

Chapter_1: DaVinci Tools (Classic AUTOSAR Development)

Ford FNV4 FordOS

Exported on 04/27/2023

Table of Contents

1	Purpose.....	10
2	Recipes	11
3	1_ Add new Bsw module to a Project.....	12
3.1	Workflow Diagram	12
3.2	Prerequisites	12
3.3	Steps	13
3.4	Steps to be performed in Davinci Configurator Tool	13
3.4.1	Step 1: Open Project Setting Tab	13
3.4.2	Step 2: Add the BSW module.....	14
3.4.3	Step 3: Resolve the upcoming errors	17
3.4.4	Step 4: Generate the Project.....	17
3.5	Troubleshooting.....	20
4	2_ Create new Tangible Software Component in a Project.....	21
4.1	Workflow Diagram	21
4.2	Prerequisites	22
4.3	Templates Files	22
4.4	Steps	23
4.5	Steps to be performed in a Text Editor	23
4.5.1	Step 1: Name change verification	23
4.6	Steps to be performed in Davinci Developer Tool.....	23
4.6.1	Step 2: Include the New Files in the Project	23
4.7	Steps to be performed in Davinci Configurator Tool	26
4.7.1	Step 3: Synchronize Project.....	26
4.7.2	Step 4: Generate the Project.....	27
4.8	Steps to be performed in Davinci Developer Tool.....	29
4.8.1	Step 5: Verification	29
4.9	Troubleshooting.....	31
4.9.1	Not able to edit the newly added SW Component in the Davinci Developer.....	31
4.10	Workflow Diagram	32
4.11	Prerequisites	32

4.12	Templates Files	33
4.13	Steps	33
4.14	Steps to be performed in a Text Editor	33
4.14.1	Step 1: Name change verification	33
4.15	Steps to be performed in Davinci Developer Tool.....	34
4.15.1	Step 2: Include the New Files in the Project	34
4.16	Steps to be performed in Davinci Configurator Tool	37
4.16.1	Step 3: Synchronize Project.....	37
4.16.2	Step 4: Generate the Project.....	38
4.17	Steps to be performed in Davinci Developer Tool.....	40
4.17.1	Step 5: Verification	40
4.18	Troubleshooting.....	42
4.18.1	Not able to edit the newly added SW Component in the Davinci Developer.....	42
5	3_ Create a New Compu Method as Enum.....	43
5.1	Context	43
5.2	Workflow Diagram	43
5.3	Prerequisites	44
5.4	Steps	44
5.5	Steps to be performed in Davinci Developer Tool.....	44
5.5.1	Step 1: Go to package view of the project	44
5.5.2	Step 2: Create the Compu Method	46
5.5.3	Step 3: Fill the "New Compu Method" Window	49
5.6	Steps to be performed in Davinci Configurator Tool	50
5.6.1	Step 4: Synchronize Project.....	50
5.6.2	Step 5: Generate the Project.....	51
5.7	Troubleshooting.....	53
6	4_ Create New Implementation DataType	54
6.1	Workflow Diagram	54
6.2	Prerequisites	55
6.3	Steps	55
6.4	Steps to be performed in Davinci Developer Tool.....	55
6.4.1	Step 1: Go to package view of the project	55

6.4.2	Step 2: Create the Compu Method	57
6.4.3	Step 3: Fill the "New Implementation Record" Window	59
6.5	Steps to be performed in Davinci Configurator Tool	61
6.5.1	Step 4: Synchronize Project.....	61
6.5.2	Step 5: Generate the Project.....	61
6.6	Troubleshooting.....	63
7	5_ Add New SWC to the Deployment List	64
7.1	Use Case	64
7.2	Workflow Diagram	64
7.3	Prerequisites	65
7.4	Steps	65
7.5	Steps to be Performed in ARXML Editor.....	65
7.5.1	Step 1: SWC Verification.....	65
7.5.2	Step 2: Add the SW Mapping.....	65
7.5.3	Step 3: Add the SW Implemantion Mapping	67
7.6	Steps to be performed in Davinci Configurator Tool	69
7.6.1	Step 4: Synchronize Project.....	69
7.6.2	Step 5: Update ECU Configuration ARXML.....	69
7.6.3	Step 6: Generate the Project.....	70
7.7	Troubleshooting.....	72
8	6_ Create a New Composition Component	73
8.1	Workflow Diagram	73
8.2	Prerequisites	73
8.3	Steps	74
8.4	Steps to be performed in Davinci Developer Tool.....	74
8.4.1	Step 1: Create Composition Component Type.....	74
8.4.2	Step 2: Fill the "New Component Type"	75
8.4.3	Step 3: Edit Composition Component	76
8.4.4	Step 4: Create Input/Outut Ports	79
8.4.5	Step 5: Create Composition instance in the ECU Composition.....	79
8.5	Steps to be performed in Davinci Configurator Tool	82
8.5.1	Step 6: Synchronize Project.....	82
8.5.2	Step 7: Update ARXML Files in Davinci Cfg.....	84

8.5.3	Step 8: Generate the Project.....	84
8.6	Troubleshooting.....	86
9	7_ Create Connection between a SWC and a Service Component.....	87
9.1	Workflow Diagram	87
9.2	Prerequisites	88
9.3	Steps.....	89
9.4	Steps to be performed in Davinci Developer	89
9.4.1	Step 1 Create New Service Port.....	89
9.4.2	Step 2 Add the neccesary access to the runnables	93
9.5	Steps to be performed in Davinci Configurator.....	96
9.5.1	Step 3: Synchronize Project.....	96
9.5.2	Step 4: Map the newly created Port with the Service.....	96
9.5.3	Step 5 Generate the Project.....	98
9.6	Steps to be performed in and external IDE tool	100
9.6.1	Step 6 Verify the RTE Connection.....	100
9.7	Troubleshooting.....	100
10	8_ Create a New Rx CAN Signal.....	101
10.1	Workflow Diagram	101
10.2	Prerequisites	102
10.3	Steps on Davinci Configurator.....	102
10.3.1	Step 1: Create a new Pdu	102
10.3.2	Step 2: Create a new CanHardwareObject	105
10.3.3	Step 3: Create a new CanIfHrhCfgs.....	108
10.3.4	Step 4: Create a New CanIfTxPduCfgs.....	110
10.3.5	Step 5: Create a New ComSignals	113
10.3.6	Step 6: Create a new ComIPdus	116
10.3.7	Step 7: Create a New PduRRoutingPath	119
10.4	Steps to be performed directly on the ARXML File	122
10.4.1	Step 8: Create the Signals in the ARXML and the Port Mapping	122
10.4.1.1	Changes on the ECU System Extract or Communication System Extract.....	122
10.4.1.2	Changes in the SWC that will use the CAN signals.....	125
10.5	Steps on Davinci Configurator.....	126
10.5.1	Step 9: Update ARXML Files in Davinci Cfg.....	126

10.6	Steps on Davinci Developer.....	127
10.6.1	Step 10: Map the new Signal.....	127
10.7	Steps on Davinci Configurator.....	128
10.7.1	Step 11: Final Conections and Verification	128
10.8	Troubleshooting.....	128
10.8.1	UL (Upper Layer) Indication related issues:.....	128
11	9_ Create a New Tx CAN Signal.....	132
11.1	Workflow Diagram	132
11.2	Prerequisites	133
11.3	Steps on Davinci Configurator.....	133
11.3.1	Step 1: Create two new Pdu.....	133
11.3.2	Step 2: Create a new CanHardwareObject	135
11.3.3	Step 3: Create a new CanIfHthCfgs.....	138
11.3.4	Step 4: Create a New CanIfBufferCfgs	140
11.3.5	Step 5: Create a New CanIfTxPduCfgs.....	142
11.3.6	Step 6: Create a New ComSignals	145
11.3.7	Step 7: Create a new ComIPdus	148
11.3.8	Step 8: Create a New PduRRoutingPath	152
11.4	Steps to be performed directly on the ARXML File	155
11.4.1	Step 9: Create the Signals in the ARXML and the Port Mapping	155
11.4.1.1	Changes on the ECU System Extract or Communication System Extract.....	155
11.4.1.2	Changes in the SWC that will use the CAN signals.....	158
11.5	Steps on Davinci Configurator.....	159
11.5.1	Step 10: Update ARXML Files in Davinci Cfg.....	159
11.6	Steps on Davinci Developer.....	160
11.6.1	Step 11: Map the new Signal.....	160
11.7	Steps on Davinci Configurator.....	160
11.7.1	Step 12: Final Conections and Verification	160
11.8	Troubleshooting.....	161
11.8.1	UL (Upper Layer) Indication related issues:.....	161
12	10_ Multiple Signal in CAN Message Configuration.....	165
12.1	Prerequisites	165
12.2	Steps	165

12.3	Steps on Davinci Configurator.....	165
12.3.1	After creating a new Pdu (Step 1)	165
12.3.2	- Continue guide execution -	167
12.3.3	After creating a new ComSignals (Step 5 or Step 6)	167
12.3.4	- Continue guide execution -	169
12.3.5	After creating a new ComIPdus (Step 6 or 7)	169
12.3.6	- Continue guide execution -	173
12.4	Troubleshooting.....	173
13	11_ Create a New SOME/IP Signal.....	174
13.1	Workflow Diagram	174
13.2	Prerequisites	175
13.3	Steps on Davinci Configurator.....	175
13.3.1	Step 1: Create a new Pdu	175
13.3.2	Step 2: Create SoAd configuration	177
13.3.3	Step 3: Service Discovery Configuration.....	178
13.4	Steps to be performed directly on the ARXML File	180
13.4.1	Step 4: Create Signals in the ARXML.....	180
13.4.2	Step 5: Port Mapping on ARXML	180
13.5	Steps on Davinci Configurator.....	181
13.5.1	Step 6: Update ARXML Files in Davinci Cfg.....	181
13.5.2	Step 7: LdCom Configuration	182
13.5.3	Step 8: PduR Configuration	183
13.5.4	Step 9: SomelpXf Configuration(SOME/IP Transformer):	185
13.6	Steps on Davinci Developer.....	186
13.6.1	Step 10: Map the new Signal.....	186
13.7	Troubleshooting.....	186
14	12_ Add new Test Environment to CANoe	187
14.1	Workflow Diagram	187
14.2	Prerequisites	187
14.3	Steps	188
14.4	Steps to be performed in Vector CANoe tool	188
14.4.1	Step 1: Create new Test Environment.....	188
14.4.2	Step 2: Insert New Testcase File	189

14.4.3 Step 3: Run the TestCase	191
14.5 Troubleshooting.....	195

⚠ WIP

Please consider this page is still a Work in Progress for the first release. Please stay tune for any updates.

ⓘ Collaborators welcome

This is a internal initiative, created with the intention of share knowledge, so if you want a specific recipe to be created, please put it on the comments so the collaborators have them on the pipeline.

On the other hand if you want to share your recipe to improve this Cookbok, feel free to do it. Just use the Template¹.

¹ https://luxproject.luxoft.com/confluence/display/FORDFNV4FO/01_Template

1 Purpose

The intention behid this Confluence page is to provide some guide and expertise on the usage of the DaVinci Tool.

Inside of each recipe you will find some step to perform the action indicated in the recipe, as well as some screenshot of the steps in the given tool (In this case Vector's Davinci Tools).

Each recipe will also provide some pre-requisites as steps before the execution of the recipe itself, also will provide a Diagram showing the interaction between the tools and the Workflow for the recipe.

Some recipe will provide some Theoretical Background or Context to have a better understanding before executing the recipe.

Please consider that these recipes will not provide core AUTOSAR definitions or theory, it's more about performing the most common actions on the given tool.

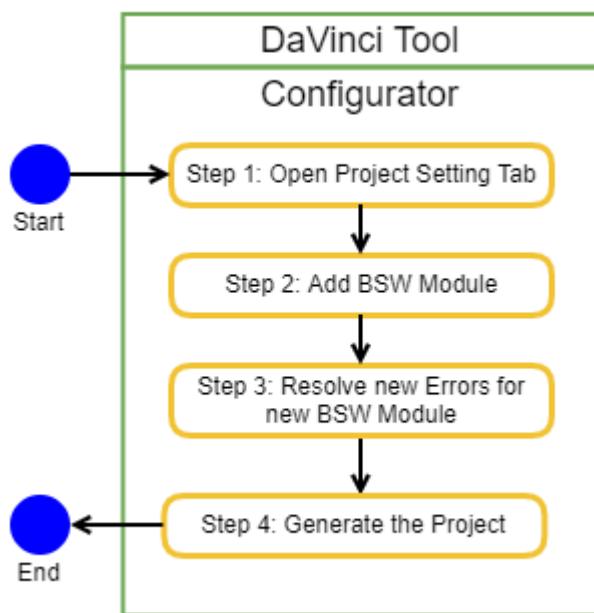
2 Recipes

- 1_ Add new Bsw module to a Project(see page 12)
- 2_ Create new Tangible Software Component in a Project(see page 21)
- 3_ Create a New Compu Method as Enum(see page 43)
- 4_ Create New Implementation DataType(see page 54)
- 5_ Add New SWC to the Deployment List(see page 64)
- 6_ Create a New Composition Component(see page 73)
- 7_ Create Connection between a SWC and a Service Component(see page 87)
- 8_ Create a New Rx CAN Signal(see page 101)
- 9_ Create a New Tx CAN Signal(see page 132)
- 10_ Multiple Signal in CAN Message Configuration(see page 165)
- 11_ Create a New SOME/IP Signal(see page 174)
- 12_ Add new Test Environment to CANoe(see page 187)

3 1_ Add new Bsw module to a Project

- Workflow Diagram(see page 12)
- Prerequisites(see page 12)
- Steps(see page 13)
- Steps to be performed in Davinci Configurator Tool(see page 13)
 - Step 1: Open Project Setting Tab(see page 13)
 - Step 2: Add the BSW module(see page 14)
 - Step 3: Resolve the upcoming errors(see page 17)
 - Step 4: Generate the Project(see page 17)
- Troubleshooting(see page 20)

3.1 Workflow Diagram



3.2 Prerequisites

1. Have a project up and running (having the input file and every previous detail).

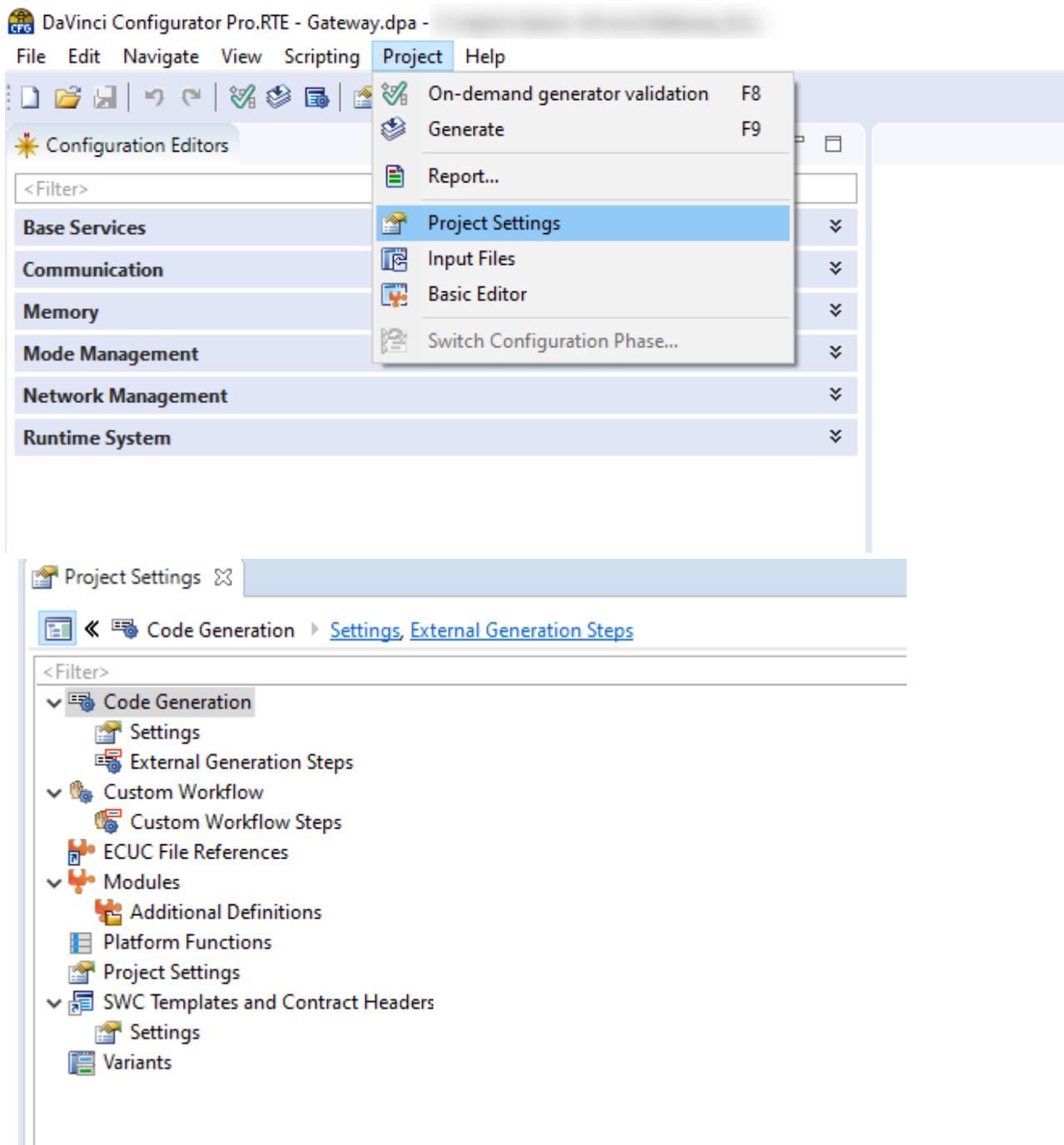
3.3 Steps

3.4 Steps to be performed in Davinci Configurator Tool

3.4.1 Step 1: Open Project Setting Tab

With the Project Open follow the next path:

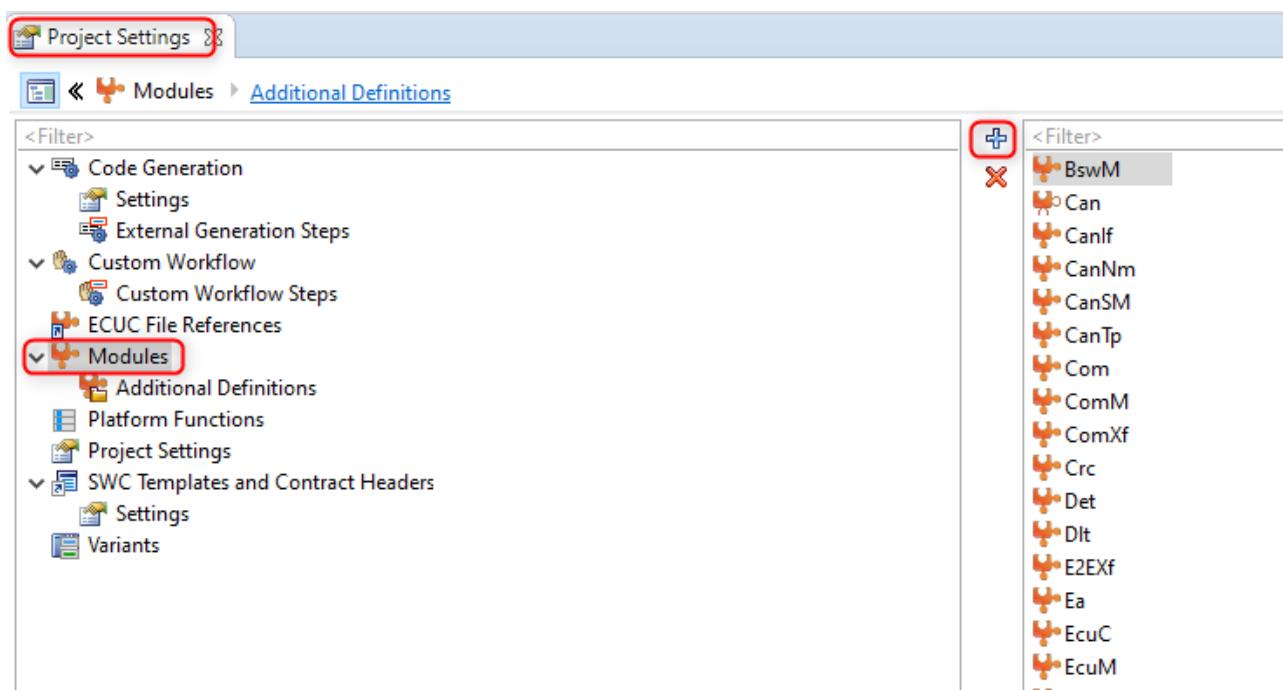
Project → Project Setting.



3.4.2 Step 2: Add the BSW module

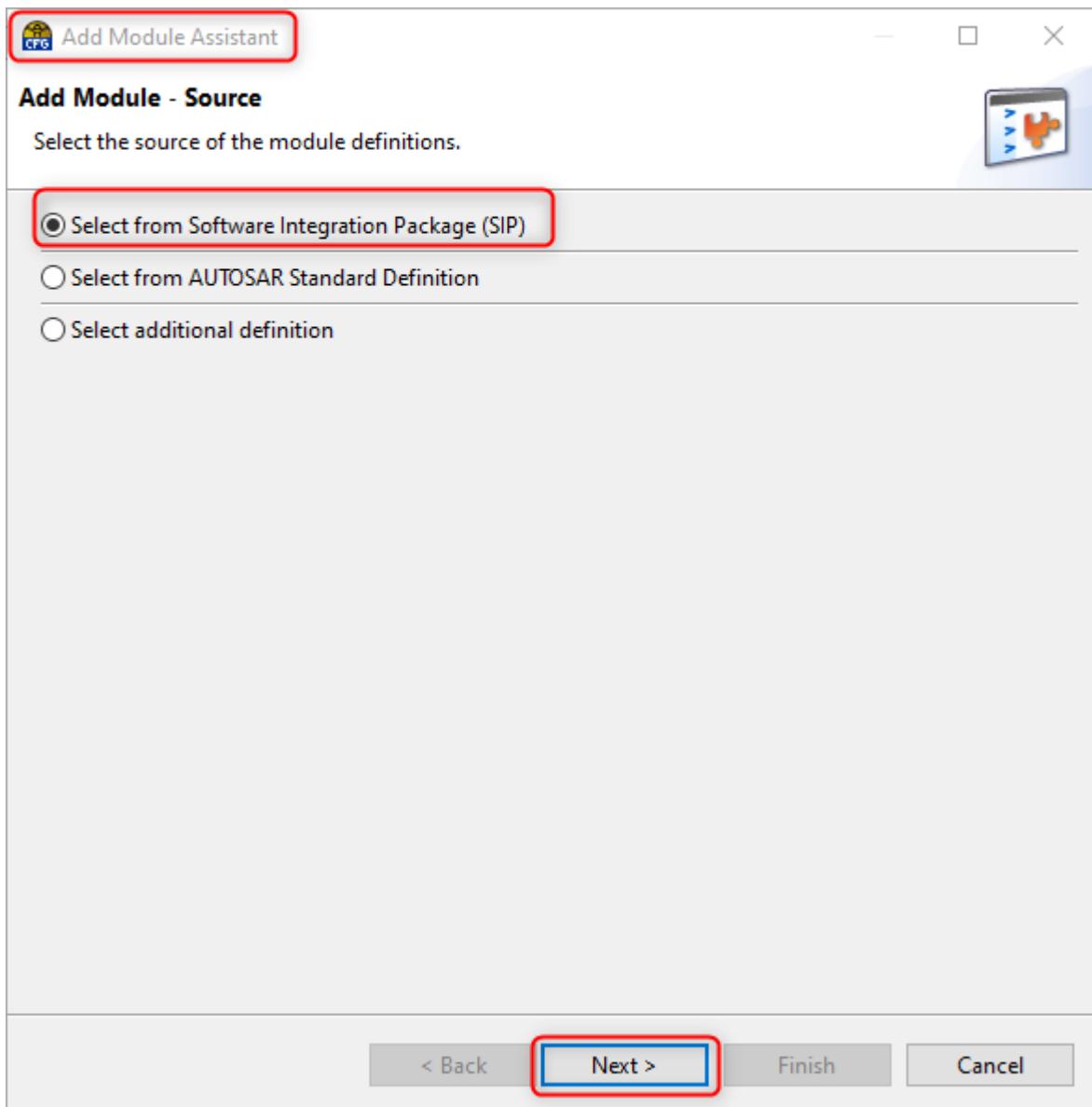
To add the module please follow the next path:

Project Setting → Modules → "+"



A new window called "Add Module Assistant" shall be open.

There select the First bullet point called "Select from Software Integration Package (SIP)" and click Next.

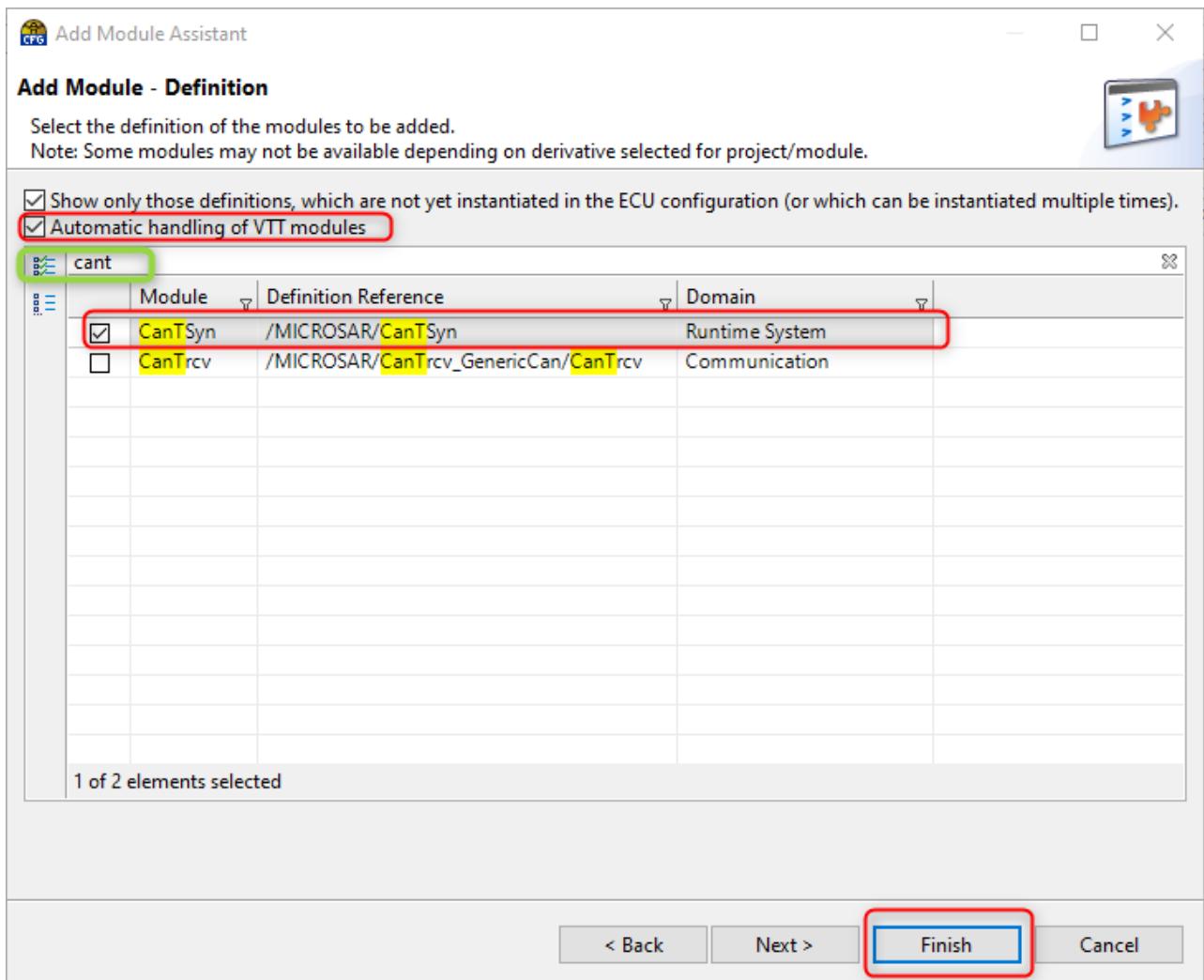


Now select the BSW to be added, clicking in the check box. Also make sure to have selected the check box for "Automatic handling of VTT modules".

You can use the filter for easy search. Green box bellow.

For this example were using CANTsync. But this steps can be performed for any BSW Module.

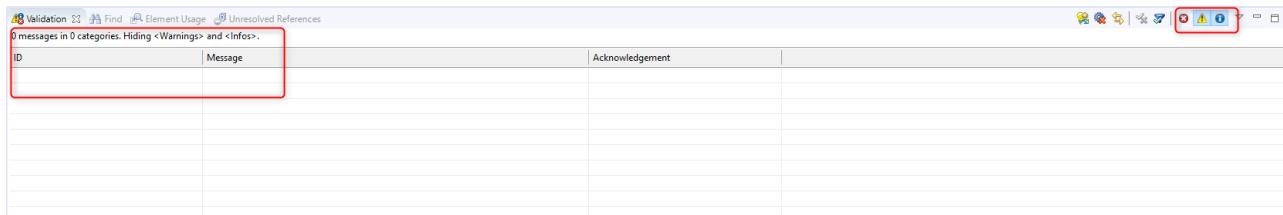
After selecting the module/s to be added, please click Finish.



3.4.3 Step 3: Resolve the upcoming errors

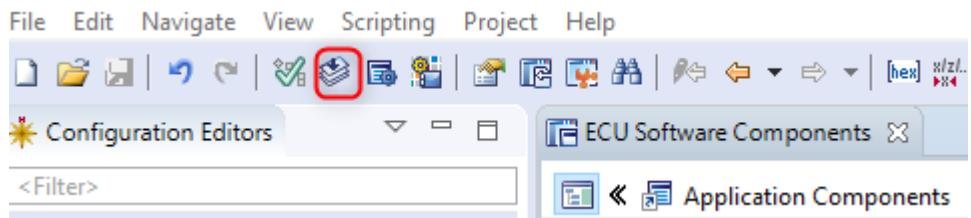
After adding the BSW Module there is the possibility of some (and new) error appear.

For this step there is the need to clean all of these errors and make before generating the code.

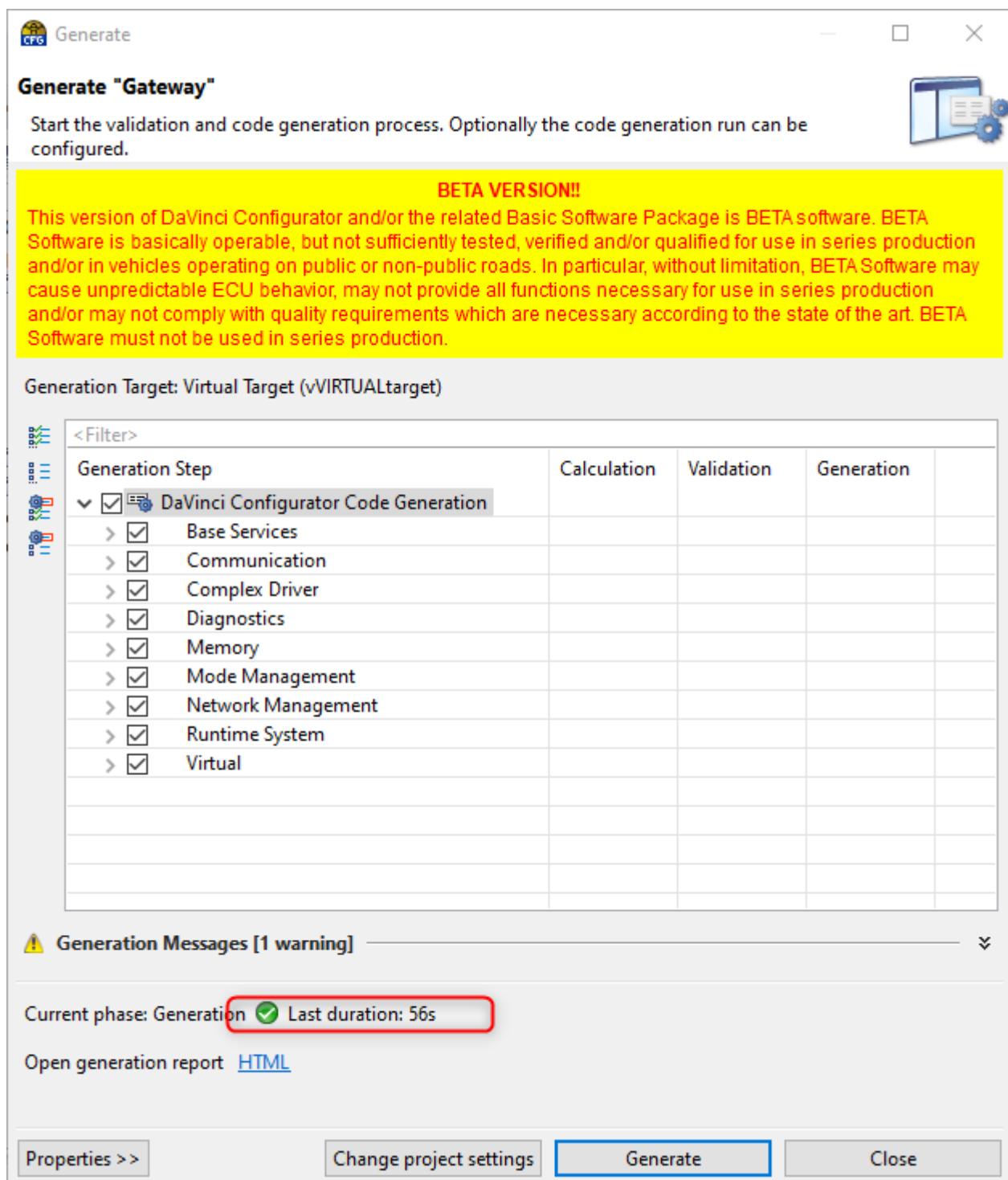


3.4.4 Step 4: Generate the Project

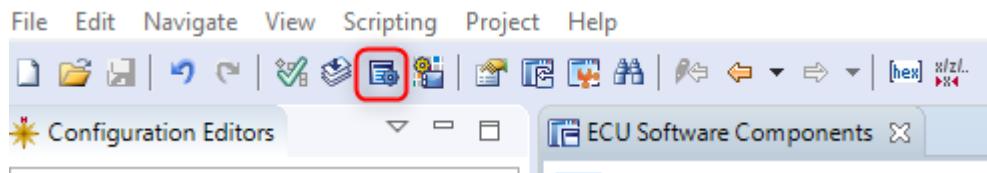
Go to Generate (Or click F9)



And Generate as usual



And finally generate the corresponding templates

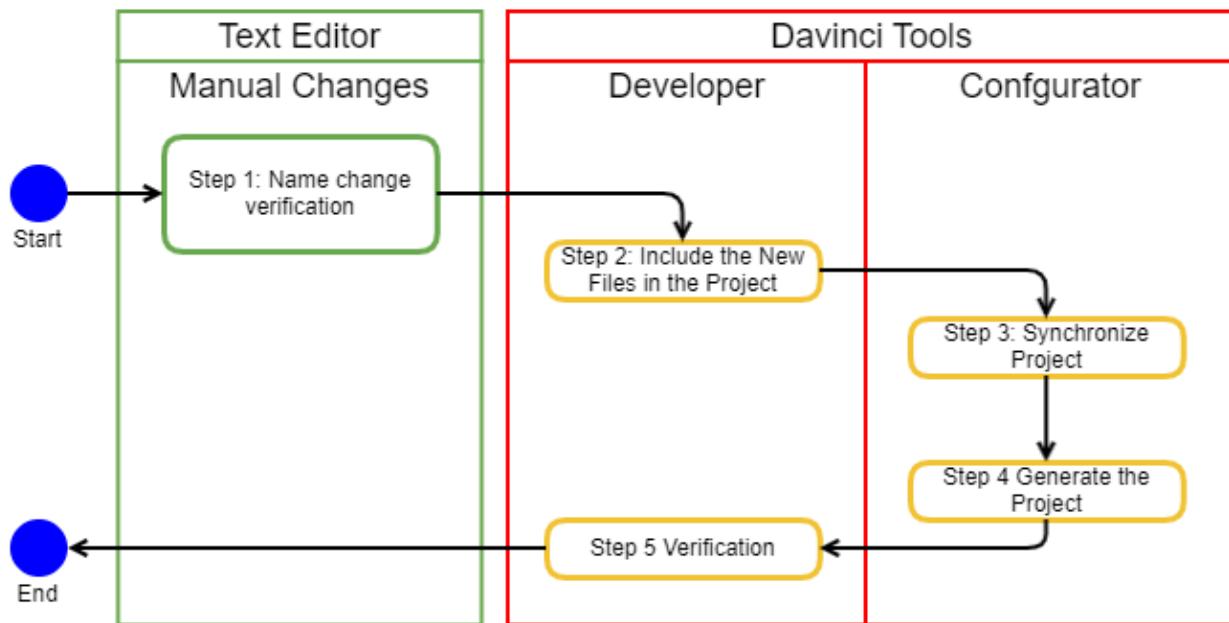


3.5 Troubleshooting

4 2_ Create new Tangible Software Component in a Project

- [Workflow Diagram\(see page 21\)](#)
- [Prerequisites\(see page 22\)](#)
- [Templates Files\(see page 22\)](#)
- [Steps\(see page 23\)](#)
- [Steps to be performed in a Text Editor\(see page 23\)
 - Step 1: Name change verification\(see page 23\)](#)
- [Steps to be performed in Davinci Developer Tool\(see page 23\)
 - Step 2: Include the New Files in the Project\(see page 23\)](#)
- [Steps to be performed in Davinci Configurator Tool\(see page 26\)
 - Step 3: Synchronize Project\(see page 26\)
 - Step 4: Generate the Project\(see page 27\)](#)
- [Steps to be performed in Davinci Developer Tool\(see page 29\)
 - Step 5: Verification\(see page 29\)](#)
- [Troubleshooting\(see page 31\)
 - Not able to edit the newly added SW Component in the Davinci Developer\(see page 31\)](#)
- [Workflow Diagram\(see page 32\)](#)
- [Prerequisites\(see page 32\)](#)
- [Templates Files\(see page 33\)](#)
- [Steps\(see page 33\)](#)
- [Steps to be performed in a Text Editor\(see page 33\)
 - Step 1: Name change verification\(see page 33\)](#)
- [Steps to be performed in Davinci Developer Tool\(see page 34\)
 - Step 2: Include the New Files in the Project\(see page 34\)](#)
- [Steps to be performed in Davinci Configurator Tool\(see page 37\)
 - Step 3: Synchronize Project\(see page 37\)
 - Step 4: Generate the Project\(see page 38\)](#)
- [Steps to be performed in Davinci Developer Tool\(see page 40\)
 - Step 5: Verification\(see page 40\)](#)
- [Troubleshooting\(see page 42\)
 - Not able to edit the newly added SW Component in the Davinci Developer\(see page 42\)](#)

4.1 Workflow Diagram



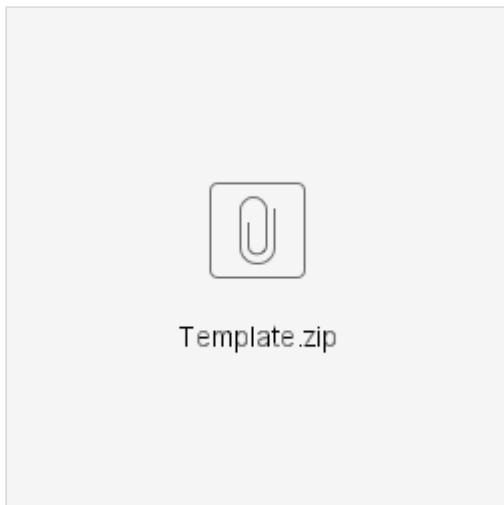
4.2 Prerequisites

1. Have a project up and running (having the input file and every previous detail).
2. Download the Template Files in this Confluence Page and rename the Files and the AR Packages in the

4.3 Templates Files

In the following ZIP file is the Template for a New SWC

Please consider the Template has a Name, Please change it both the file name and the AR-Package for every file.



4.4 Steps

4.5 Steps to be performed in a Text Editor

4.5.1 Step 1: Name change verification

After un zip the Templete Files, please copy them in the following path

..\\Config\\Developer\\ComponentTypes

Before starting the integration process please verify all the names are chnaged to the proper one. Use the following image as guide

```

FOLDERS
Template
  Template_ApplicationDataTypes.arxml
  Template_ComponentTypes.arxml
  Template_Compumethods.arxml
  Template_DataConstraints.arxml
  Template_DatatypeMappingSets.arxml
  Template_ImplementationDataTypes.arxml
  Template_Interfaces.arxml
  Template_ModeGroups.arxml
  Template_SWC.arxml
  Template_SWCType.arxml
  Template_Systems.arxml

Template_ComponentTypes.arxml
1 <?xml version="1.0" encoding="utf-8"?>
2 <!--This file was saved with a tool from Vector Informatik GmbH-->
3 <AUTOSAR xsi:schemaLocation="http://autosar.org/schema/r4.0
AUTOSAR_00048.xsd" xmlns="http://autosar.org/schema/r4.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
4   <AR-PACKAGES>
5     <AR-PACKAGE UUID="8f01c409-2ae1-454a-a0ca-fa3db8a3e34b">
6       <SHORT-NAME>Template</SHORT-NAME>
7       <AR-PACKAGES>
8         <AR-PACKAGE UUID="9000AAFD-B7E6-491E-8AA3-CBFF18E9A8E2">
9           <SHORT-NAME>ComponentTypes</SHORT-NAME>
10        </AR-PACKAGES>
11      </AR-PACKAGE>
12    </AR-PACKAGES>
13  </AR-PACKAGES>
14 </AUTOSAR>
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35

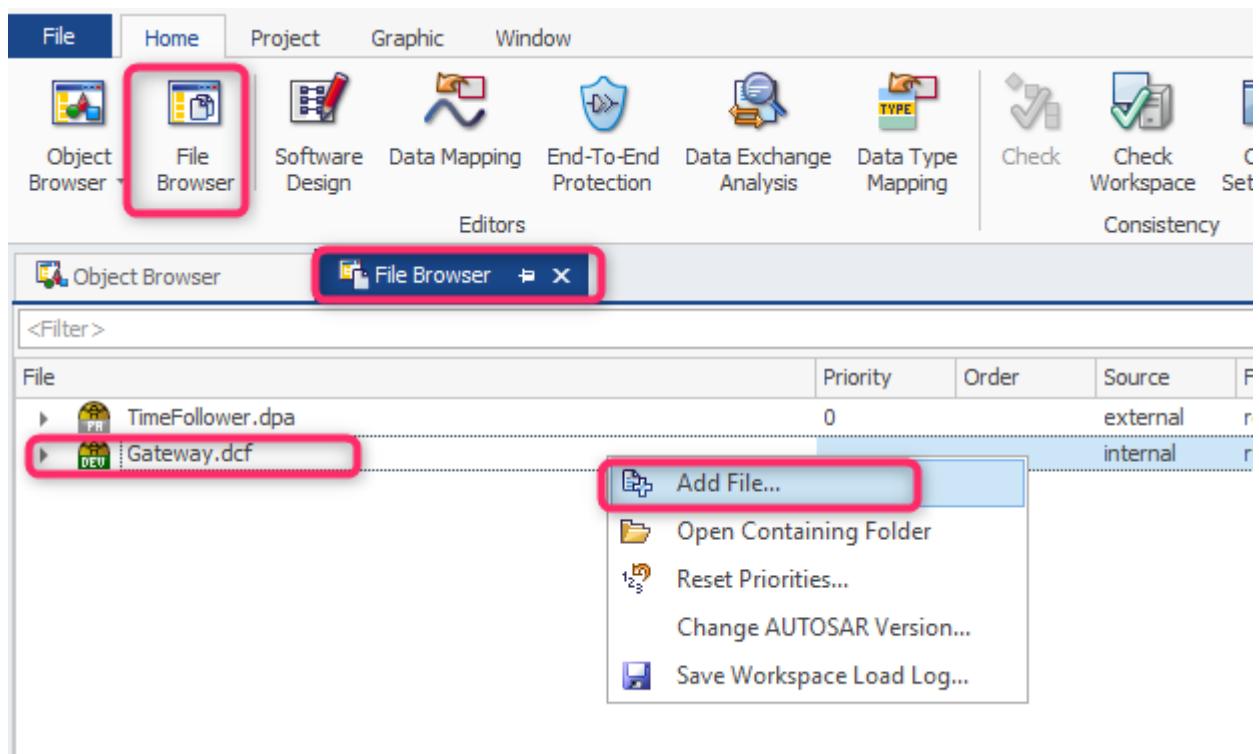
Template_SWCType.arxml
1 <?xml version="1.0" encoding="utf-8"?>
2 <!--This file was saved with a tool from Vector Informatik GmbH-->
3 <AUTOSAR xsi:schemaLocation="http://autosar.org/schema/r4.0
AUTOSAR_00048.xsd" xmlns="http://autosar.org/schema/r4.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
4   <AR-PACKAGES>
5     <AR-PACKAGE UUID="5f011f9b-14a1-4596-9e80-e1d315619065">
6       <SHORT-NAME>Template</SHORT-NAME>
7       <ELEMENTS>
8         <APPLICATION-SW-COMPONENT-TYPE UUID="6D4EC6C62-A1E9-4900-B2D9-FF8506A12776">
9           <SHORT-NAME>Template_Swctype</SHORT-NAME>
10          <PORTS>
11            </PORTS>
12          <INTERNAL-BEHAVIORS>
13            <SWC-INTERNAL-BEHAVIOR UUID="1662D7D5-10B5-4007-BE33-F47E04BAE26">
14              <SHORT-NAME>Template_Swctype_InternalBehavior</SHORT-NAME>
15              <DATA-TYPE-MAPPING-REFS>
16                </DATA-TYPE-MAPPING-REFS>
17              <EVENTS>
18                </EVENTS>
19              <PORT-API-OPTIONS>
20                </PORT-API-OPTIONS>
21              <RUNNABLES>
22                </RUNNABLES>
23                <SUPPORTS-MULTIPLE-INSTANTIATION>false</SUPPORTS-MULTIPLE-INSTANTIATION>
24                <SWC-INTERNAL-BEHAVIOR>
25                  <INTERNAL-BEHAVIORS>
26                    <APPLICATION-SW-COMPONENT-TYPE>
27                      <SWC-IMPLEMENTATION UUID="000342D4-A511-4F53-85CD-776C1075479">
28                        <SHORT-NAME>Template_Swctype_Implementation</SHORT-NAME>
29                        <PROGRAMMING-LANGUAGE>C</PROGRAMMING-LANGUAGE>
30                          <BEHAVIOR-REF DEST="SWC-INTERNAL-BEHAVIOR">Template_Swctype/Template_Swctype_InternalBehavior</BEHAVIOR-REF>
31                        </SWC-IMPLEMENTATION>
32                      </ELEMENTS>
33                    </AR-PACKAGE>
34                  </AR-PACKAGES>
35 </AUTOSAR>

```

4.6 Steps to be performed in Davinci Developer Tool

4.6.1 Step 2: Include the New Files in the Project

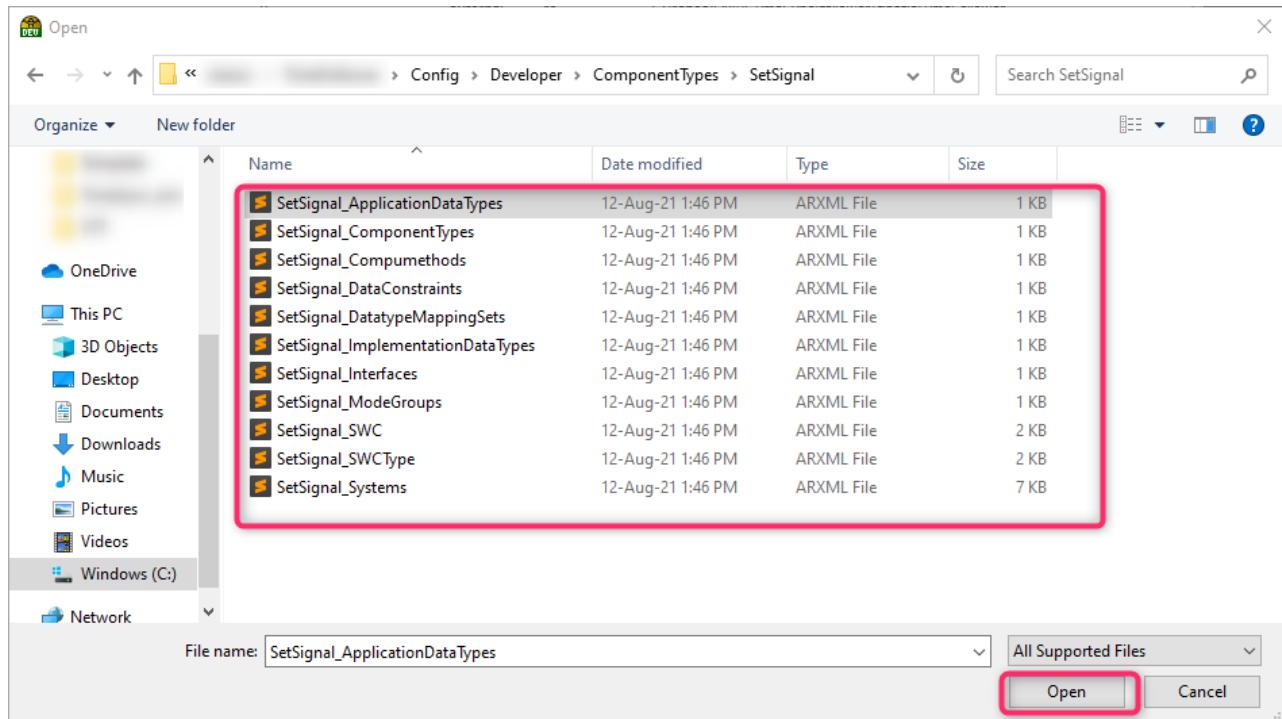
Go to File Browser → (Project Name).dfc → Right Click and select Add File.



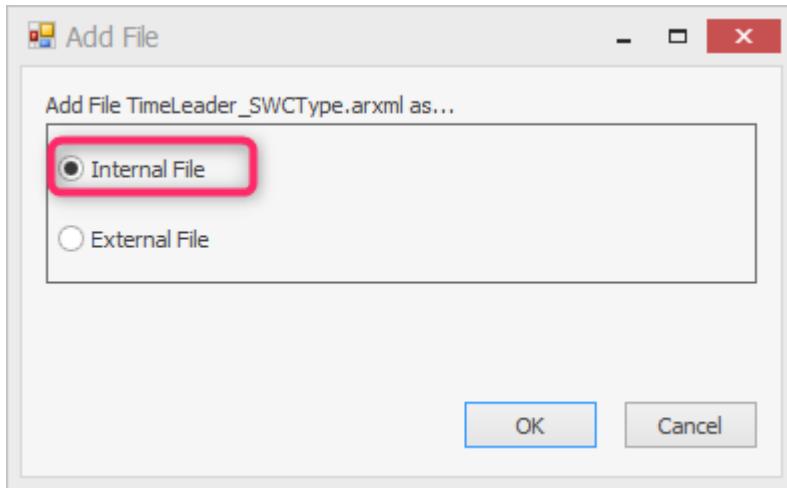
This will open a Window pop-up to select the File to be added.

Go to the Path in which the SWC Files are and select it and Click Open

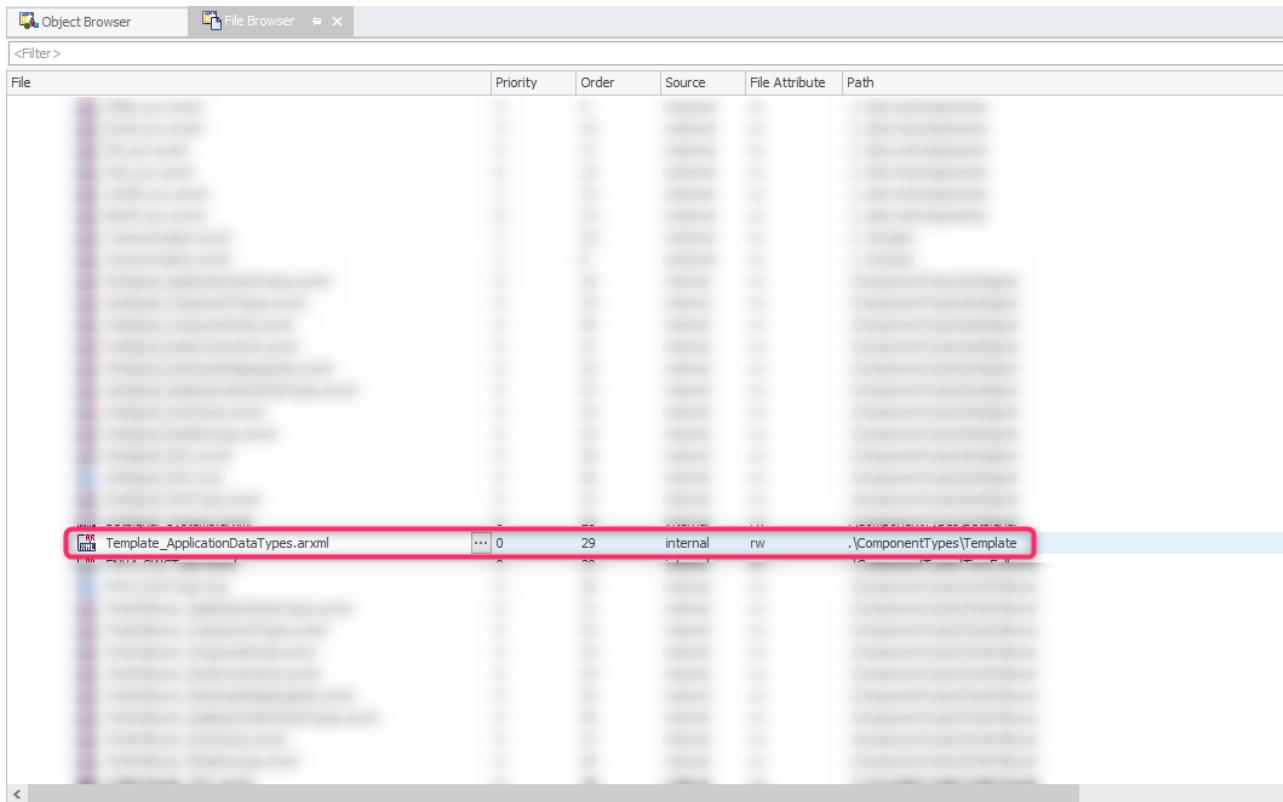
Note: The Davinci Tool only allow one file at the time, so this will need to be repeat it until all the files are added.



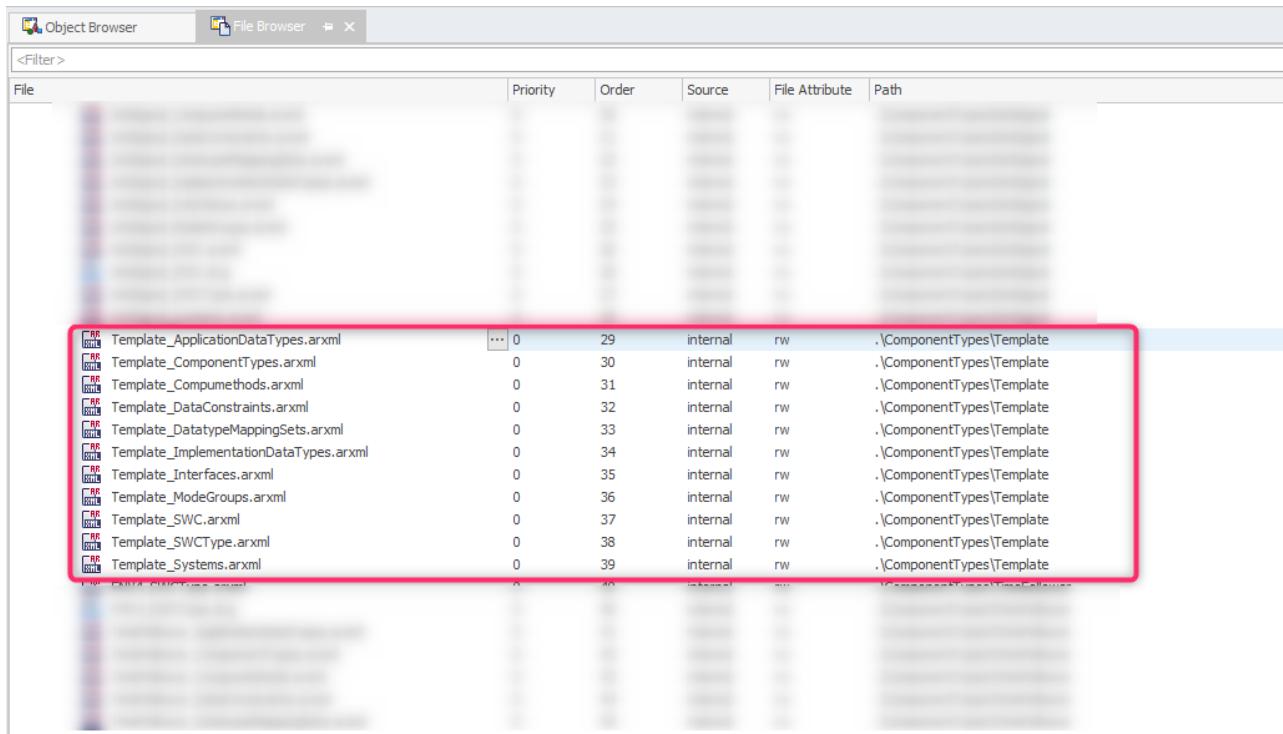
After this, the Davinci will show a Popup woth two option, please select **Internal (this will allow to edit the SWC)**.



After this the file shall be added as the image below



Now repeat the process for all the files until all are added, you shouls end with something like the following image.

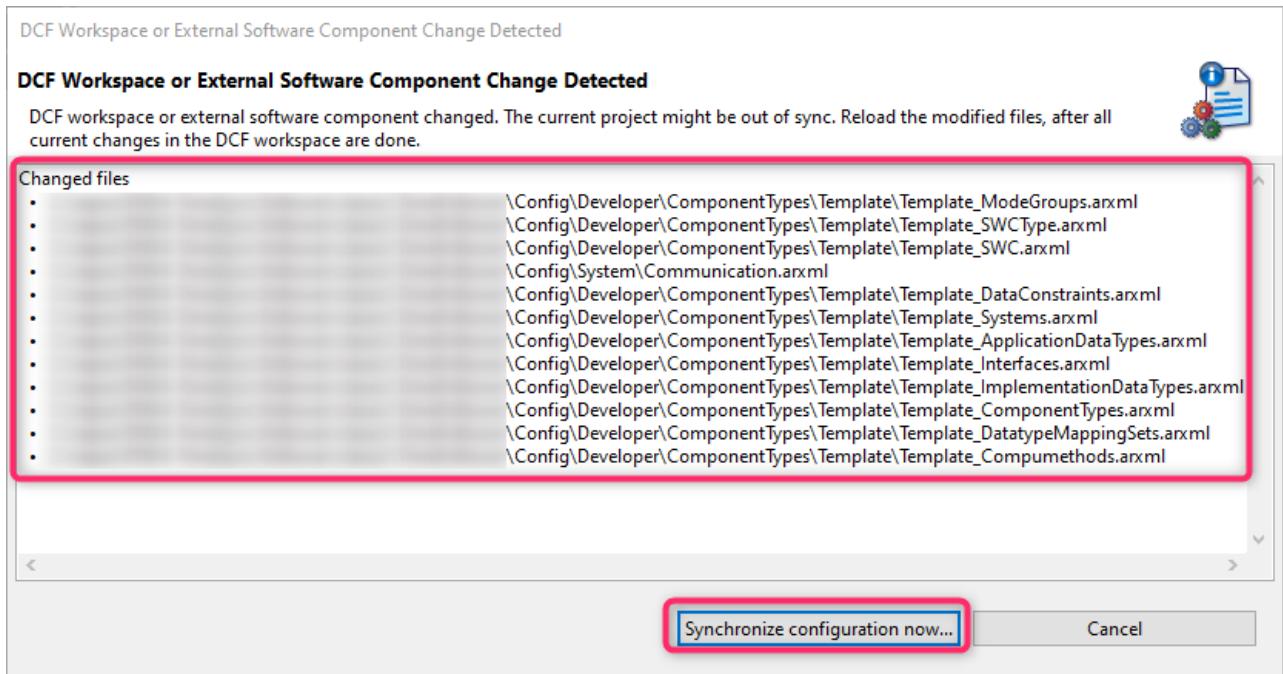


File	Priority	Order	Source	File Attribute	Path
Template_ApplicationDataTypes.arxml	[...]	0	29	internal	rw .\ComponentTypes\Template
Template_ComponentTypes.arxml		0	30	internal	rw .\ComponentTypes\Template
Template_Compumethods.arxml		0	31	internal	rw .\ComponentTypes\Template
Template_DataConstraints.arxml		0	32	internal	rw .\ComponentTypes\Template
Template_DatatypeMappingSets.arxml		0	33	internal	rw .\ComponentTypes\Template
Template_ImplementationDataTypes.arxml		0	34	internal	rw .\ComponentTypes\Template
Template_Interfaces.arxml		0	35	internal	rw .\ComponentTypes\Template
Template_ModeGroups.arxml		0	36	internal	rw .\ComponentTypes\Template
Template_SWC.arxml		0	37	internal	rw .\ComponentTypes\Template
Template_SWCType.arxml		0	38	internal	rw .\ComponentTypes\Template
Template_Systems.arxml		0	39	internal	rw .\ComponentTypes\Template

4.7 Steps to be performed in Davinci Configurator Tool

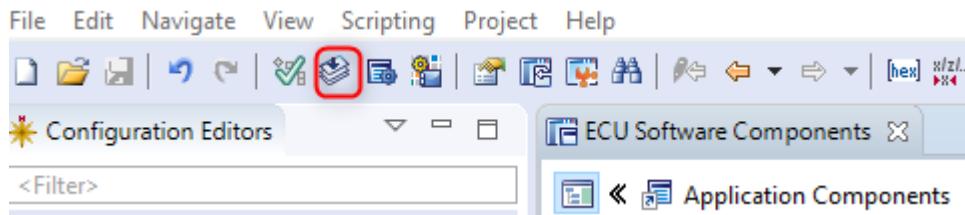
4.7.1 Step 3: Synchronize Project

After saving the Project in Davinci Developer, the Davinci Configurator will show a PopUp window requesting the Synchronization.

