terry.taylor.ext@siemens.com
2016-09-09

Automation Designer Getting Started

New September 2016 version

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

This is what the next version (second half 2017) of GS should look like (from what I understand of how the product works with minimal project experience myself). It focuses on realistic hands-on step-by-step instructions to help someone get started.

Ch 1 “Concepts” is my own way of explaining what AD is, from what I can understand with little hands-on with actual realistic customer examples).

TOC

1. Concepts (2016-09-07)

This chapter introduces basic AD concepts and how you learn them hands-on in this GS.

1.1. the goal. What is the customer trying to do? (install factory equipment (mech), wire it up (elec), program PLCs (auto)).

1.2. the problem with NO AD: 3 parties not integrated. Mech, elec, auto work separately. Manual exchange of info. Time-consuming, error-prone.

1.3. the solution With AD: 3 parties integrated. With AD, if mech adds a conveyor, then elec and auto can be created with ease. If auto changes the PLC modules, then elec can be updated with ease.

1.4. create CDs and reuse library LD_DEs / AD_EODefs (part 1, ch3-5). Create TC (database) elements you will use to create your project: (1) LD CD, (2) AD CD, (3) LD DE's (should already be in RL), and (4) AD EODefs (in RL).

1.5. associate RL conveyor DE <-> AD GL EODef (manage type mapping) (part 1, ch 6). The (3) LD DE's and (4) AD EODefs should be associated (TERRY: there needs to be some clear unique term for this … "manage type mapping" is confusing). This ensures that LD objects and corresponding AD EO's (not every AD EO corresponds to an LD object) are linked ("mapped": this is the only connection between LD and AD).

1.6. create LD (mechanical design) (part 2, ch 7). with 2 conveyors. Simple.

1.7. create AD EO aspect tree (part 2, ch 8). for 1 conveyor. this tree models the functional structure of the entire plant, with detail above and below the LD DE conveyor (includes conveyors subcomponents). This tree is then used to create the globally unique ID's used in elec (EPLAN) and auto (TIA).

1.8. create EPLAN (electrical design) (part 2, ch 9). import macro, manually set variables (don't use expressions to get values from the aspect tree yet, because its too complex; this is OK for only 1 conveyor), and generate report.

1.9. create TIA (automation design) (part 2, ch 10). Import SW-tags. Fix the SW (dynamize). manually set SW-tags names (don't use expressions to get values from the aspect tree yet, because its too complex; this is OK for only 1 conveyor), and send SW-tags to TIA.

1.10. build the factory floor. you now have mech (LD), elec (EPLAN), and auto (TIA). You could build factory now.

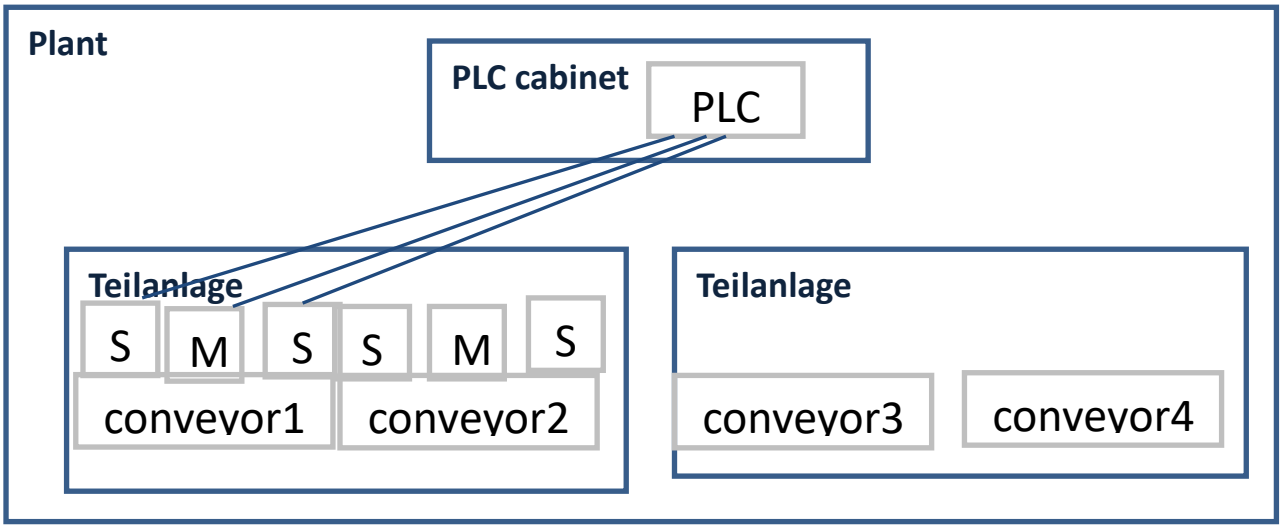
1.11. synch changes (no templates) (part 2, ch 11). not sure about this section, what to write. But should demo how changes in LD and TIA can easily be propagated to LD, TIA, and EPLAN.

1.12. templates (part 3, ch 12-16). Now you need to (1) use expressions to create unique IDs for EPLAN and TIA based on the aspect tree. Then (2) create a template of your AD conveyor EO (which includes TIA and EPLAN), and store in the RL. Then (3) add a conveyors in LD and (4) instantiate templates in the EO aspect tree for the added conveyors. Steps (3) and (4) are quick and easy (or will be in a future version), and demonstrate why you went to all the trouble to do all this.

1.1. the goal

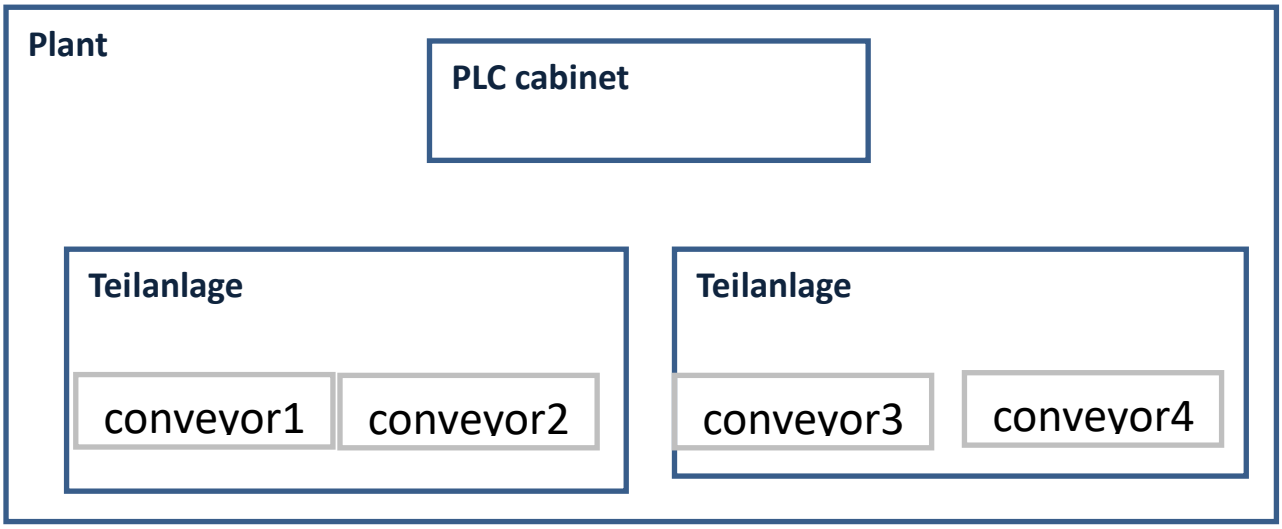
What is the customer trying to do? (install factory equipment (mech), wire it up (elec), program PLCs (auto)).

1_06

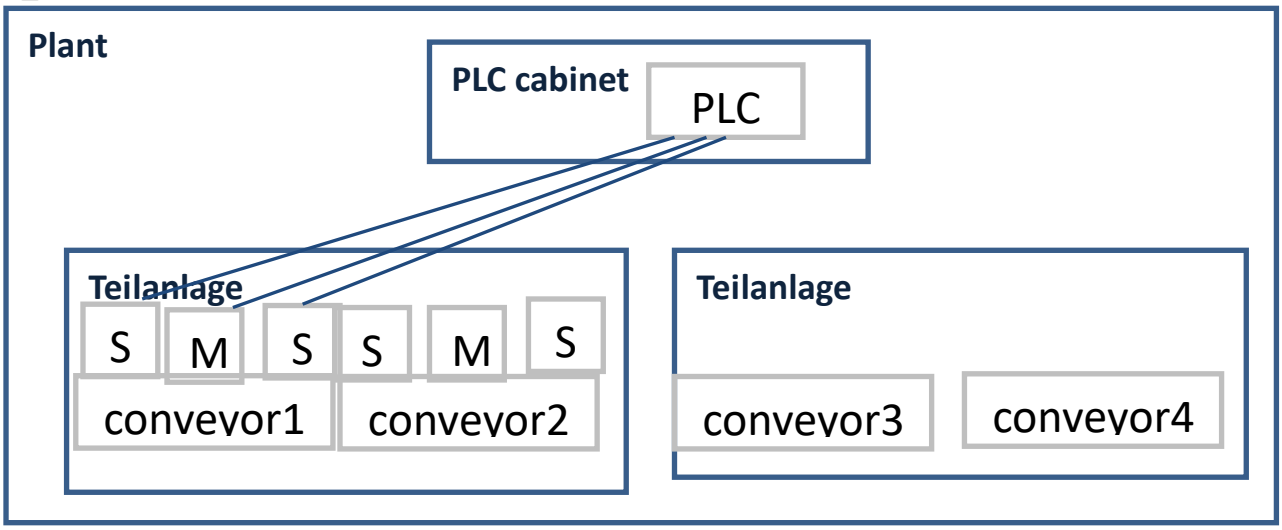


Need to do 3 design processes
1. mechanical
2. electrical wiring (EPLAN)
3. automation (TIA, PLC programming).

Mechanical (LD) designs mechanical plant.
1_02



Electrical (EPLAN) defines wiring
1_04



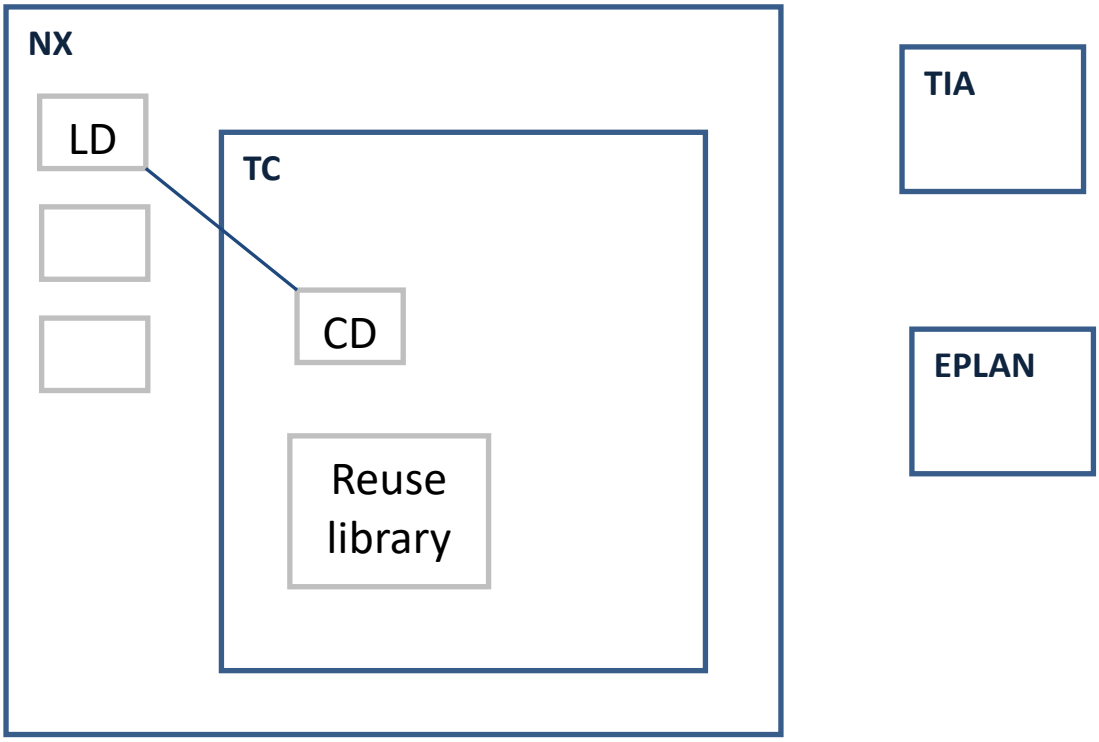
Automation (TIA) creates PLC software.
1_05

1.2. the problem with NO AD: 3 parties not integrated

Mech, elec, auto work separately. Manual exchange of info. Time-consuming, error-prone.

All 3 parties (LD, TIA, EPLAN) work separately.

1_01

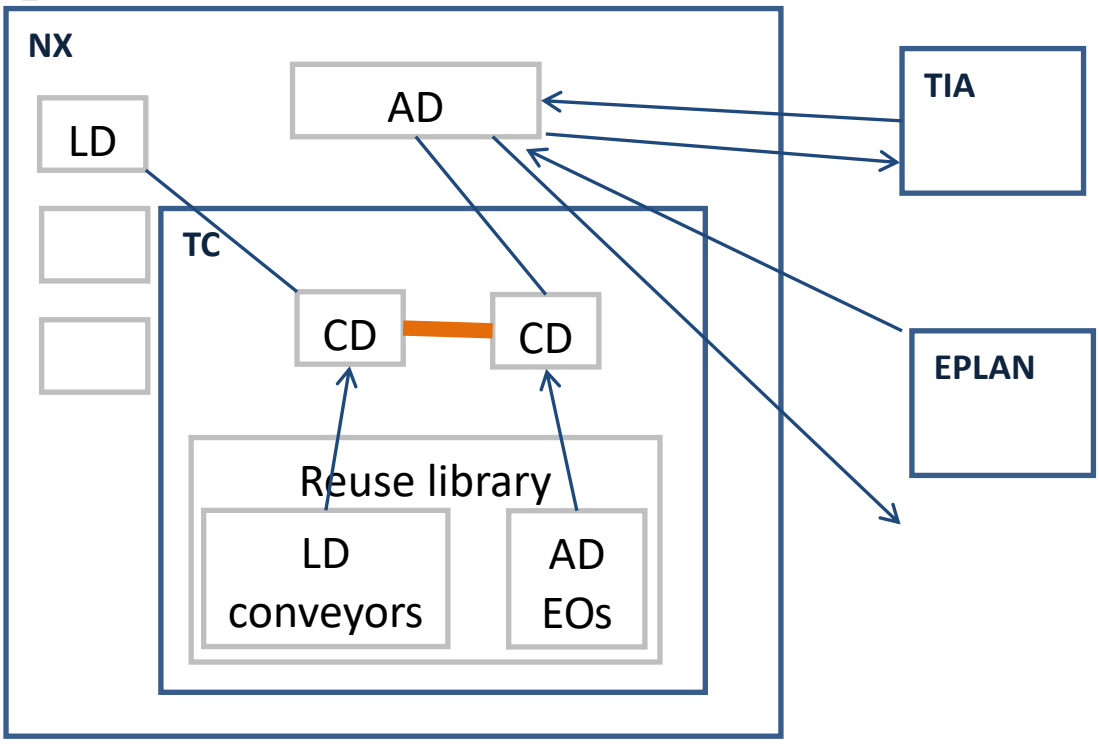


main problems:
1. disconnected design processes for all 3 (mech, elec, auto) .
2. no detailed EO model of structure of plant (super- and sub-conveyor engineering objects). Only what LD creates.
3. no unique naming for electrical (eplan wiring).
4. no unique names for software (TIA).

1.3. the solution With AD: 3 parties integrated

With AD, if mech adds a conveyor, then elec and auto can be created with ease. If auto changes the PLC modules, then elec can be updated with ease.

1_07



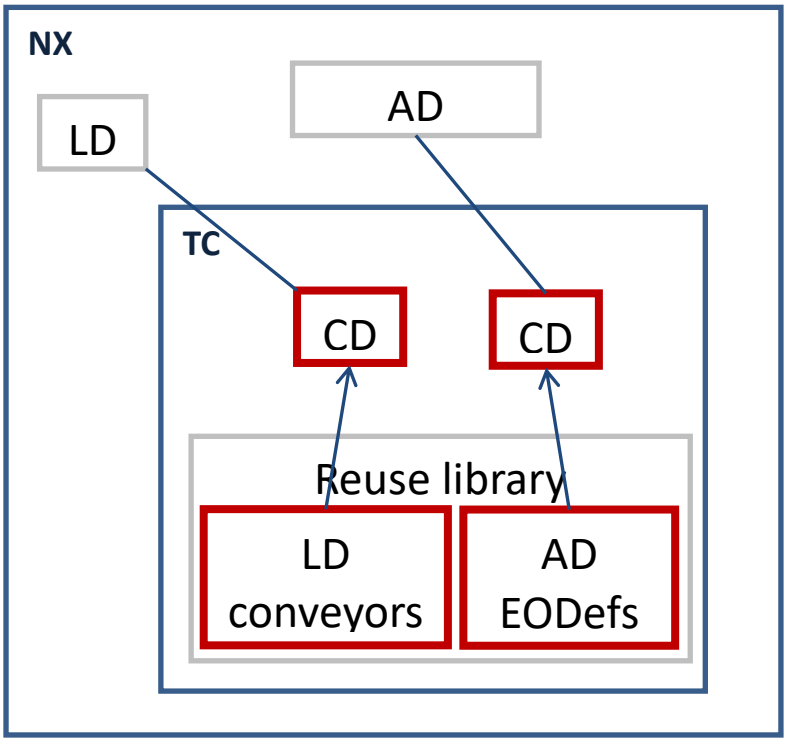
Solved main problems:

1. connected design processes for all 3 (mech, elec, auto).
2. created more detailed EO model of structure of plant (super- and sub-conveyor engineering objects) than what LD creates.
3. unique naming for electrical (eplan wiring) based on EO model.
4. unique names for software (TIA) based on EO model.

1.4. create CDs and reuse library LD_DEs / AD_EODefs (part 1, ch3-5)

In this GS Part 1 do some configuration in TC.
Ch3-4. create LD CD (and WS, SS, etc.).
Ch5. create AD CD (and WS, SS).
Ch5. create EODefs and naming rules.
x. assume LD conveyors already in reuse library.

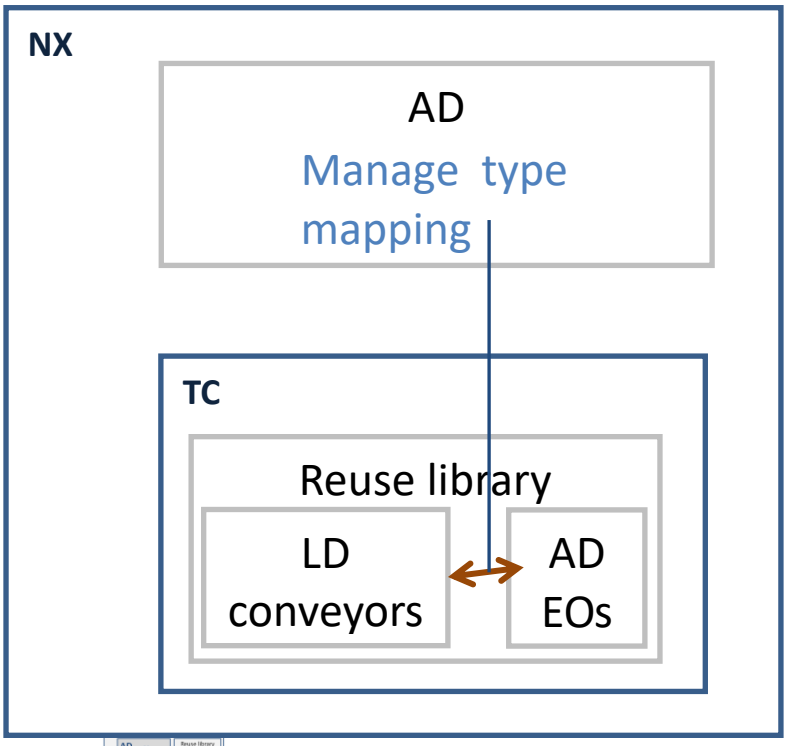
1_22



1.5. associate RL conveyor DE <-> AD GL EODef (manage type mapping) (part 1, ch 6)

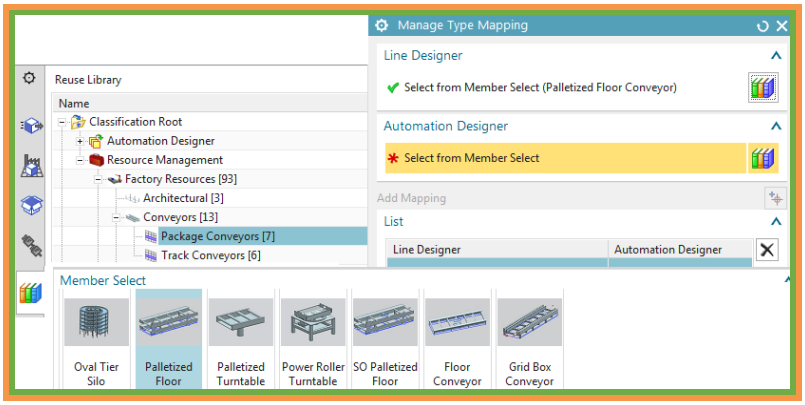
The LD DE's and AD EODefs should be associated (TERRY: there needs to be some clear unique term for this … "manage type mapping" is confusing). This ensures that LD objects and corresponding AD EO's (not every AD EO corresponds to an LD object) are linked ("mapped": this is the only connection between LD and AD).

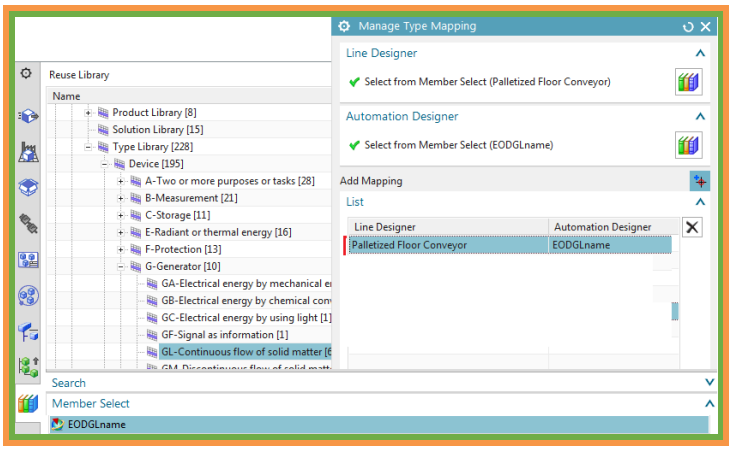
1_09



Old …

You specify which AD EOs represent which LD factory resources.



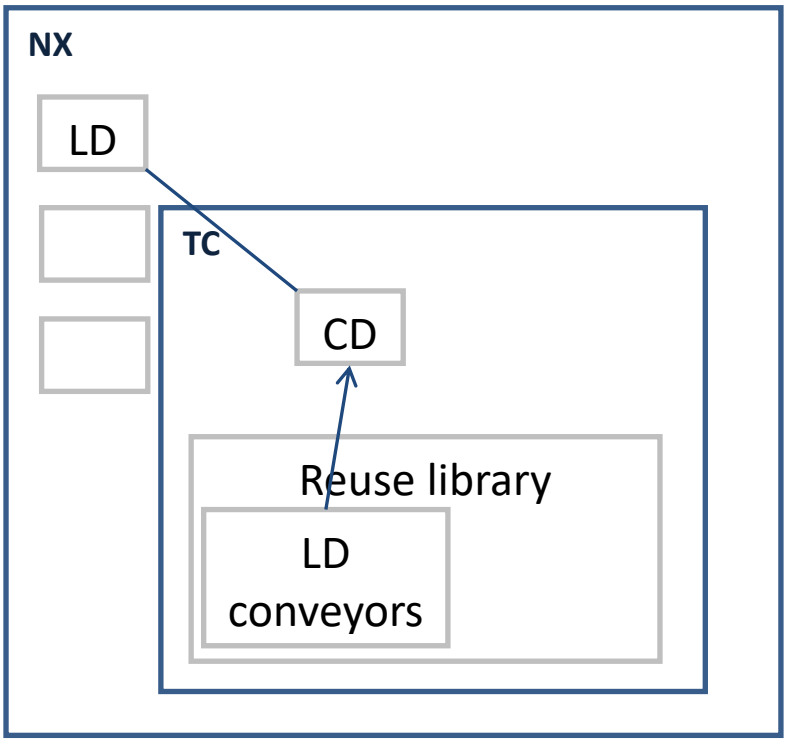


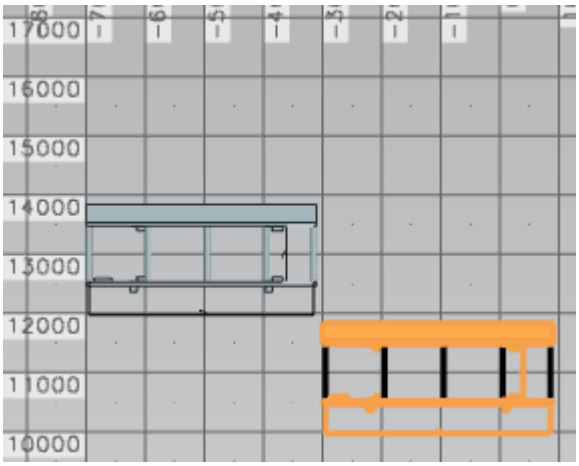
1.6. create LD (mechanical design) (part 2, ch 7)

TERRY: do this before create AD EO tree, because then conveyors show up in map to new dialog.

In ch7 you create LD design stored in LD CD.

1_10

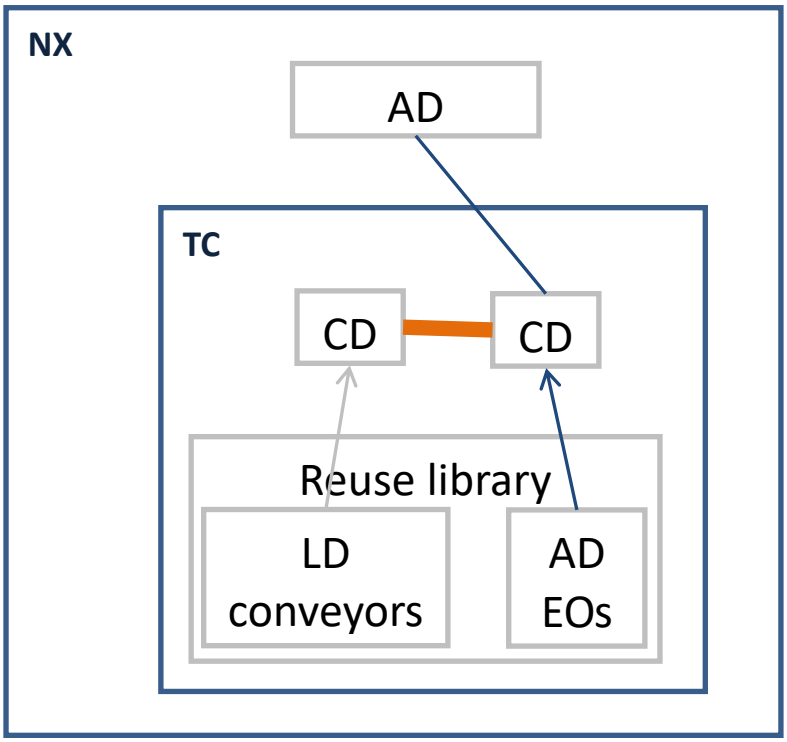




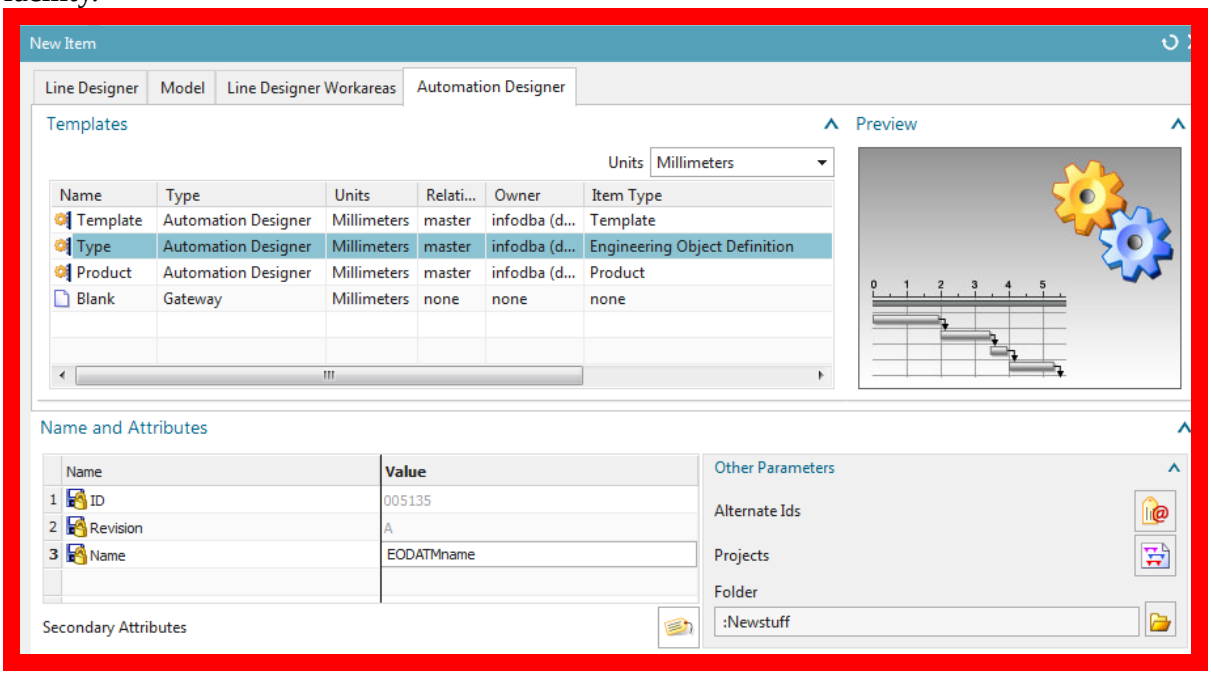
1.7. create AD EO aspect tree (part 2, ch 8)

In ch 8 You create AD aspect tree (stored in AD CD) for 1 conveyor. this tree models the functional structure of the entire plant, with detail above and below the LD DE conveyor (includes conveyors subcomponents). This tree is then used to create the globally unique ID's used in elec (EPLAN) and auto (TIA). The designs are linked (orange line in diagram below), so that changes on LD or AD side can be noticed by the other side.

1_11

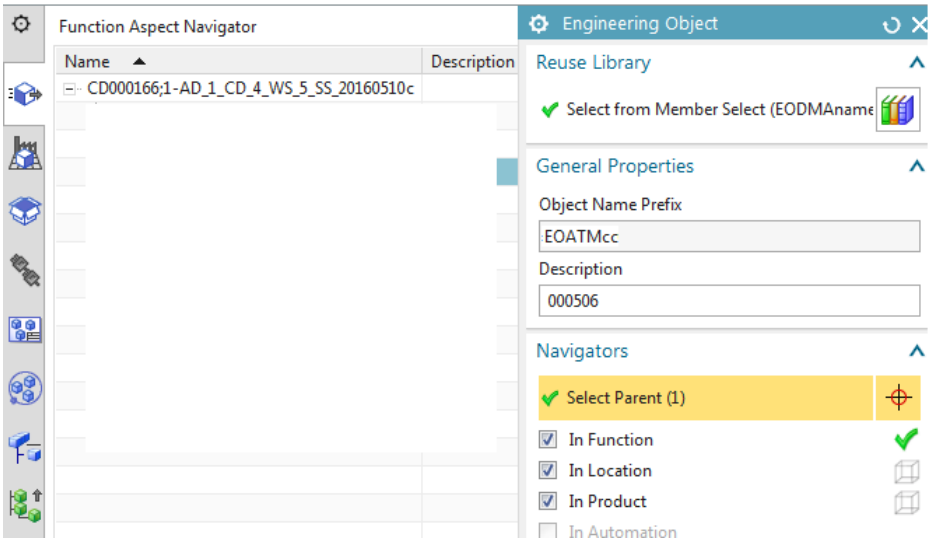


NOTE: Naming rules (created in ch5 when you created EODefs) depend on conventions for the particular facility.

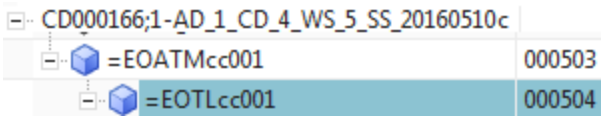


1.7.1. add top level EOs

Add the EOs above the conveyor.



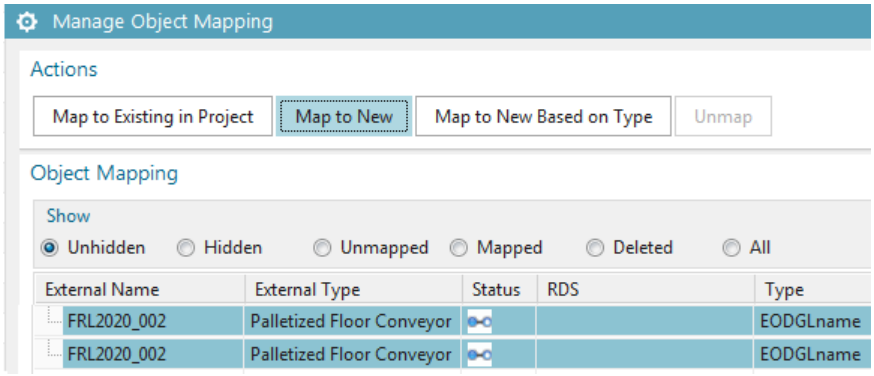
ATM is the facility, TL is teilanlage.



1.7.2. Map to new (add EO that corresponds to CONVEYOR)

The mapping you set up a few sections earlier (“4b. associate RL conveyors <-> AD EODefs (manage type mapping)”).

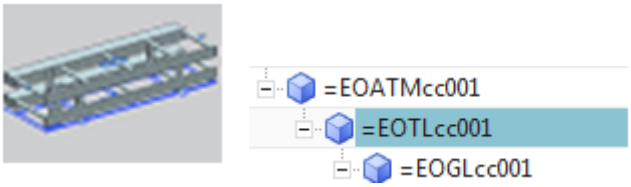
2. Click **Map to new**. You see the LD objects not mapped to anything in AD. But the type mapping is set to EODGLname (the terminology is quite confusing… object mapping should be called “linking” or something less confusing).



3. Select the Engineering Object from reuse library

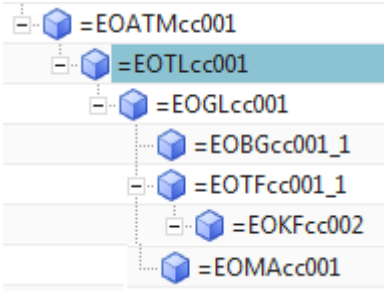
4. For the parent select TL.

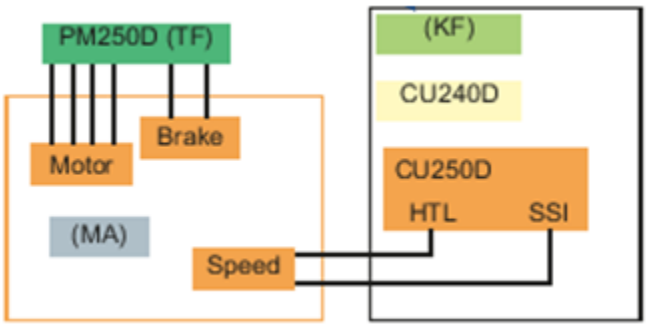
5. Click **OK**. A new Engineering Object is created and mapped to the conveyor. That represents the conveyor.



1.7.3. add sub-conveyor EOs

Add rest of EOs, creating an aspect chain that reflects structure of conveyor.

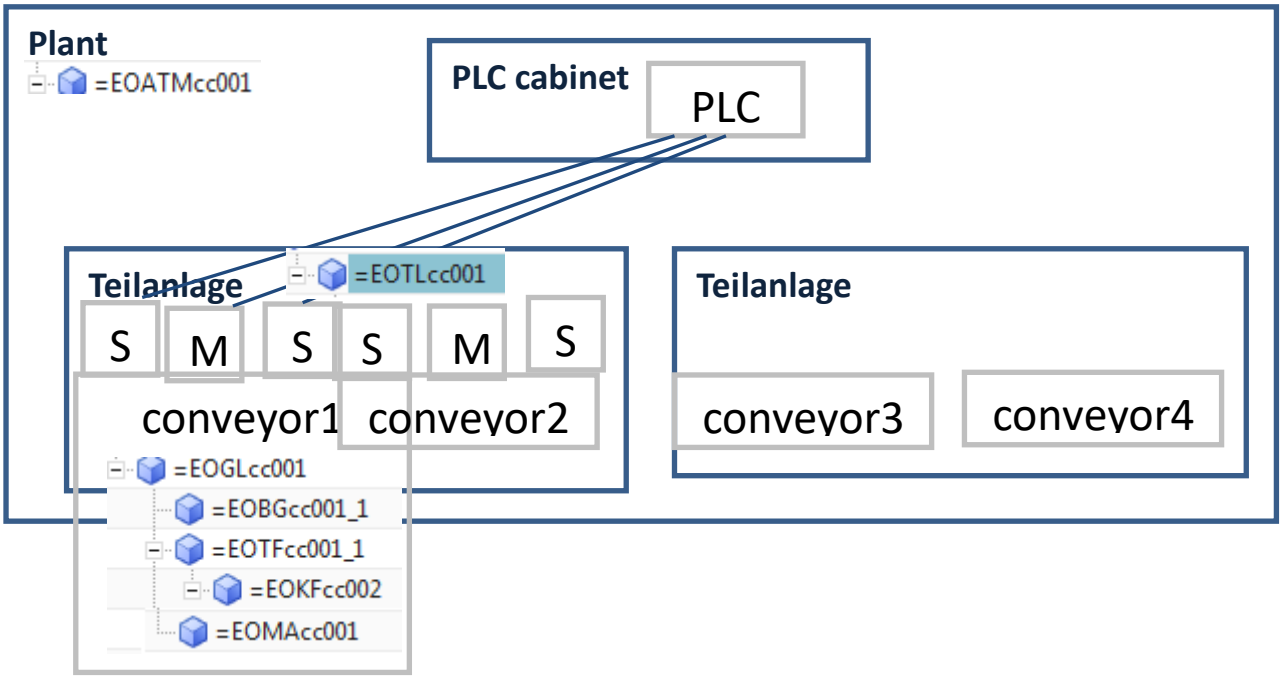




1.7.4. result

Now you have a model with more detailed structure than in LD.
This can now be used for electrical (EPLAN) and auto (TIA).

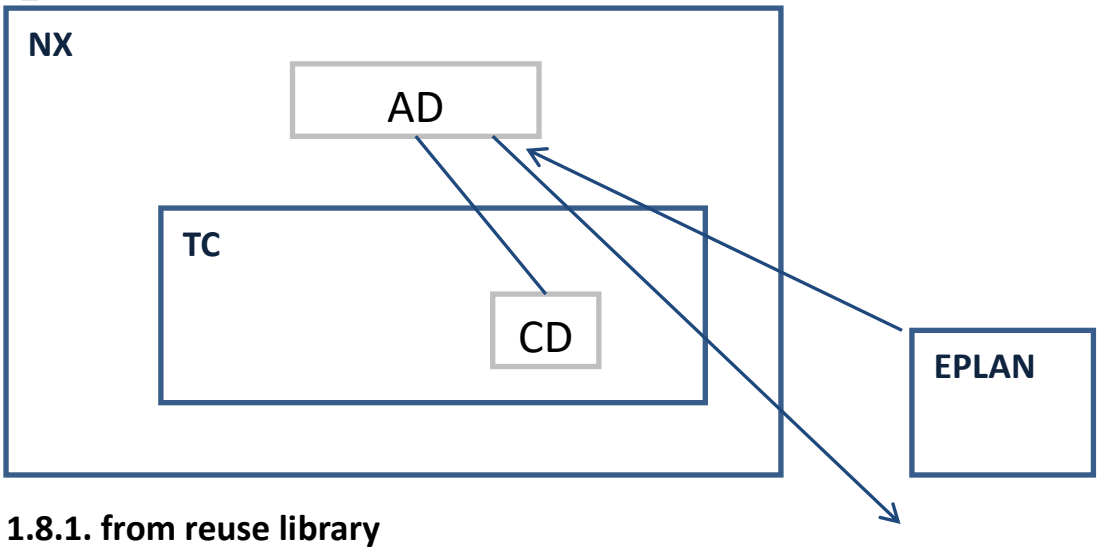
1_08



1.8. create EPLAN (electrical design) (part 2, ch 9)

import macro, manually set variables (don't use expressions to get values from the aspect tree yet, because its too complex; this is OK for only 1 conveyor), and generate report.

1_12



1.8.1. from reuse library

Best way….

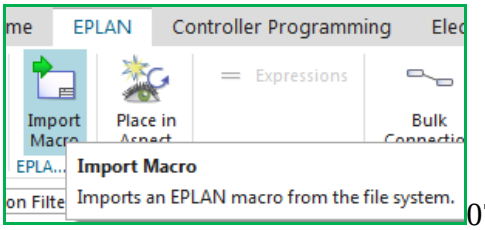
(TERRY: should demo first just using RL, but I did not have this setup)

1.8.2. OR create yourself / no expressions (no aspect chain names) (ch9)

…. but this GS shows how to do yourself.

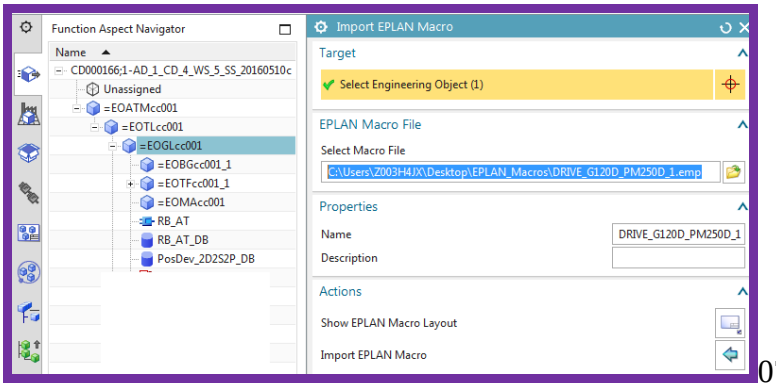
In part1 ch9 you create with no expressions.

1. Click on **Import Macro**.



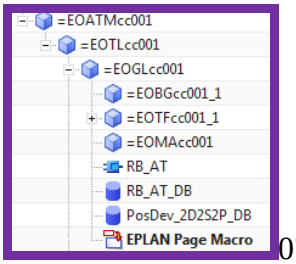
07b_05

2. Import DRIVE_G120D_PM250D_1.emp under Engineering Object GL.



07b_06

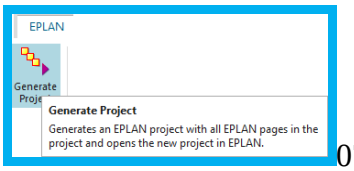
3. Click Close. The macro appears in the aspect tree.



07b_07

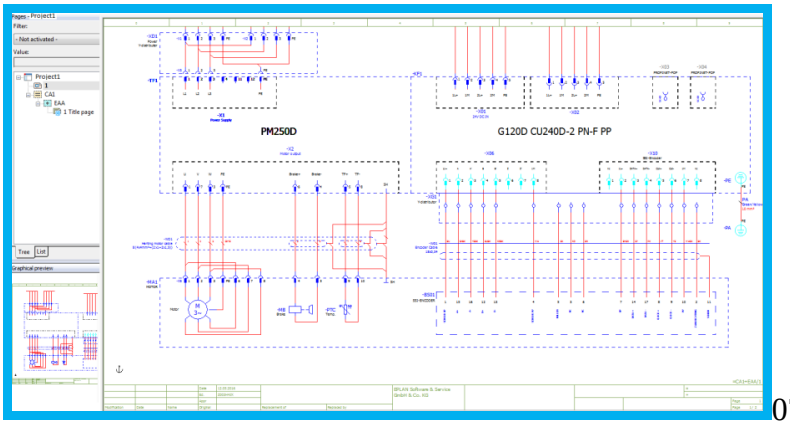
1.8.3. generate EPLAN reports

1. Click Generate Project.



07b_10

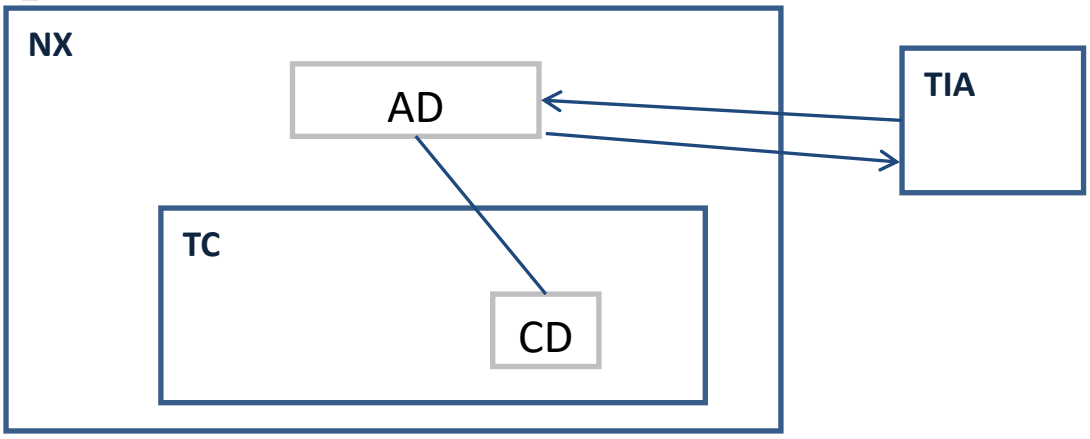
3. Click **Generate**. The project is opened in EPLAN.