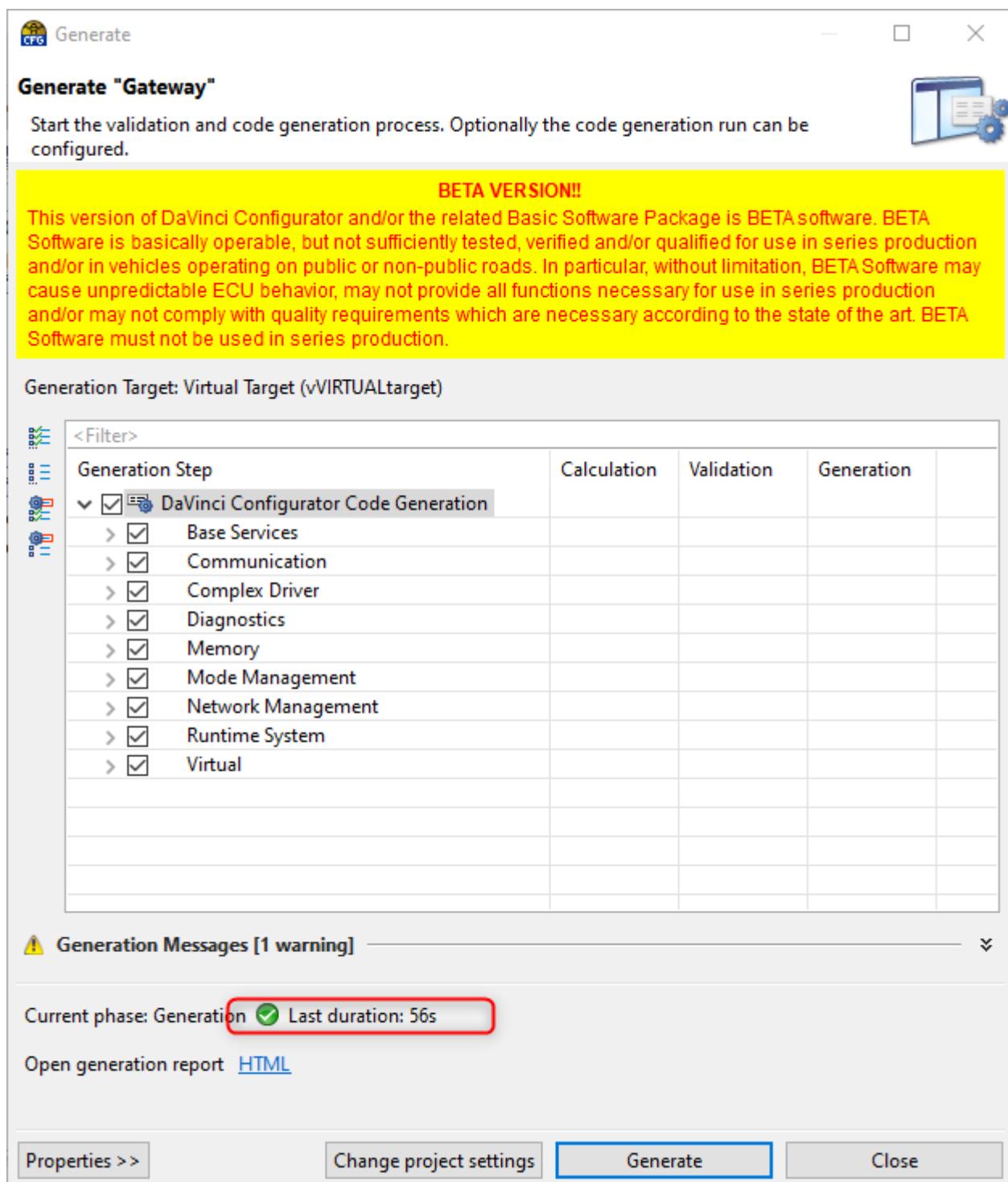


4.7.2 Step 4: Generate the Project

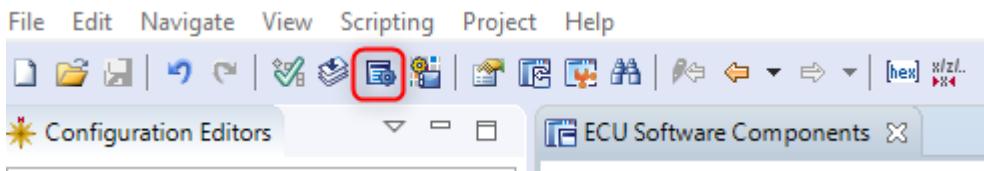
Go to Generate (Or click F9)



And Generate as usual



And finally generate the corresponding templates

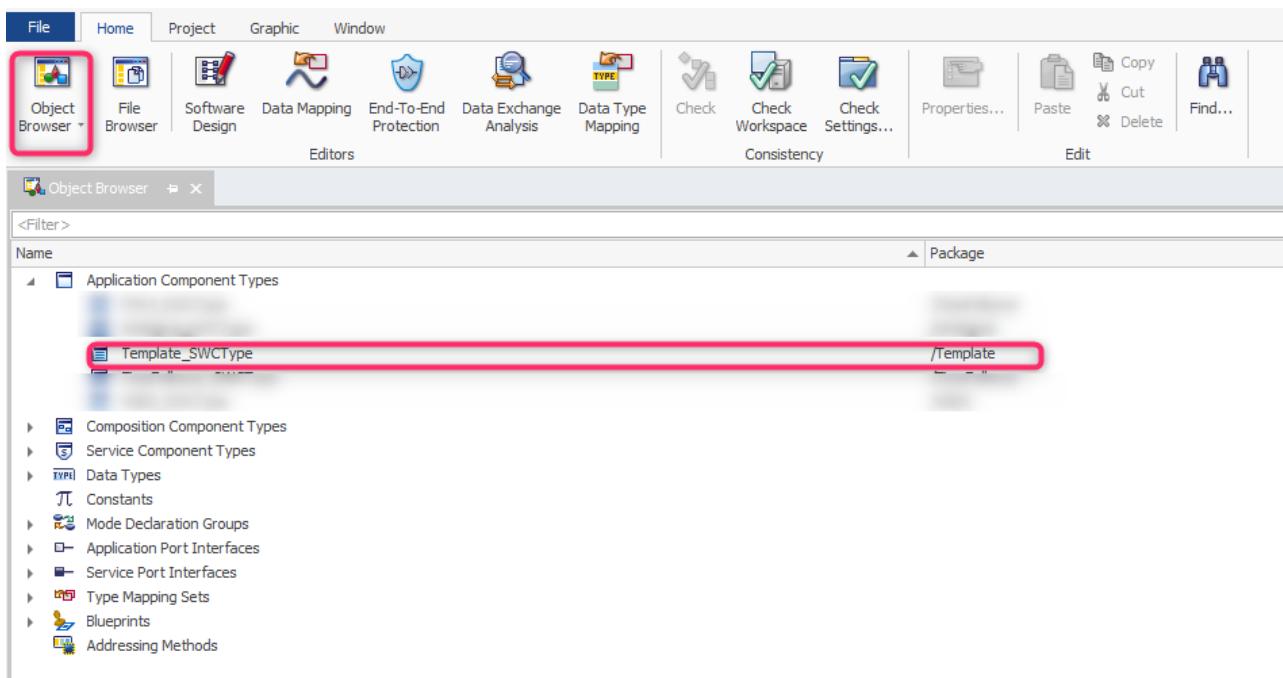


4.8 Steps to be performed in Davinci Developer Tool

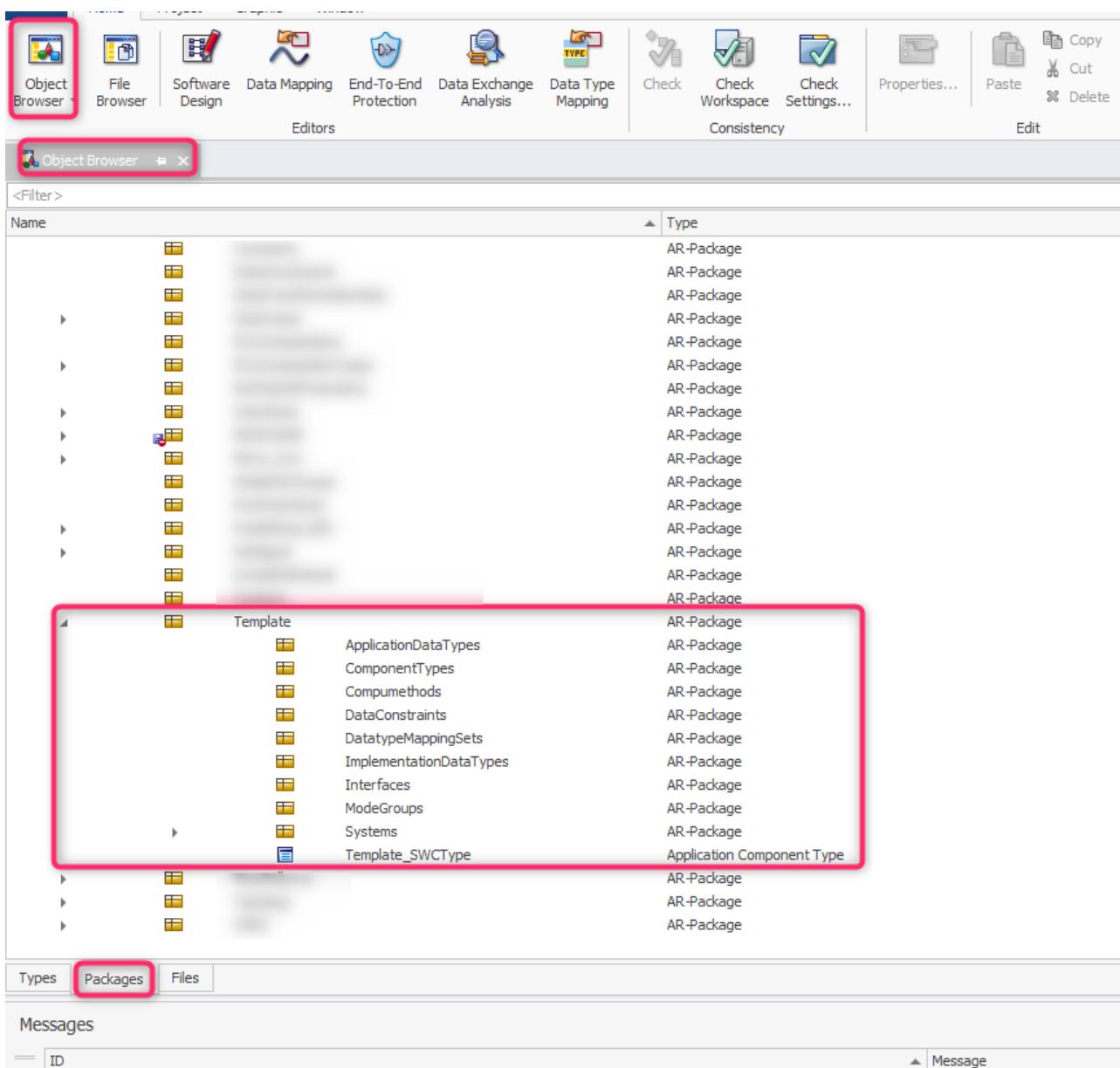
4.8.1 Step 5: Verification

After Generation please close and Reopen the Project using the Davinci Developer Tools.

Now go to Object Browser → Application Component Types and verify the New SWC is there.



Now go to Object Browser → Package and verify the SWC packages are created as show.



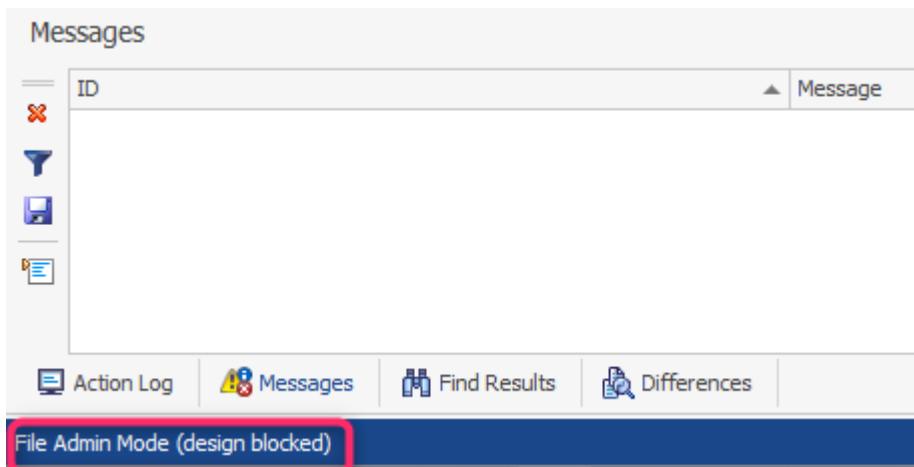
Note

This confluence page is only to create the Tangible SWC.
 If during the modification of this SWC (like adding ports or interfaces, etc) those changes are not done in the added files (See Step 1) please close all the DaVinci tools and move the changes manually to the correct files.
 This is a risky process but is the way to ensure a Tangible way of work.

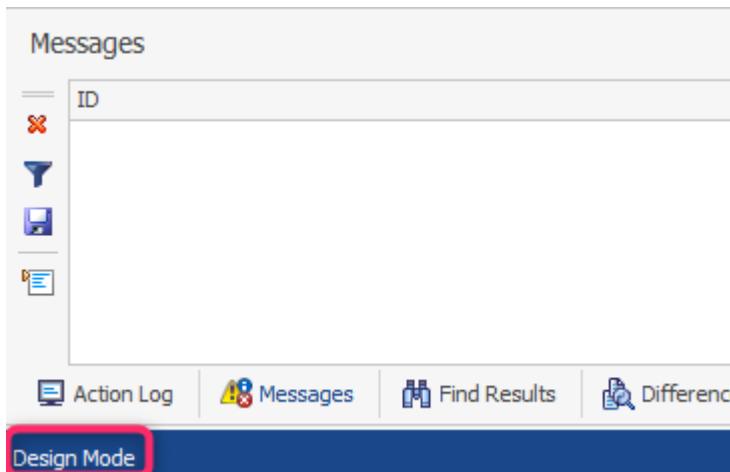
4.9 Troubleshooting

4.9.1 Not able to edit the newly added SW Component in the Davinci Developer

Verify the current mode be in Design mode. After modifying the files of the project, the Tool will change to Admin Mode which only allow to modify the files.



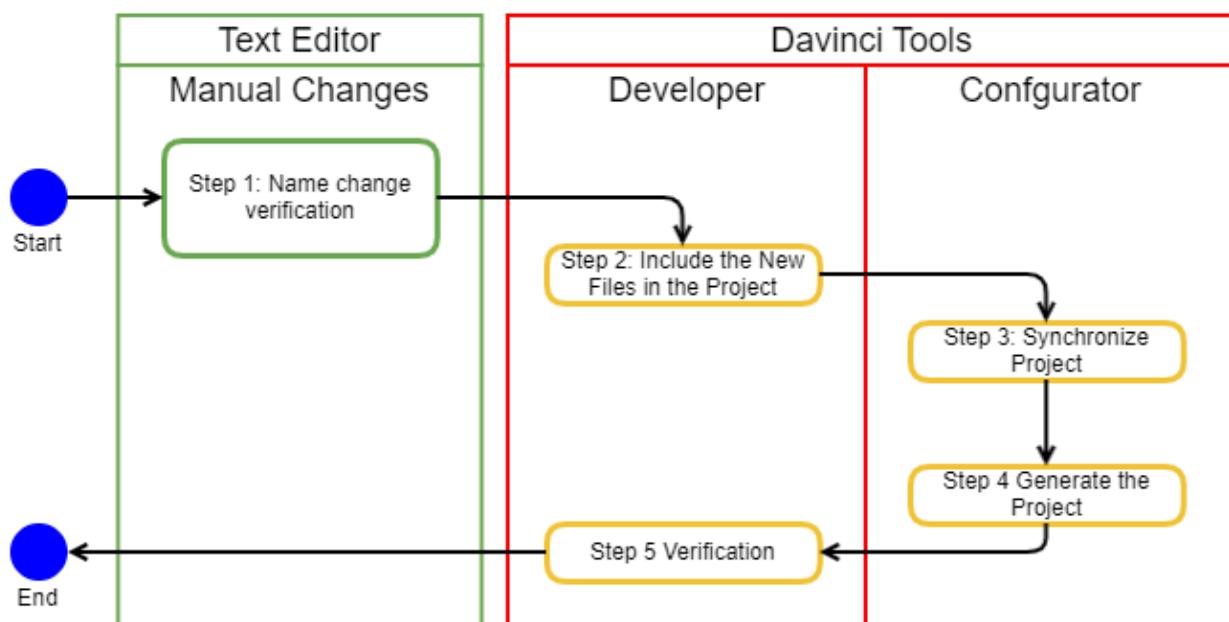
To change it back to Design mode just close the Project and re-open it.



- Workflow Diagram(see page 21)
- Prerequisites(see page 22)
- Templates Files(see page 22)
- Steps(see page 23)
- Steps to be performed in a Text Editor(see page 23)
 - Step 1: Name change verification(see page 23)
- Steps to be performed in Davinci Developer Tool(see page 23)
 - Step 2: Include the New Files in the Project(see page 23)
- Steps to be performed in Davinci Configurator Tool(see page 26)
 - Step 3: Synchronize Project(see page 26)
 - Step 4: Generate the Project(see page 27)

- Steps to be performed in Davinci Developer Tool(see page 29)
 - Step 5: Verification(see page 29)
- Troubleshooting(see page 31)
 - Not able to edit the newly added SW Component in the Davinci Developer(see page 31)
- Workflow Diagram(see page 32)
- Prerequisites(see page 32)
- Templates Files(see page 33)
- Steps(see page 33)
- Steps to be performed in a Text Editor(see page 33)
 - Step 1: Name change verification(see page 33)
- Steps to be performed in Davinci Developer Tool(see page 34)
 - Step 2: Include the New Files in the Project(see page 34)
- Steps to be performed in Davinci Configurator Tool(see page 37)
 - Step 3: Synchronize Project(see page 37)
 - Step 4: Generate the Project(see page 38)
- Steps to be performed in Davinci Developer Tool(see page 40)
 - Step 5: Verification(see page 40)
- Troubleshooting(see page 42)
 - Not able to edit the newly added SW Component in the Davinci Developer(see page 42)

4.10 Workflow Diagram



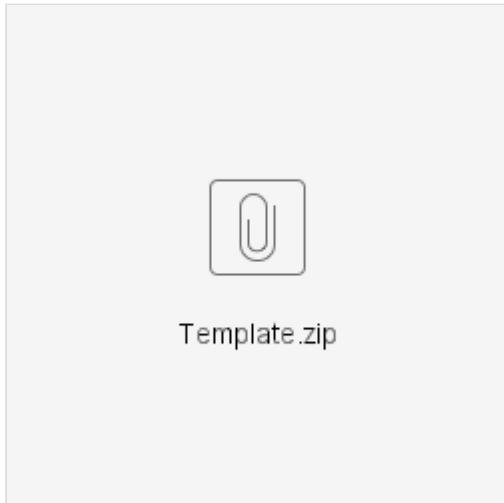
4.11 Prerequisites

1. Have a project up and running (having the input file and every previous detail).
2. Download the Template Files in this Confluence Page and rename the Files and the AR Packages in the

4.12 Templates Files

In the following ZIP file is the Template for a New SWC

Please consider the Template has a Name, Please change it both the file name and the AR-Package for every file.



4.13 Steps

4.14 Steps to be performed in a Text Editor

4.14.1 Step 1: Name change verification

After un zip the Templeate Files, please copy them in the following path

..\\Config\\Developer\\ComponentTypes

Before starting the integration process please verify all the names are chnaged to the proper one. Use the following image as guide

```

FOLDERS
Template
  Template_ApplicationDataTypes.arxml
  Template_ComponentTypes.arxml
  Template_Compumethods.arxml
  Template_DataConstraints.arxml
  Template_DatatypeMappingSets.arxml
  Template_ImplementationDataTypes.arxml
  Template_Interfaces.arxml
  Template_ModeGroups.arxml
  Template_SWC.arxml
  Template_SWType.arxml
  Template_Systems.arxml

Template_ComponentTypes.arxml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <!--This file was saved with a tool from Vector Informatik GmbH-->
3 <AUTOSAR xsi:schemalocation="http://autosar.org/schema/r4.0
AUTOSAR_00048.xsd" xmlns="http://autosar.org/schema/r4.0"
http://www.w3.org/2001/XMLSchema-instance">
4   <AR-PACKAGES>
5     <AR-PACKAGE UUID="8F01c490-2aae-4241-8C41-fa3db8a3e34b">
6       <SHORT-NAME>Template</SHORT-NAME>
7     <AR-PACKAGE><!-->
8       <AR-PACKAGE UUID="9000AAFD-B7E6-491E-8AA3-CBFF18E9A8E2">
9         <SHORT-NAME>ComponentTypes</SHORT-NAME>
10        </AR-PACKAGE>
11      </AR-PACKAGES>
12    </AR-PACKAGE>
13  </AR-PACKAGES>
14 </AUTOSAR>

```

```

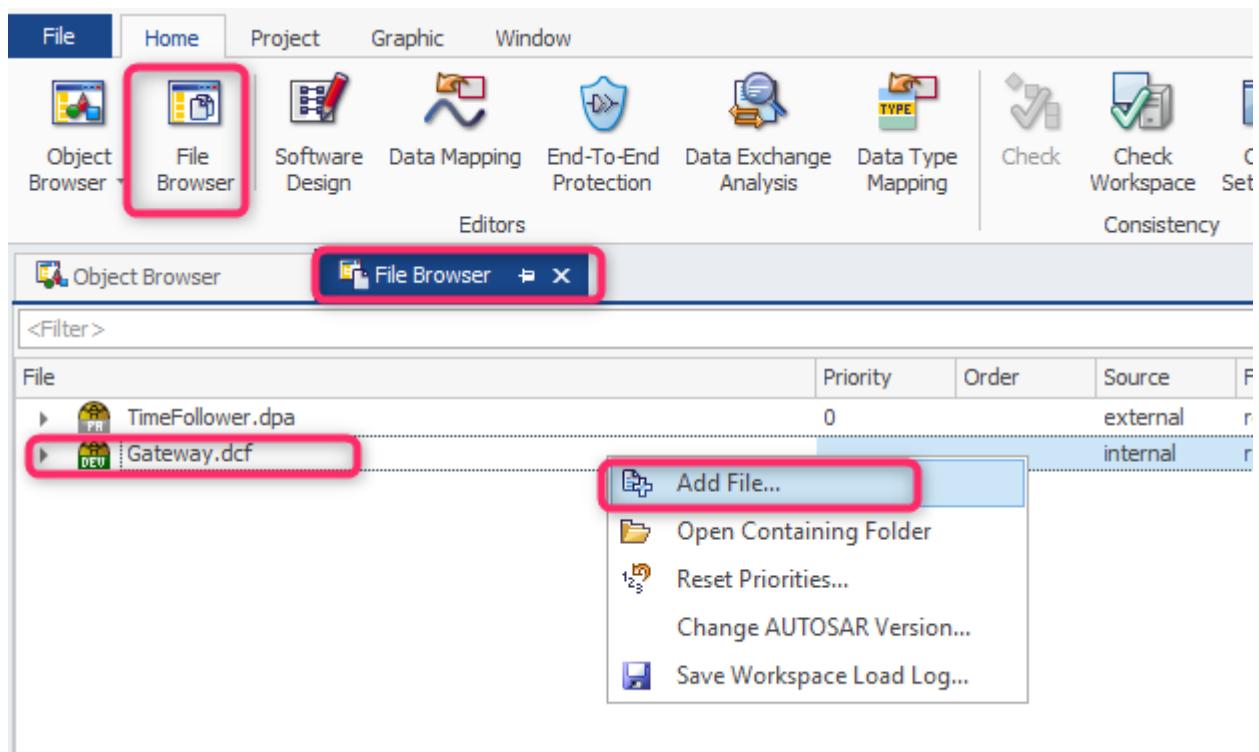
Template_SWType.arxml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <!--This file was saved with a tool from Vector Informatik GmbH-->
3 <AUTOSAR xsi:schemalocation="http://autosar.org/schema/r4.0
AUTOSAR_00048.xsd" xmlns="http://autosar.org/schema/r4.0"
http://www.w3.org/2001/XMLSchema-instance">
4   <AR-PACKAGES>
5     <AR-PACKAGE UUID="Ef01169b-14a1-4996-9e80-e1d315619065">
6       <SHORT-NAME>Template</SHORT-NAME>
7       <ELEMENTS>
8         <APPLICATION-SW-COMPONENT-TYPE UUID="6D4EC662-A1E9-4900-93DD-EF8506A12776">
9           <SHORT-NAME>Template_SWType</SHORT-NAME>
10          <PORTS>
11        </PORTS>
12        <INTERNAL-BEHAVIORS>
13          <SMC-INTERNAL-BEHAVIOR UUID="1662D7D5-1DB5-4897-BE32-FE47E04BAE26">
14            <SHORT-NAME>Template_SWType_InternalBehavior</SHORT-NAME>
15            <DATA-TYPE-MAPPING-REFS>
16            <EVENTS>
17            <PORT-API-OPTIONS>
18            <PORT-API-OPTIONS>
19            <RUNNABLES>
20            <RUNNABLES>
21            <SUPPORTS-MULTIPLE-INSTANTIATION>false</SUPPORTS-MULTIPLE-INSTANTIATION>
22            <SMC-INTERNAL-BEHAVIOR>
23            <INTERNAL-BEHAVIORS>
24              <APPLICATION-SW-COMPONENT-TYPE>
25                <SMC-IMPLEMENTATION UUID="00A3A2D4-A511-4A9E-94E0-5776C1075479">
26                  <SHORT-NAME>Template_SWType_Implementation</SHORT-NAME>
27                  <PROGRAMMING-LANGUAGE>C</PROGRAMMING-LANGUAGE>
28                  <BSM/ATOP-REF DEST="SMC-INTERNAL-BEHAVIOR">/Template/
29                    <Template_SWType/Template_SWType_InternalBehavior</Template_SWType>
30                  <DEPLOYMENT-NET>
31                </SMC-IMPLEMENTATION>
32              </INTERNAL-BEHAVIORS>
33            </ELEMENTS>
34          </AR-PACKAGE>
35        </AR-PACKAGES>
36 </AUTOSAR>

```

4.15 Steps to be performed in Davinci Developer Tool

4.15.1 Step 2: Include the New Files in the Project

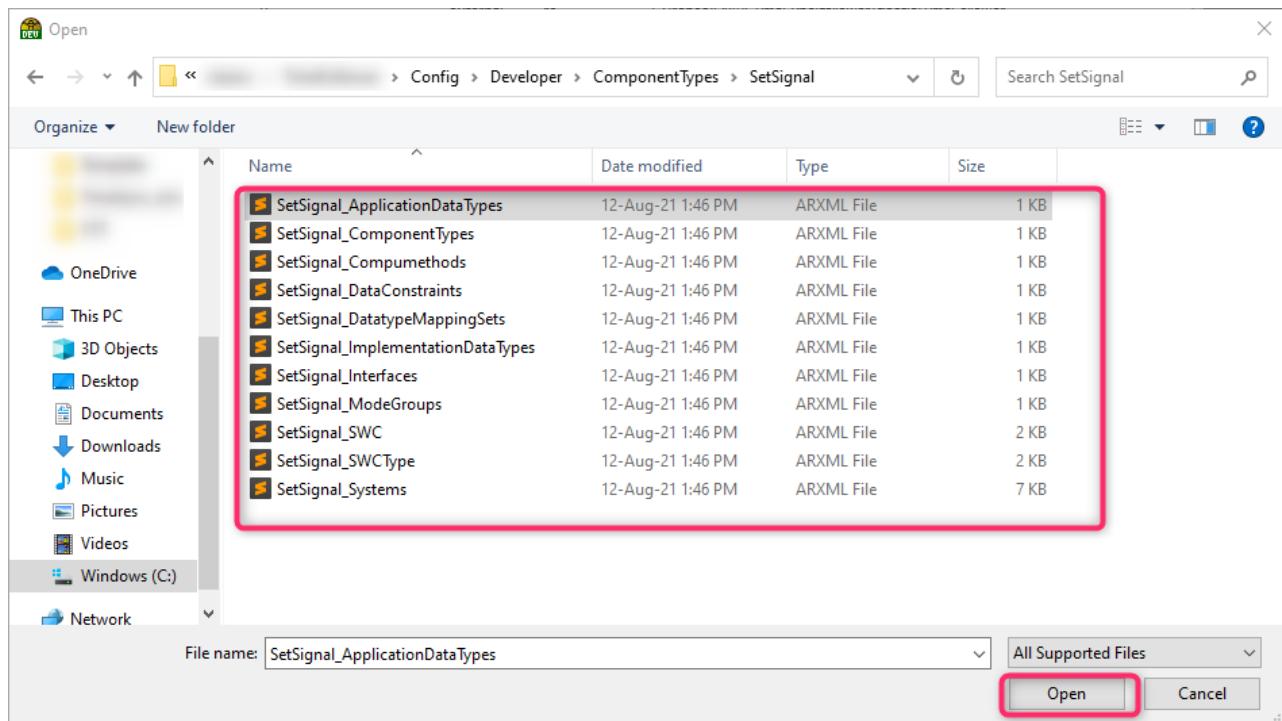
Go to File Browser → (Project Name).dcf → Right Click and select Add File.



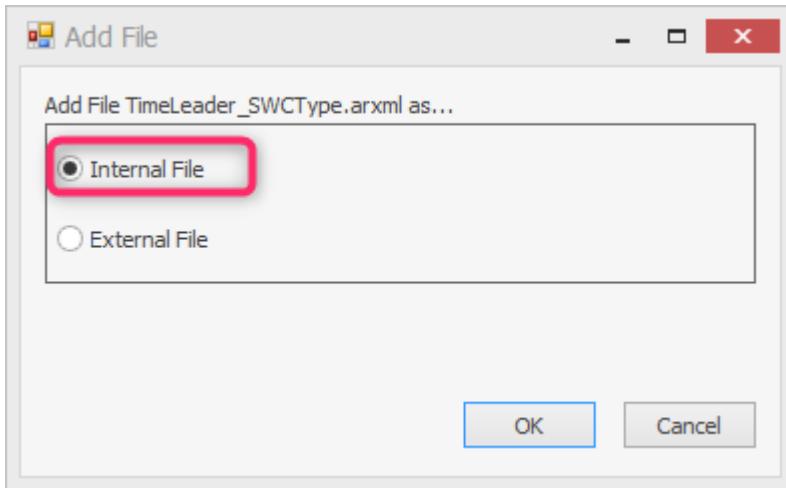
This will open a Window pop-up to select the File to be added.

Go to the Path in which the SWC Files are and select it and Click Open

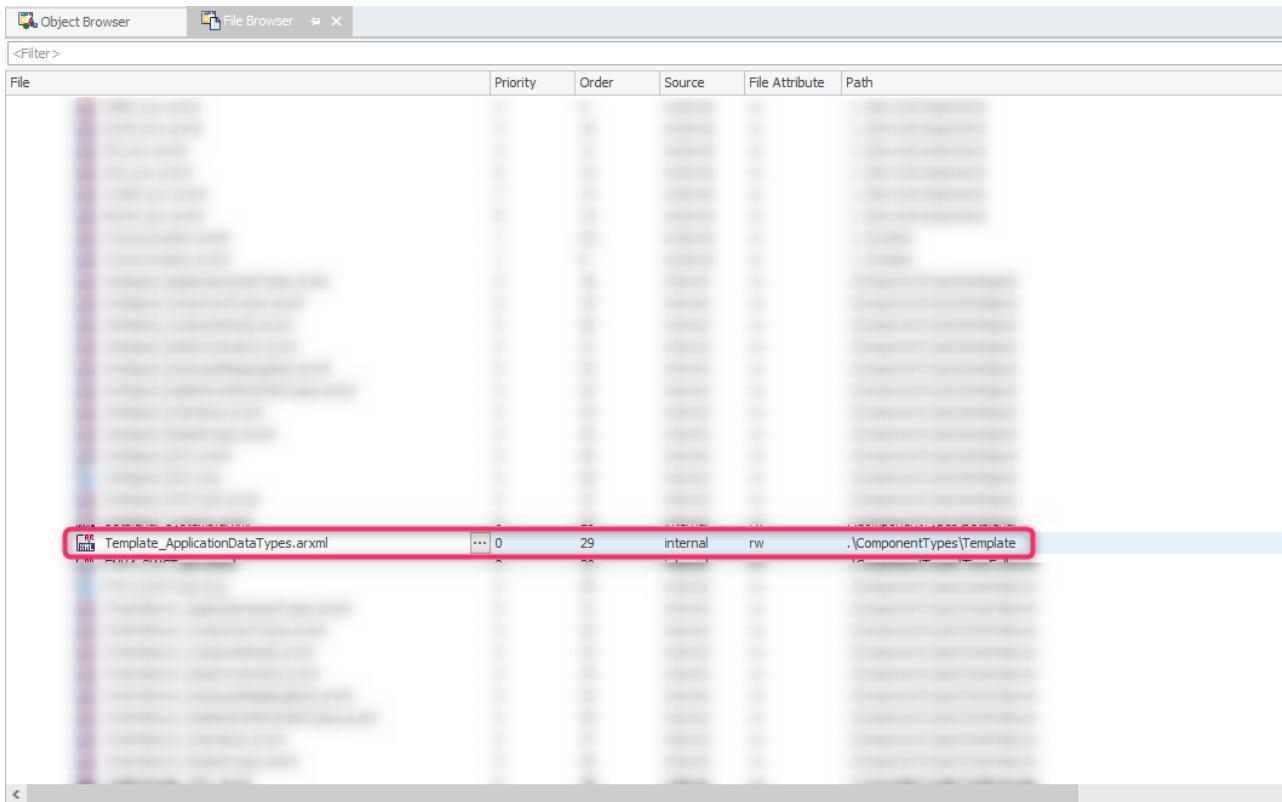
Note: The Davinci Tool only allow one file at the time, so this will need to be repeat it until all the files are added.



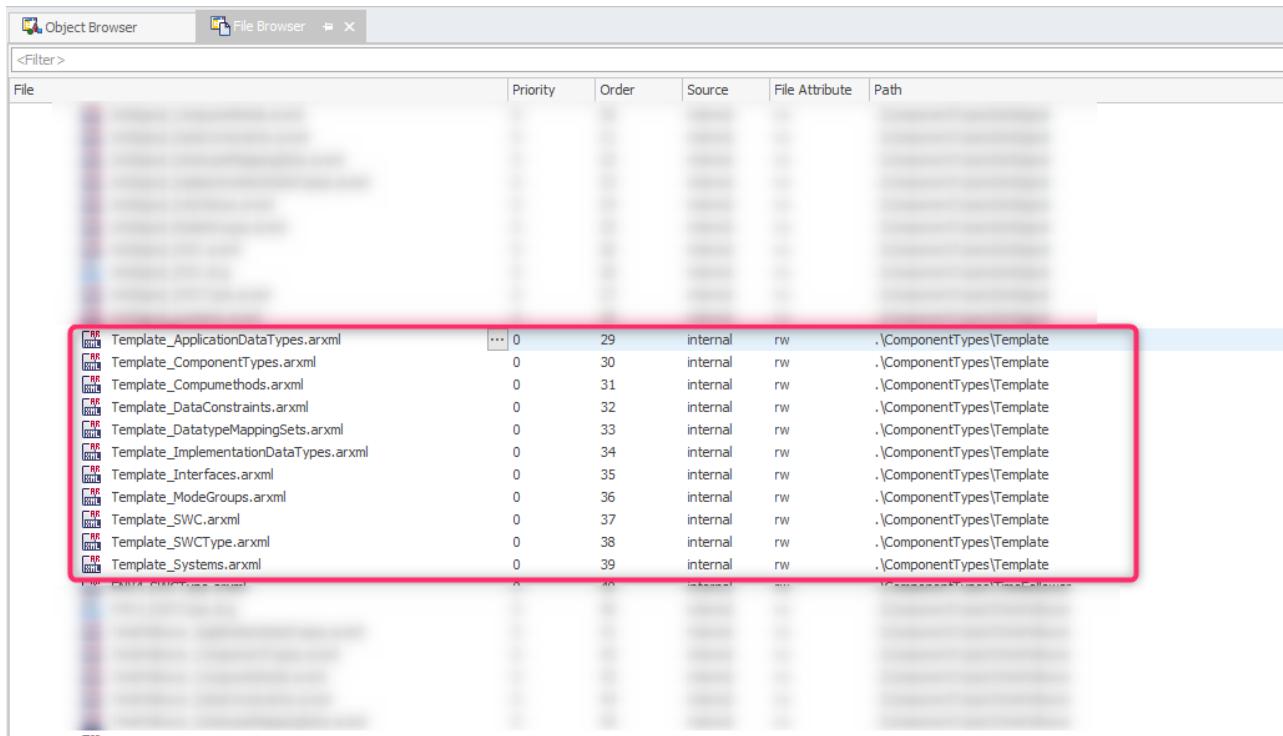
After this, the Davinci will show a Popup woth two option, please select **Internal (this will allow to edit the SWC)**.



After this the file shall be added as the image below



Now repeat the process for all the files until all are added, you shouls end with something like the following image.

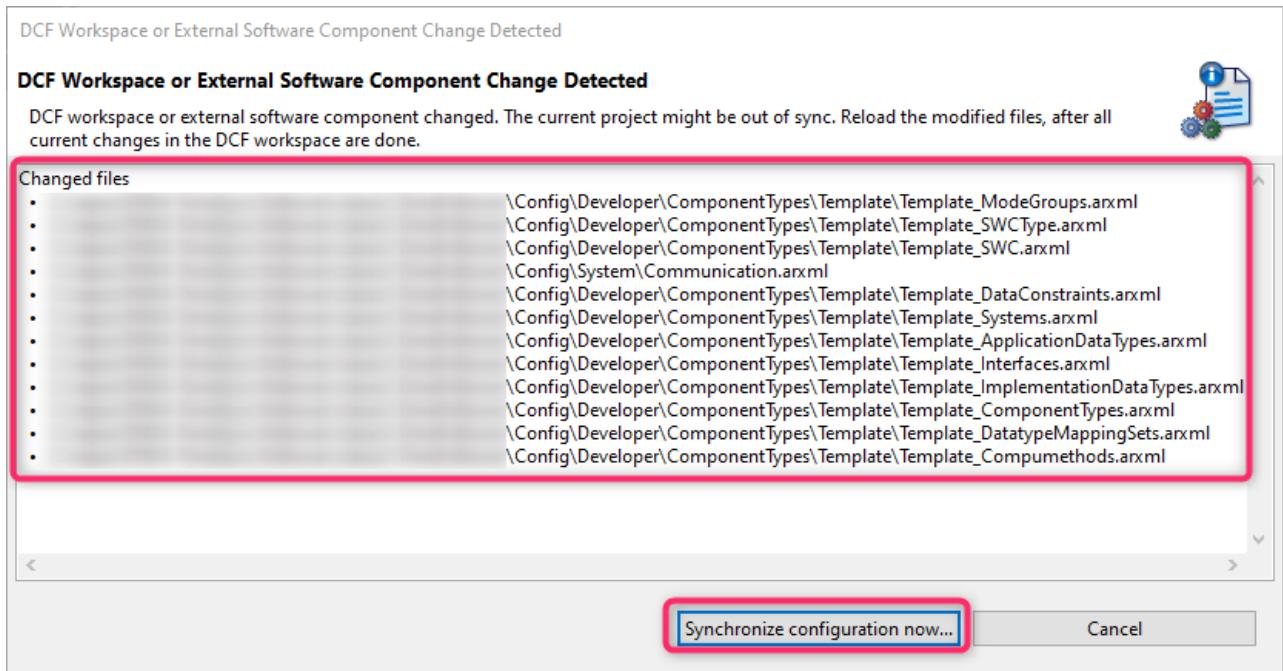


File	Priority	Order	Source	File Attribute	Path
Template_ApplicationDataTypes.arxml	0	29	internal	rw	.\\ComponentTypes\\Template
Template_ComponentTypes.arxml	0	30	internal	rw	.\\ComponentTypes\\Template
Template_Compumethods.arxml	0	31	internal	rw	.\\ComponentTypes\\Template
Template_DataConstraints.arxml	0	32	internal	rw	.\\ComponentTypes\\Template
Template_DatatypeMappingSets.arxml	0	33	internal	rw	.\\ComponentTypes\\Template
Template_ImplementationDataTypes.arxml	0	34	internal	rw	.\\ComponentTypes\\Template
Template_Interfaces.arxml	0	35	internal	rw	.\\ComponentTypes\\Template
Template_ModeGroups.arxml	0	36	internal	rw	.\\ComponentTypes\\Template
Template_SWC.arxml	0	37	internal	rw	.\\ComponentTypes\\Template
Template_SWCType.arxml	0	38	internal	rw	.\\ComponentTypes\\Template
Template_Systems.arxml	0	39	internal	rw	.\\ComponentTypes\\Template

4.16 Steps to be performed in Davinci Configurator Tool

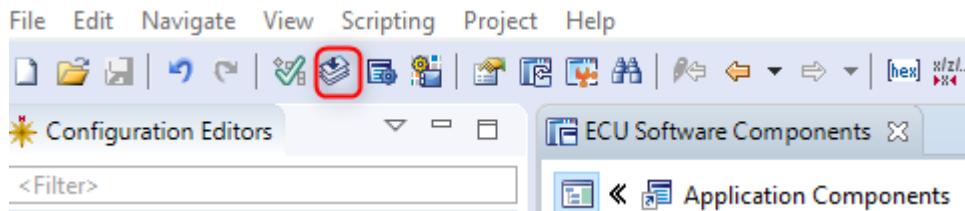
4.16.1 Step 3: Synchronize Project

After saving the Project in Davinci Developer, the Davinci Configurator will show a PopUp window requesting the Synchronization.

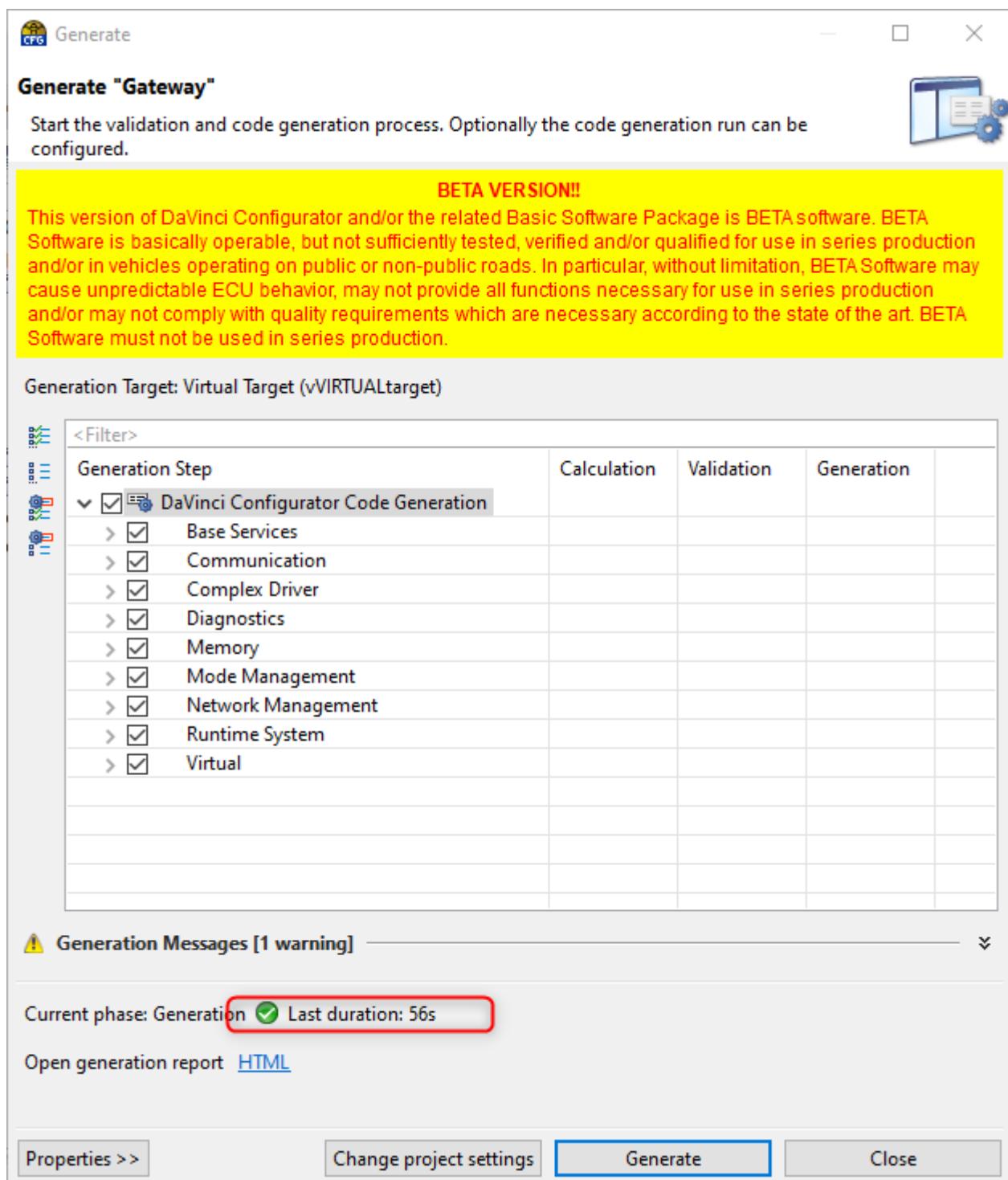


4.16.2 Step 4: Generate the Project

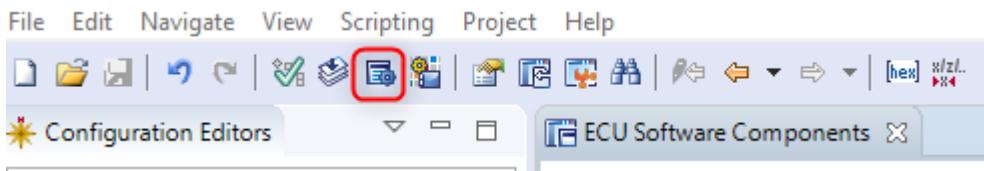
Go to Generate (Or click F9)



And Generate as usual



And finally generate the corresponding templates

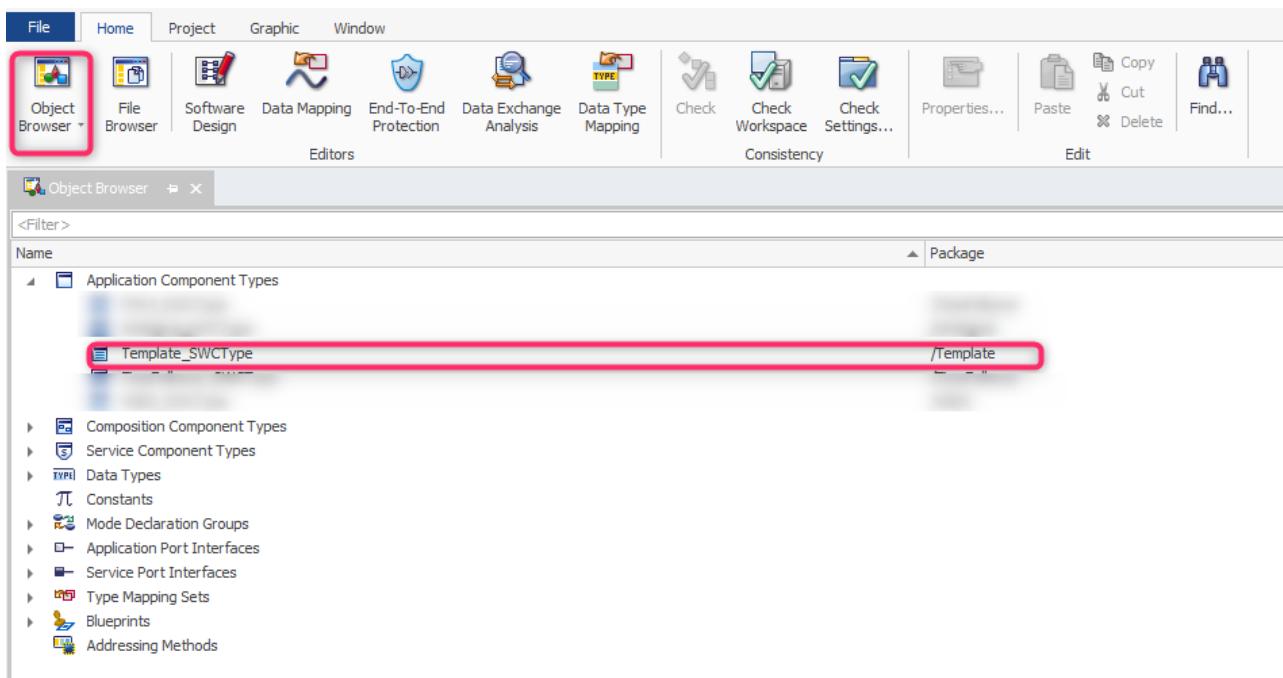


4.17 Steps to be performed in Davinci Developer Tool

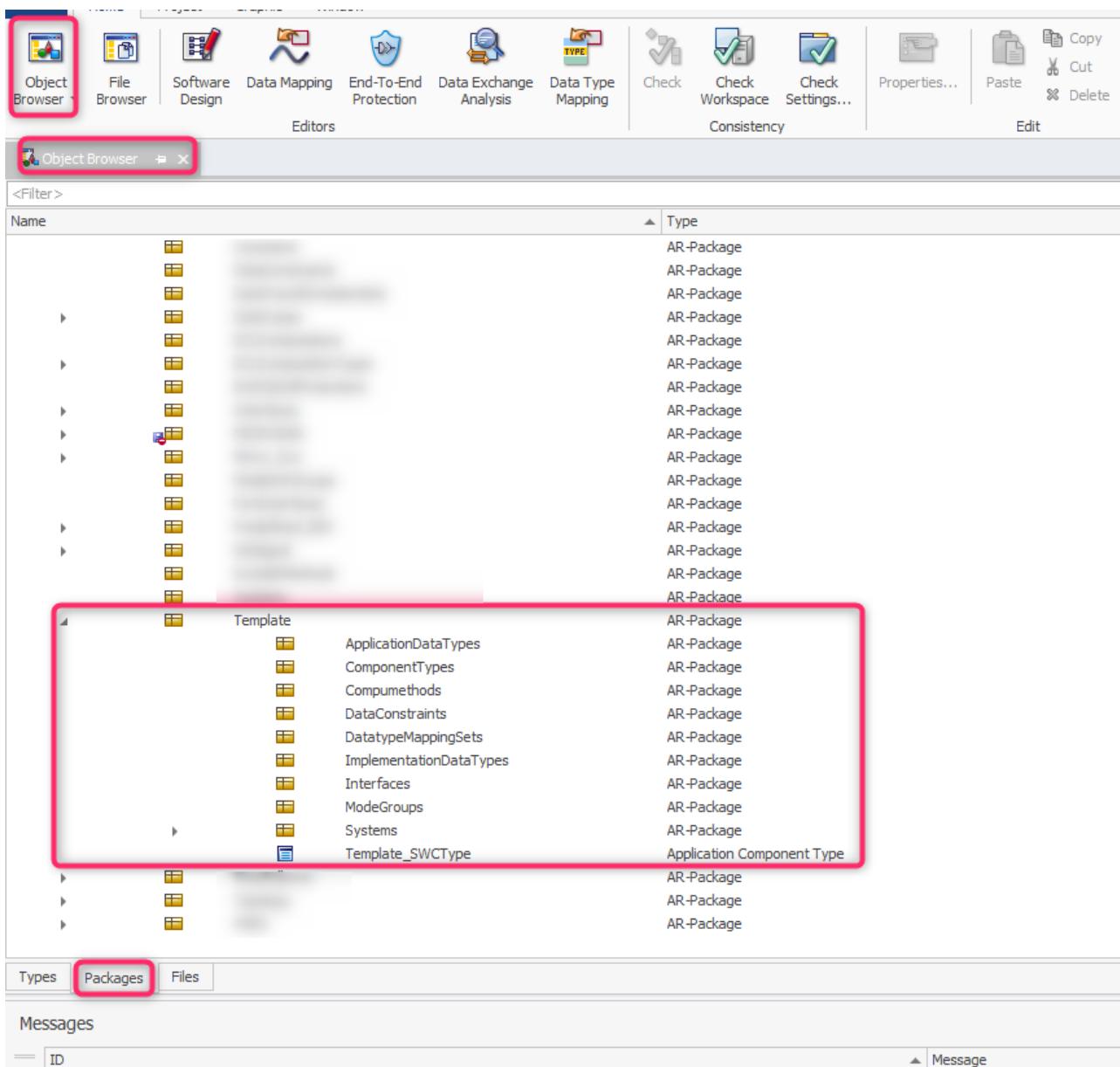
4.17.1 Step 5: Verification

After Generation please close and Reopen the Project using the Davinci Developer Tools.

Now go to Object Browser → Application Component Types and verify the New SWC is there.



Now go to Object Browser → Package and verify the SWC packages are created as show.



Note

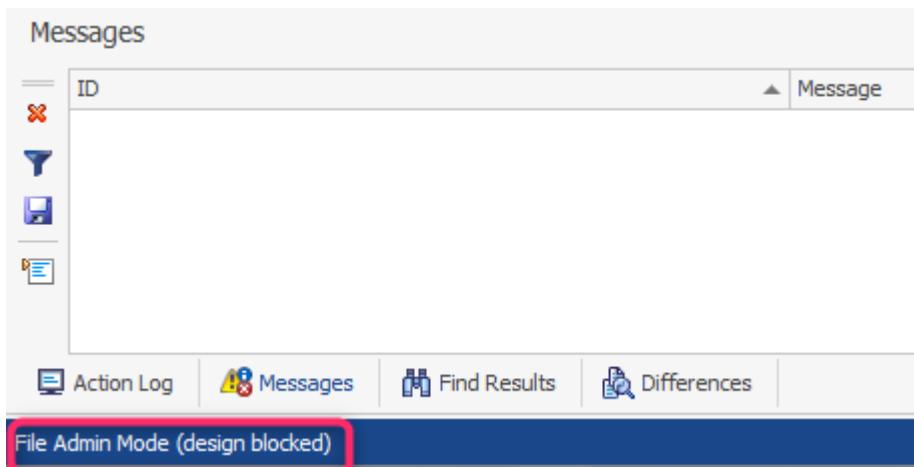
This confluence page is only to create the Tangible SWC.
 If during the modification of this SWC (like adding ports or interfaces, etc) those changes are not done in the added files (See Step 1) please close all the DaVinci tools and move the changes manually to the correct files.

This is a risky process but is the way to ensure a Tangible way of work.

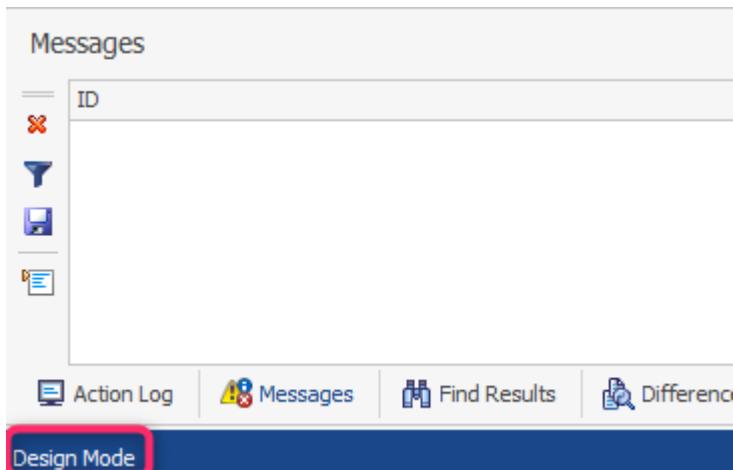
4.18 Troubleshooting

4.18.1 Not able to edit the newly added SW Component in the Davinci Developer

Verify the current mode be in Design mode. After modifying the files of the project, the Tool will change to Admin Mode which only allow to modify the files.



To change it back to Design mode just close the Project and re-open it.



5 3_Create a New Compu Method as Enum

- [Context](#)(see page 43)
- [Workflow Diagram](#)(see page 43)
- [Prerequisites](#)(see page 44)
- [Steps](#)(see page 44)
- [Steps to be performed in Davinci Developer Tool](#)(see page 44)
 - [Step 1: Go to package view of the project](#)(see page 44)
 - [Step 2: Create the Compu Method](#)(see page 46)
 - [Step 3: Fill the "New Compu Method" Window](#)(see page 49)
- [Steps to be performed in Davinci Configurator Tool](#)(see page 50)
 - [Step 4: Synchronize Project](#)(see page 50)
 - [Step 5: Generate the Project](#)(see page 51)
- [Troubleshooting](#)(see page 53)

5.1 Context

AUTOSAR Def

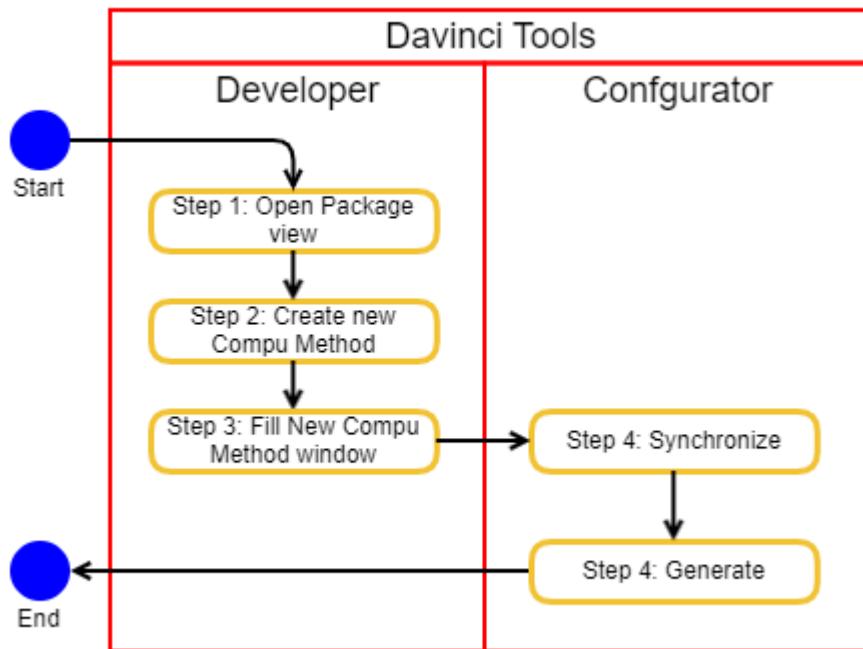
This meta-**class** represents the ability to express the relationship between a physical value and the mathematical representation.