07b_13

1.9. create TIA (automation design) (part 2, ch 10)

Import SW-tags. Fix the SW (dynamize). manually set SW-tags names (don't use expressions to get values from the aspect tree yet, because its too complex; this is OK for only 1 conveyor), and send SW-tags to TIA.
1_13?



1.9.1. from reuse library

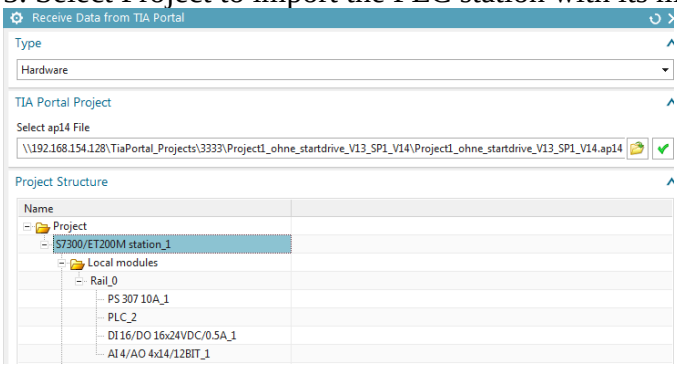
Best way….
(TERRY: should demo first just using RL, but I did not have this setup)

1.9.2. OR create yourself / no expressions (no aspect chain names) (ch10)

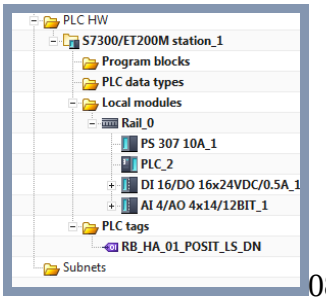
….but this GS shows how to do yourself.
In part1 ch10 you create with no expressions.

1. Import HW/SW

HW
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.

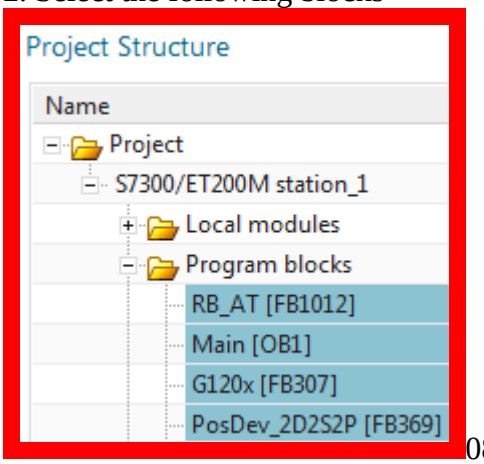


6. Click **Receive from TIA Portal**. The station is imported.



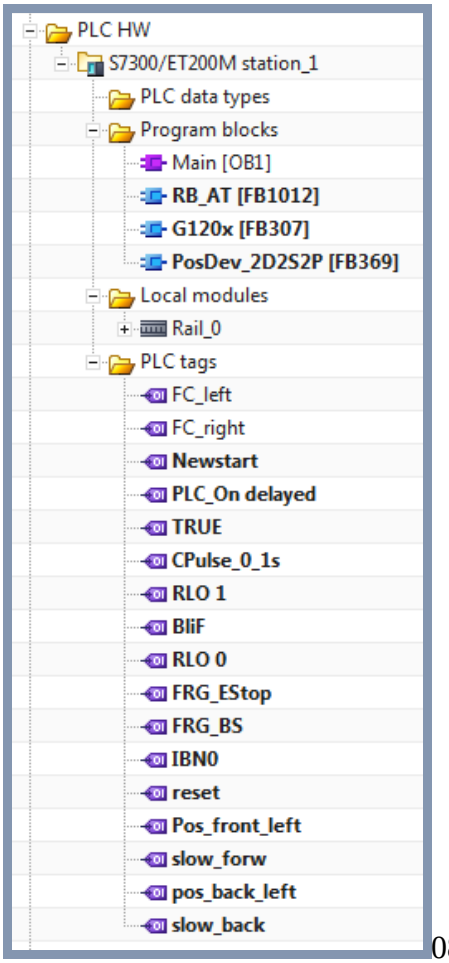
SW

1. For **Type** select software.
2. Select the following blocks



08c_01

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.

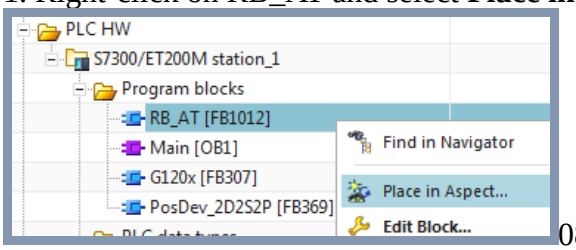


08b_03

2. place SW in the aspect, create IDBs

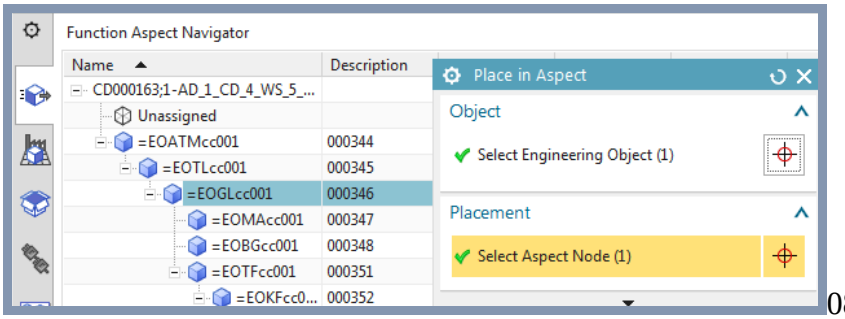
You now need to place the function blocks in the aspects and create IDBs in the aspects. Where you place the function blocks determines the aspect chain that will be used to create unique indentifiers (symbolic names). In this Getting Started you focus only on the Function aspect.

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



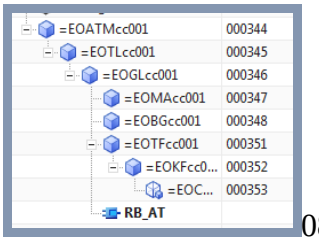
08b_04

2. For Placement select GL.



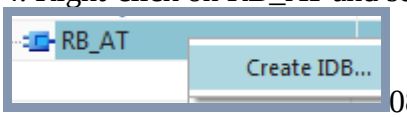
08b_05

3. Click **OK**.



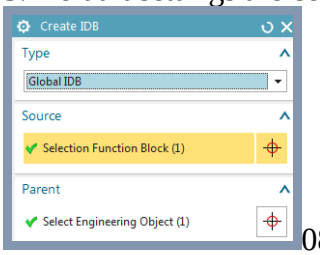
08b_06

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



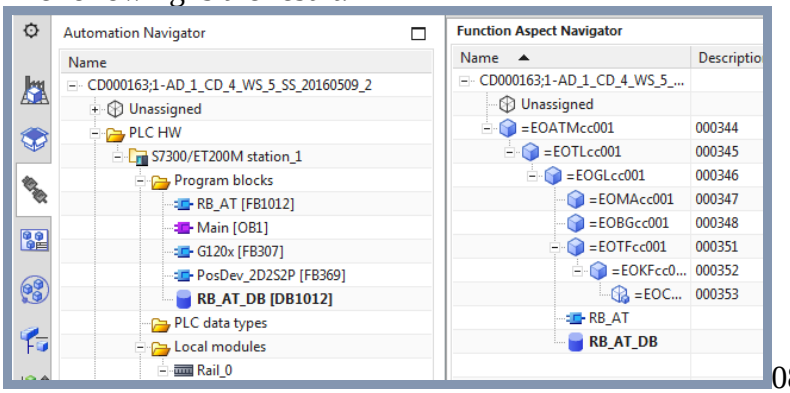
08b_07

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



08b_08

The following is the result.



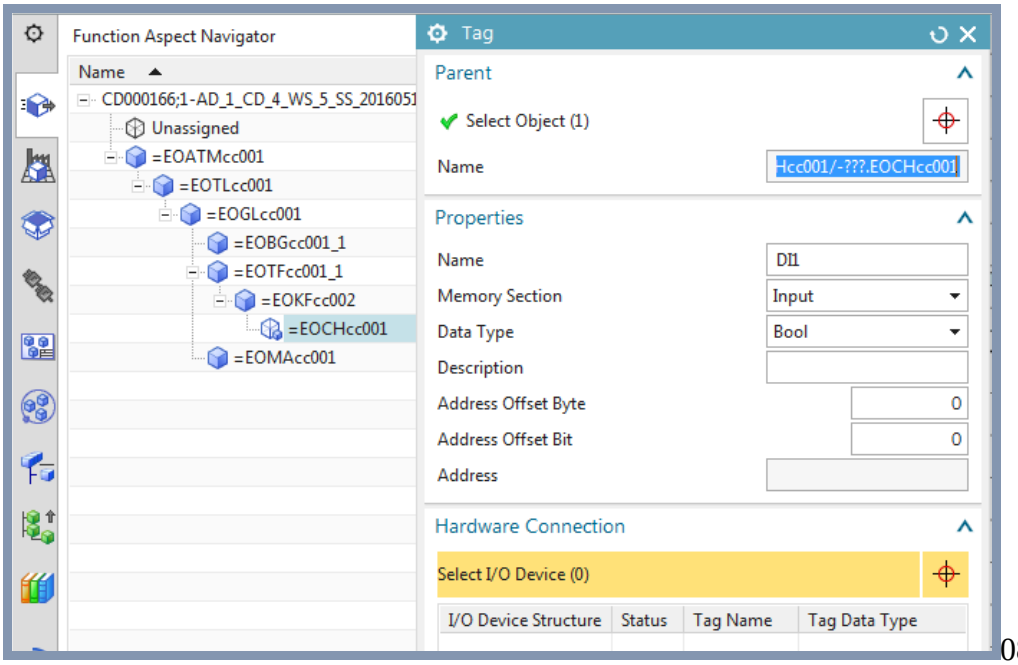
08b_09

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

3. place / add tags (assign to I/O device)

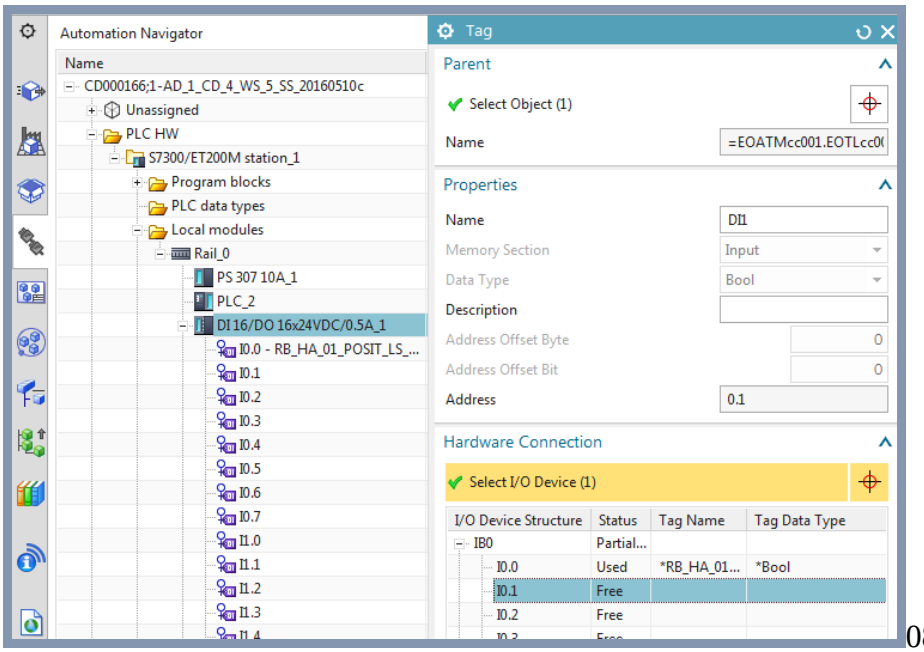
Add the sensor tag DI1 (boolean input).

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



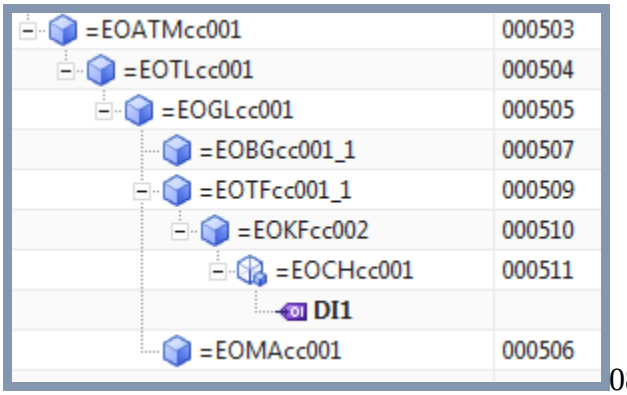
08b_11

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.



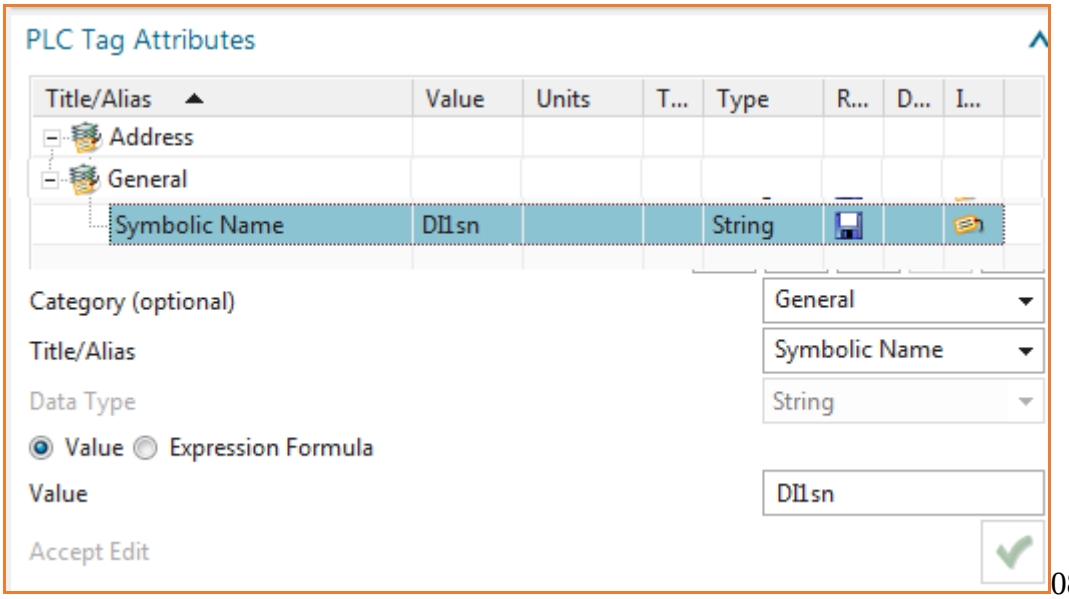
08b_12

6. Click **OK**.



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).



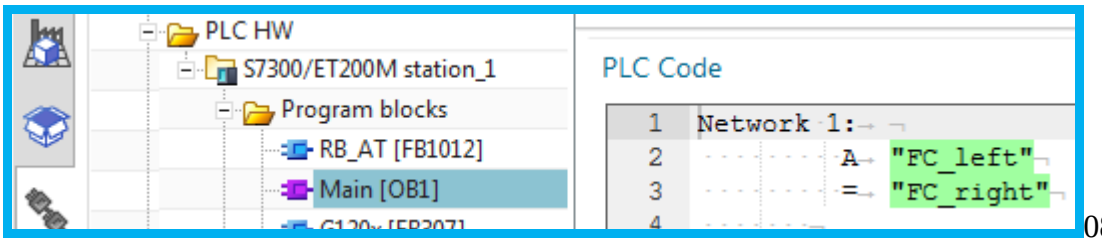
08b_14

4. dynamize (fix the calls of imported sw)

Fix anything not green (here is green, but fix anyway).

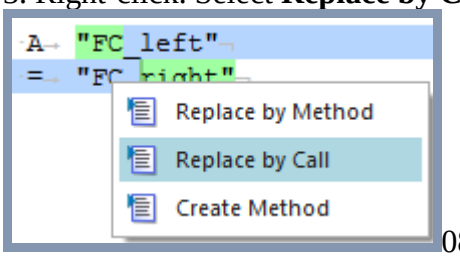
10.5.1. OB1->RB_AT_DB replace by call

OB1 calls the RB_AT IDB.
1. Double-click on **OB1**.



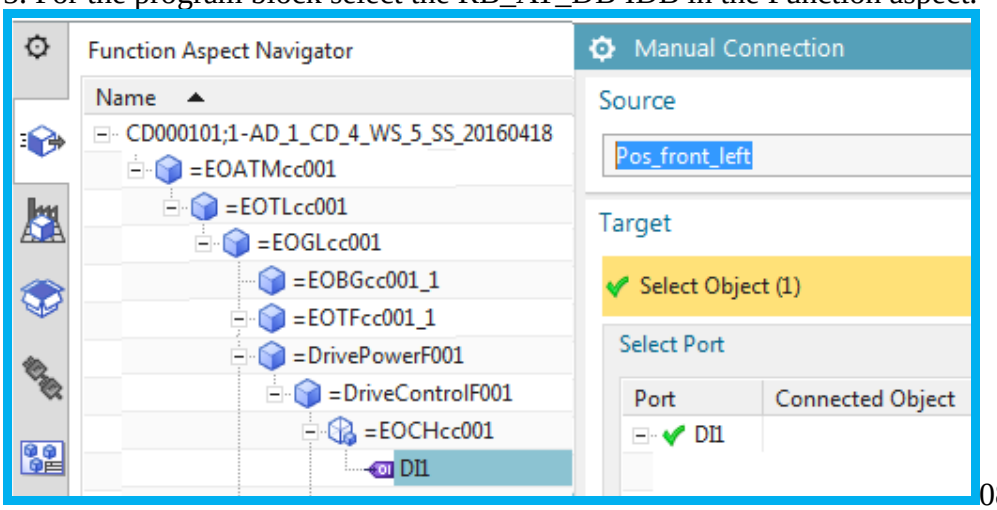
08b_22

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



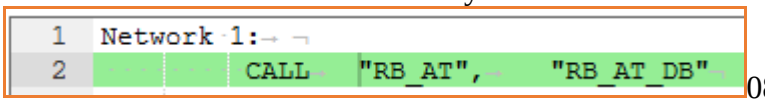
08b_23

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



08b_24

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

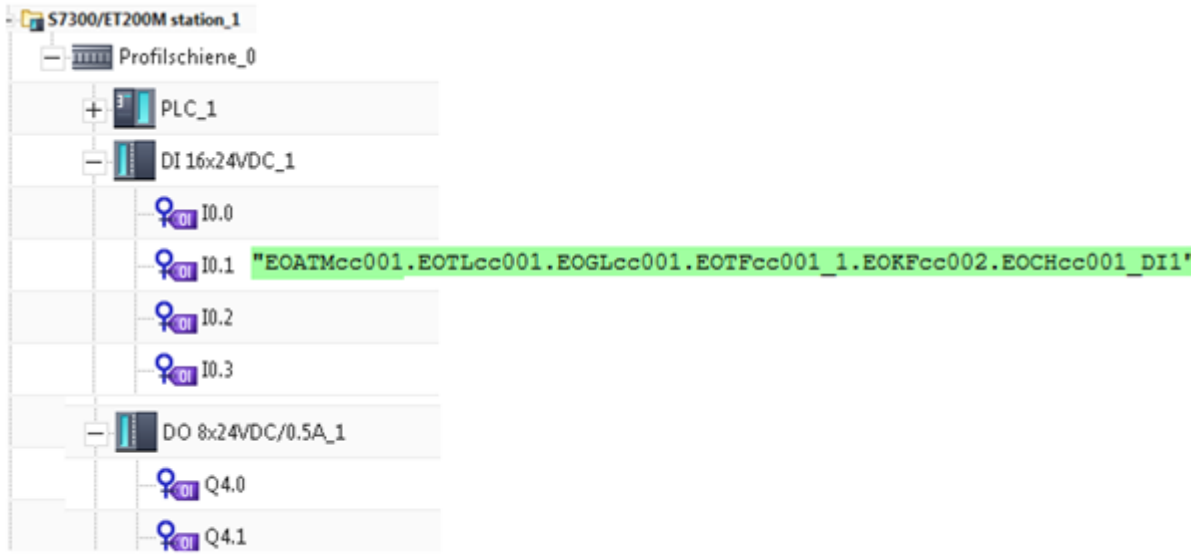


08b_25

1.9.3. assign sw-tags to hw (connect sw)

1. Select **Bulk Connection**.
2. For **Source** select **GL**.
3. Select the ports.
4. Under **Target** select **Select Object**.
5. Select the station.
6. Select the target ports.
8. Click **OK**.

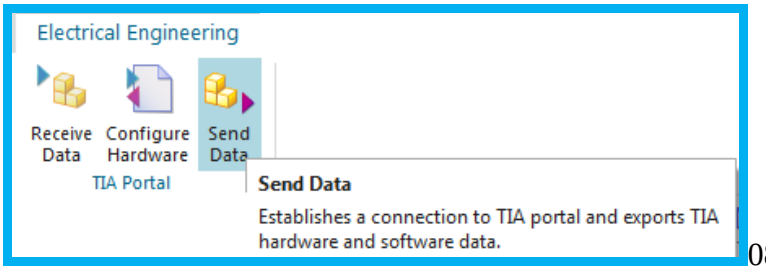
The following shows the result.



08b_54

1.9.4. export hw/sw ad->tia

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

NEED PIC OF SW IN TIA with aspect names.

1.10. build the factory floor

You now have

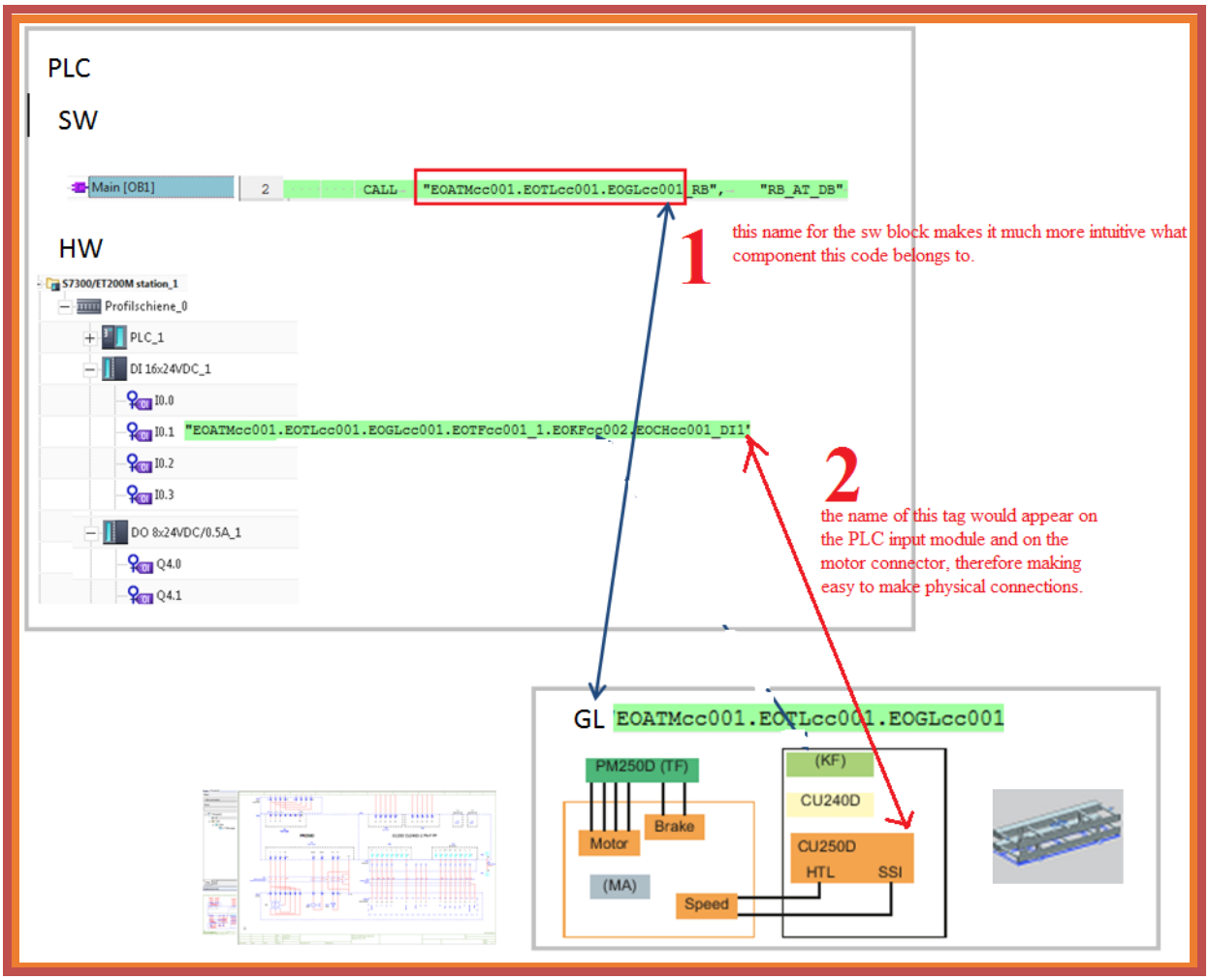
1. Mechanical (LD)
2. automation (TIA)
3. electrical (EPLAN)

So… You could build factory now.

1. mechanical builds the line (including PLCs).
2. Electrical connects line equipment.
3. Automation programs the PLCs.

This is all made much easier because you used AD: Following shows how

1. PLC SW-tag names match unique IDs of equipment on factory floor (understand code much easier).
2. PLC in/out panel names match those marked on equipment (GL) (marked when electrics connected).



If any changes then easy to propagate between LD, TIA and EPLAN.

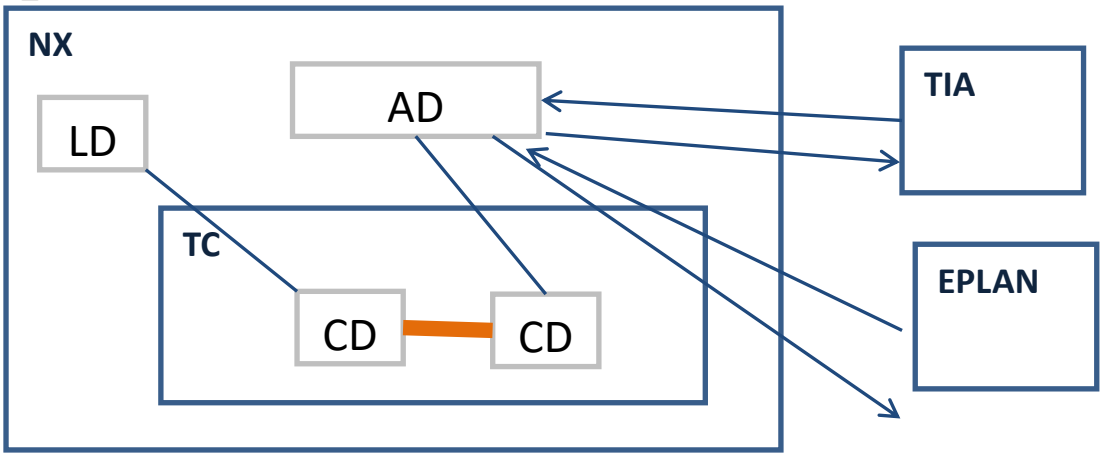
Note: you could add TIA and EPLAN for new conveyors. Simply copy and paste the conveyor EO (and all sub-EOs). But the unique IDs you would have to manually change. This you fix with templates (later).

1.11. synch changes (no templates) (part 2, ch 11)

not sure about this section, what to write. But should demo how changes in LD and TIA can easily be propagated to LD, TIA, and EPLAN.

If make changes Then
1. make sure LD-AD mapping complete.
2. export to TIA again.
3. generate EPLAN reports again.

1_14?



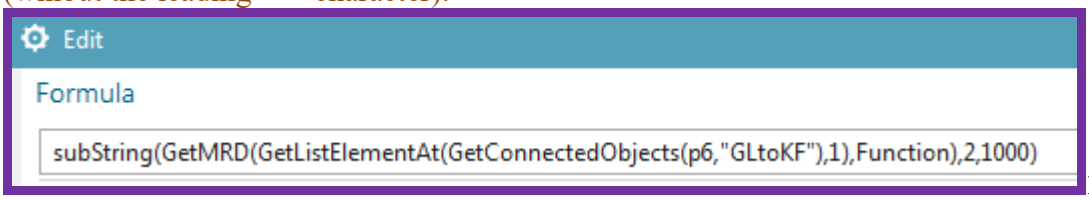
1.12. templates (part 3, ch 12-16)

Now you need to
(1) use expressions to create unique IDs for EPLAN and TIA based on the aspect tree.
(2) create a template of your AD conveyor EO (which includes TIA and EPLAN), and store in the RL.
(3) add a conveyors in LD and
(4) instantiate templates in the EO aspect tree for the added conveyors.

Steps (3) and (4) are quick and easy (or will be in a future version), and demonstrate why you went to all the trouble to do all this.

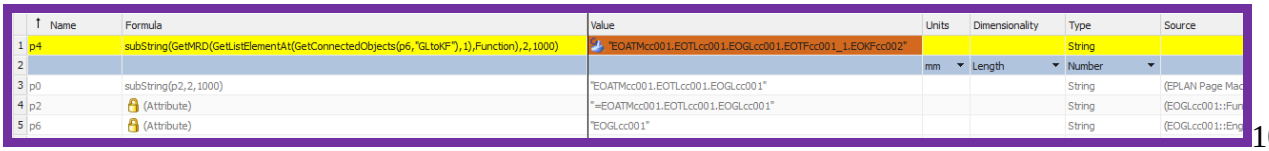
1.12.1. Make EPLAN template ready (use expressions for unique aspect chain names) (ch13)

In part2 ch13 you add expressions, which make your project template-ready.
1. Open the properties for the EPLAN macro.
2. For **Apply to** select **EPLAN page**.
3. Click on Power Module Function Text.
4. Click **Expression Formula**.
5. Click on the arrow for **Expression Formula**. A drop-down list appears.
6. Click **Formula**. The Expressions dialog appears.
7. Under **Formula** right-click and select **Edit**.
8. Enter “subString(GetMulti Reference Designation(GetListElementAt(GetConnectedObjects(“.
9. Click on **Reference Object Attribute**.
10. Select the conveyor **Object name**.
11. Complete expression with “,"GLtoKF"),1),Function),2,1000)”. This expression gets the connected objects at port GLtoKF, gets the list elements at that port, then gets the MRD, then returns the substring (wihout the leading “=” character).



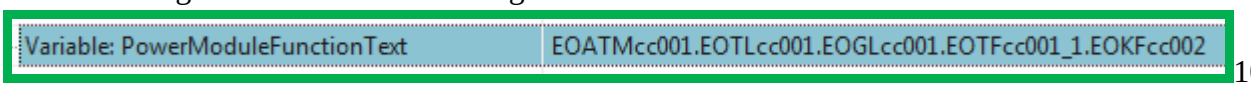
10b_17

12. Click **OK**.



10b_18

13. Click **OK**.
14. Click the green arrow. The following shows the result.



10b_20

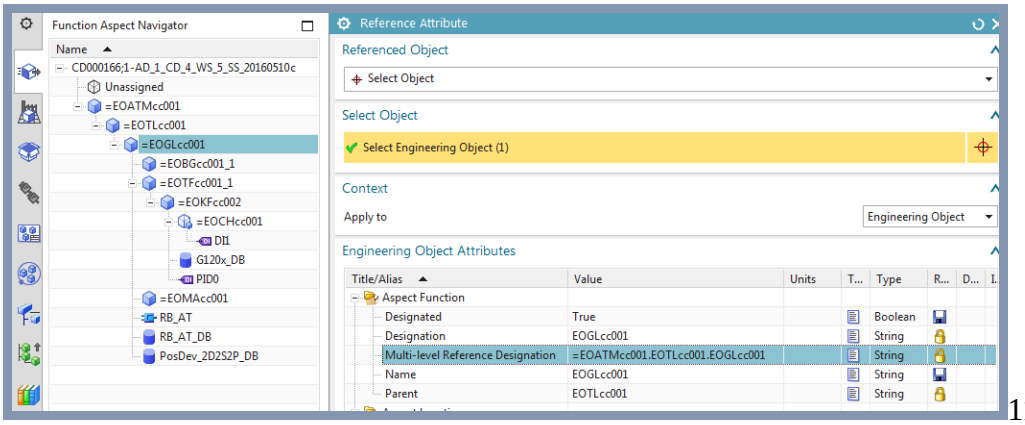
15. Click **OK**.

1.12.2. Make TIA template ready (use expressions for unique aspect chain names) (ch14)

In part2 ch14 you add expressions, which make your project template-ready.

SW

1. Open RB_AT properties.
2. Select **Symbolic Name**.
3. Click **Expression Formula**.
4. Right-click on the drop-down box and select **Formula**.
8. Select the GL Function aspect **Multi Reference Designation**.



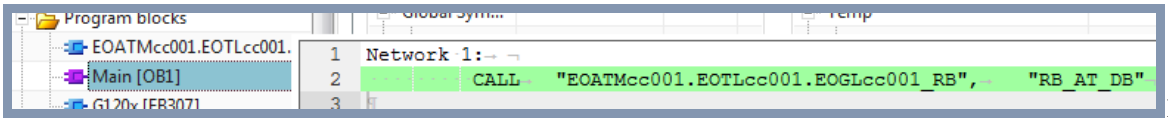
11b_06

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

11b_09

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

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

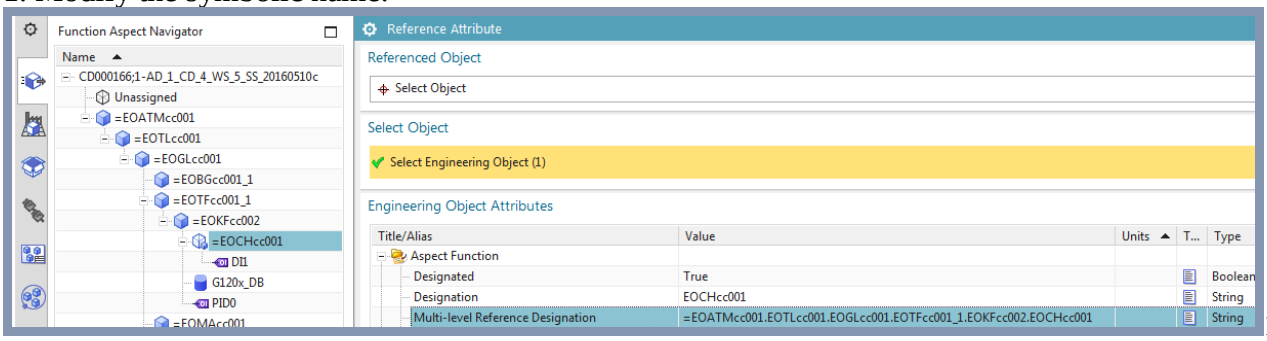


11b_12

TAGS

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.

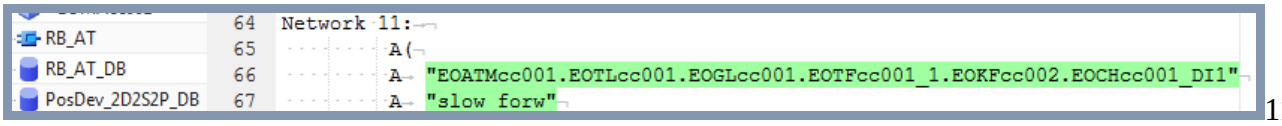
1. Modify the symbolic name.



11b_23

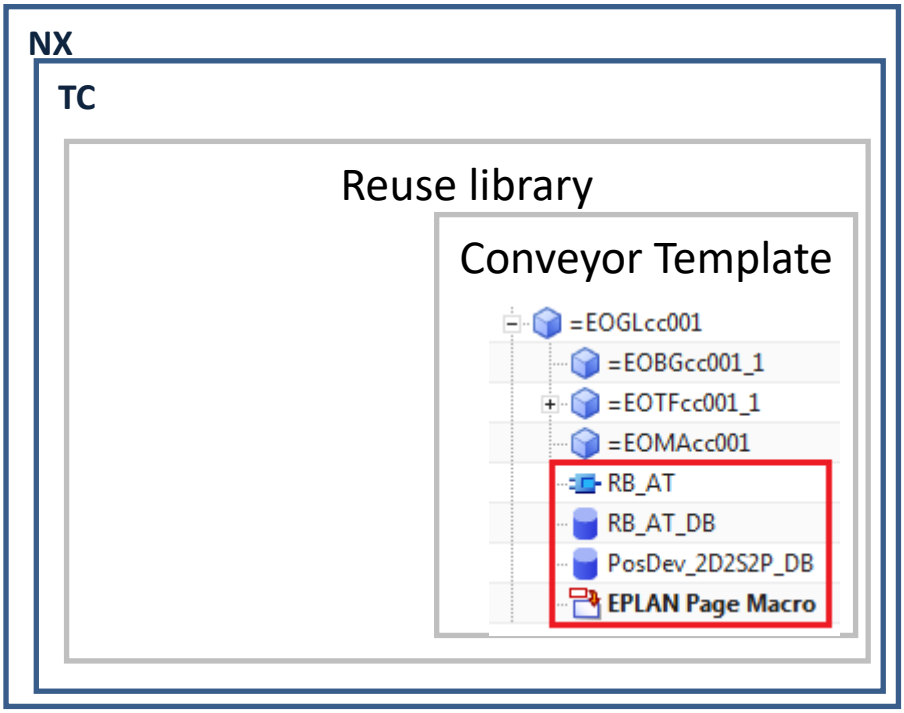
11b_24

The following shows the result.

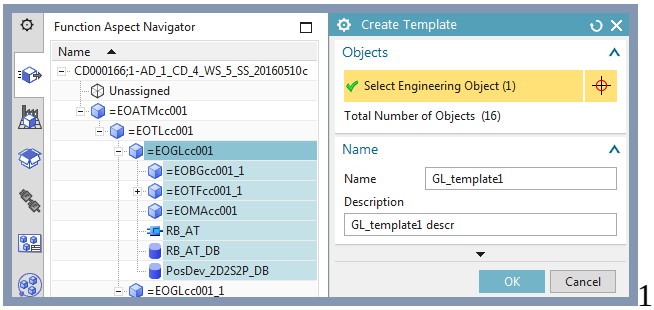


11b_25

1.12.3. Create template / add to RL (15.1)

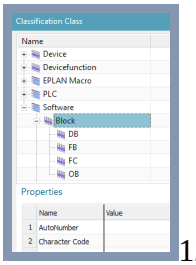


1. Select GL01. Click **System Design→Create Template**.
2. For **Name** enter **GL_Template**.



12b_01

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

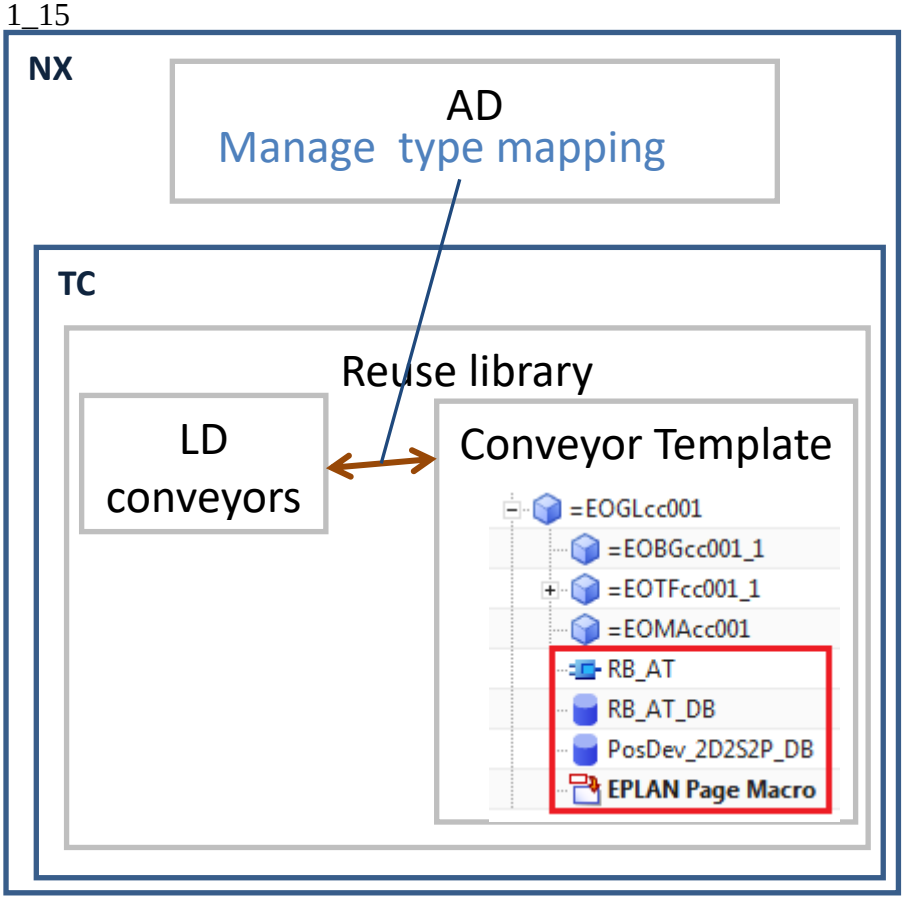


12b_03

6. Click **OK**. You are now in the template editor.
7. Choose **File→Close→Close template**.
8. Choose **Yes – Save and Exit**.

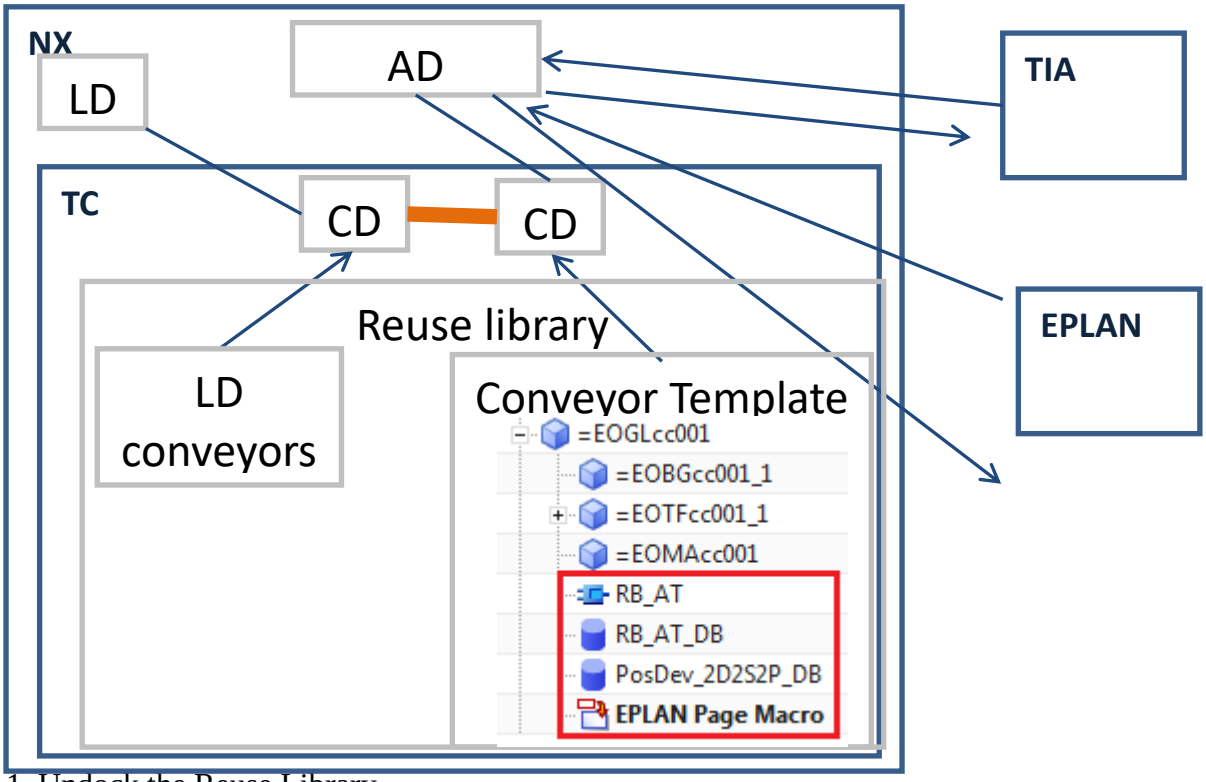
1.12.4. type mapping to template (15.2)

1_15

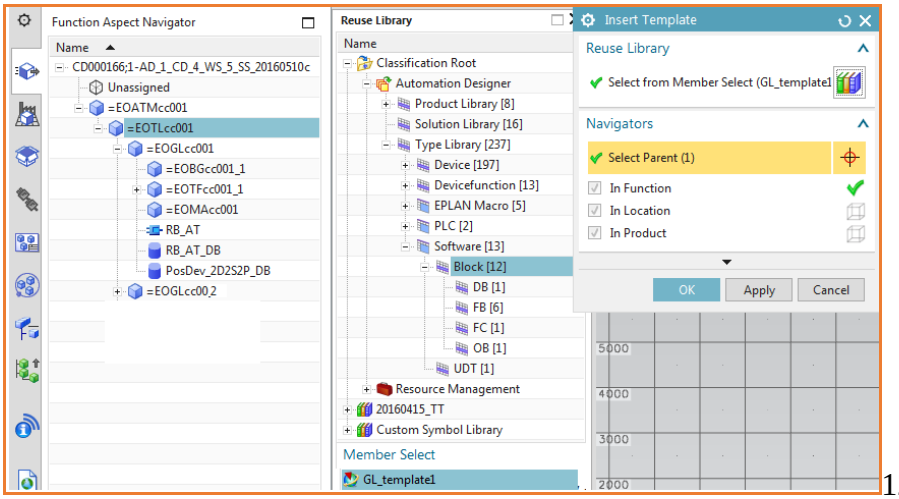


1.12.5. Insert template (15.3)

1_16

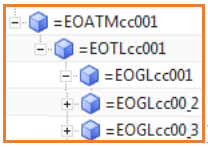


1. Undock the Reuse Library.
2. Drag & drop the template. The **Insert Template** dialog appears.
3. For **Parent** select **Engineering Object TL**.



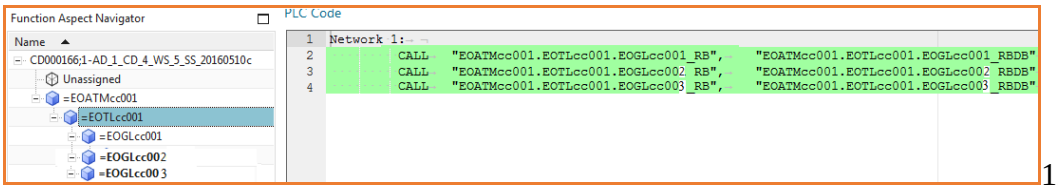
12b_06xxxxx

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.



12b_17b

1.12.6. Synch changes (templates) (ch16)

I just wrote a half page, not sure how to do.

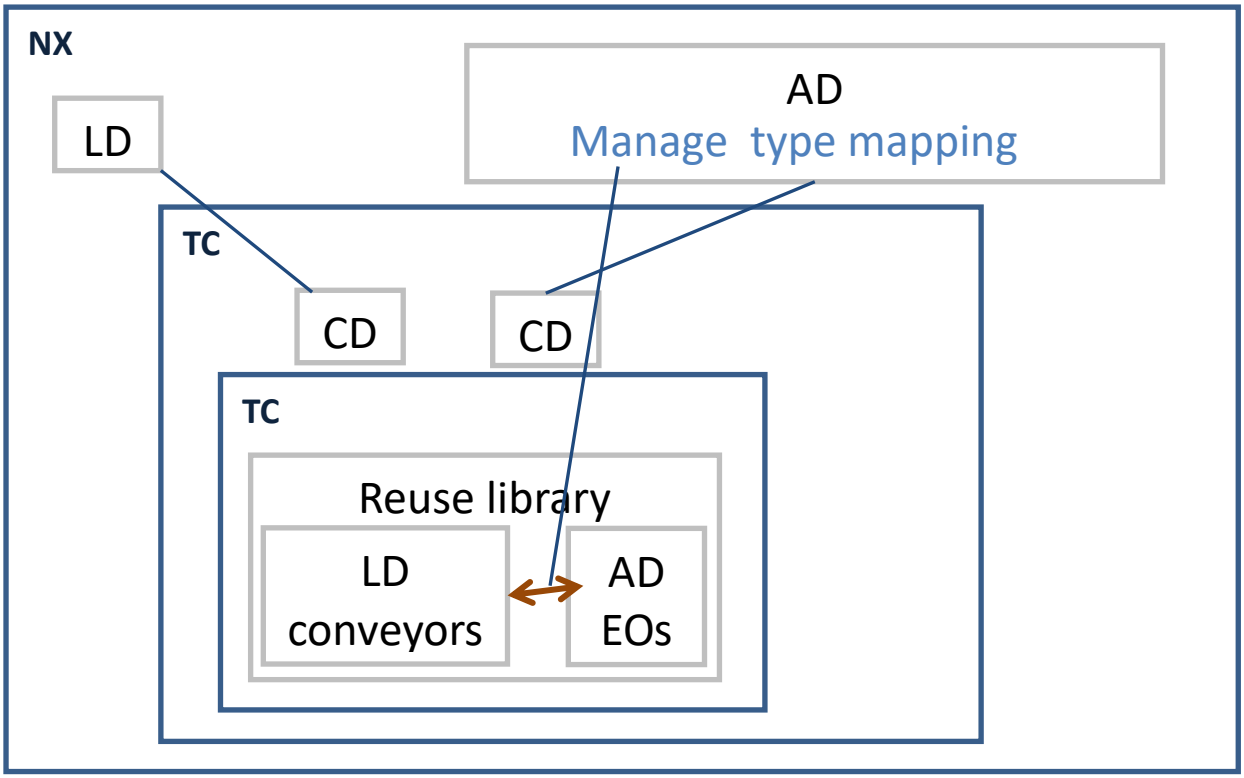
2. overview of this GS (new, 2016-09-07)

in this GS you start from clean system, create all yourself.
Reuse library does not have templates, SW, HW, EPLAN.

(TERRY: GS should start out with typical system that has these in reuse library, but I never knew how to set this up.. this GS is based on the automotive example which is all I had to work with).

Part 1: Create CDs, Eodefs, map.

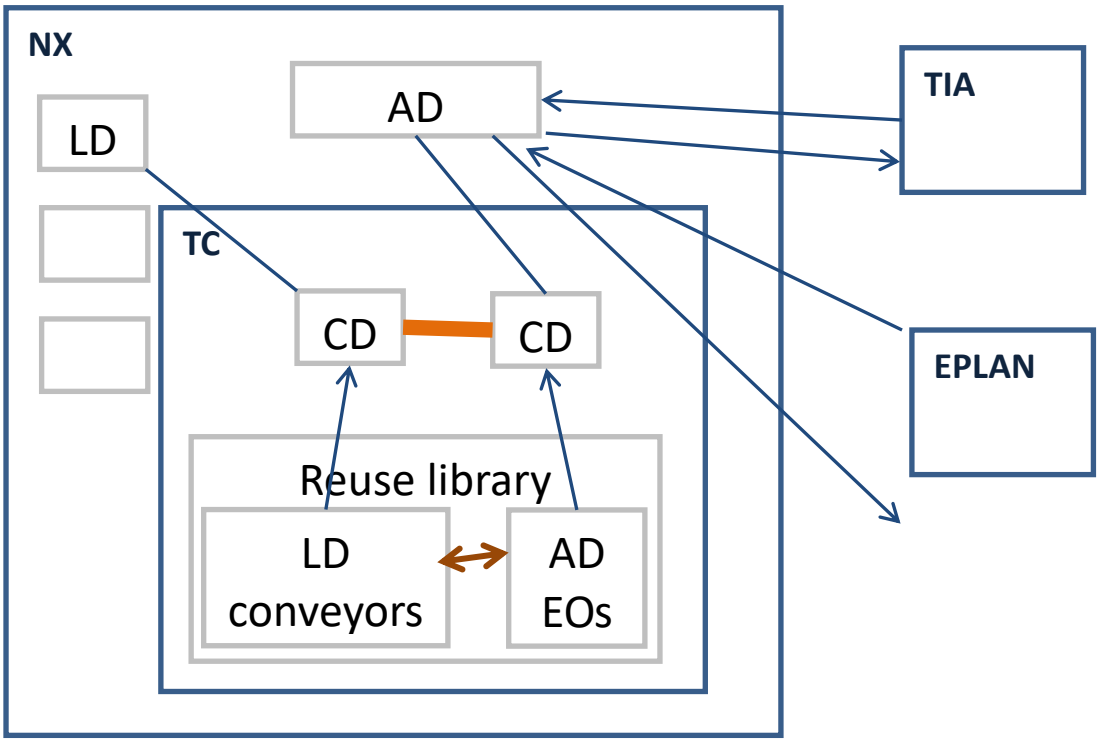
1_19



Part 2. Create LD DEs (conveyors), AD EO aspect tree, EPLAN, TIA Portal

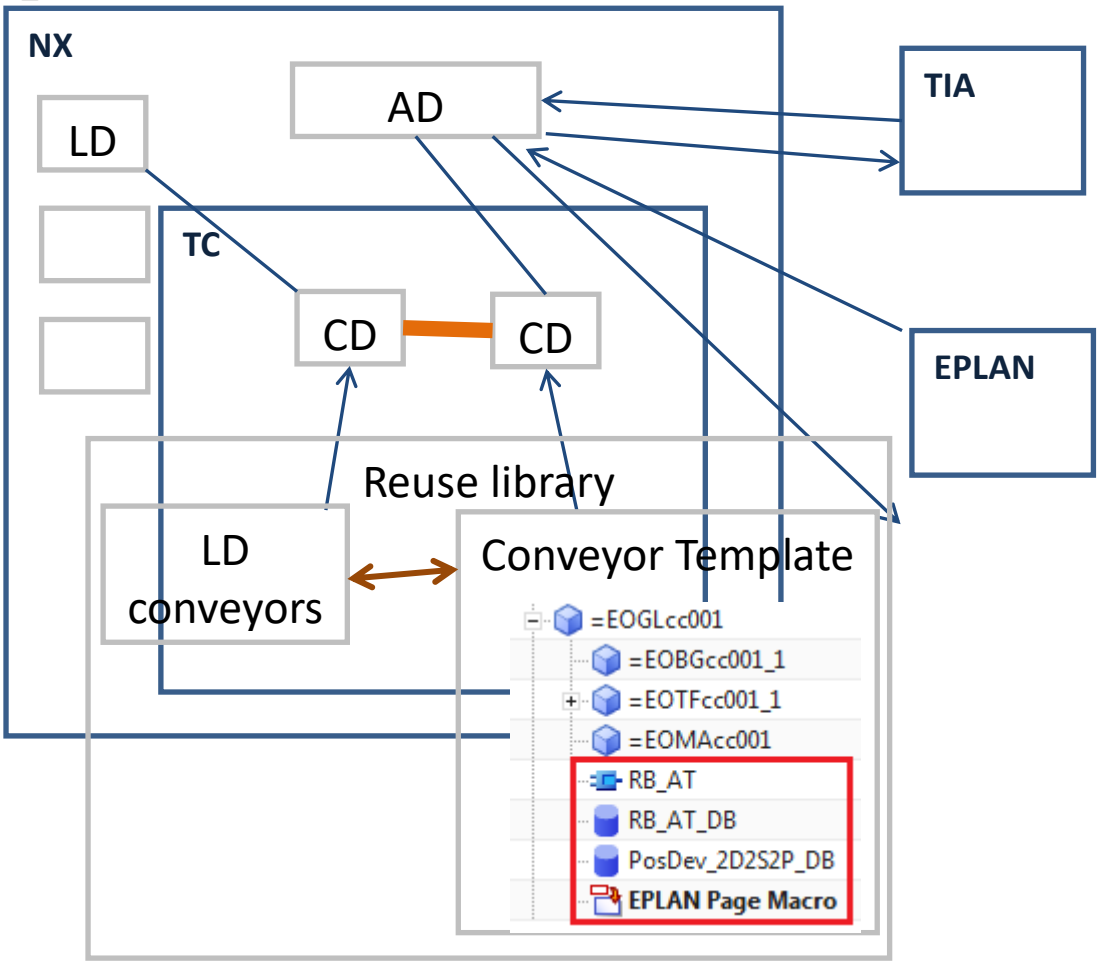
Add LD conveyors.
AD EOs
Import/generate EPLAN.
Import/generate TIA.

1_21



Part 3. Create expressions, template, instantiate.

1_20



Part 1. Create LD/AD TC components (CDs, EODefs), mapping

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

3. TeamCenter: Create Line Designer Collaborative Design.
4. Line Designer: Create Line Designer workset + Design Elements.
5. Automation Designer: Create Automation Designer workset (and Collaborative Design, SS) + Engineering Objects.
6. Map LD-AD

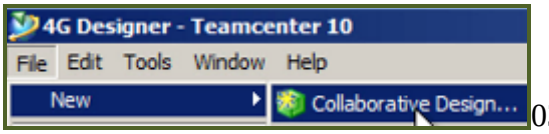
3. TeamCenter: Create Line Designer Collaborative Design

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

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

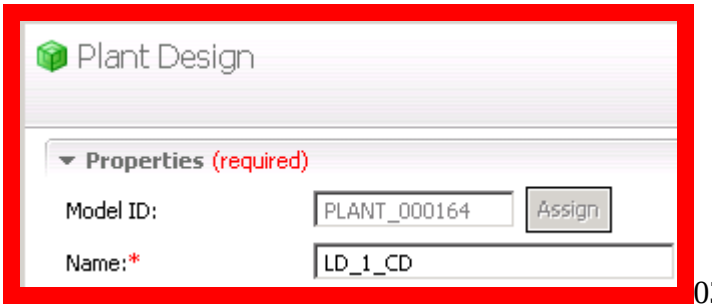
2. Select **Plant Design**.

03b_02

3. Click **Assign**.

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

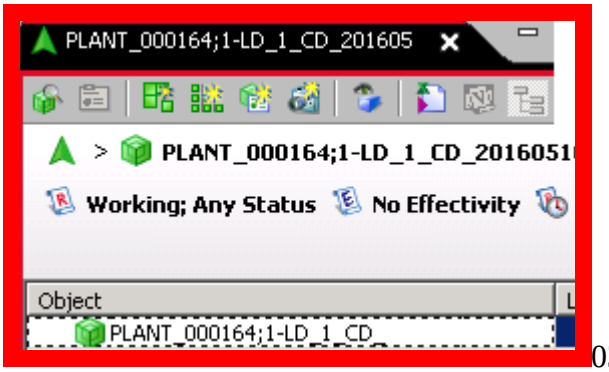
these names make it easier to follow in TeamCenter what is being created. Normally I add the date and/or initials to the name, such as LD_1_CD_TT_20160509



03b_03

5. Click **Finish**.

6. Click **Close**.

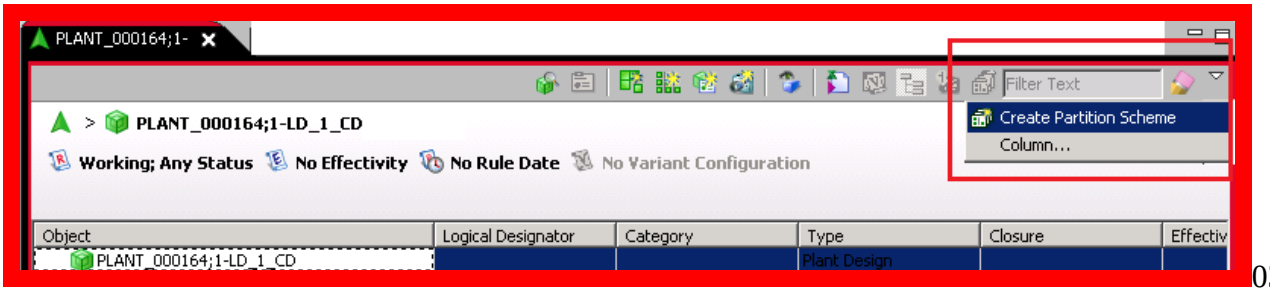


03b_04

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**.



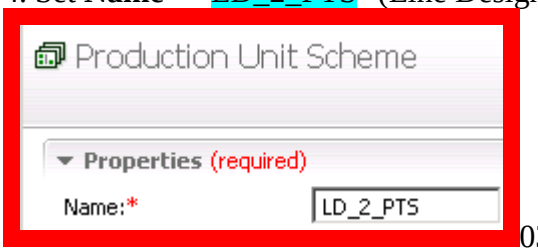
03b_05

2. Select **Production Unit Scheme**.

03b_06

3. Click **Next**.

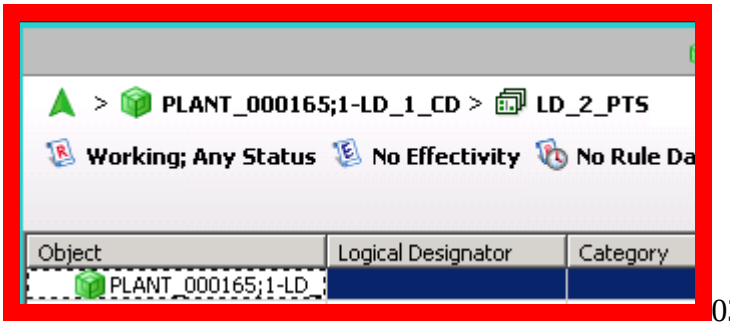
4. Set **Name** = "LD_2_PTS" (Line Designer Partition Scheme).



03b_07

5. Click **Finish**.

6. Click **Close**.



03b_08

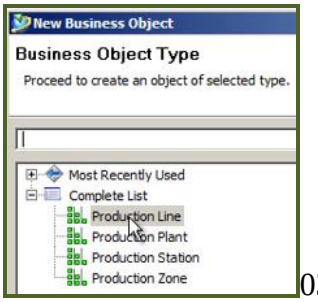
3.3. Create partition objects (and send to 4GD)

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

1. Click on **Create partition**.


03b_09

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

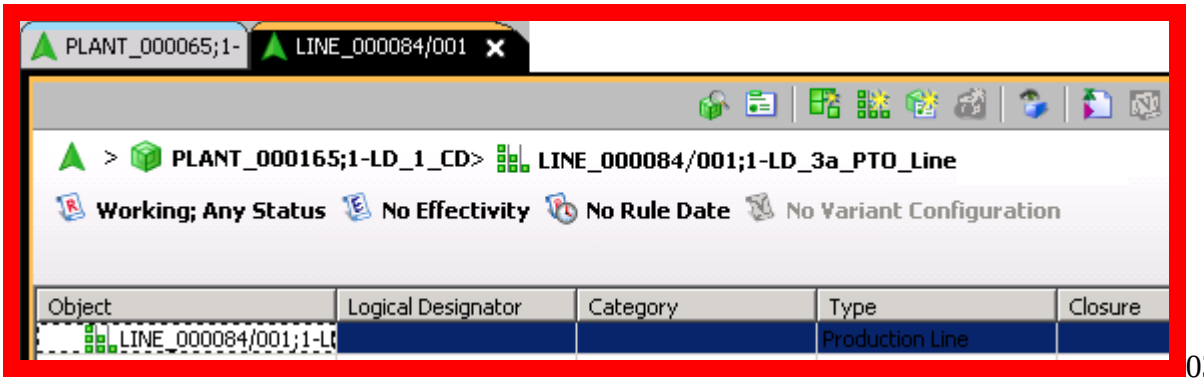


03b_10

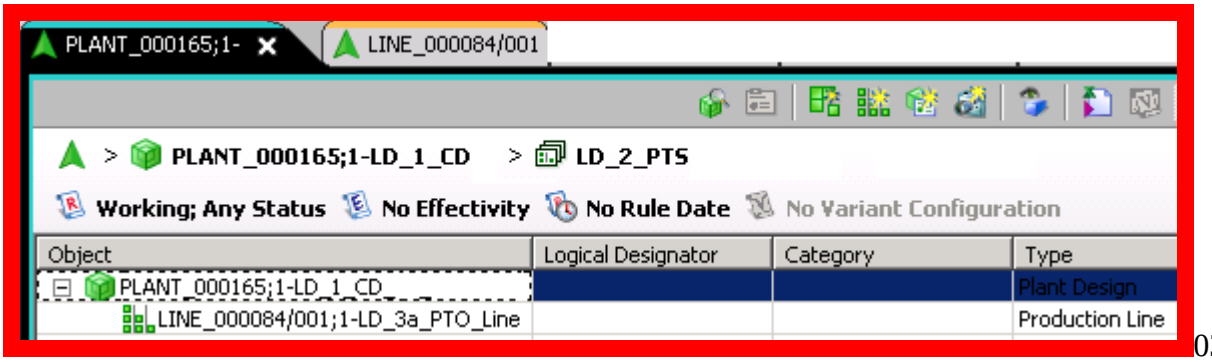
4. Click **Assign**.
5. Set **Name** = "LD_3a_PTO_Line".

03b_11

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

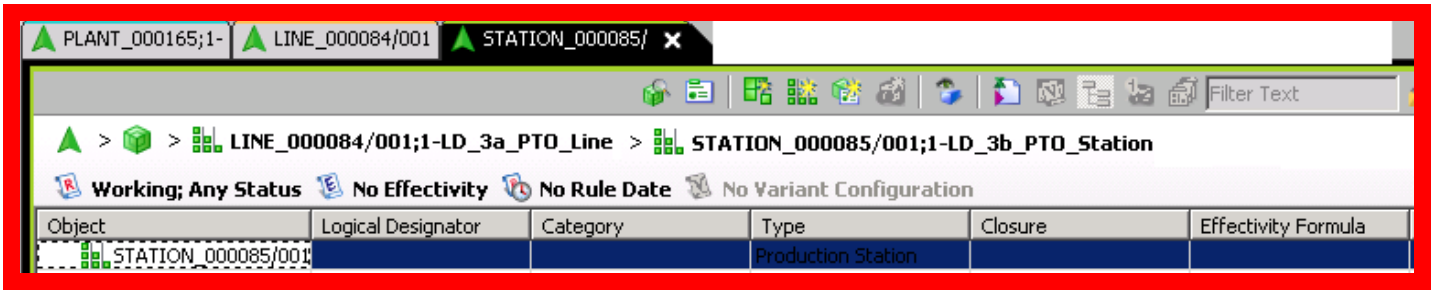


03b_12

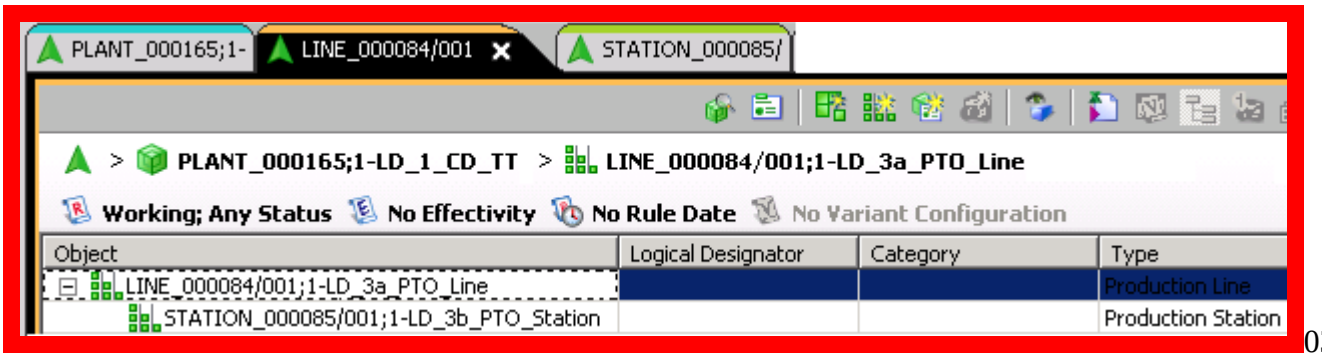


03b_13

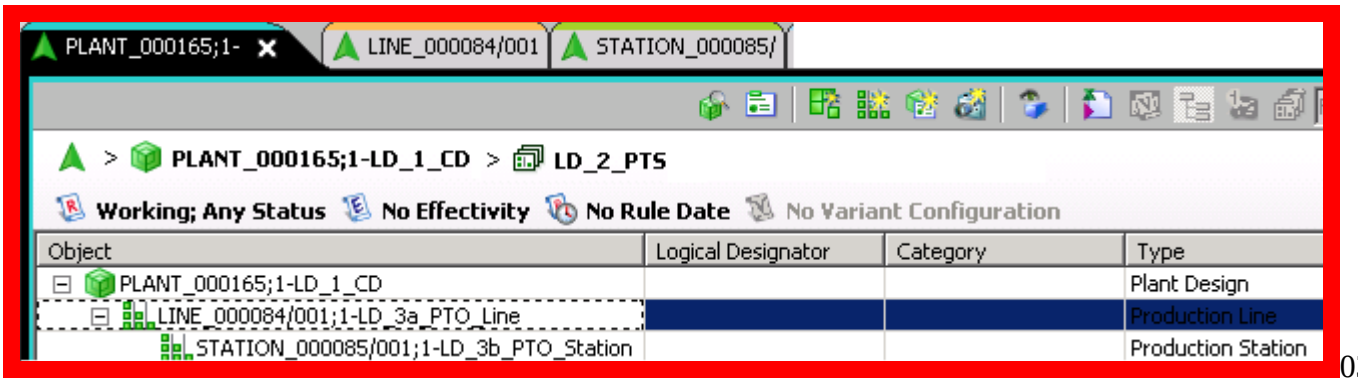
8. Click on the line under Plant in the Plant tab.
9. Click the **Create Partition** Icon.
10. Select **Production Station**.
11. Click **Next**.
12. Click **Assign**.
13. Enter **Name** = "LD_3b_PTO_Station".
14. Click **Finish**.
15. Click **Close**. The following shows what you have created so far.



03b_14

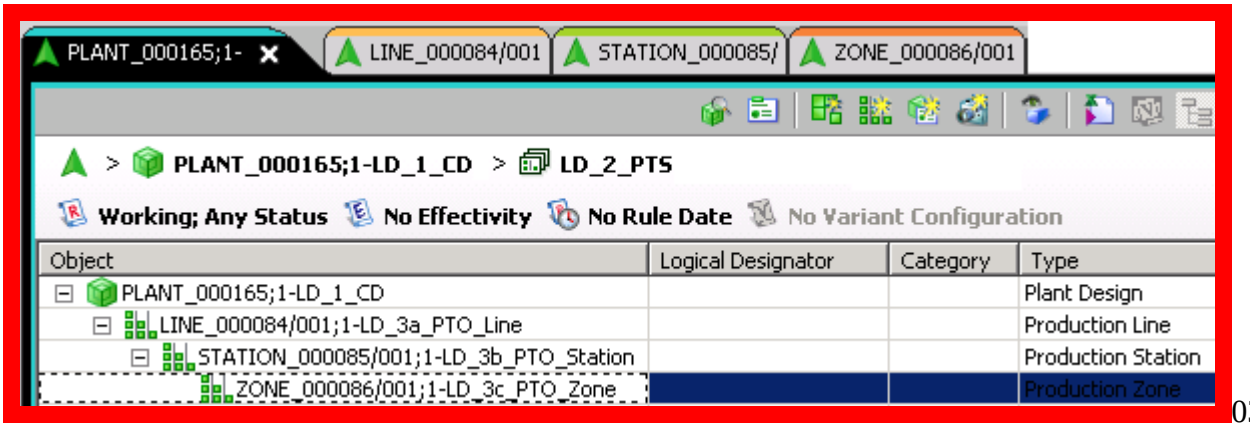


03b_15



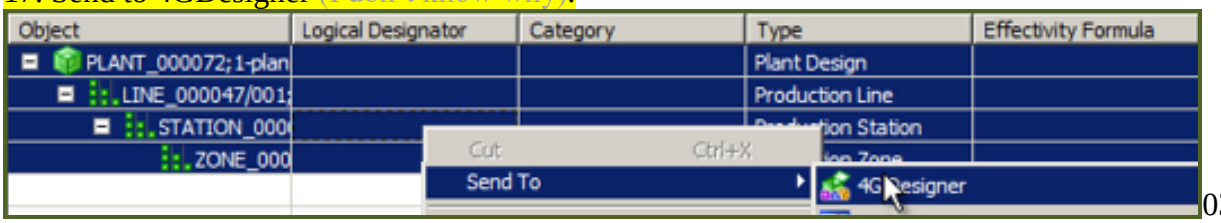
03b_16

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



03b_17

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



03b_18

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

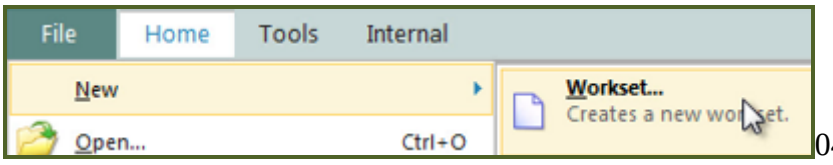
For 4GD requirements you must do the following:

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

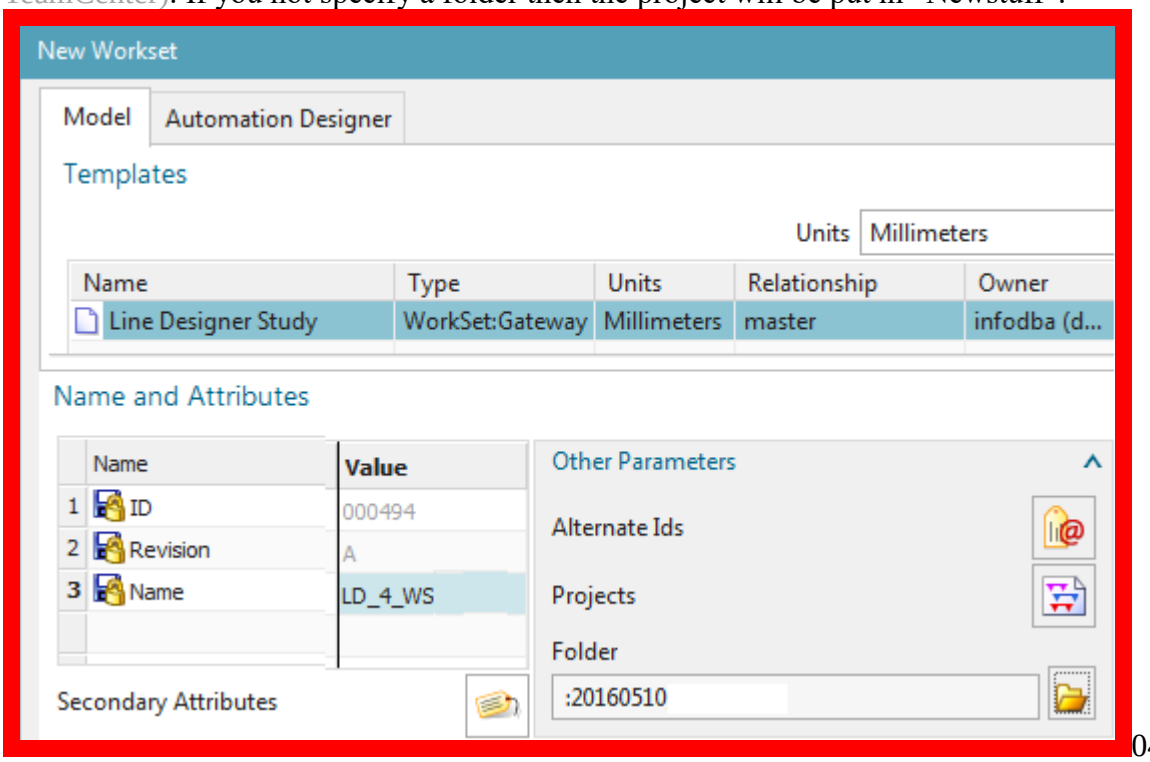
A workset object is the collection of Design Elements in your NX session. A workset is defined by one or more subsets. There may be many Design Elements within the workset you work on in your NX session.

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".



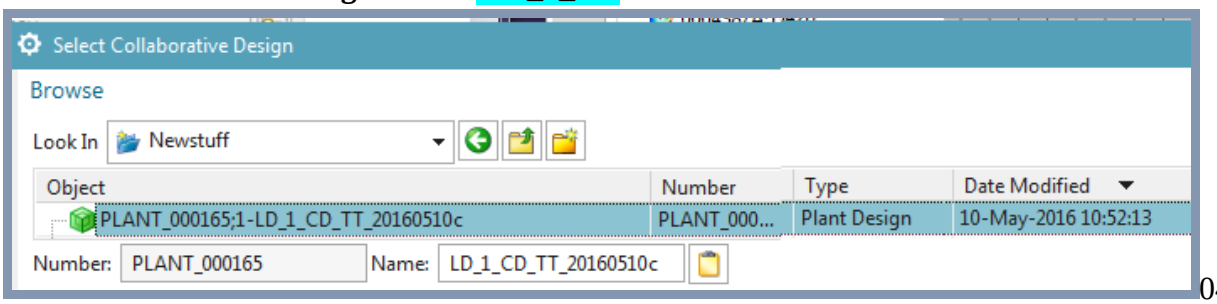
04b_02

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

4.2. Create Line Designer subset and add partitions to recipe

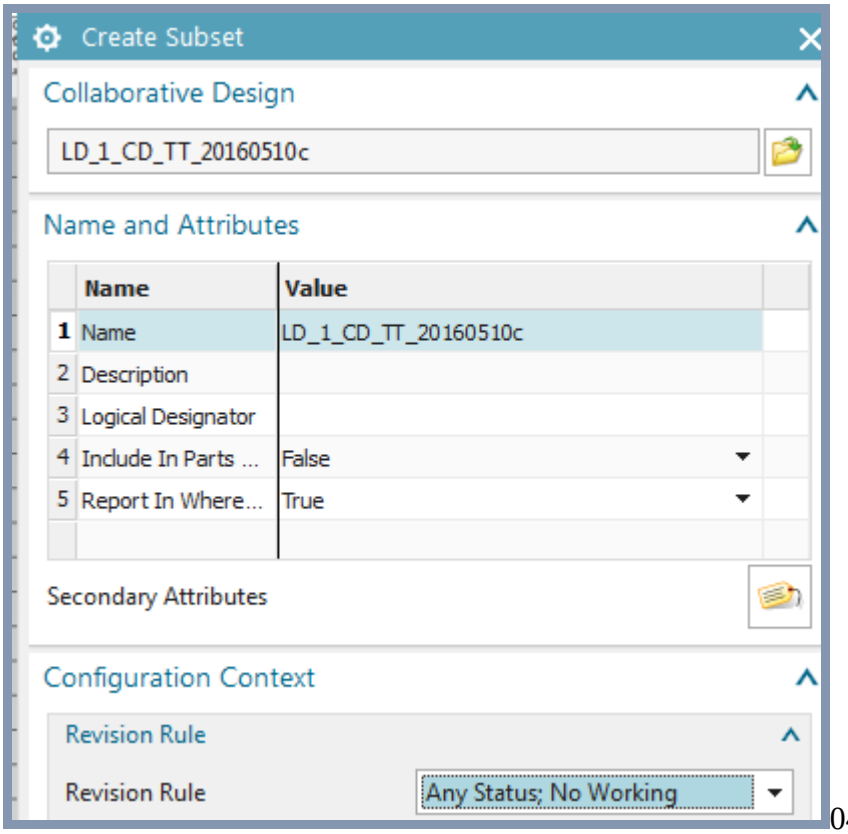
A subset object selects the design elements for a workset. The subset may include specific Design Elements, or it may contain a dynamic recipe which defines partitions to search. The diagram above shows a session with 2 subsets.

1. For **Collaborative Design** select " LD_1_CD".



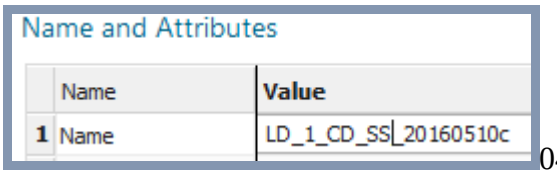
04b_03

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



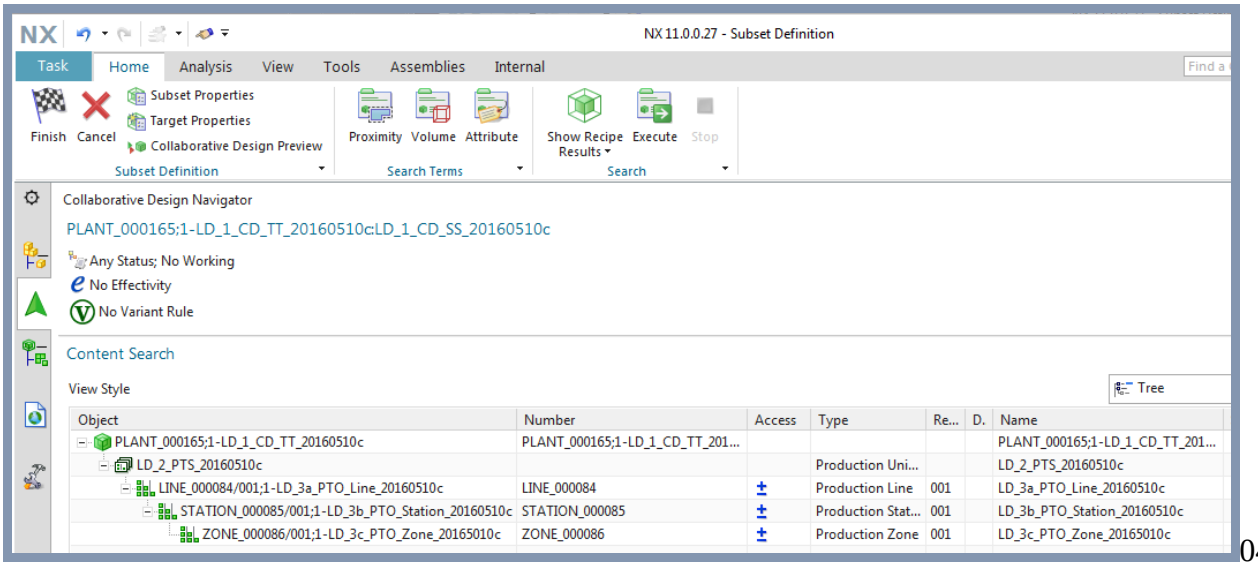
04b_04

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



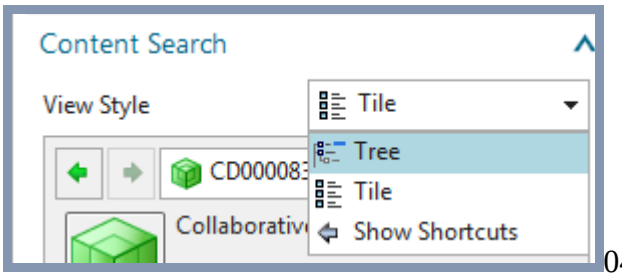
04b_05

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



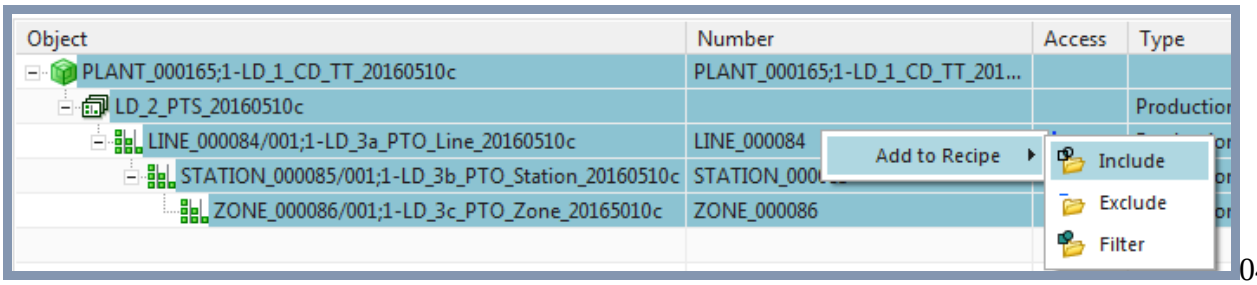
04b_06

6. Select the tree if not shown as above.



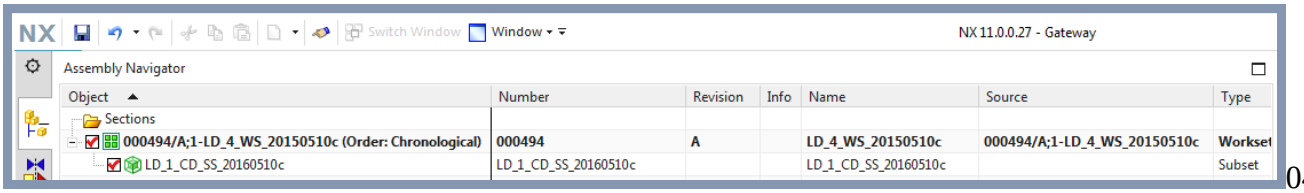
04b_07

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



04b_08

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



04b_09

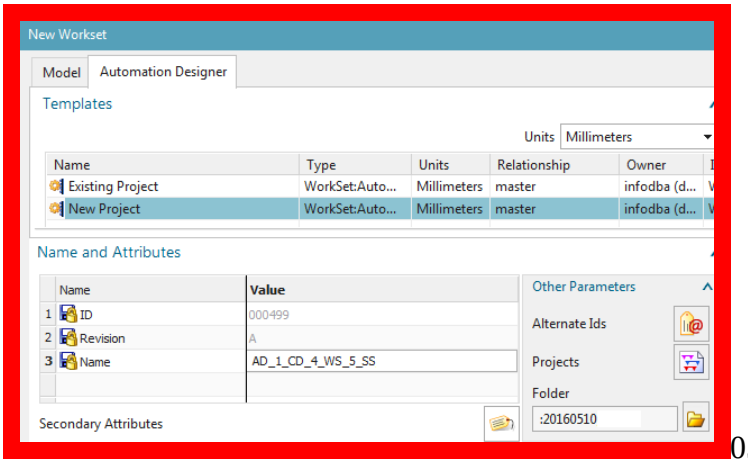
5. Create Automation Designer workset (and Collaborative Design, subset) and Engineering Objects

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

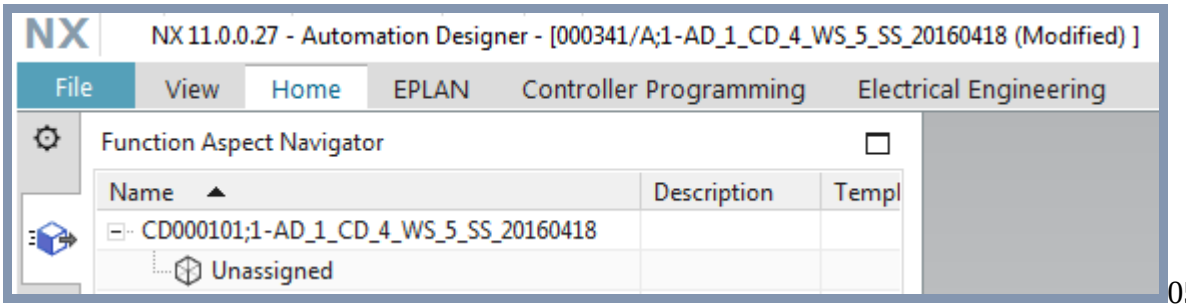
5.1. Create project workset (and Collaborative Design + subset)

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".



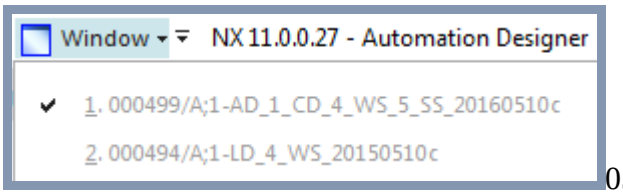
05b_01

4. Click **OK**.



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.