Note that **this** is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant.

CompuMethods in general shall be reuseable and not fixed to a specific data type. CompuMethods can be referenced by any kind of datatype which need such described computational features; the reuse is strongly recommended and mandatory **for float** datatypes definition.

With Compu Method is possible to create enum structure like in C language.

5.2 Workflow Diagram



5.3 Prerequisites

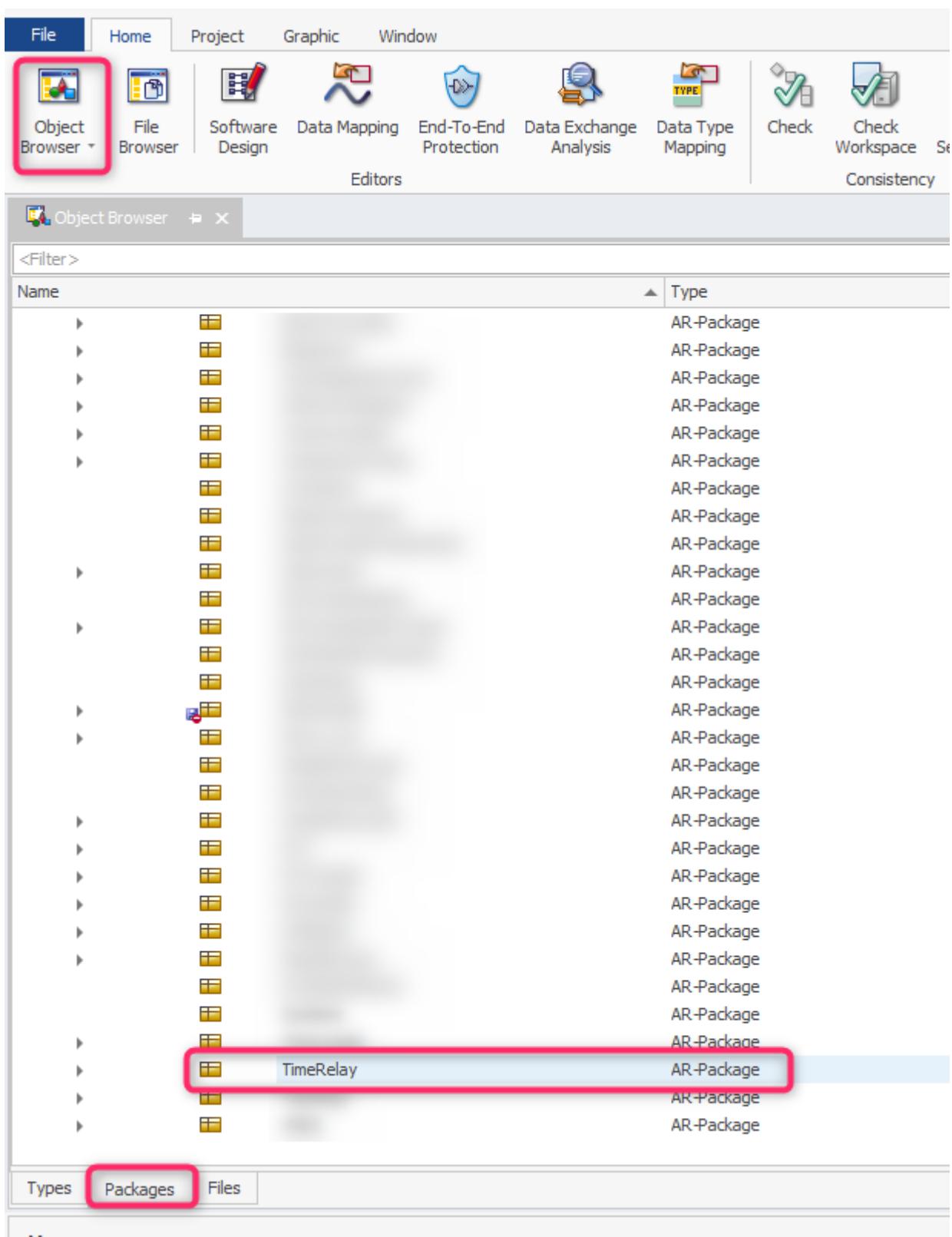
1. Have a project up and running (having the input file and every previous detail).

5.4 Steps

5.5 Steps to be performed in Davinci Developer Tool

5.5.1 Step 1: Go to package view of the project

Go to Object Browser → Package Tab.



5.5.2 Step 2: Create the Compu Method

Open the package whch will contain the new Compu Method → Then right click to show menu.

Screenshot of the Ford FNV4 FordOS DaVinci Tools interface showing the Object Browser.

The Object Browser is highlighted with a red box in the top ribbon bar.

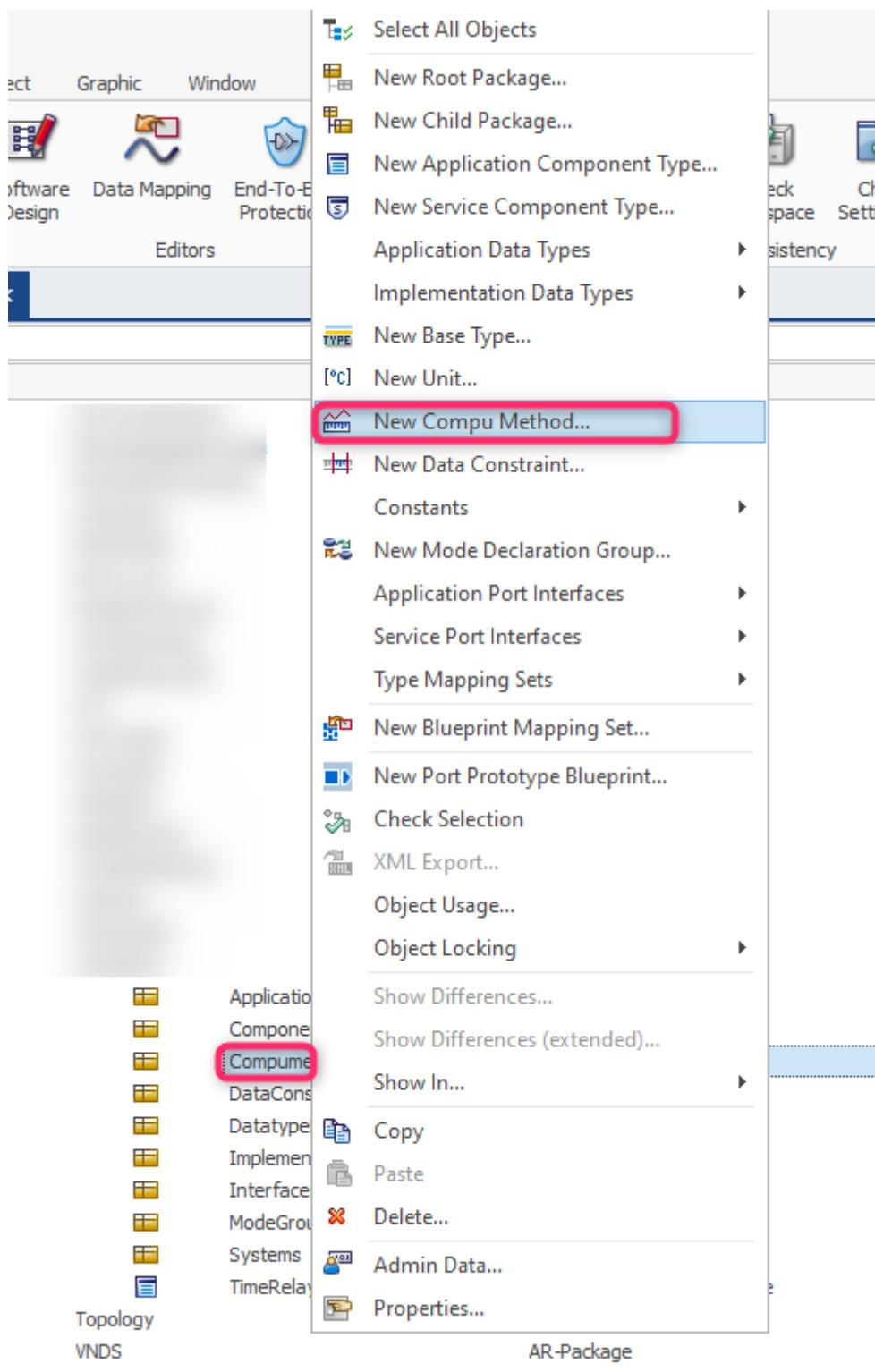
The Object Browser window displays a tree view of packages under the 'TimeRelay' package. The 'Compumethods' package is also highlighted with a red box.

Name	Type
TimeRelay	AR-Package
ApplicationDataTypes	AR-Package
ComponentTypes	AR-Package
Compumethods	AR-Package
DataConstraints	AR-Package
DatatypeMappingSets	AR-Package
ImplementationDataTypes	AR-Package
Interfaces	AR-Package
ModeGroups	AR-Package
Systems	AR-Package
TimeRelay_SWCType	Application Component Type
Topology	AR-Package
VNDS	AR-Package

The 'Packages' tab in the bottom navigation bar is highlighted with a red box.

Messages:

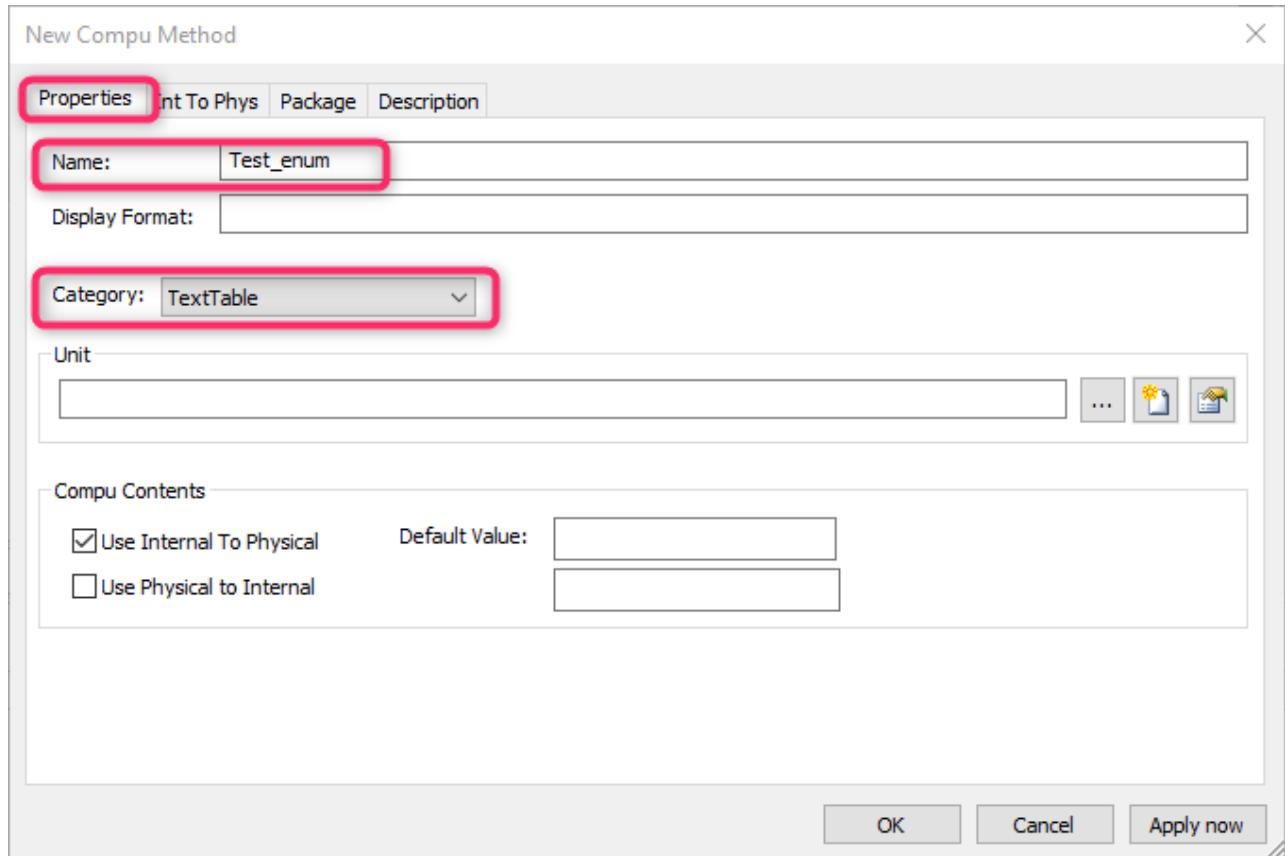
Select "New Compu Method"



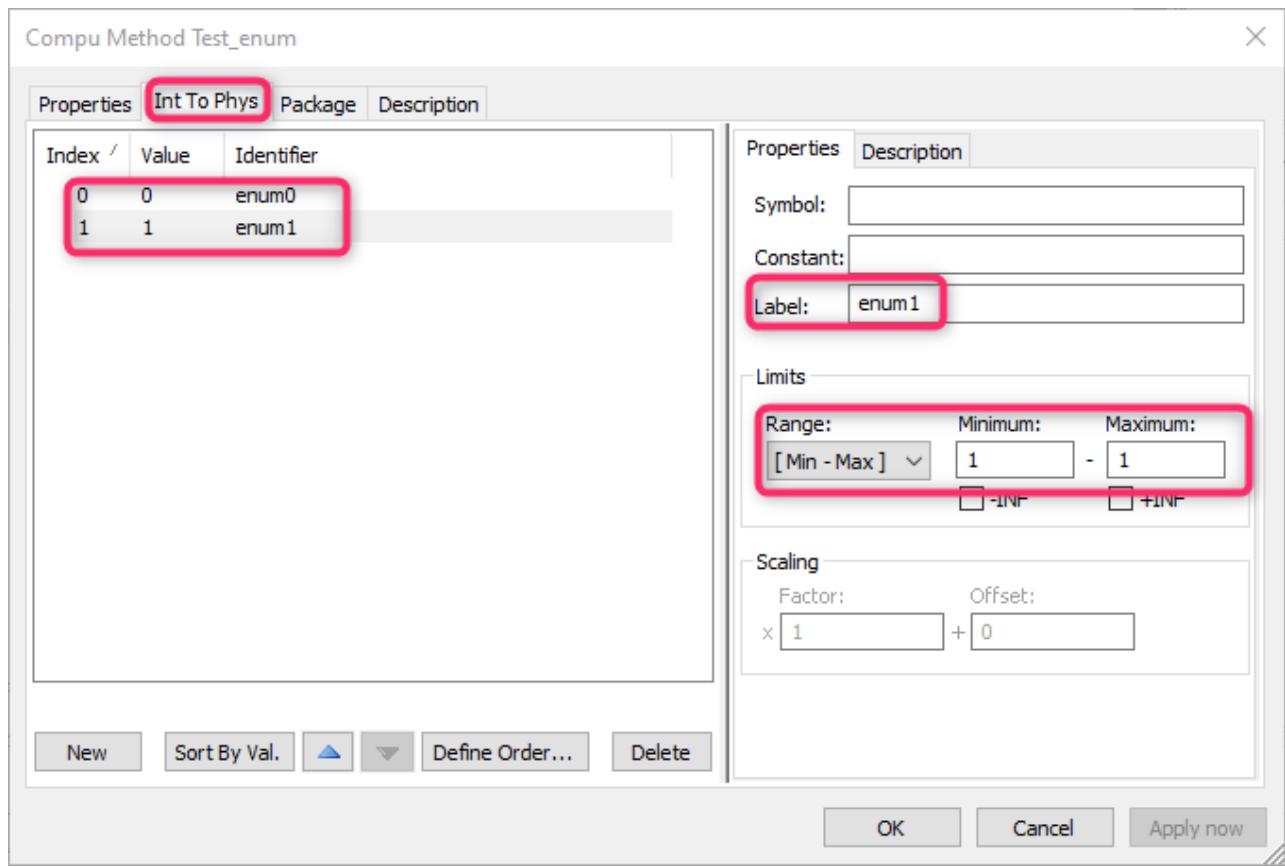
5.5.3 Step 3: Fill the "New Compu Method" Window

In the Properties Tab, fill the following:

- Create a new name for the Compu Method.
- Category: **TextTable**



In the Int To Phys Tab, fill the enum name and value, i.e.:



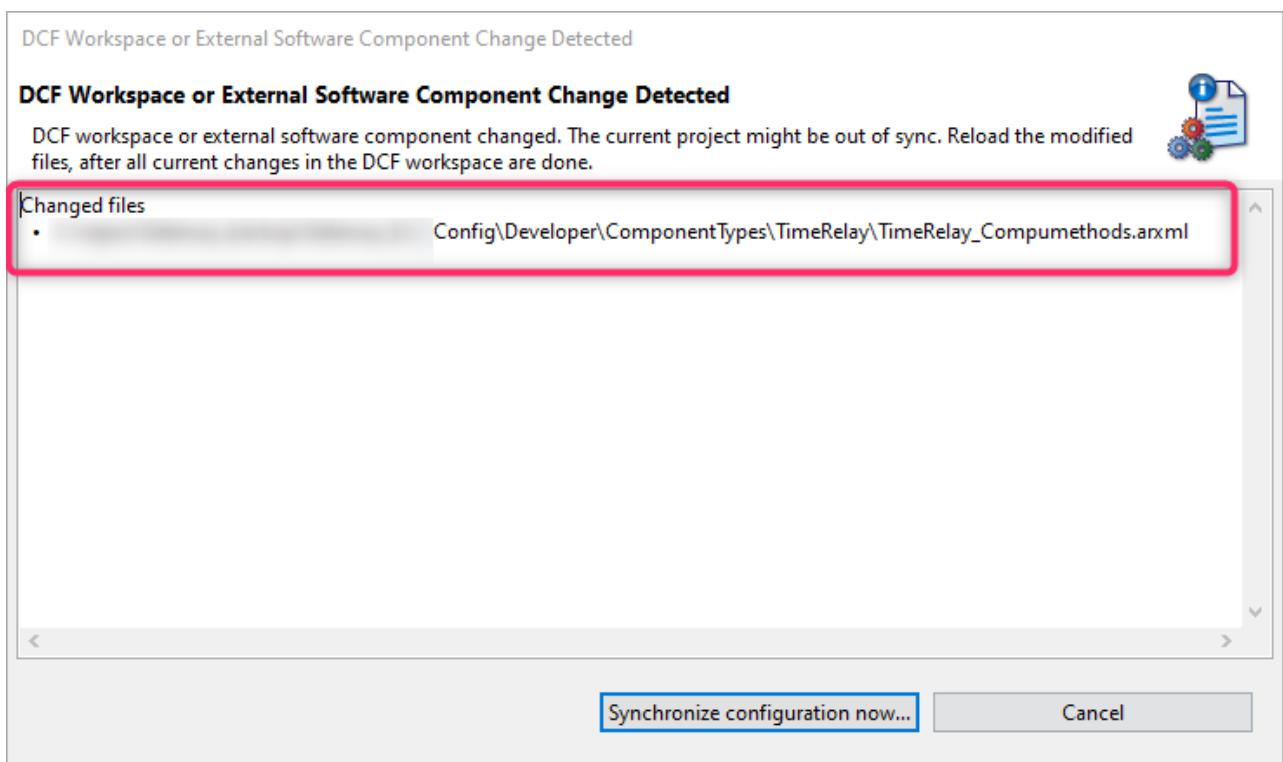
⚠ Note

Make sure the Range limits are set to the expected value in the enum or some errors could occur.

5.6 Steps to be performed in Davinci Configurator Tool

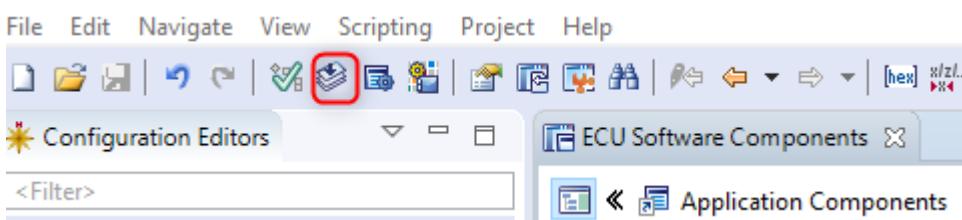
5.6.1 Step 4: Synchronize Project

After saving the Project in Davinci Developer, the Davinci Configurator will show a PopUp window requesting the Synchronization.

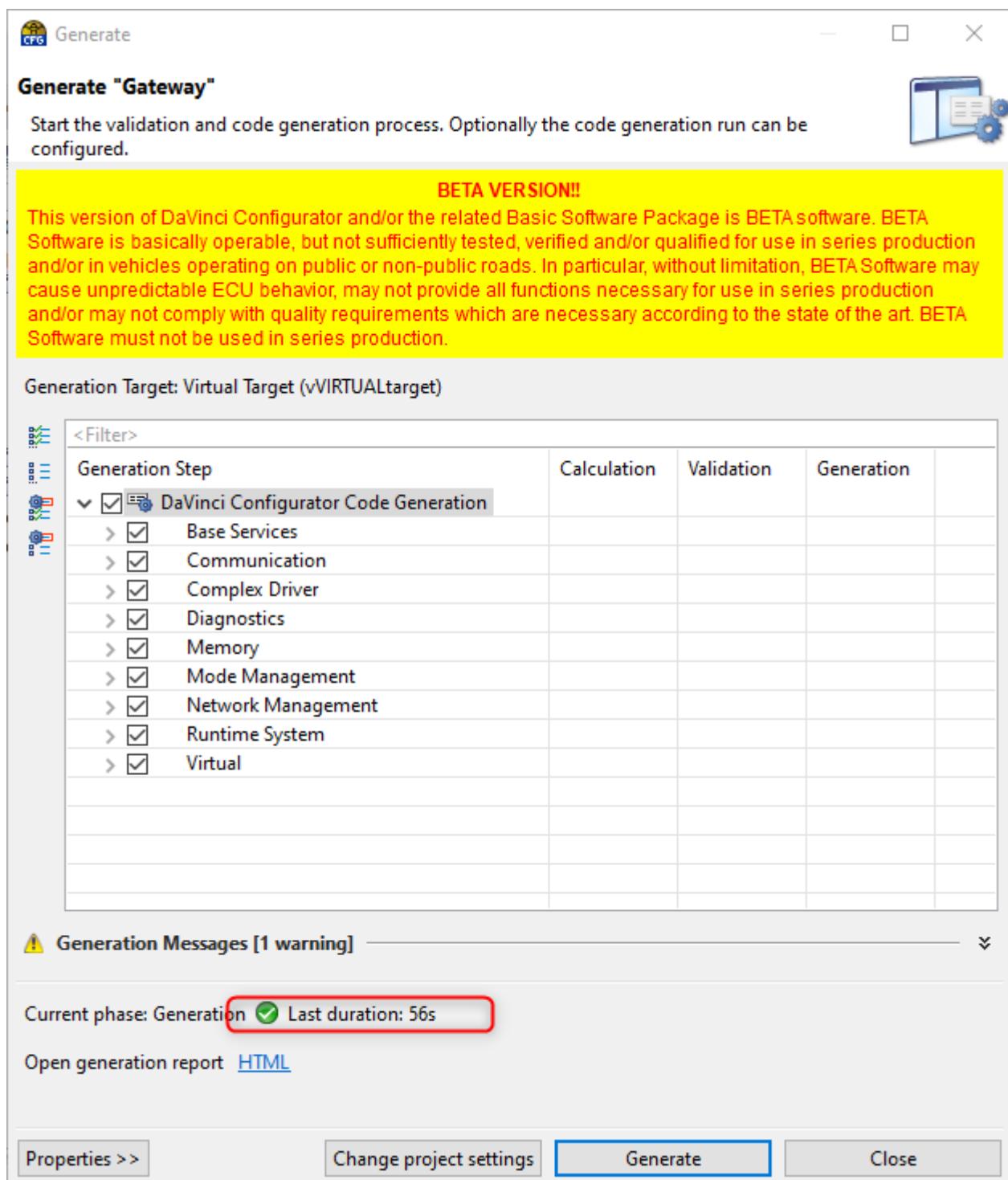


5.6.2 Step 5: Generate the Project

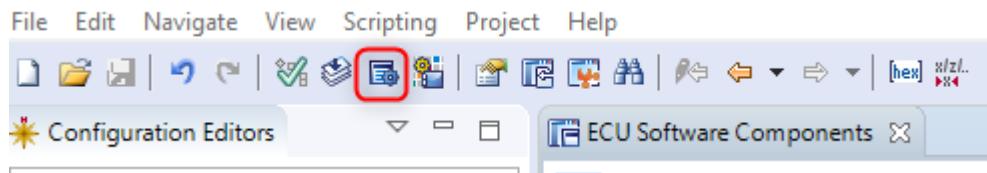
Go to Generate (Or click F9)



And Generate as usual



And finally generate the corresponding templates



5.7 Troubleshooting

6 4_Create New Implementation DataType

- Workflow Diagram(see page 54)
- Prerequisites(see page 55)
- Steps(see page 55)
- Steps to be performed in Davinci Developer Tool(see page 55)
 - Step 1: Go to package view of the project(see page 55)
 - Step 2: Create the Compu Method(see page 57)
 - Step 3: Fill the "New Implementation Record" Window(see page 59)
- Steps to be performed in Davinci Configurator Tool(see page 61)
 - Step 4: Synchronize Project(see page 61)
 - Step 5: Generate the Project(see page 61)
- Troubleshooting(see page 63)

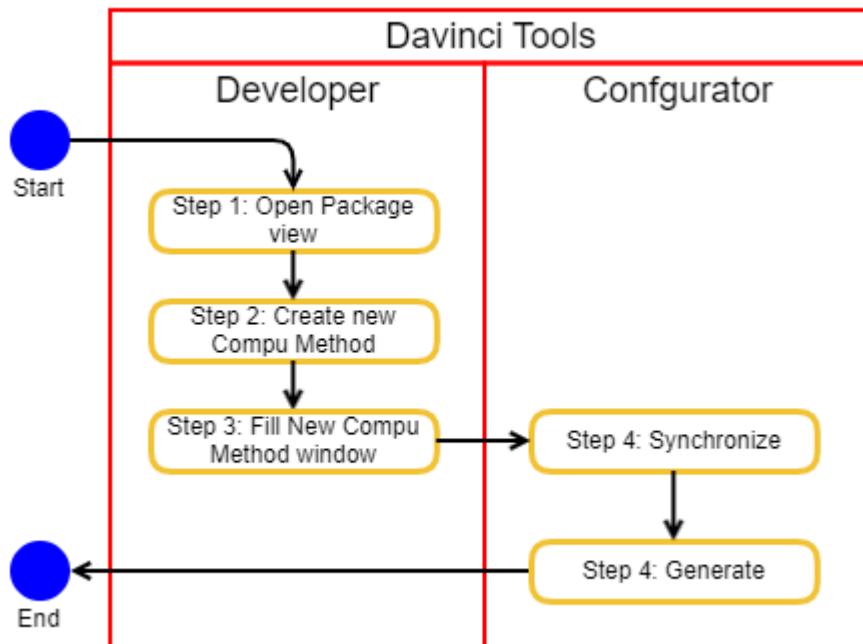
Context

AUTOSAR Def

Implementation Data Types is a very generic container and can define almost any data type.

With Implementation DataTypes is possible to create TypeDef structure like in C language.

6.1 Workflow Diagram



6.2 Prerequisites

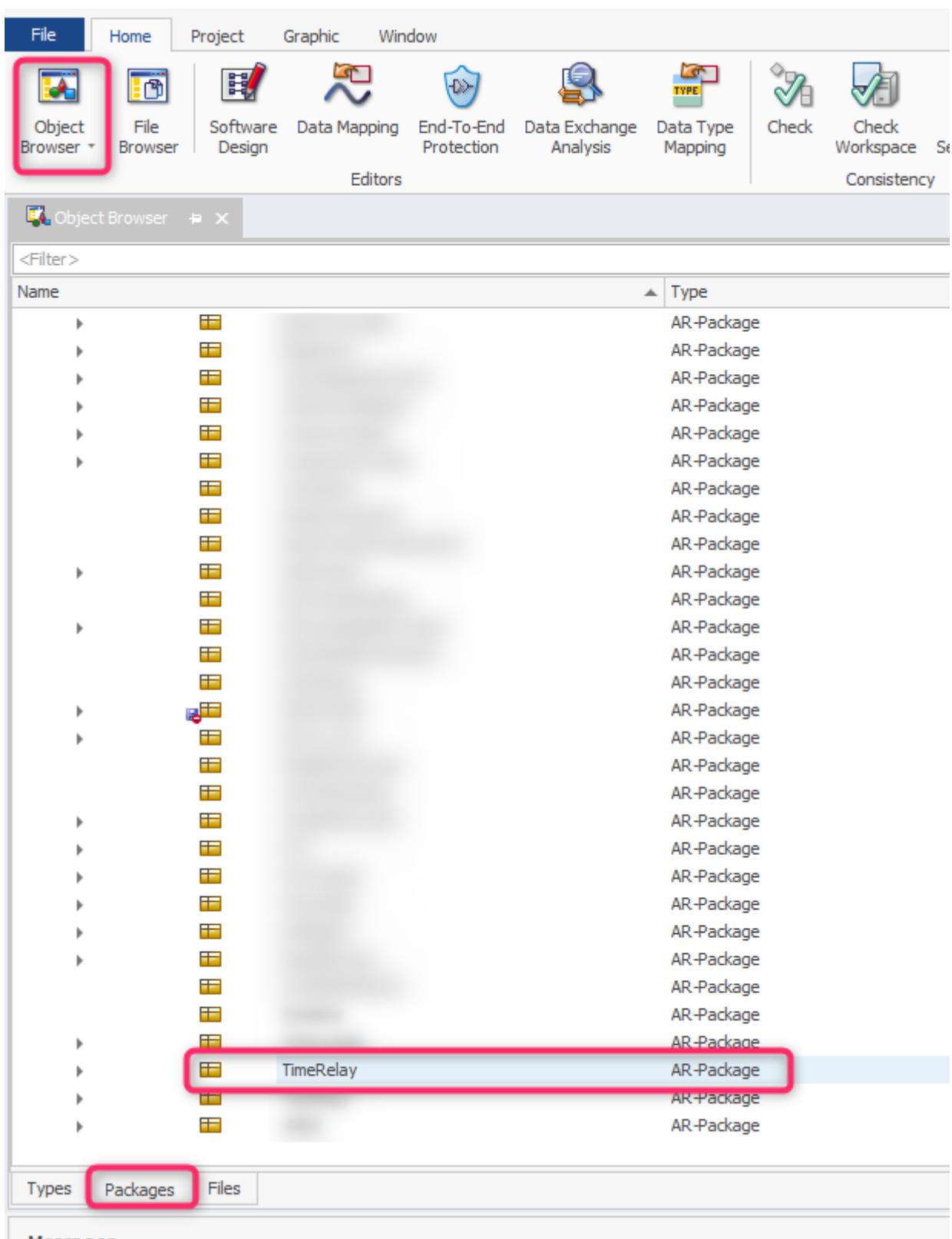
1. Have a project up and running (having the input file and every previous detail).

6.3 Steps

6.4 Steps to be performed in Davinci Developer Tool

6.4.1 Step 1: Go to package view of the project

Go to Object Browser → Package Tab.



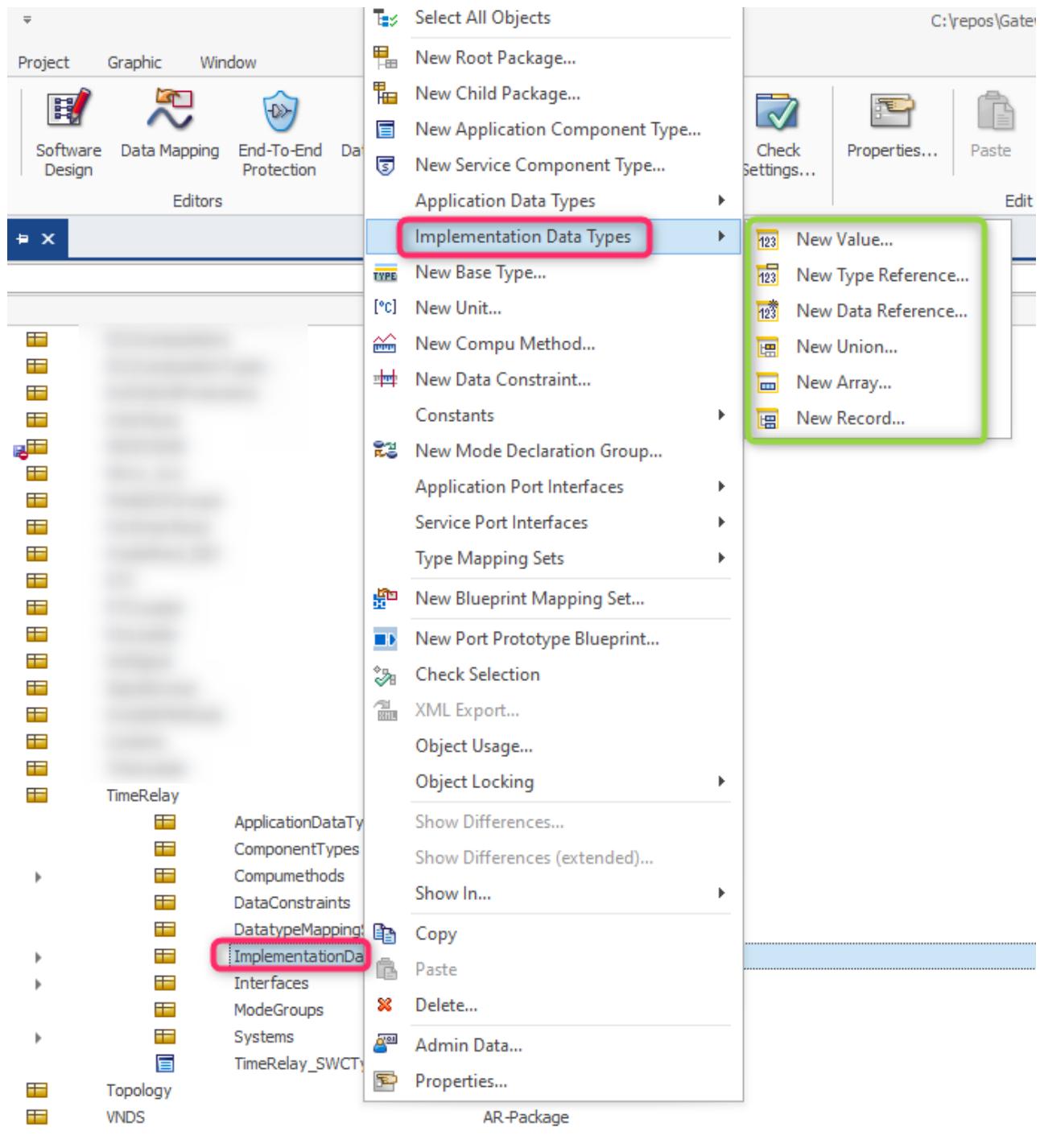
6.4.2 Step 2: Create the Compu Method

Open the package which will contain the new Implementation DataType → Then right click to show menu.

The screenshot shows the DaVinci Tools interface with the Object Browser tab selected. The main pane displays the structure of the TimeRelay package. The ImplementationDataTypes package is highlighted with a red box. The Packages tab is selected at the bottom of the browser window.

Name	Type
TimeRelay	AR-Package
ApplicationDataTypes	AR-Package
ComponentTypes	AR-Package
Compumethods	AR-Package
DataConstraints	AR-Package
DatatypeMappingSets	AR-Package
ImplementationDataTypes	AR-Package
Interfaces	AR-Package
ModeGroups	AR-Package
Systems	AR-Package
TimeRelay_SWCType	Application Component Type

Select "New Implementation DataType" and Select the Type of Implementation DataType



Note

- Value → Simple dataType like uint8, uintn16,...

- Union → special data type available in C that allows to store different data types in the same memory location.
- Type Reference → special data type that reference to another Data Type.
- Data Reference → similar to a pointer, save the reference to an specific variable.
- Array → collection of similar data items stored at contiguous memory locations and elements can be accessed randomly using indices of an array.
- Record → Collection of different data types stored at contiguous memory locations.

⚠ Advice

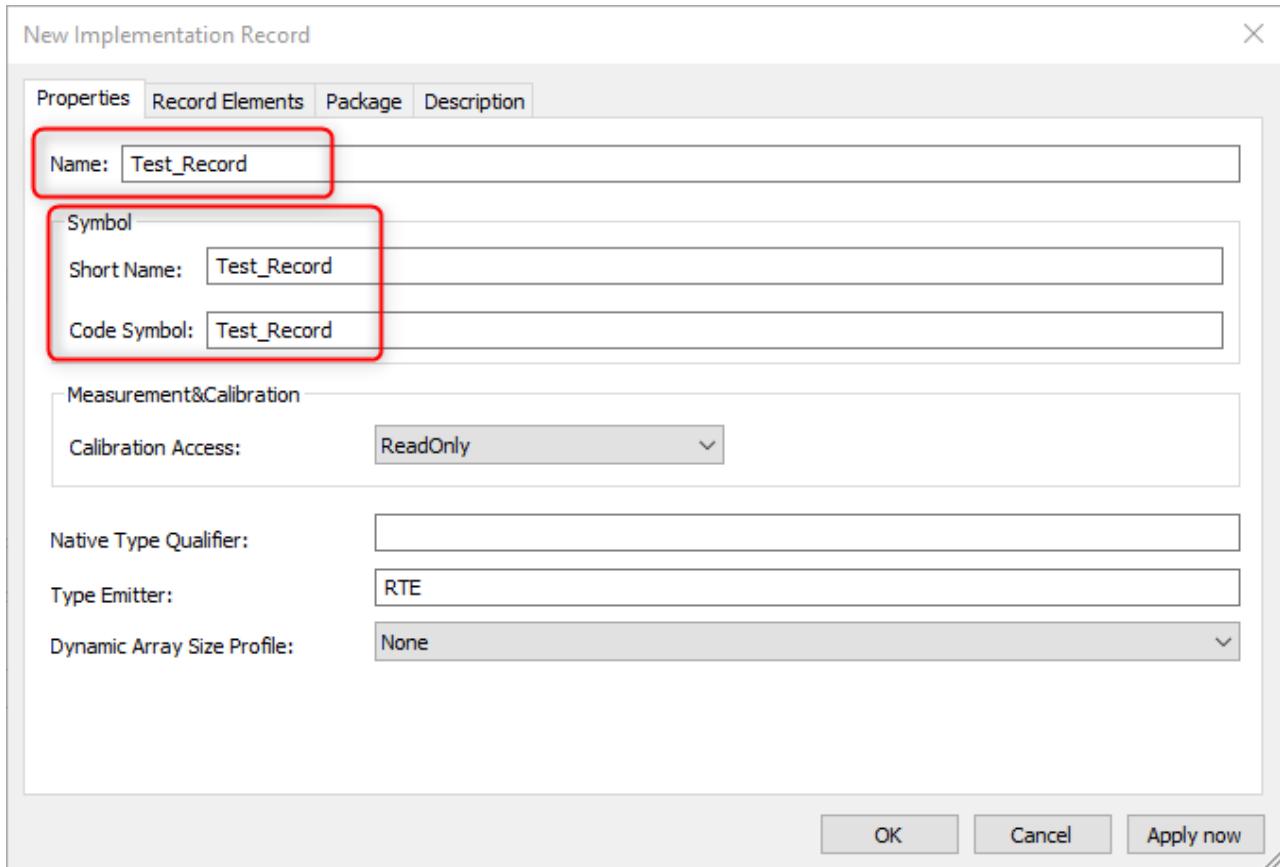
To reduce the possibilities of generation errors, it's recommended to use as first option Type Reference.

For this example we will use Record as it's the most one found in the project.

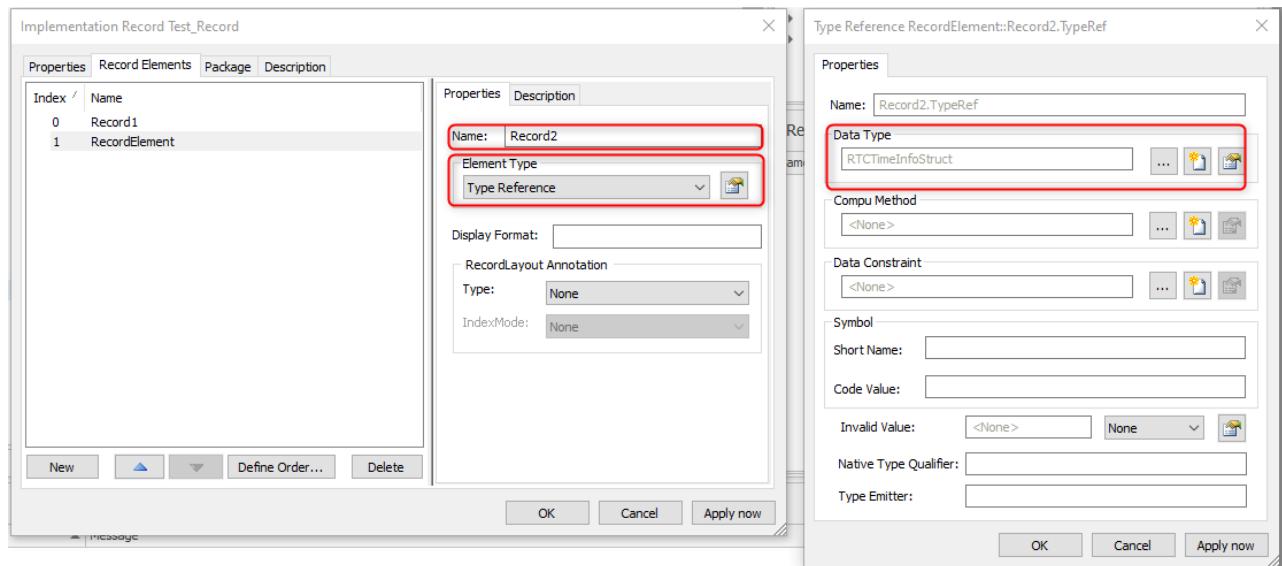
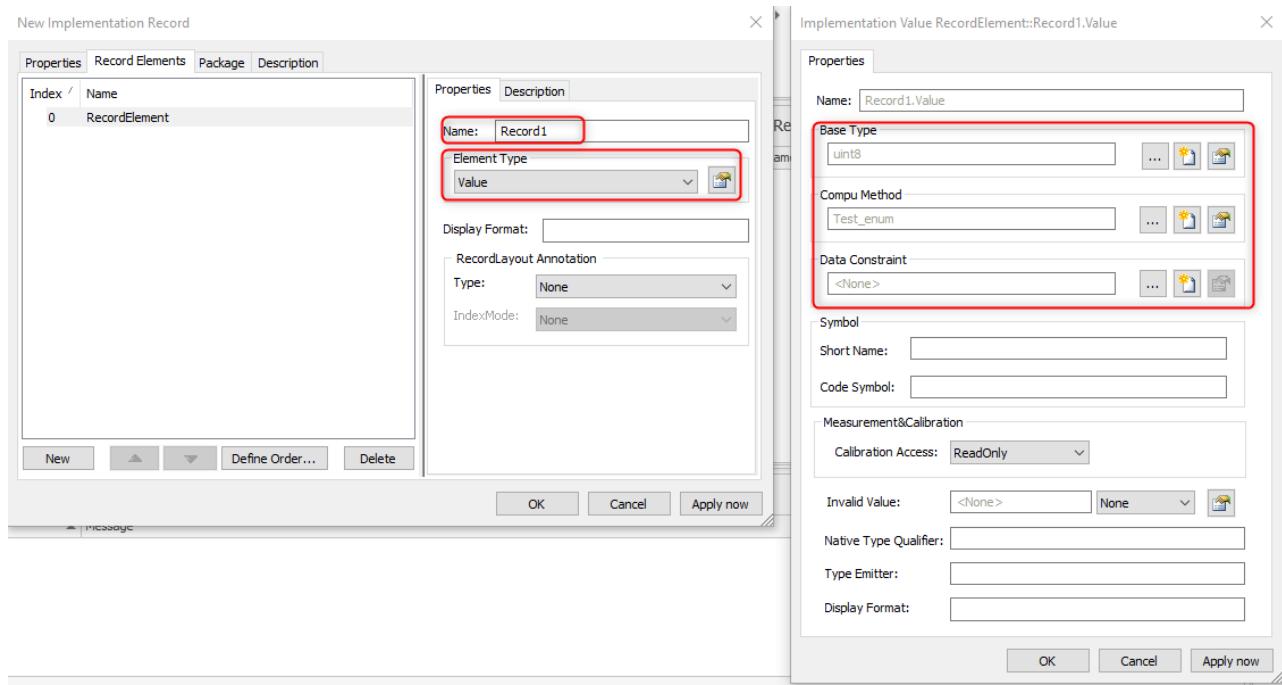
6.4.3 Step 3: Fill the "New Implementation Record" Window

In the Properties Tab, fill the following:

- Create a new Name for the Implementation Record.
- Short Name: Same as Name.
- Code Symbol: Same as Name.



In the Record Elements Tab, fill the value according to the use case i.e.:



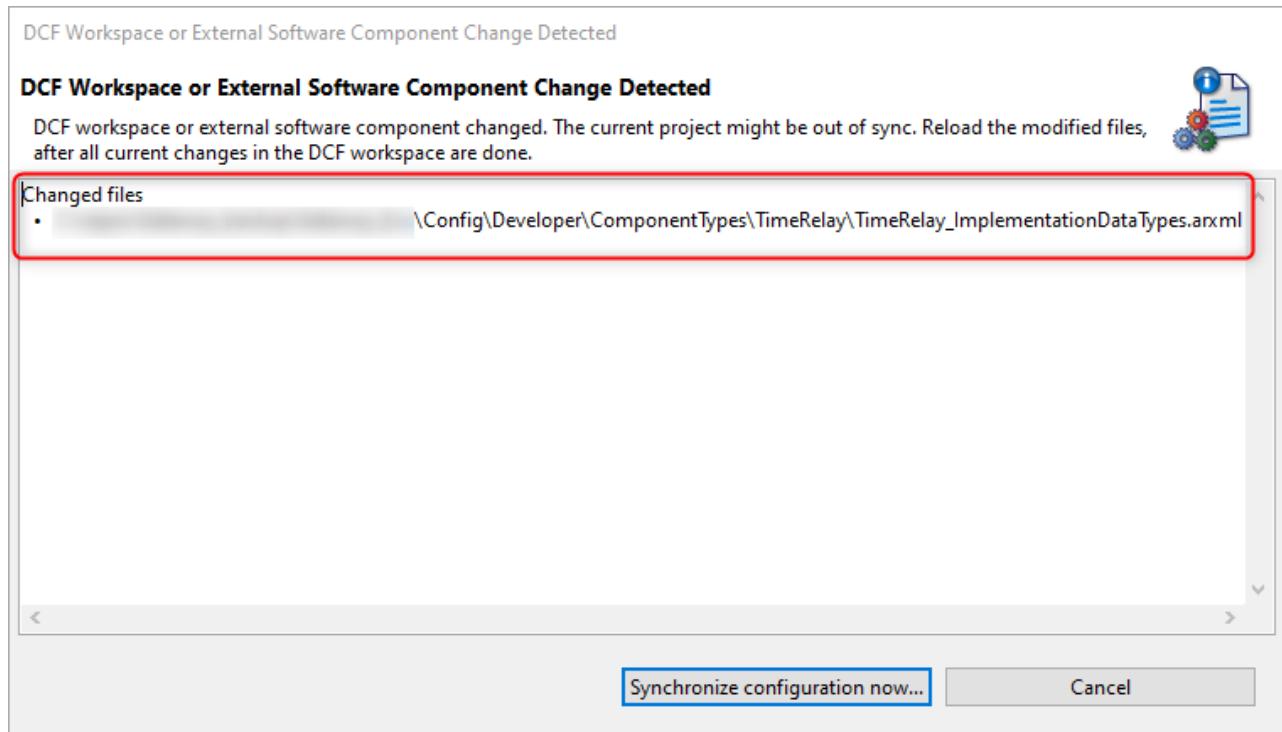
⚠ Note

Please consider that if you want to have a record within another record, the nested record can be created using Element Type: **Record**, or if the nested record is already created please use Element Type: **Type Reference**.

6.5 Steps to be performed in Davinci Configurator Tool

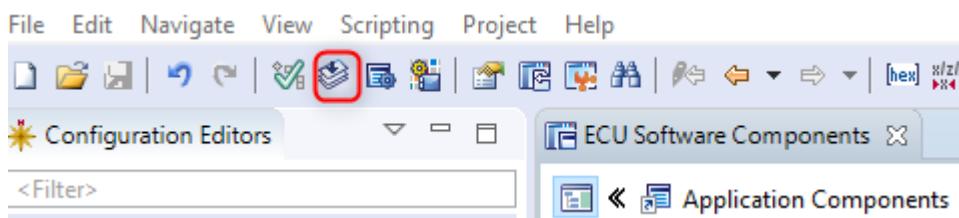
6.5.1 Step 4: Synchronize Project

After saving the Project in Davinci Developer, the Davinci Configuratoe will show a PopUp window requesting the Synchronization.

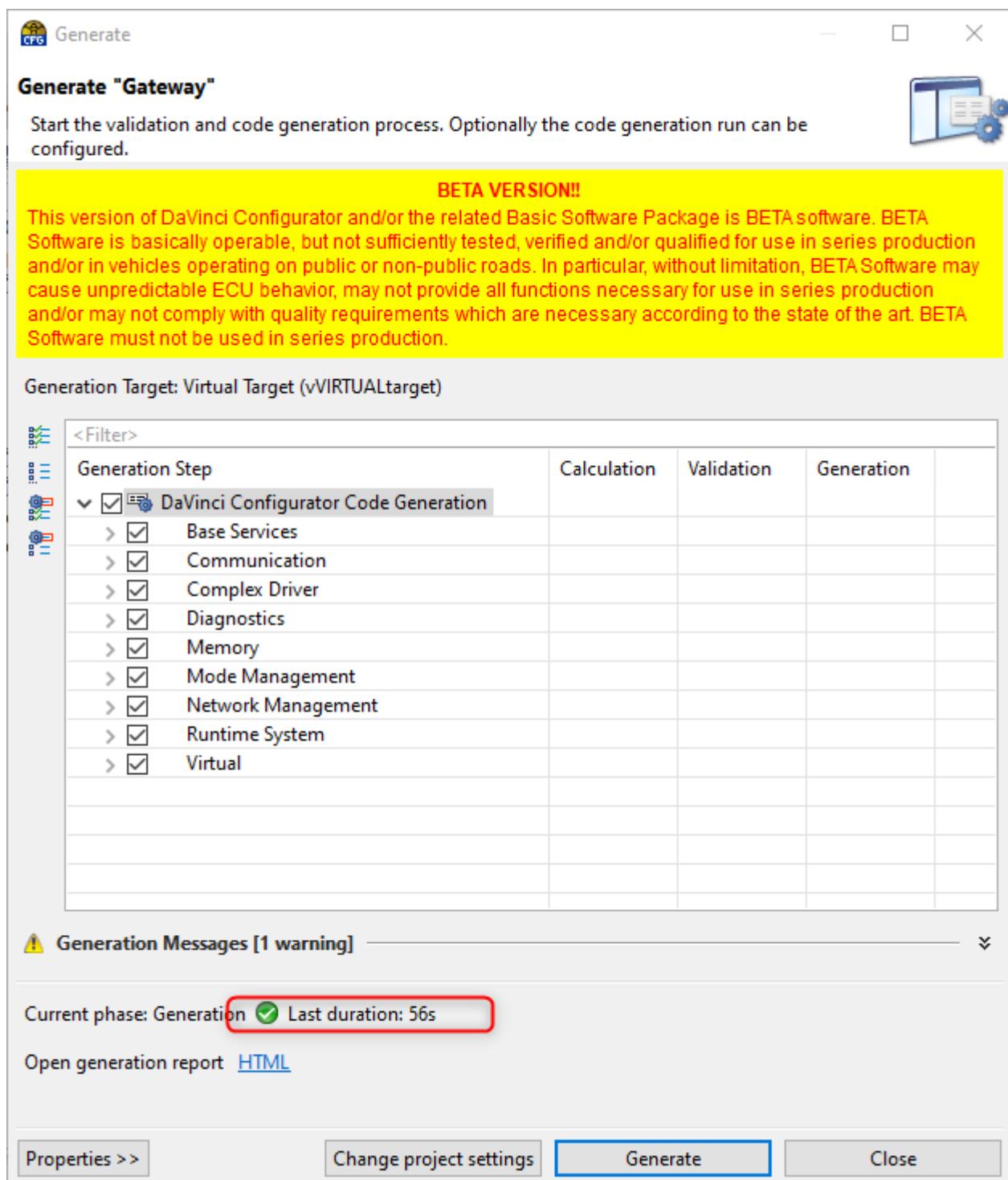


6.5.2 Step 5: Generate the Project

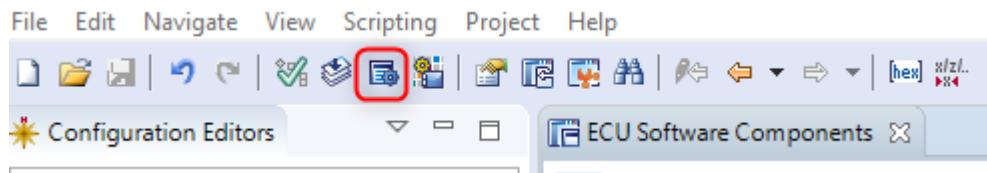
Go to Generate (Or click F9)



And Generate as usual



And finally generate the corresponding templates



6.6 Troubleshooting

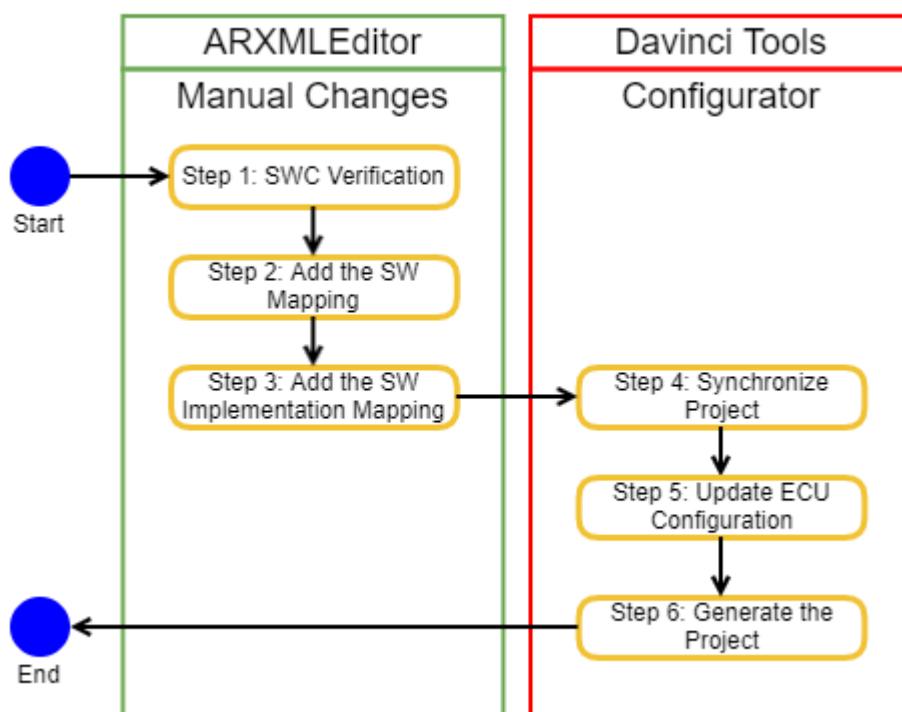
7 5_ Add New SWC to the Deployment List

- Use Case(see page 64)
- Workflow Diagram(see page 64)
- Prerequisites(see page 65)
- Steps(see page 65)
- Steps to be Performed in ARXML Editor(see page 65)
 - Step 1: SWC Verification(see page 65)
 - Step 2: Add the SW Mapping(see page 65)
 - Step 3: Add the SW Implementation Mapping(see page 67)
- Steps to be performed in Davinci Configurator Tool(see page 69)
 - Step 4: Synchronize Project(see page 69)
 - Step 5: Update ECU Configuration ARXML(see page 69)
 - Step 6: Generate the Project(see page 70)
- Troubleshooting(see page 72)

7.1 Use Case

- Adding a SWC to the Deployment List in the ECU extract could solve some RTE Hooks issue.
 - This is not a final solution, but in practice have proved to solve sometimes the RTE Hook Generation issues.

7.2 Workflow Diagram



7.3 Prerequisites

1. Having the SWC already Created with all the necessary elements (Runnable, Task, Ports, etc).
2. Have a ARXML Editor (or XML with Schema Validator), like AUTOSAR XML Editor from Vector Informatik.

7.4 Steps

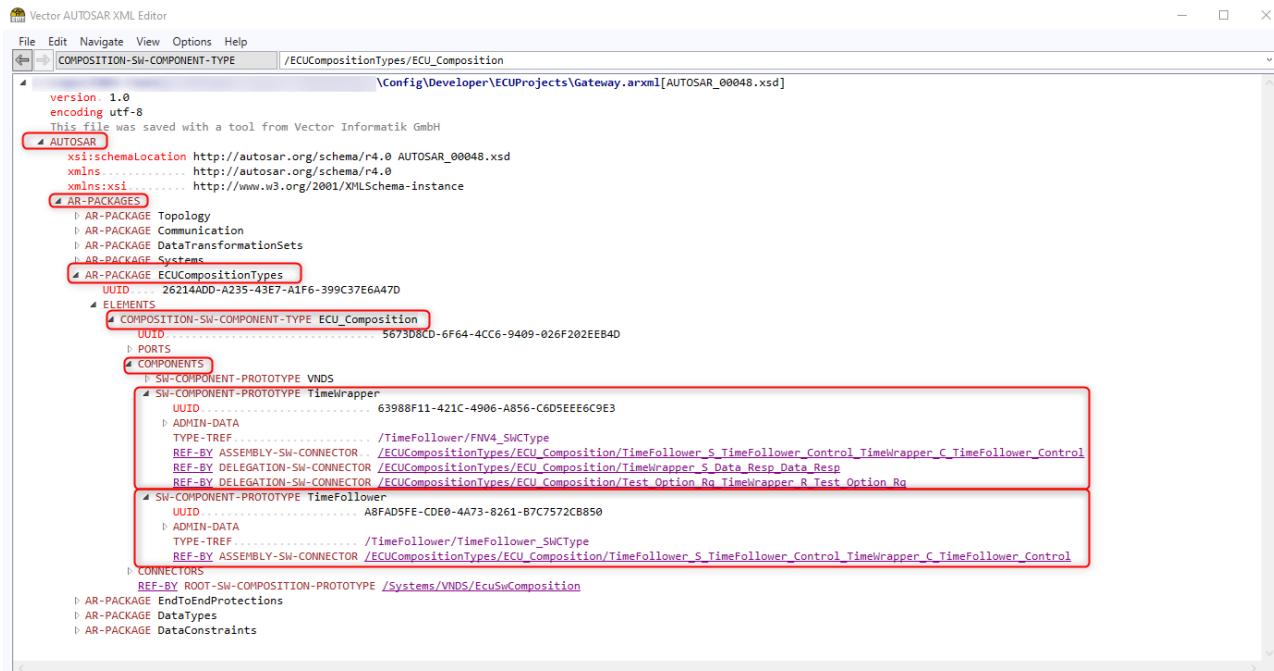
7.5 Steps to be Performed in ARXML Editor

7.5.1 Step 1: SWC Verification

First we need to verify the SWC to be added to the Deployment Listi properly added.