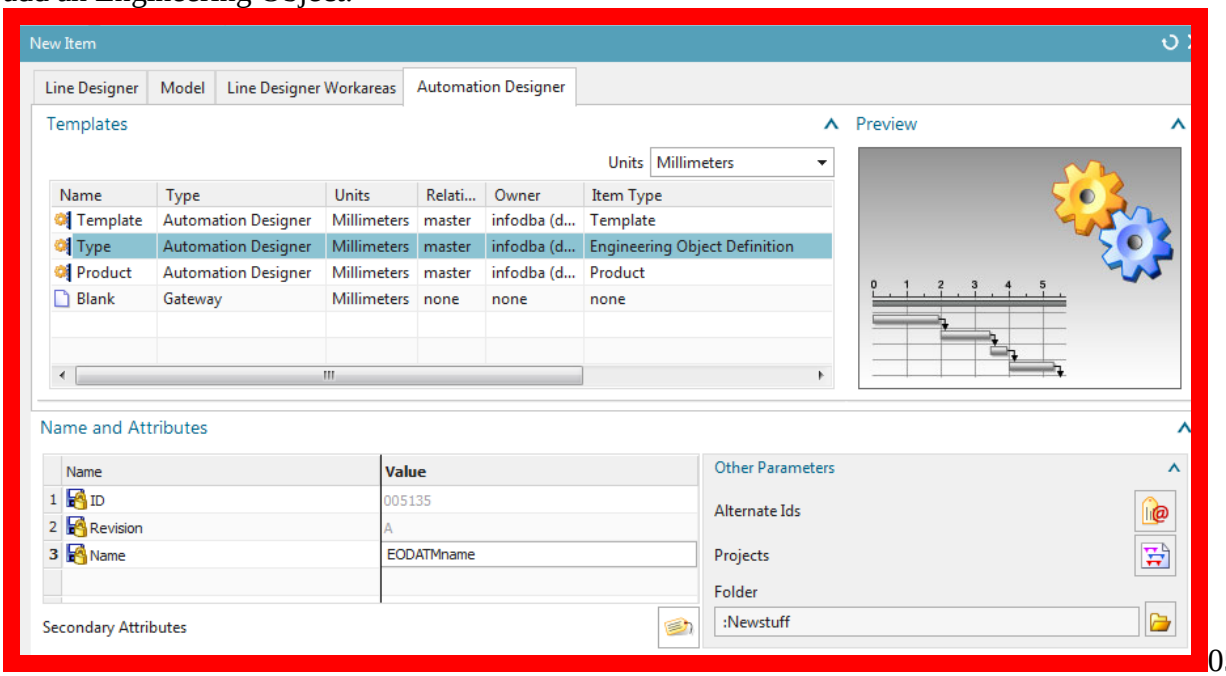
05_03

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.

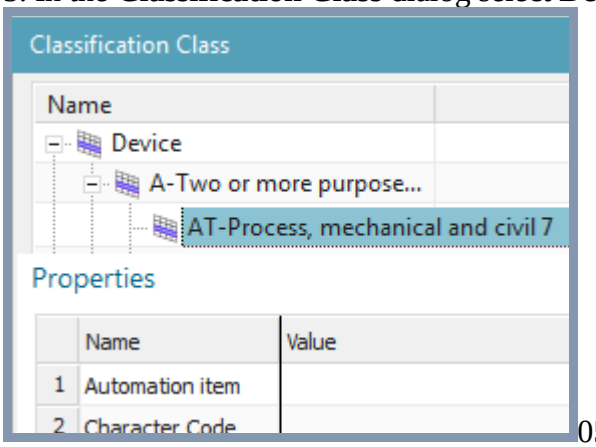
Create the first Engineering Object Definition.

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.

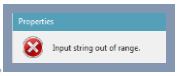


05b_04

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

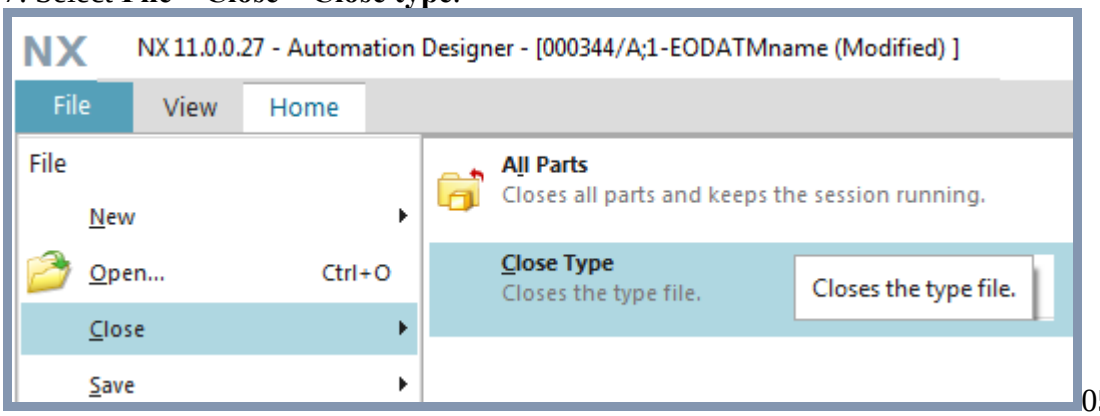


05b_05

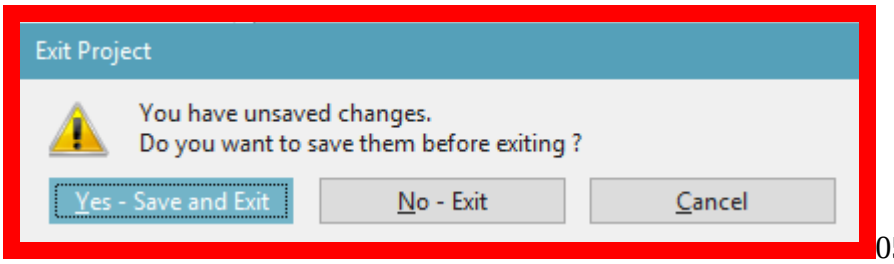
Click OK. 

6. Click **OK**.

7. Select **File→Close→Close type**.

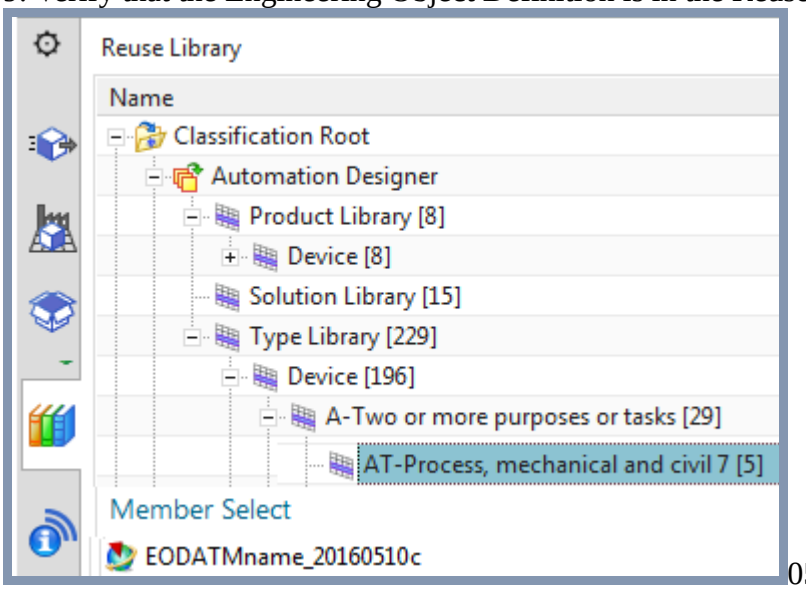


05b_06



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:

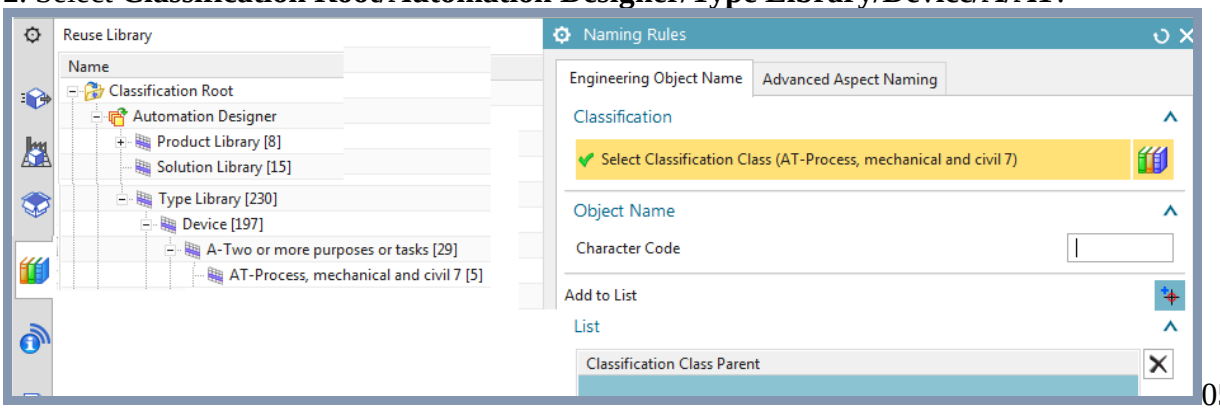
Type	Classification Root
1. EODATMname (created above)	Device / A ->1 purpose or task / AT
2. EODTLname	Device / U-Keep
3. EODGLname	Device / G-Generator / GL-Continuous flow
4. EODMAname	Device / M-Motor / MA-Electromagnetic
5. EODBGname	Device / B-Measurement / BG-Gauge,position
6. EODTFname	Device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

5.3. Create Engineering Object names

Engineering Object names

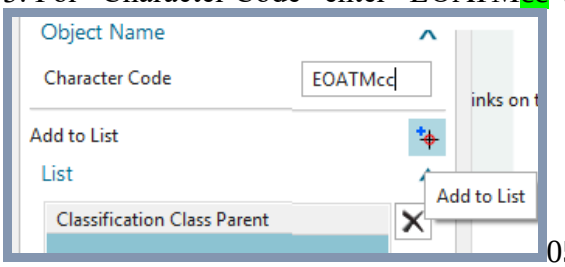
You now create the Engineering Object names that will display in the aspect tree.

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



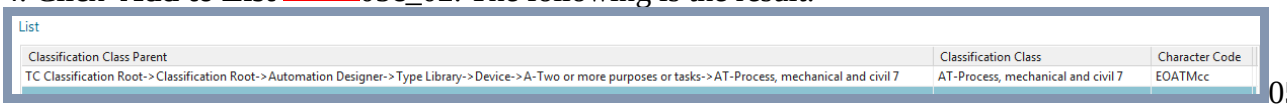
05b_08

3. For "Character Code" enter "EOATMcc".



05b_09

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

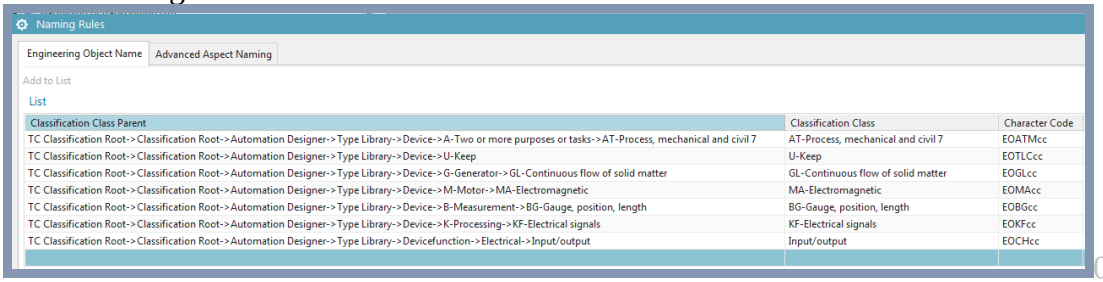


05b_10

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.



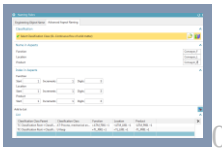
05b_11

Advanced 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.

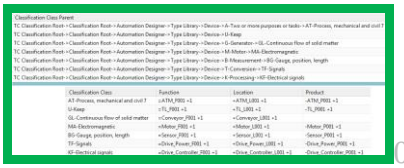


05b_12

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:



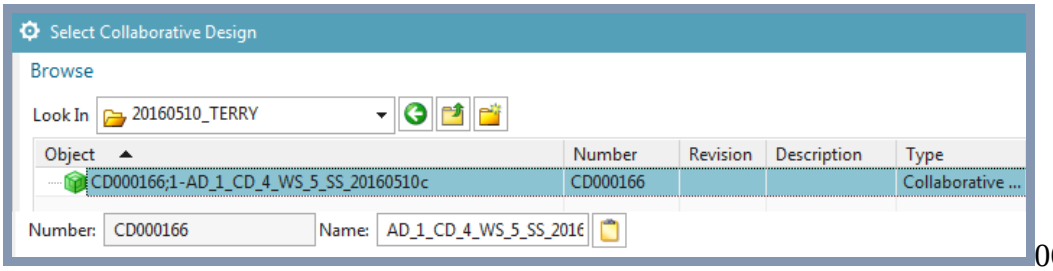
05b_13

6. Map Line Designer-Automation Designer

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

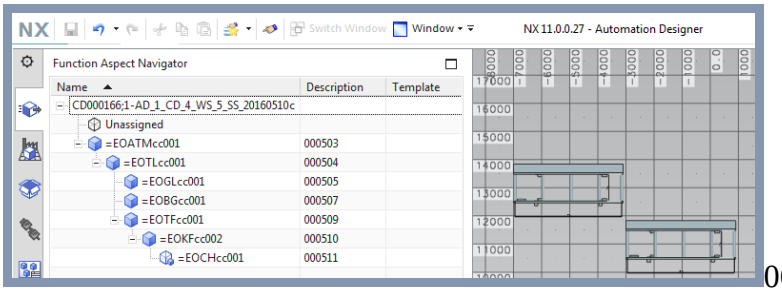
1. Close the Automation Designer project you created previously.
2. Open the Line Designer Collaborative Design.
3. Select **File→All Applications→Automation Designer**.
4. Select the Automation Designer Collaborative Design.



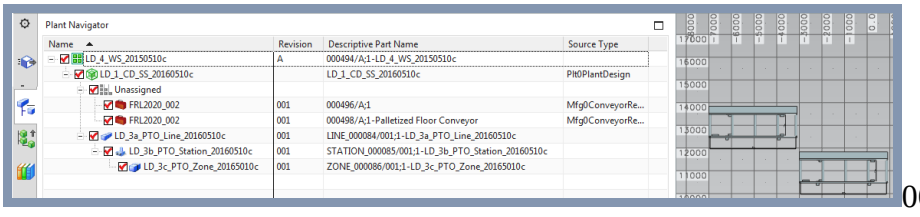
06b_02

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.



06b_03



06b_04

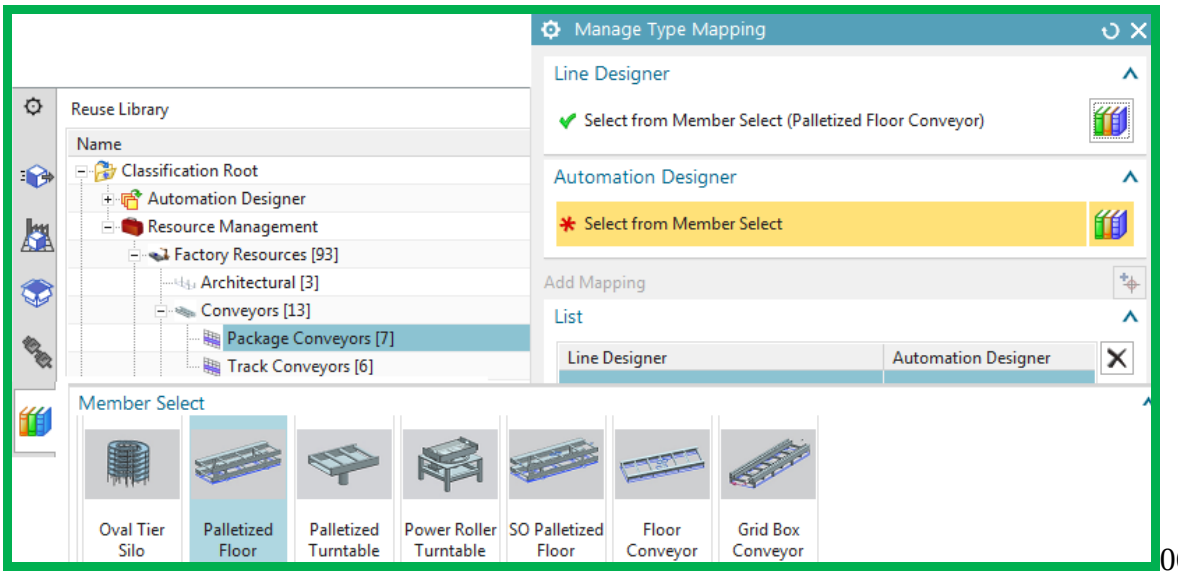
20160509 they were not there.

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.

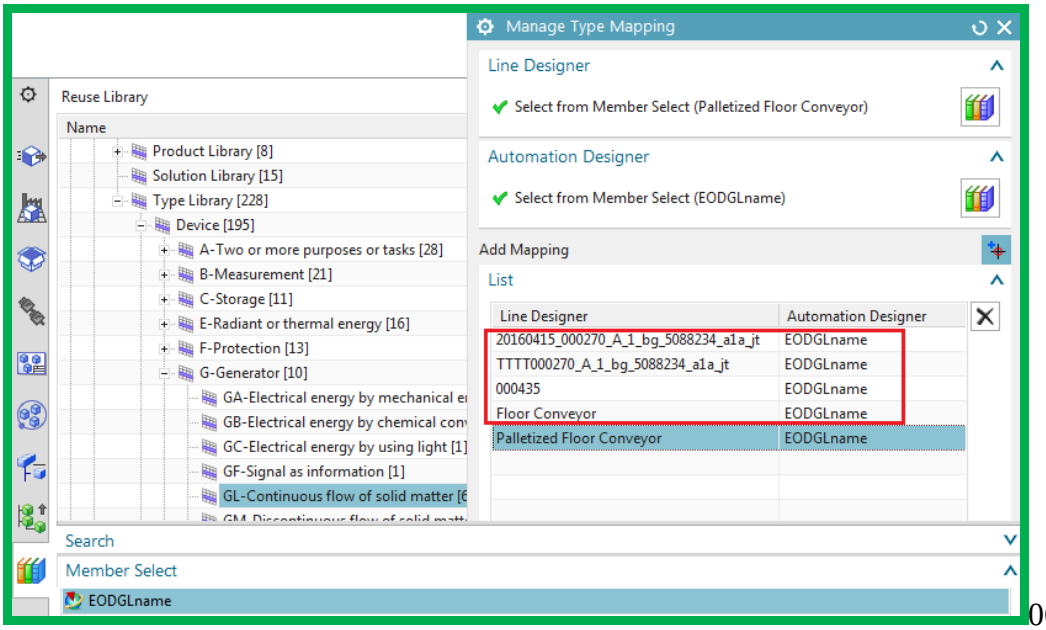
You now will define the type mapping. Type mapping determines for which type of Line Designer object what type of Engineering Object will be created when you use **Map to new**.

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



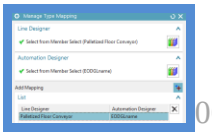
06b_05

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



06b_06

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


06b_07

Part 2. Config (non-template) LD DEs (conveyors), AD EO aspect tree, EPLAN, TIA Portal

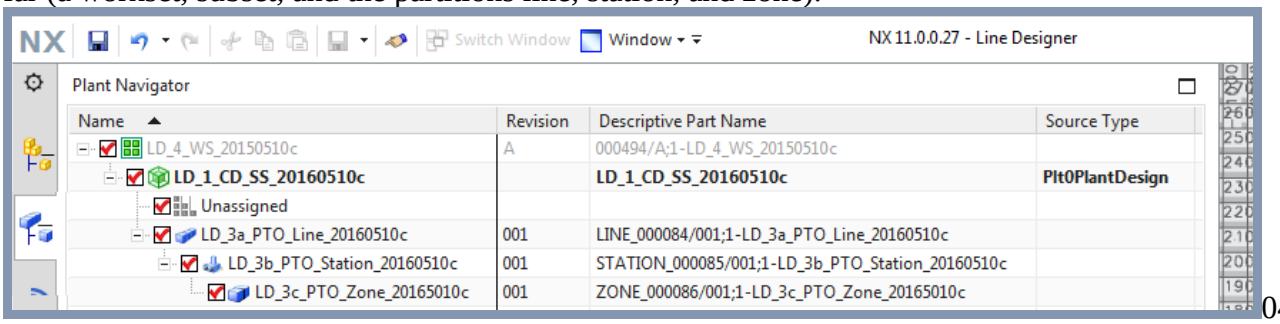
This part shows how to create the mapping and TIA Portal software for a single conveyor. You will not create a template and will not use expressions or ports until part 3 (to keep things simple).

7 (4.3). Add 2 Line Designer conveyors
8 (5.4). Add Engineering Objects
9. Configure a basic AD project for EPLAN
10. Configure a basic AD project for TIA Portal

7. Add 2 Line Designer conveyors

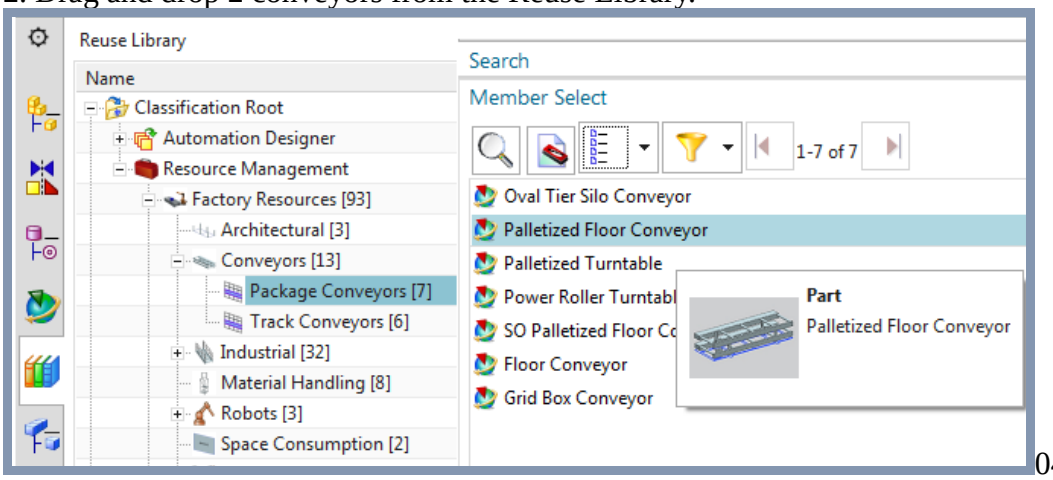
You now add two conveyors that will be linked (mapped) later to Engineering Objects.

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).



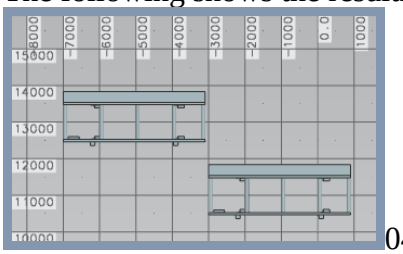
04b_10

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

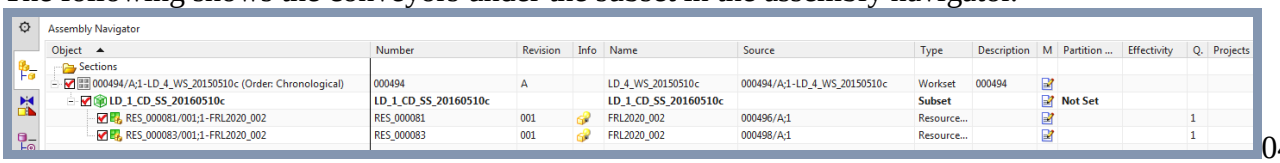


04b_11

The following shows the resulting conveyors.

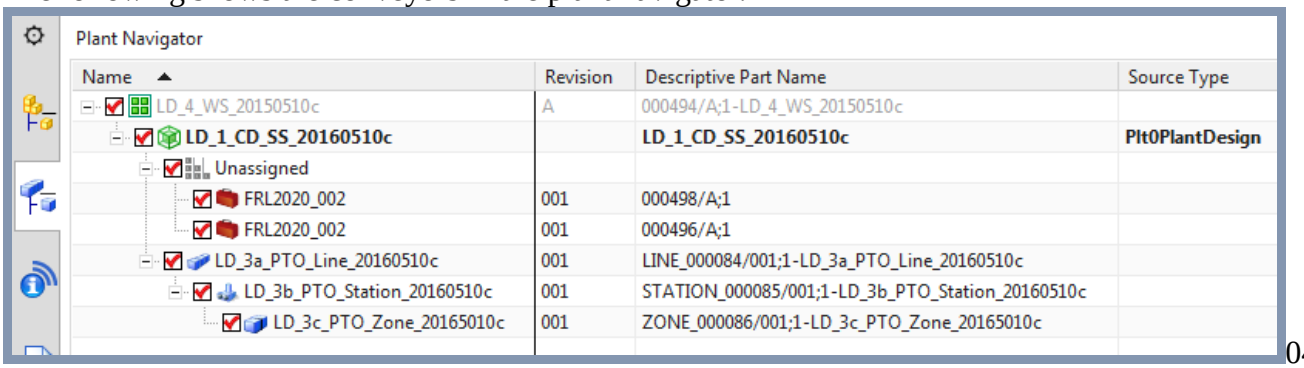

04b_12

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



04b_13

The following shows the conveyors in the plant navigator.



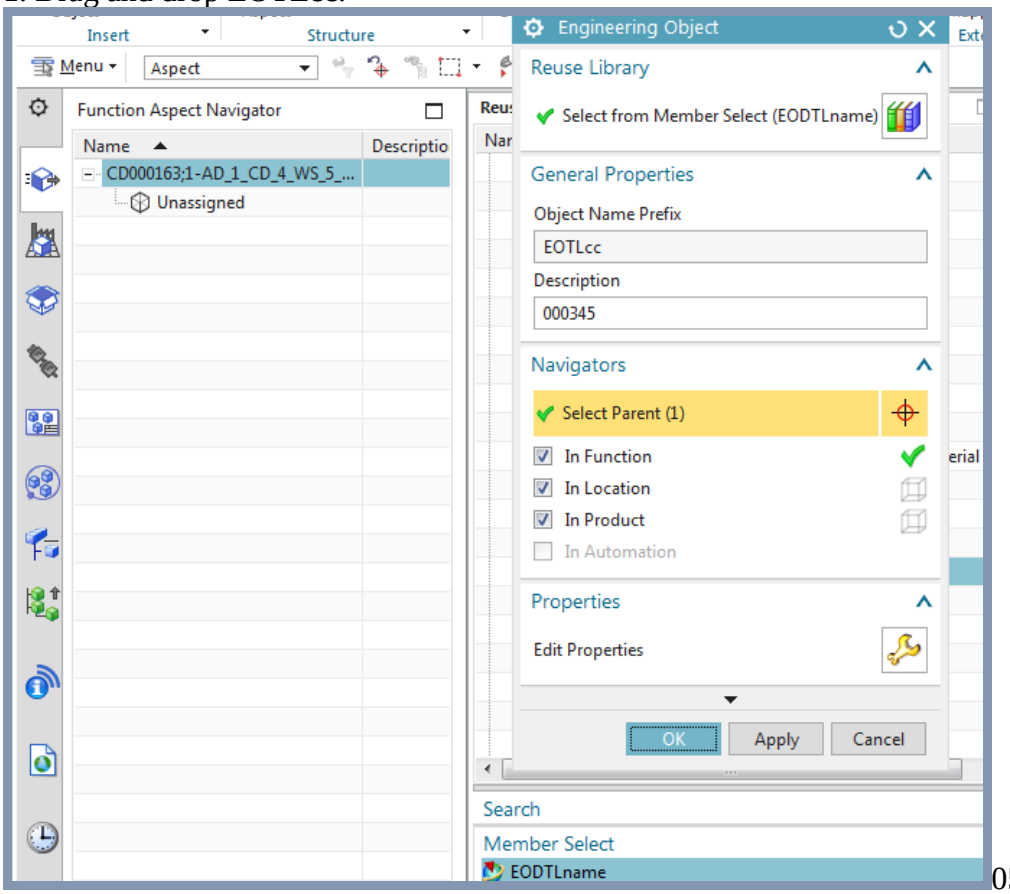
04b_14

8. Add Engineering Objects

8.1. add upper level (ATM, TL) to Function aspects

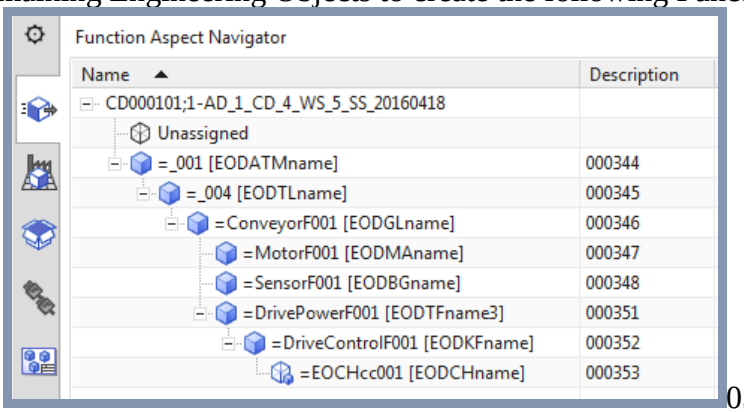
In this section you will only add in the Function aspect.

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



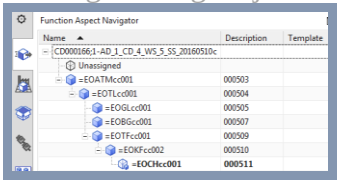
05b_14

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

ONLY TOP 2 

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.



8.2. 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.

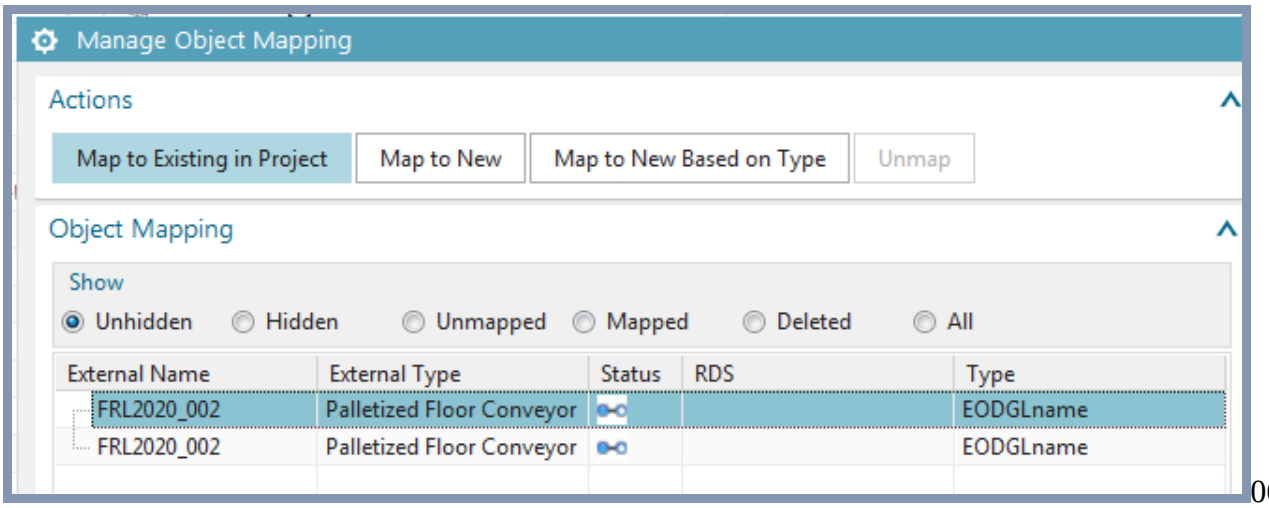
Use the **Manage Object Mapping** dialog box to map single external objects to single Engineering Objects or templates. Then continue engineering with the mapped Engineering Object or template.
There are 3 ways to map objects:

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

1. Map to existing

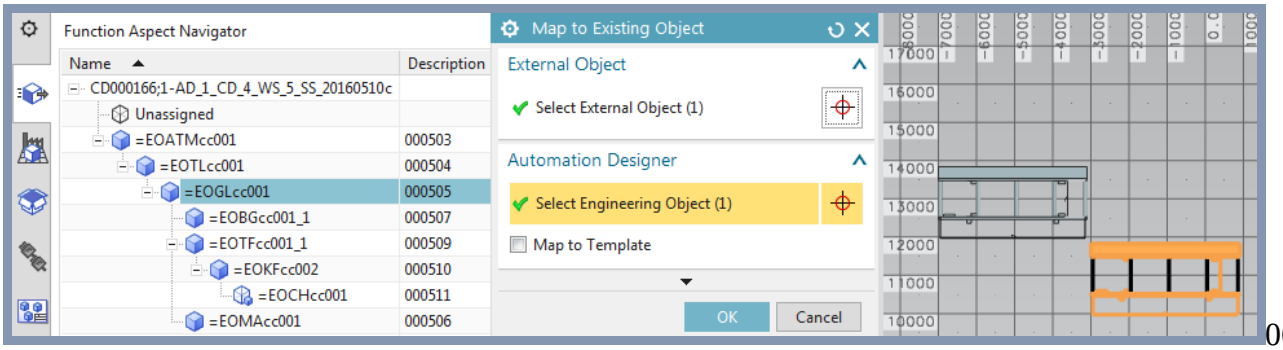
You can map external objects to existing Engineering Objects or templates. You can choose an Engineering Object or template whose type matches the type mapping defined for all projects. Or you choose an Engineering Object or template whose definition deviates from the type mapping. In that case, the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

1. Click **Manage object mapping**. 2 conveyors appear.
2. Select a conveyor.



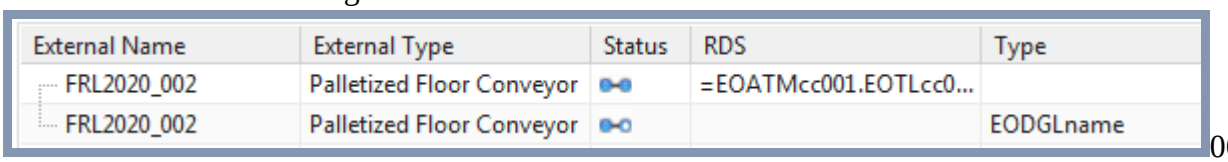
06b_08

3. Click **Map to existing in project**.
4. Select GL.



06b_09

5. Click **OK**. The following is the result.



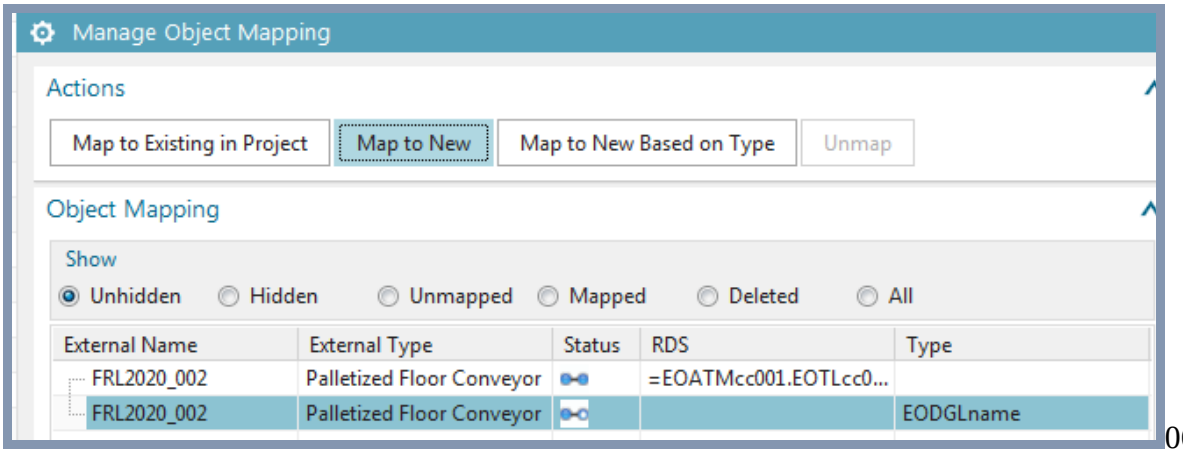
06b_10

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.

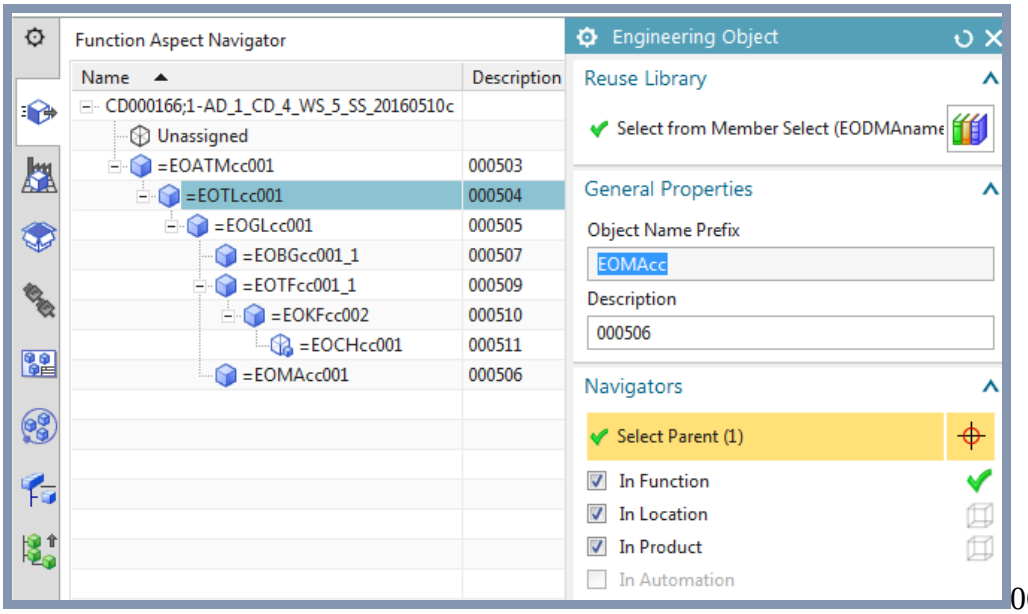
Now try to create a new Engineering Object based on the mapping.

1. Select the unmapped Engineering Object.
2. Click **Map to new**.



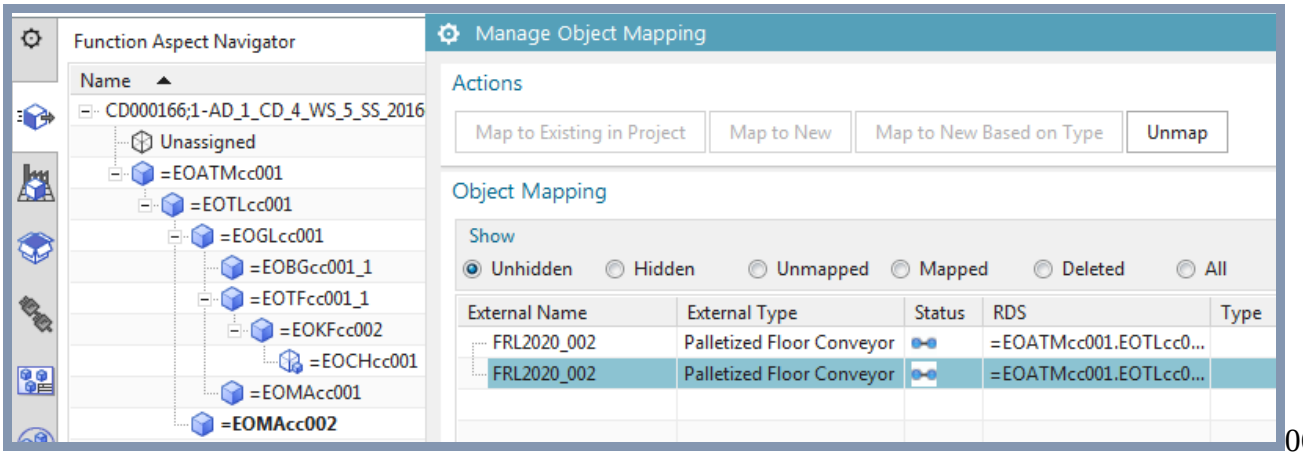
06b_11

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



06b_12

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

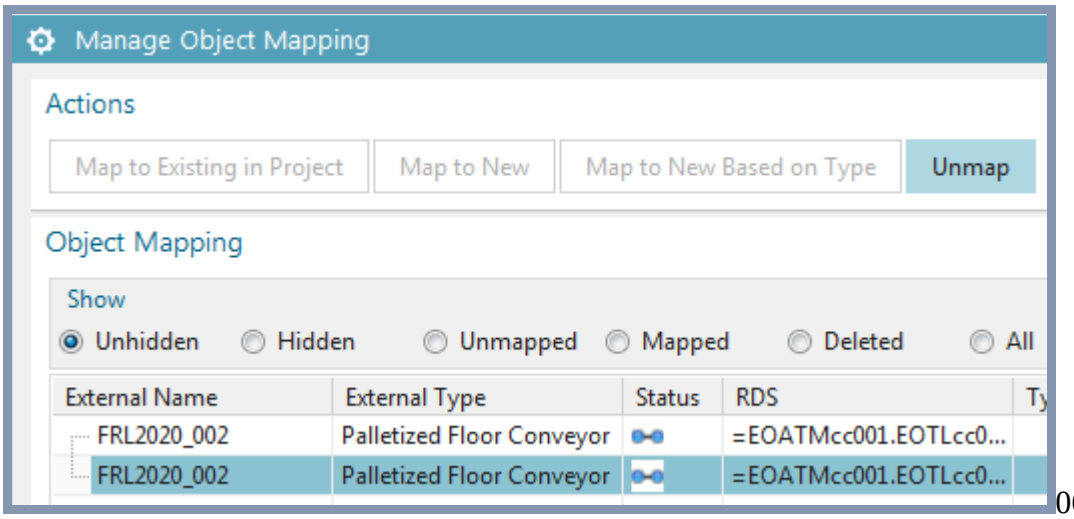


06b_13

3. Map to new based on type

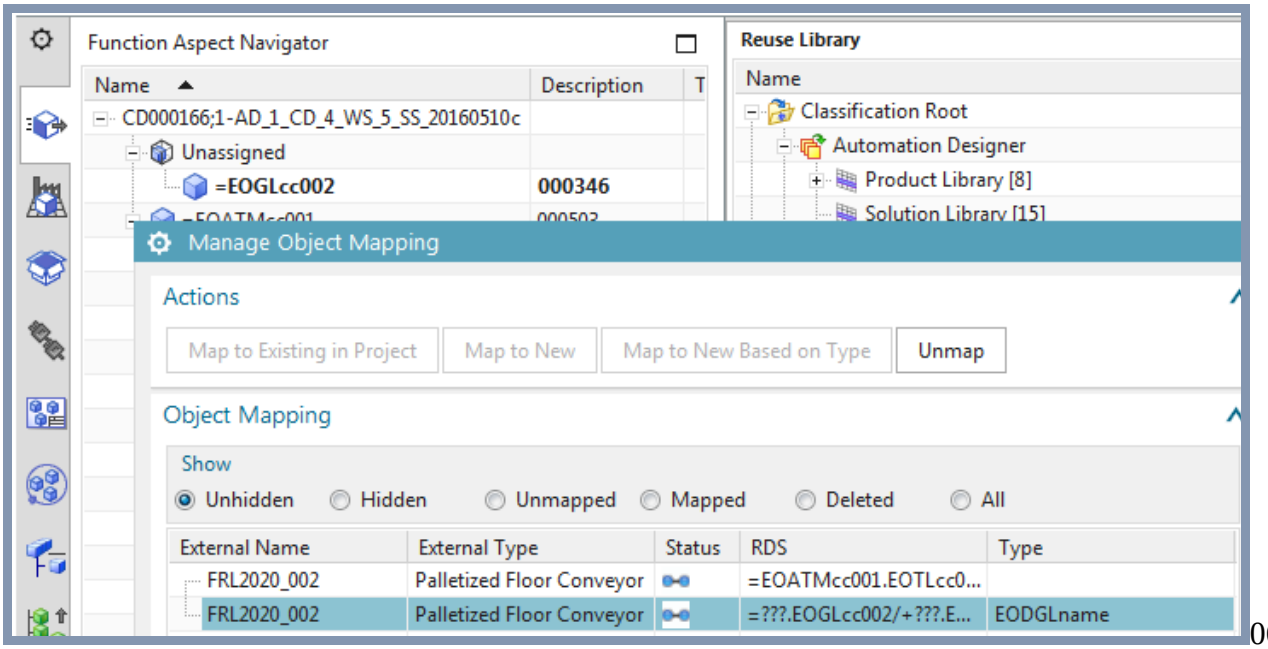
If you want to map external objects to new Engineering Objects, you can use the type mappings that were defined as default for all projects.

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



06b_14

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

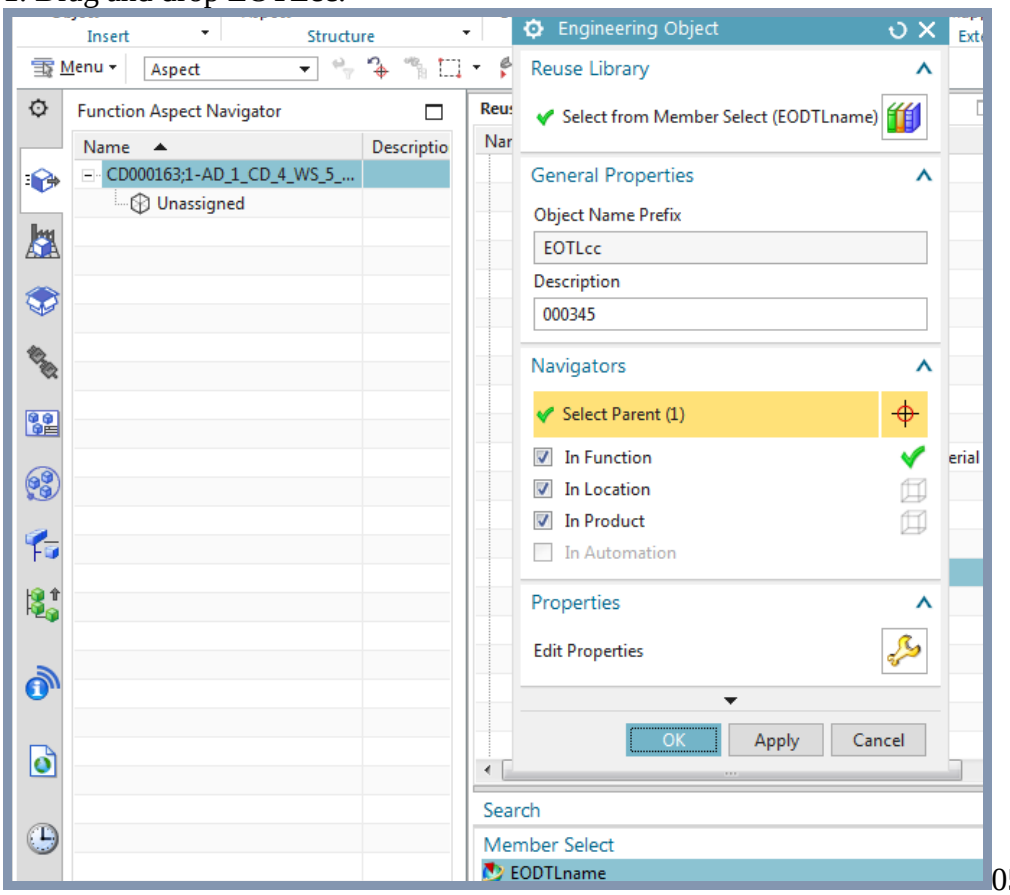


06b_15

8.3. add lower level to Function aspects

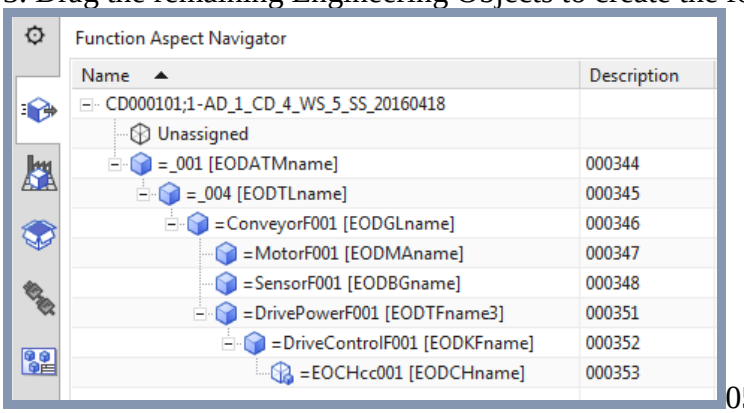
In this section you will only add in the Function aspect.

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



05b_14

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

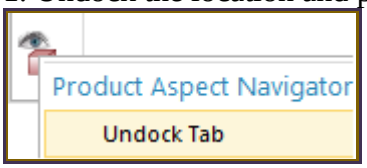


05b_15

8.4. Location-Product aspects

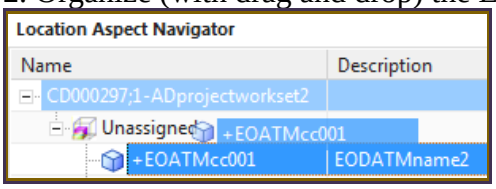
Now you will configure the location and product aspects.

1. Undock the location and product aspects.



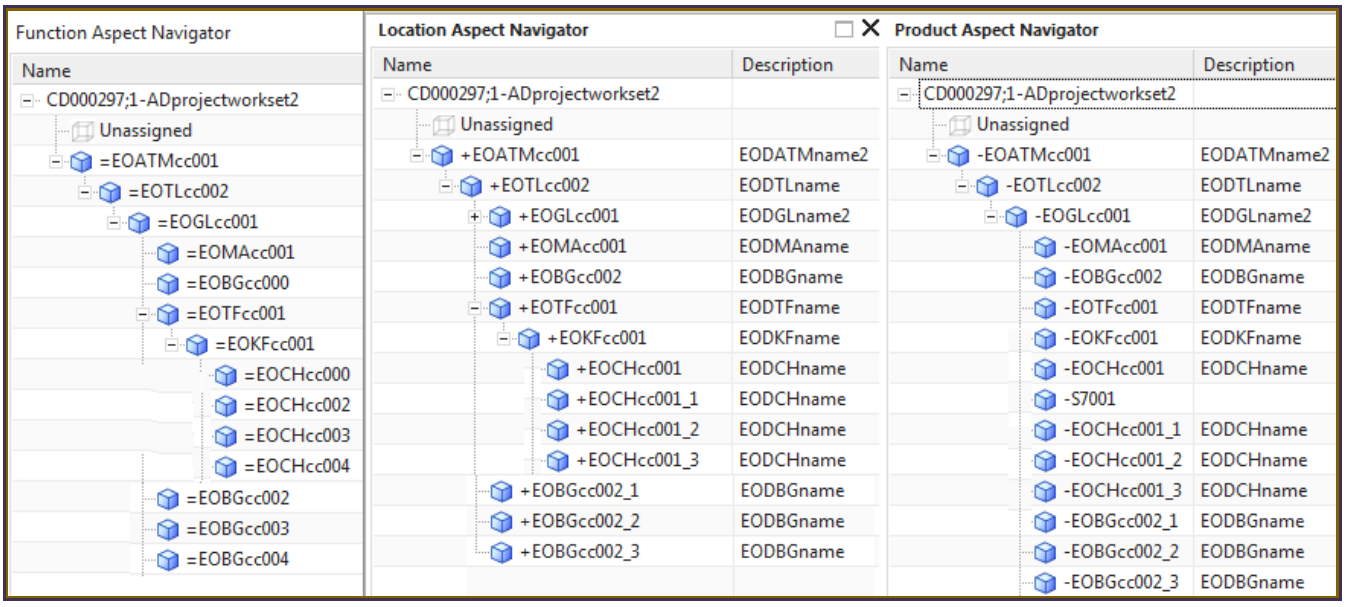
05_28

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



05_29

3. The result should be like this.



05_30

9. Configure a basic AD project for EPLAN

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

9.1. Import EPLAN project template

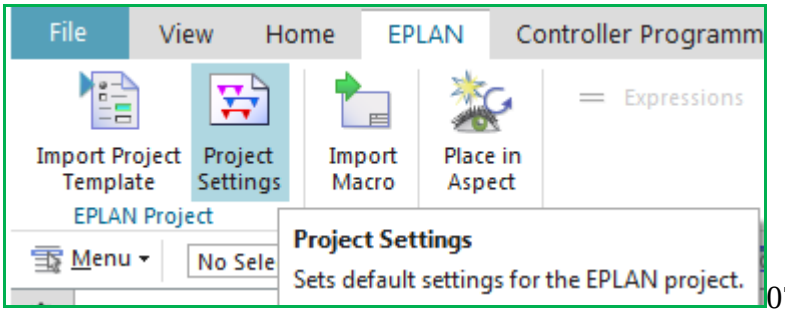
9.2. Add PM250D macro

9.3. 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

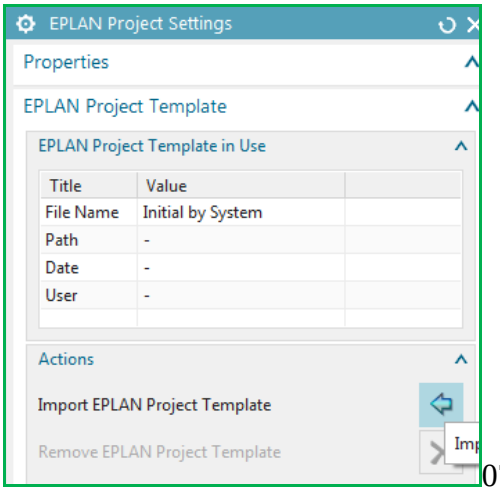
9.1. Import EPLAN project template

1. Click on Project Settings.

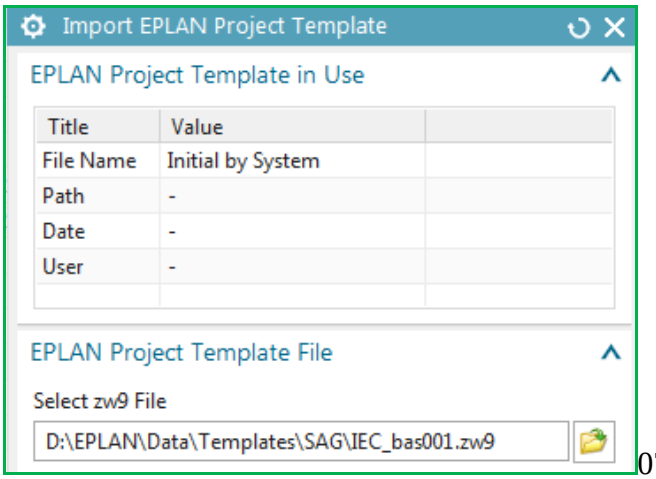


07b_01

2. Import the template IEC_bas001.zw9.

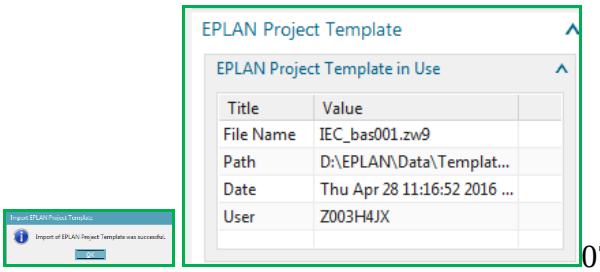


07b_02



07b_03

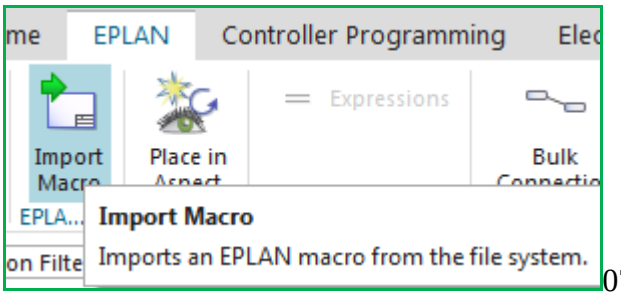
3. Click OK.



07b_04

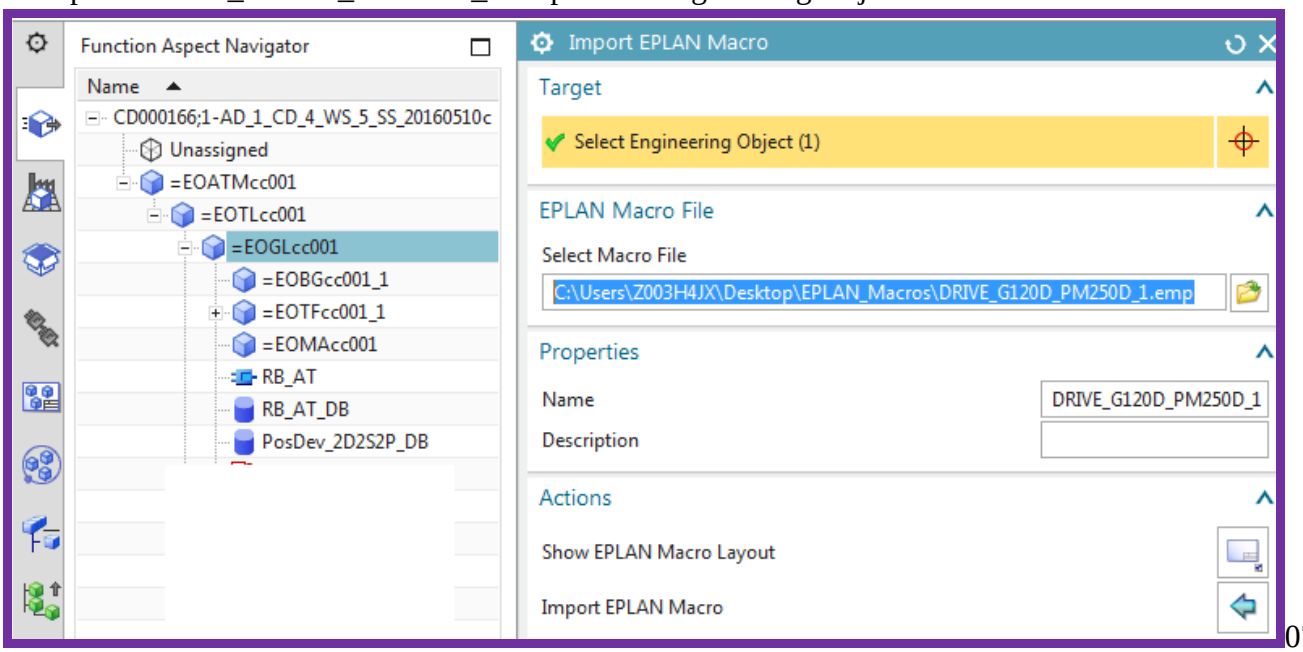
9.2. Add PM250D macro

1. Click on **Import Macro**.



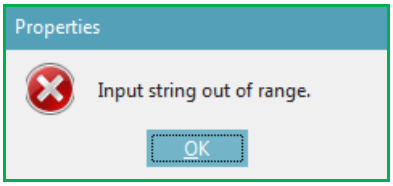
07b_05

2. Import DRIVE_G120D_PM250D_1.emp under Engineering Object GL.

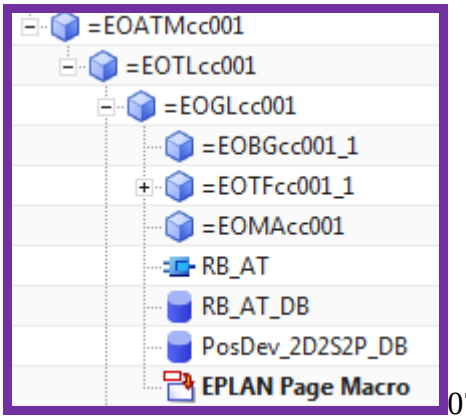


07b_06

3. Ignore this error.

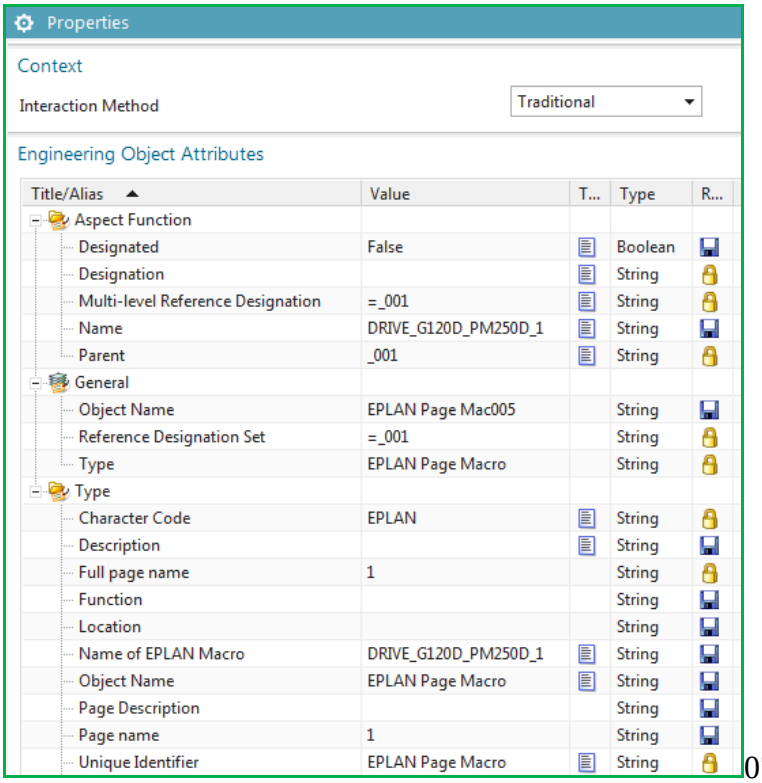


3. Click Close. The macro appears in the aspect tree.

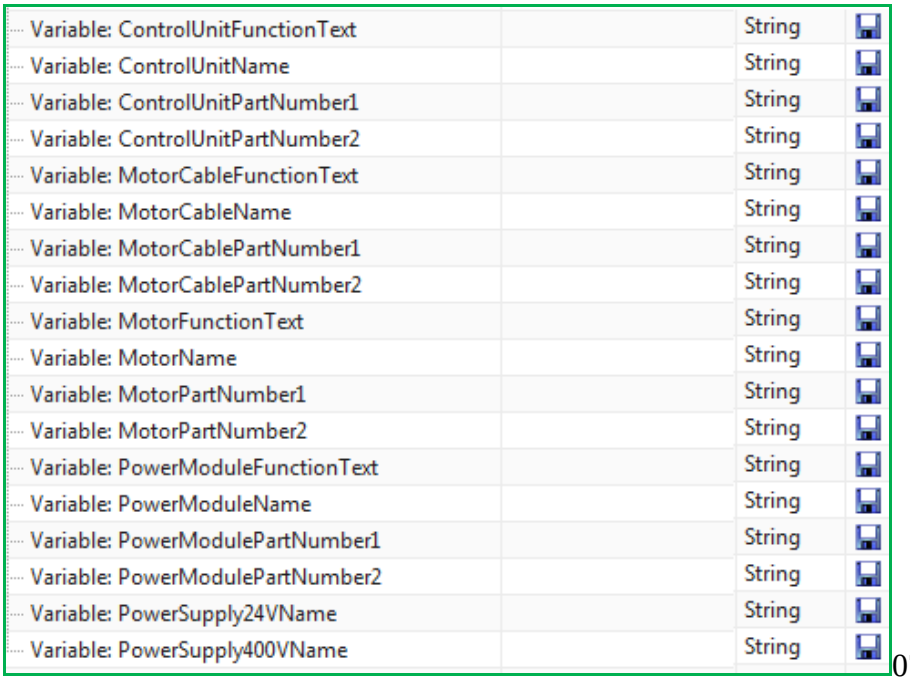


07b_07

Note the default properties.



07b_08



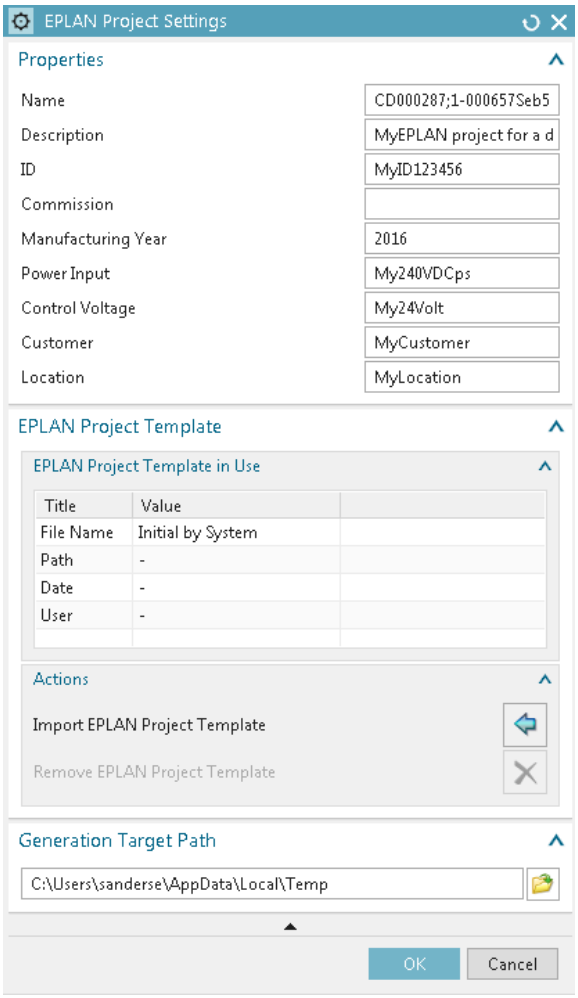
07b_09

9.3. 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.



9.4. 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)

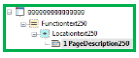
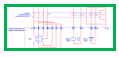
Assume you replaced the values as shown below (you don't need to actually do this). Values in grey you cannot modify.

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

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  07b_15  07b_16
- old pics:      

9.5. Generate

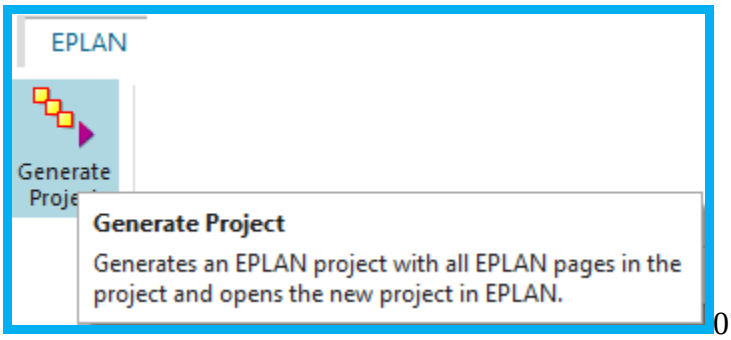
Generate an EPLAN report with

9.3.1. Default values

9.3.2. Simple text values

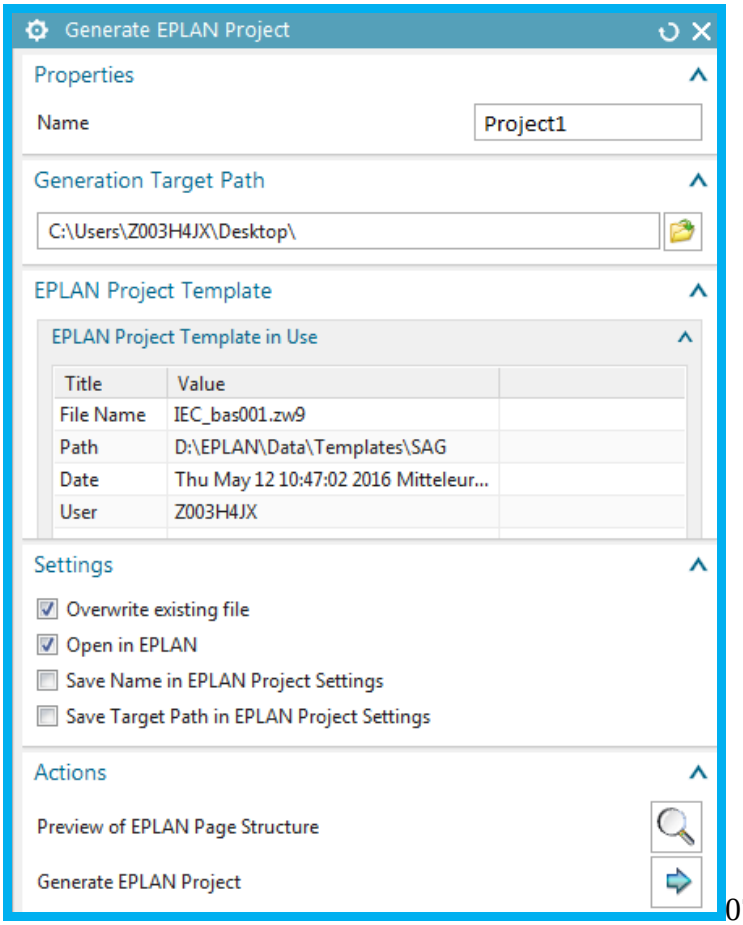
9.3.1. Default values

1. Click Generate Project.



07b_10

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



07b_11

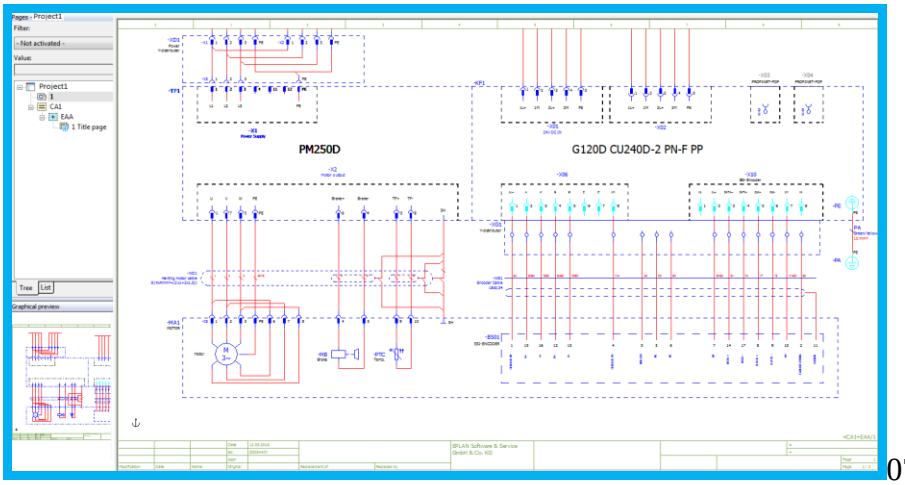
3. Click **Generate**. The project is opened in EPLAN.

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;1-000657Seb5
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

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



07b_13

10. Configure a basic AD project for TIA Portal

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

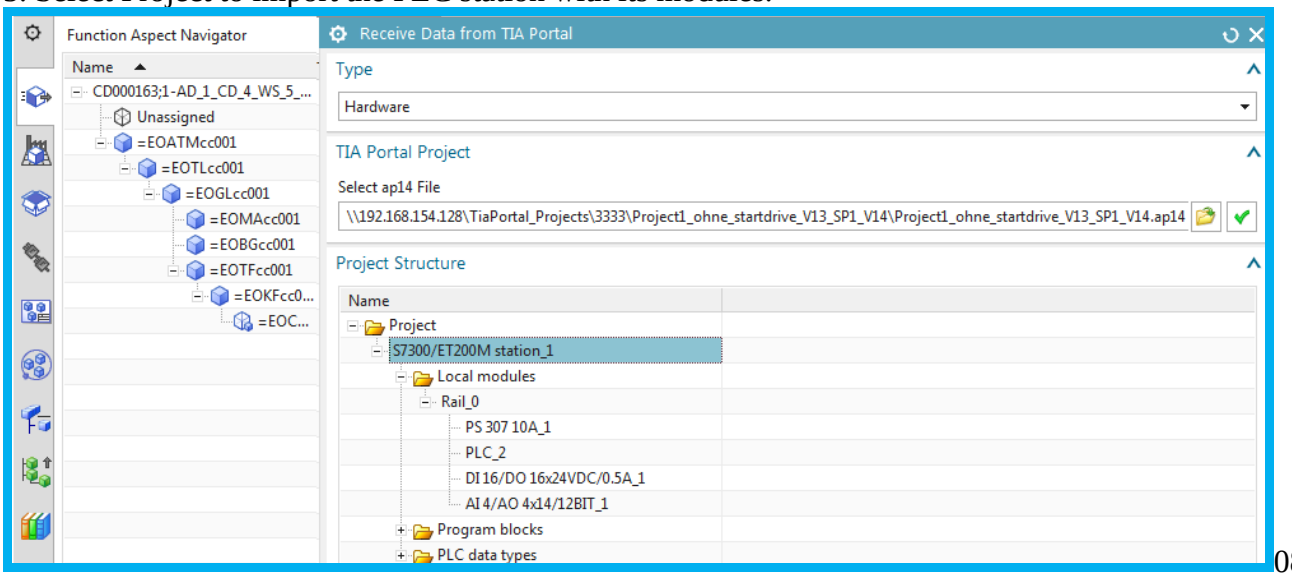
10.1. Receive hardware/software
10.2. Place FB's in aspect and create IDB's
10.3. Add/delete tags
10.4. Create TL constant value
10.5. Dynamize software
10.6. Assign software to hardware
10.7. Generate TIA Portal

10.1. Receive hardware, software

10.1.1. Receive hardware

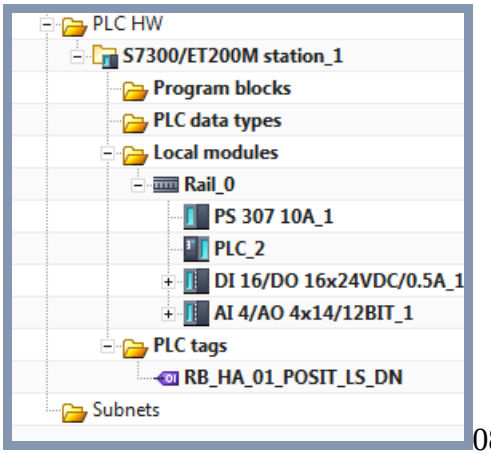
From STEP 7 or WinCC V13 SP1 onward, TIA Portal Openness is included in the delivery of STEP 7 or WinCC in TIA Portal. This enables you to program the applications which automate the engineering in TIA Portal.

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.



08b_01

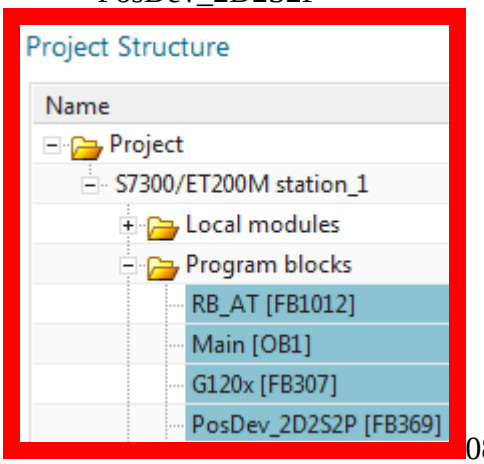
6. Click **Receive from TIA Portal**. The station is imported.



08b_02

10.1.2. Import software

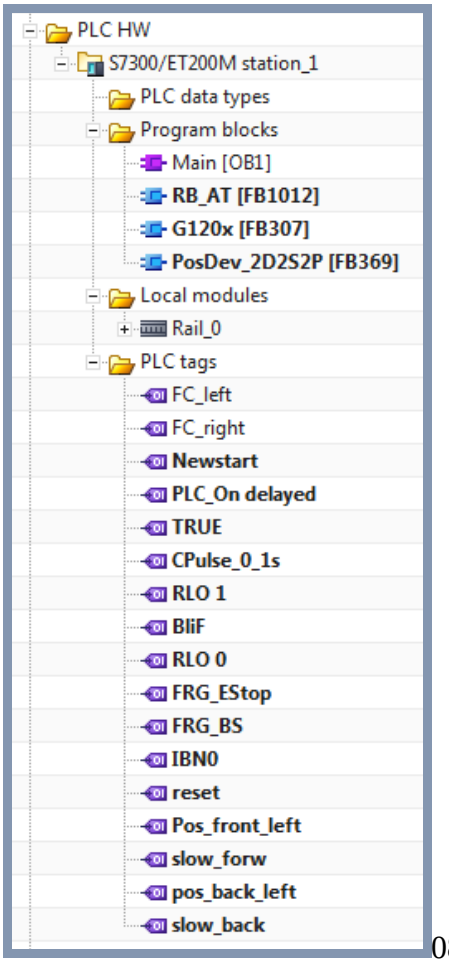
1. For **Type** select software.
2. Select the following blocks
   - Main [OB1]
   - RB_AT
   - G120x
   - PosDev_2D2S2P



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.

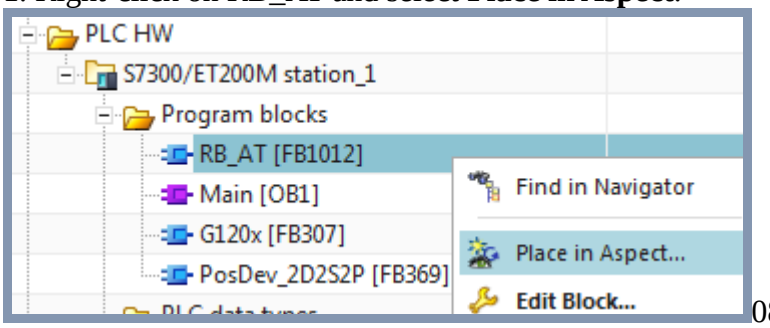


08b_03

10.2. Place the function blocks in aspects and create IDBs

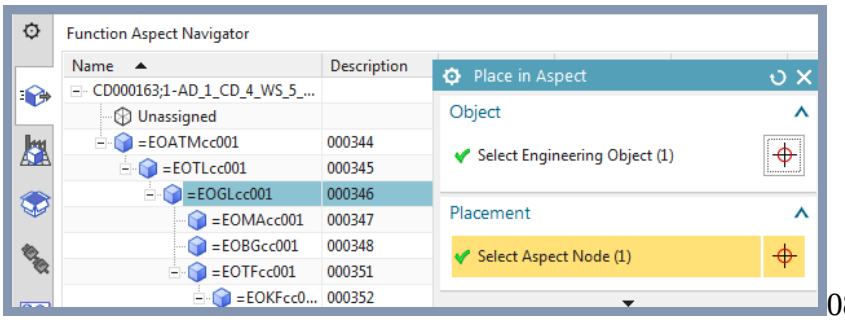
You now need to place the function blocks in the aspects and create IDBs in the aspects. Where you place the function blocks determines the aspect chain that will be used to create unique indentifiers (symbolic names). In this Getting Started you focus only on the Function aspect.

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



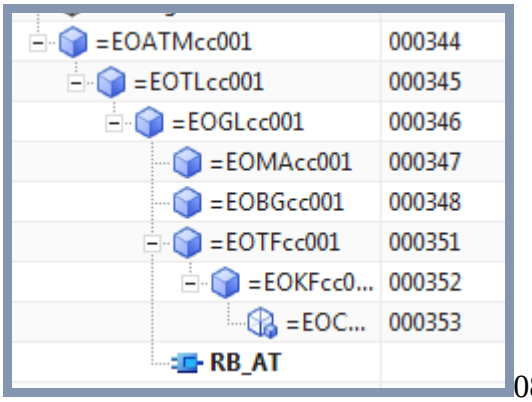
08b_04

2. For Placement select GL.



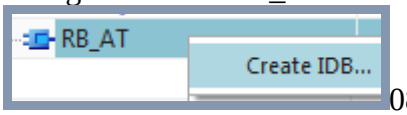
08b_05

3. Click **OK**.



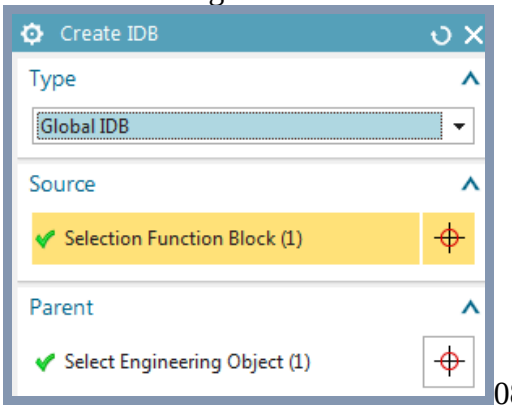
08b_06

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



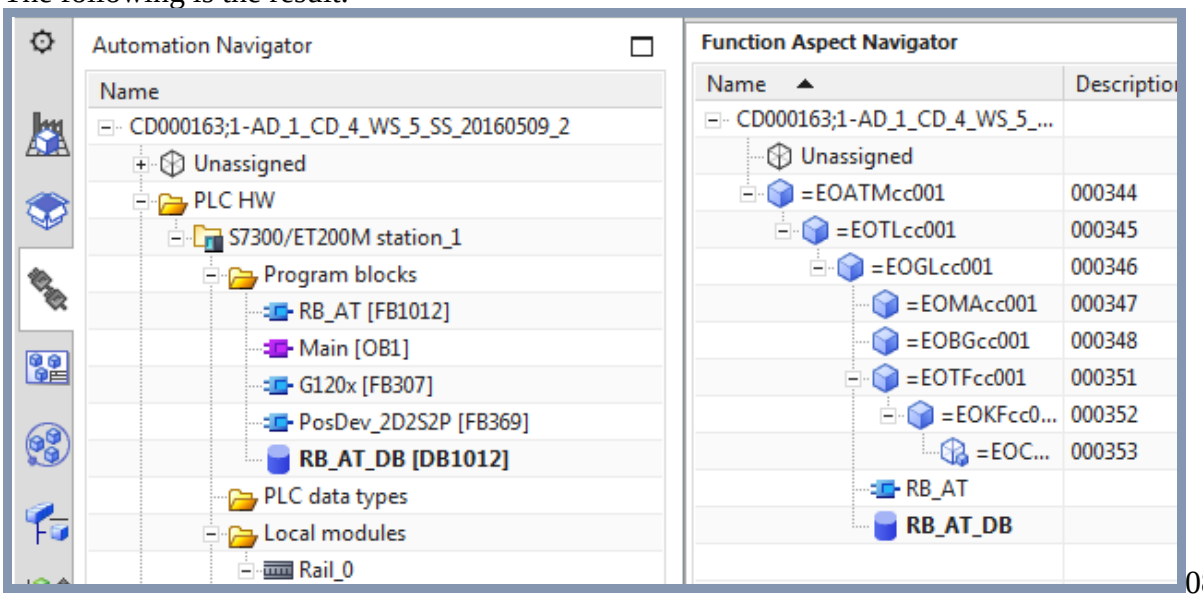
08b_07

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



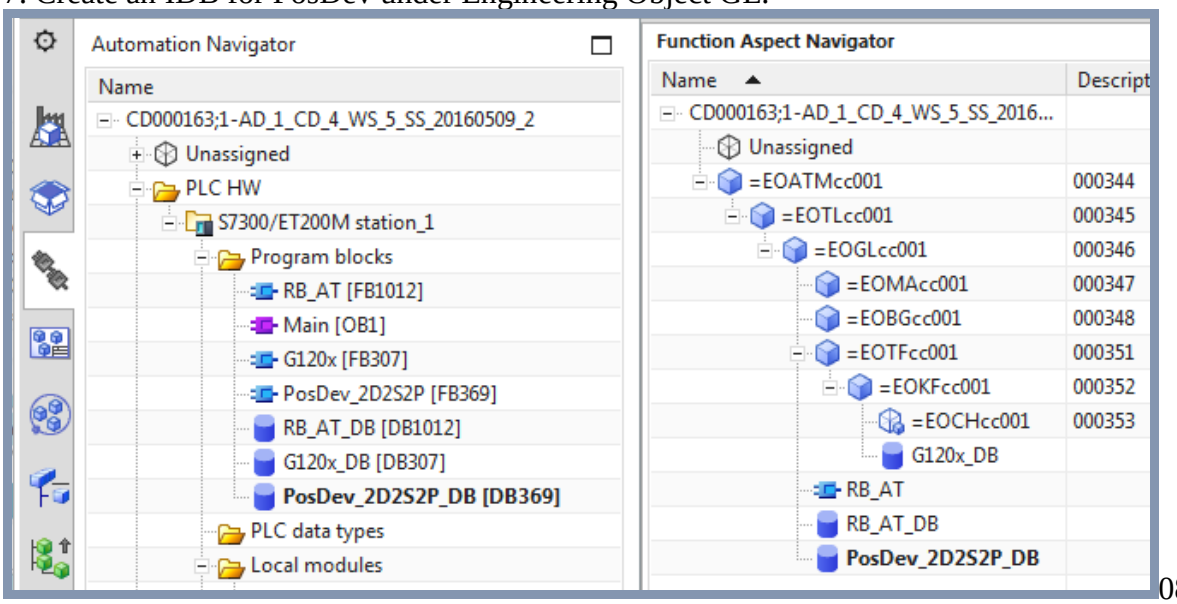
08b_08

The following is the result.



08b_09

6. Create an IDB for G120x under Engineering Object KF.
7. Create an IDB for PosDev under Engineering Object GL.



08b_10

10.3. Add tags

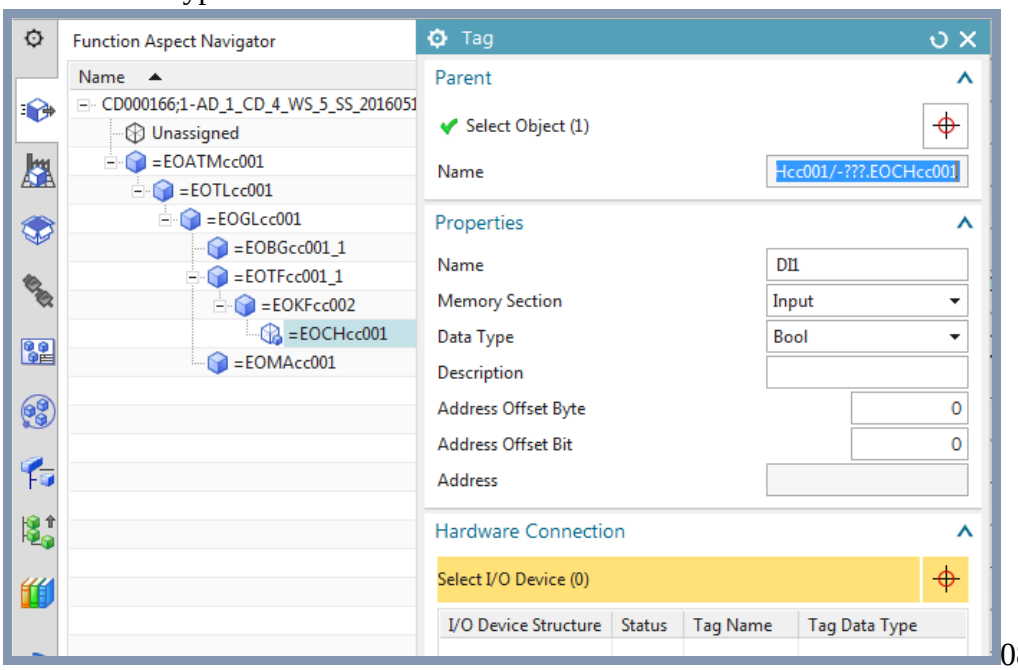
For this Getting Started we only add 2 tags.

8.2.1. DI1
8.2.2. PID0

10.3.1. DI1

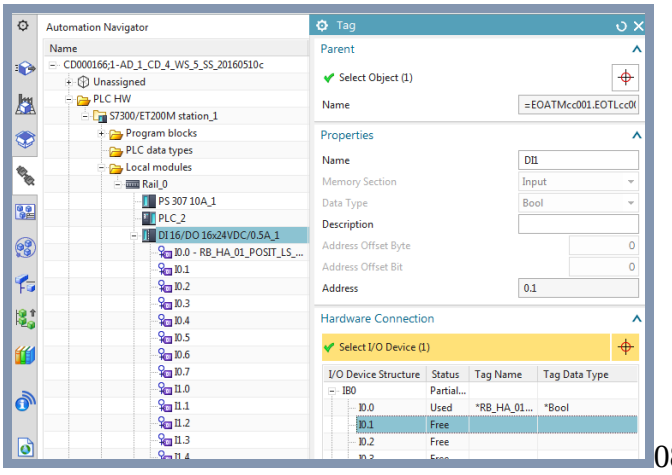
Add the sensor tag DI1 (boolean input).

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



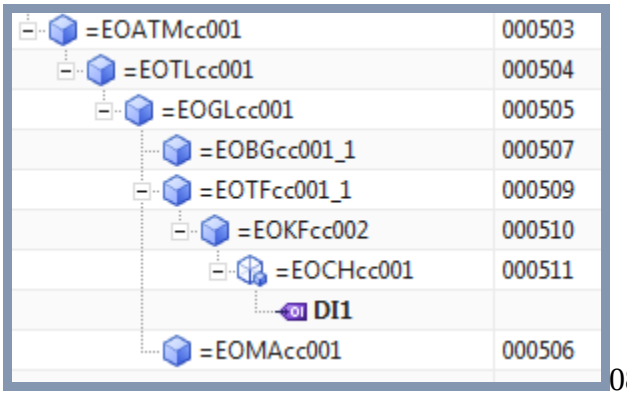
08b_11

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.