For that using the ARXML Editor follow the path:

AUTOSAR → AR-PACKAGES → ECUCompositionTypes → ECU_Composition → (expand to veridfy the components added into the project)



7.5.2 Step 2: Add the SW Mapping

For that using the ARXML Editor follow the path:

AUTOSAR → AR-PACKAGES → Systems → <Project Name> (in this example VNDs) → MAPPINGS → SYSTEM MAPPING → SW-MAPPING → SWC-TO-ECU-MAPPING → COMPONENT-IREFS → COMPONENT-IREF → (Here you will add a new IREF for your SWC)

You will add the following information

1. Path to the ECU Composition of the Project
2. Path to the SWC like Step 1
3. Path to the SWC like Step 1

ARXML code to be added

```
<COMPONENT-IREF>
  <CONTEXT-COMPOSITION-REF DEST="ROOT-SW-COMPOSITION-PROTOTYPE">/Systems/GWM/
EcuSwComposition</CONTEXT-COMPOSITION-REF>
  <CONTEXT-COMPONENT-REF DEST="SW-COMPONENT-PROTOTYPE">/ECUCompositionTypes/
ECU_Composition/SWC_NAME</CONTEXT-COMPONENT-REF>
  <TARGET-COMPONENT-REF DEST="SW-COMPONENT-PROTOTYPE">/ECUCompositionTypes/
ECU_Composition/SWC_NAME</TARGET-COMPONENT-REF>
</COMPONENT-IREF>
```

```

version: 1.0
encoding: utf-8
This file was saved with a tool from Vector Informatik GmbH
AUTOSAR
  xsi:schemaLocation: http://autosar.org/schema/r4.0 AUTOSAR_00048.xsd
  xmlns: http://autosar.org/schema/r4.0
  xmlns:xsi: http://www.w3.org/2001/XMLSchema-instance
  AR-PACKAGES
    AR-PACKAGE Topology
    AR-PACKAGE Communication
    AR-PACKAGE DataTransformationSets
    AR-PACKAGE Systems
      UUID: 6B031DB9-82A9-4DF9-AF96-72D5B5A47D50
      ELEMENTS
        SYSTEM.. VNDs
          UUID: CA58116E-3443-427B-B580-7D4B9BF98CDD
          CATEGORY: ECU_EXTRACT
        FIBEX-ELEMENTS
          MAPPINGS
            SYSTEM-MAPPING Mapping
              UUID: 462D4855-48B1-4F5E-B572-DC454E5329A7
            DATA-MAPPINGS
            SW-IMPL-MAPPINGS
              SW-MAPPINGS
                SWC-TO-ECU-MAPPING Mapping_VNDS
                  UUID: 7C56B43D-E98A-4F0F-99CC-1022FC3F0BE1
            COMPONENT-IREFS
              COMPONENT-IREF
              COMPONENT-IREF
              COMPONENT-IREF
              COMPONENT-IREF
              COMPONENT-IREF
              COMPONENT-IREF
              COMPONENT-IREF
                CONTEXT-COMPOSITION-REF: /Systems/GWM/EcuSwComposition
                CONTEXT-COMPONENT-REF: /ECUCompositionTypes/ECU_Composition/TimeFollower
                TARGET-COMPONENT-REF: /ECUCompositionTypes/ECU_Composition/TimeFollower
              COMPONENT-IREF
                CONTEXT-COMPOSITION-REF: /Systems/GWM/EcuSwComposition
                CONTEXT-COMPONENT-REF: /ECUCompositionTypes/ECU_Composition/TimeWrapper
                TARGET-COMPONENT-REF: /ECUCompositionTypes/ECU_Composition/TimeWrapper
              ECU-INSTANCE-REF: /Topology/HardwareComponents/GWM
            ROOT-SOFTWARE-COMPOSITIONS

```

7.5.3 Step 3: Add the SW Implemantion Mapping

For that using the ARXML Editor follow the path:

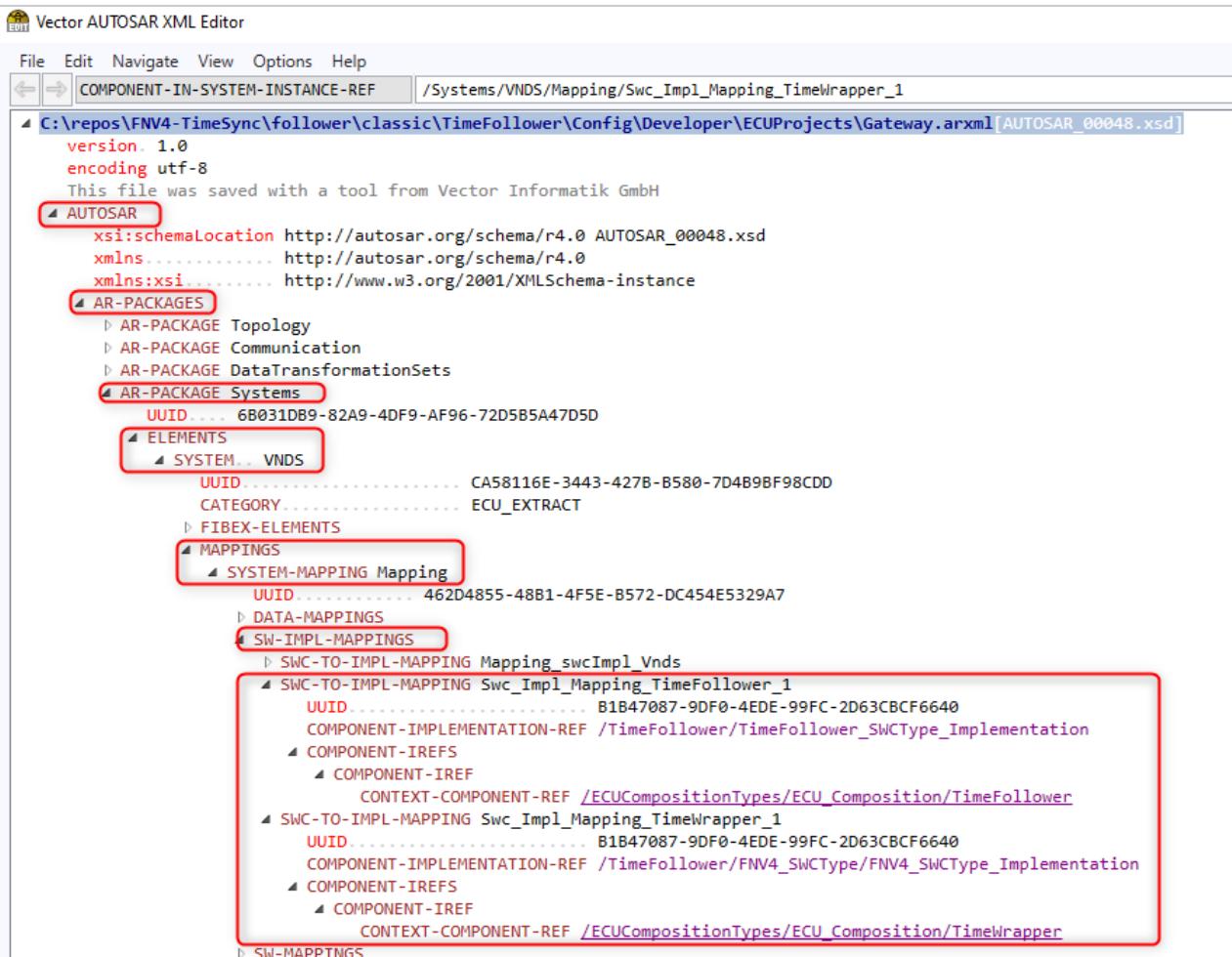
AUTOSAR → AR-PACKAGES → Systems → <Project Name> (in this example VNDs) → MAPPINGS → SYSTEM MAPPING → SW-IMPL-MAPPING → SWC-TO-IMPL-MAPPINGS → SWC-TO-IMPL-MAPPING → (Here you will add a new MAPPING for your SWC)

You will add the following information

1. Name the mapping (Swc_Impl_Mapping_SWC_Name_1)
2. Path to the SWC Implemantation
3. Path to the SWC like Step 1

ARXML code to be added

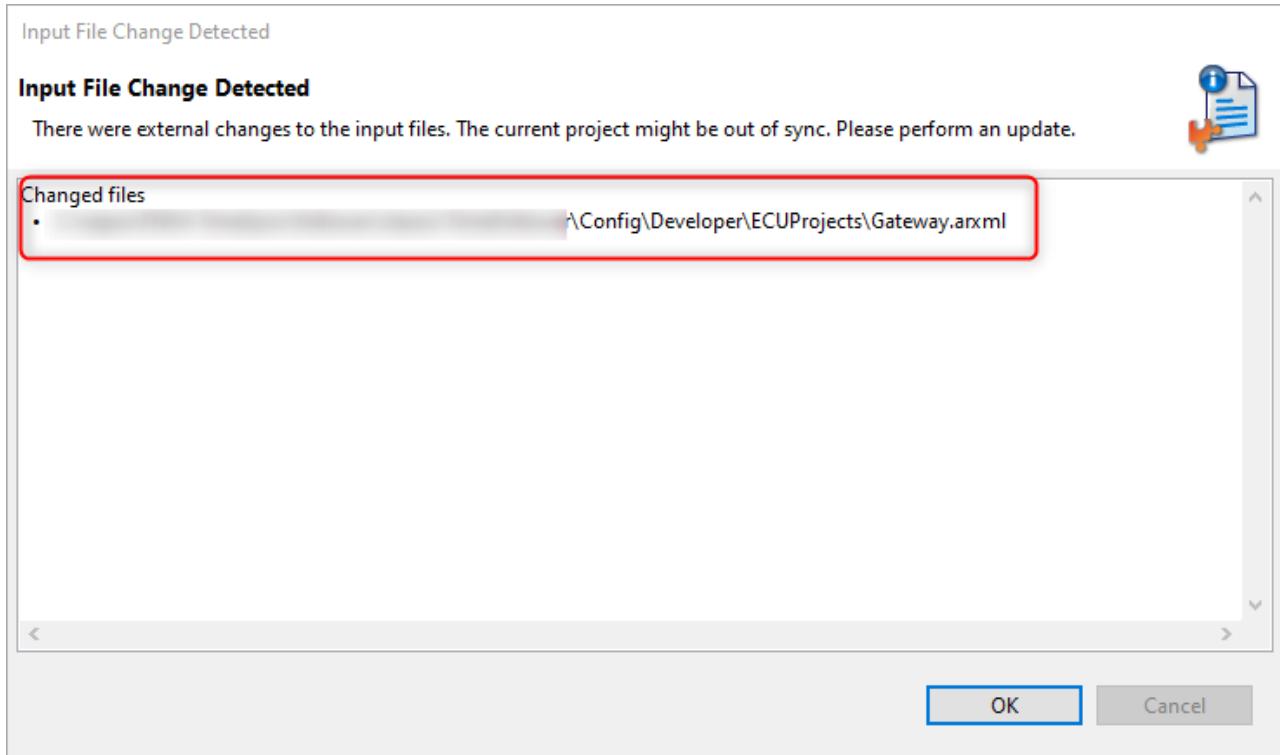
```
<SWC-TO-IMPL-MAPPING UUID="B1B47087-9DF0-4EDE-99FC-2D63CBCF6640">
  <SHORT-NAME>Swc_Impl_Mapping_SWC_Name_1</SHORT-NAME>
  <COMPONENT-IMPLEMENTATION-REF DEST="SWC-IMPLEMENTATION">/SWC_Name/
  SWC_Name_SWCType_Implementation</COMPONENT-IMPLEMENTATION-REF>
  <COMPONENT-IREFS>
    <COMPONENT-IREF>
      <CONTEXT-COMPONENT-REF DEST="SW-COMPONENT-PROTOTYPE">/ECUCompositionTypes/
      ECU_Composition/SWC_Name</CONTEXT-COMPONENT-REF>
    </COMPONENT-IREF>
  </COMPONENT-IREFS>
</SWC-TO-IMPL-MAPPING>
```



7.6 Steps to be performed in Davinci Configurator Tool

7.6.1 Step 4: Synchronize Project

After saving the Project in Davinci Developer, the Davinci Configurator will show a PopUp window requesting the Synchronization.

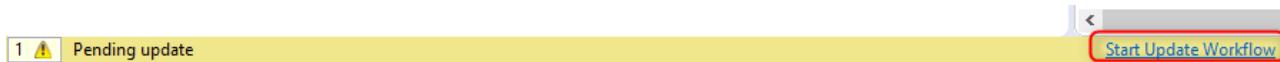


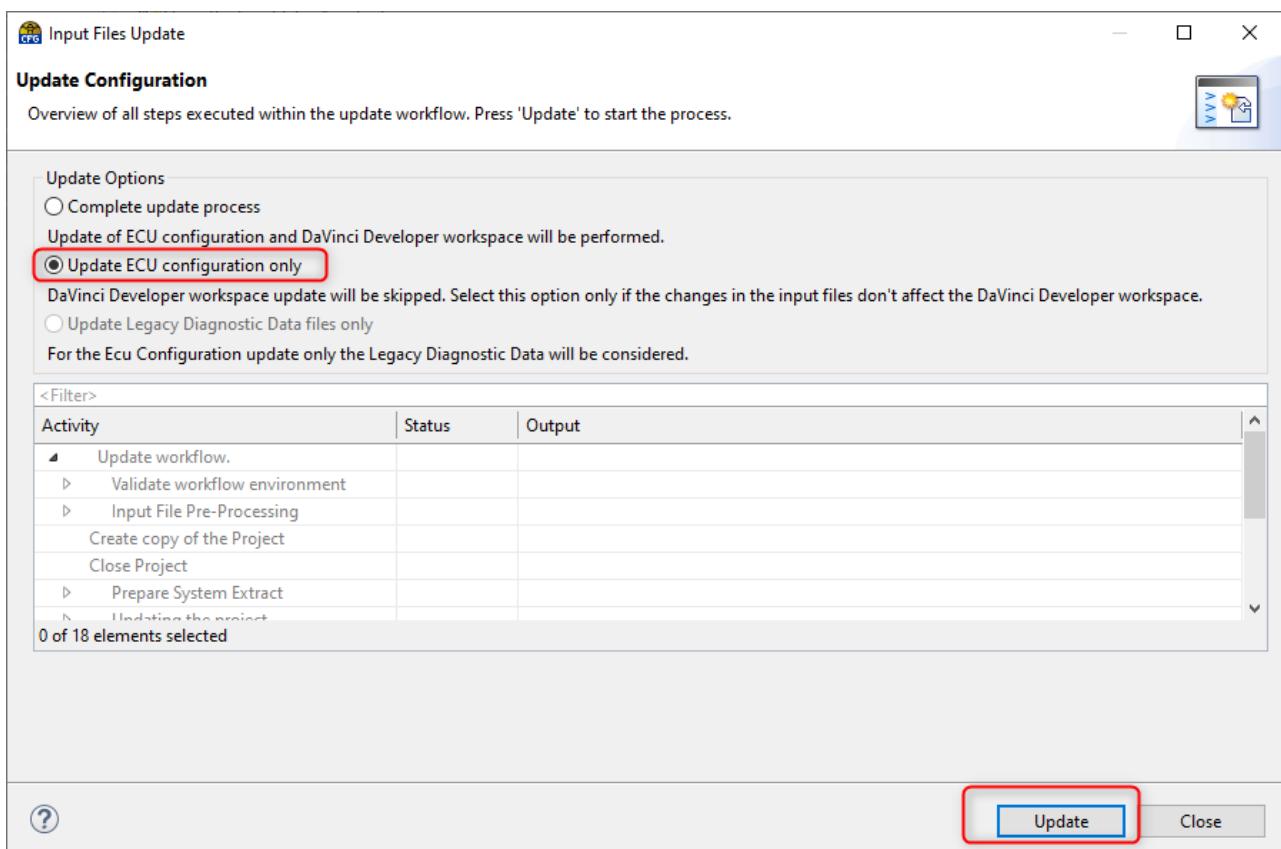
7.6.2 Step 5: Update ECU Configuration ARXML

⚠ Sync Needed

Please consider that after modifying (and saving) the ARXML file, an Update shall be needed, please follow the recommended Updating process by clicking the Yellow ribbon in the Vector Davinci Configurator Tool.

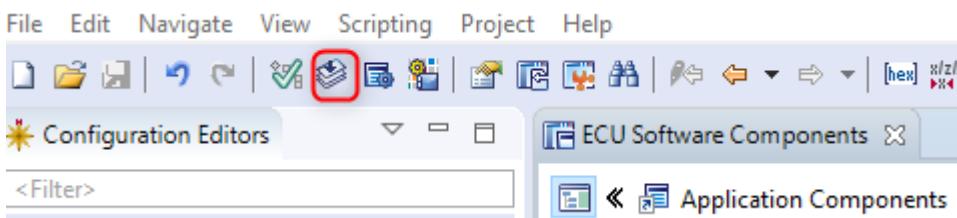
Please make use to follow the Update show in the Following Images:



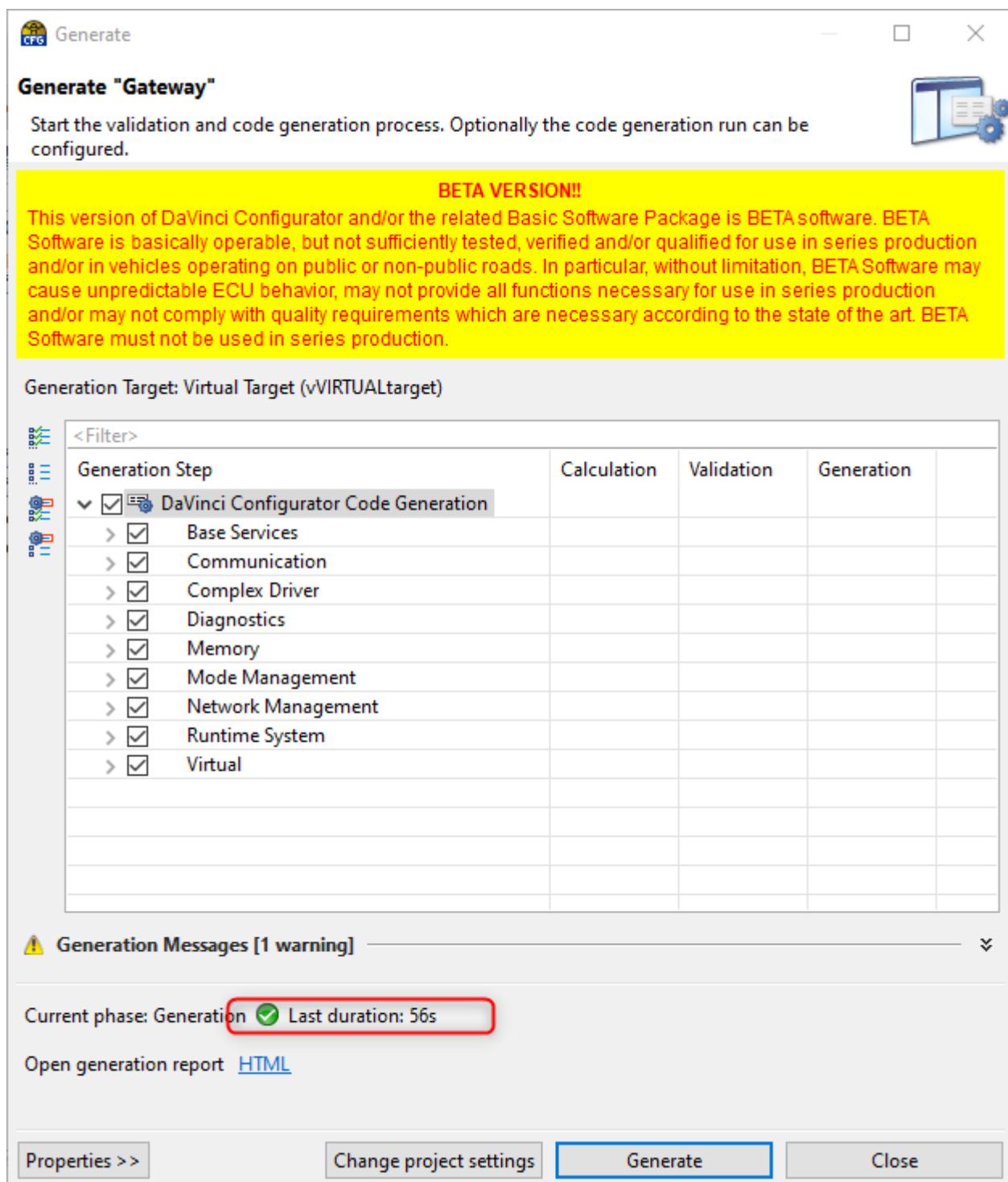


7.6.3 Step 6: Generate the Project

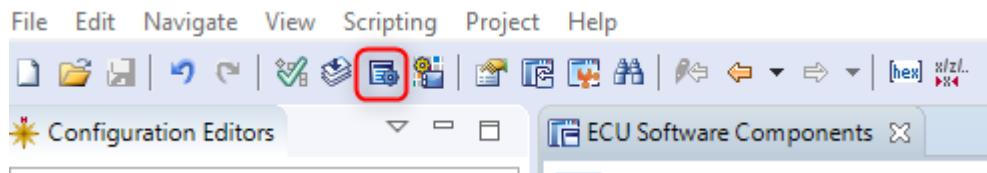
Go to Generate (Or click F9)



And Generate as usual



And finally generate the corresponding templates

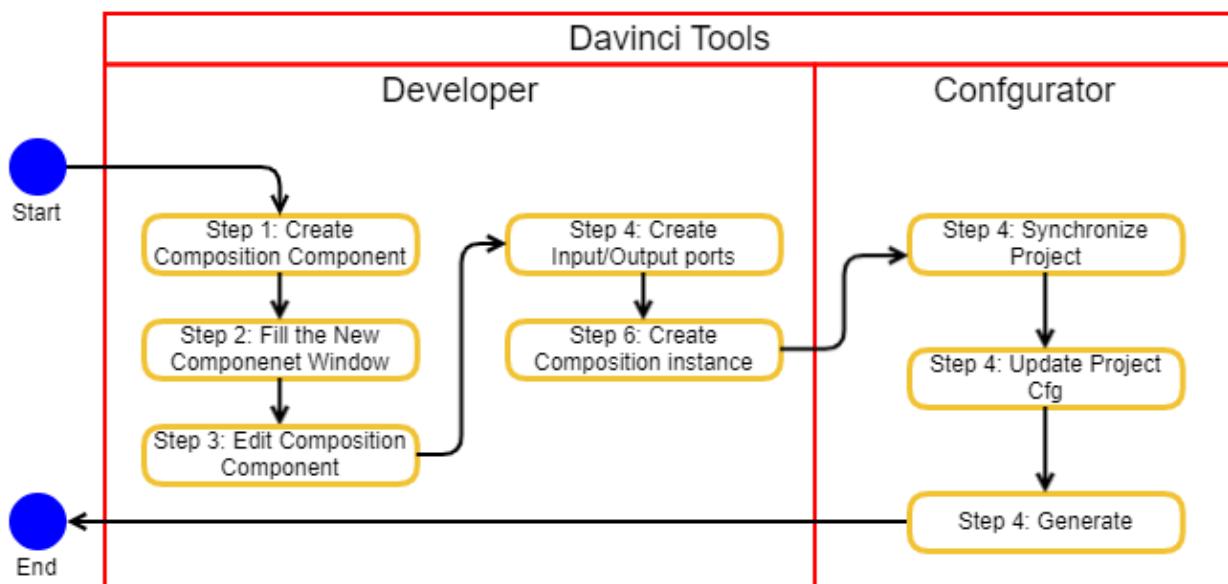


7.7 Troubleshooting

8 6_Create a New Composition Component

- Workflow Diagram(see page 73)
- Prerequisites(see page 73)
- Steps(see page 74)
- Steps to be performed in Davinci Developer Tool(see page 74)
 - Step 1: Create Composition Component Type(see page 74)
 - Step 2: Fill the "New Component Type"(see page 75)
 - Step 3: Edit Composition Component(see page 76)
 - Step 4: Create Input/Outut Ports(see page 79)
 - Step 5: Create Composition instance in the ECU Composition.(see page 79)
- Steps to be performed in Davinci Configurator Tool(see page 82)
 - Step 6: Synchronize Project(see page 82)
 - Step 7: Update ARXML Files in Davinci Cfg(see page 84)
 - Step 8: Generate the Project(see page 84)
- Troubleshooting(see page 86)

8.1 Workflow Diagram



8.2 Prerequisites

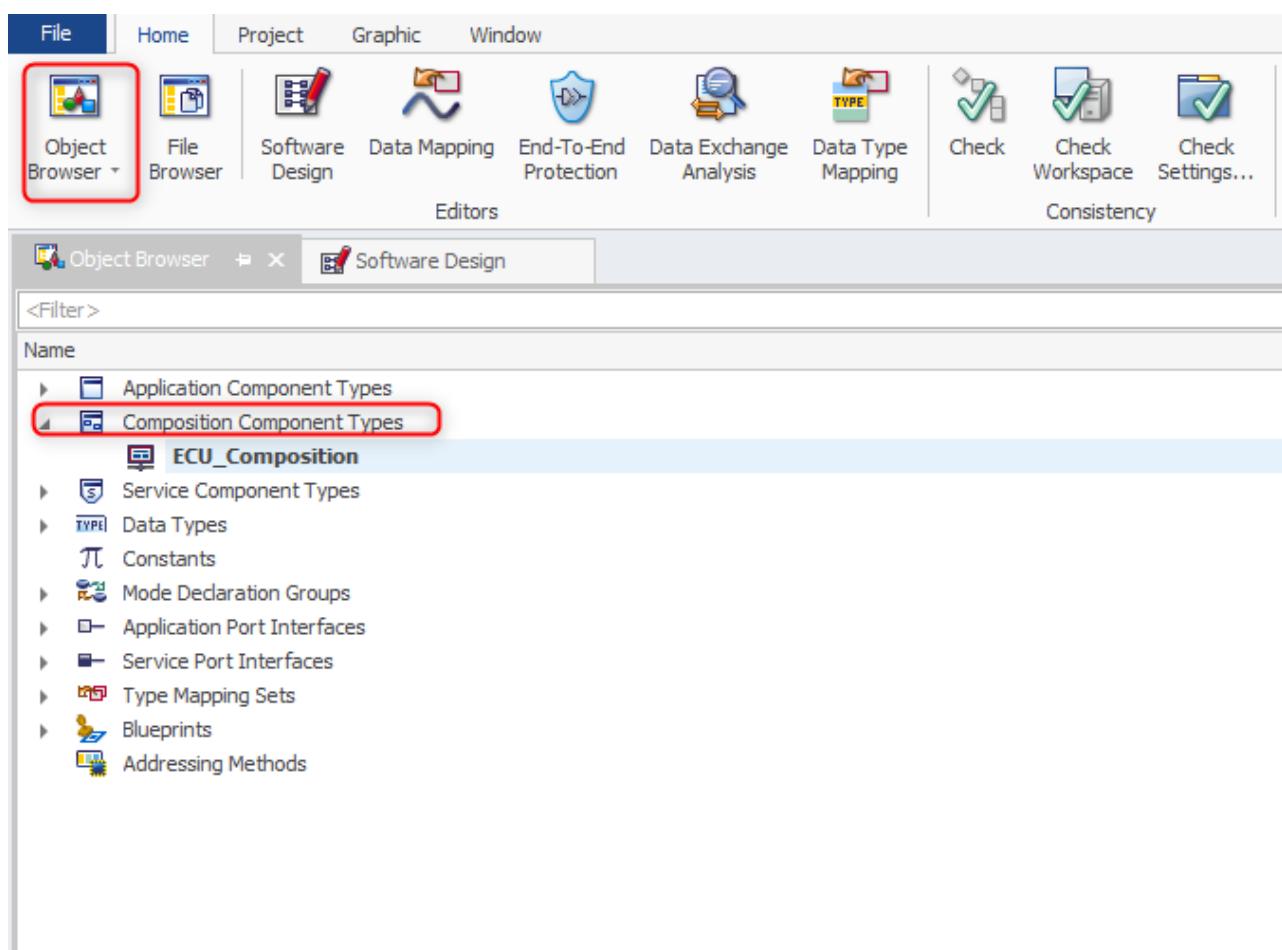
1. Have a project up and running (having the input file and every previous detail).
2. Have two or more SWC that will be inside the Composition Component.

8.3 Steps

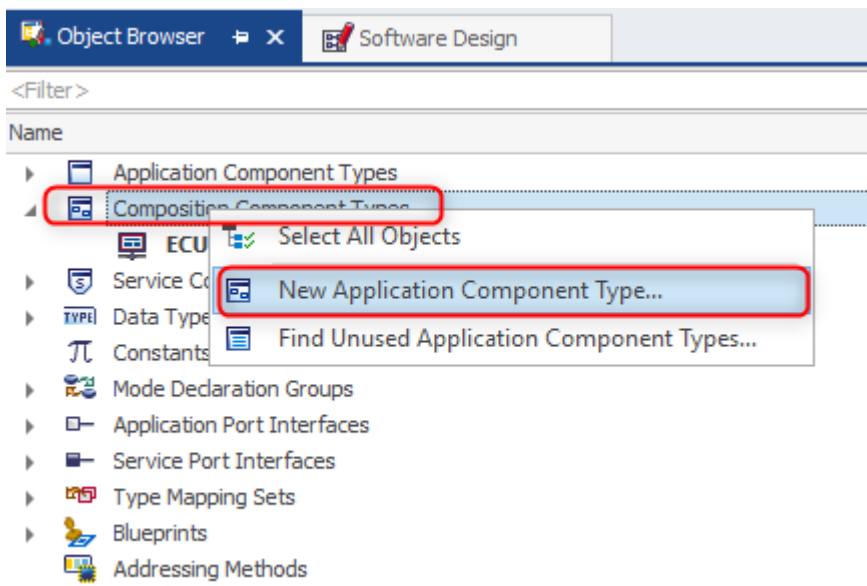
8.4 Steps to be performed in DaVinci Developer Tool

8.4.1 Step 1: Create Composition Component Type

Go to Object Browser → Composition Component Type.



The right click → New Application Componenet Type.

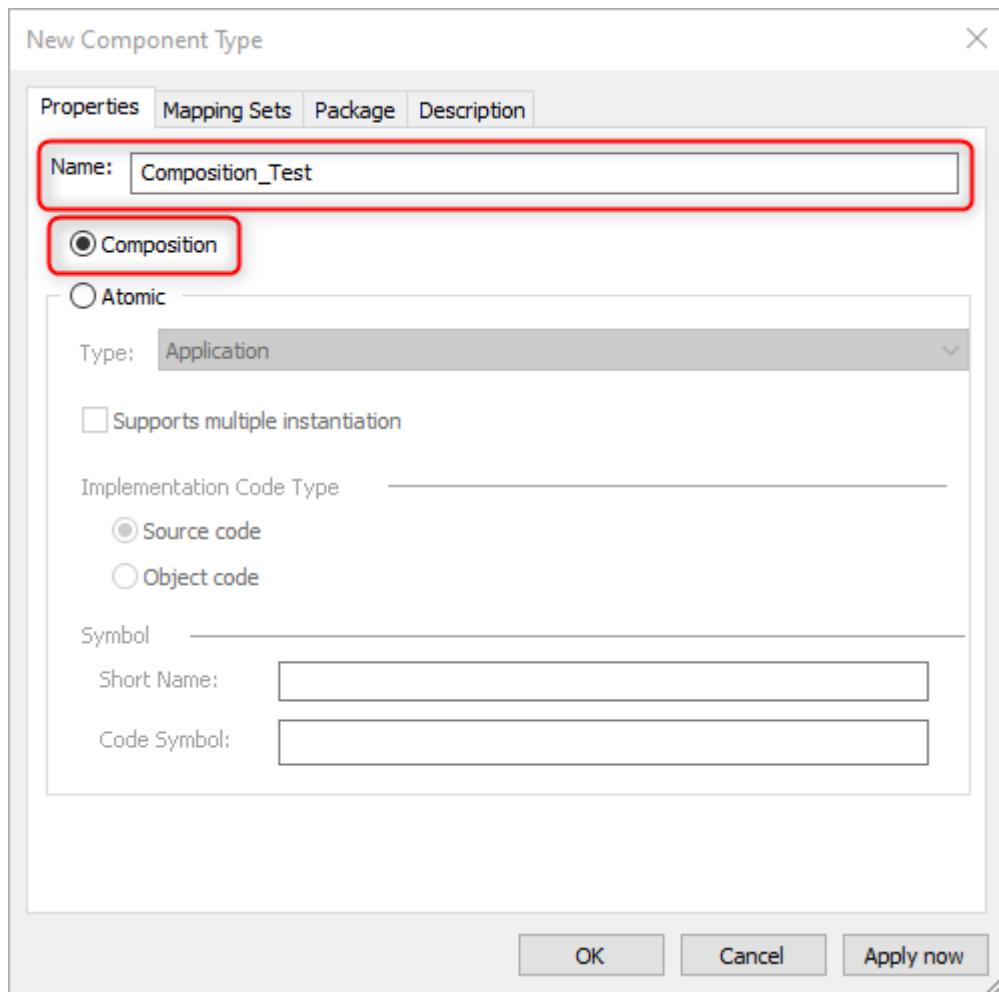


8.4.2 Step 2: Fill the "New Component Type"

In the pop-up window, fill the following Information: and click **OK**

In the Properties tab:

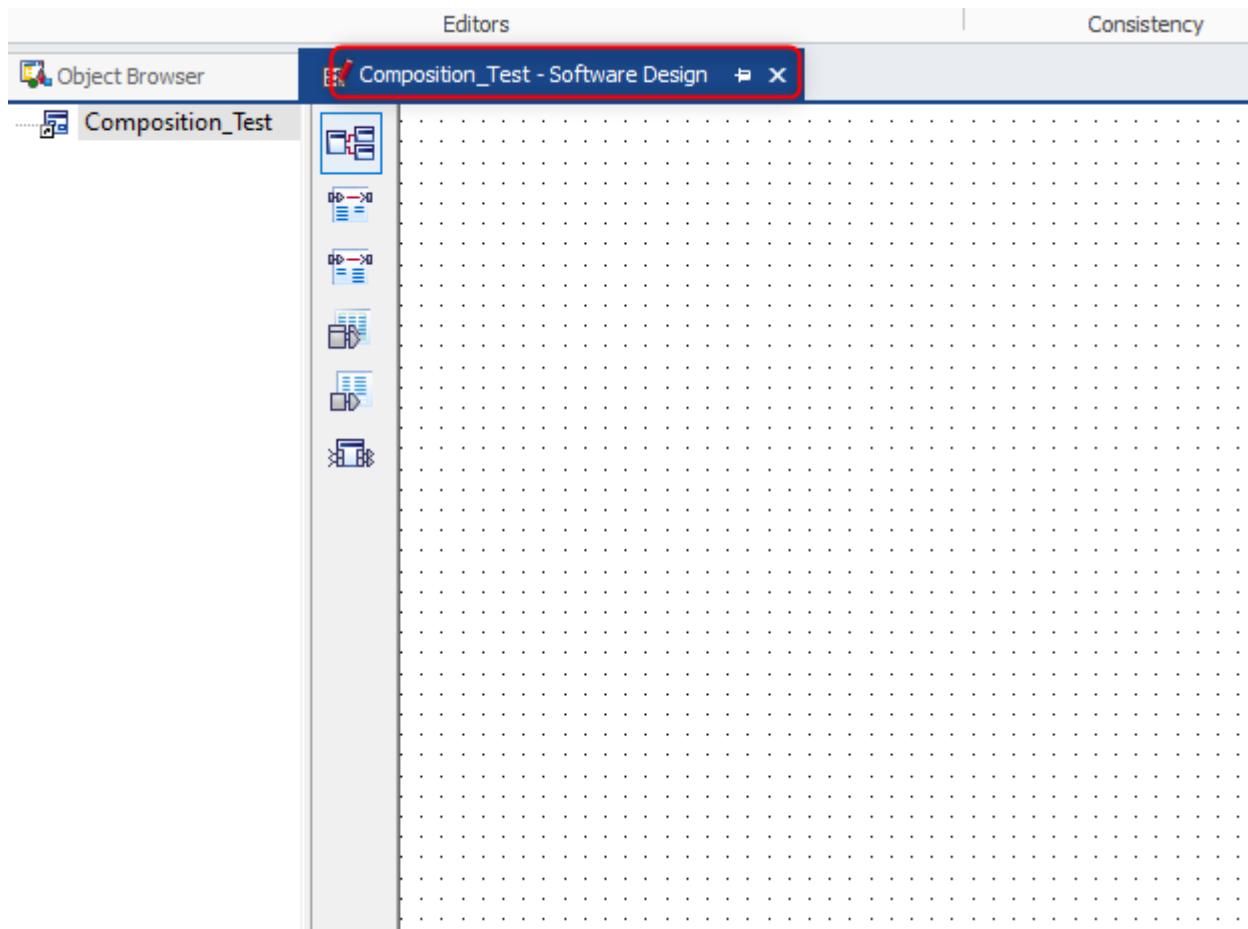
- Name: Create a new name for your Composition SWC.
- Make sure the "Composition" check box is selected.



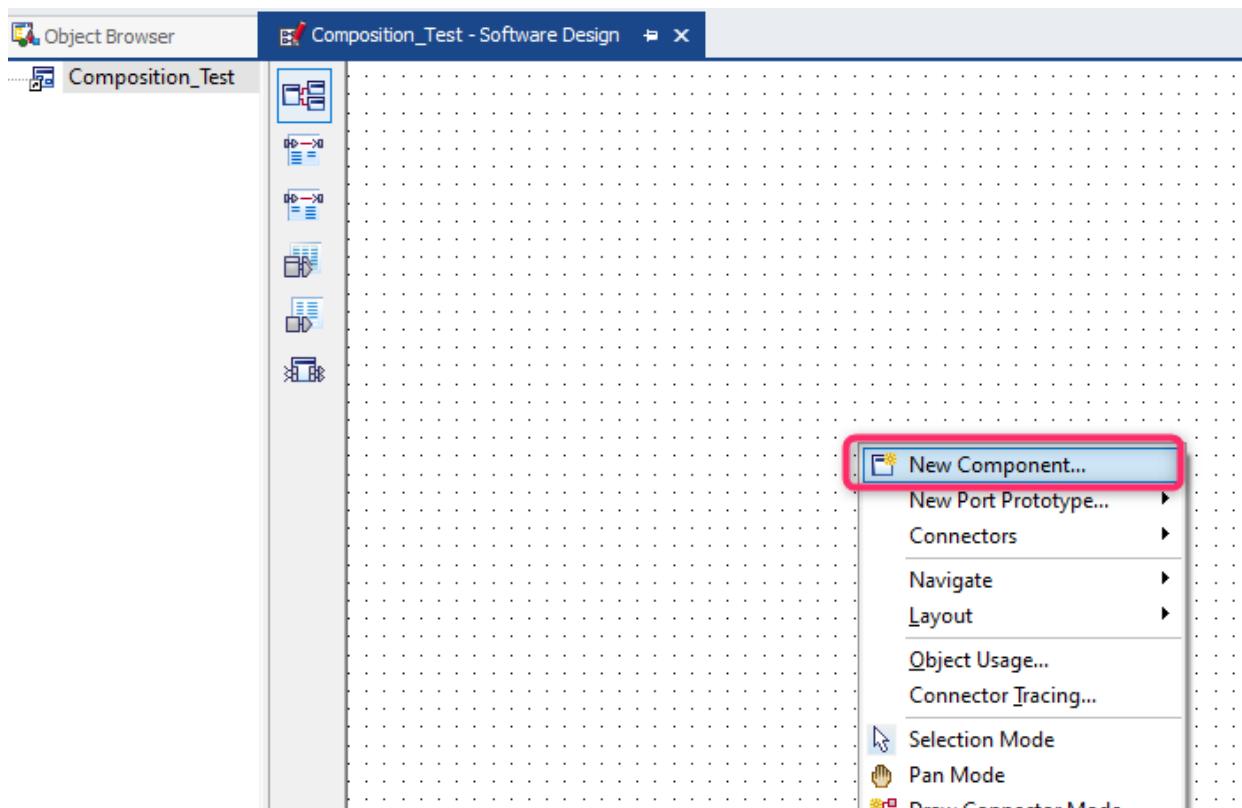
8.4.3 Step 3: Edit Composition Component

To do so, first go to Object Browser → Composition Component Type and Doble-click in the new Composition created.

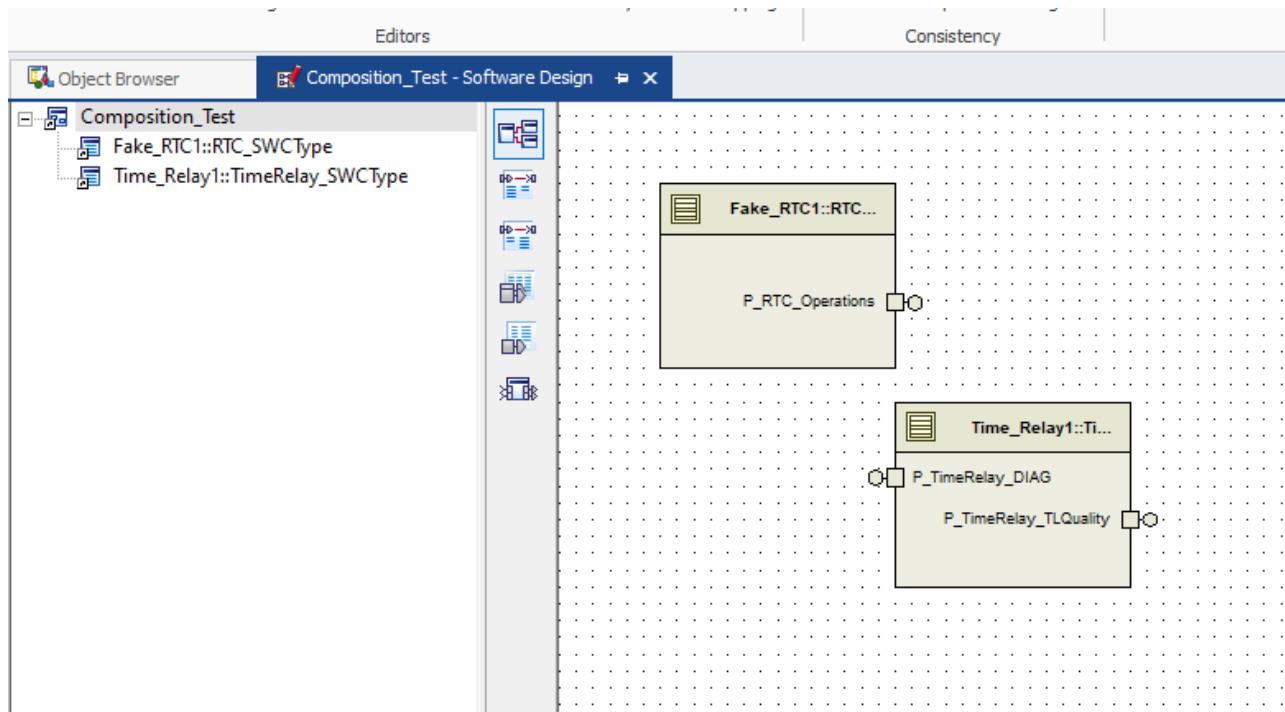
This will open the Software Design for this Composition SWC (which at this moment is empty).



Now, in the canvas right-click → New Component and add the SWC that shall be contained in the new Composition SWC.

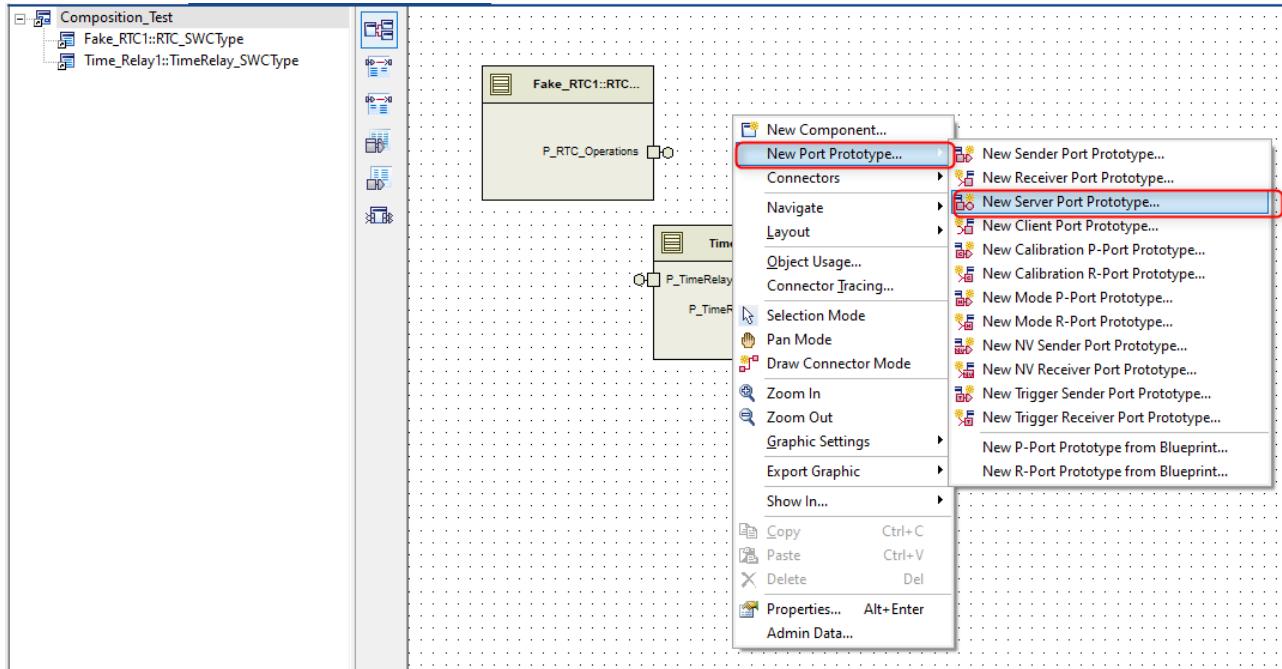


Repeate the adding process untill no more SWC are needed.



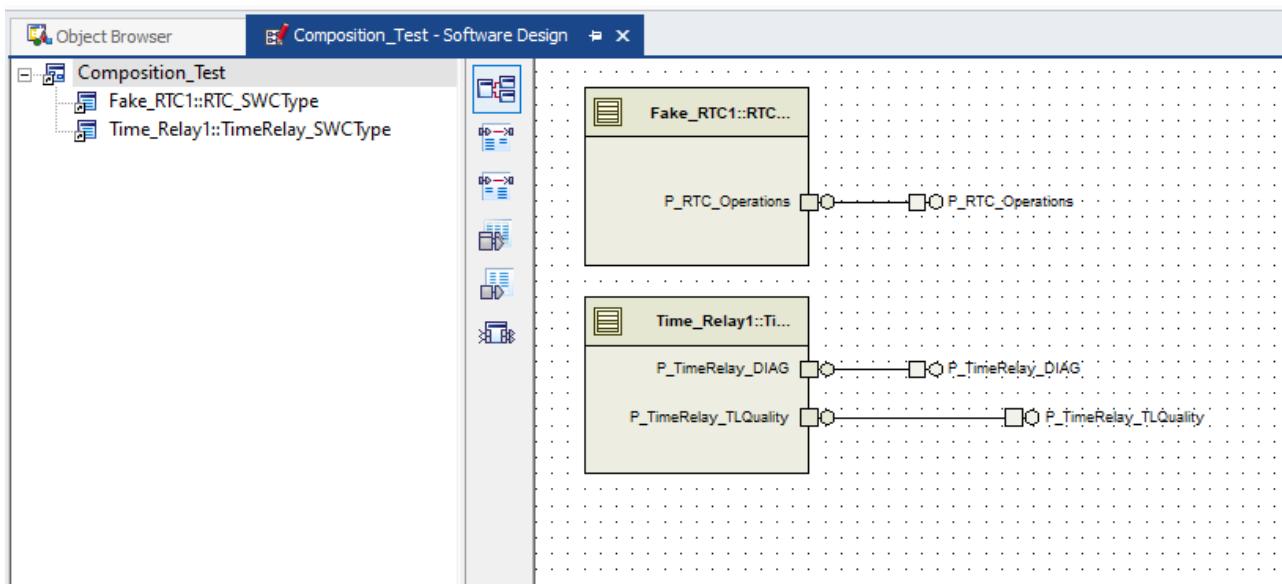
8.4.4 Step 4: Create Input/Outut Ports

In the canvas, right-click → New Port Prototype and add all the necesary ports for the inner SWC to work.



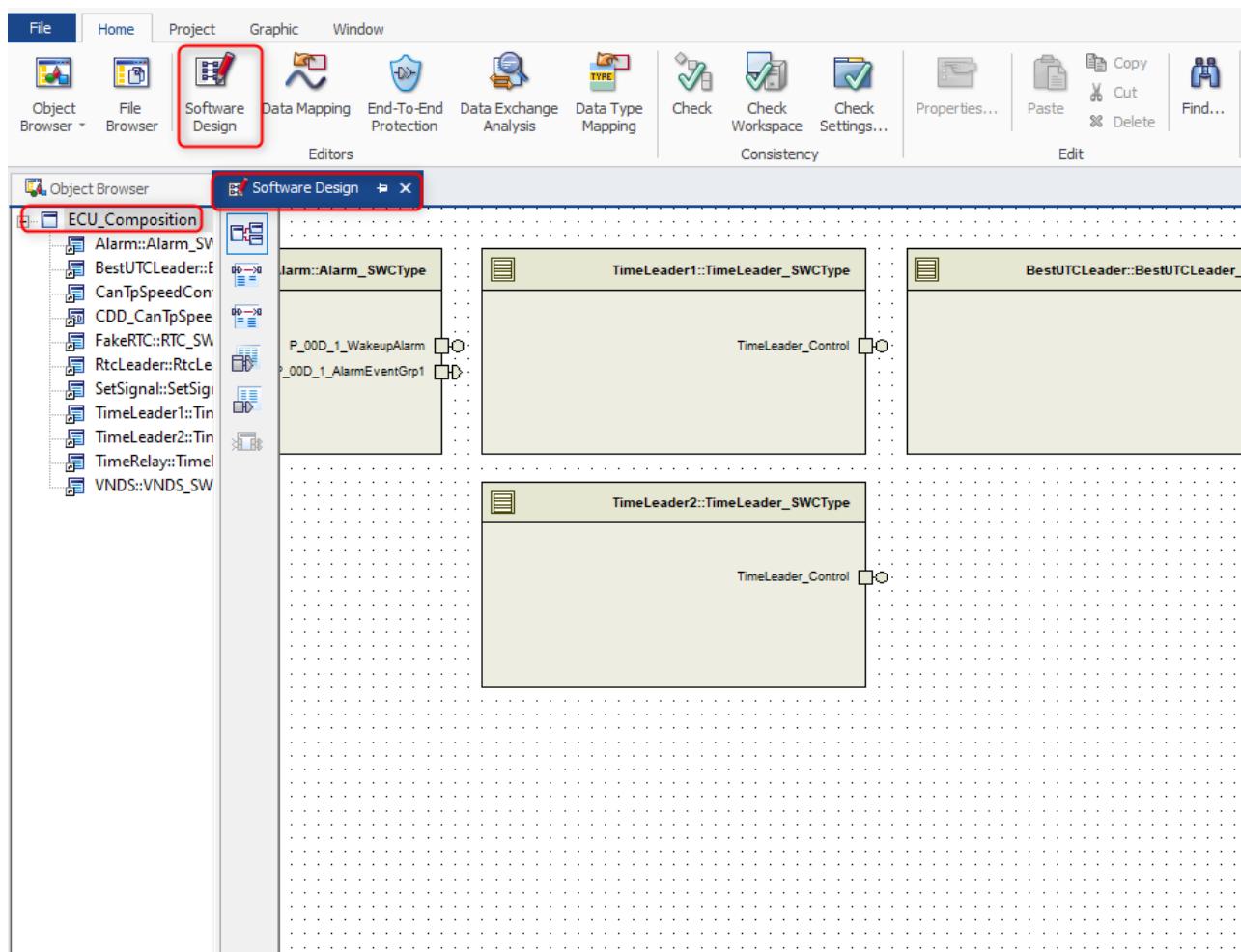
Then connect the ports to the inner SWC.

For this example, the canvas shall look like:

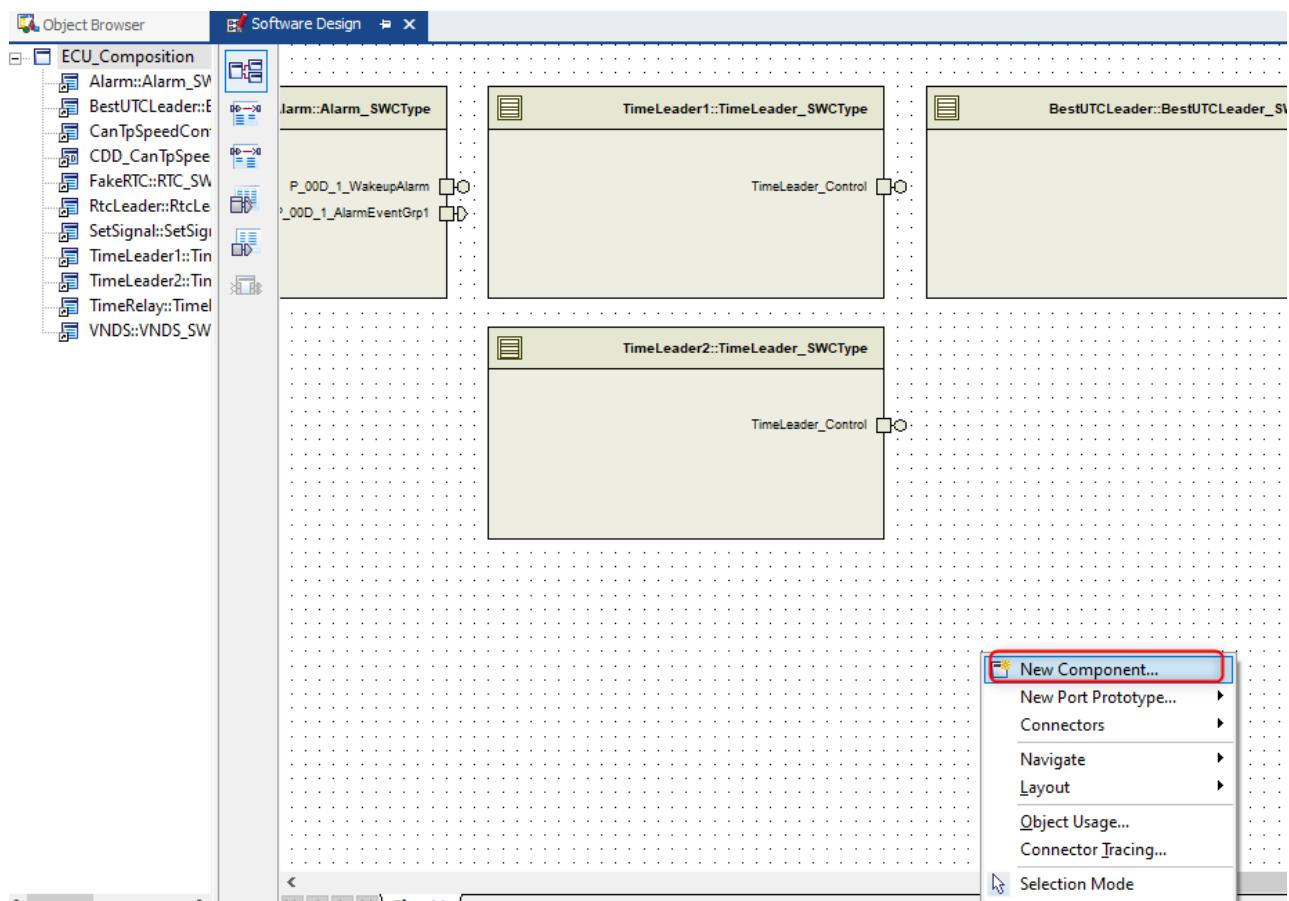


8.4.5 Step 5: Create Composition instance in the ECU Composition.

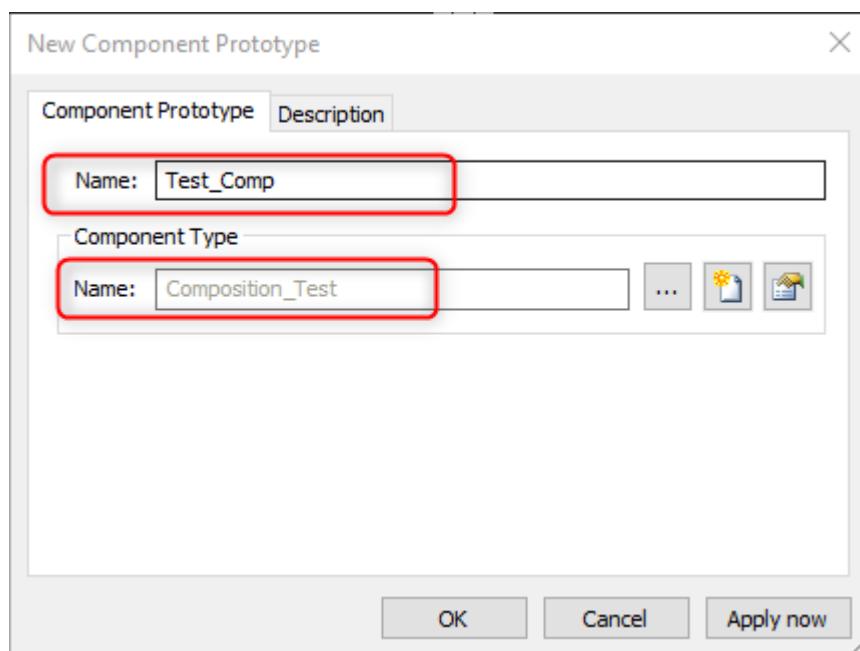
Now go to Home → Software Design → ECU_Composition.



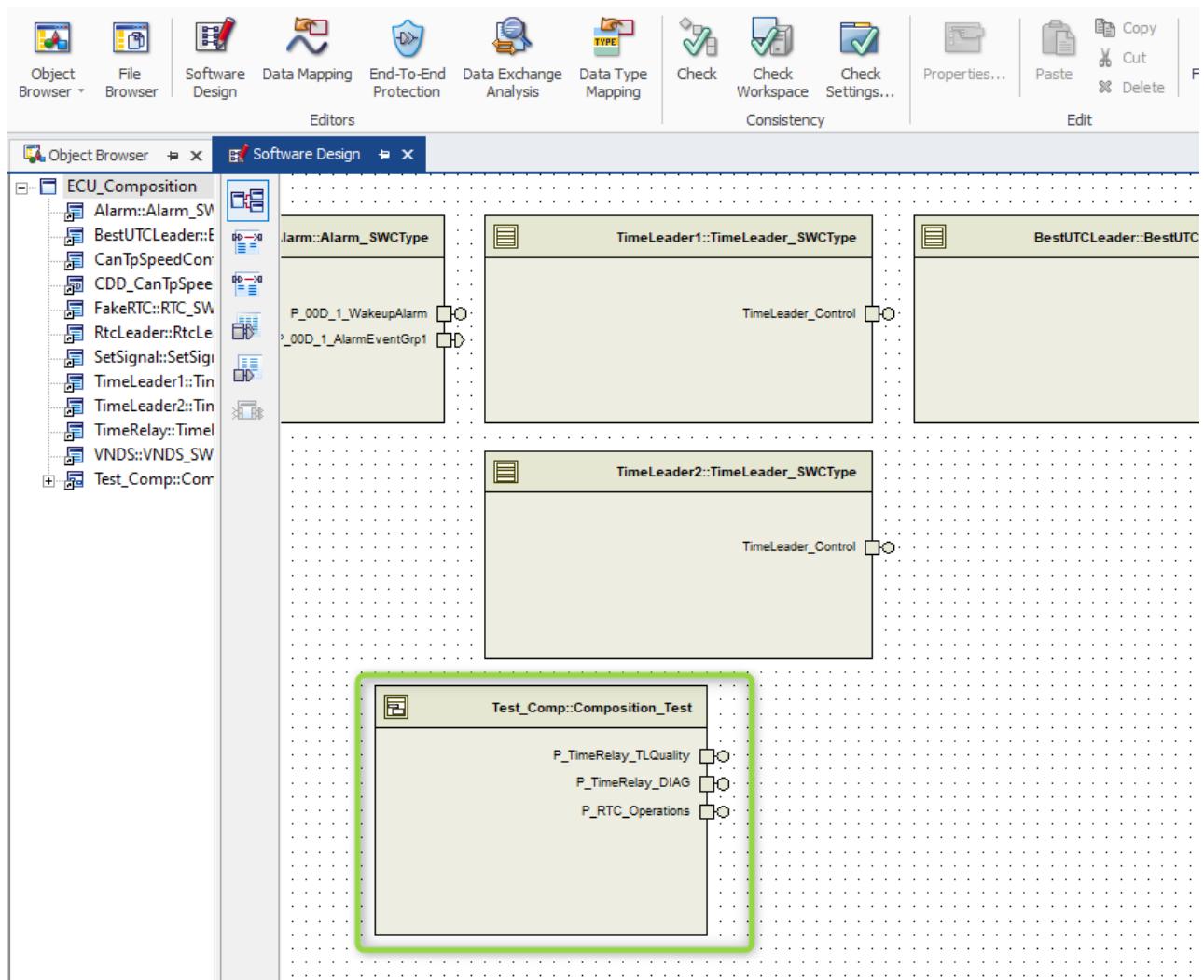
In the canvas, right-click → New Componenet.



In the New Component Window select the new Composition SWC with a unique name.



Now you shall have something like.

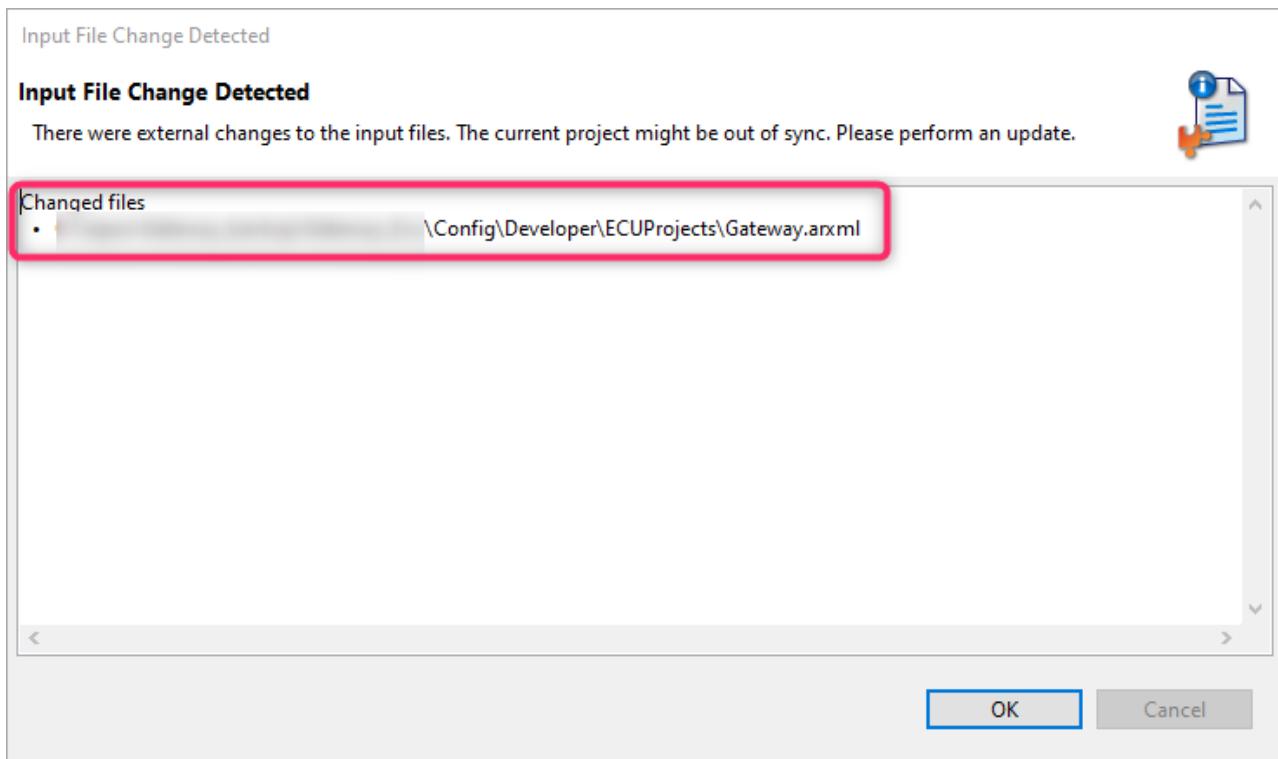


Now just connect the port as necessary.

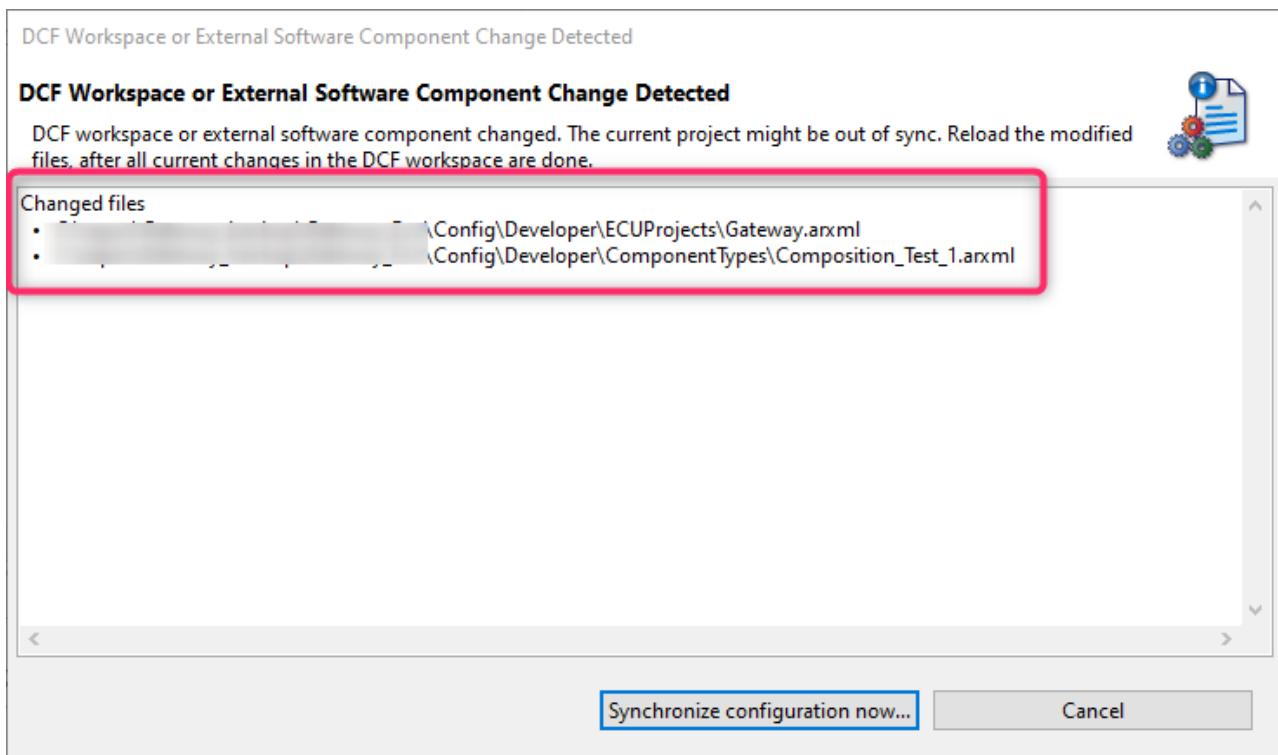
8.5 Steps to be performed in Davinci Configurator Tool

8.5.1 Step 6: Synchronize Project

After saving the Project in Davinci Developer, the Davinci Configurator will show a PopUp window requesting the Synchronization.



After the general Synchronization a more general one is needed.

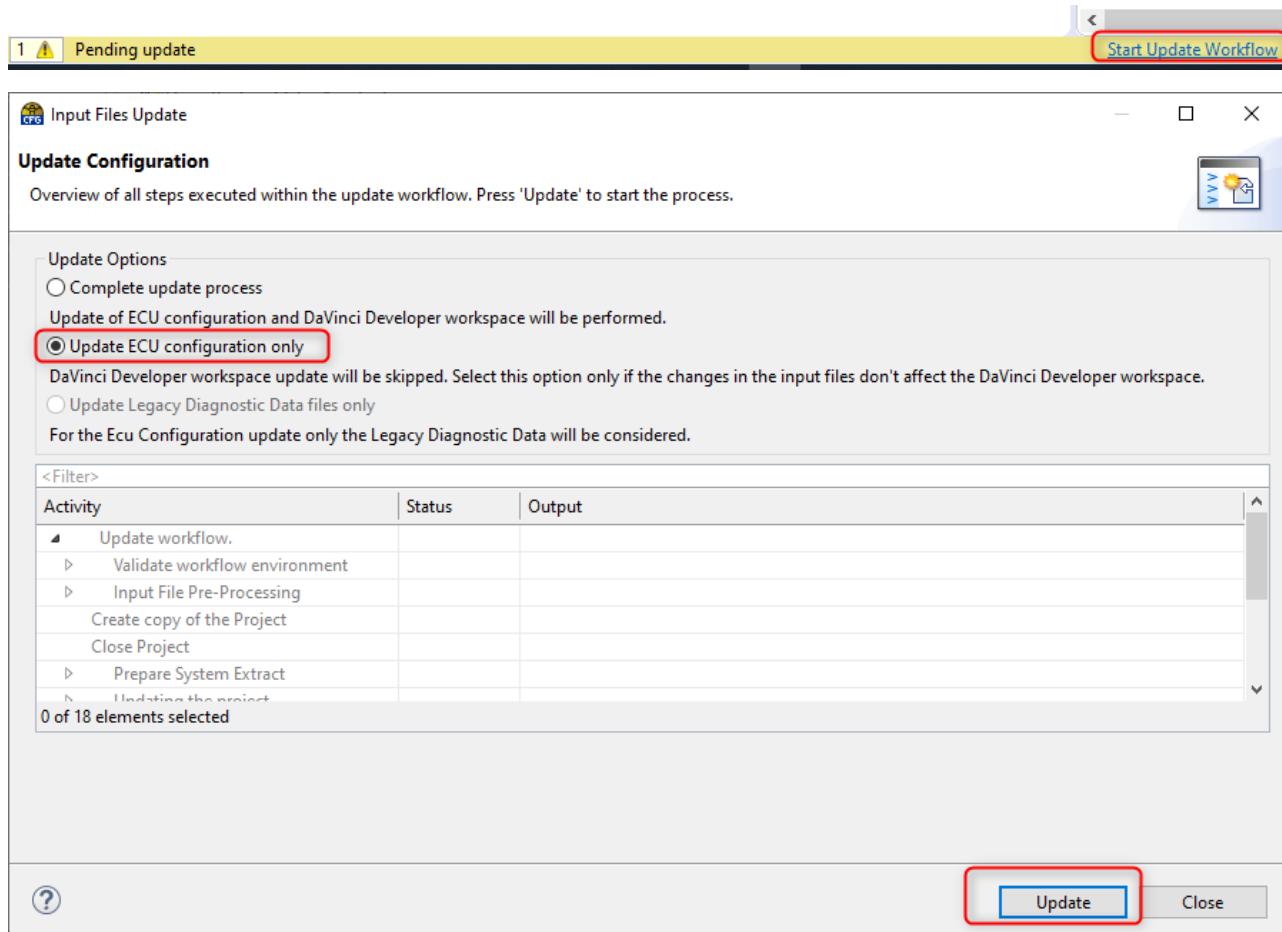


8.5.2 Step 7: Update ARXML Files in Davinci Cfg

⚠ Sync Needed

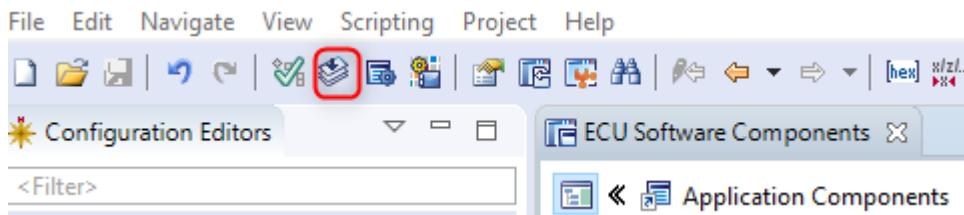
Please consider that after modifying (and saving) the ARXML file, an Update shall be need, please follow the recommended Updating process by clicking the Yellow ribbon in the Vector Davinci Configurator Tool. Please make use to follow the Update show in the Following Images:

Click, start Update Workflow, and select in the Checkbox "Update ECU Configuration only".

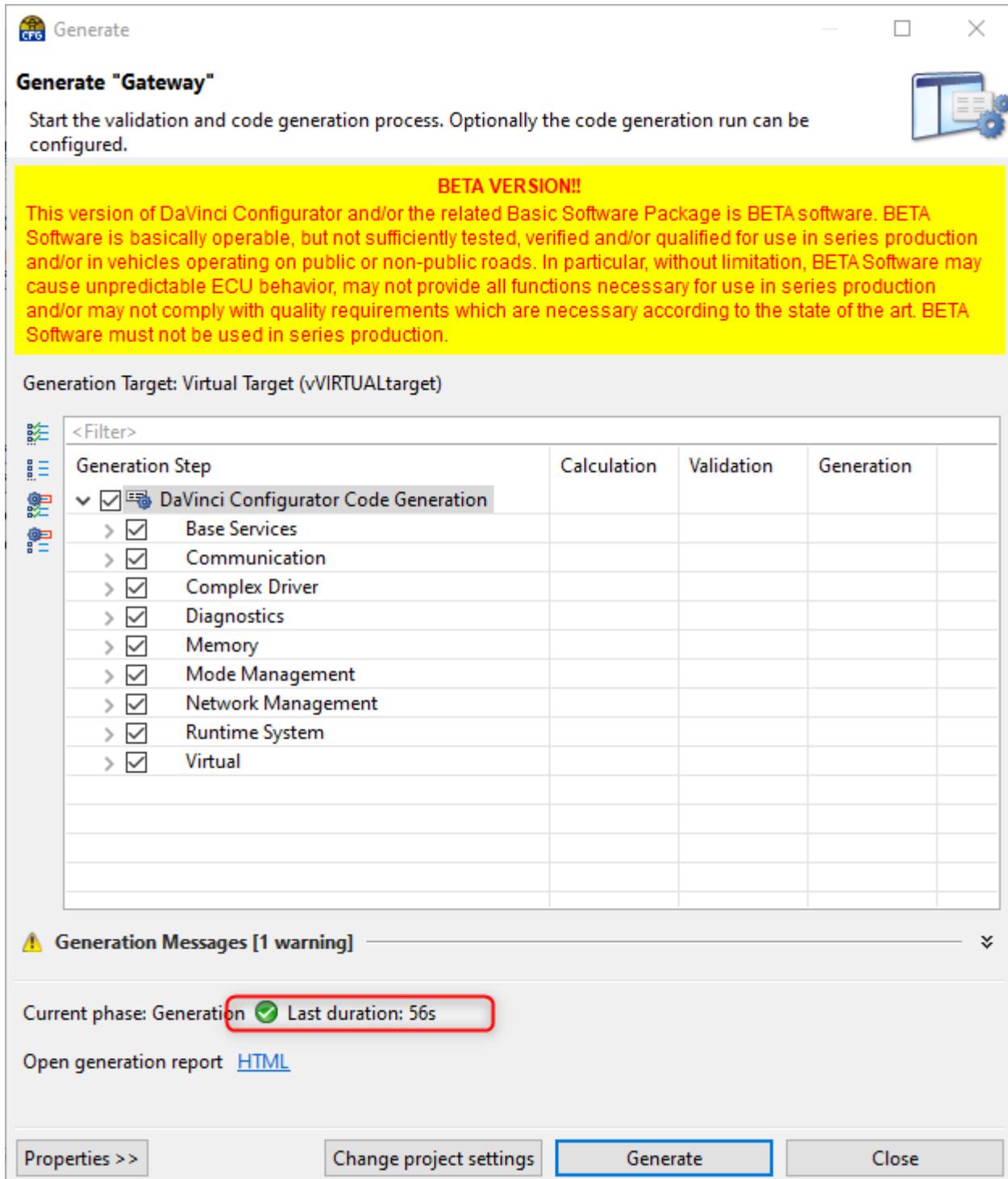


8.5.3 Step 8: Generate the Project

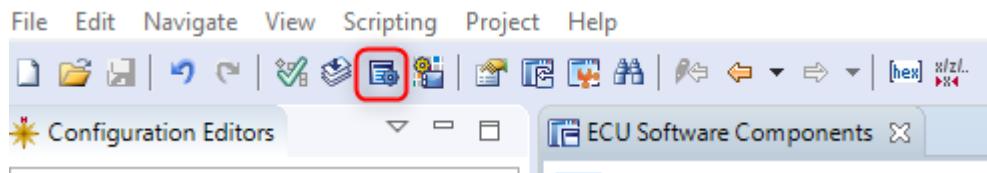
Go to Generate (Or click F9)



And Generate as usual



And finally generate the corresponding templates



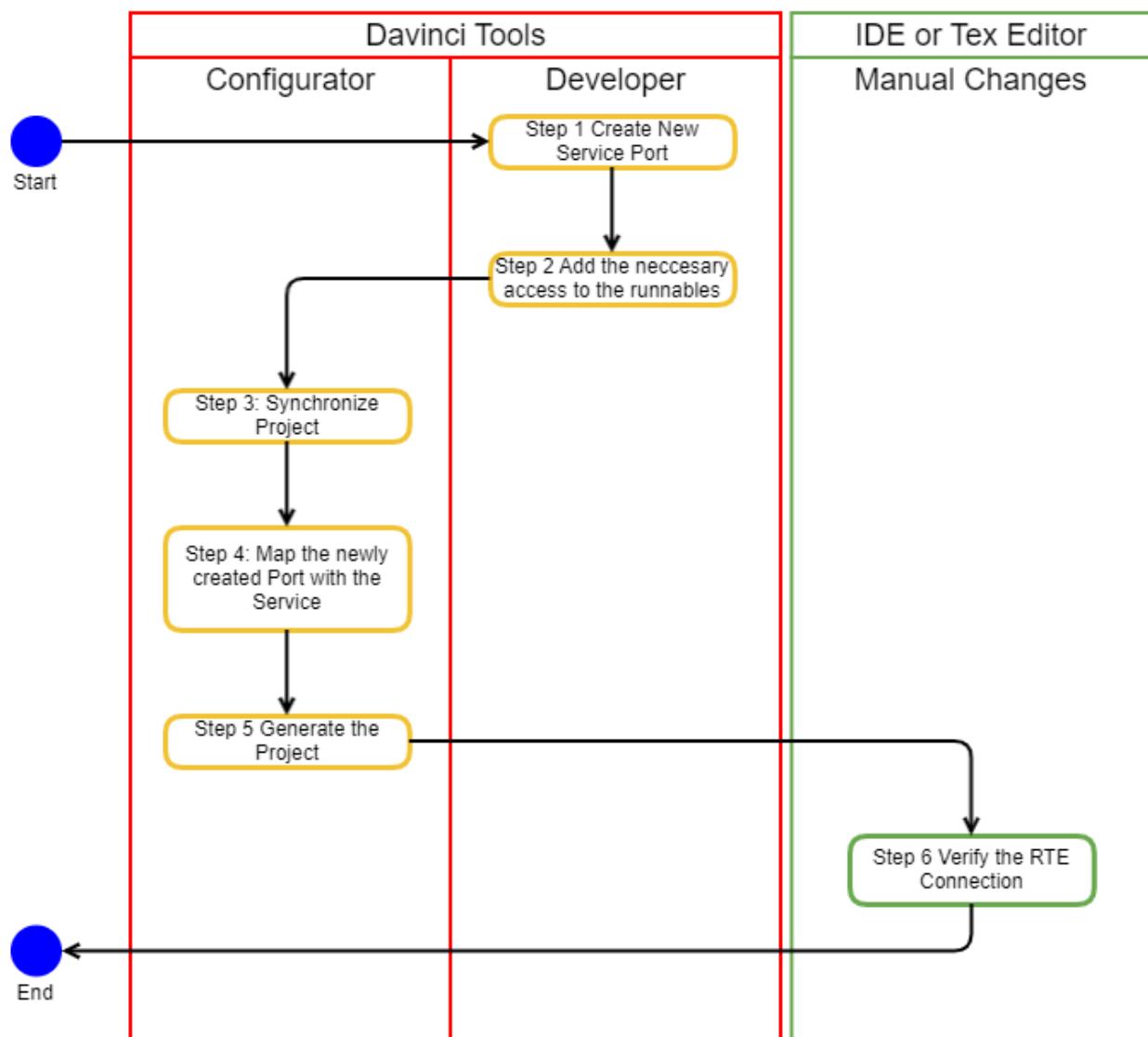
8.6 Troubleshooting

9 7_Create Connection between a SWC and a Service Component

⚠ Please consider this recipe is using the Stbm as example, but the steps are the same for every Service.

- [Workflow Diagram\(see page 87\)](#)
- [Prerequisites\(see page 88\)](#)
- [Steps\(see page 89\)](#)
- [Steps to be performed in Davinci Developer\(see page 89\)
 - \[Step 1 Create New Service Port\\(see page 89\\)\]\(#\)
 - \[Step 2 Add the neccesary access to the runnables\\(see page 93\\)\]\(#\)](#)
- [Steps to be performed in Davinci Configurator\(see page 96\)
 - \[Step 3: Synchronize Project\\(see page 96\\)\]\(#\)
 - \[Step 4: Map the newly created Port with the Service\\(see page 96\\)\]\(#\)
 - \[Step 5 Generate the Project\\(see page 98\\)\]\(#\)](#)
- [Steps to be performed in and external IDE tool\(see page 100\)
 - \[Step 6 Verify the RTE Connection\\(see page 100\\)\]\(#\)](#)
- [Troubleshooting\(see page 100\)](#)

9.1 Workflow Diagram



9.2 Prerequisites

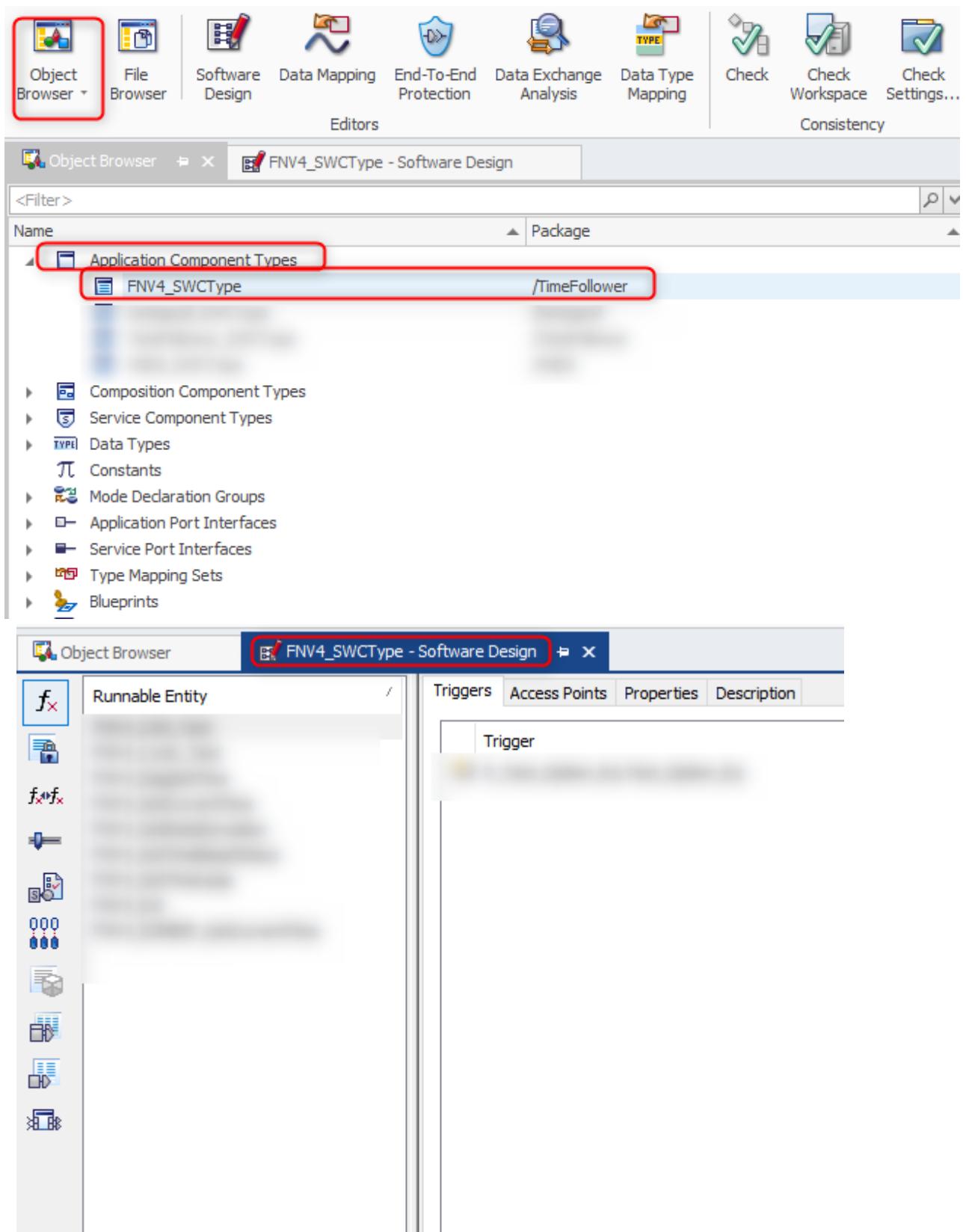
1. Have the Service Component properly added and configured.
2. Have the Software component properly created and configured.

9.3 Steps

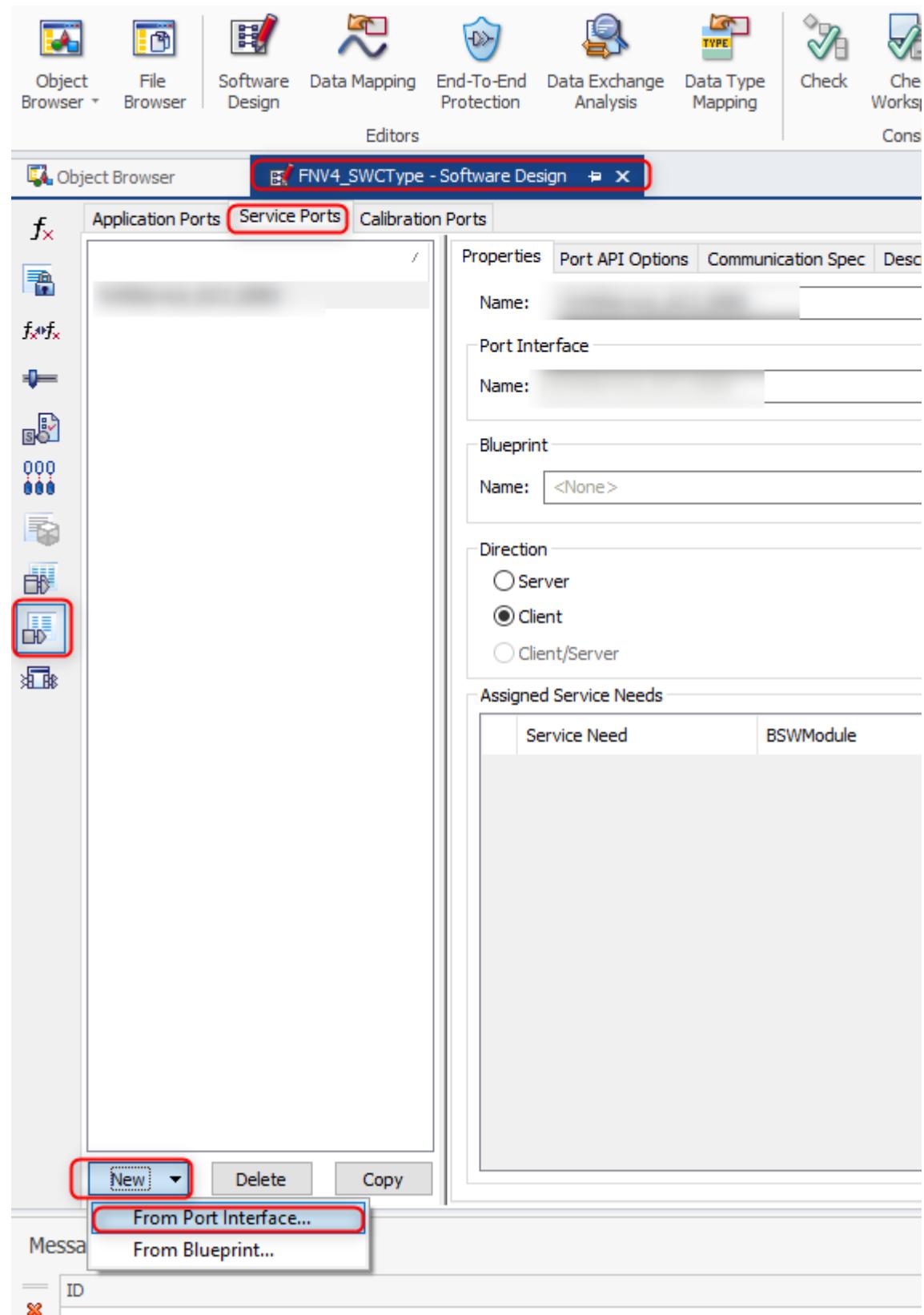
9.4 Steps to be performed in Davinci Developer

9.4.1 Step 1 Create New Service Port

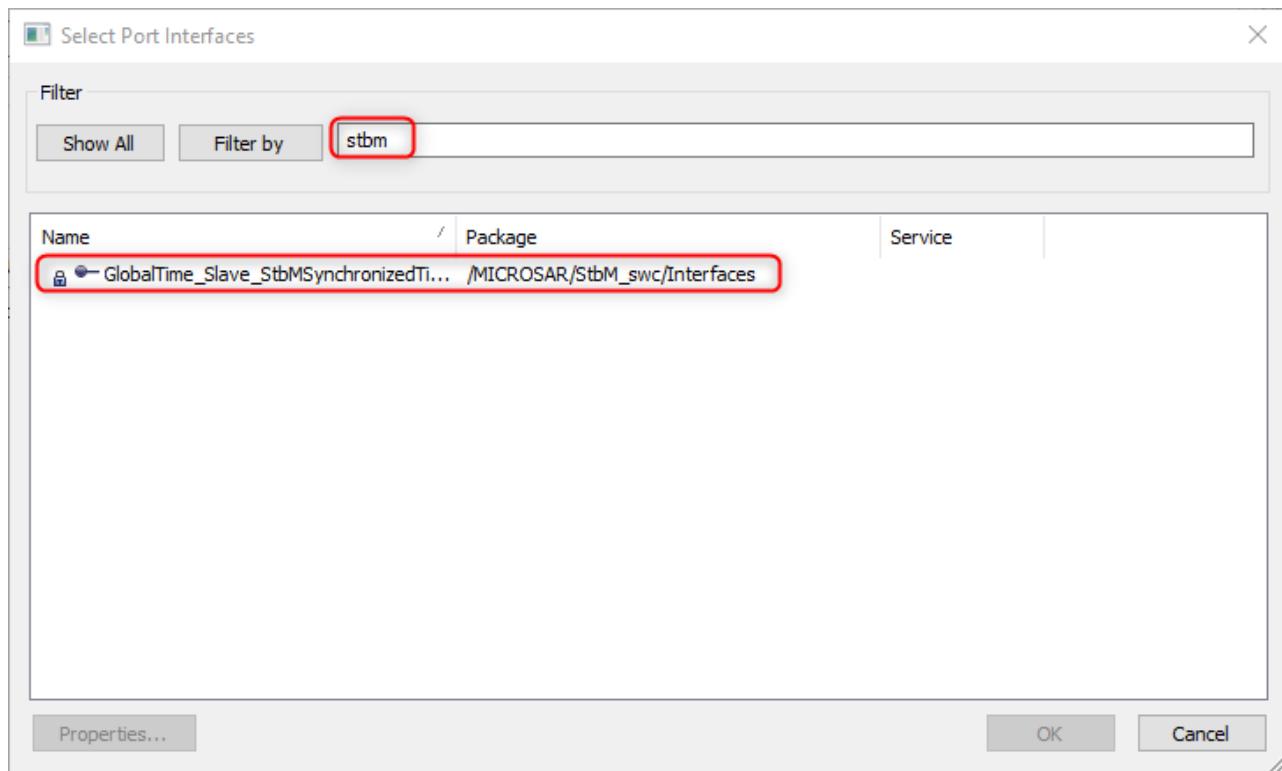
Go to Object Browser → Application Component Type → Doble click on the SWC to be connected.



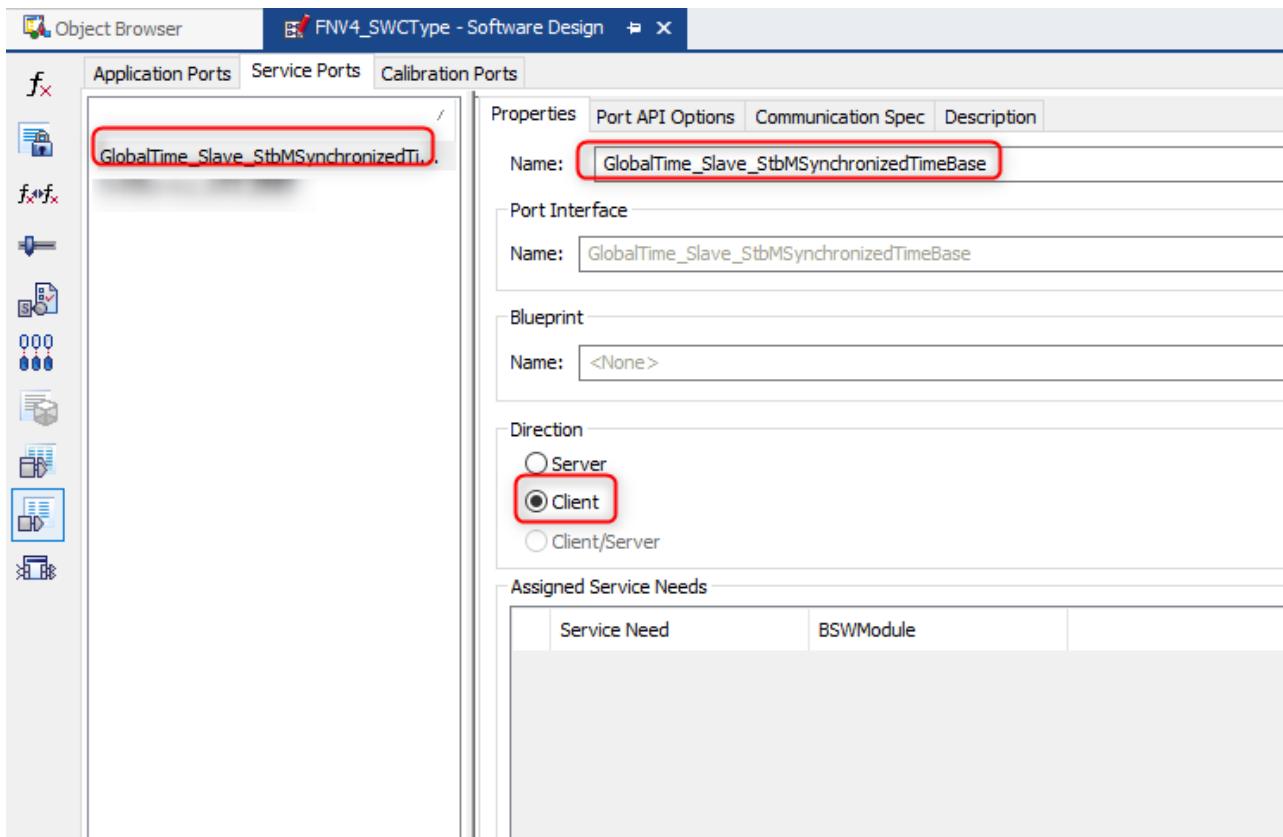
Go to Port Prototype List → Service Ports → New and Click in From Port Interface



Now from the PopUp menu, select the Ports from the service needed (in this Example Stbm Component) and click Ok



The result shall look like the following image

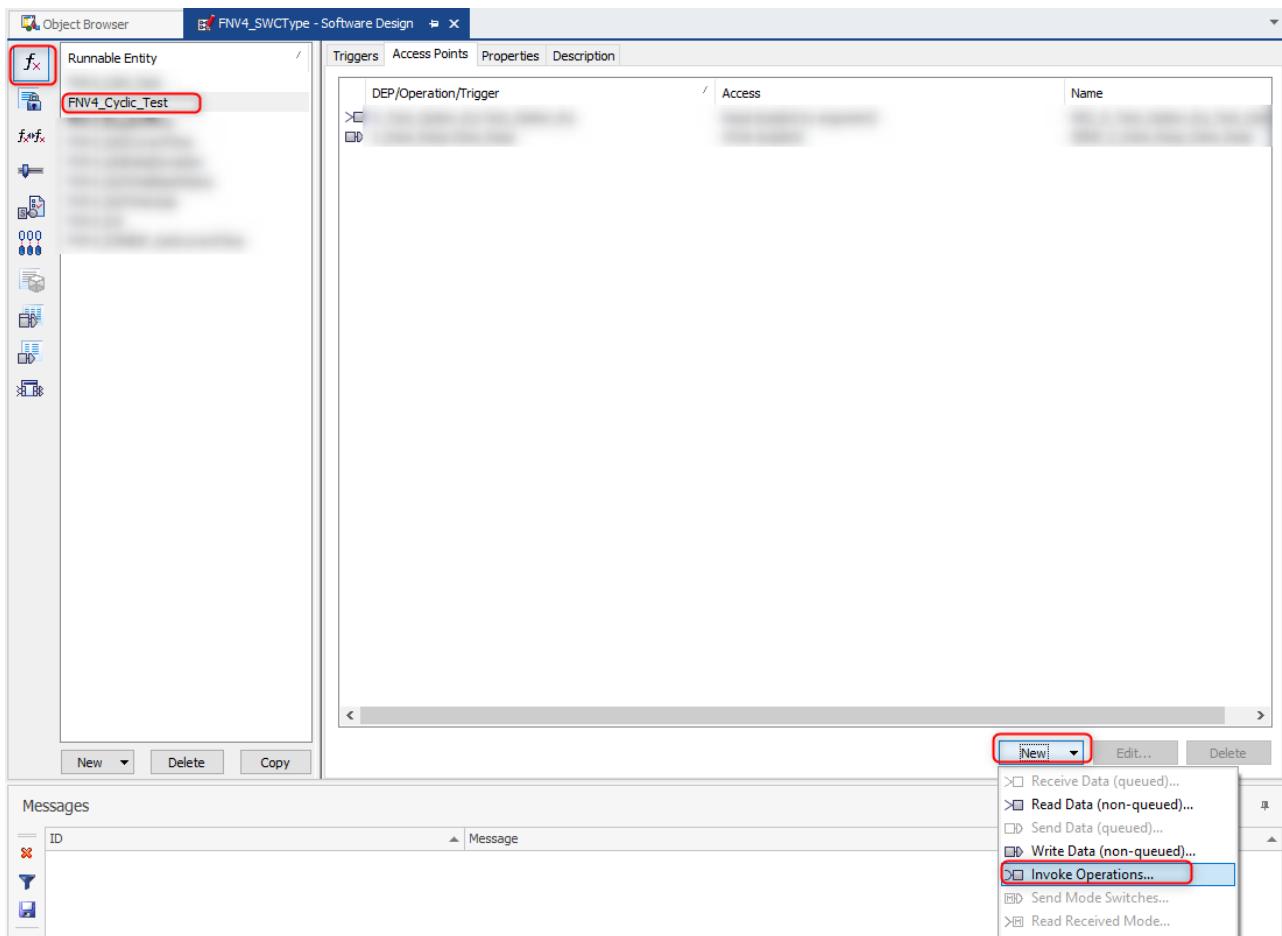


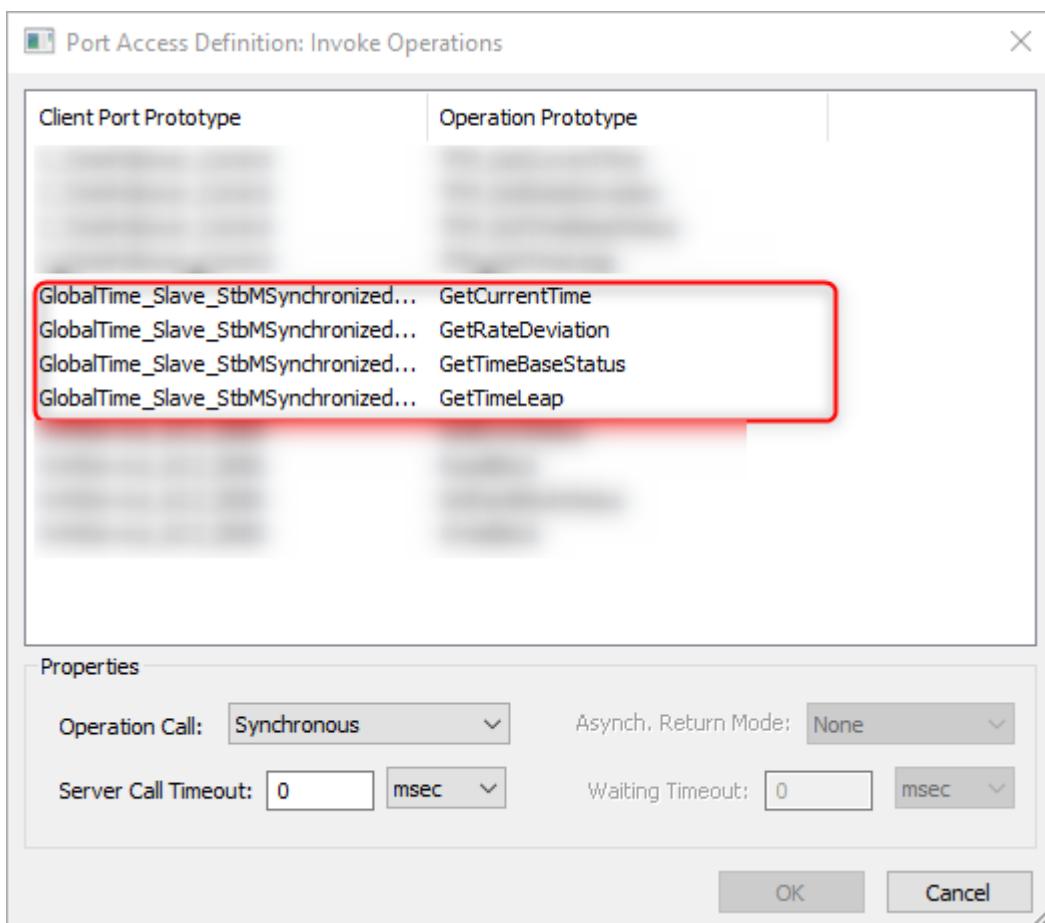
Note: Please consider that if your service is not listed on the Menu, is because it needs to be configured prior this recipe.

9.4.2 Step 2 Add the neccesary access to the runnables

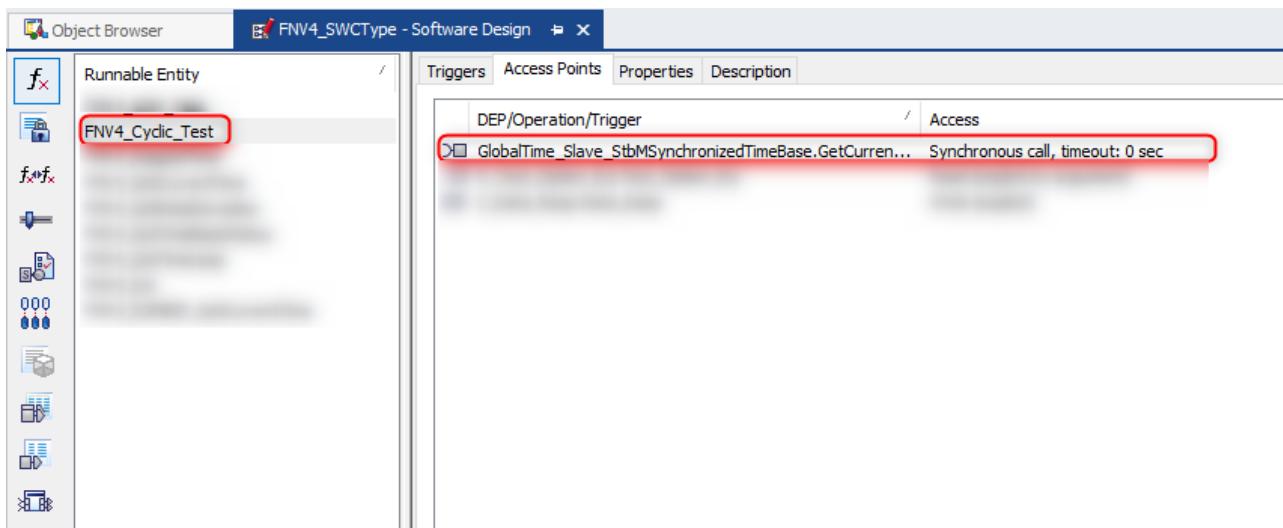
Go to Runnable Entityy List → (Runnbale) → Access Points → New, and provide Invoke Operation access to hte proper runnable.

Example:





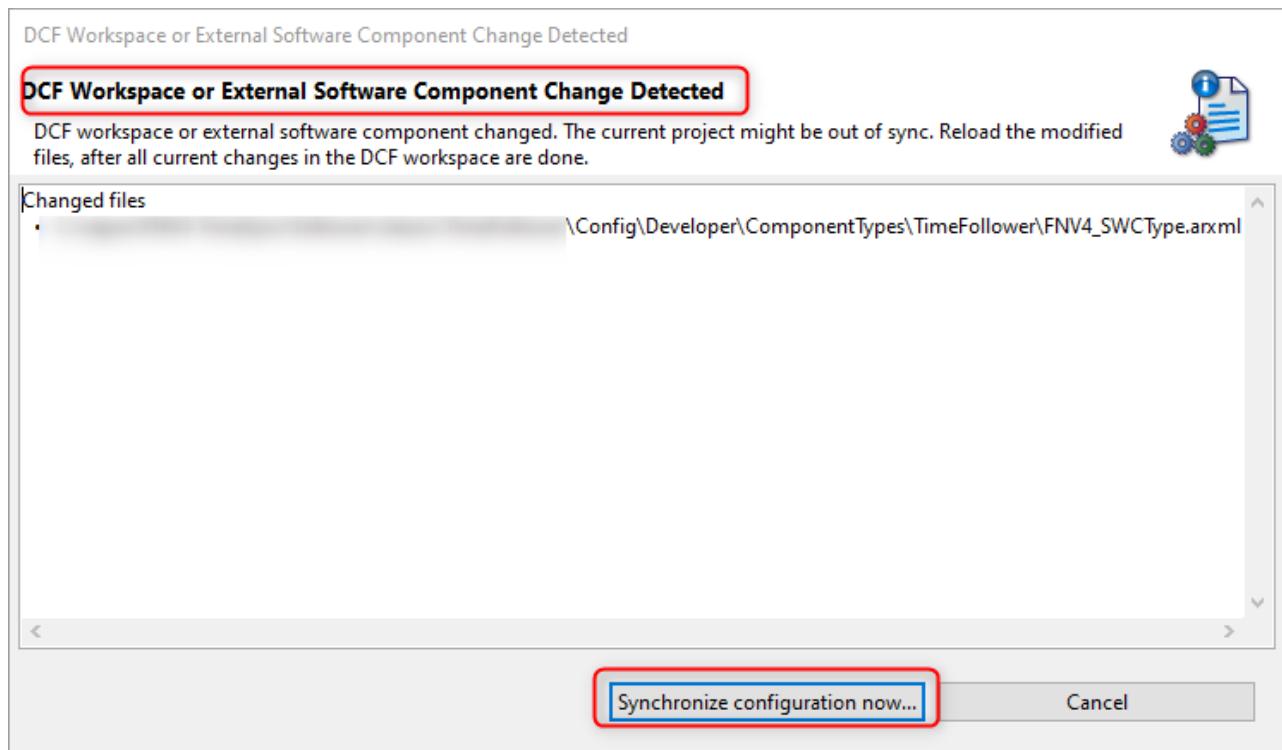
The end result shall look something like the following image



9.5 Steps to be performed in Davinci Configurator

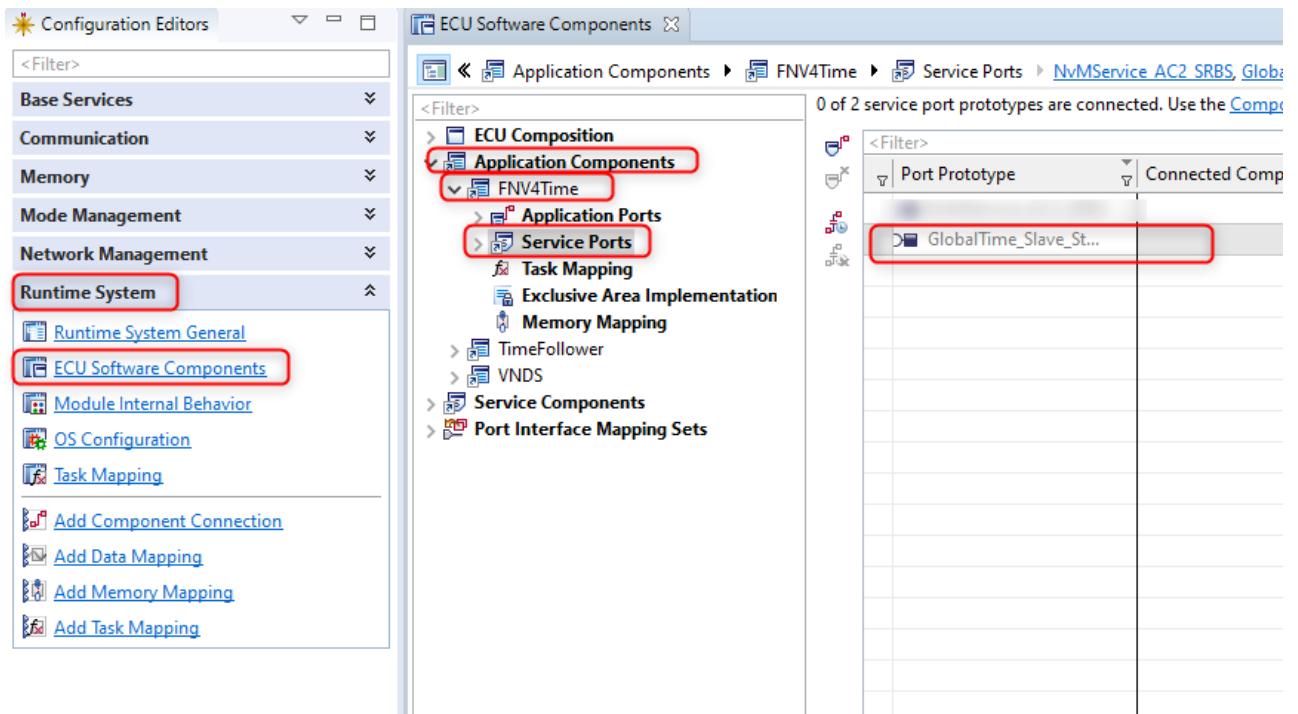
9.5.1 Step 3: Synchronize Project

After saving the Project in Davinci Developer, the Davinci Configurator will show a PopUp window requesting the Synchronization.

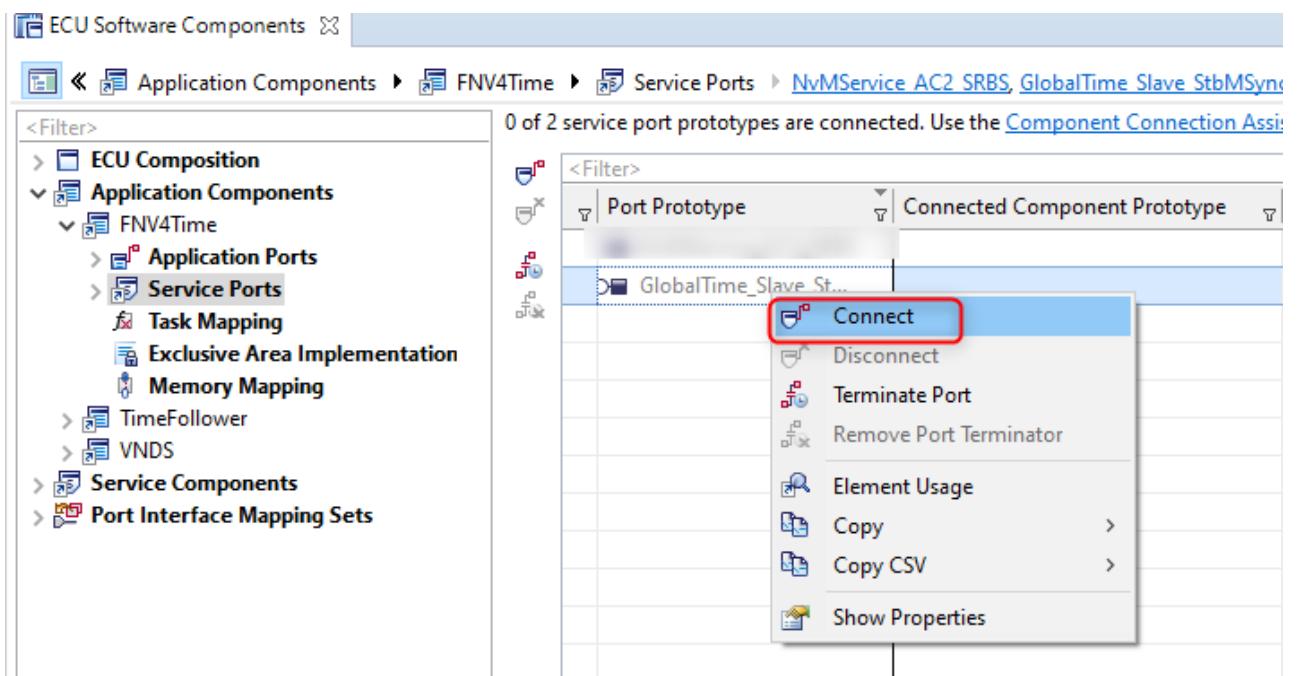


9.5.2 Step 4: Map the newly created Port with the Service

Go to Runtime Systems =>ECU Software Components → Application Components → (Your component) → Service Ports:



Now select the Port you just created and Right click and Select the option Connect



Now select the Service Component related to the created Port and click Finish

Select Port Prototype

Please select port prototype.

 Show unconnected ports only Use also the origin context for name matching

<Filter>	Compatibility	Component Prototype	Port Prototype	Port Interface	Port Interface Mapping
<input checked="" type="checkbox"/>	600	StbM	GlobalTime_Slave_StbMSynchronizedTimeBase	GlobalTime_Slave_StbMSynchronizedTimeBase	-
<input type="checkbox"/>	0	ComM	modeLimitation	ComM_ECUModeLimitation	-
<input type="checkbox"/>	0	ComM	CL_con_FD1_GWM_fe528cca	ComM_ChannelLimitation	-
<input type="checkbox"/>	0	ComM	CW_con_FD1_GWM_fe528cca	ComM_ChannelWakeUp	-
<input type="checkbox"/>	0	ComM	CL_con_Eth_VNDS_GWM	ComM_ChannelLimitation	-
<input type="checkbox"/>	0	ComM	CW_con_Eth_VNDS_GWM	ComM_ChannelWakeUp	-
<input type="checkbox"/>	0	ComM	UR_con_FD1_GWM_3b4ddca8	ComM_UserRequest	-
<input type="checkbox"/>	0	ComM	UR_user_Eth_VNDS_GWM	ComM_UserRequest	-
<input type="checkbox"/>	0	Det	VNDS_DetModuleInstance	DETService	-
<input type="checkbox"/>	0	EcuM	EcuM_ShutdownTarget	EcuM_ShutdownTarget	-
<input type="checkbox"/>	0	EcuM	EcuM_BootTarget	EcuM_BootTarget	-

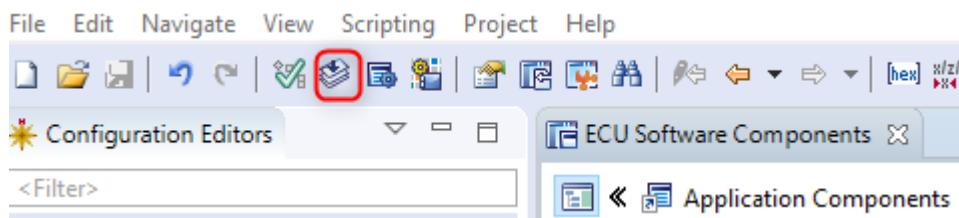
1 of 11 elements selected. Sorting by <Compatibility>

Finish

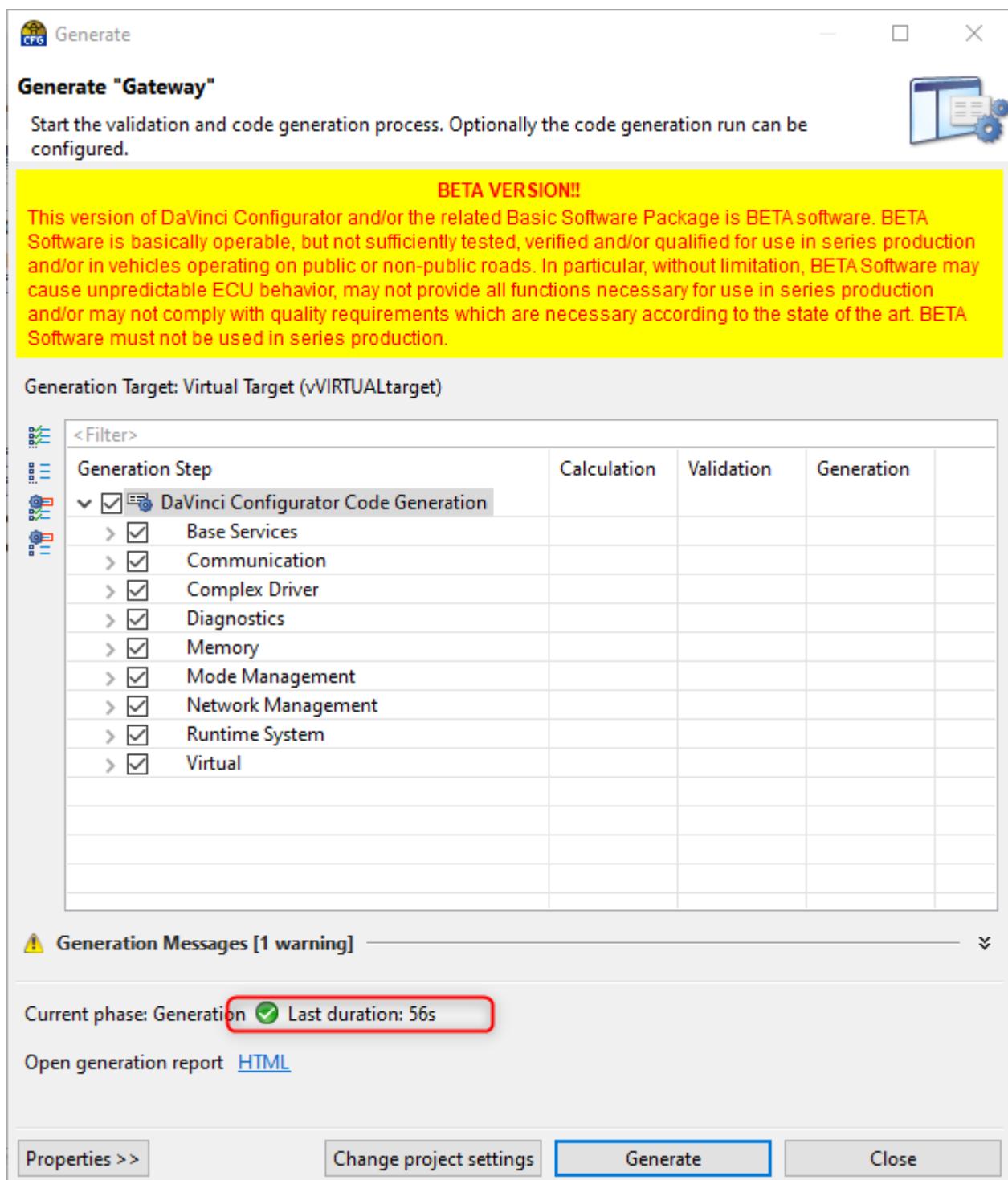
Cancel

9.5.3 Step 5 Generate the Project

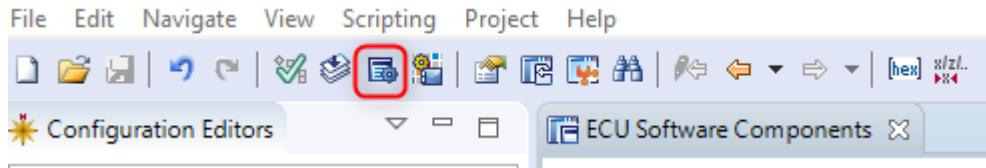
Go to Generate (Or click F9)



And Generate as usual



And finally generate the corresponding templates



9.6 Steps to be performed in and external IDE tool

9.6.1 Step 6 Verify the RTE Connection

Using an IDE or a Text Editor please make sure the RTE connections are benn done correctly

Go to ..\Appl\GenDataVtt\Components\Rte_XXXXXXXXXXXXXX_SWCType.h this file contain all the RTE connection of your component.