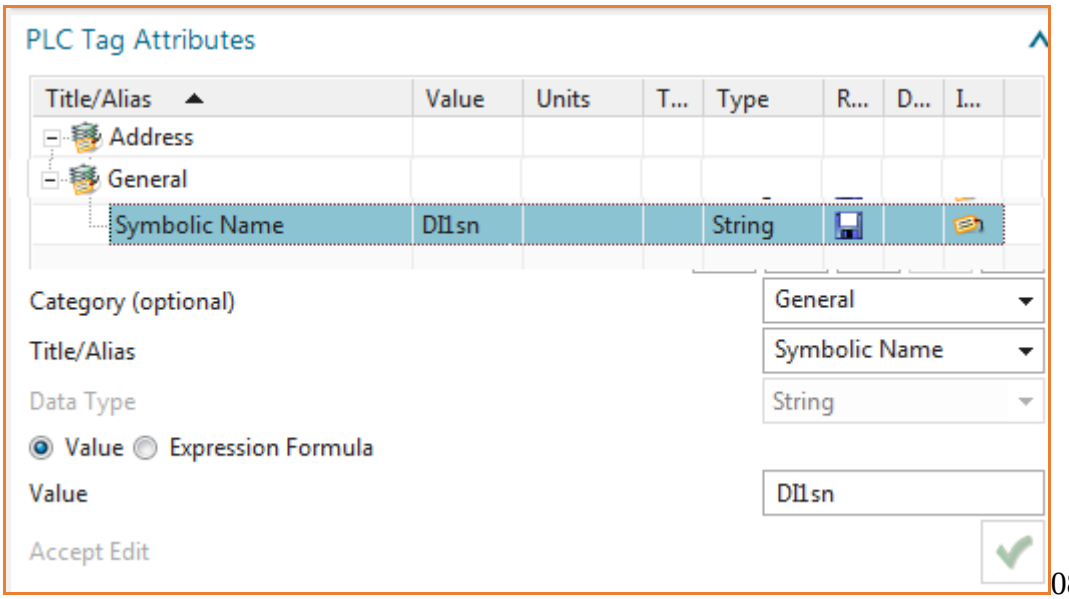
08b_12

6. Click **OK**.



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).

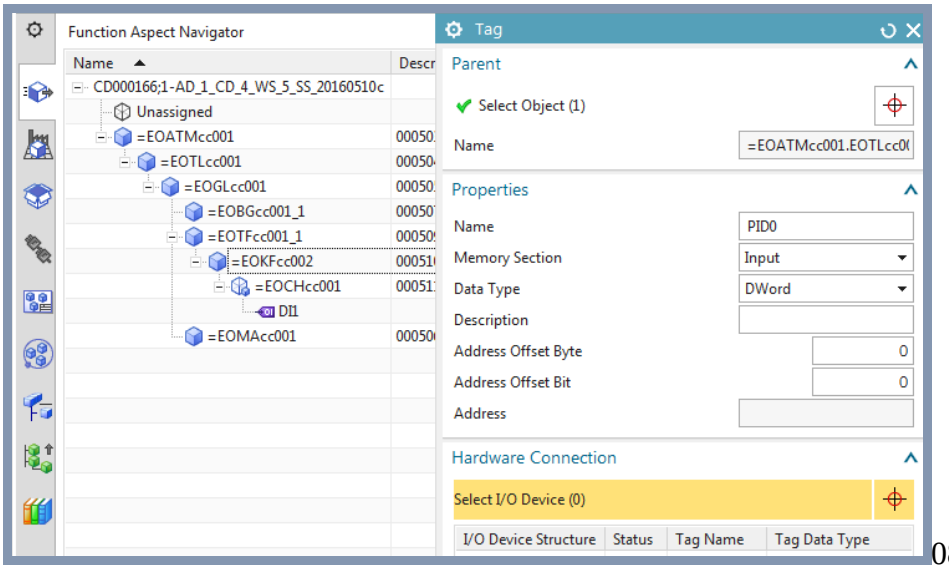


08b_14

10.3.2. PID0

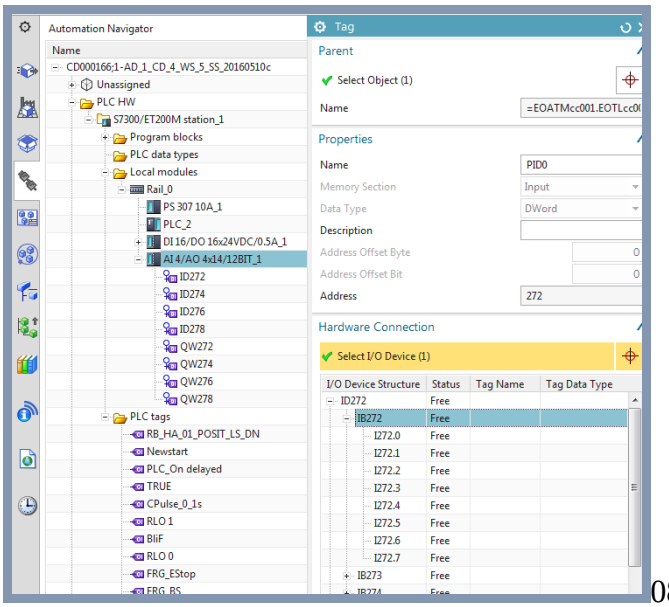
Add the tag PID0 (digital input).

1. Click **Tag**.
2. Specify the tag properties:
   - Parent Engineering Object = KF
   - Name = PID0
   - Memory section = Input
   - Data Type = DWord



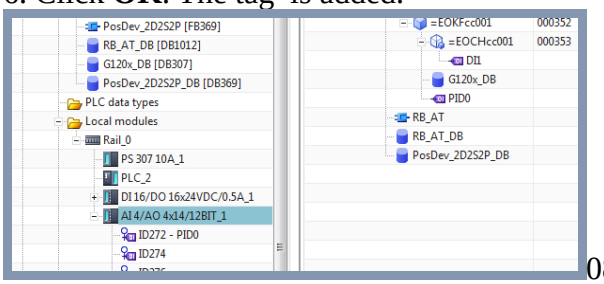
08b_15

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



08b_16

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

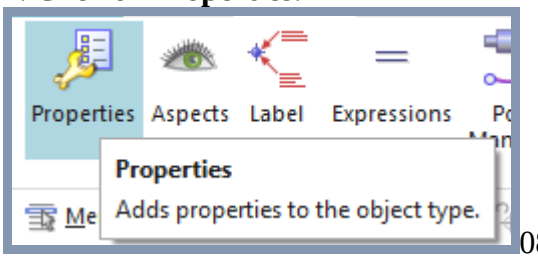


08b_17

10.4. Create TL constant value

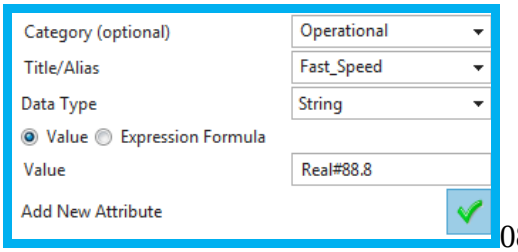
The constant values in the top aspect Engineering Object TL can be referenced in Engineering Objects. This allows you to modify the constant value and the value changes for all Engineering Objects that reference that value.

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



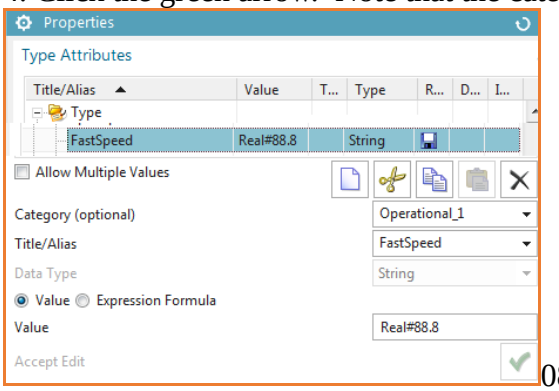
08b_18

3. Create a property with following :
   - Category = Operational_1
   - Title/Alias = Fast_Speed
   - Data Type = String
   - Value = Real#88.8



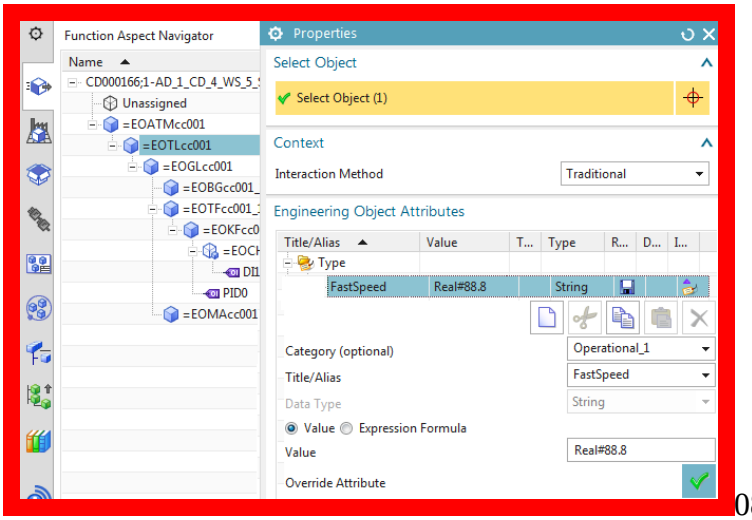
08b_19

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



08b_20

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



08c_02

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

10.5. Dynamize software

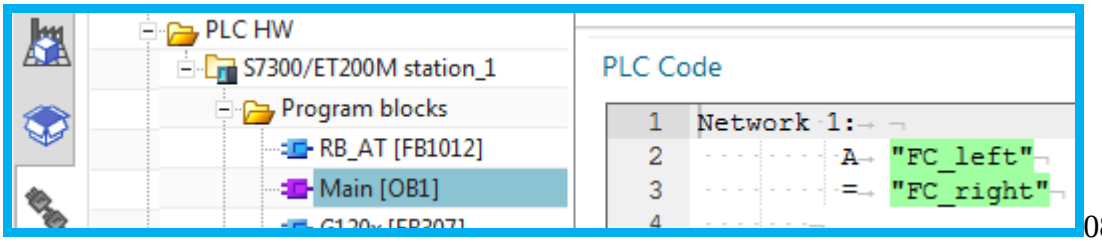
Dynamization of software means that the imported software is enhanced in such a way that it can be used in templates for auto-generation of software.

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

10.5.1. OB1->RB_AT_DB replace by call

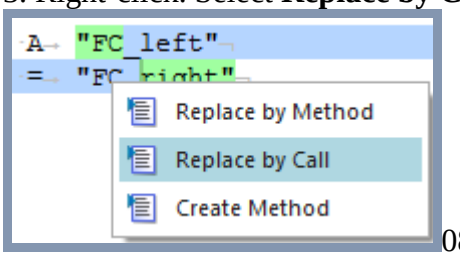
OB1 calls the RB_AT IDB.

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



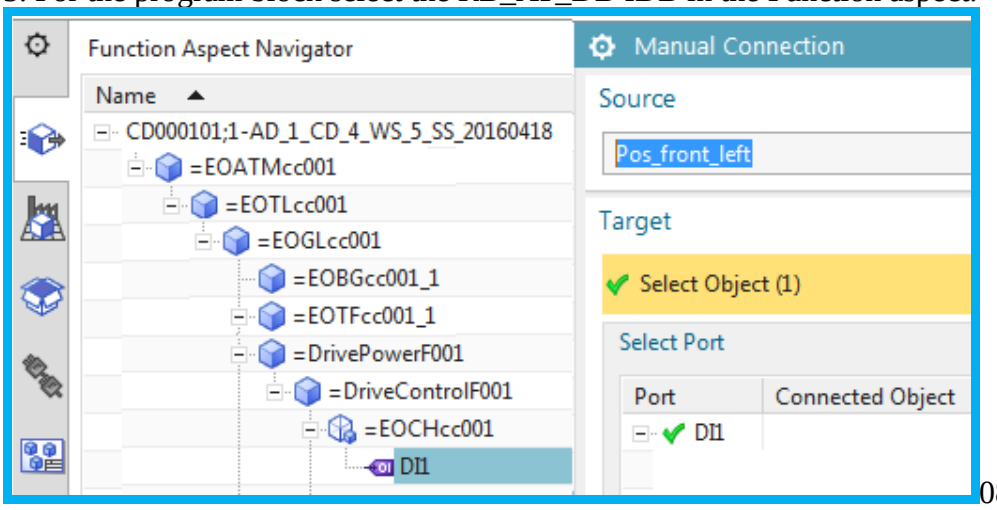
08b_22

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



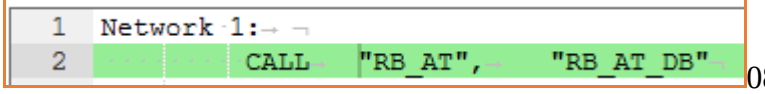
08b_23

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



08b_24

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

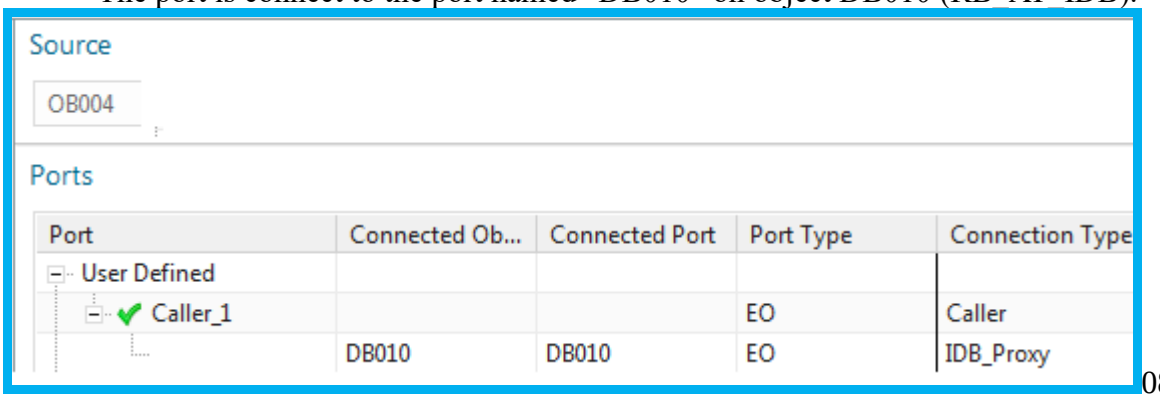


08b_25

7. Right-click on **OB Main**.
8. Select **Ports Manager**. A list of ports is displayed.

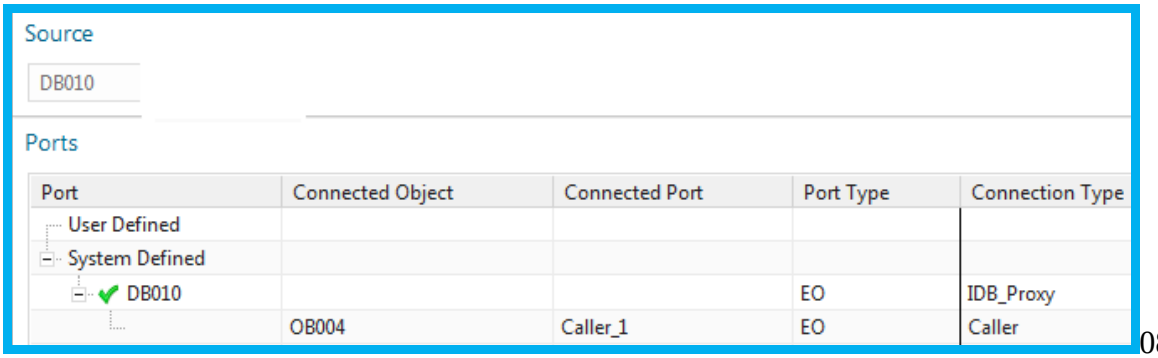
The following diagram shows the OB Main call to the RB_AT IDB.

- 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).



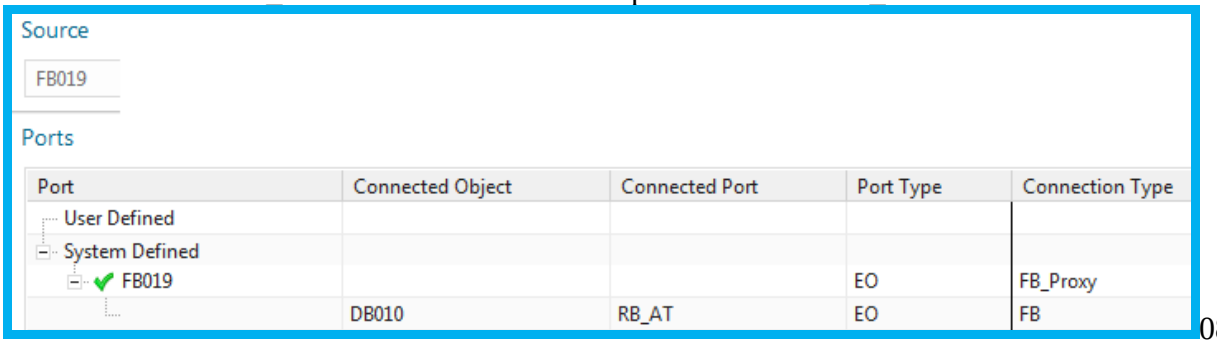
08b_26

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.



08b_27

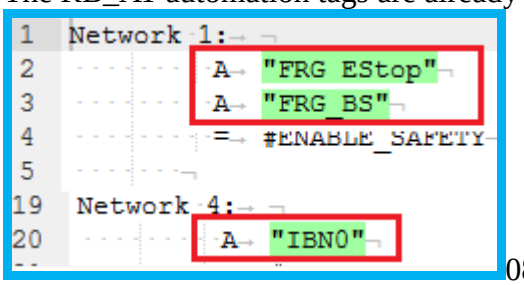
10. List the ports for the RB_AT FB. This shows the port that was created earlier when you created the IDB.
    - FB019 = RB_AT FB.
    - The port on FB019 is named “FB019” and is system-defined.
    - DB010 = RB_AT IDB and the connected port is named “RB_AT”.



08b_28

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

The RB_AT automation tags are already connected.



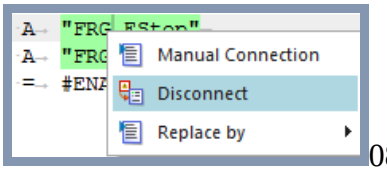
08b_29

The following shows the ports manager for RB_AT.

08b_30

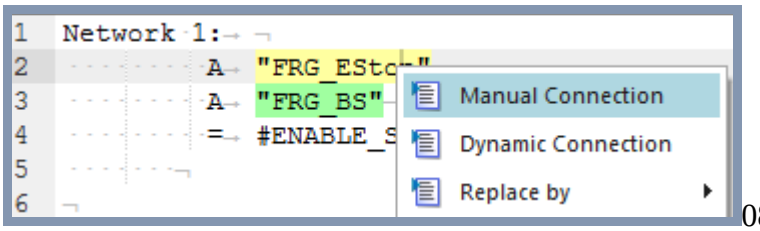
The following shows you how to make this connection.

1. Disconnect.



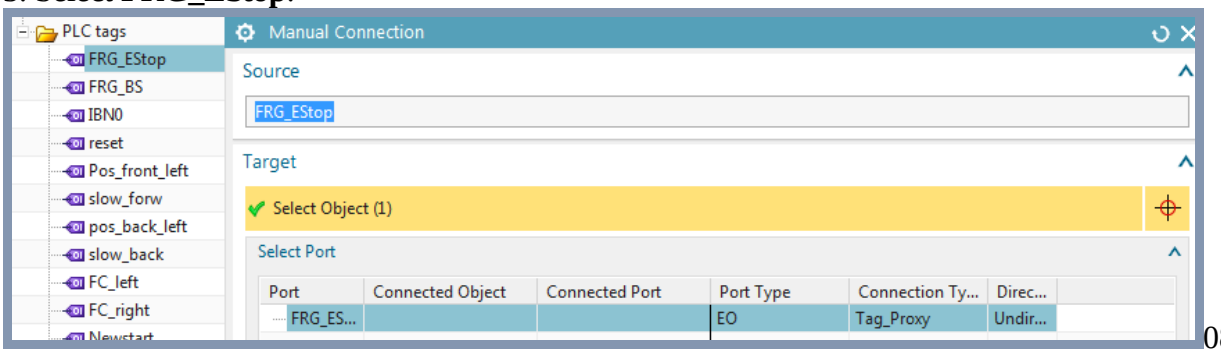
08b_31

2. Select **Manual Connection**.



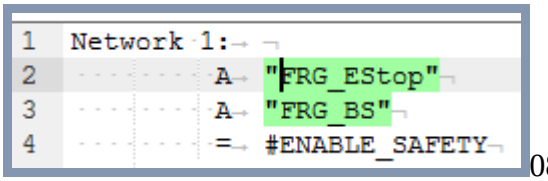
08b_32

3. Select **FRG_EStop**.



08b_33

4. Click **OK**.

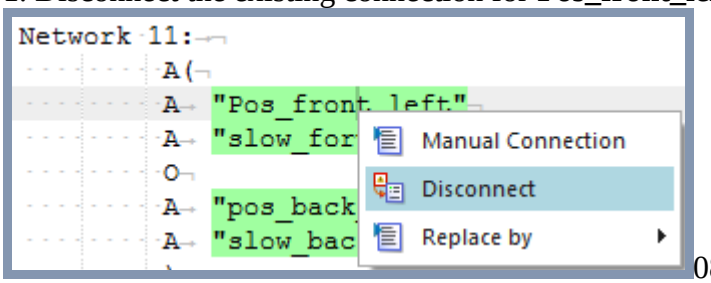


08b_34

10.5.3. RB_AT manual connect to CH DI tags

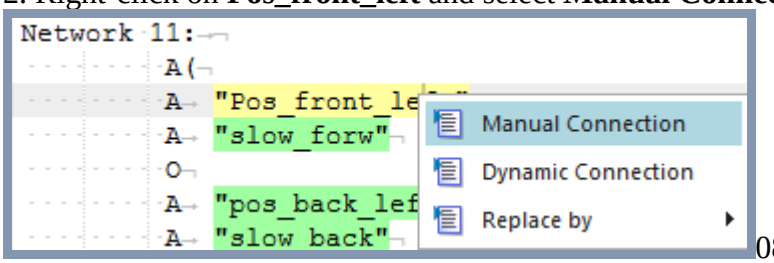
This section shows you how to make a connection to the channel tag.

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



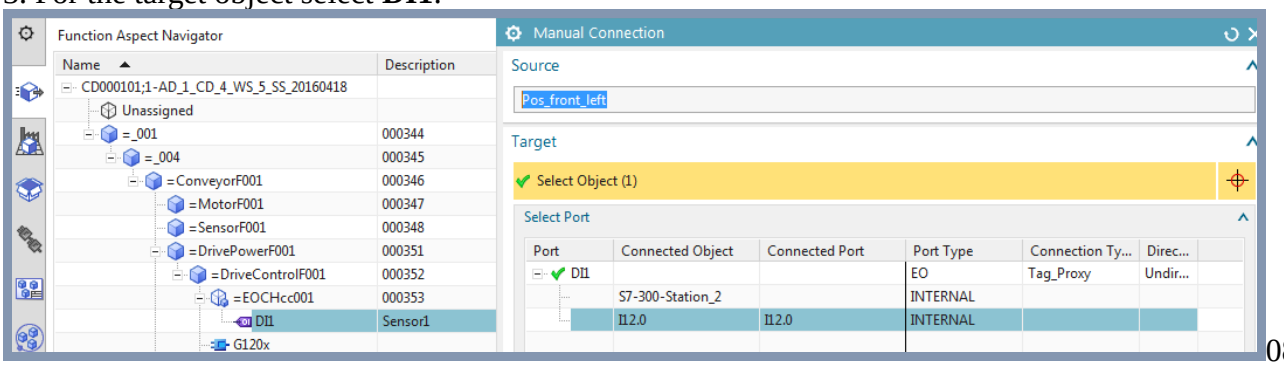
08b_35

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



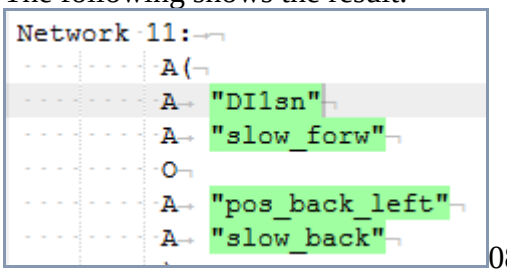
08b_36

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



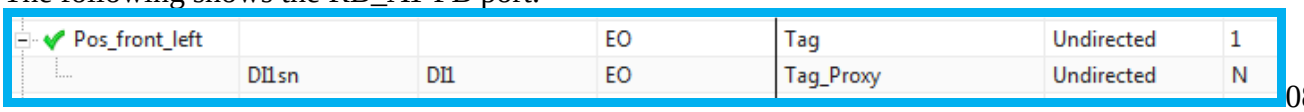
08b_37

The following shows the result.



08b_38

The following shows the RB_AT FB port.

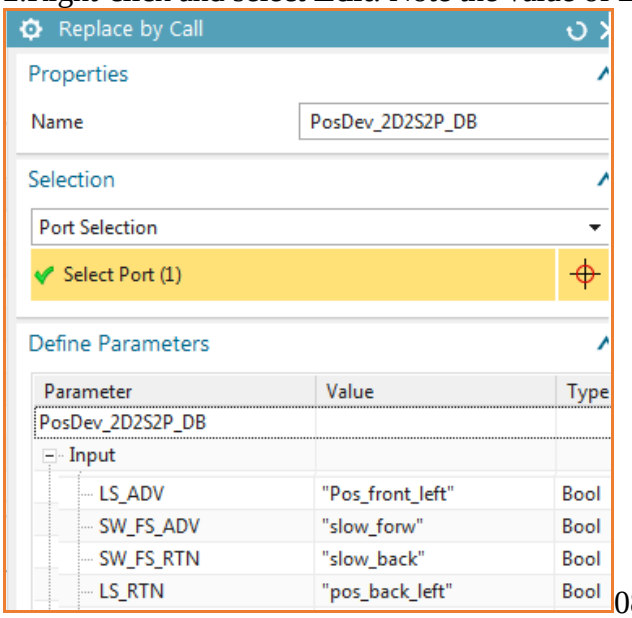


08b_39

10.5.4. RB_AT->PosDev replace by call

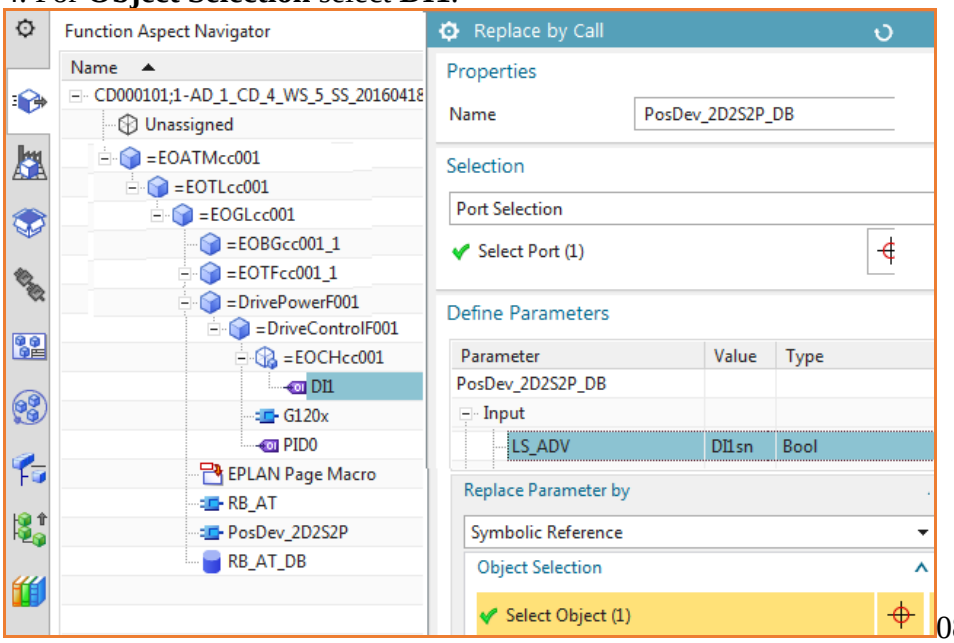
You now want to create your first replace by call. You use this to create a call from RB_AT to Pos_Dev. You will also reference the DI tag in a call.

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



08b_40

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



08b_41

5. Click **OK**.

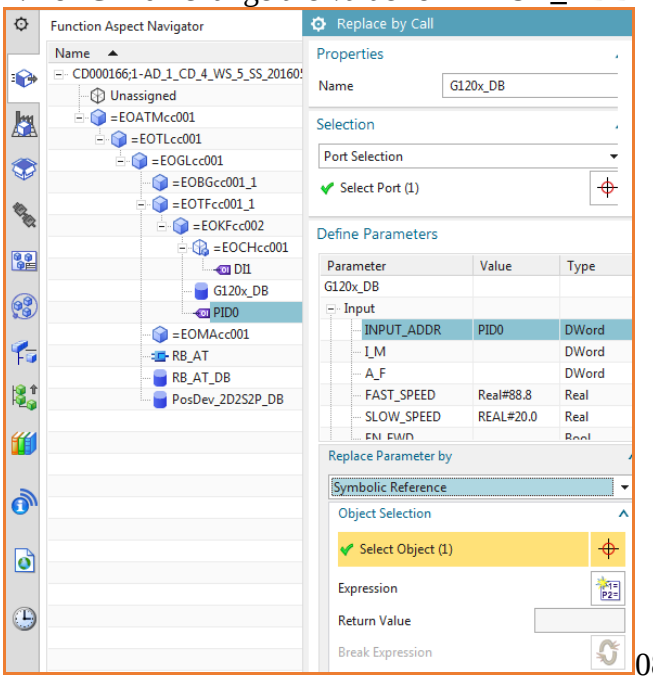
```
CALL "PosDev_2D2S2P", "PosDev_2D2S2P_DB"
   LS_ADV := "DI1sn"
   SW_FS_ADV := "slow_forw"
   SW_FS_RTN := "slow_back"
   LS_RTN := "pos_back_left"
```

08b_42

10.5.5. RB_AT->G120x replace by call

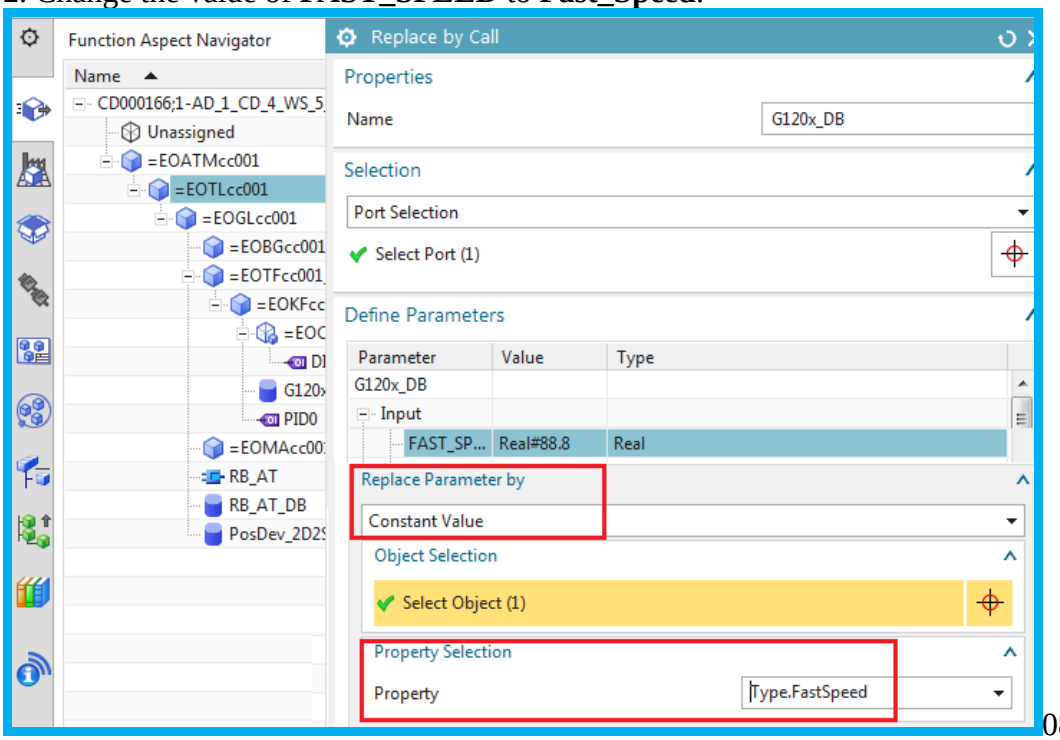
You now want to create your second replace by call. You use this to create a call from RB_AT to G120x. You will also reference a the PID0 tag and the constant value you created earlier.

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



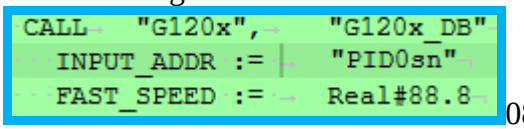
08b_43

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



08b_44

The following shows the result.



08b_45

10.5.6. 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"
```

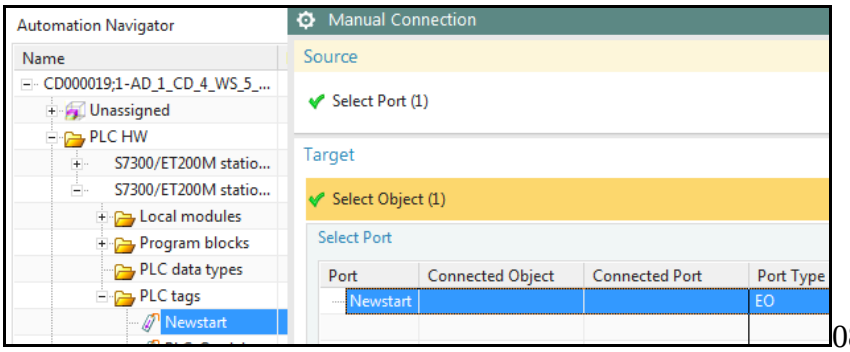
08b_46

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

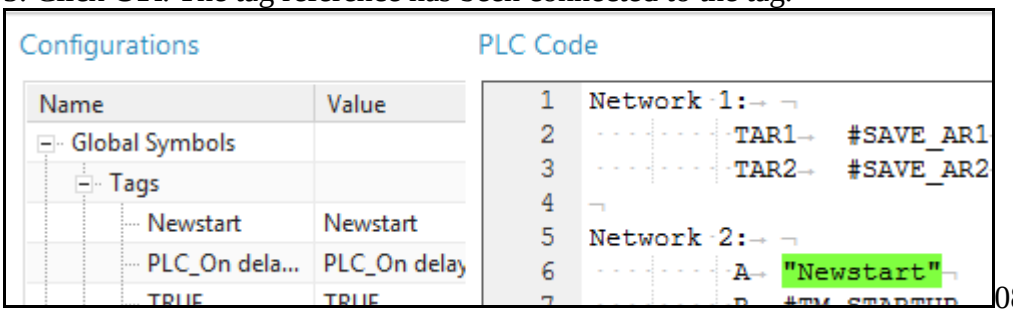
If they are not connected, the do the following:

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



08b_48

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

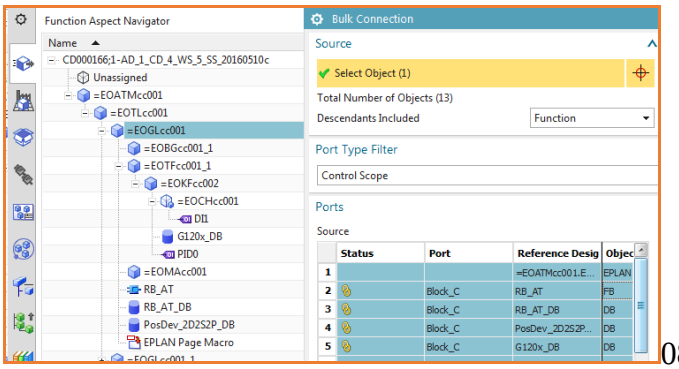


08b_49

10.6. Assign software to hardware

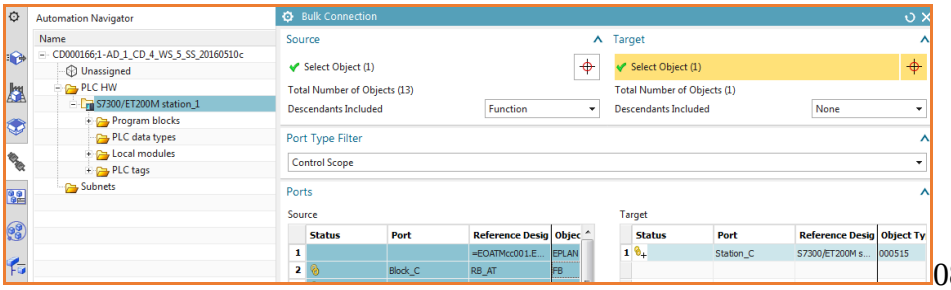
You have created the software blocks. But you still need to assign these blocks to your hardware in the automation tab. After this you can export to TIA Portal.

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



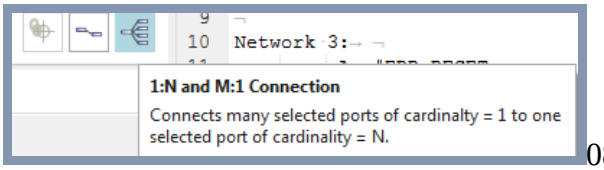
08b_50

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



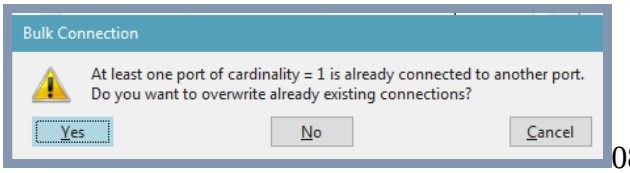
08b_51

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



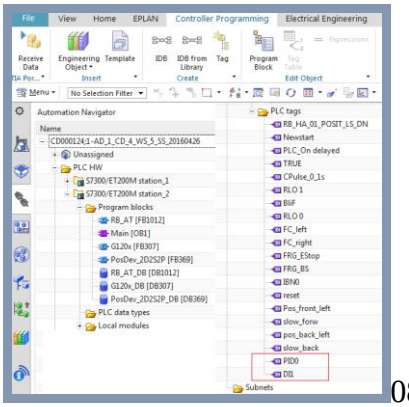
08b_52

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



08b_53

The following shows the result.

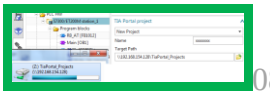


08b_54

10.7. 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

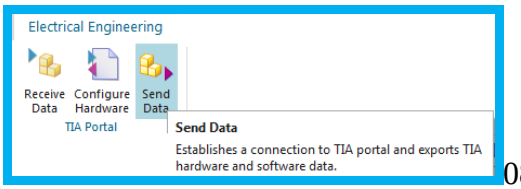
Note: Mapped drive to the VM must be connected or get error.


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.

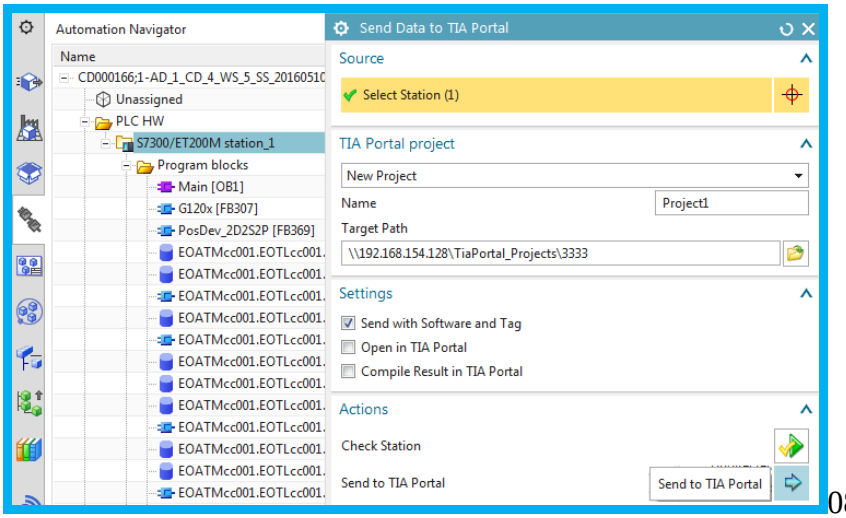
You now want to send the data to TIA portal.

1. Click **Send Data**.



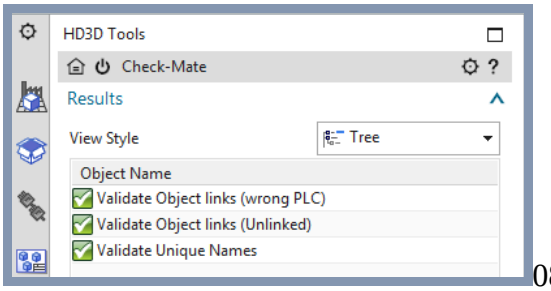
08b_56

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**.



08b_57

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



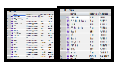
08b_58

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

Always get this eror. 

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

11. Synchronize changes (no templates)

Should add a chapter about how to sync changes.

Part 3. Create/instantiate template

12. Template-related concepts
13. Configure a template-ready AD project for EPLAN
14. Configure a template-ready AD project for TIA Portal
15. Create/instantiate template
16. Synchronize changes (with templates)

12. Template-related concepts

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

12.1. 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.

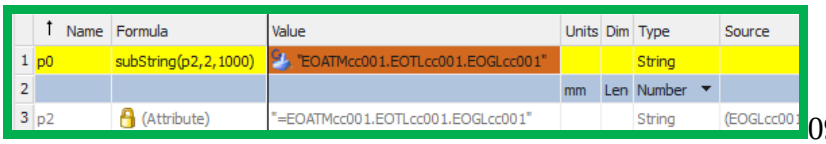
EPLAN

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.

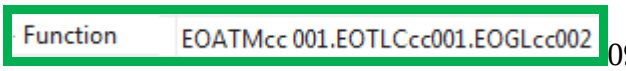
The solution is to use expressions to make the value of the **Function** property depend on the Engineering Object (GL) the macro was placed under in the aspect tree. Thus the property value is unique. This is not required to generate a report, but is required to avoid confusion when actually running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

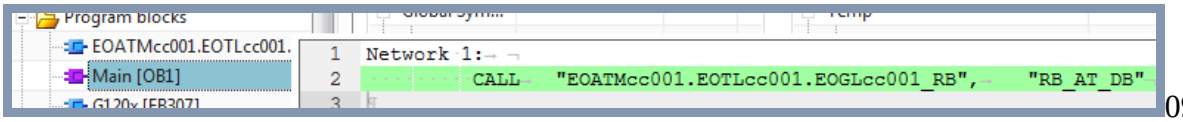
Previously you created only one conveyor, so all software block and tag names were globally unique. You did not need to use the aspect chain. But if you add or copy a conveyor with same values, the symbolic name of software blocks and tags will be repeated (which will generate an error in TIA Portal).

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

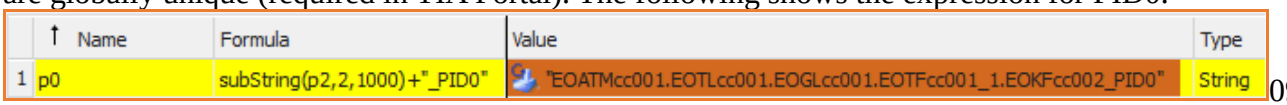
09b_03

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



09b_04

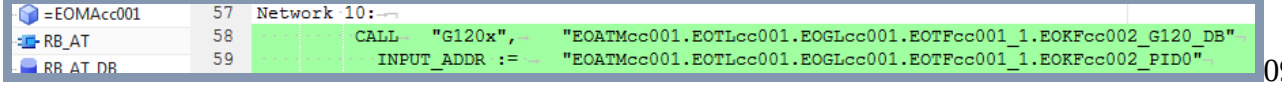
The solution for tags is to use expressions to make the symbolic name of the tags (PID0 and DI1) depend on the Engineering Objects (KF and CH) they were placed under in the aspect tree. Thus the symbolic names are globally unique (required in TIA Portal). The following shows the expression for PID0.



	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.

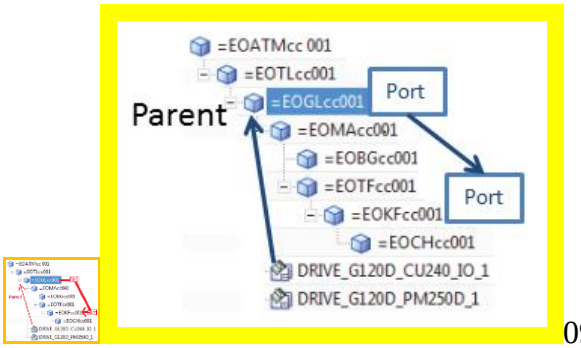


09b_07

12.2. 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

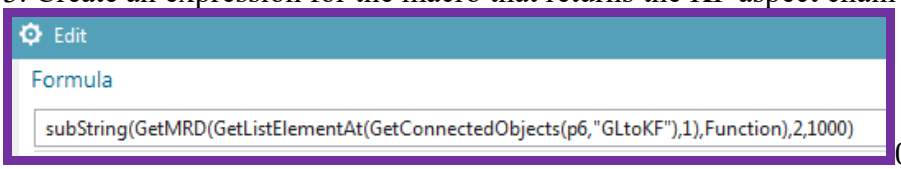
To configure this you do the following:

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

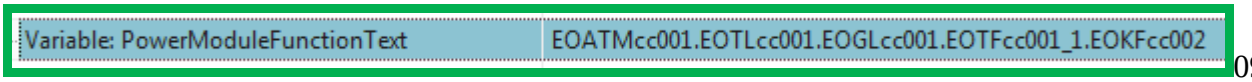
~~In 7.2 "Add PM250 macro" you simply entered text for device names.~~ Since the Engineering Objects representing the devices are not the parent Engineering Objects of the macro, you do not have a link to the Engineering Objects to retrieve their Multi Reference Designation. ~~In 13.2 "KF01.name ports, 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).



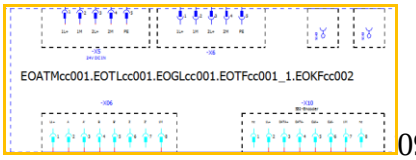
09b_11

The following shows the resulting property value.



09b_13

The following shows the result in the generated macro report.



09b_14

TIA Portal (software blocks and tags)

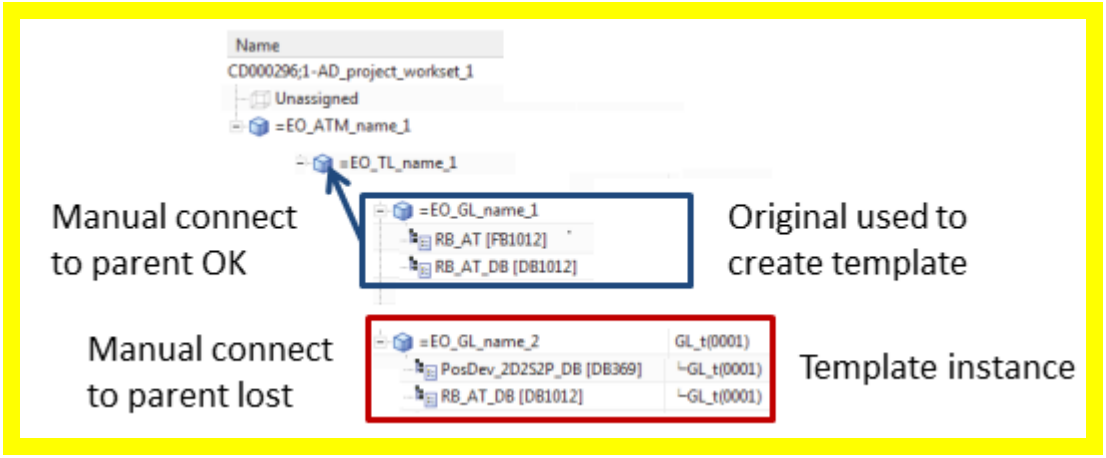
In this Getting Started you don't have an example where a software block uses a port to access the aspect ID of a non-parent Engineering Object to use as the symbolic name. Normally you would not do this.
For tags you might want to have the symbolic name for a tag set to the aspect chain of a non-parent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

12.3. 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).

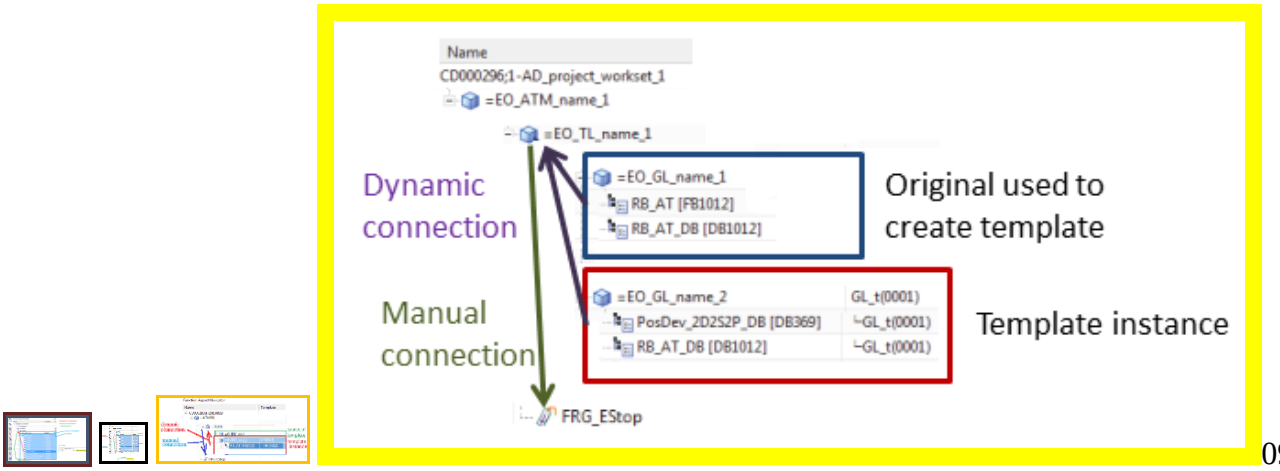


09b_15

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 (14.2)

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

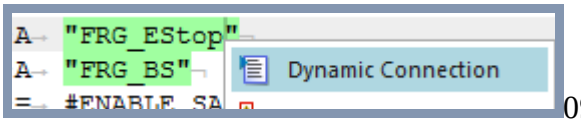
3. Create the dynamic connection from RBAT FB to the tag via the TL port using the following expressions (P6 is RB_AT FB object name).

	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.



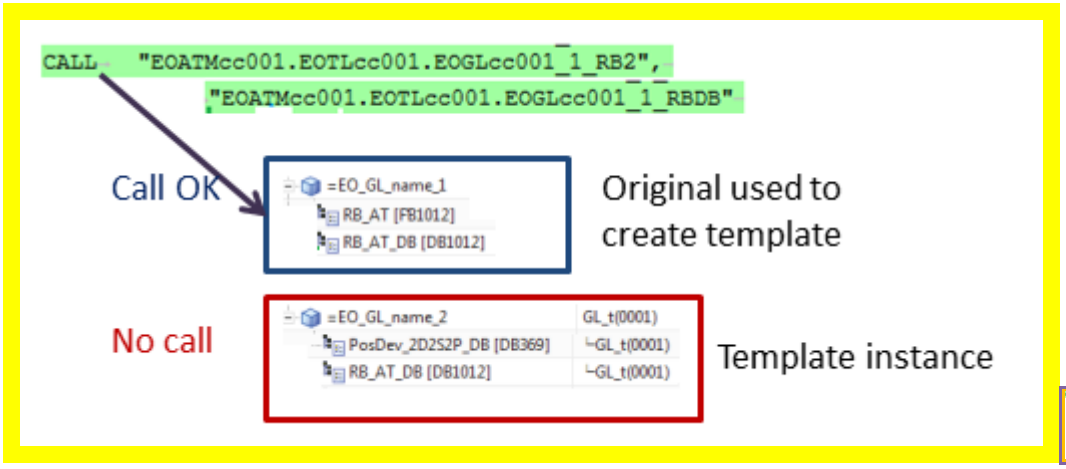
09b_19

12.4. 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.




09b_20xx

Solution (14.3)

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

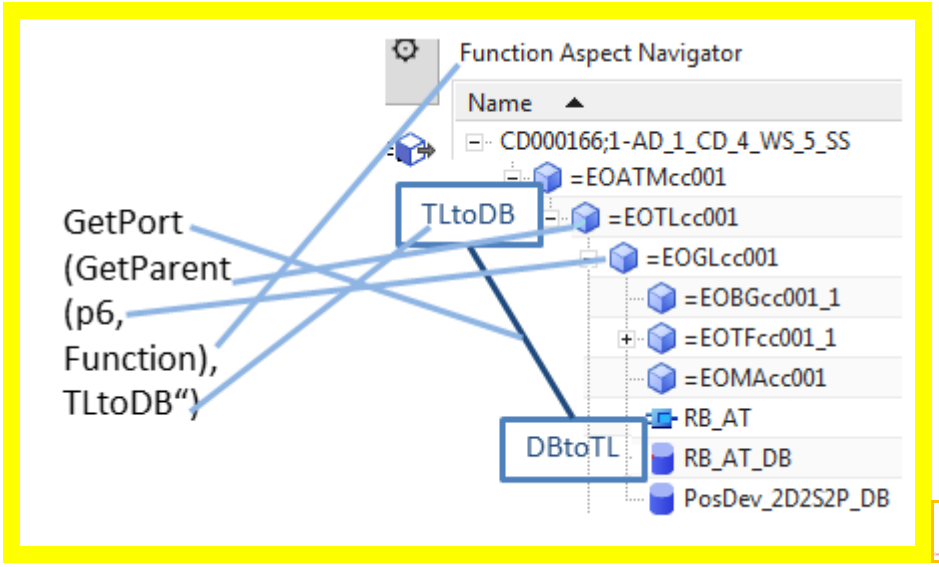
1. Define ports TLtoDB (in TL) and DBtoTL (in RBAT IDB).
2. In RBAT DB create a connection between the ports with the following expressions (p6 is the GL object name).

	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

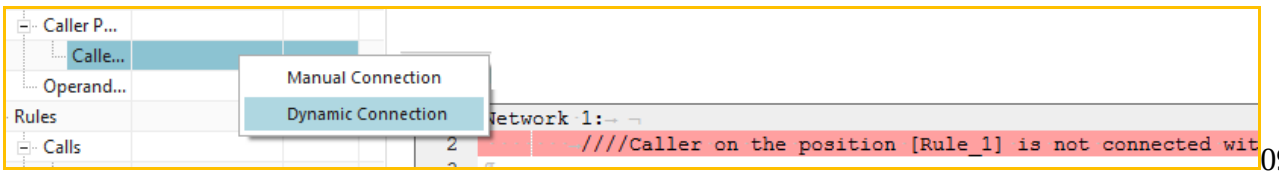

09b_23

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.

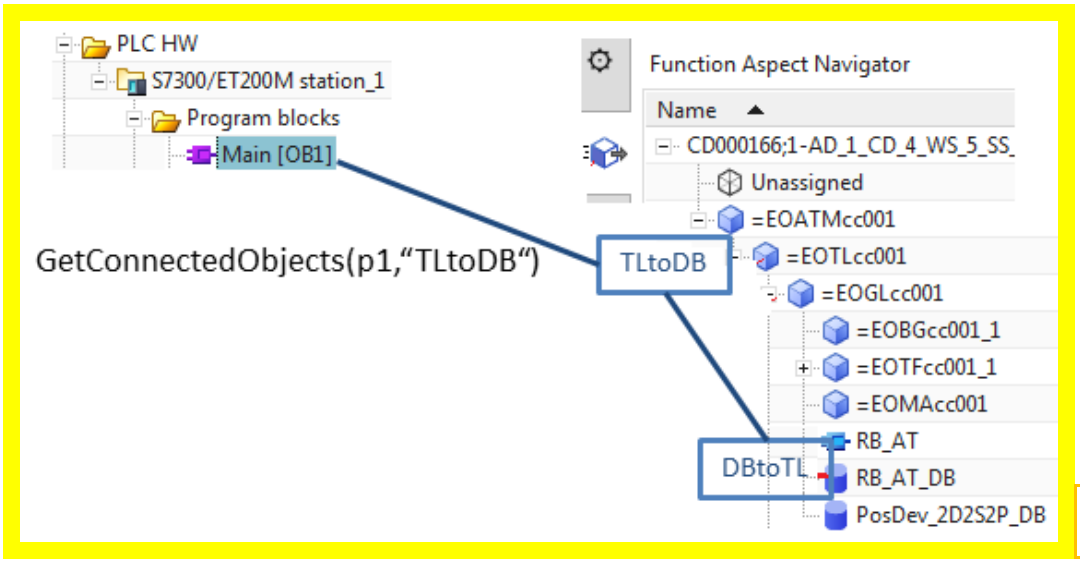


09b_25

	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.



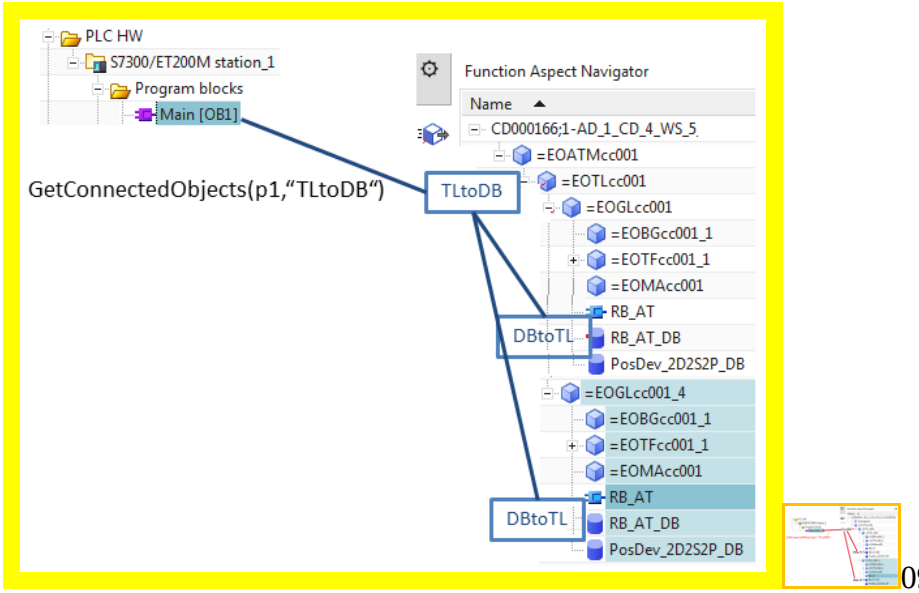
09b_24

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

13. Configure a template-ready AD project for EPLAN

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

13.1. Function expression

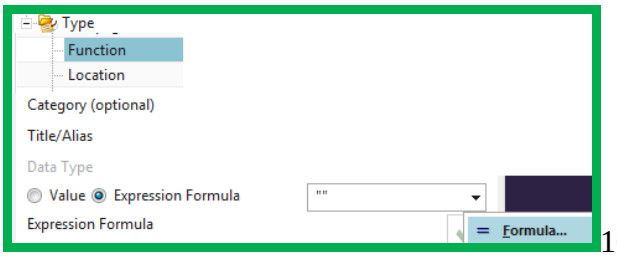
13.2. KF01.name ports, link, and expression

13.3. Test

13.1. Function expression

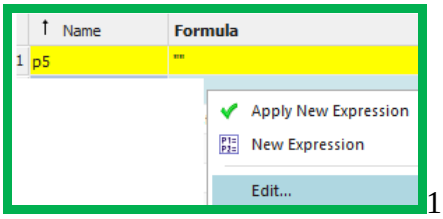
This section shows a simple example of how to use the parent Engineering Object aspect tree for the value of an EPLAN property.

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…**.



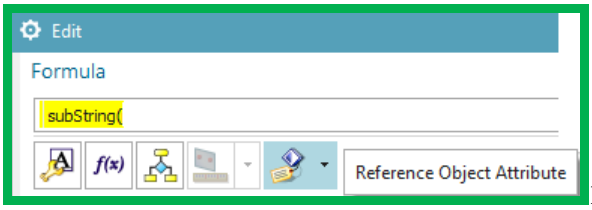
10b_01

2. Click **Edit**.



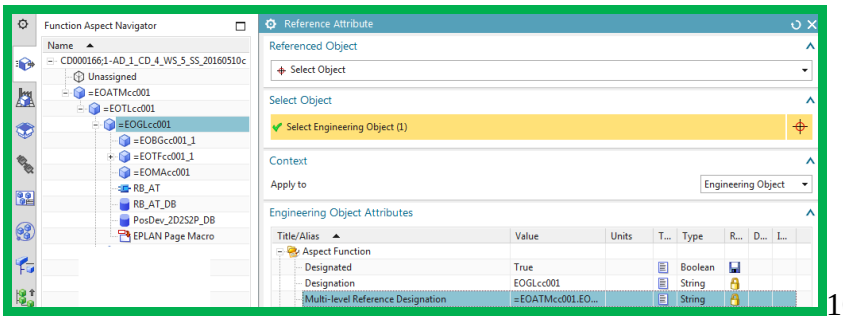
10b_02

3. Enter “subString(“ .



10b_03

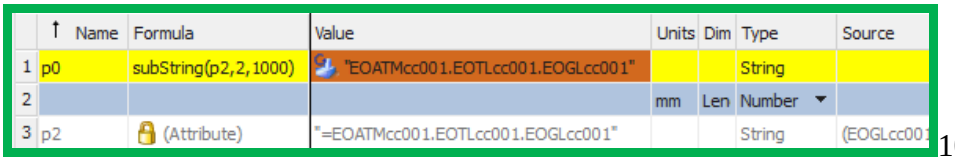
4. For **Reference Object Attribute** select in the Function aspect the Engineering Object GL **Multi Reference Designation**.



10b_04

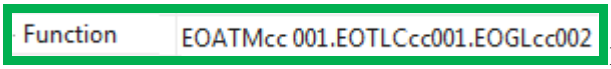
5. Add the the formula “,2,1000)”. This specifies to create a substring from character 2 and including the first 1000 characters.

6. Click **OK**.



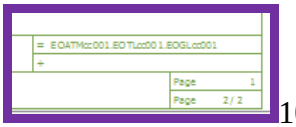
10b_05

The following shows the result.



10b_06

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



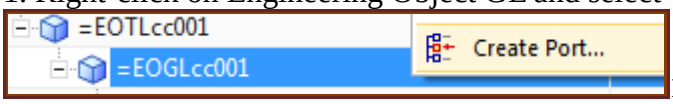
10b_23

13.2. KF01.name ports, link, and expression

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

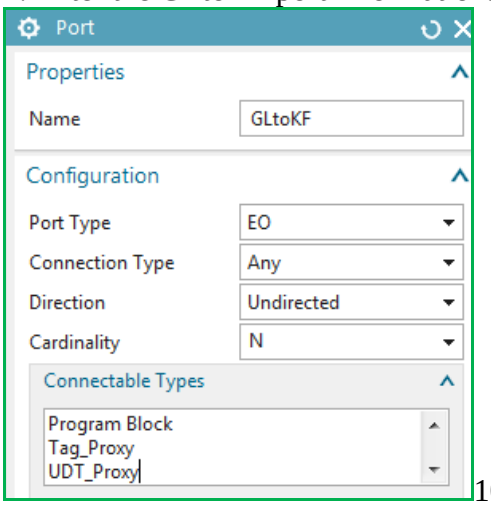
Ports

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



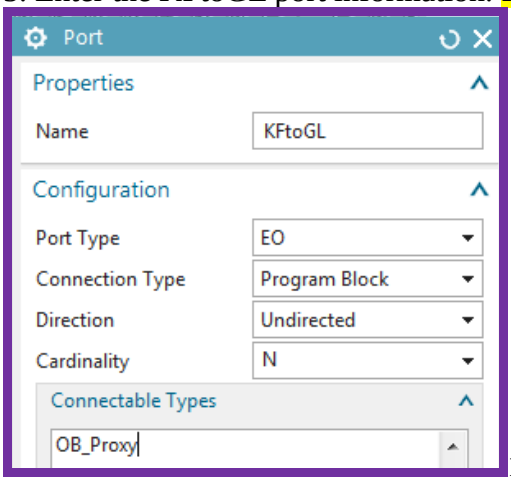
10b_07

2. Enter the GLtoKF port information.



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**.



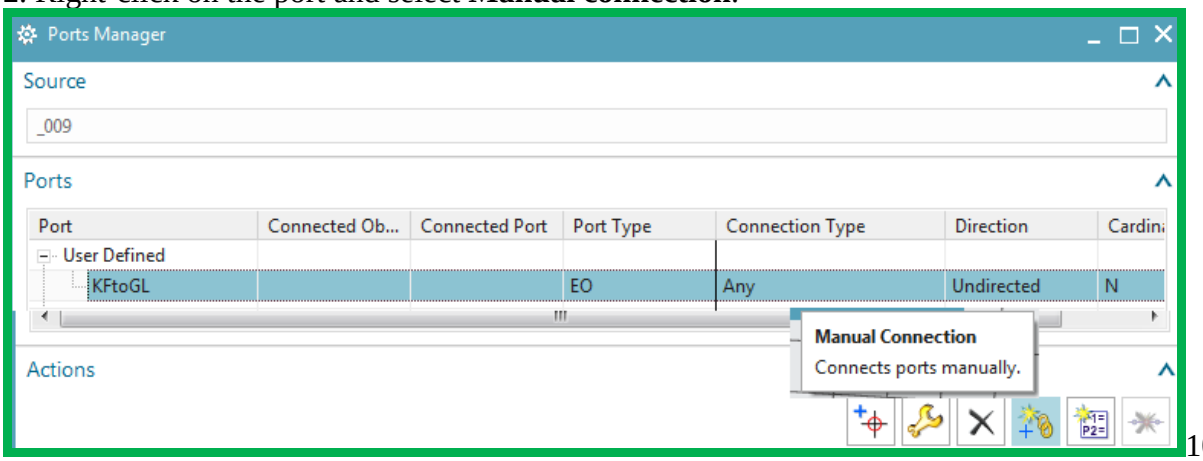
10b_09

6. Click **OK**.

Connection (link)

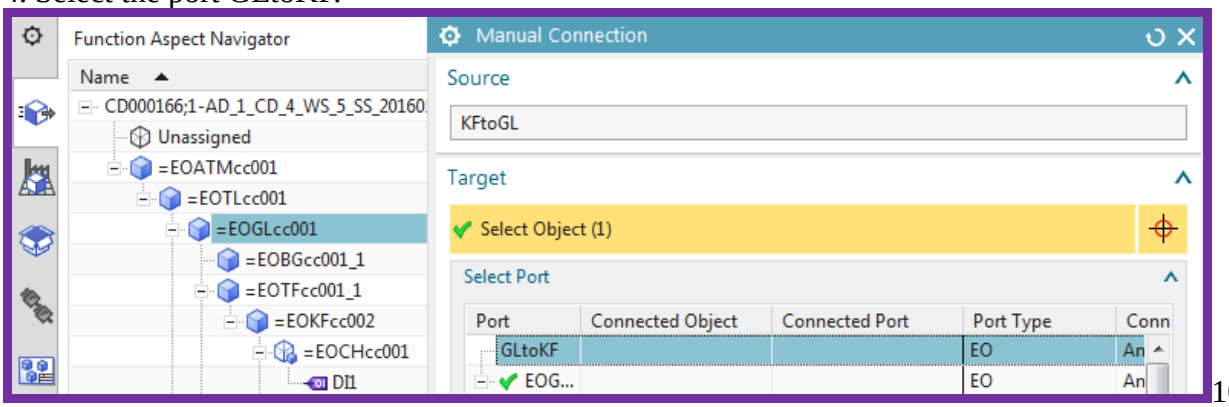
You now need to link the 2 ports with a manual connection, since this connection will be within the template you create later. The link can be created starting from either Engineering Object.

1. Right-click on Engineering Object KF and select **Ports manager…**.
2. Right-click on the port and select **Manual connection**.



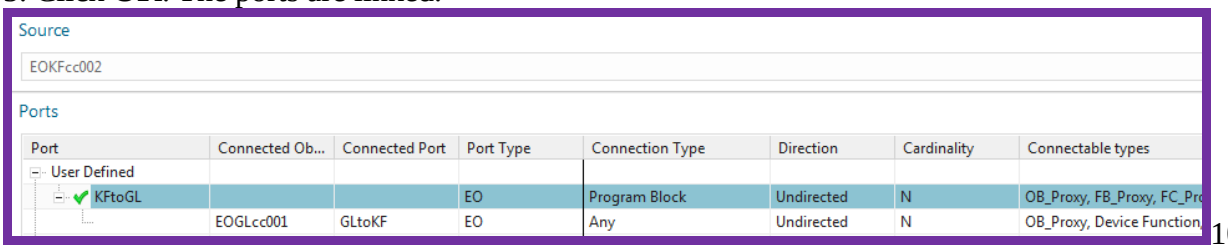
10b_10

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



10b_11

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

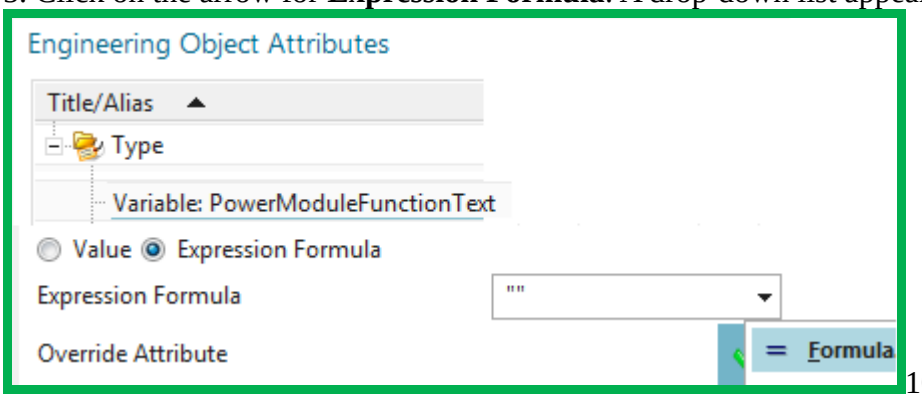


10b_12

Expression

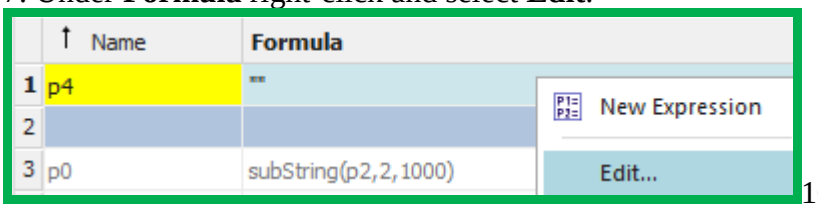
This section shows how to use an expression to get the Function aspect Multi Reference Designation of an Engineering Object that is not a parent of the macro and assign to a macro variable.

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



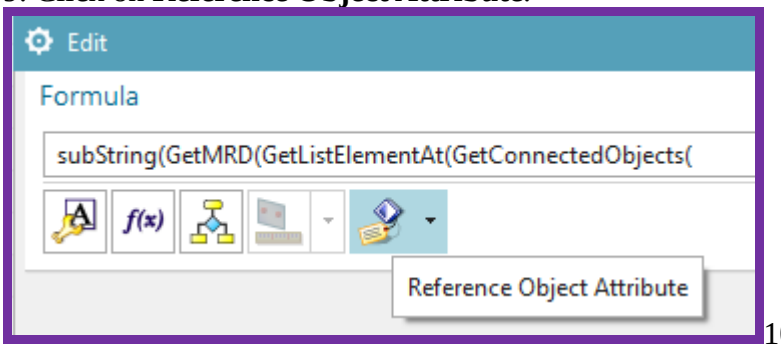
10b_13

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



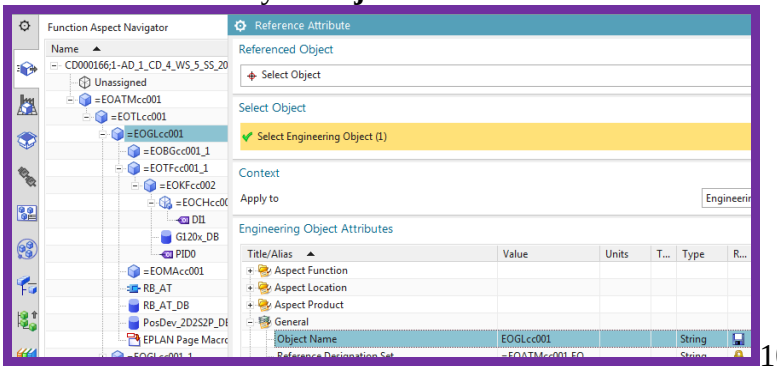
10b_14

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



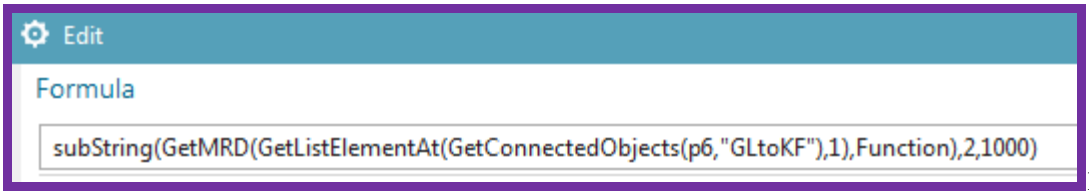
10b_15

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



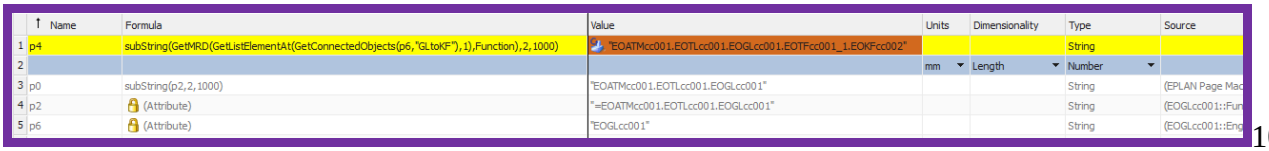
10b_16

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).



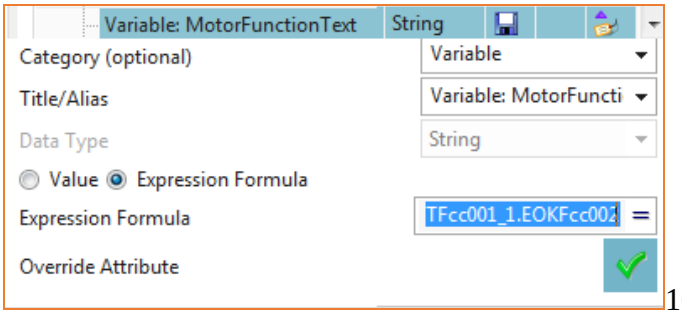
10b_17

12. Click **OK**.



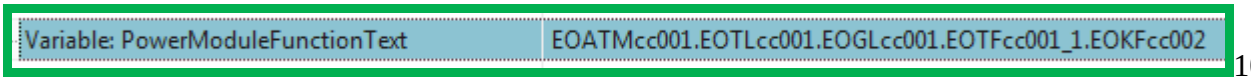
10b_18

13. Click **OK**.



10b_19

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

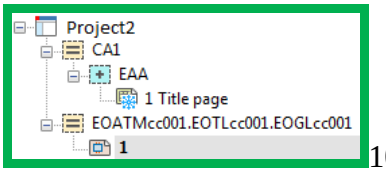


10b_20

15. Click **OK**.

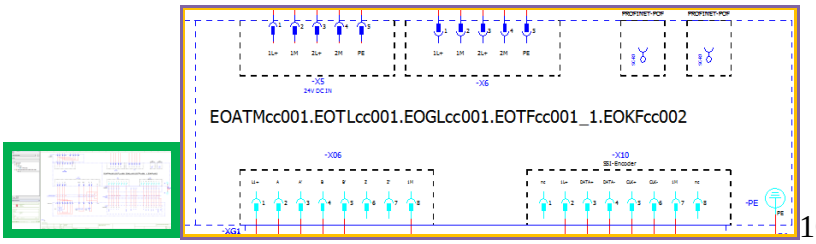
13.3. Generate

1. Generate EPLAN.



10b_21

Note the property for KF “MotorFunctionText”.



10b_22 xxxxxxxxxx

14. 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:

14.1. 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

14.1.1. RB_AT FB symbolic name
14.1.2. RB_AT IDB symbolic name
14.1.3. PosDev_DB symbolic name
14.1.4. G120 IDB symbolic name
14.1.5. PID0 symbolic name
14.1.6. DI1 symbolic name

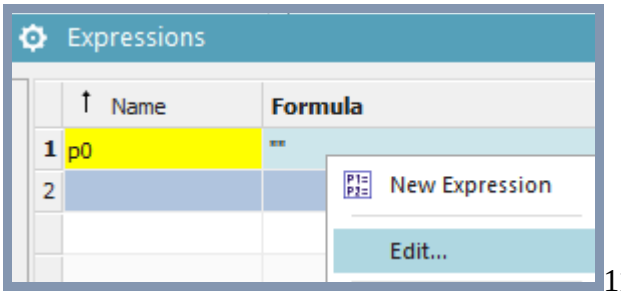
14.1.1. RB_AT FB symbolic name

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**.

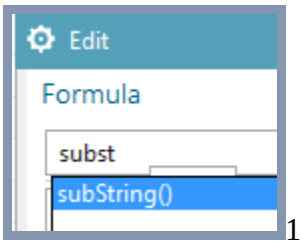
11b_02

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



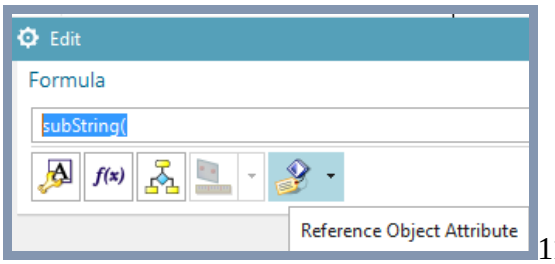
11b_03

6. Enter “subst” and accept the suggestion.



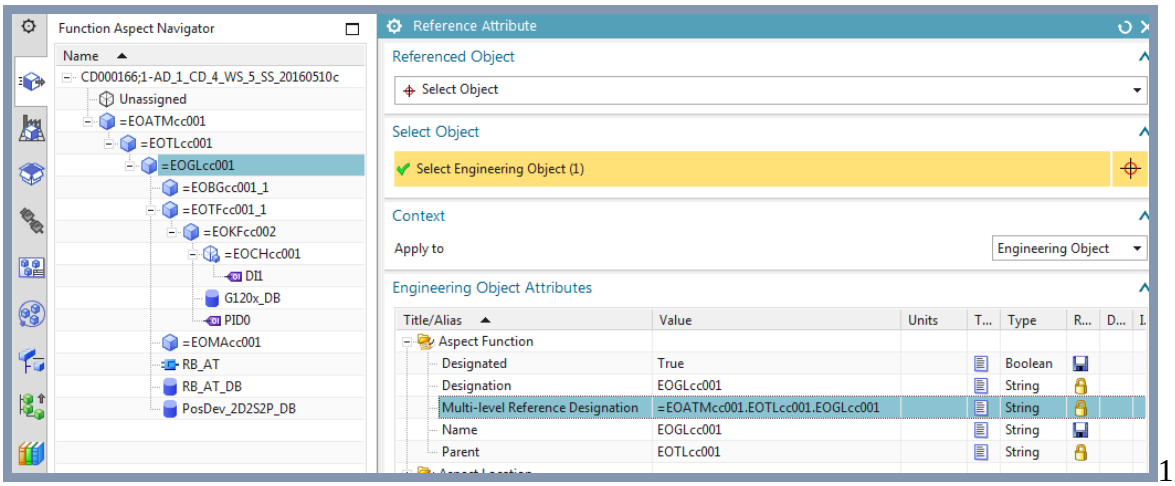
11b_04

7. Click **Reference Object Attribute**.



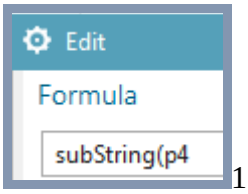
11b_05

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



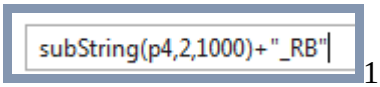
11b_06

P4 = GL Function Multi Reference Designation.



11b_07

9. Complete the expression.

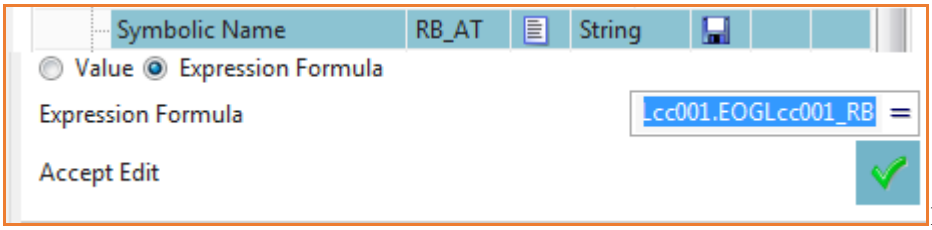


11b_08

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

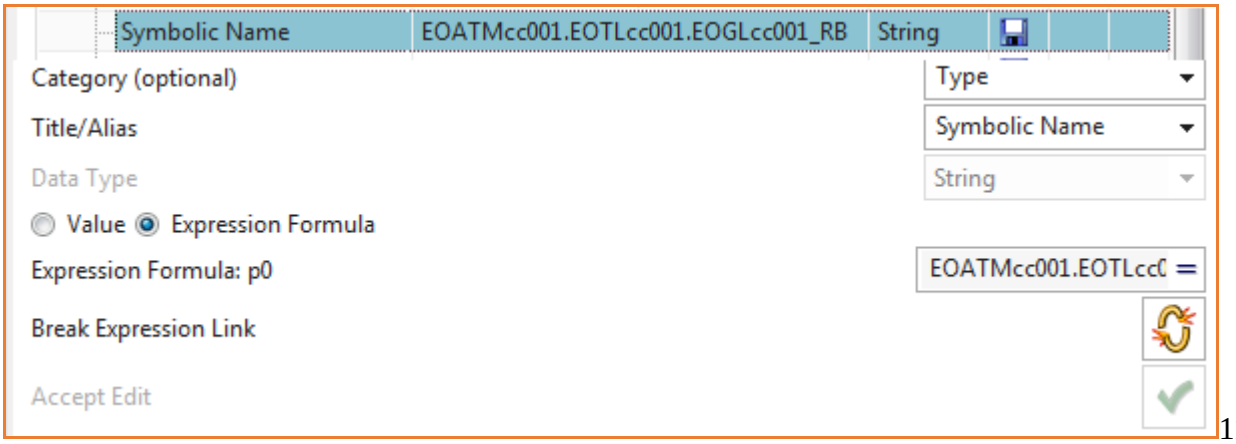
11b_09

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



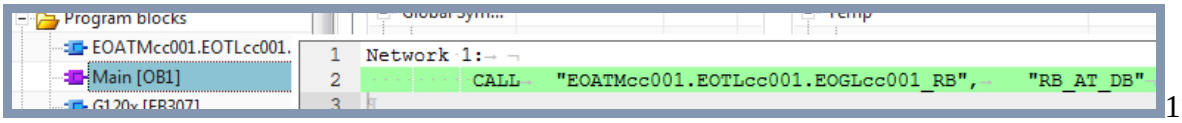
11b_10

12. Click the green arrow.



11b_11

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



11b_12

14.1.2. RB_AT IDB symbolic name

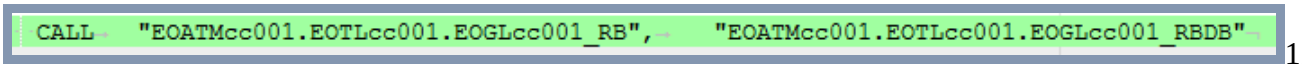
The RB_AT IDB must have a symbolic name that is unique when a template is instantiated. You do this by using an expression to assign a name based on the Function aspect Multi Reference Designation of RB_AT IDB.

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.

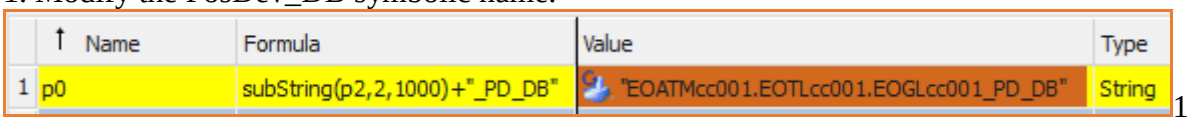


```
CALL   "EOATMcc001.EOTLcc001.EOGLcc001_RB",   "EOATMcc001.EOTLcc001.EOGLcc001_RBDB"
```

11b_14

14.1.3. PosDev_DB symbolic name

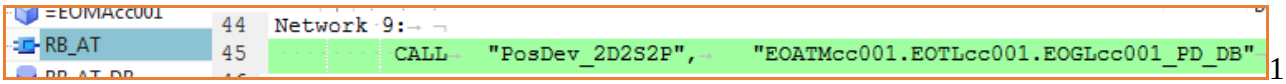
1. Modify the PosDev_DB symbolic name.



	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.



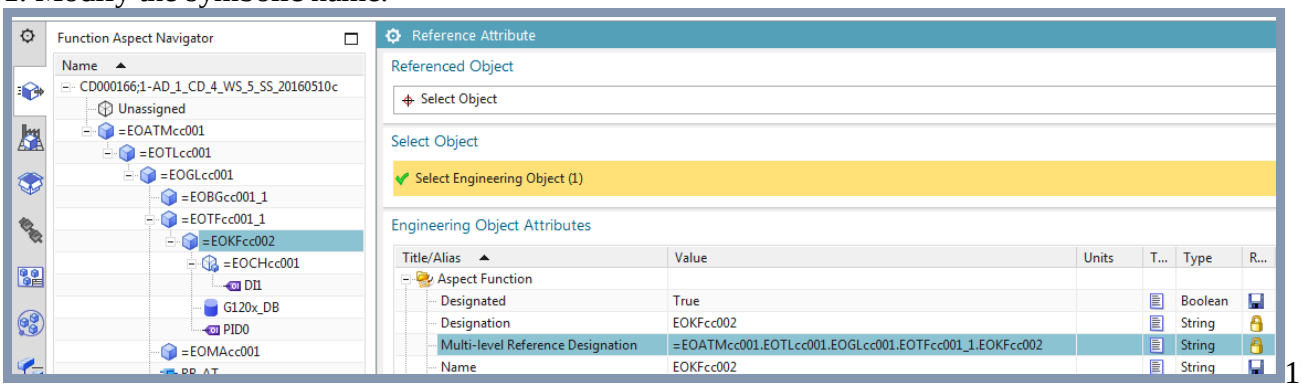
```
RB_AT      44  Network 9:
           45      CALL   "PosDev_2D2S2P",   "EOATMcc001.EOTLcc001.EOGLcc001_PD_DB"
```

11b_16

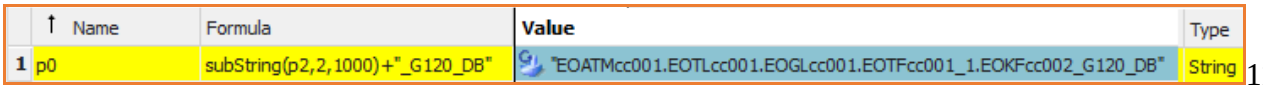
14.1.4. G120 IDB symbolic name

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.



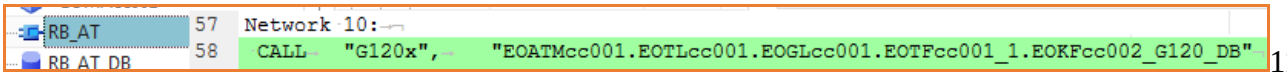
11b_17



	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.



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

11b_19

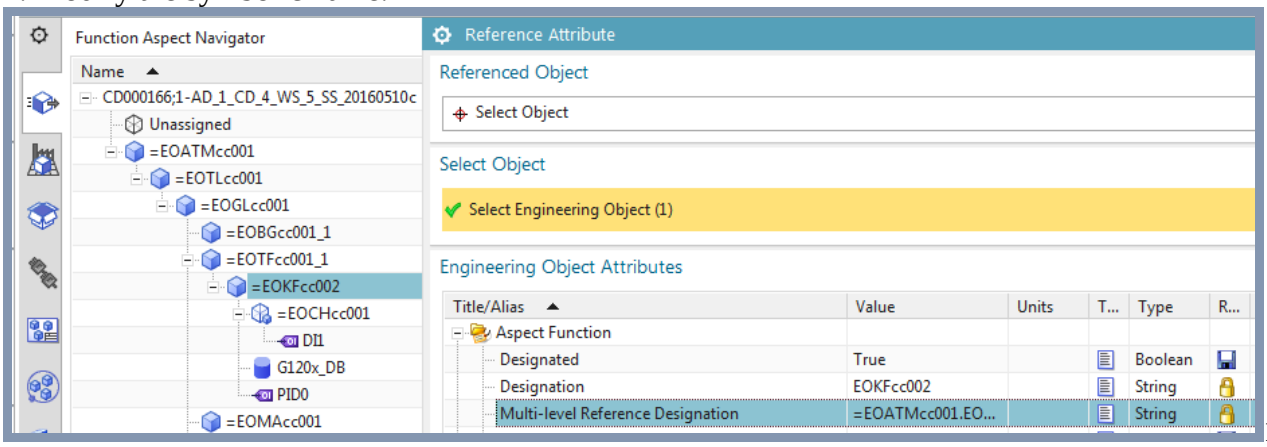
>>>>>>>>>>>>>> TERRY… did I forgot to do this previously???