Look for the Rte_Call section and make sure the connection is created as shown below

```
/*******************************************************************************
 * Rte_Call_C_P<>_O<> (unmapped) for synchronous C/S communication
 ******************************************************************************/
#define Rte_Call_C_TimeFollower_Control_TFM_GetCurrentTime Rte_Call_FN4_SWCType_C_TimeFollower_Control_TFM_GetCurrentTime
#define Rte_Call_C_TimeFollower_Control_TFM_GetRateDeviation Rte_Call_FN4_SWCType_C_TimeFollower_Control_TFM_GetRateDeviation
#define Rte_Call_C_TimeFollower_Control_TFM_GetTimeBaseStatus Rte_Call_FN4_SWCType_C_TimeFollower_Control_TFM_GetTimeBaseStatus
#define Rte_Call_C_TimeFollower_Control_TFM_GetTimeLeap Rte_Call_FN4_SWCType_C_TimeFollower_Control_TFM_GetTimeLeap
#define RTE_START_SEC_STBM_APPL_CODE
#include "MemMap.h" /* PRQA S 5087 */ /* MD_MSR_MemMap */
FUNC(Std_ReturnType, RTE_STBM_APPL_CODE) StbM_GetCurrentTime(StbM_SynchronizedTimeBaseType param0, P2VAR(StbM_TimeStampType, AUTOMATIC, RTE_STBM_APPL_VAR) timeStampPtr, P2VAR(StbM_UserDataType, AUTOMATIC, RTE_STBM_APPL_VAR) userParam);
#define RTE_STOP_SEC_STBM_APPL_CODE
#include "MemMap.h" /* PRQA S 5087 */ /* MD_MSR_MemMap */

#define Rte_Call_GlobalTime_Slave_StbMSynchronizedTimeBase_GetCurrentTime(arg1, arg2) (StbM_GetCurrentTime((StbM_SynchronizedTimeBaseType)0, arg1, arg2)) /* PRQA S 3453 */ /* MD_MSR_FctLikeMacro */
```

Example of a RTE Connection missing

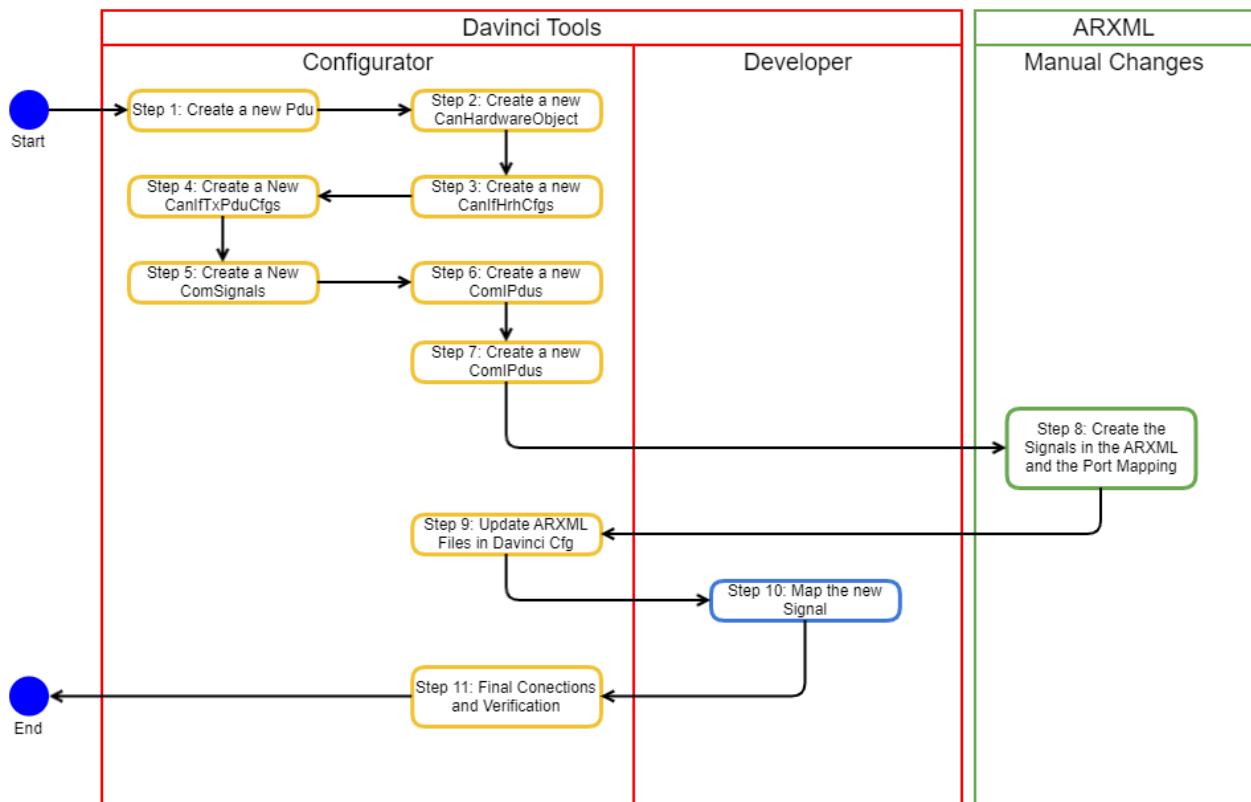
```
/*******************************************************************************
 * Rte_Call_C_P<>_O<> (unmapped) for synchronous C/S communication
 ******************************************************************************/
#define Rte_Call_GlobalTime_Slave_StbMSynchronizedTimeBase_GetCurrentTime(arg1, arg2) (RTE_E_UNCONNECTED) /* PRQA S 3453 */ /* MD_MSR_FctLikeMacro */
#define Rte_Call_GlobalTime_Slave_StbMSynchronizedTimeBase_GetRateDeviation(arg1) (RTE_E_UNCONNECTED) /* PRQA S 3453 */ /* MD_MSR_FctLikeMacro */
#define Rte_Call_GlobalTime_Slave_StbMSynchronizedTimeBase_GetTimeBaseStatus(arg1, arg2) (RTE_E_UNCONNECTED) /* PRQA S 3453 */ /* MD_MSR_FctLikeMacro */
#define Rte_Call_GlobalTime_Slave_StbMSynchronizedTimeBase_GetTimeLeap(arg1) (RTE_E_UNCONNECTED) /* PRQA S 3453 */ /* MD_MSR_FctLikeMacro */
```

9.7 Troubleshooting

10 8_ Create a New Rx CAN Signal

- [Workflow Diagram](#)(see page 101)
- [Prerequisites](#)(see page 102)
- [Steps on Davinci Configurator](#)(see page 102)
 - [Step 1: Create a new Pdu](#)(see page 102)
 - [Step 2: Create a new CanHardwareObject](#)(see page 105)
 - [Step 3: Create a new CanIfHrhCfgs](#)(see page 108)
 - [Step 4: Create a New CanIfTxPduCfgs](#)(see page 110)
 - [Step 5: Create a New ComSignals](#)(see page 113)
 - [Step 6: Create a new ComIPdus](#)(see page 116)
 - [Step 7: Create a New PduRRoutingPath](#)(see page 119)
- [Steps to be performed directly on the ARXML File](#)(see page 122)
 - [Step 8: Create the Signals in the ARXML and the Port Mapping](#)(see page 122)
 - [Changes on the ECU System Extract or Communication System Extract.](#)(see page 122)
 - [Changes in the SWC that will use the CAN signals](#)(see page 125)
- [Steps on Davinci Configurator](#)(see page 126)
 - [Step 9: Update ARXML Files in Davinci Cfg](#)(see page 126)
- [Steps on Davinci Developer](#)(see page 127)
 - [Step 10: Map the new Signal](#)(see page 127)
- [Steps on Davinci Configurator](#)(see page 128)
 - [Step 11: Final Conections and Verification](#)(see page 128)
- [Troubleshooting](#)(see page 128)
 - [UL \(Upper Layer\) Indication related issues:](#)(see page 128)

10.1 Workflow Diagram



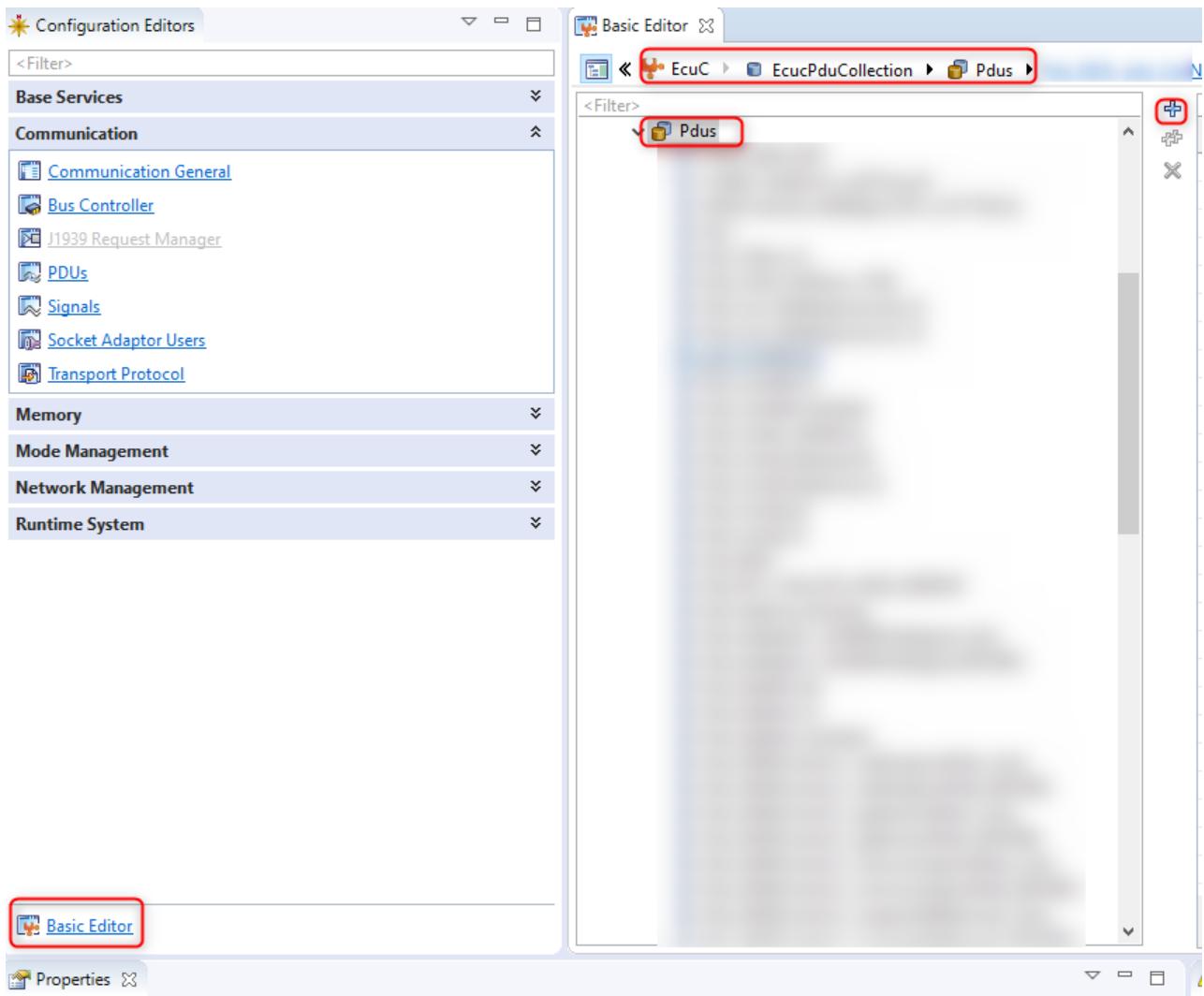
10.2 Prerequisites

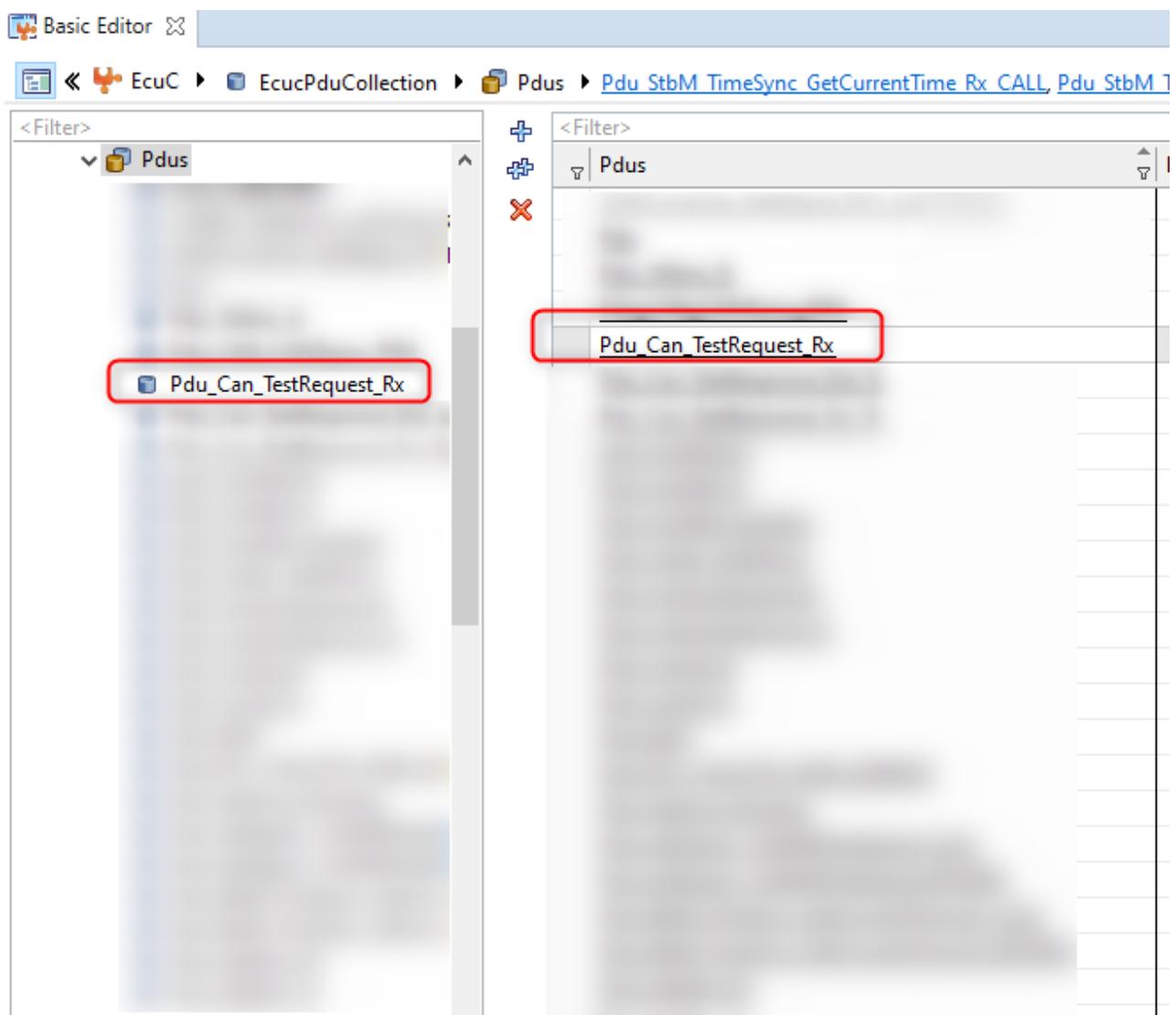
1. Make sure you have an up an running CAN configuration.
2. Make sure you are able to transmit a any CAN Message (not the one to be configure in this recipe).
3. Make sure to have a CANoe simulation or any other kind of verification tool.

10.3 Steps on Davinci Configurator

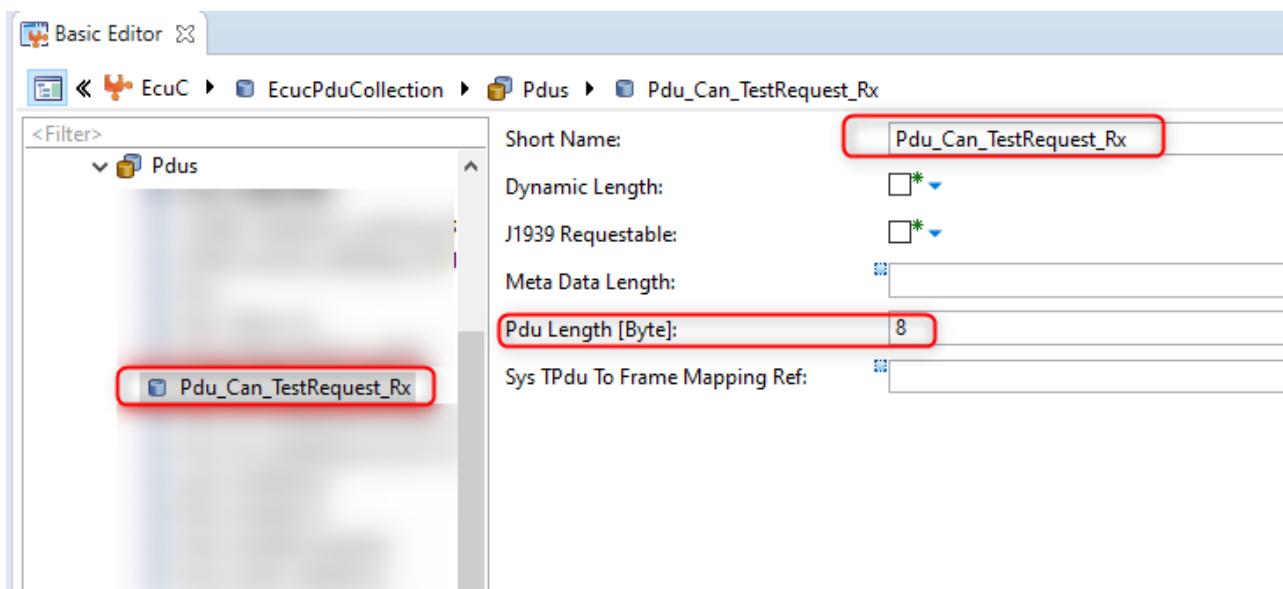
10.3.1 Step 1: Create a new Pdu

Go to Basic Editor → EcuC → EcucPduCollection → Pdu and Clic on the "+" sign.



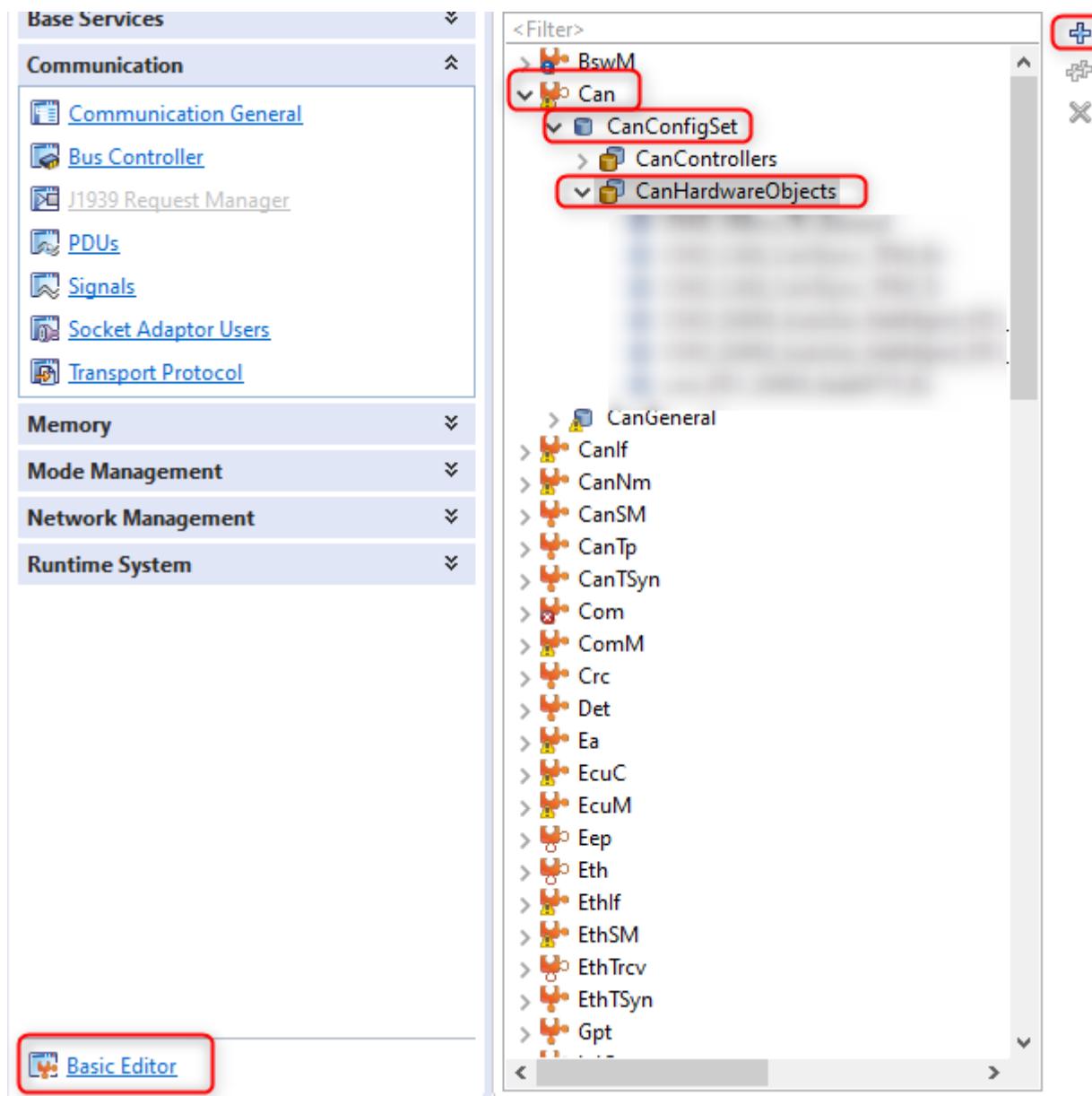


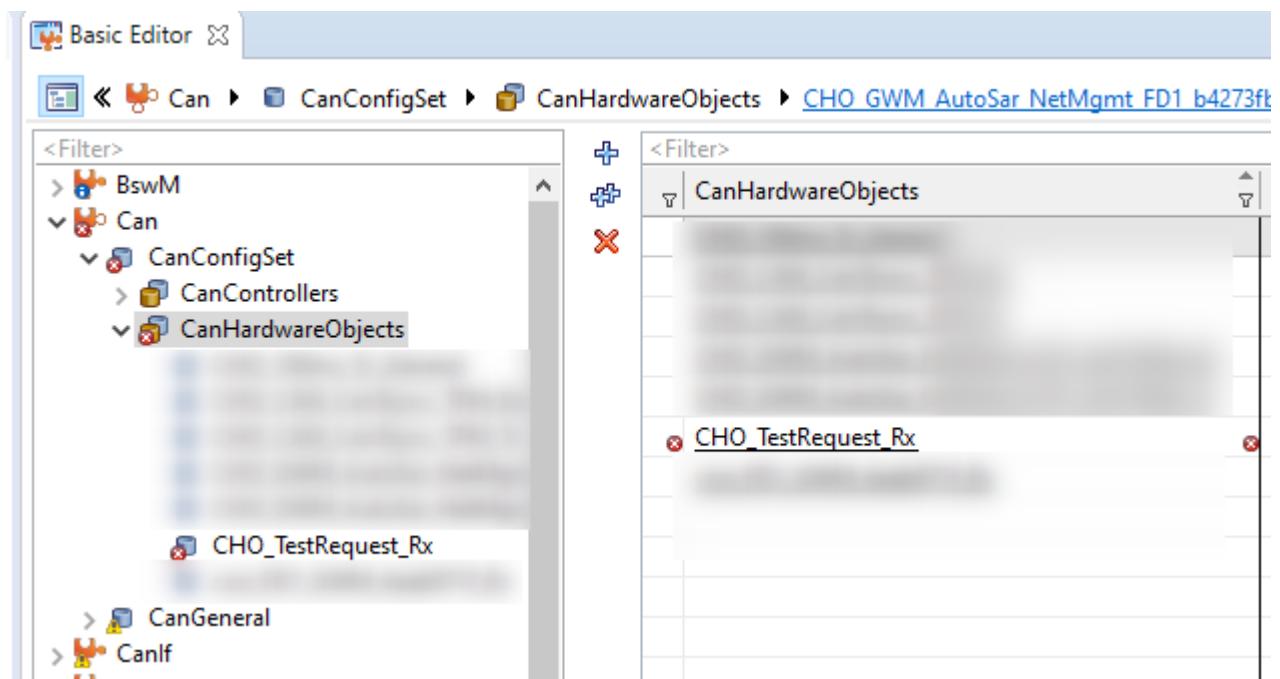
After creating the Pdu with a meaningful and unique name, configure the Data Length and any other field needed.



10.3.2 Step 2: Create a new CanHardwareObject

Go to Basic Editor → Can → CanConfigSet → CanHardwareObject, and click on the "+" sign.





After Creating the new CanHardwareObject with a meaningful and unique name, configure the following fields:

- Controller Ref: **(Use the previously configured CAN Controller)**
- Handle Type: **FULL**
- Id Value: **(ID from your application to be received)**
- Object Type: **RECEIVE**

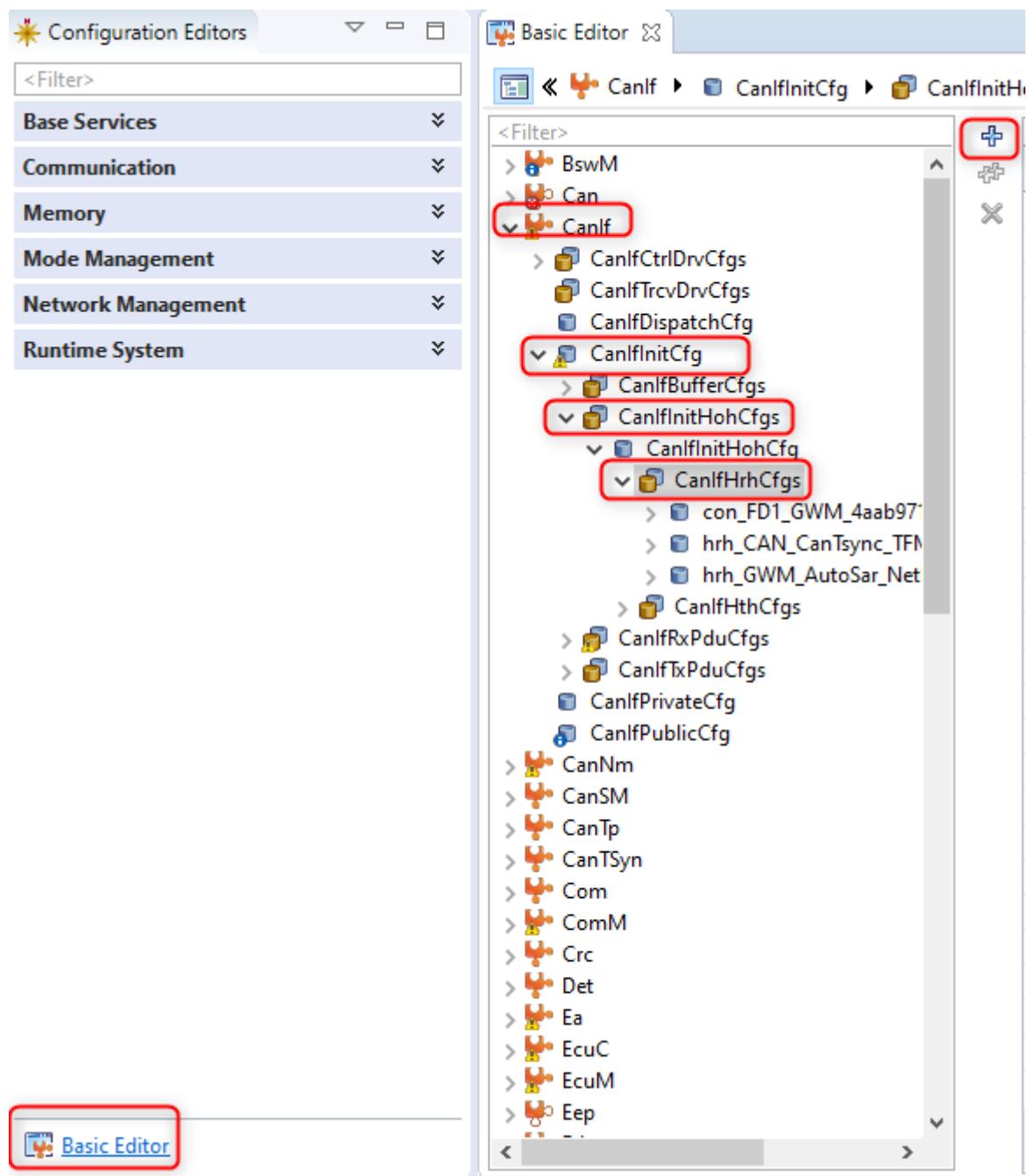
Short Name:	CHO_TestRequest_Rx
Common CAN Node:	
Controller Ref:	/ActiveEcuC/Can/CanConfigSet/FD1_CAN_GWM_6ab810d5 [...]
Enable Polling Processing:	<input type="checkbox"/> *
Fd Padding Value:	0x0 hex *
Handle Type:	FULL
Hardware FIFO:	<input type="checkbox"/> *
Id Type:	STANDARD
Id Value:	0x18 hex
Main Function RW Period Ref:	
Max Data Len:	64 dec
Multiplexed Transmission:	<input type="checkbox"/> *
Object Hw Handle:	3 dec
Object Hw Size:	1 dec
Object Id:	5 dec
Object Type:	RECEIVE

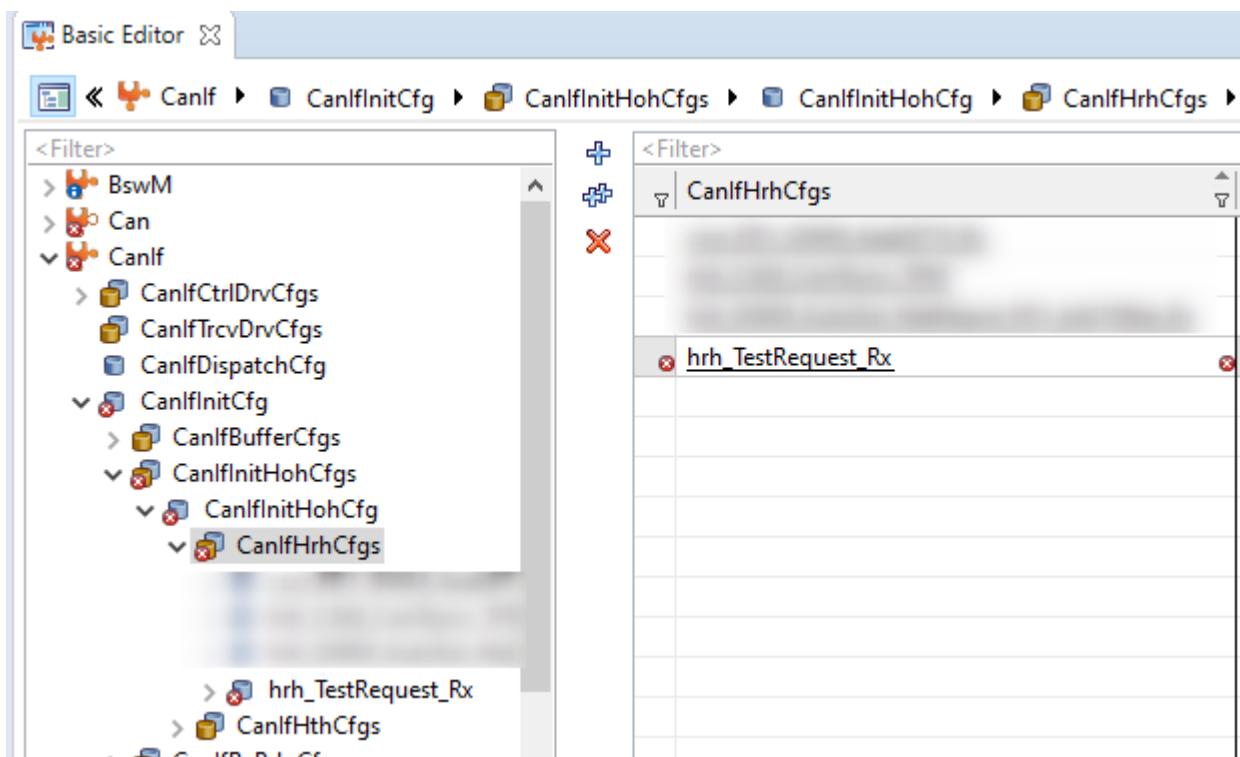
Filter Mask Ref

+ Use '+' to add parameters
✖

10.3.3 Step 3: Create a new CanIfHrhCfgs

Go to Basi Editor → CanIf → CanIfInitCfg → CanIfInitHohCfgs → CanIfInitHohCfg → CanIfHrhCfgs, and click the "+" sign.





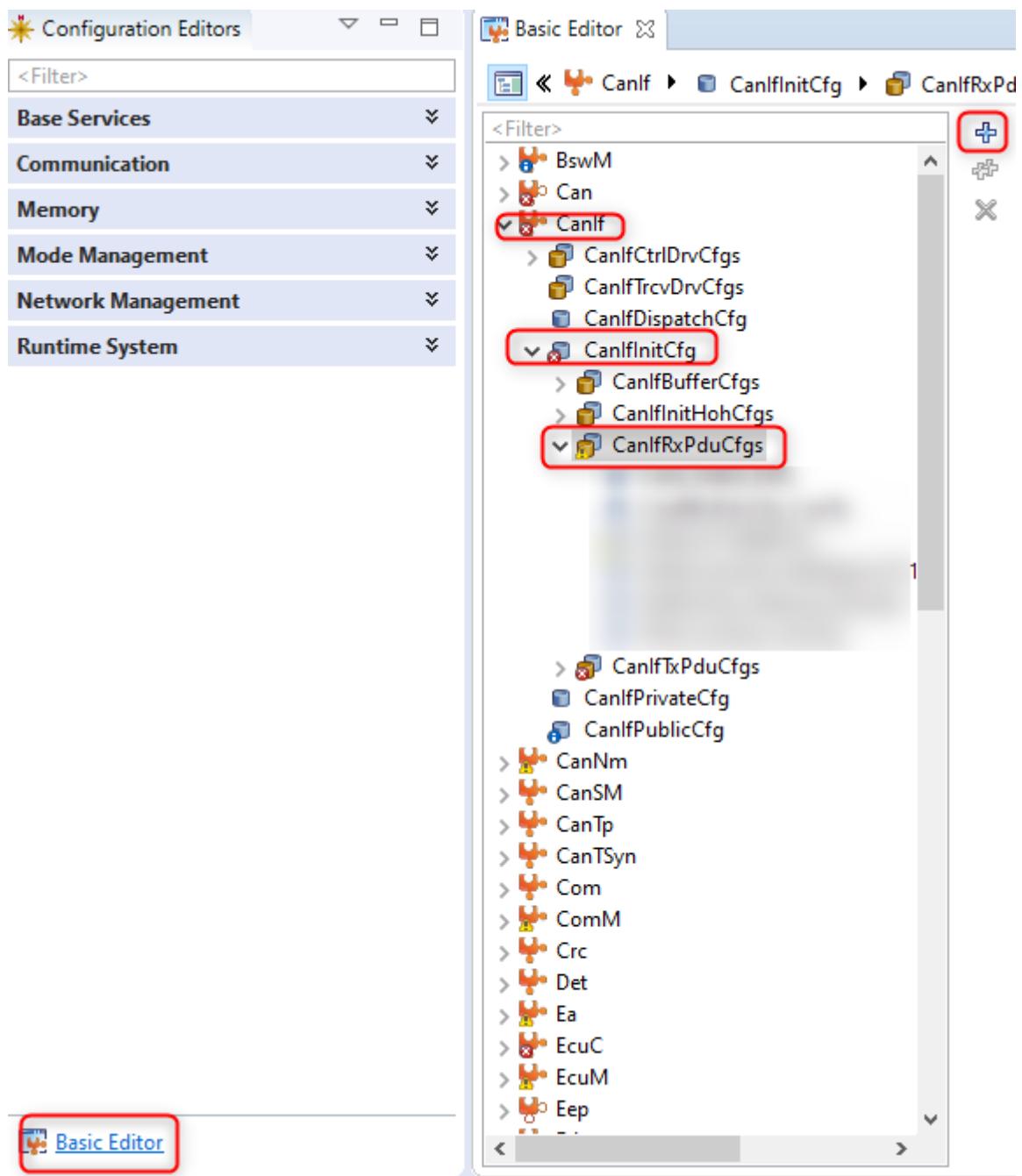
After creating a new CanIfHthCfgs with a meaningful and unique name, configure the following fields:

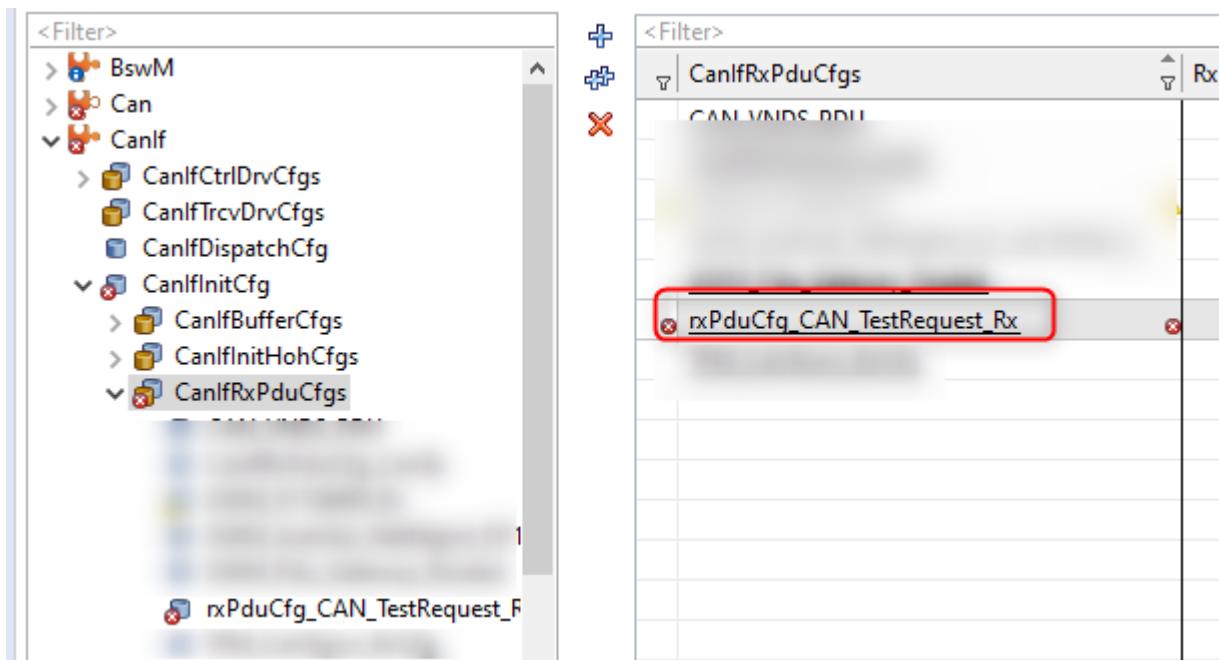
- Hrh Can Ctrl Id Ref: **(Use the previously created CAN Controller)**
- Hrh Id Sym Ref: **(Use the CanHardwareObject created in the previous step 2)**

Short Name:	<input type="text" value="hrh_TestRequest_Rx"/>	[...]
Hrh Can Ctrl Id Ref:	<input type="text" value="/ActiveEcuC/CanIf/CanIfCtrlDrvCfg_7d254554/FD1_CAN_GWM_6ab810d5"/>	[...]
Hrh Can Handle Type Ref:	<input type="text"/>	[...]
Hrh Id Sym Ref:	<input type="text" value="/ActiveEcuC/Can/CanConfigSet/CHO_TestRequest_Rx"/>	[...]
Hrh Software Filter:	<input checked="" type="checkbox"/> *	

10.3.4 Step 4: Create a New CanIfTxPduCfgs

Go to Basic Editor → CanIf → CanIfInitCfg → CanIfRxPduCfgs and click the "+" sign.





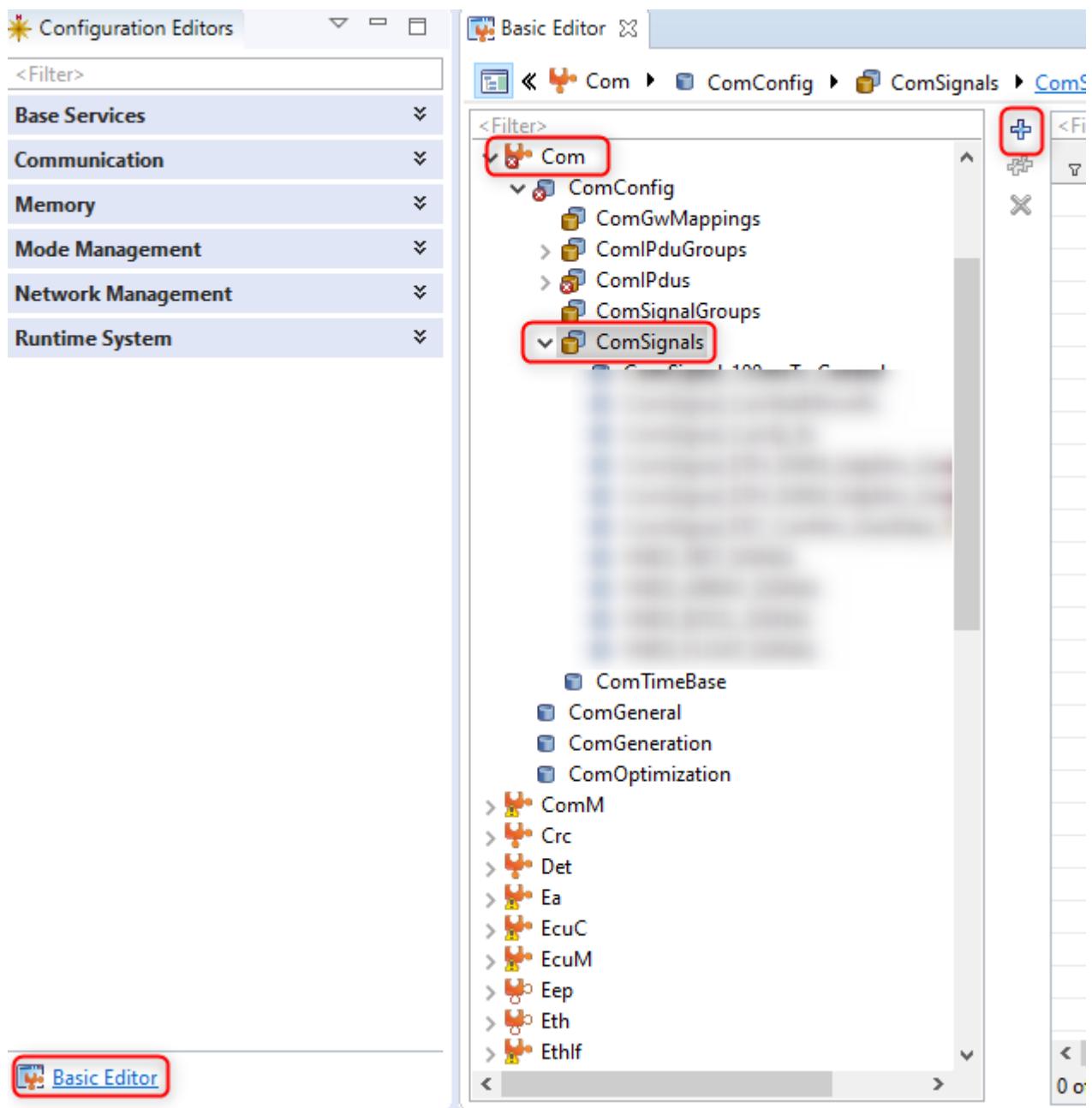
After creating a new CanIfTxPduCfgs with a meaningful and unique name, configure the following fields:

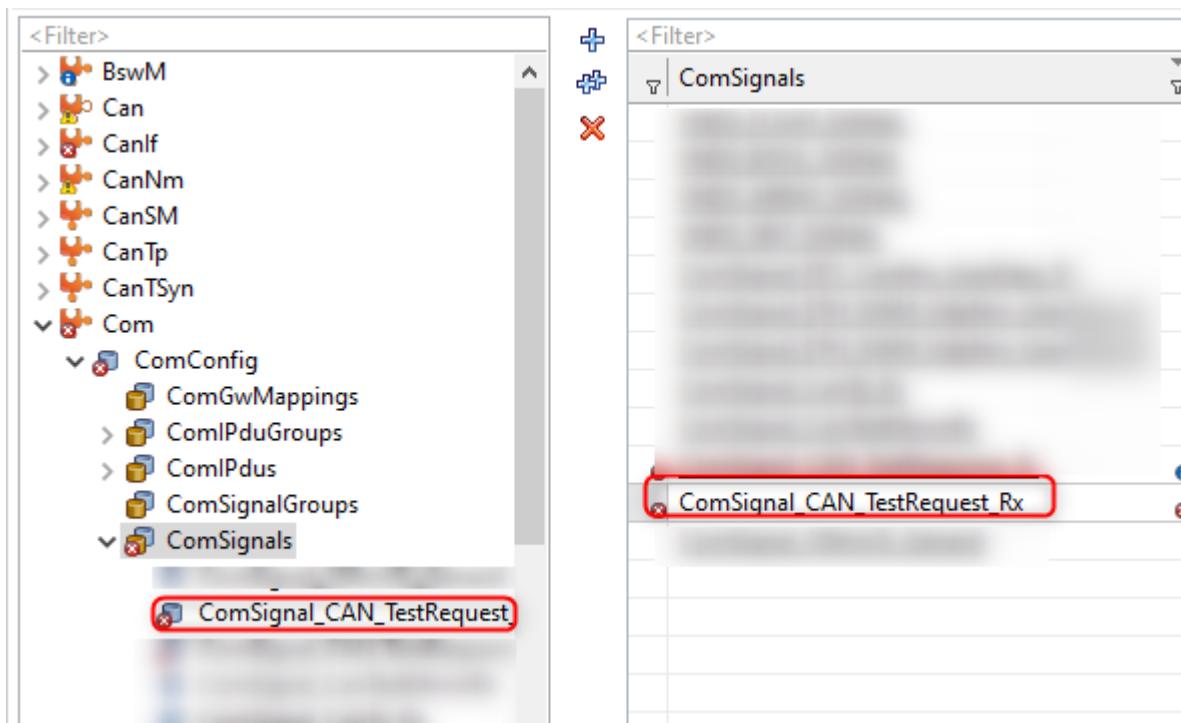
- Rx Pdu Can Id: **(Create one base on your requirements)**
- Rx Pdu Can Id Type: **STANDARD_CAN**
- Rx Pdu Dlc (Byte): **(Create one base on your requirements)**
- Rx Pdu Hrh Id Reference: **(Use the previously one created on step 3)**
- Rx Pdu Ref: **(Use the previously one created on step 1)**
- Rx Pdu User Tx Confirmation UL: **PDUR**

Short Name:	rxPduCfg_CAN_TestRequest_Rx
Rx Pdu Bsw Sch Excl Area Id Ref:	[...]
Rx Pdu Can Handle Type Locked:	<input checked="" type="checkbox"/>
Rx Pdu Can Id:	0x18
Rx Pdu Can Id Mask:	
Rx Pdu Can Id Type:	STANDARD_CAN
Rx Pdu Data Checksum Pdu:	<input checked="" type="checkbox"/>
Rx Pdu Dlc [Byte]:	8
Rx Pdu Dlc Check:	<input checked="" type="checkbox"/> *
Rx Pdu Hrh Id Ref:	/ActiveEcuC/CanIf/CanIfInitCfg/CanIfInitHohCfg/hrh_TestRequest_Rx
Rx Pdu Id:	0
Rx Pdu Read Data:	<input type="checkbox"/> *
Rx Pdu Read Notify Status:	<input type="checkbox"/> *
Rx Pdu Ref:	/ActiveEcuC/EcuC/EcucPduCollection/Pdu_Can_TestRequest_Rx
Rx Pdu Set Reception Mode Pdu:	<input checked="" type="checkbox"/>
Rx Pdu User Rx Indication Name:	PduR_CanIfRxIndication
Rx Pdu User Rx Indication Type:	
Rx Pdu User Rx Indication UL:	PDUR

10.3.5 Step 5: Create a New ComSignals

Go to Basic Editor → Com → ComConfig → ComSignals and Click the "+" sign





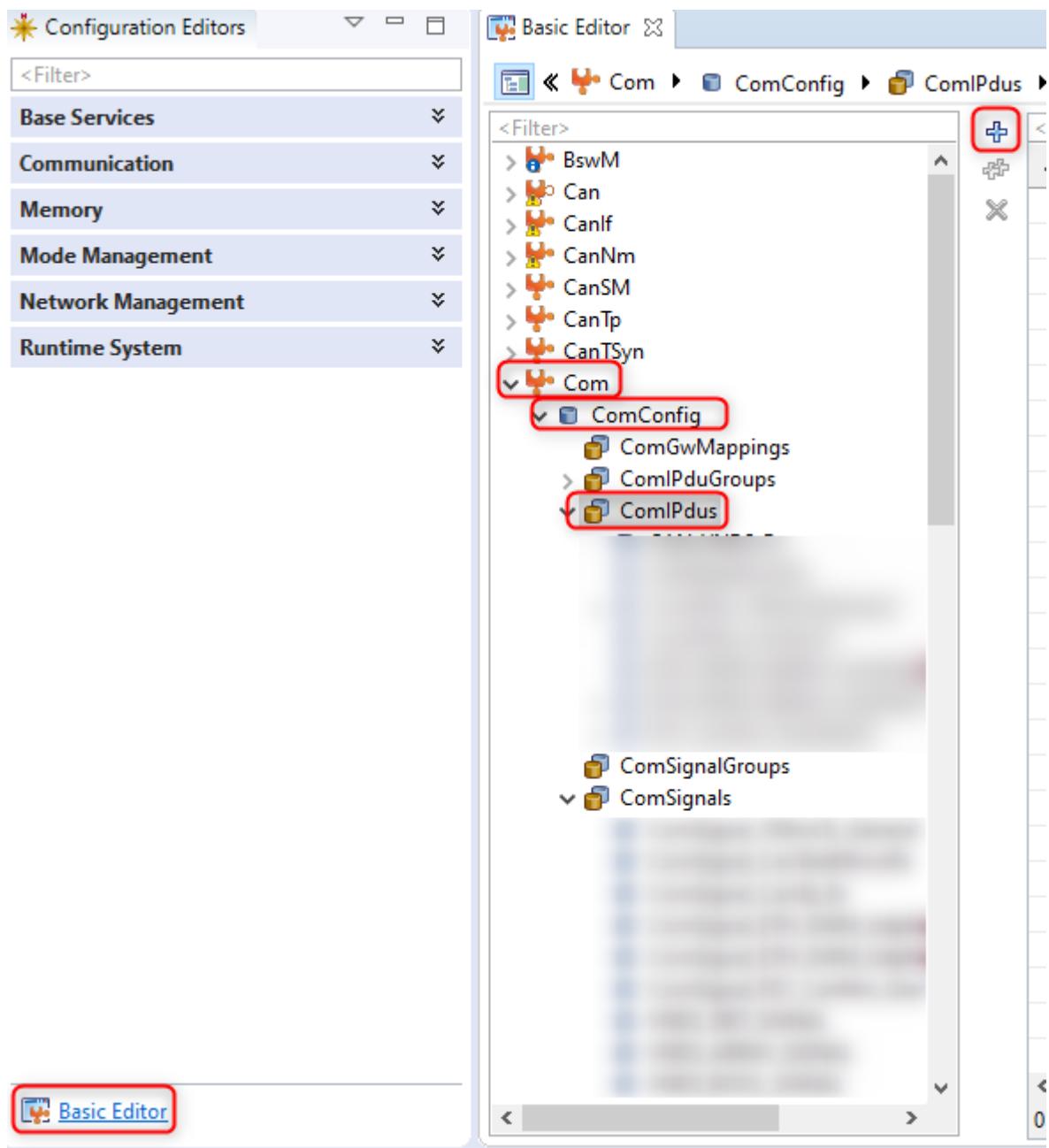
After creating the ComSignals with a meaningful and unique name, configure the following fields:

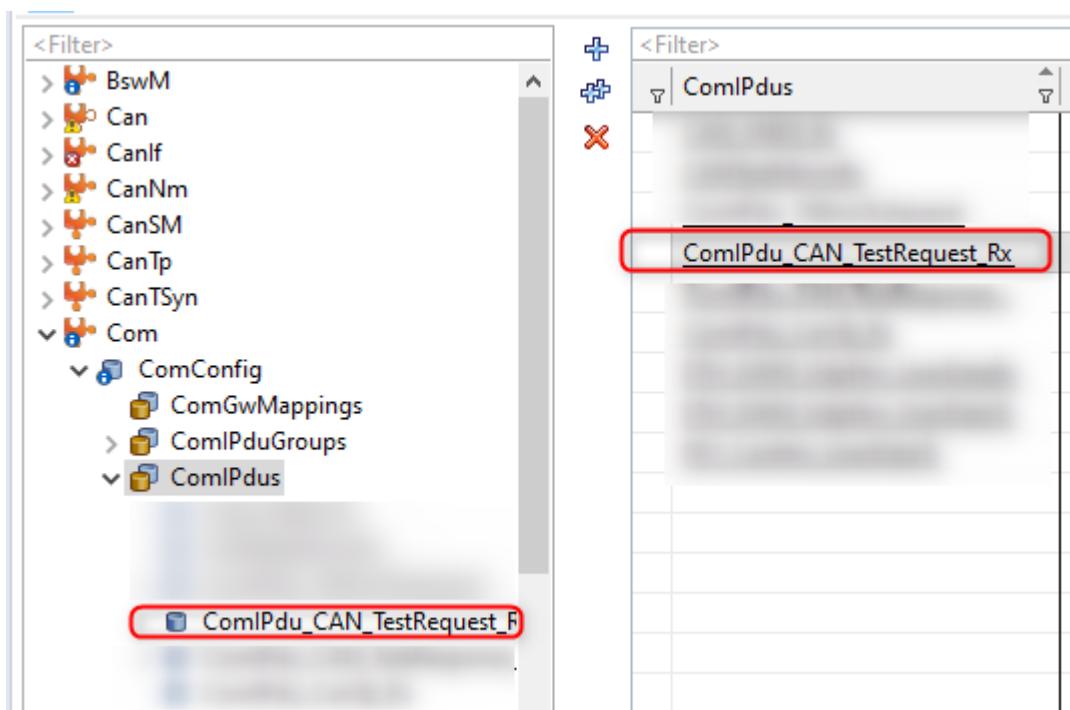
- Notification: **(Right click in the downward arrow and select "Create Parameter")**
- Rx Data Timeout Action: **NONE**
- Signal Access: **ACCESS_NEEDED_BY_SWC_OR_COM**
- Signal Endianess: **OPAQUE**.
- Signal Init Value: **0 0 0 0 0 0 0 0**
- Signal Length (Byte): **8**
- SignalType: **UINT8_N**

Short Name:	ComSignal_CAN_TestRequest_Rx	▼
Bit Position:	0	dec *▼
Bit Size [Bit]:	8	dec ▼
Data Invalid Action:		▼
Error Notification:		▼
First Timeout [ms]:		dec ▼
Handle Id:	0	dec *▼
Invalid Notification:		▼
Notification:	Rte_COMCbk_ComSignal_CAN_TestRequest_Rx	▼
Rx Data Timeout Action:	NONE	*▼
Rx Data Timeout Substitution Value:		▼
Signal Access:	ACCESS_NEEDED_BY_SWC_OR_COM	▼
Signal Data Invalid Value:		▼
Signal Endianness:	OPAQUE	▼
Signal Init Value:	00000000	▼
Signal Length [Byte]:	8	dec ▼
Signal Type:	UINT8_N	▼
System Template System Signal Ref:	/TimeFollower/Systems/Signals/Pdu_Can_TestRequest_Rx/Data_Request_TFM_Rx	[..] ▼
Timeout [ms]:		dec ▼
Timeout Notification:		▼
Transfer Property:		▼
Update Bit Position:		dec ▼

10.3.6 Step 6: Create a new ComIPdus

Go to Basic Editor → Com → ComConfig → ComIPdus and Click the "+" sign.





After the creating the ComIPdus with a meaningful and unique name, configure the following fields:

- IPdu Direction: **RECEIVE**
- Pdu Id Ref: **(Use the previously one created on step 1)**
- IPdu Signal Ref: **(Use the previously one created on step 5)**

Short Name:	ComIPdu_CAN_TestRequest_Rx
IPdu Callout:	
IPdu Cancellation Support:	<input type="checkbox"/> *
IPdu Direction:	RECEIVE
IPdu Handle Id:	2
IPdu Signal Processing:	DEFERRED
IPdu Trigger Transmit Callout:	
IPdu Type:	NORMAL
Pdu Id Ref:	/ActiveEcuC/EcuC/EcucPduCollection/Pdu_Can_TestRequest_Rx

IPdu Group Ref

+ Use '+' to add parameters

×

IPdu Signal Group Ref

+ Use '+' to add parameters

×

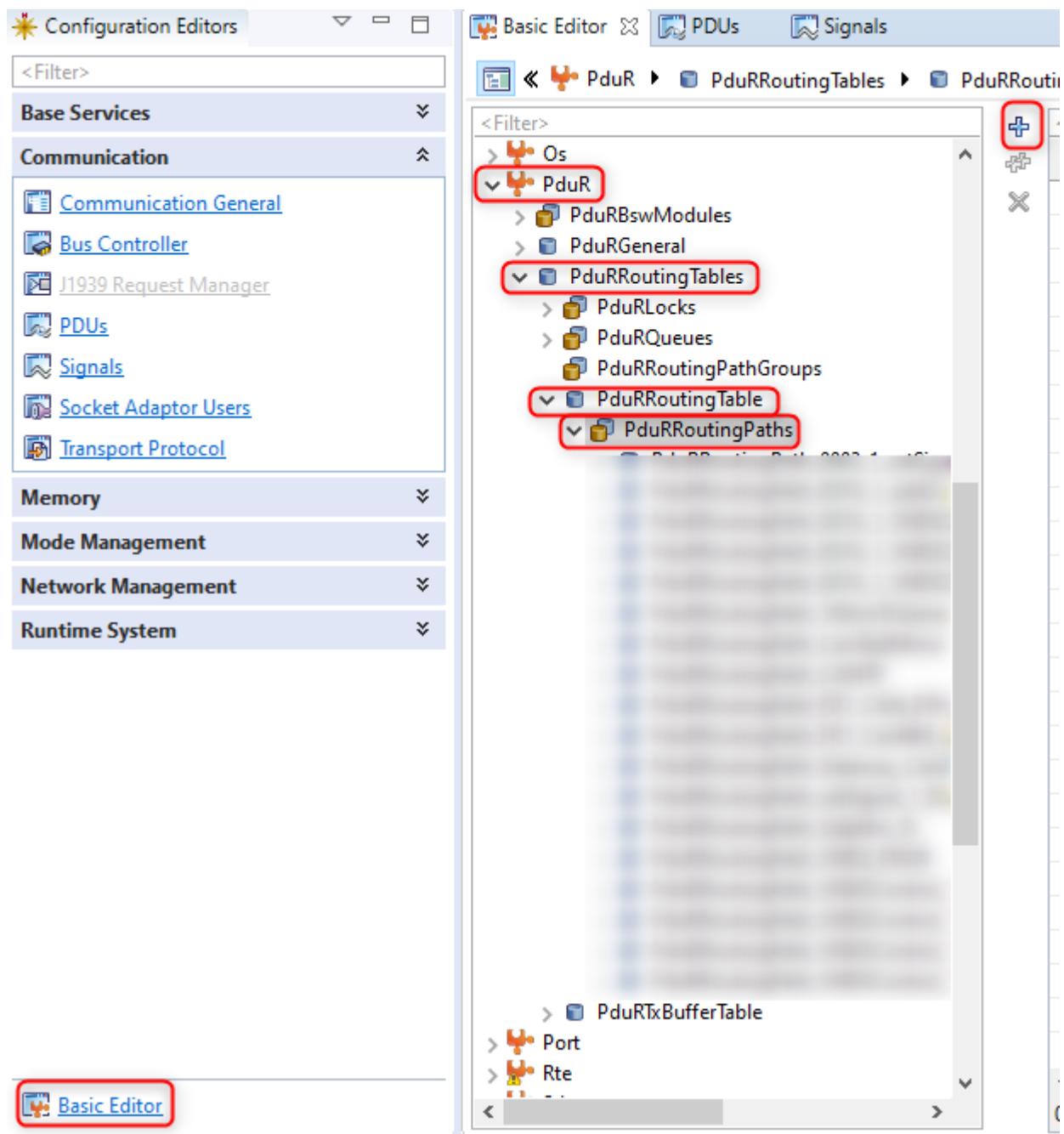
IPdu Signal Ref

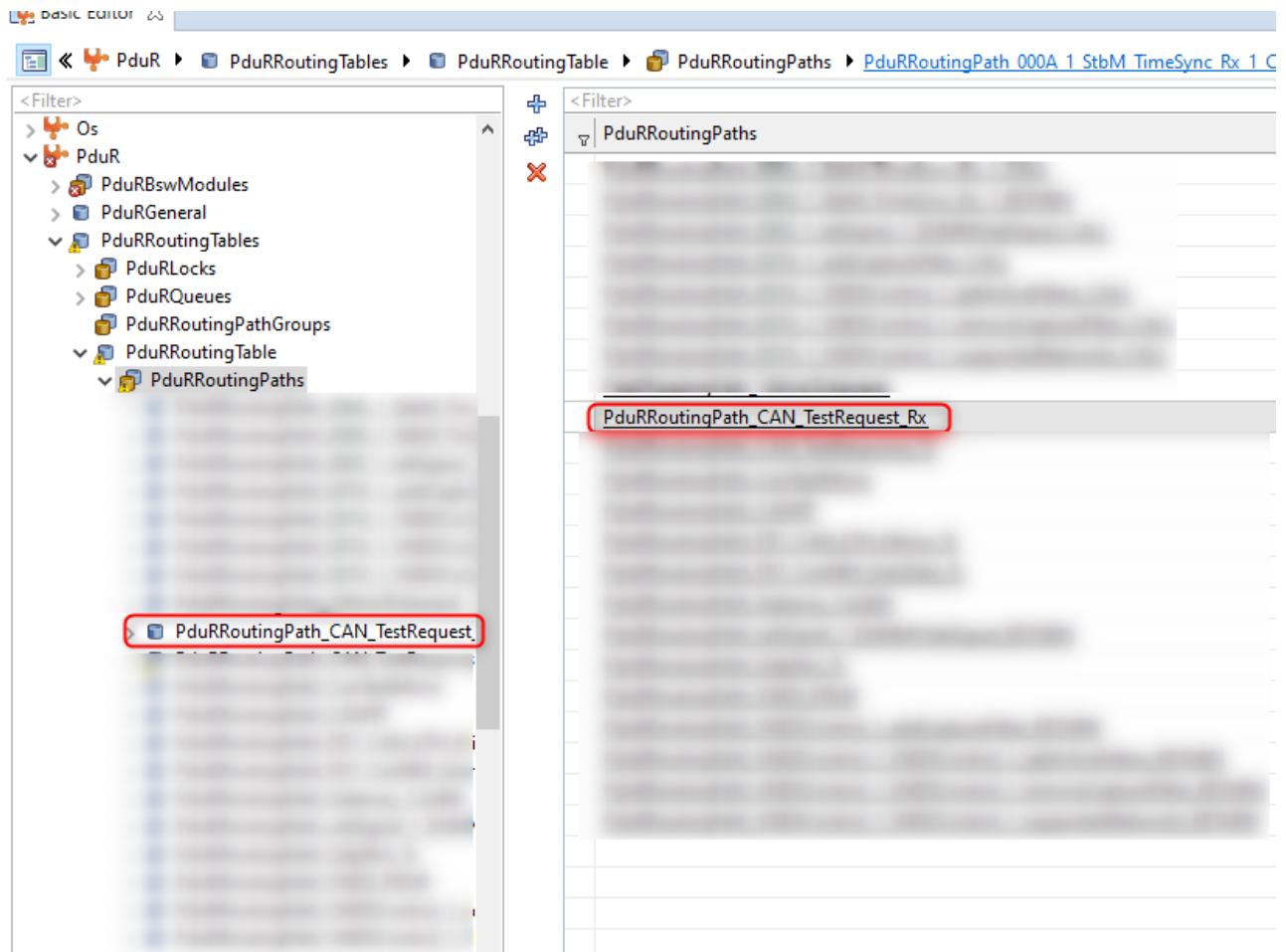
+ /ActiveEcuC/Com/ComConfig/ComSignal_CAN_TestRequest_Rx

×

10.3.7 Step 7: Create a New PduRRoutingPath

Go to Basi Editor → PduR → PduRRoutingTables → PduRRoutingTable → PduRRoutingPaths and Click the "+" sign.





After creating the PduRRoutingPath with a meaningful and unique name, a container called PduRSrcPdu_1 shall be created please configure the following:

- Src Pdu Ref: **(Use the previously one created on step 1)**

Short Name:	<input type="text" value="PduRSrcPdu_2"/>	▼
Source Pdu Handle Id:	<input type="text" value="6"/> dec	▼
Src Pdu Direction:	<input type="text" value="RECEIVE"/>	▼
Src Pdu PduRBswModules Ref:	<input type="text" value="/ActiveEcuC/PduR/CanIf"/> [...] ▼	
Src Pdu Ref:	<input type="text" value="/ActiveEcuC/EcuC/EcucPduCollection/Pdu_Can_TestRequest_Rx"/> [...] ▼	

Then after expand the PduRDestPdu container of the created PduRRoutingPath and configure the following:

- Dest Pdu Ref: **(Use the previously one created on step 1 with the)**
- Pdu Len Handling Strategy: **IGNORE**

Short Name:	PduRDestPdu_2
Dest Pdu Data Processing:	IMMEDIATE
Dest Pdu Data Provision:	
Dest Pdu Direction:	RECEIVE
Dest Pdu Handle Id:	2
Dest Pdu PduRBSwModules Ref:	/ActiveEcuC/PduR/Com
Dest Pdu Queue Depth:	
Dest Pdu Ref:	/ActiveEcuC/EcuC/EcucPduCollection/Pdu_Can_TestRequest_Rx
Dest Pdu Routing Type:	API_FORWARDING
Dest Tx Buffer Ref:	
Pdu Length Handling Strategy:	IGNORE
Queue Ref:	
Tp Threshold:	
Transmission Confirmation:	<input type="checkbox"/>

10.4 Steps to be performed directly on the ARXML File

10.4.1 Step 8: Create the Signals in the ARXML and the Port Mapping

Add the following ARXML blocks with the proper Short name and Paths in the Application Software Component which will handle the Signal transmission.

Please Consider

The following are just examples and templates, you may need to add more "ARXML Sections" to make this one work.

i.e:

In the following code snapshot some Port references are used, these port nee to be created elsewhere (not in the scope of this Recipe)

10.4.1.1 Changes on the ECU System Extract or Communication System Extract.

```
<CAN-FRAME-TRIGGERING UUID="BE91C722-B114-4149-89CA-158FBBEE014D">
  <SHORT-NAME>CF_TFM_Test_Request_Rx</SHORT-NAME>
  <FRAME-PORT-REFS>
    <FRAME-PORT-REF DEST="FRAME-PORT">/Topology/HardwareComponents/GWM/con_FD1_GWM/
      framePort_CANRx_Test_Request</FRAME-PORT-REF>
    </FRAME-PORT-REFS>
    <FRAME-REF DEST="CAN-FRAME">/Communication/Frames/CF_TFM_Test_Request_Rx</FRAME-
      REF>
    <CAN-ADDRESSING-MODE>STANDARD</CAN-ADDRESSING-MODE>
    <IDENTIFIER>24</IDENTIFIER>
  </CAN-FRAME-TRIGGERING>
```

```
<I-SIGNAL-TRIGGERING UUID="BEC3138A-2E7E-4CD2-8B31-71408F66D6C2">
  <SHORT-NAME>isigTrig_CANRx_Test_Request</SHORT-NAME>
  <I-SIGNAL-PORT-REFS>
    <I-SIGNAL-PORT-REF DEST="I-SIGNAL-PORT">/Topology/HardwareComponents/GWM/
con_FD1_GWM/isignalPort_CANRx_Test_Request</I-SIGNAL-PORT-REF>
  </I-SIGNAL-PORT-REFS>
    <I-SIGNAL-REF DEST="I-SIGNAL">/TimeFollower/Systems/Signals/Data_Request_TFM_Rx</I-
SIGNAL-REF>
  </I-SIGNAL-TRIGGERING>
```

```
<PDU-TRIGGERING UUID="F1D29002-45E9-4820-BAB2-CC49BF70A931">
  <SHORT-
NAME>FD1_CAN_FD1_CAN_FD1_CAN_FD1_CAN_FD1_CAN_FD1_CAN_FD1_CAN_D_F62AF3D8_8C18_
4779_A0C0_9BD39632CEB5_VFrame_VFrame</SHORT-NAME>
  <I-PDU-PORT-REFS>
    <I-PDU-PORT-REF DEST="I-PDU-PORT">/Topology/HardwareComponents/GWM/con_FD1_GWM/
ipduPort_CANRx_Test_Request</I-PDU-PORT-REF>
  </I-PDU-PORT-REFS>
    <I-PDU-REF DEST="I-SIGNAL-I-PDU">/TimeFollower/Systems/Signals/
Pdu_Can_TestRequest_Rx</I-PDU-REF>
  </PDU-TRIGGERING>
```

```
<FRAME-PORT UUID="C0769E8C-687A-49B3-88E1-4207076C5557">
  <SHORT-NAME>framePort_CANRx_Test_Request</SHORT-NAME>
  <COMMUNICATION-DIRECTION>IN</COMMUNICATION-DIRECTION>
</FRAME-PORT>
```

```
<I-PDU-PORT UUID="8EE13968-7E3B-4486-9160-6F9573C7A235">
  <SHORT-NAME>ipduPort_CANRx_Test_Request</SHORT-NAME>
  <COMMUNICATION-DIRECTION>IN</COMMUNICATION-DIRECTION>
</I-PDU-PORT>
```

```
<CAN-FRAME UUID="1988BD2A-B059-4057-8A9B-A3042F45F3F7">
  <SHORT-NAME>CF_TFM_Test_Request_Rx</SHORT-NAME>
</CAN-FRAME>
```

```
<FIBEX-ELEMENT-REF-CONDITIONAL>
  <FIBEX-ELEMENT-REF DEST="I-SIGNAL">/TimeFollower/Systems/Signals/
Data_Request_TFM_Rx</FIBEX-ELEMENT-REF>
</FIBEX-ELEMENT-REF-CONDITIONAL>
<FIBEX-ELEMENT-REF-CONDITIONAL>
  <FIBEX-ELEMENT-REF DEST="I-SIGNAL-I-PDU">/TimeFollower/Systems/Signals/
Pdu_Can_TestRequest_Rx</FIBEX-ELEMENT-REF>
```

```
</FIBEX-ELEMENT-REF-CONDITIONAL>
```

```
<SENDER-RECEIVER-TO-SIGNAL-MAPPING>
  <DATA-ELEMENT-IREF>
    <CONTEXT-PORT-REF DEST="R-PORT-PROTOTYPE">/ECUCompositionTypes/ECU_Composition/
Test_Option_Rq</CONTEXT-PORT-REF>
    <TARGET-DATA-PROTOTYPE-REF DEST="VARIABLE-DATA-PROTOTYPE">/TimeFollower/
Interfaces/Test_Option_Rq/Test_Option_Rq</TARGET-DATA-PROTOTYPE-REF>
  </DATA-ELEMENT-IREF>
  <SYSTEM-SIGNAL-REF DEST="SYSTEM-SIGNAL">/TimeFollower/Systems/Signals/
sysSignal_Data_Request_TFM_Rx</SYSTEM-SIGNAL-REF>
</SENDER-RECEIVER-TO-SIGNAL-MAPPING>
```

```
<R-PORT-PROTOTYPE UUID="2EAEA4D2-4362-4186-BA92-20AD79DC9968">
  <SHORT-NAME>Test_Option_Rq</SHORT-NAME>
  <ADMIN-DATA>
    <SDGS>
      <SDG GID="DV:DEV">
        <SD GID="DV:ImportModePreset">Keep</SD>
      </SDG>
    </SDGS>
  </ADMIN-DATA>
  <REQUIRED-COM-SPECS>
    <NONQUEUED-RECEIVER-COM-SPEC>
      <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/TimeFollower/Interfaces/
Test_Option_Rq/Test_Option_Rq</DATA-ELEMENT-REF>
      <ALIVE-TIMEOUT>0</ALIVE-TIMEOUT>
      <ENABLE-UPDATE>false</ENABLE-UPDATE>
      <FILTER>
        <DATA-FILTER-TYPE>ALWAYS</DATA-FILTER-TYPE>
      </FILTER>
      <HANDLE-NEVER-RECEIVED>false</HANDLE-NEVER-RECEIVED>
      <INIT-VALUE>
        <ARRAY-VALUE-SPECIFICATION>
          <ELEMENTS>
            <NUMERICAL-VALUE-SPECIFICATION>
              <VALUE>0</VALUE>
            </NUMERICAL-VALUE-SPECIFICATION>
            <NUMERICAL-VALUE-SPECIFICATION>
              <VALUE>0</VALUE>
            </NUMERICAL-VALUE-SPECIFICATION>
            <NUMERICAL-VALUE-SPECIFICATION>
              <VALUE>0</VALUE>
            </NUMERICAL-VALUE-SPECIFICATION>
            <NUMERICAL-VALUE-SPECIFICATION>
              <VALUE>0</VALUE>
            </NUMERICAL-VALUE-SPECIFICATION>
            <NUMERICAL-VALUE-SPECIFICATION>
              <VALUE>0</VALUE>
            </NUMERICAL-VALUE-SPECIFICATION>
          </ELEMENTS>
        </ARRAY-VALUE-SPECIFICATION>
      </INIT-VALUE>
    </NONQUEUED-RECEIVER-COM-SPEC>
  </REQUIRED-COM-SPECS>
</R-PORT-PROTOTYPE>
```

```

</NUMERICAL-VALUE-SPECIFICATION>
<NUMERICAL-VALUE-SPECIFICATION>
  <VALUE>0</VALUE>
</NUMERICAL-VALUE-SPECIFICATION>
<NUMERICAL-VALUE-SPECIFICATION>
  <VALUE>0</VALUE>
</NUMERICAL-VALUE-SPECIFICATION>
<NUMERICAL-VALUE-SPECIFICATION>
  <VALUE>0</VALUE>
</NUMERICAL-VALUE-SPECIFICATION>
</ELEMENTS>
</ARRAY-VALUE-SPECIFICATION>
</INIT-VALUE>
</NONQUEUED-RECEIVER-COM-SPEC>
</REQUIRED-COM-SPECS>
<REQUIRED-INTERFACE-TREF DEST=""SENDER-RECEIVER-INTERFACE">/TimeFollower/Interfaces/
Test_Option_Rq</REQUIRED-INTERFACE-TREF>
</R-PORT-PROTOTYPE>

```

```

<DELEGATION-SW-CONNECTOR UUID=""285100A0-E353-4820-9CB1-AA6B60EC3220"">
  <SHORT-NAME>Test_Option_Rq_FNV4Time_R_Test_Option_Rq</SHORT-NAME>
  <INNER-PORT-IREF>
    <R-PORT-IN-COMPOSITION-INSTANCE-REF>
      <CONTEXT-COMPONENT-REF DEST=""SW-COMPONENT-PROTOTYPE">/ECUCompositionTypes/
ECU_Composition/FNV4Time</CONTEXT-COMPONENT-REF>
      <TARGET-R-PORT-REF DEST=""R-PORT-PROTOTYPE">/TimeFollower/FNV4_SWCType/
R_Test_Option_Rq</TARGET-R-PORT-REF>
    </R-PORT-IN-COMPOSITION-INSTANCE-REF>
  </INNER-PORT-IREF>
  <OUTER-PORT-REF DEST=""R-PORT-PROTOTYPE">/ECUCompositionTypes/ECU_Composition/
Test_Option_Rq</OUTER-PORT-REF>
</DELEGATION-SW-CONNECTOR>

```

10.4.1.2 Changes in the SWC that will use the CAN signals

```

<I-SIGNAL UUID=""514877EF-B893-40E8-A86A-0C099353C53A"">
  <SHORT-NAME>Data_Request_TFM_Rx</SHORT-NAME>
  <DATA-TYPE-POLICY>LEGACY</DATA-TYPE-POLICY>
  <I-SIGNAL-TYPE>ARRAY</I-SIGNAL-TYPE>
  <LENGTH>8</LENGTH>
  <NETWORK-REPRESENTATION-PROPS>
    <SW-DATA-DEF-PROPS-VARIANTS>
      <SW-DATA-DEF-PROPS-CONDITIONAL>
        <BASE-TYPE-REF DEST=""SW-BASE-TYPE">/TimeFollower/ImplementationDataTypes/
uint8_8_none</BASE-TYPE-REF>
      </SW-DATA-DEF-PROPS-CONDITIONAL>
    </SW-DATA-DEF-PROPS-VARIANTS>
  </NETWORK-REPRESENTATION-PROPS>

```

```

<SYSTEM-SIGNAL-REF DEST="SYSTEM-SIGNAL">/TimeFollower/Systems/Signals/
sysSignal_Data_Request_TFM_Rx</SYSTEM-SIGNAL-REF>
</I-SIGNAL>
<I-SIGNAL-I-PDU UUID="D0AC5774-3653-4F80-BFDD-789FF85A8D20">
  <SHORT-NAME>Pdu_Can_TestRequest_Rx</SHORT-NAME>
  <LENGTH>8</LENGTH>
  <I-SIGNAL-TO-PDU-MAPPINGS>
    <I-SIGNAL-TO-I-PDU-MAPPING UUID="01D10CA6-25A9-4C78-B15F-7909D5F17202">
      <SHORT-NAME>Data_Request_TFM_Rx</SHORT-NAME>
      <I-SIGNAL-REF DEST="I-SIGNAL">/TimeFollower/Systems/Signals/
Data_Request_TFM_Rx</I-SIGNAL-REF>
      <PACKING-BYTE-ORDER>OPAQUE</PACKING-BYTE-ORDER>
      <START-POSITION>0</START-POSITION>
      <TRANSFER-PROPERTY>TRIGGERED</TRANSFER-PROPERTY>
    </I-SIGNAL-TO-I-PDU-MAPPING>
  </I-SIGNAL-TO-PDU-MAPPINGS>
  <UNUSED-BIT-PATTERN>0</UNUSED-BIT-PATTERN>
</I-SIGNAL-I-PDU>
<SYSTEM-SIGNAL UUID="9bf677cf-1503-4f38-8658-3d32aa785faD">
  <SHORT-NAME>sysSignal_Data_Request_TFM_Rx</SHORT-NAME>
  <DYNAMIC-LENGTH>false</DYNAMIC-LENGTH>
  <PHYSICAL-PROPS>
    <SW-DATA-DEF-PROPS-VARIANTS>
      <SW-DATA-DEF-PROPS-CONDITIONAL>
        <COMPU-METHOD-REF DEST="COMPU-METHOD">/TimeFollower/Compumethods/
CM_Data_Resp</COMPU-METHOD-REF>
        <DATA-CONSTR-REF DEST="DATA-CONSTR">/TimeFollower/DataConstraints/
DC_Data_Resp</DATA-CONSTR-REF>
      </SW-DATA-DEF-PROPS-CONDITIONAL>
    </SW-DATA-DEF-PROPS-VARIANTS>
  </PHYSICAL-PROPS>
</SYSTEM-SIGNAL>

```

10.5 Steps on Davinci Configurator

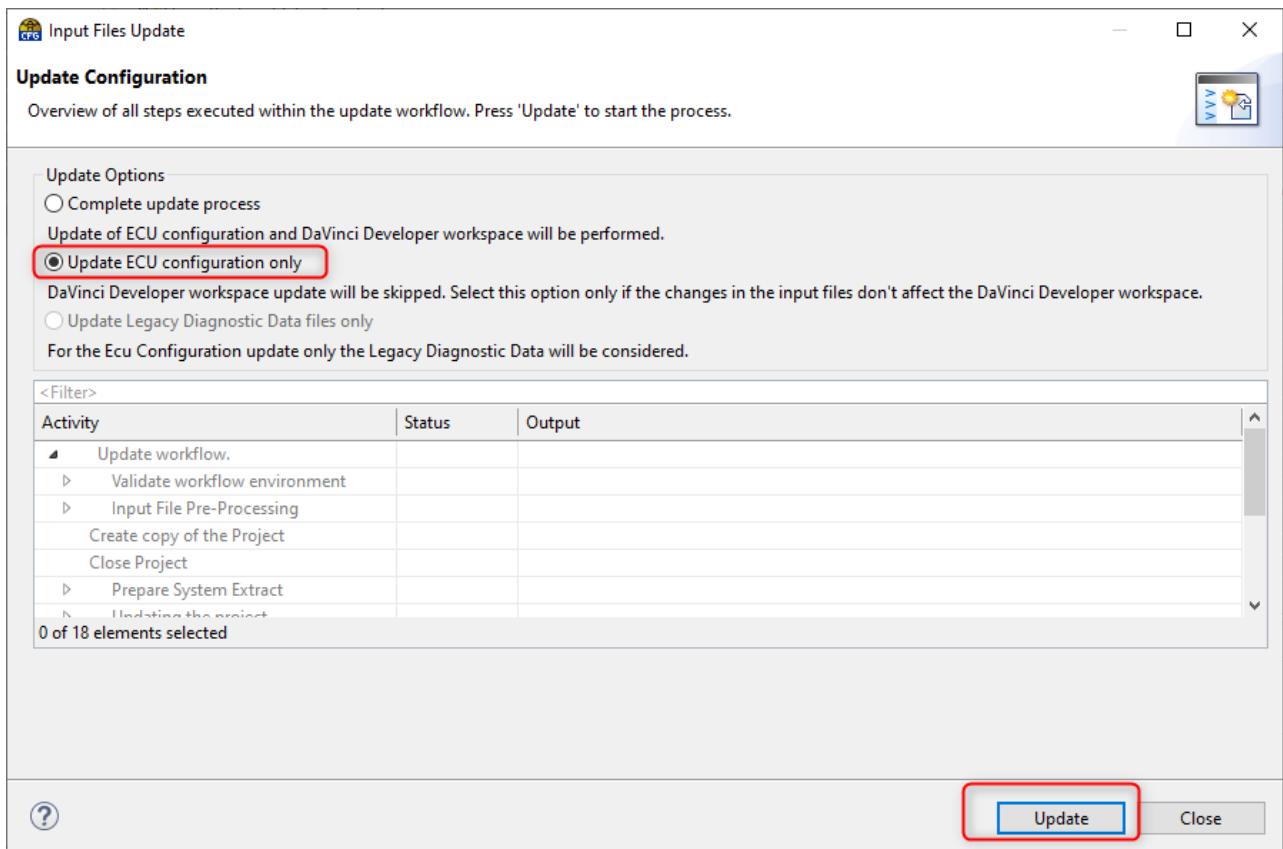
10.5.1 Step 9: Update ARXML Files in Davinci Cfg

⚠ Sync Needed

Please consider that after modifying (and saving) the ARXML file, an Update shall be need, please follow the recomended Updating process by clicking the Yellow ribbon in the Vector Davinci Configurator Tool.

Please make use to follow the Update show in the Following Images:





10.6 Steps on Davinci Developer

10.6.1 Step 10: Map the new Signal

After Step 9 the Mapping shall be done automatically, please verify that the mapping looks like the following image:

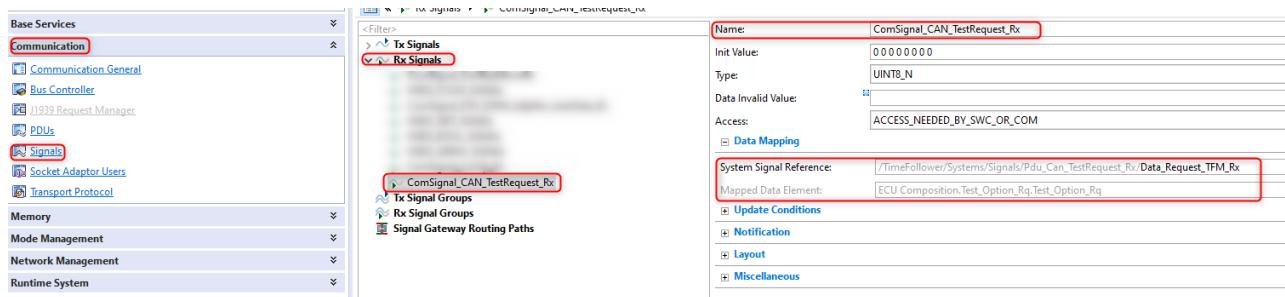
10.7 Steps on Davinci Configurator

10.7.1 Step 11: Final Connections and Verification

Using the Left panel move the following path:

Communication → Signals → Tx Signal → Select the ComSignals from step 6 → Data Mapping.

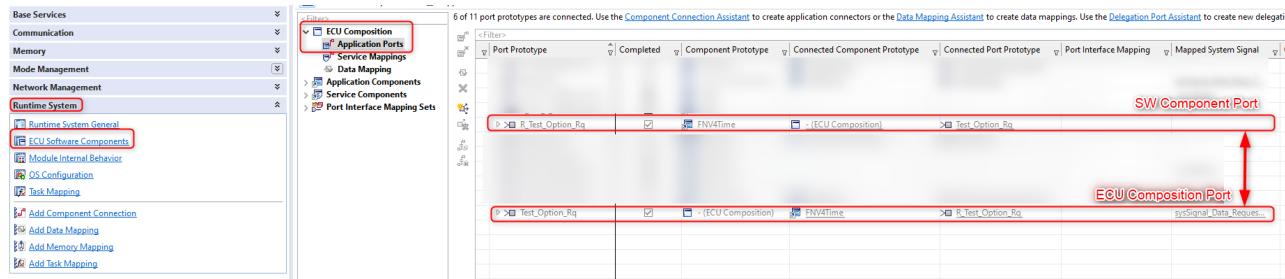
- System Signal Reference: (**Select the System Signal from Step 9**)
- Mapped Data Element: (**Shall be updated as well**)



Using the Left panel move the following path:

Runtime Systems → ECU Software Components → ECU Composition → Application Ports.

Make sure the System Signal is mapped to the correct port only on the ECU Composition Port:



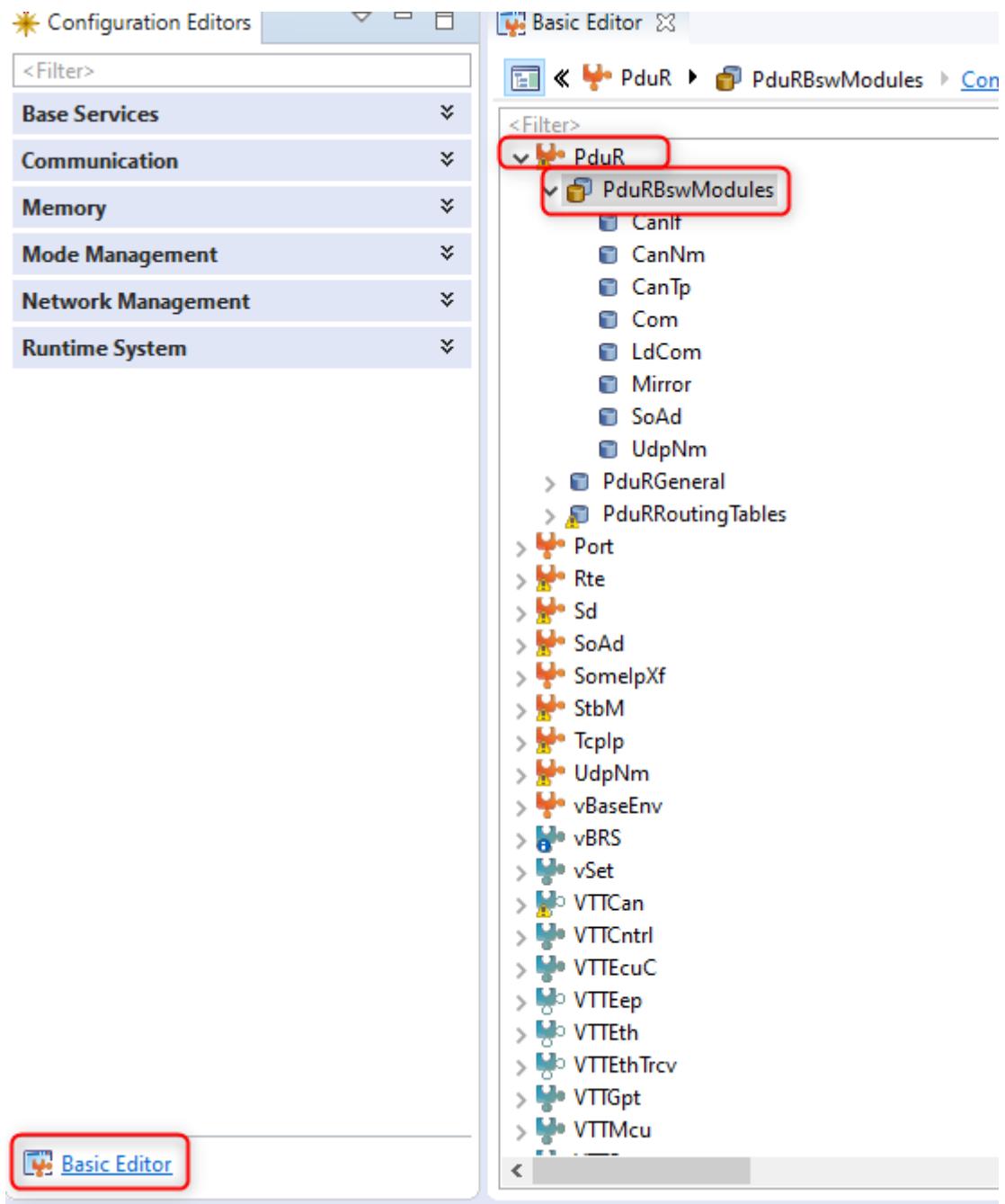
10.8 Troubleshooting

10.8.1 UL (Upper Layer) Indication related issues:

Some default configuration from Autosar Project don't provide any Tx Confirmation or Upper Module Configuration, hence some validation errors:

To modify this please refer to the following Path:

Basic Editor → PduR → PduRBswModules:



In here you can configure any missing Upper Layer or Tx Confirmation issues, like the image below:

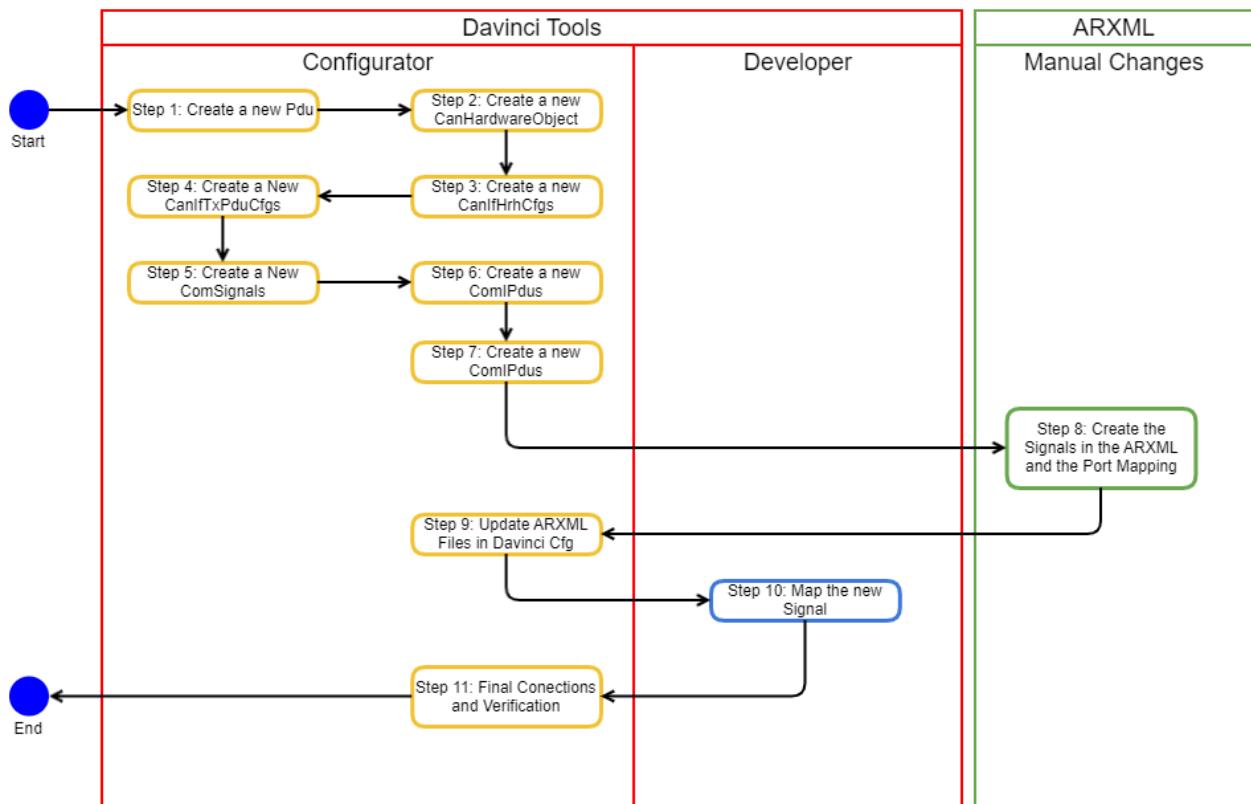
Short Name:	Com
Bsw Module Ref:	/ActiveEcuC/Com
Cancel Receive:	<input type="checkbox"/> *
Cancel Transmit:	<input type="checkbox"/> *
Change Parameter Request Api:	<input type="checkbox"/> *
Communication Interface:	<input checked="" type="checkbox"/>
Lower Module:	<input type="checkbox"/> *
Os Application Ref:	<input type="checkbox"/>
Retransmission:	<input type="checkbox"/>
Transport Protocol:	<input type="checkbox"/> *
Triggertransmit:	<input checked="" type="checkbox"/>
Tx Confirmation:	<input checked="" type="checkbox"/>
Upper Module:	<input checked="" type="checkbox"/>
Use Tag:	<input type="checkbox"/> *

Short Name:	CanIf
Bsw Module Ref:	/ActiveEcuC/CanIf
Cancel Receive:	<input type="checkbox"/> *
Cancel Transmit:	<input type="checkbox"/> *
Change Parameter Request Api:	<input type="checkbox"/> *
Communication Interface:	<input checked="" type="checkbox"/>
Lower Module:	<input checked="" type="checkbox"/>
Os Application Ref:	
Retransmission:	<input type="checkbox"/>
Transport Protocol:	<input type="checkbox"/> *
Triggertransmit:	<input type="checkbox"/> *
Tx Confirmation:	<input checked="" type="checkbox"/>
Upper Module:	<input type="checkbox"/> *
Use Tag:	<input type="checkbox"/> *

11 9_ Create a New Tx CAN Signal

- [Workflow Diagram](#)(see page 132)
- [Prerequisites](#)(see page 133)
- [Steps on Davinci Configurator](#)(see page 133)
 - Step 1: Create two new Pdu(see page 133)
 - Step 2: Create a new CanHardwareObject(see page 135)
 - Step 3: Create a new CanIfHthCfgs(see page 138)
 - Step 4: Create a New CanIfBufferCfgs(see page 140)
 - Step 5: Create a New CanIfTxPduCfgs(see page 142)
 - Step 6: Create a New ComSignals(see page 145)
 - Step 7: Create a new ComIPdus(see page 148)
 - Step 8: Create a New PduRRoutingPath(see page 152)
- [Steps to be performed directly on the ARXML File](#)(see page 155)
 - Step 9: Create the Signals in the ARXML and the Port Mapping(see page 155)
 - Changes on the ECU System Extract or Communication System Extract.(see page 155)
 - Changes in the SWC that will use the CAN signals(see page 158)
- [Steps on Davinci Configurator](#)(see page 159)
 - Step 10: Update ARXML Files in Davinci Cfg(see page 159)
- [Steps on Davinci Developer](#)(see page 160)
 - Step 11: Map the new Signal(see page 160)
- [Steps on Davinci Configurator](#)(see page 160)
 - Step 12: Final Conections and Verification(see page 160)
- [Troubleshooting](#)(see page 161)
 - UL (Upper Layer) Indication related issues:(see page 161)

11.1 Workflow Diagram



11.2 Prerequisites

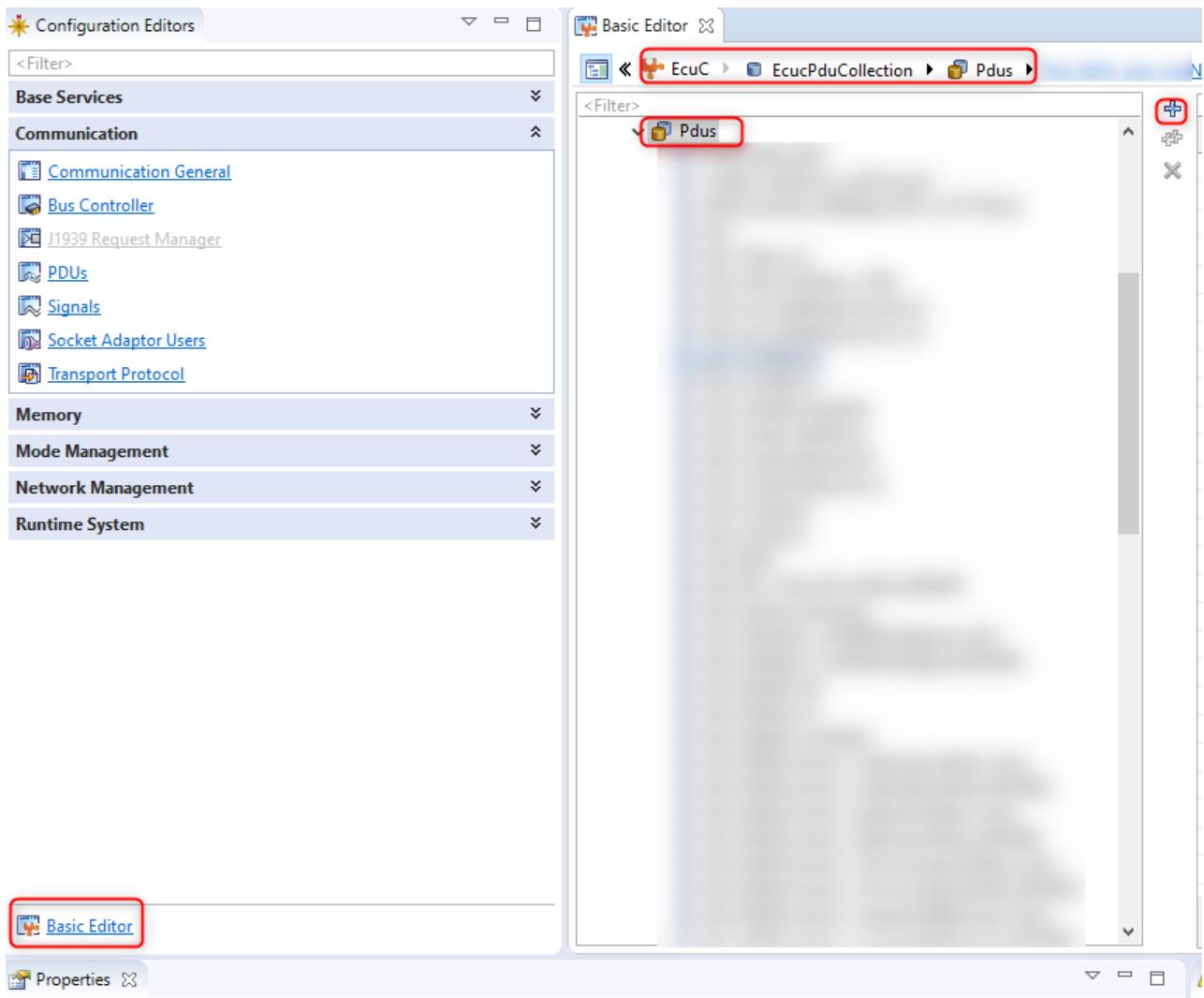
1. Make sure you have an up and running CAN configuration.
2. Make sure you are able to transmit any CAN Message (not the one to be configured in this recipe).
3. Make sure to have a CANoe simulation or any other kind of verification tool.

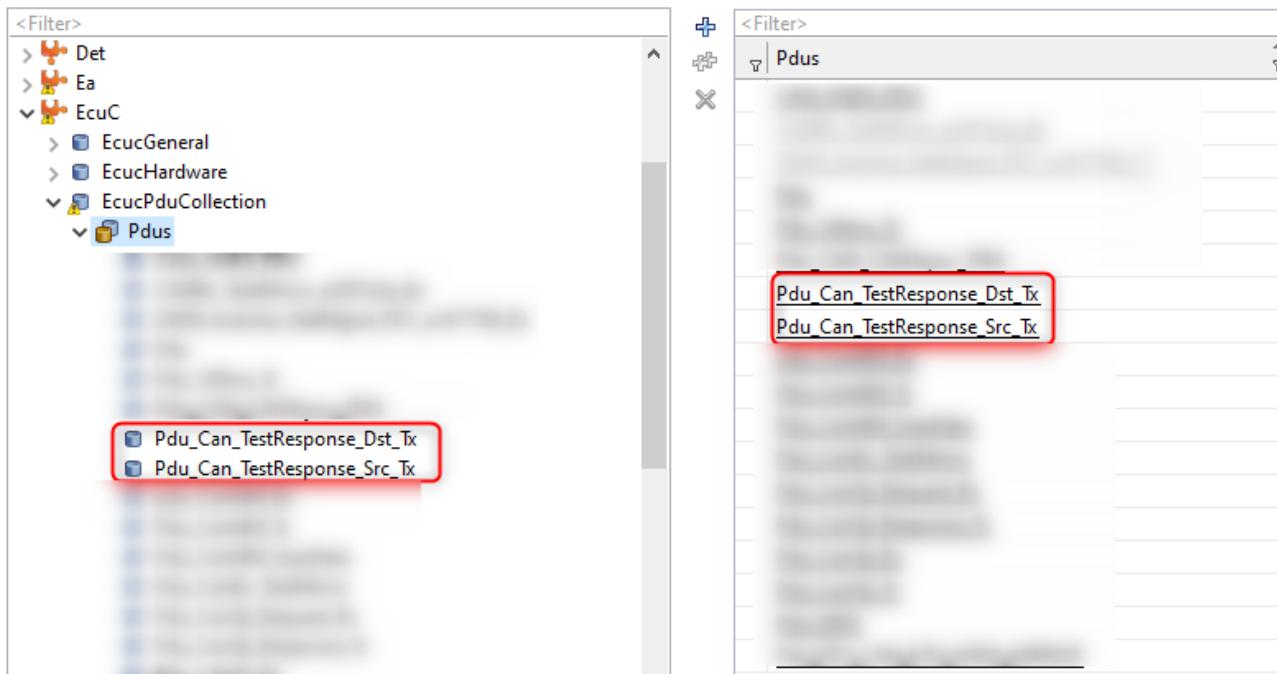
11.3 Steps on Davinci Configurator

11.3.1 Step 1: Create two new Pdu

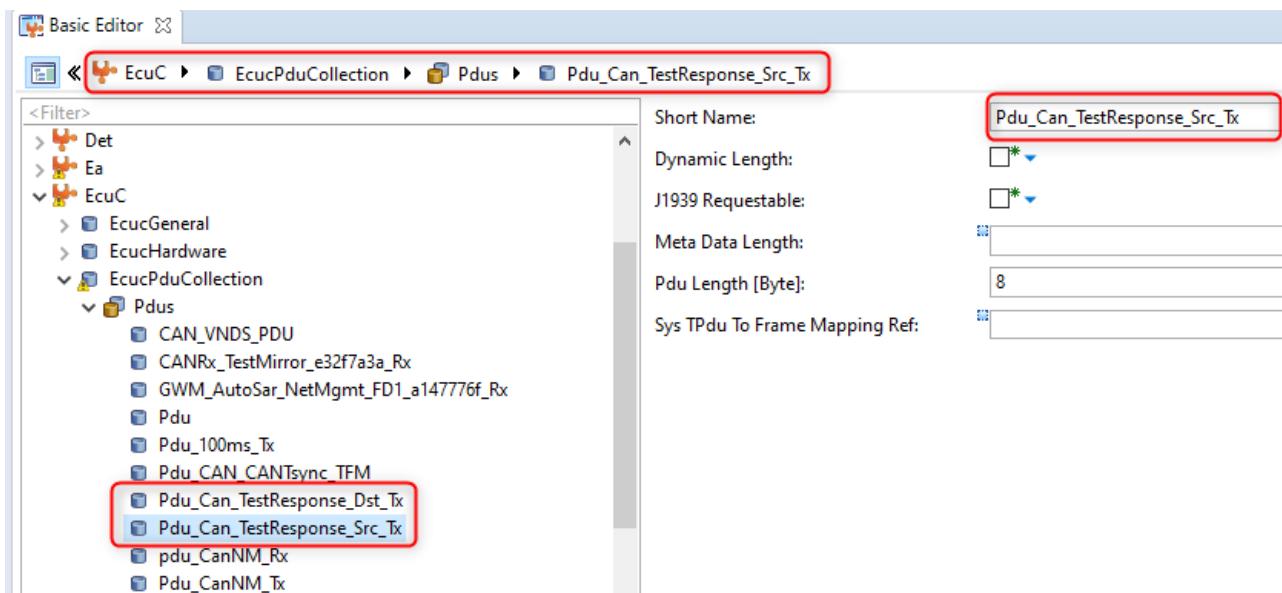
Go to Basic Editor → EcuC → EcucPduCollection → Pdu and Click on the "+" sign.

You need to create two, use the Suffix _Dst and _Src to differentiate them





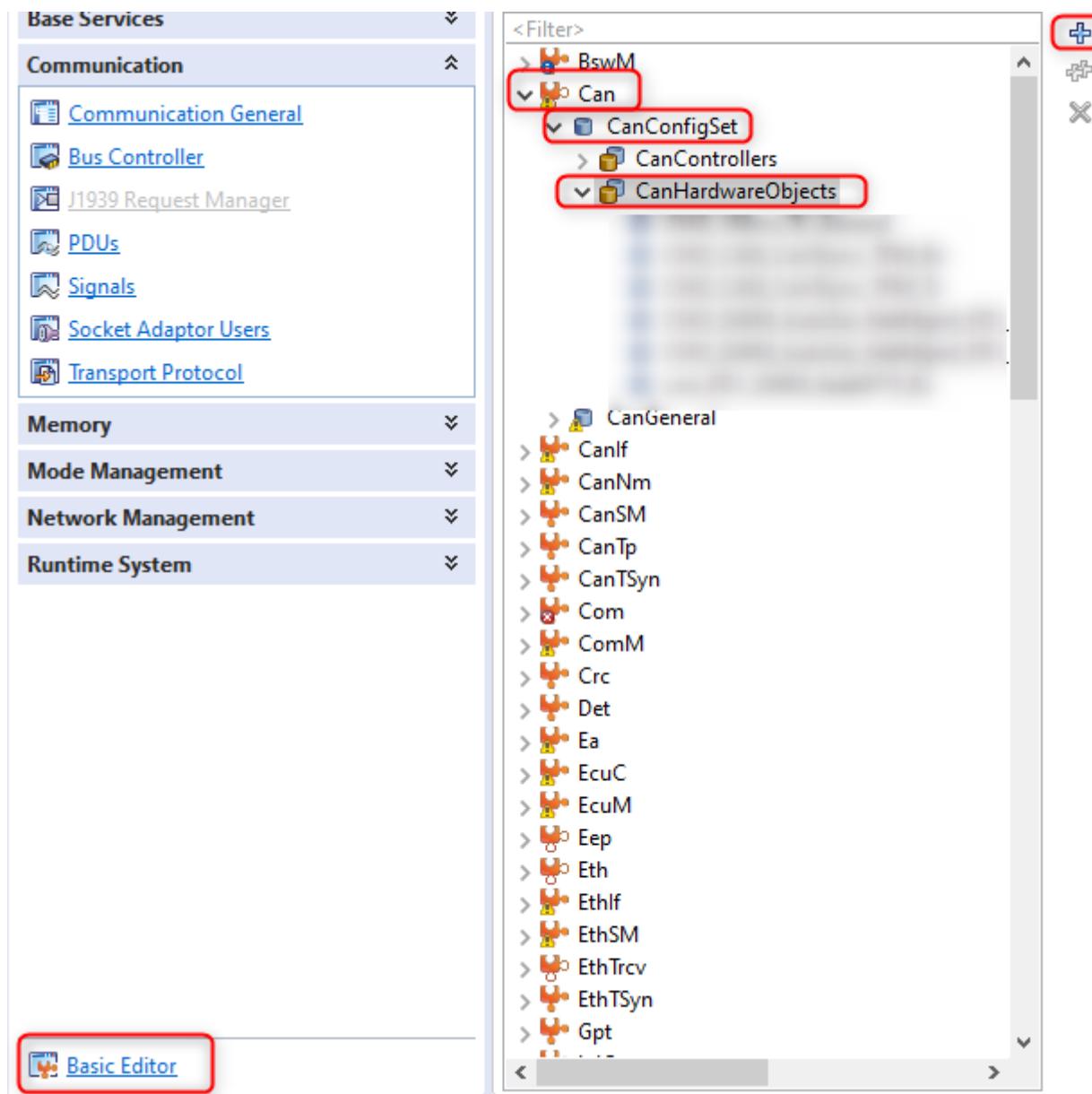
After creating the Pdu with a meaningful and unique name, configure the Data Length and any other field needed.

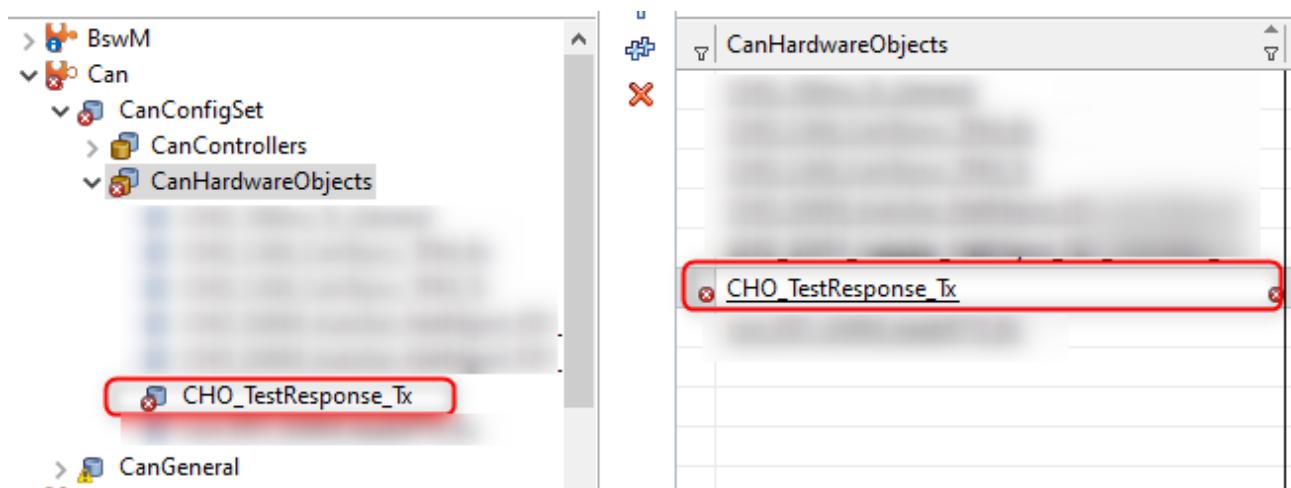


Test

11.3.2 Step 2: Create a new CanHardwareObject

Go to Basic Editor → Can → CanConfigSet → CanHardwareObject, and click on the "+" sign.





After Creating the new CanHardwareObject with a meaningfull and unique name, configure the following fields:

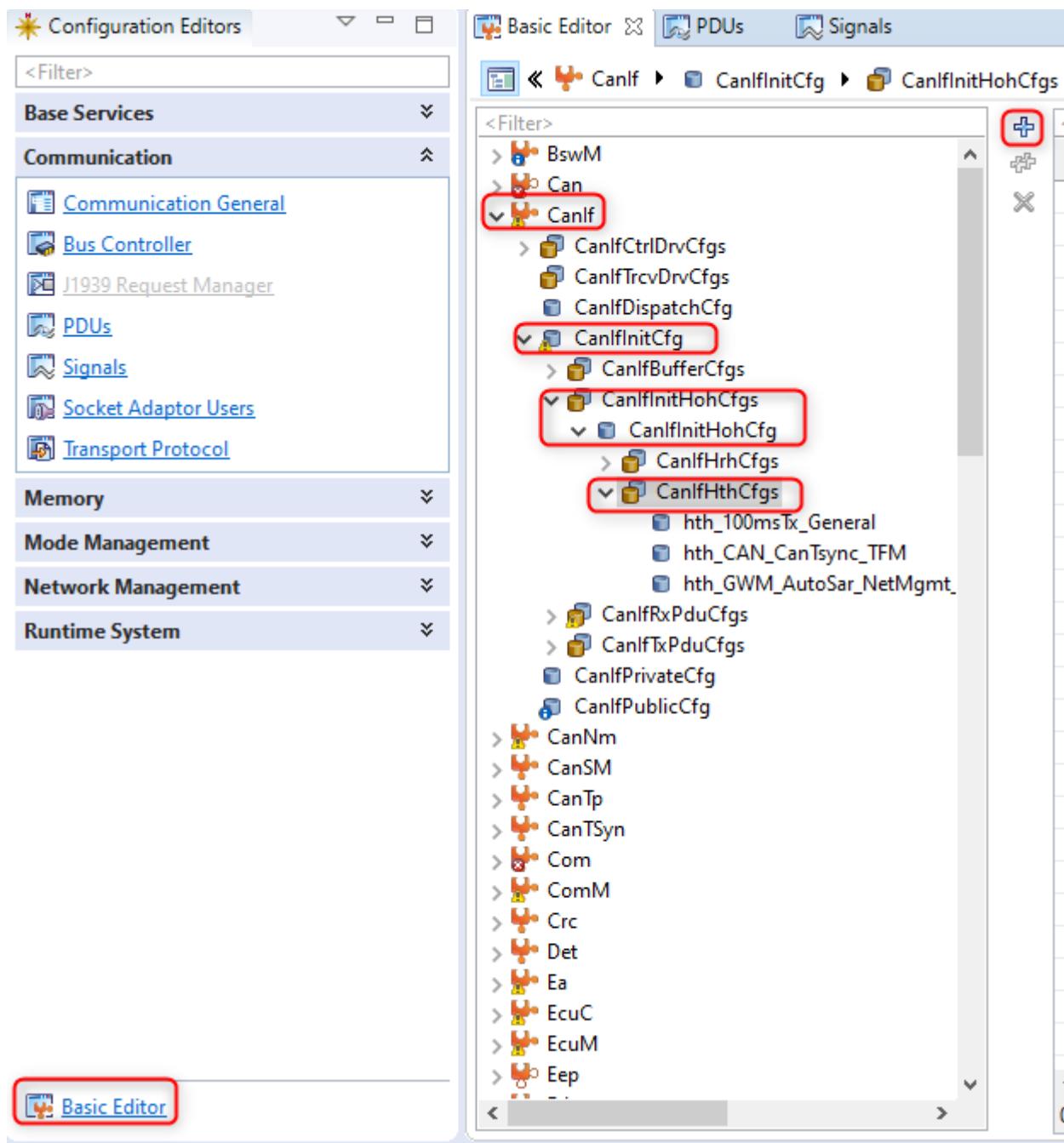
- Controller Ref: **(Use the previously configure CAN Controller)**
- Handle Type: **FULL**
- Id Value: **(ID from your aplicacion to be received)**
- Object Type: **TRANSMIT**

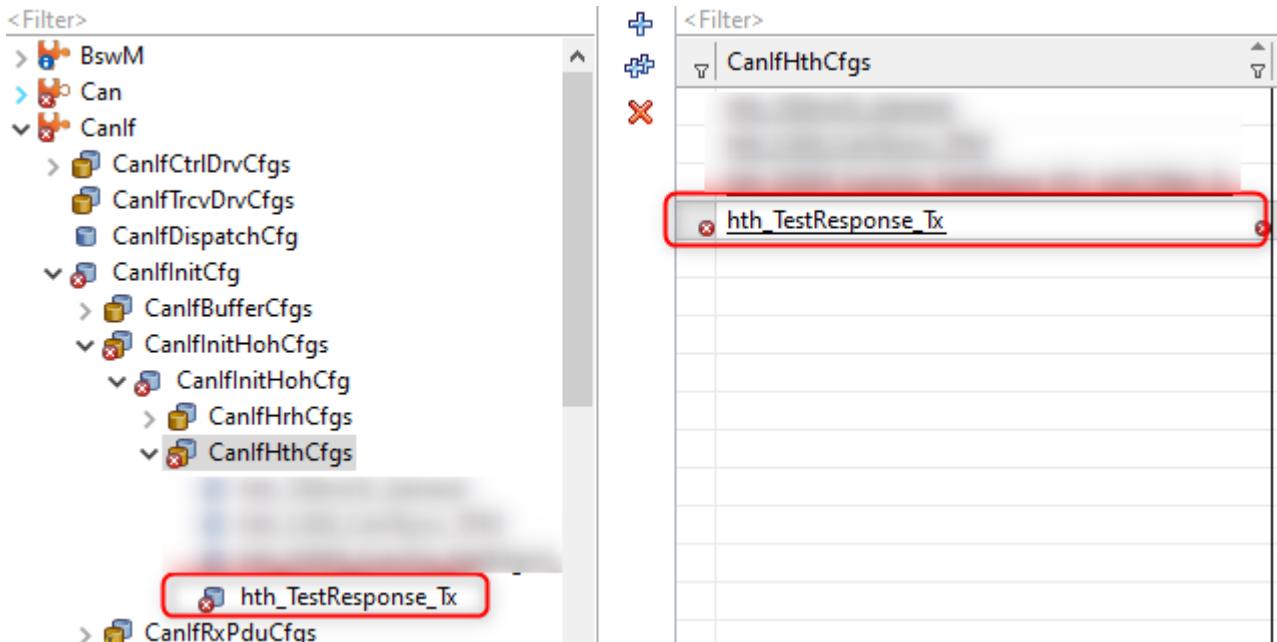
anHardwareObjects ▶ CHO_TestResponse_Tx

Short Name:	CHO_TestResponse_Tx
Common CAN Node:	
Controller Ref:	/ActiveEcuC/Can/CanConfigSet/FD1_CAN_GWM_6ab810d5 [...]
Enable Polling Processing:	<input type="checkbox"/> *
Fd Padding Value:	0x0 hex *
Handle Type:	FULL
Hardware FIFO:	<input type="checkbox"/> *
Id Type:	STANDARD
Id Value:	0x19 hex
Main Function RW Period Ref:	
Max Data Len:	64 dec
Multiplexed Transmission:	<input type="checkbox"/> *
Object Hw Handle:	0 dec *
Object Hw Size:	1 dec
Object Id:	2 dec
Object Type:	TRANSMIT
Filter Mask Ref	
<input style="margin-right: 10px;" type="button" value="+"/> Use '+' to add parameters <input type="button" value="X"/>	

11.3.3 Step 3: Create a new CanIfHthCfgs

Go to Basi Editor → CanIf → CanIfInitCfg → CanIfInitHohCfgs → CanIfInitHohCfg → CanIfHthCfgs, and click the "+" sign.