```
Network 9:
      CALL   "PosDev_2D2S2P",   "EOATMcc001.EOTLcc001.EOGLcc001_P
       LS_ADV :=   "Pos_front_left"
       SW_FS_ADV :=   "slow_forw"
```

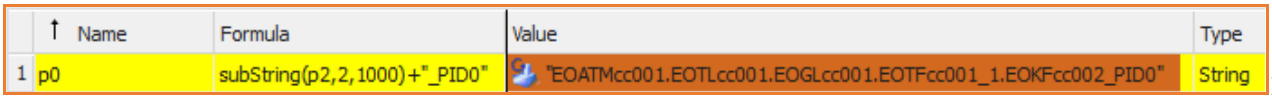
14.1.5. PID0 symbolic name

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.

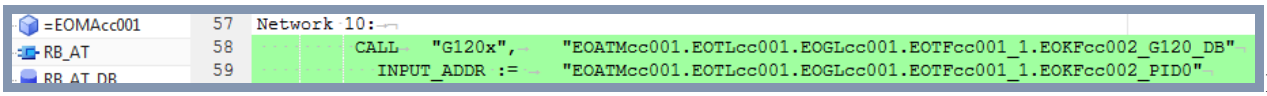


11b_20



11b_21

The following shows the result.

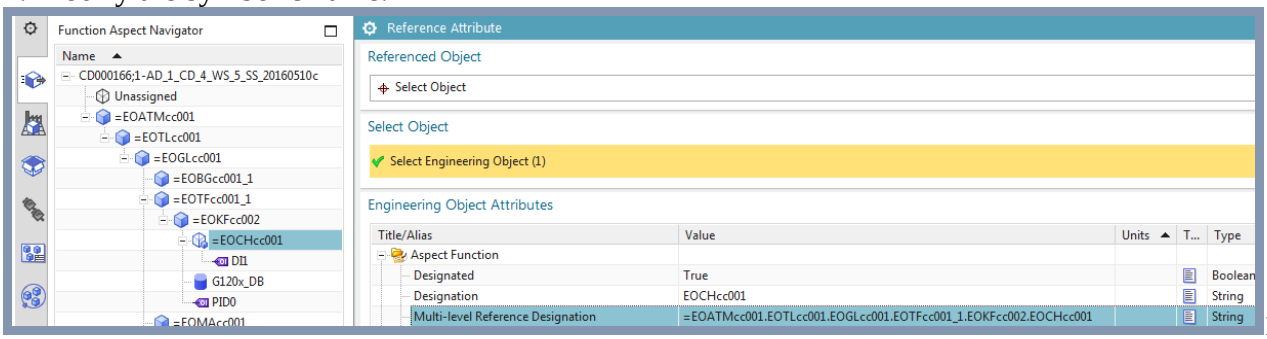


11b_22

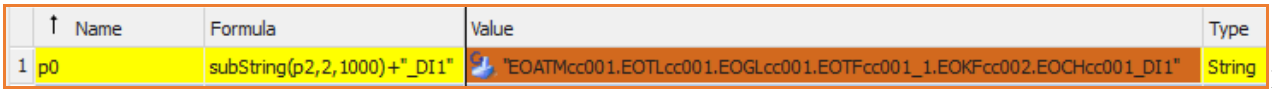
14.1.6. DI1 symbolic name

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.

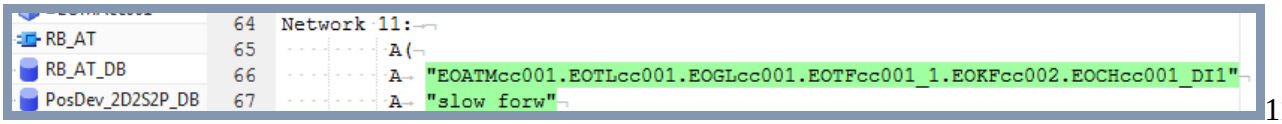


11b_23



11b_24

The following shows the result.

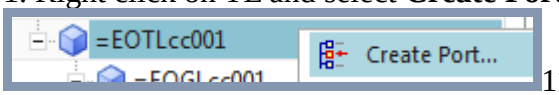


11b_25

14.2. FRGEStop dynamic connection

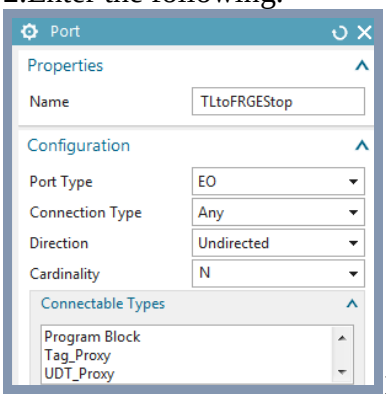
First create a port from TL to the tag.

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



11b_26

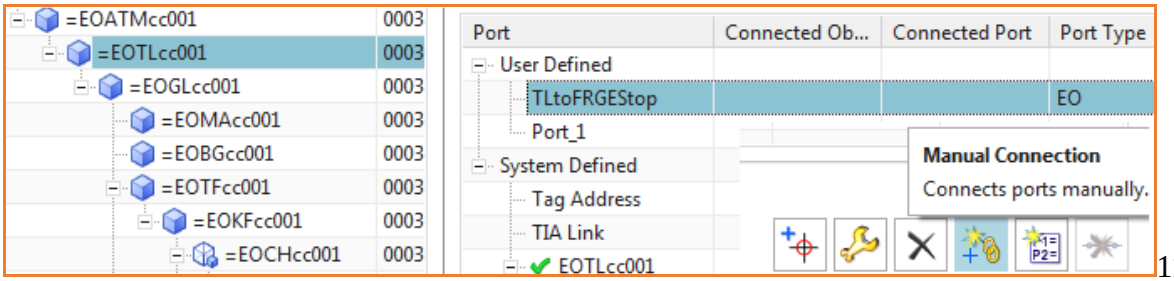
2.Enter the following.



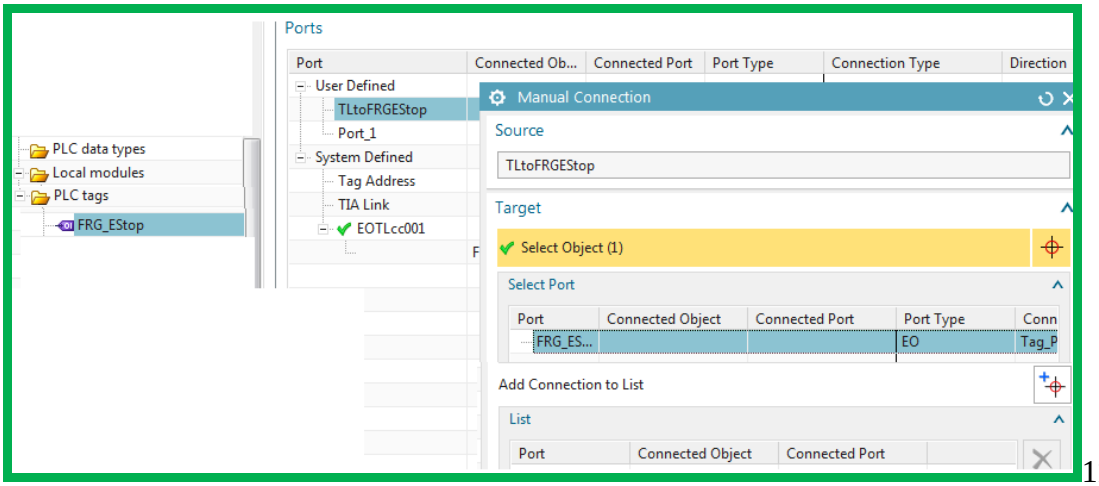
11b_27

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.

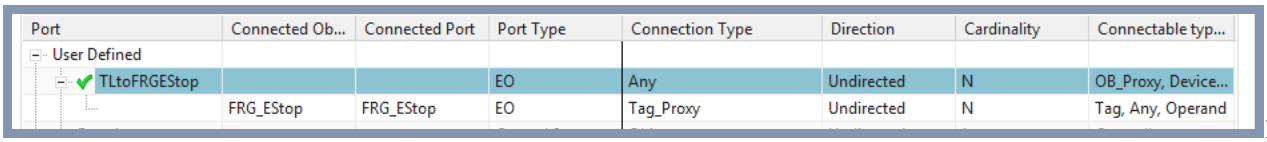


11b_28



11b_29

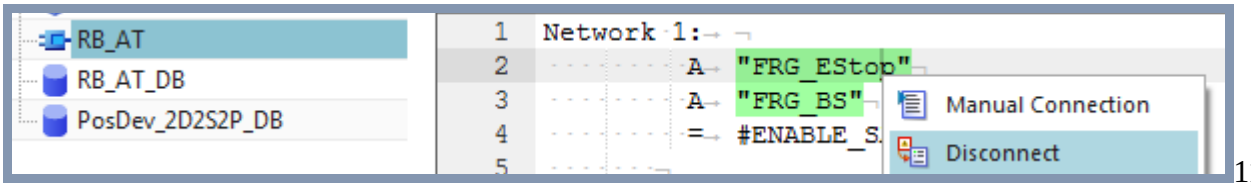
The following shows the result.



11b_30

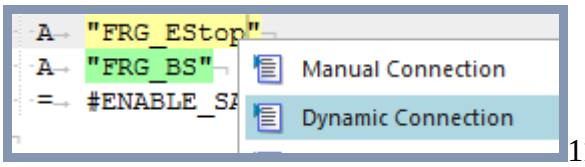
Now create the dynamic connection from RBAT FB to the tag via the TL port.

5. Disconnect the manual connection.



11b_31

6. Create a dynamic connection.



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

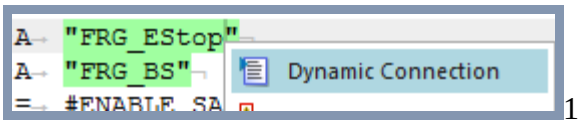
8. Add an expression to get the tag.

	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.



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

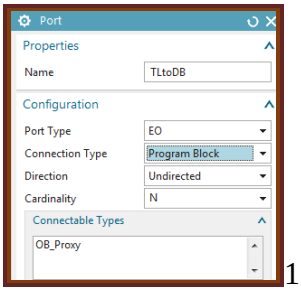
10. Show all.


11b_38

14.3. OB Main calls

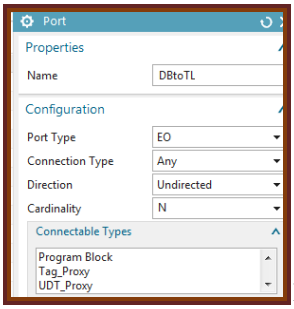
Now you need to enable OB Main to automatically add a call to an added GL. The following is my guess at how to do this, Not sure if it's the official way.

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**).



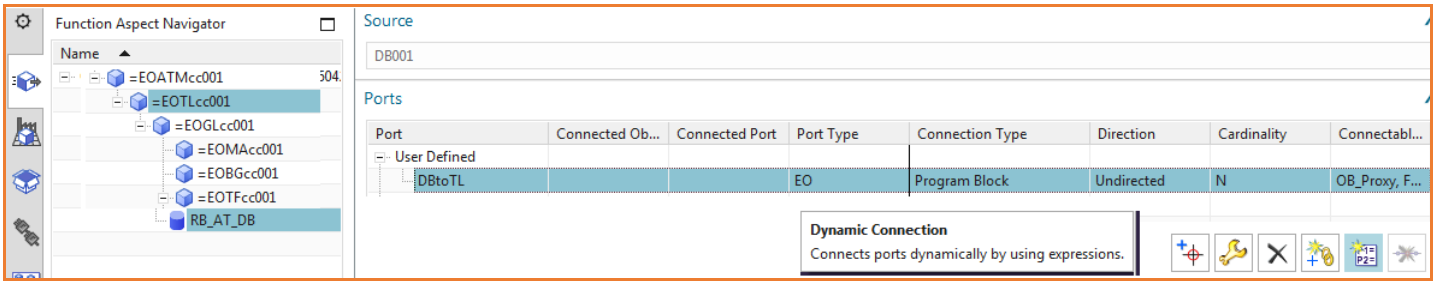
11b_39

2. Create a port on RBAT IDB.

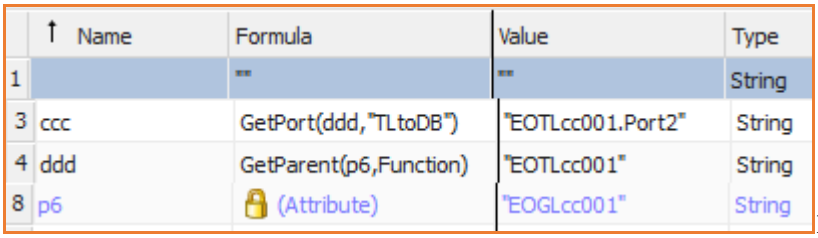


11b_40

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



11b_41 xxxxxx

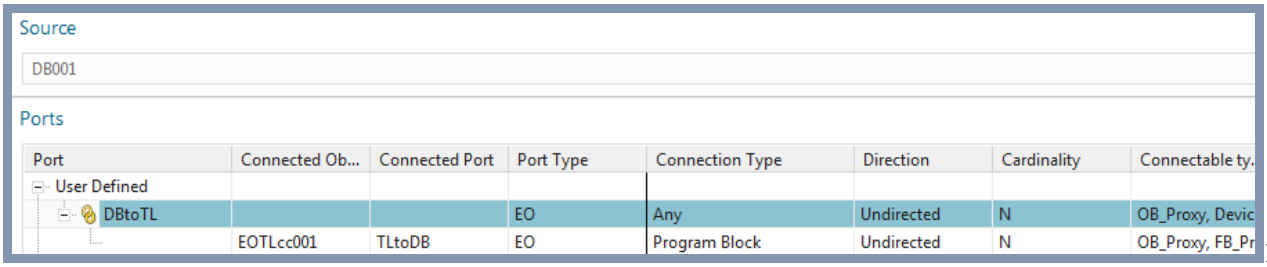


	Name	Formula	Value	Type
1		""	""	String
3	ccc	GetPort(ddd,"TLtoDB")	"EOTLcc001.Port2"	String
4	ddd	GetParent(p6,Function)	"EOTLcc001"	String
8	p6	(Attribute)	"EOGLcc001"	String

11b_42


11b_43

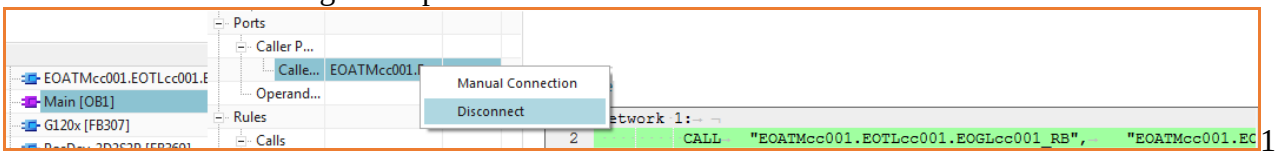
The following shows the result.



11b_44

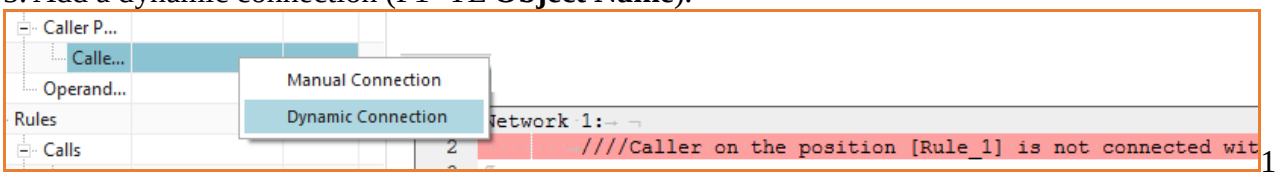
Now change the OB main to RBAT DB call port.

4. Disconnect the existing caller port.

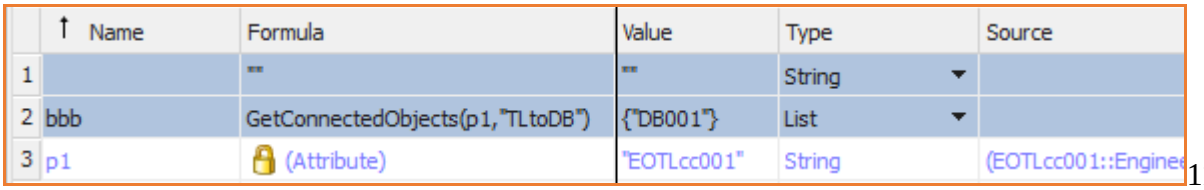


11b_45

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

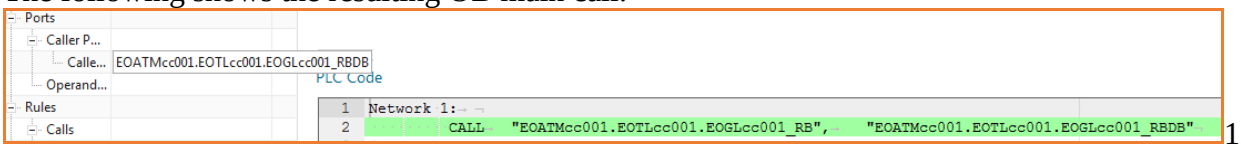


11b_46



11b_47

The following shows the resulting OB main call.



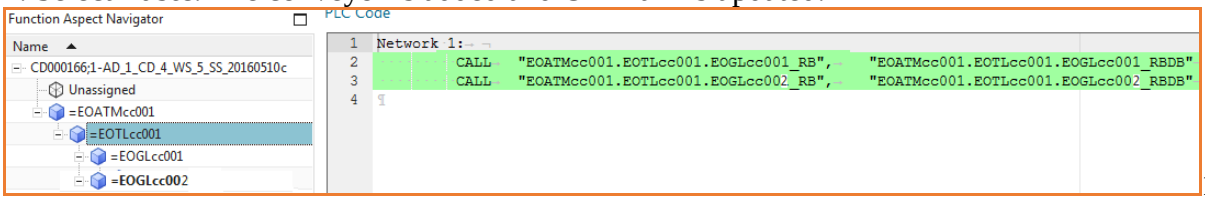
11b_48

14.4. 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.

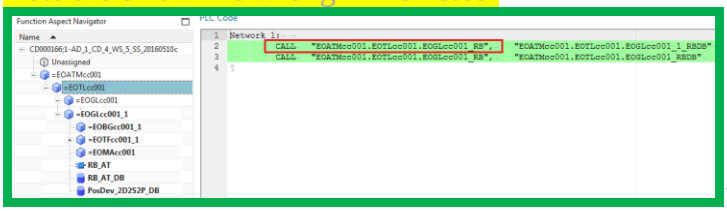


11b_49 xxxxx

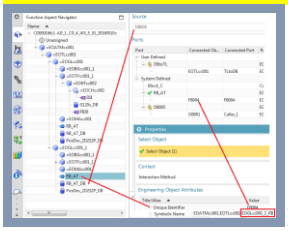
14.2.1. OB main bug

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

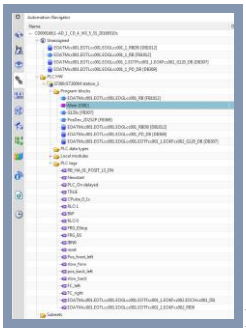
Note the error. The wrong FB is listed.


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.

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

11b_54

14.2.2. 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"
)
```

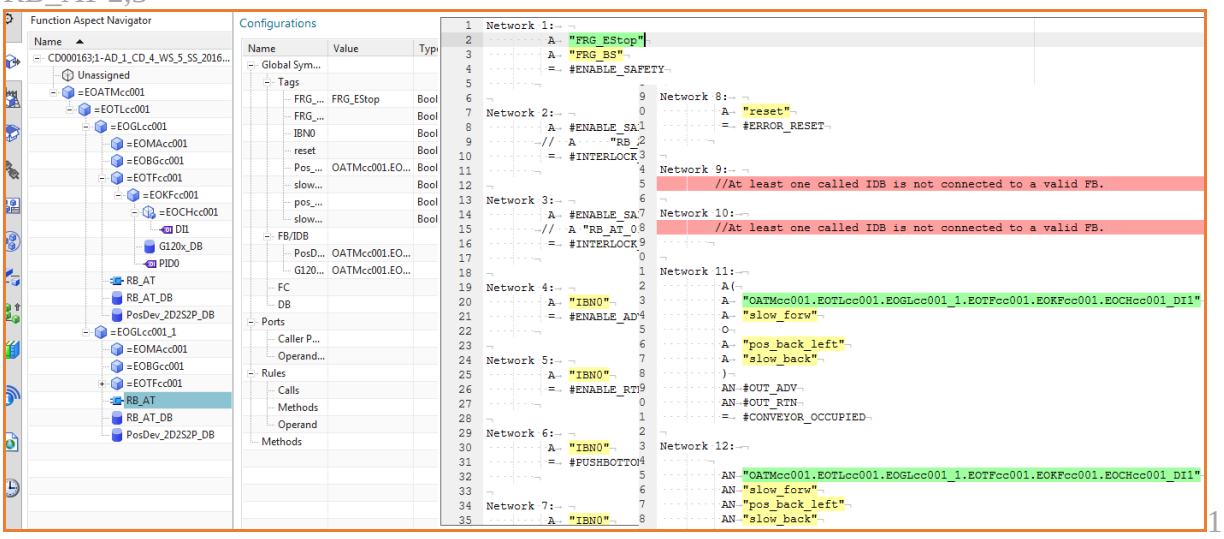
11b_55

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.EOTFcc002.EOCHcc001_DI1"
A "slow_forw"
O
A "pos_back_left"
A "slow_back"
)
```

11b_56

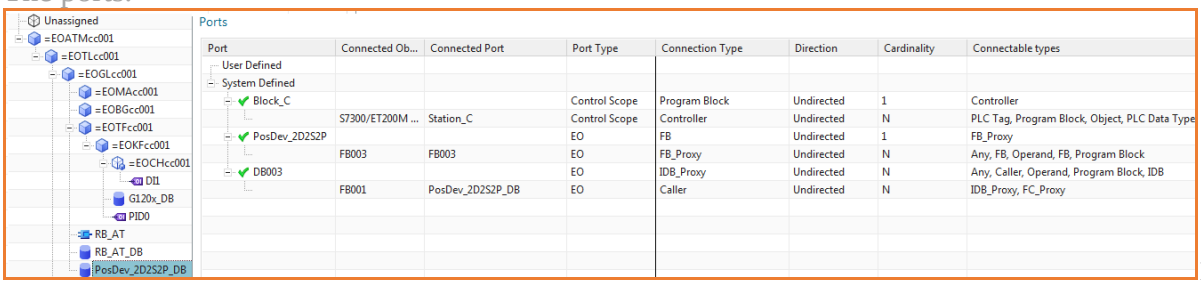
RB_AT 2,3



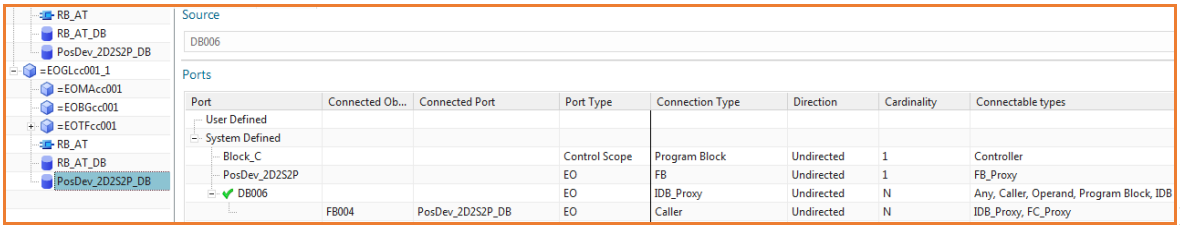
11b_57

Following shows how I previously manually fixed this.

The ports.

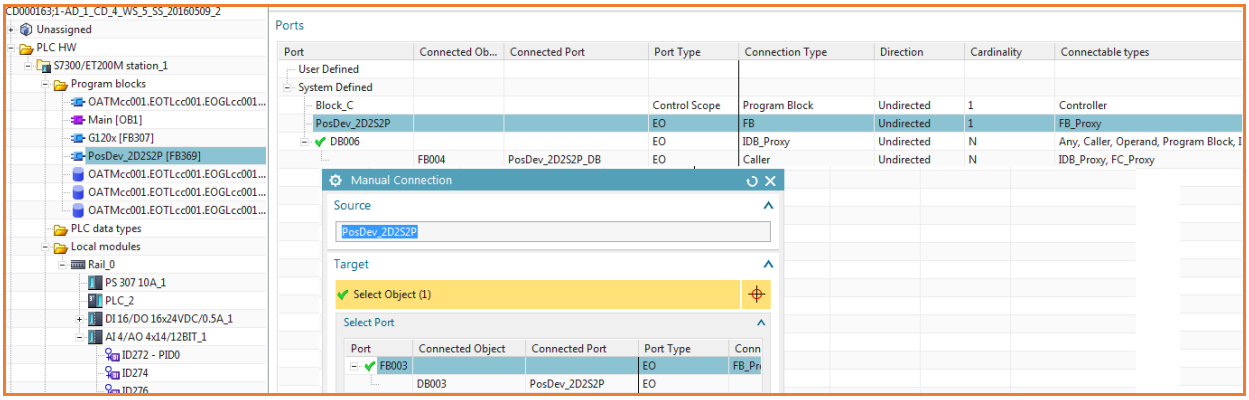


11b_58



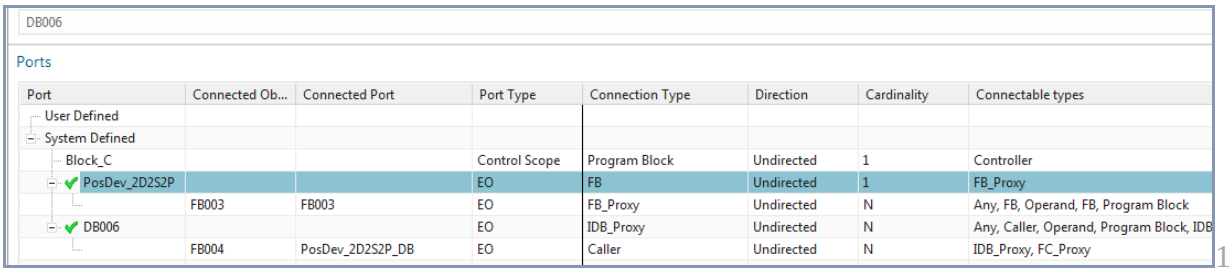
11b_59

1. manually connect the ports.

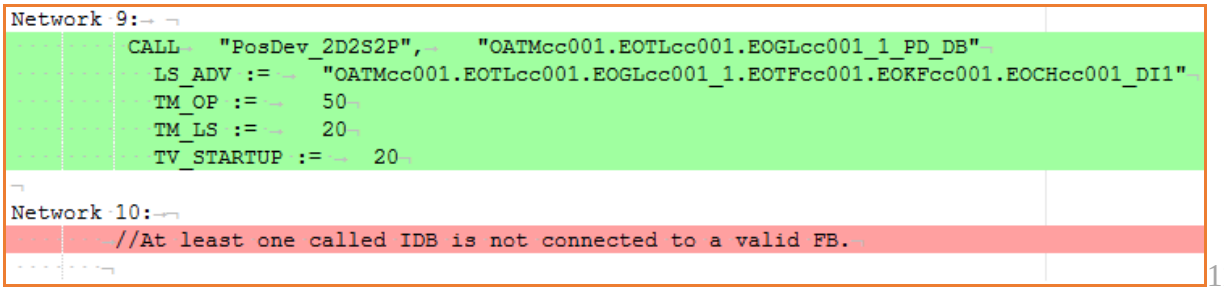


11b_60

Result.



11b_61



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.

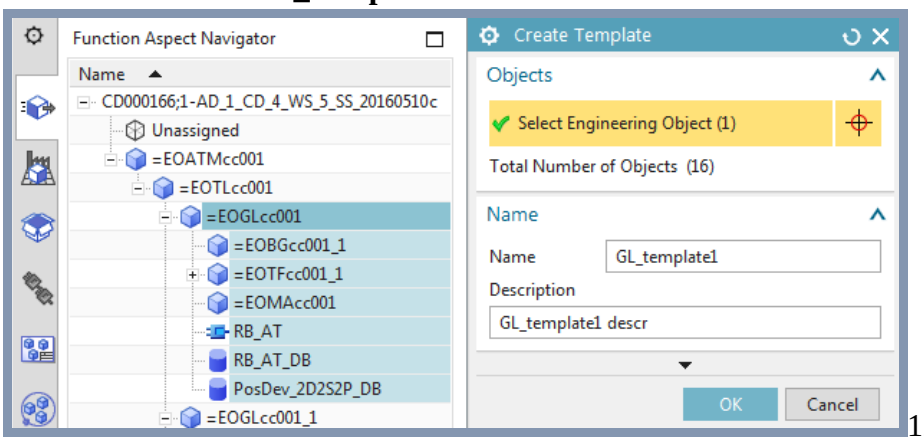
15. Create/instantiate template

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.

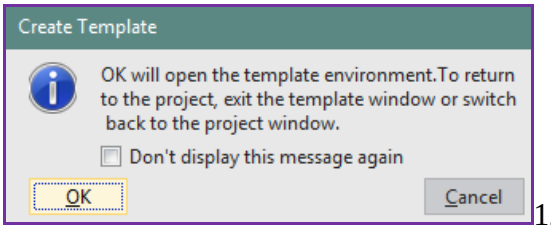
15.1. Create template

1. Select GL01. Click **System Design→Create Template**.
2. For **Name** enter **GL_Template**.



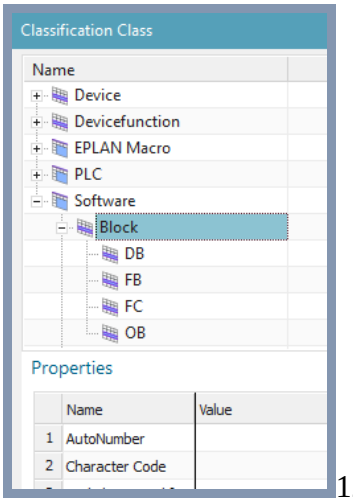
12b_01

3. Click **OK**.



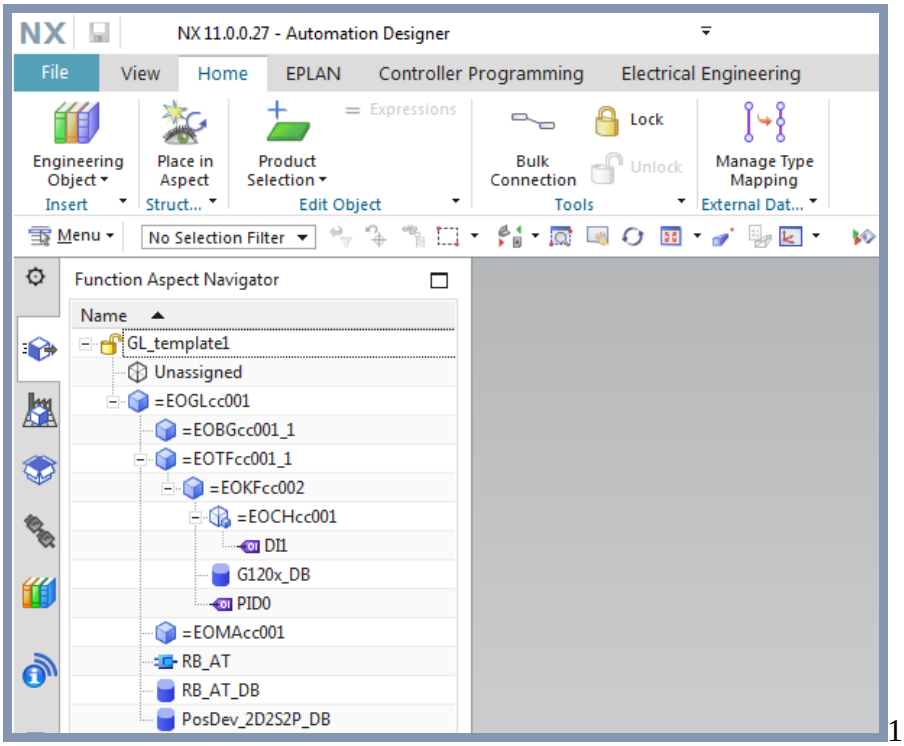
12b_02

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



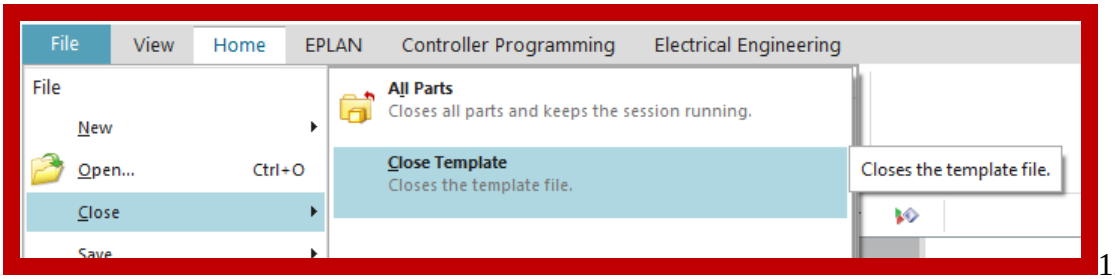
12b_03

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



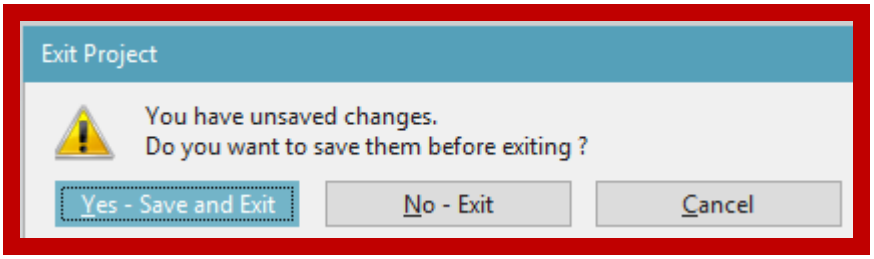
12b_04

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

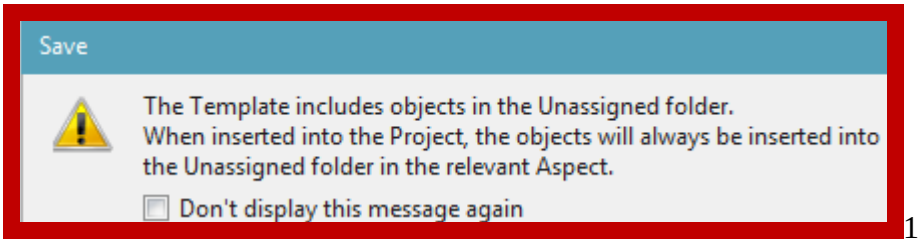


12c_01

8. Choose **Yes – Save and Exit**.



12c_02



12c_03

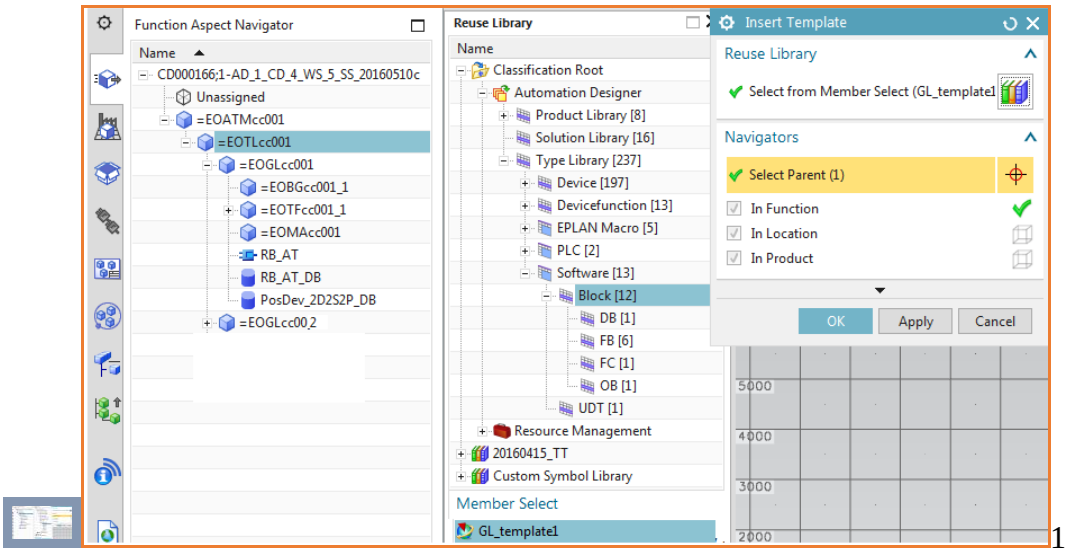
15.2. Add template to mapping xxxx

15.3. Insert template

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

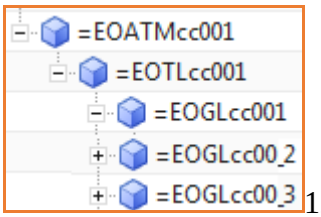
15.3.1. Drag&drop template

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



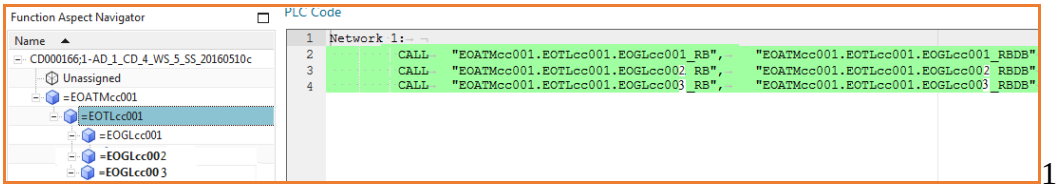
12b_06xxxxx

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.



12b_17b

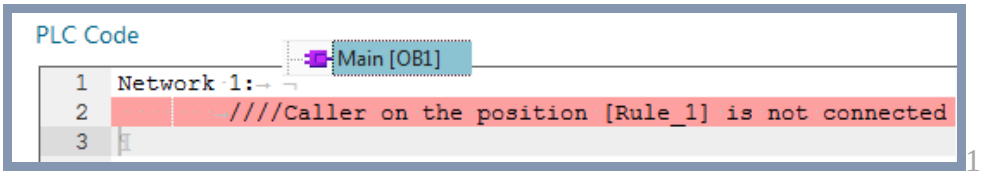

12b_08

RBAT FB:


12b_09

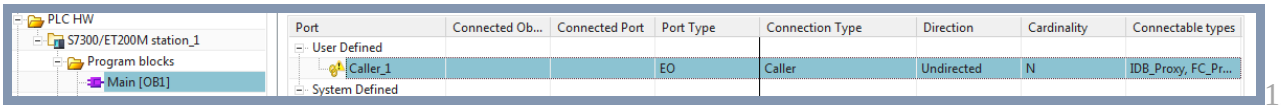
xxx15.3.2. Fix errors

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

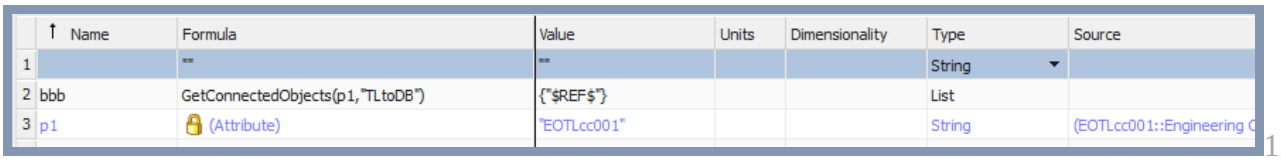


12b_10

2. Open the port



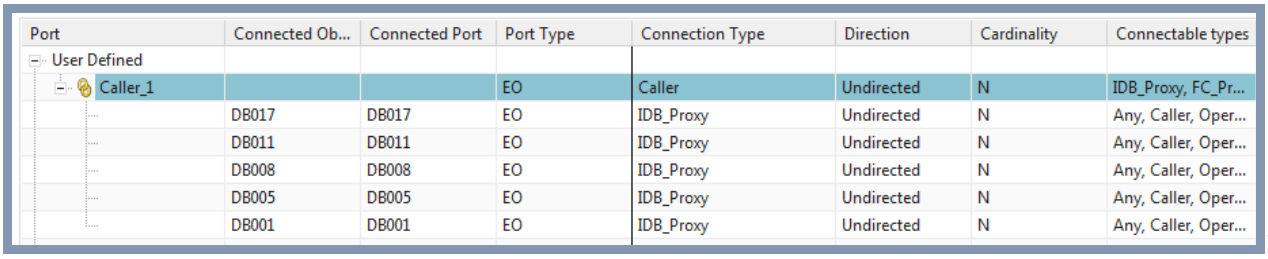
12b_11



12b_12

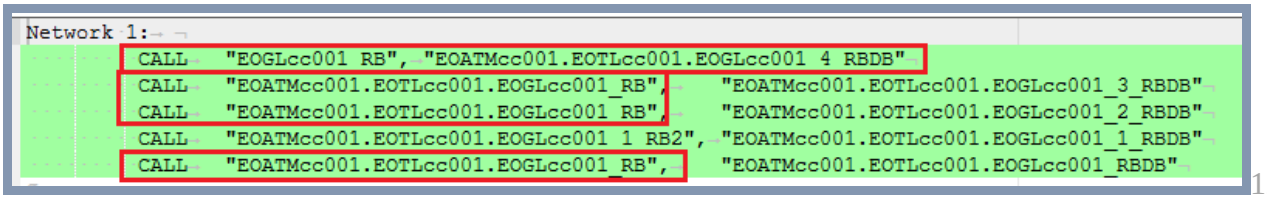
3. Select bbb.

4. Click ok. The 5 conveyors appear.



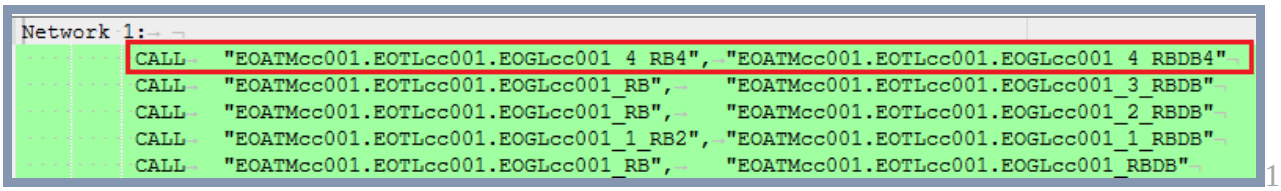
12b_13

Note the errors in main.



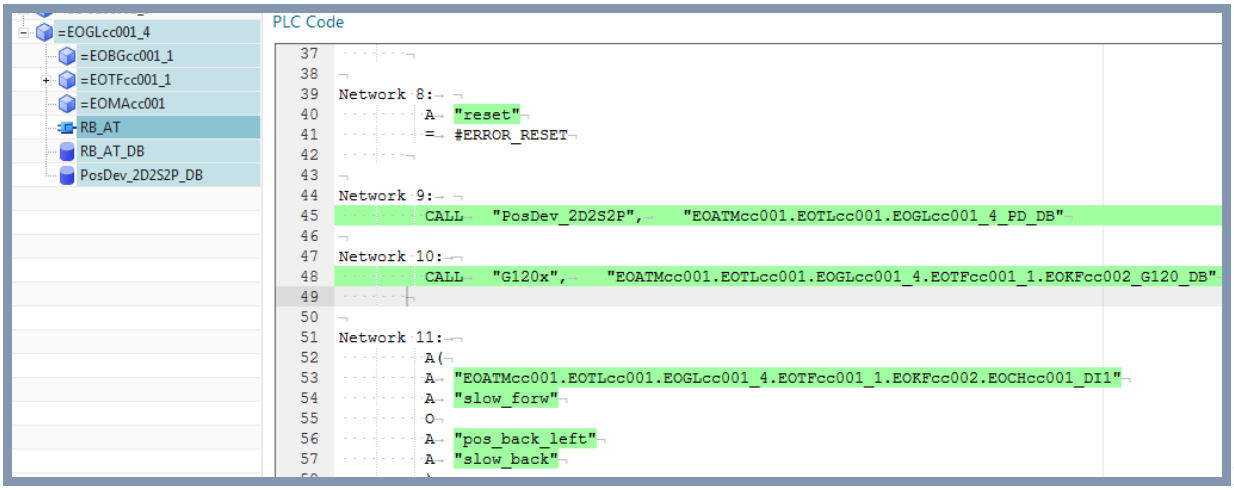
12b_14

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



12b_15

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



12b_16

16. Synchronize changes (with templates)

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**.