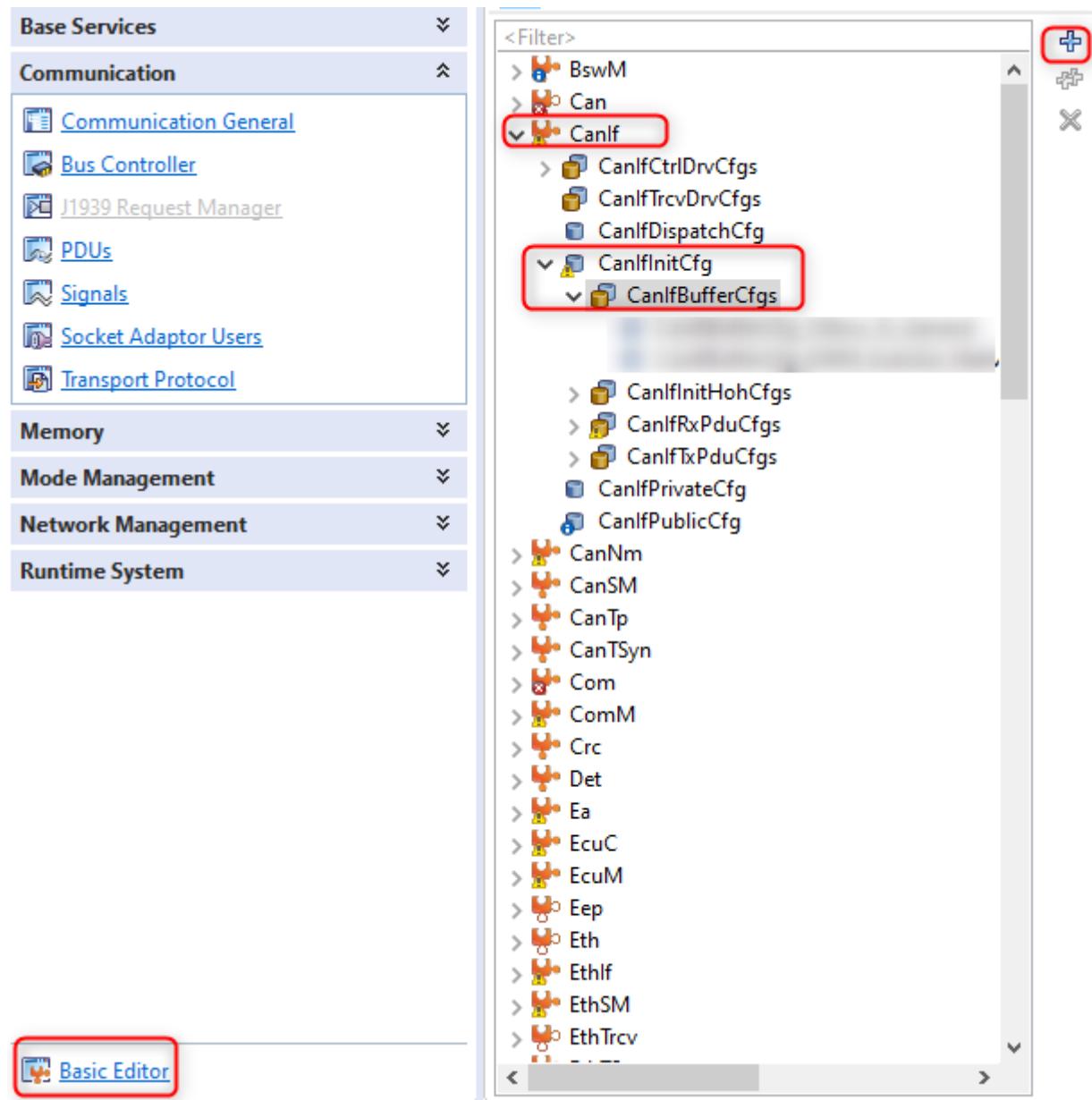
After creating a new CanIfHthCfgs with a meaningful and unique name, configure the following fields:

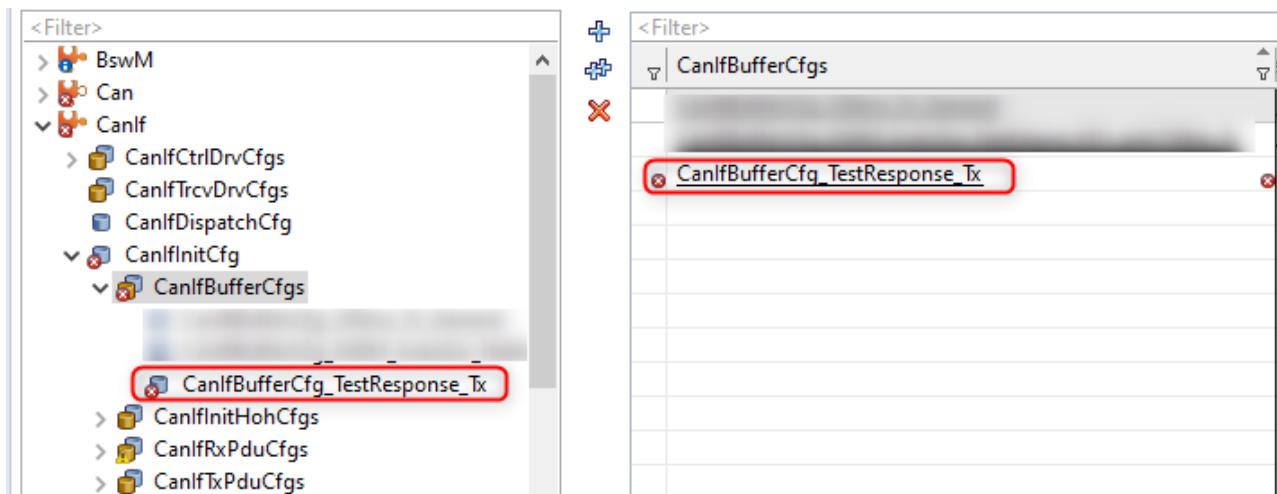
- Hth Can Ctrl Id Ref: **(Use the previously created CAN Controller)**
- Hth Id Sym Ref: **(Use the CanHardwareObject created in the previous step 2)**

Short Name:	<input type="text" value="hth_TestResponse_Tx"/>
Hth Can Ctrl Id Ref:	<input type="text" value="/ActiveEcuC/CanIf/CanIfCtrlDrvCfg_7d254554/FD1_CAN_GWM_6ab810d5"/> [...]
Hth Can Handle Type Ref:	<input type="text"/> [...]
Hth Id Sym Ref:	<input type="text" value="/ActiveEcuC/Can/CanConfigSet/CHO_TestResponse_Tx"/> [...]

11.3.4 Step 4: Create a New CanIfBufferCfgs

Go to Basi Editor → CanIf → CanIfInitCfg → CanIfBufferCfgs and click the "+" sign.





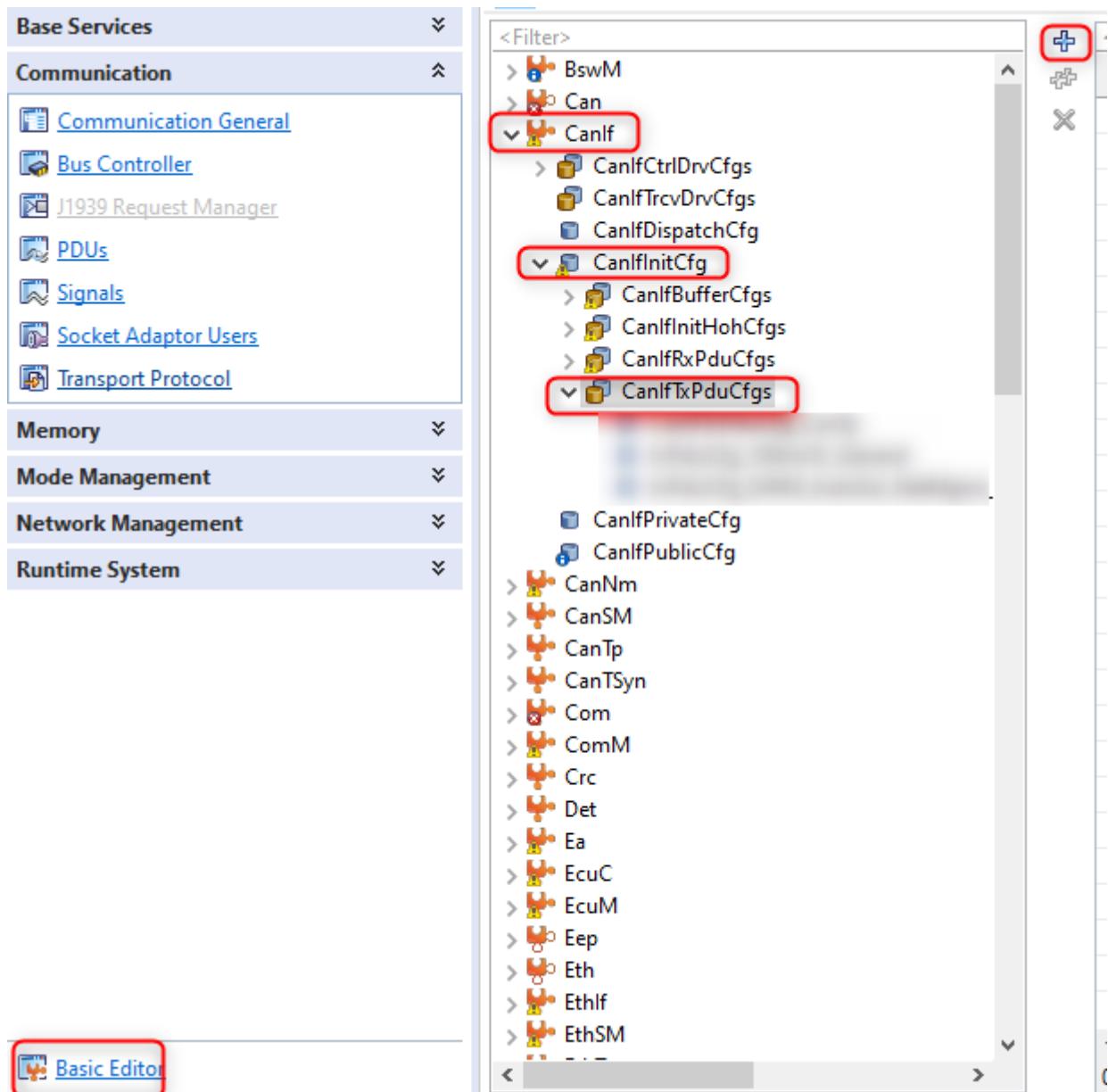
After creating a new CanIfBufferCfgs with a meaningful and unique name, configure the following fields:

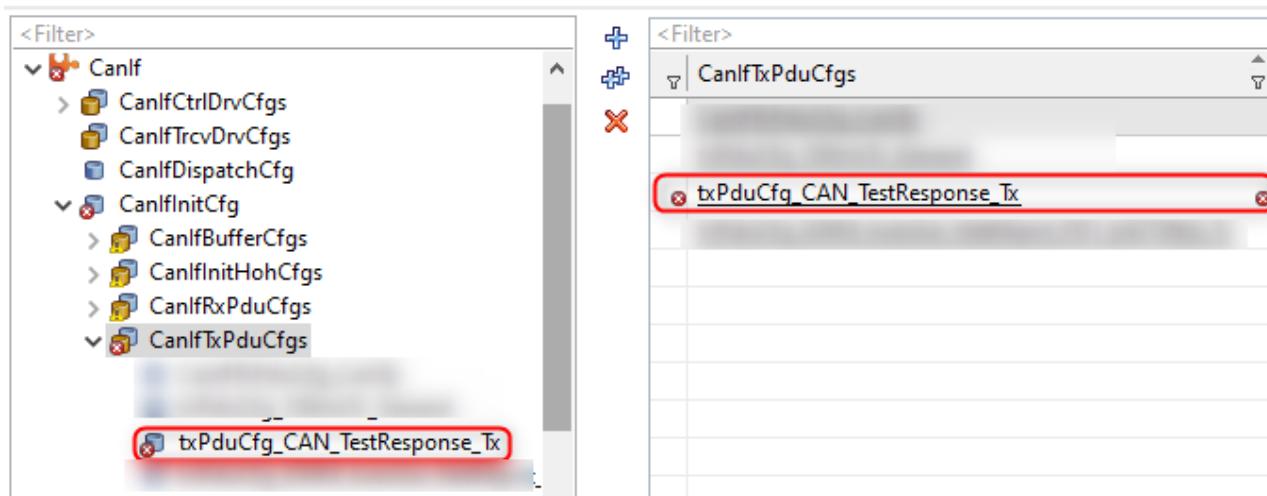
- Buffer Hth Ref: **(Use the Hth created in the previous Step 3)**

Short Name:	<input type="text" value="CanIfBufferCfg_TestResponse_Tx"/>
Buffer Hth Ref:	<input type="text" value="/ActiveEcuC/CanIf/CanIfInitCfg/CanIfInitHohCfg/hth_TestResponse_Tx"/> [...]
Buffer Size:	<input type="text" value="0"/> dec *
Tx Buffer Handling Type:	<input type="text" value="NONE"/> *
Tx Buffer Mapped Tx Pdus:	<input type="text" value="0"/> dec *
Tx Buffer Max Pdu Length [Byte]:	<input type="text" value="8"/> dec *

11.3.5 Step 5: Create a New CanIfTxPduCfgs

Go to Basi Editor → CanIf → CanIfInitCfg → CanIfTxPduCfgs and click the "+" sign.





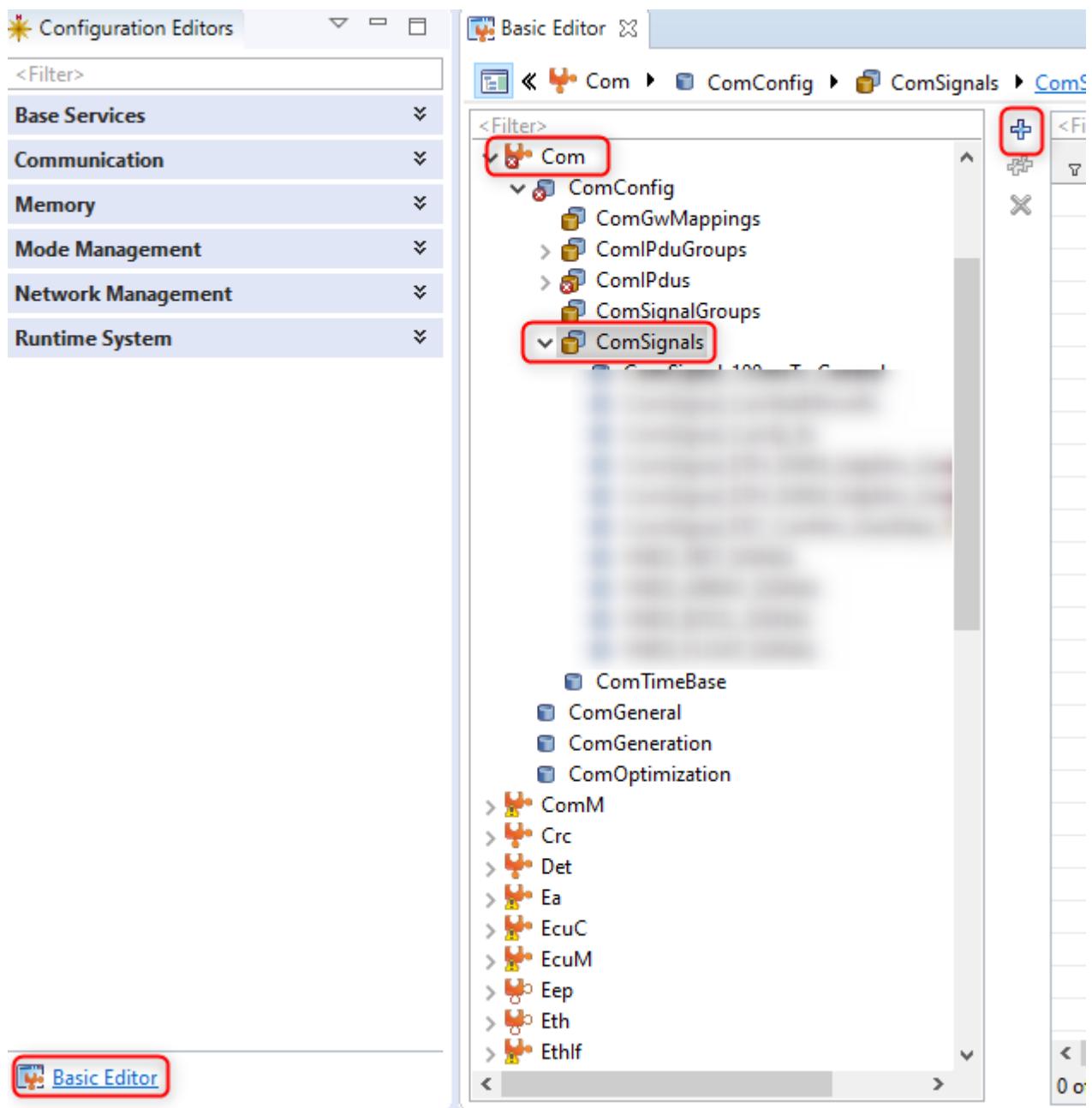
After creating a new CanIfTxPduCfgs with a meaningful and unique name, configure the following fields:

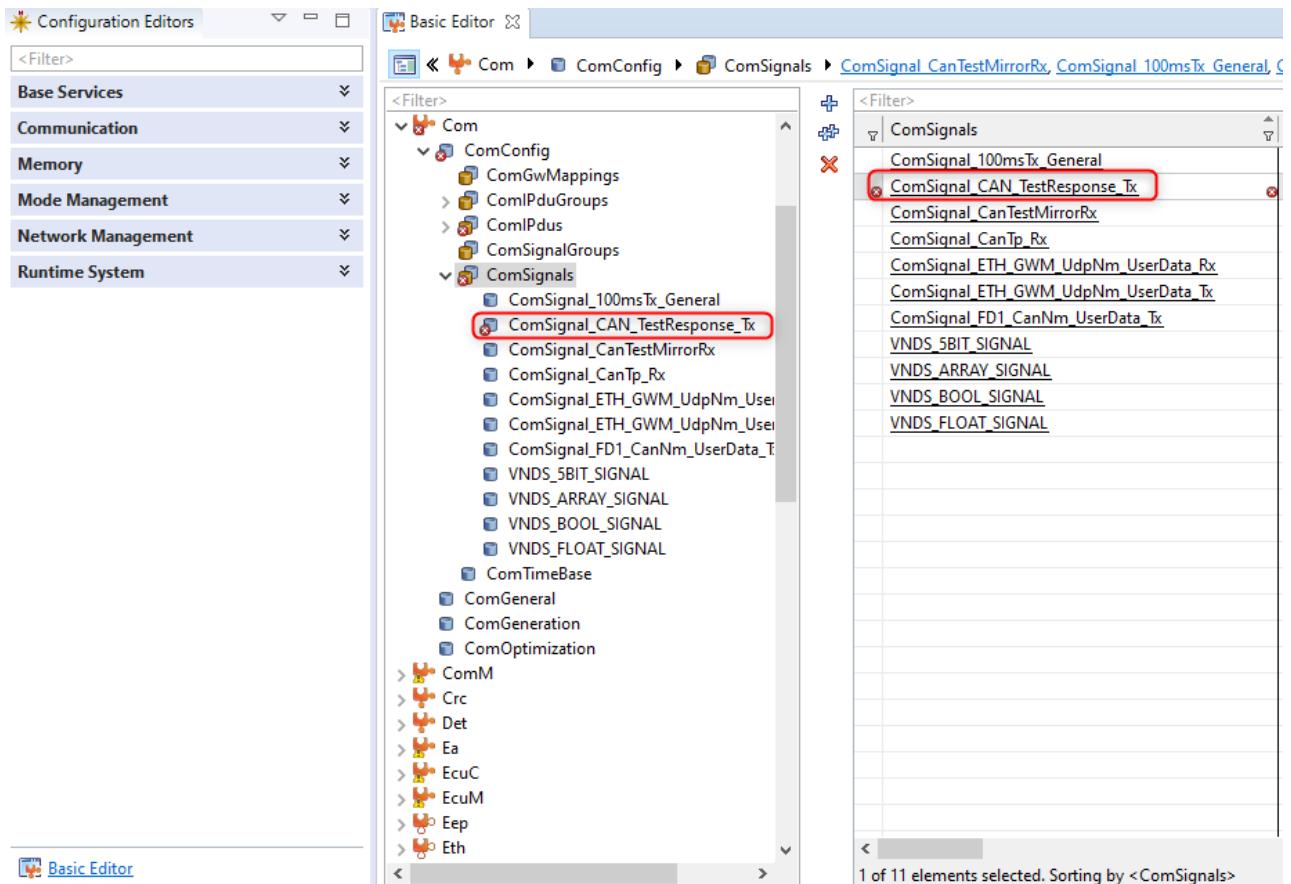
- Tx Pdu Buffer Reference: **(Use the previously one created on step 4)**
- Tx Pdu Can Id: **(Create one base on your requirements)**
- Tx Pdu Can Id Type: **STANDARD_CAN**
- Tx Pdu Ref: **(Use the previously one created on step 1 with _Dst suffix)**
- Tx Pdu User Tx Confirmation UL: **PDUR**

Short Name:	txPduCfg_CAN_TestResponse_Tx
Tx Pdu Bsw Sch Excl Area Id Ref:	
Tx Pdu Buffer Ref:	/ActiveEcuC/CanIf/CanIfInitCfg/CanIfBufferCfg_TestResponse_Tx
Tx Pdu Can Id:	0x19
Tx Pdu Can Id Mask:	
Tx Pdu Can Id Type:	STANDARD_CAN
Tx Pdu Data Checksum Pdu:	<input checked="" type="checkbox"/>
Tx Pdu Dlc [Byte]:	8
Tx Pdu Id:	3
Tx Pdu Pn Filter Pdu:	<input checked="" type="checkbox"/>
Tx Pdu Read Notify Status:	<input checked="" type="checkbox"/>
Tx Pdu Ref:	/ActiveEcuC/EcuC/EcucPduCollection/Pdu_Can_TestResponse_Dst_Tx
Tx Pdu Truncation:	<input checked="" type="checkbox"/>
Tx Pdu Type:	STATIC
Tx Pdu User Tx Confirmation Name:	PduR_CanIfTxConfirmation
Tx Pdu User Tx Confirmation UL:	PDUR

11.3.6 Step 6: Create a New ComSignals

Go to Basic Editor → Com → ComConfig → ComSignals and Click the "+" sign





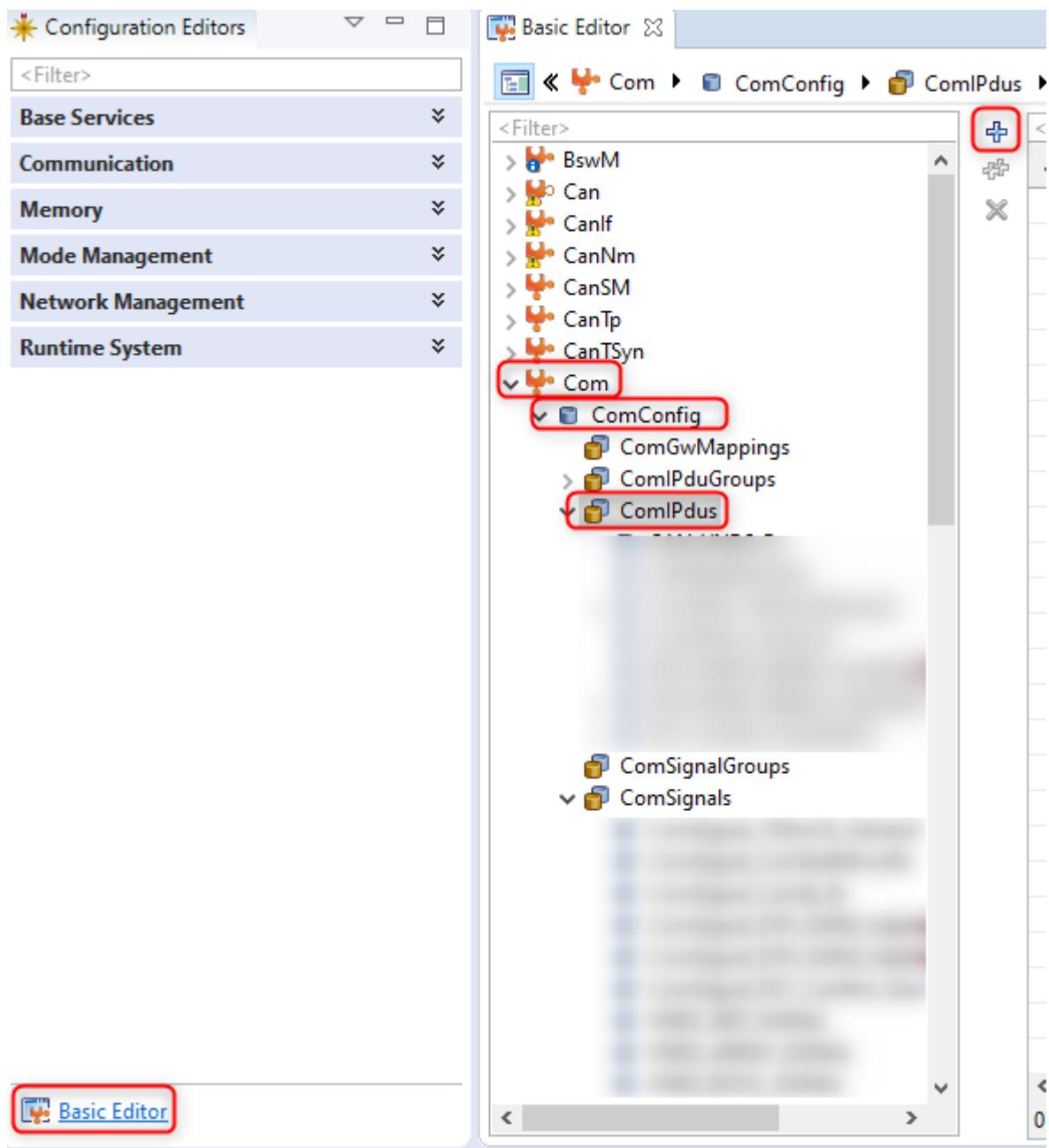
After the creating the ComSignals with a meaningfull and unique name, Configure the following fields:

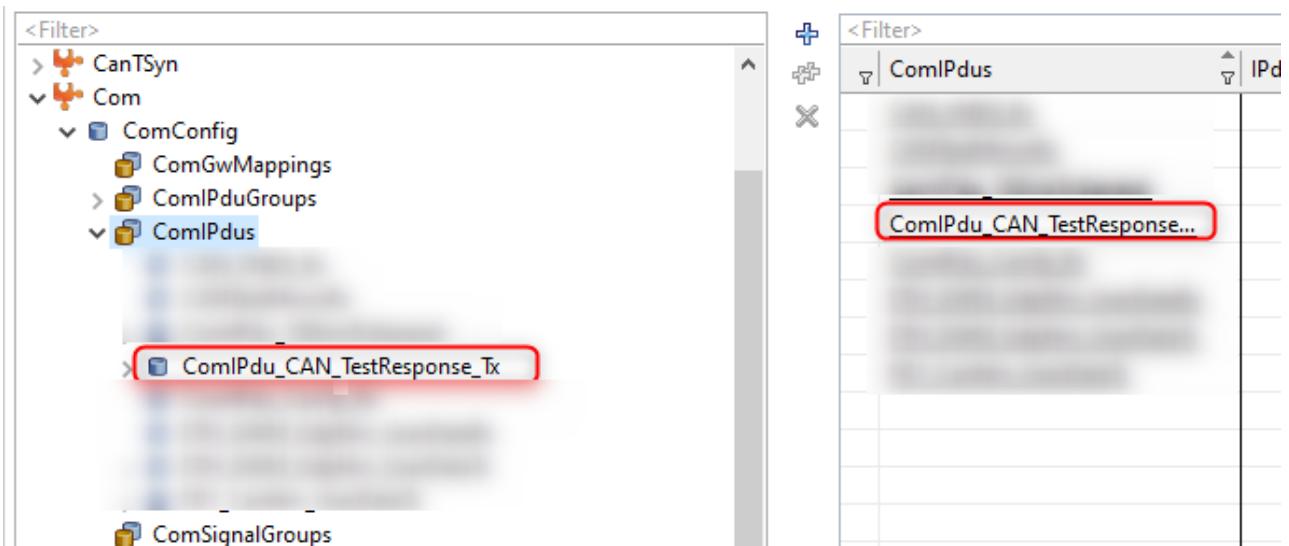
- Signal Access: **ACCESS_NEEDED_BY_SWC_OR_COM**
- Sigaln Endianess: **OPAQUE**.
- Signal Init Value: **0 0 0 0 0 0 0 0**
- Signal Length (Byte): **8**
- SignalType: **UINT8_N**
- Transfer Property: **PENDING**

Short Name:	ComSignal_CAN_TestResponse_Tx	▼
Bit Position:	0	dec *▼
Bit Size [Bit]:	8	dec ▼
Data Invalid Action:	▼	▼
Error Notification:	▼	▼
First Timeout [ms]:	▼	dec ▼
Handle Id:	0	dec *▼
Invalid Notification:	▼	▼
Notification:	▼	▼
Rx Data Timeout Action:	▼	▼
Rx Data Timeout Substitution Value:	▼	▼
Signal Access:	ACCESS_NEEDED_BY_SWC_OR_COM	▼ ▾
Signal Data Invalid Value:	▼	▼
Signal Endianness:	OPAQUE	▼ ▾
Signal Init Value:	00000000	▼
Signal Length [Byte]:	8	dec ▼
Signal Type:	UINT8_N	▼ ▾
System Template System Signal Ref:	▼	[...] ▾
Timeout [ms]:	▼	dec ▼
Timeout Notification:	▼	▼
Transfer Property:	PENDING	▼ *▼
Update Bit Position:	▼	dec ▼

11.3.7 Step 7: Create a new ComIPdus

Go to Basic Editor → Com → ComConfig → ComIPdus and Click the "+" sign.





After the creating the ComIPdus with a meaningfull and unique name, configure the following fields:

- IPdu Direction: **SEND**
- Pdu Id Ref: **(Use the previously one created on step 1 with the _Src suffix)**
- IPdu Signal Ref: **(Use the previously one created on step 6)**

Short Name:	ComIPdu_CAN_TestResponse_Tx	▼
IPdu Callout:		▼
IPdu Cancellation Support:	<input type="checkbox"/> *	▼
IPdu Direction:	SEND	▼
IPdu Handle Id:	0	dec *▼
IPdu Signal Processing:	DEFERRED	*▼
IPdu Trigger Transmit Callout:		▼
IPdu Type:	NORMAL	*▼
Pdu Id Ref:	/ActiveEcuC/EcuC/EcucPduCollection/Pdu_Can_TestResponse_Src_Tx	[...] ▼

IPdu Group Ref

+ Use '+' to add parameters

X

IPdu Signal Group Ref

+ Use '+' to add parameters

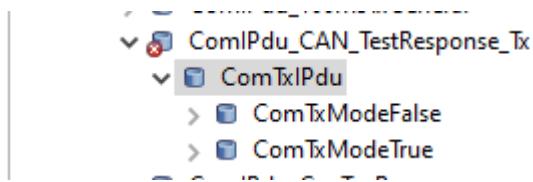
X

IPdu Signal Ref

+ /ActiveEcuC/Com/ComConfig/ComSignal_CAN_TestResponse_Tx

X

Now, press right click on the created ComIPdu and create a new ComTxIPdu container, which will create the necessary sub containers (ComTxModeFalse and ComTxModeTrue).



Inside each of the sub-containers there is another subcontainer called ComTxMode, please Configure both following the image bellow:

Short Name:	ComTxMode
Tx Mode Mode:	MIXED
Tx Mode Number Of Repetitions:	0
Tx Mode Repetition Period [ms]:	
Tx Mode Time Offset [ms]:	0
Tx Mode Time Period [ms]:	1000

(i) Deviation

The previous images are for a Periodic Message only, using as period the Tx Mode Time Period. If the Message to be created is Triggered depending of the runnables change the configuration on thi step as the following list:

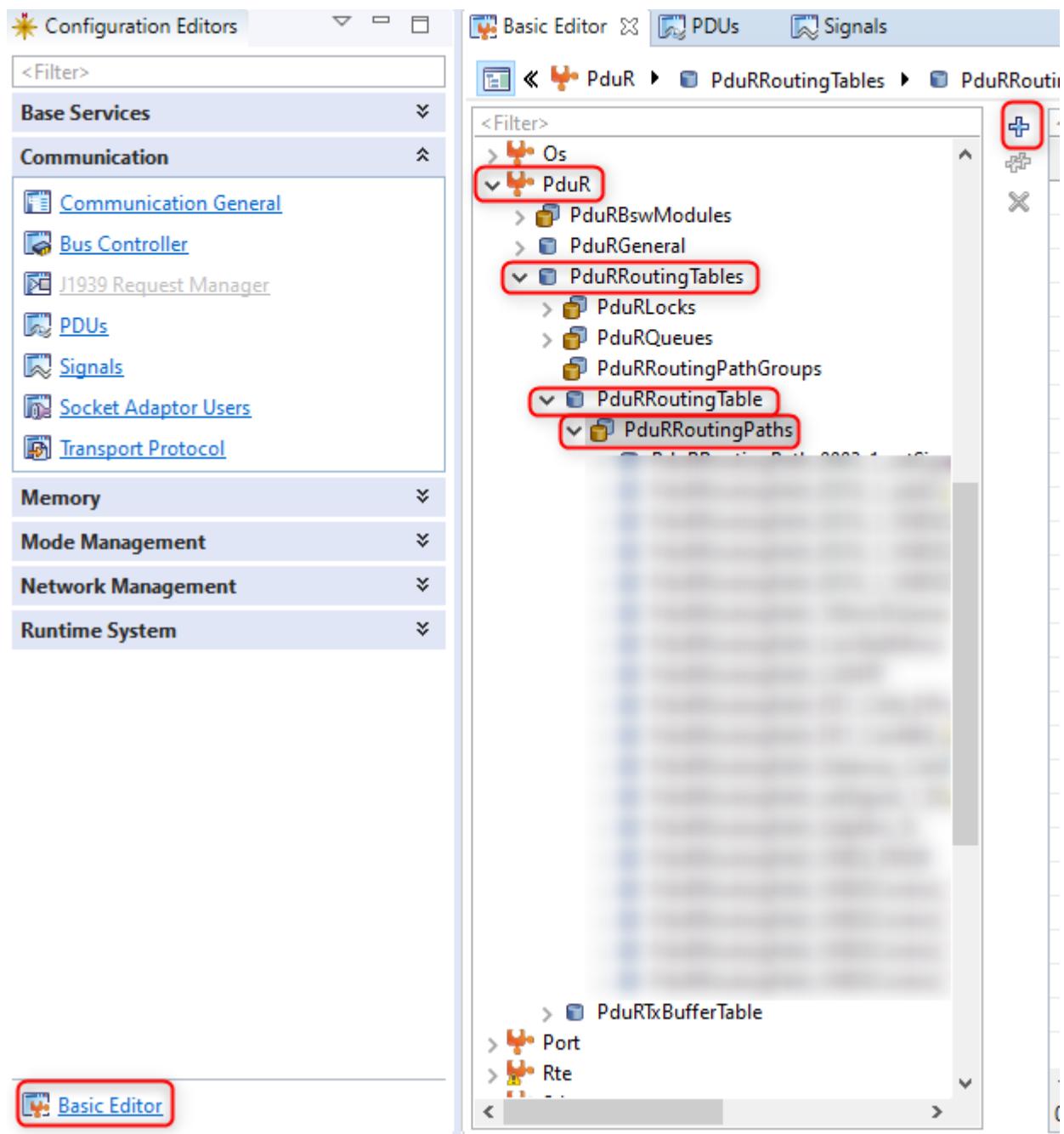
- Tx Mode: **Direct**.
- Tx Mode Time Period: **0**.

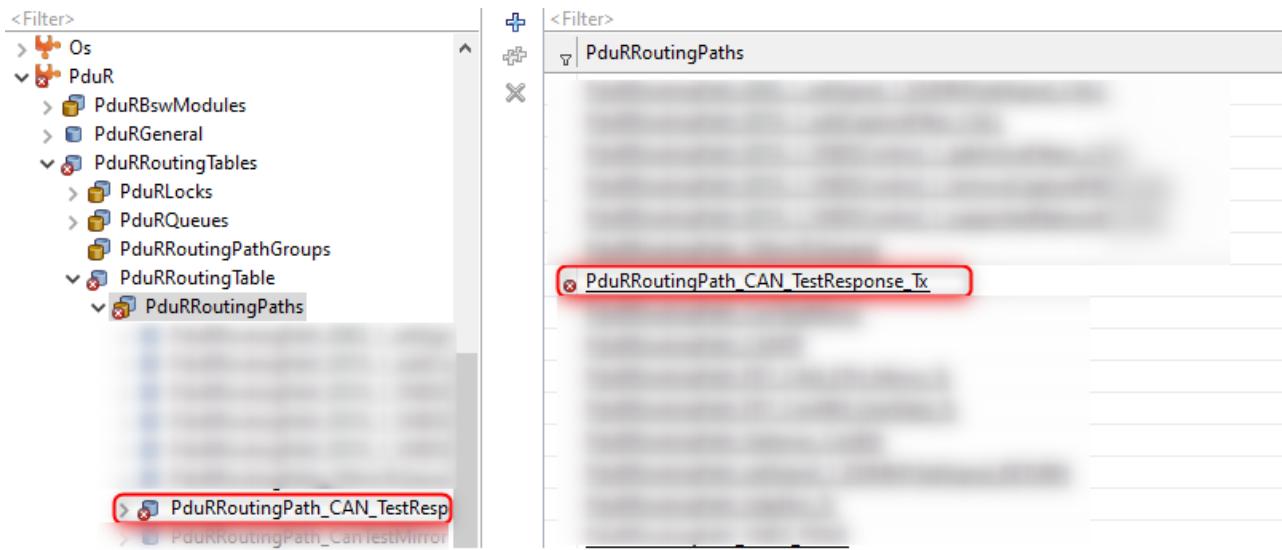
And under ComSingals (Step 6)

- Transfer Property: **TRIGGERED**

11.3.8 Step 8: Create a New PduRRoutingPath

Go to Basi Editor → PduR → PduRRoutingTables → PduRRoutingTable → PduRRoutingPaths and Click the "+" sign.





After creating the PduRRoutingPath with a meaningful and unique name, a container called PduRSrcPdu_1 shall be created please configure the following:

- Src Pdu Ref: **(Use the previously one created on step 1 with the _Src Suffix)**

Short Name:	PduRSrcPdu_1
Source Pdu Handle Id:	1
Src Pdu Direction:	TRANSMIT
Src Pdu PduRBswModules Ref:	/ActiveEcuC/PduR/Com
Src Pdu Ref:	/ActiveEcuC/EcuC/EcucPduCollection/Pdu_Can_TestResponse_Src_Tx

Then after expand the PduRDestPdu container of the created PduRRoutingPath and configure the following:

- Dest Pdu Data Provision: **PDUR_DIRECT**
- Dest Pdu Ref: **(Use the previously one created on step 1 with the _Dst suffix)**
- Pdu Length Handling Strategy: **IGNORE**

Short Name:	PduRDestPdu_1
Dest Pdu Data Processing:	IMMEDIATE
Dest Pdu Data Provision:	PDUR_DIRECT
Dest Pdu Direction:	TRANSMIT
Dest Pdu Handle Id:	2
Dest Pdu PduRBswModules Ref:	/ActiveEcuC/PduR/CanIf
Dest Pdu Queue Depth:	5
Dest Pdu Ref:	/ActiveEcuC/EcuC/EcucPduCollection/Pdu_Can_TestResponse_Dst_Tx
Dest Pdu Routing Type:	API_FORWARDING
Dest Tx Buffer Ref:	5
Pdu Length Handling Strategy:	IGNORE
Queue Ref:	5
Tp Threshold:	5
Transmission Confirmation:	<input checked="" type="checkbox"/> *

11.4 Steps to be performed directly on the ARXML File

11.4.1 Step 9: Create the Signals in the ARXML and the Port Mapping

Add the following ARXML blocks with the proper Short name and Paths in the Application Software Component which will handle the Signal transmission.

⚠ Please Consider

The following are just examples and templates, you may need to add more "ARXML Sections" to make this one work.

i.e:

In the following code snapshot some Port references are used, these port nee to be created elsewhere (not in the scope of this Recipe)

11.4.1.1 Changes on the ECU System Extract or Communication System Extract.

```
<CAN-FRAME-TRIGGERING UUID="BE91C722-B114-4149-89CA-158FBBEE014D">
<SHORT-NAME>CF_TFM_Test_Resp_Tx</SHORT-NAME>
<FRAME-PORT-REFS>
  <FRAME-PORT-REF DEST="FRAME-PORT">/Topology/HardwareComponents/GWM/con_FD1_GWM/
framePort_CANTx_Test_Resp</FRAME-PORT-REF>
</FRAME-PORT-REFS>
```

```
<FRAME-REF DEST="CAN-FRAME">/Communication/Frames/CF_TFM_Test_Resp_Tx</FRAME-REF>
<CAN-ADDRESSING-MODE>STANDARD</CAN-ADDRESSING-MODE>
<IDENTIFIER>25</IDENTIFIER>
</CAN-FRAME-TRIGGERING>
```

```
<I-SIGNAL-TRIGGERING UUID="BEC3138A-2E7E-4CD2-8B31-71408F66D6C2">
  <SHORT-NAME>isigTrig_CANTx_Test_Resp</SHORT-NAME>
  <I-SIGNAL-PORT-REFS>
    <I-SIGNAL-PORT-REF DEST="I-SIGNAL-PORT">/Topology/HardwareComponents/GWM/
con_FD1_GWM/isignalPort_CANTx_Test_Resp</I-SIGNAL-PORT-REF>
  </I-SIGNAL-PORT-REFS>
  <I-SIGNAL-REF DEST="I-SIGNAL">/TimeFollower/Systems/Signals/Data_Resp_TFM_Tx</I-
SIGNAL-REF>
</I-SIGNAL-TRIGGERING>
```

```
<PDU-TRIGGERING UUID="F1D29002-45E9-4820-BAB2-CC49BF70A931">
  <SHORT-
NAME>FD1_CAN_FD1_CAN_FD1_CAN_FD1_CAN_FD1_CAN_FD1_CAN_FD1_CAN_D_F62AF3D8_8C18_
4779_A0C0_9BD39632CEB4_VFrame_VFrame_VFr2A821DA0</SHORT-NAME>
  <I-PDU-PORT-REFS>
    <I-PDU-PORT-REF DEST="I-PDU-PORT">/Topology/HardwareComponents/GWM/con_FD1_GWM/
ipduPort_CANTx_Test_Resp</I-PDU-PORT-REF>
  </I-PDU-PORT-REFS>
  <I-PDU-REF DEST="I-SIGNAL-I-PDU">/TimeFollower/Systems/Signals/
Pdu_Can_TestResponse_Tx</I-PDU-REF>
</PDU-TRIGGERING>
```

```
<FRAME-PORT UUID="C0769E8C-687A-49B3-88E1-4207076C5557">
  <SHORT-NAME>framePort_CANTx_Test_Resp</SHORT-NAME>
  <COMMUNICATION-DIRECTION>OUT</COMMUNICATION-DIRECTION>
</FRAME-PORT>
```

```
<I-PDU-PORT UUID="8EE13968-7E3B-4486-9160-6F9573C7A235">
  <SHORT-NAME>ipduPort_CANTx_Test_Resp</SHORT-NAME>
  <COMMUNICATION-DIRECTION>OUT</COMMUNICATION-DIRECTION>
</I-PDU-PORT>
```

```
<CAN-FRAME UUID="1988BD2A-B059-4057-8A9B-A3042F45F3F7">
  <SHORT-NAME>CF_TFM_Test_Resp_Tx</SHORT-NAME>
</CAN-FRAME>
```

```
<FIBEX-ELEMENT-REF-CONDITIONAL>
```

```

<FIBEX-ELEMENT-REF DEST="I-SIGNAL"/>/TimeFollower/Systems/Signals/Data_Resp_TFM_Tx</
FIBEX-ELEMENT-REF>
</FIBEX-ELEMENT-REF-CONDITIONAL>
<FIBEX-ELEMENT-REF-CONDITIONAL>
  <FIBEX-ELEMENT-REF DEST="I-SIGNAL-I-PDU"/>/TimeFollower/Systems/Signals/
Pdu_Can_TestResponse_Tx</FIBEX-ELEMENT-REF>
</FIBEX-ELEMENT-REF-CONDITIONAL>

```

```

<SENDER-RECEIVER-TO-SIGNAL-MAPPING>
  <DATA-ELEMENT-IREF>
    <CONTEXT-PORT-REF DEST="P-PORT-PROTOTYPE"/>/ECUCompositionTypes/ECU_Composition/
Data_Resp</CONTEXT-PORT-REF>
    <TARGET-DATA-PROTOTYPE-REF DEST="VARIABLE-DATA-PROTOTYPE"/>/TimeFollower/
Interfaces/Data_Resp/Data_Resp</TARGET-DATA-PROTOTYPE-REF>
  </DATA-ELEMENT-IREF>
  <SYSTEM-SIGNAL-REF DEST="SYSTEM-SIGNAL"/>/TimeFollower/Systems/Signals/
sysSignal_Data_Resp_TFM_Tx</SYSTEM-SIGNAL-REF>
</SENDER-RECEIVER-TO-SIGNAL-MAPPING>

```

```

<P-PORT-PROTOTYPE UUID="1C12227A-8195-4660-820B-9C37D77023A9">
  <SHORT-NAME>Data_Resp</SHORT-NAME>
  <ADMIN-DATA>
    <SDGS>
      <SDG GID="DV:DEV">
        <SD GID="DV:ImportModePreset">Keep</SD>
      </SDG>
    </SDGS>
  </ADMIN-DATA>
  <PROVIDED-INTERFACE-TREF DEST="SENDER-RECEIVER-INTERFACE"/>/TimeFollower/Interfaces/
Data_Resp</PROVIDED-INTERFACE-TREF>
</P-PORT-PROTOTYPE>

```

```

<DELEGATION-SW-CONNECTOR UUID="761DB2DA-6821-48B8-861C-40F381671AC5">
  <SHORT-NAME>FNV4Time_S_Data_Resp_Data_Resp</SHORT-NAME>
  <INNER-PORT-IREF>
    <P-PORT-IN-COMPOSITION-INSTANCE-REF>
      <CONTEXT-COMPONENT-REF DEST="SW-COMPONENT-PROTOTYPE"/>/ECUCompositionTypes/
ECU_Composition/FNV4Time</CONTEXT-COMPONENT-REF>
      <TARGET-P-PORT-REF DEST="P-PORT-PROTOTYPE"/>/TimeFollower/FNV4_SWCType/
S_Data_Resp</TARGET-P-PORT-REF>
    </P-PORT-IN-COMPOSITION-INSTANCE-REF>
  </INNER-PORT-IREF>
  <OUTER-PORT-REF DEST="P-PORT-PROTOTYPE"/>/ECUCompositionTypes/ECU_Composition/
Data_Resp</OUTER-PORT-REF>
</DELEGATION-SW-CONNECTOR>

```

11.4.1.2 Changes in the SWC that will use the CAN signals

```

<I-SIGNAL UUID="514877EF-B893-40E8-A86A-0C099353C53A">
  <SHORT-NAME>Data_Resp_TFM_Tx</SHORT-NAME>
  <DATA-TYPE-POLICY>LEGACY</DATA-TYPE-POLICY>
  <I-SIGNAL-TYPE>ARRAY</I-SIGNAL-TYPE>
  <LENGTH>8</LENGTH>
  <NETWORK-REPRESENTATION-PROPS>
    <SW-DATA-DEF-PROPS-VARIANTS>
      <SW-DATA-DEF-PROPS-CONDITIONAL>
        <BASE-TYPE-REF DEST="SW-BASE-TYPE">/TimeFollower/ImplementationDataTypes/
        uint8_8_none</BASE-TYPE-REF>
      </SW-DATA-DEF-PROPS-CONDITIONAL>
    </SW-DATA-DEF-PROPS-VARIANTS>
  </NETWORK-REPRESENTATION-PROPS>
  <SYSTEM-SIGNAL-REF DEST="SYSTEM-SIGNAL">/TimeFollower/Systems/Signals/
  sysSignal_Data_Resp_TFM_Tx</SYSTEM-SIGNAL-REF>
</I-SIGNAL>
<I-SIGNAL-I-PDU UUID="D0AC5774-3653-4F80-BFDD-789FF85A8D20">
  <SHORT-NAME>Pdu_Can_TestResponse_Tx</SHORT-NAME>
  <LENGTH>8</LENGTH>
  <I-SIGNAL-TO-PDU-MAPPINGS>
    <I-SIGNAL-TO-I-PDU-MAPPING UUID="01D10CA6-25A9-4C78-B15F-7909D5F17202">
      <SHORT-NAME>Data_Resp_TFM_Tx</SHORT-NAME>
      <I-SIGNAL-REF DEST="I-SIGNAL">/TimeFollower/Systems/Signals/Data_Resp_TFM_Tx</
      I-SIGNAL-REF>
      <PACKING-BYTE-ORDER>OPAQUE</PACKING-BYTE-ORDER>
      <START-POSITION>0</START-POSITION>
      <TRANSFER-PROPERTY>TRIGGERED</TRANSFER-PROPERTY>
    </I-SIGNAL-TO-I-PDU-MAPPING>
  </I-SIGNAL-TO-PDU-MAPPINGS>
  <UNUSED-BIT-PATTERN>0</UNUSED-BIT-PATTERN>
</I-SIGNAL-I-PDU>
<SYSTEM-SIGNAL UUID="9bf677cf-1503-4f38-8658-3d32aa785faD">
  <SHORT-NAME>sysSignal_Data_Resp_TFM_Tx</SHORT-NAME>
  <DYNAMIC-LENGTH>false</DYNAMIC-LENGTH>
  <PHYSICAL-PROPS>
    <SW-DATA-DEF-PROPS-VARIANTS>
      <SW-DATA-DEF-PROPS-CONDITIONAL>
        <COMPU-METHOD-REF DEST="COMPU-METHOD">/TimeFollower/Compumethods/
        CM_Data_Resp</COMPU-METHOD-REF>
        <DATA-CONSTR-REF DEST="DATA-CONSTR">/TimeFollower/DataConstraints/
        DC_Data_Resp</DATA-CONSTR-REF>
      </SW-DATA-DEF-PROPS-CONDITIONAL>
    </SW-DATA-DEF-PROPS-VARIANTS>
  </PHYSICAL-PROPS>
</SYSTEM-SIGNAL>

```

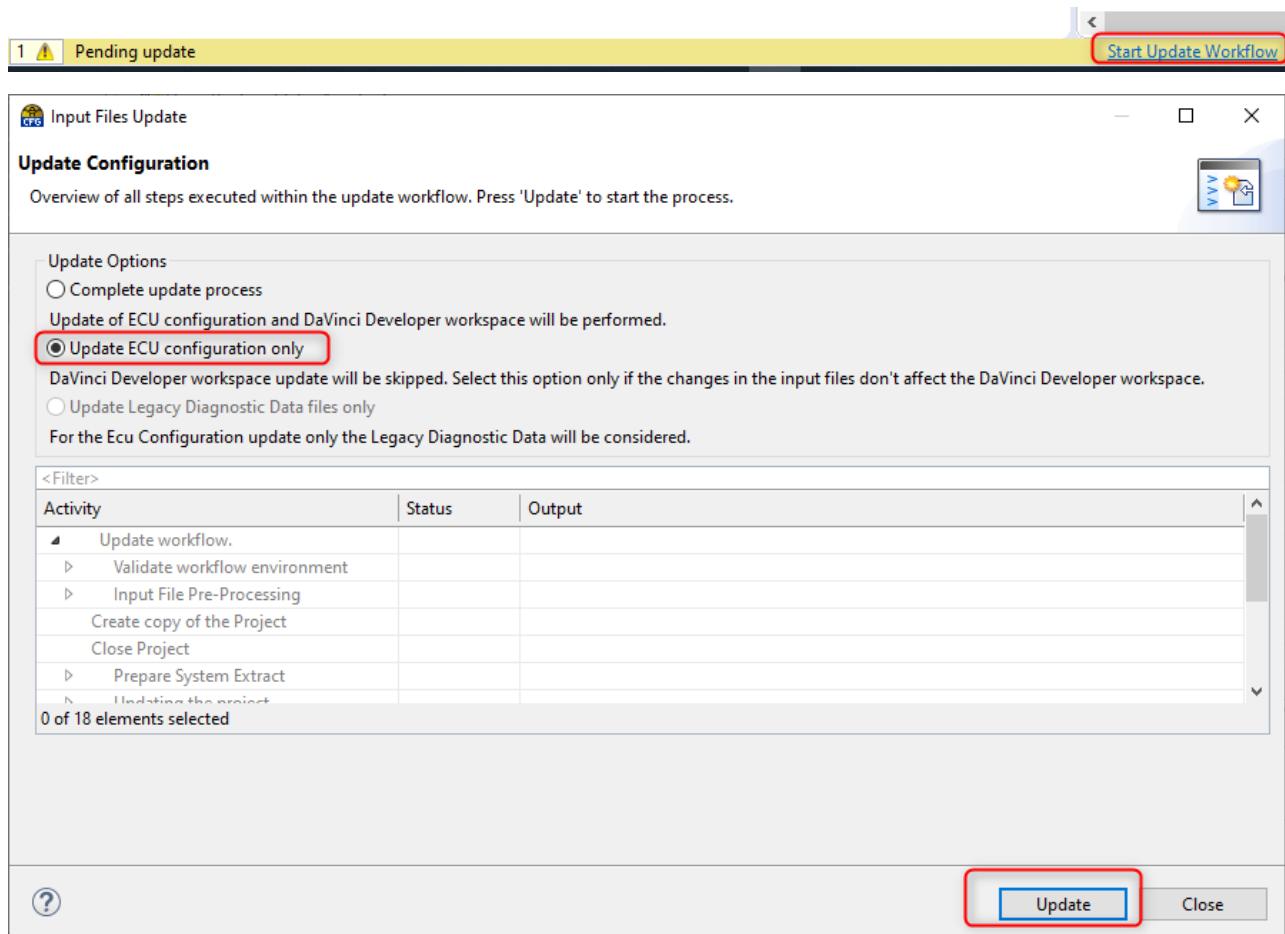
11.5 Steps on Davinci Configurator

11.5.1 Step 10: Update ARXML Files in Davinci Cfg

⚠ Sync Needed

Please consider that after modifying (and saving) the ARXML file, an Update shall be need, please follow the recommended Updating process by clicking the Yellow ribbon in the Vector Davinci Configurator Tool.

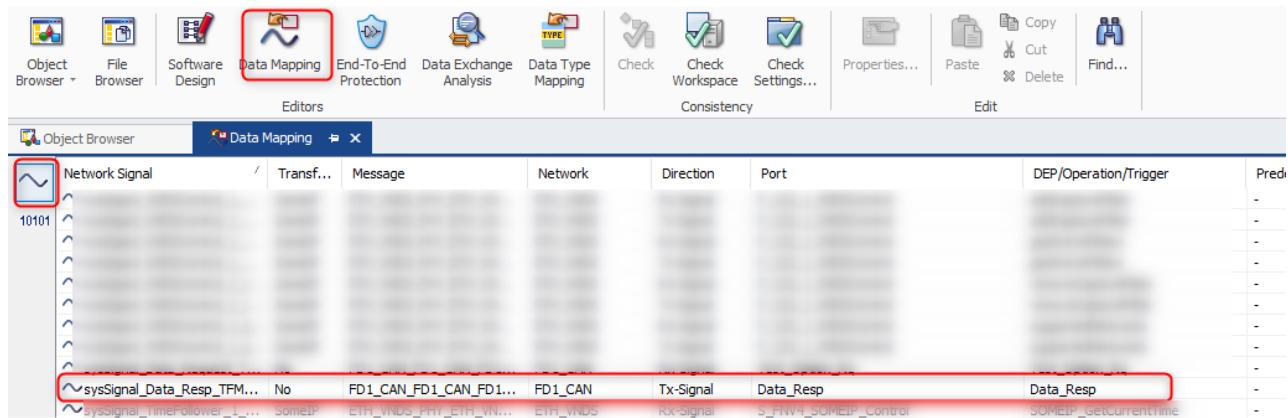
Please make use to follow the Update show in the Following Images:



11.6 Steps on Davinci Developer

11.6.1 Step 11: Map the new Signal

After Step 10 the Mapping shall be done automatically, please verify that the mapping looks like the following image:



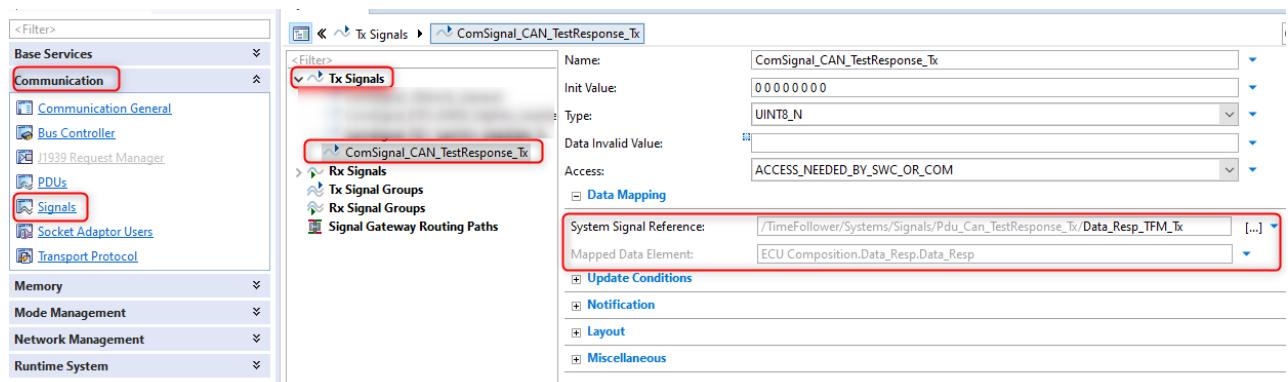
11.7 Steps on Davinci Configurator

11.7.1 Step 12: Final Connections and Verification

Using the Left panel move the following path:

Communication → Signals → Tx Signal → Select the ComSignals from step 6 → Data Mapping.

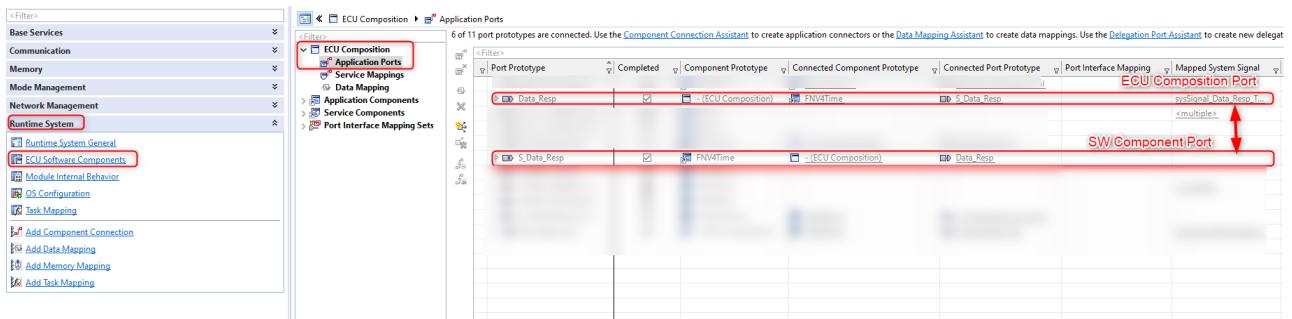
- System Signal Reference: (**Select the System Signal from Step 9**)
- Mapped Data Element: (**Shall be updated as well**)



Using the Left panel move the following path:

Runtime Systems → ECU Software Components → ECU Composition → Application Ports.

Make sure the System Signal is mapped to the correct port only on the ECU Composition Port:



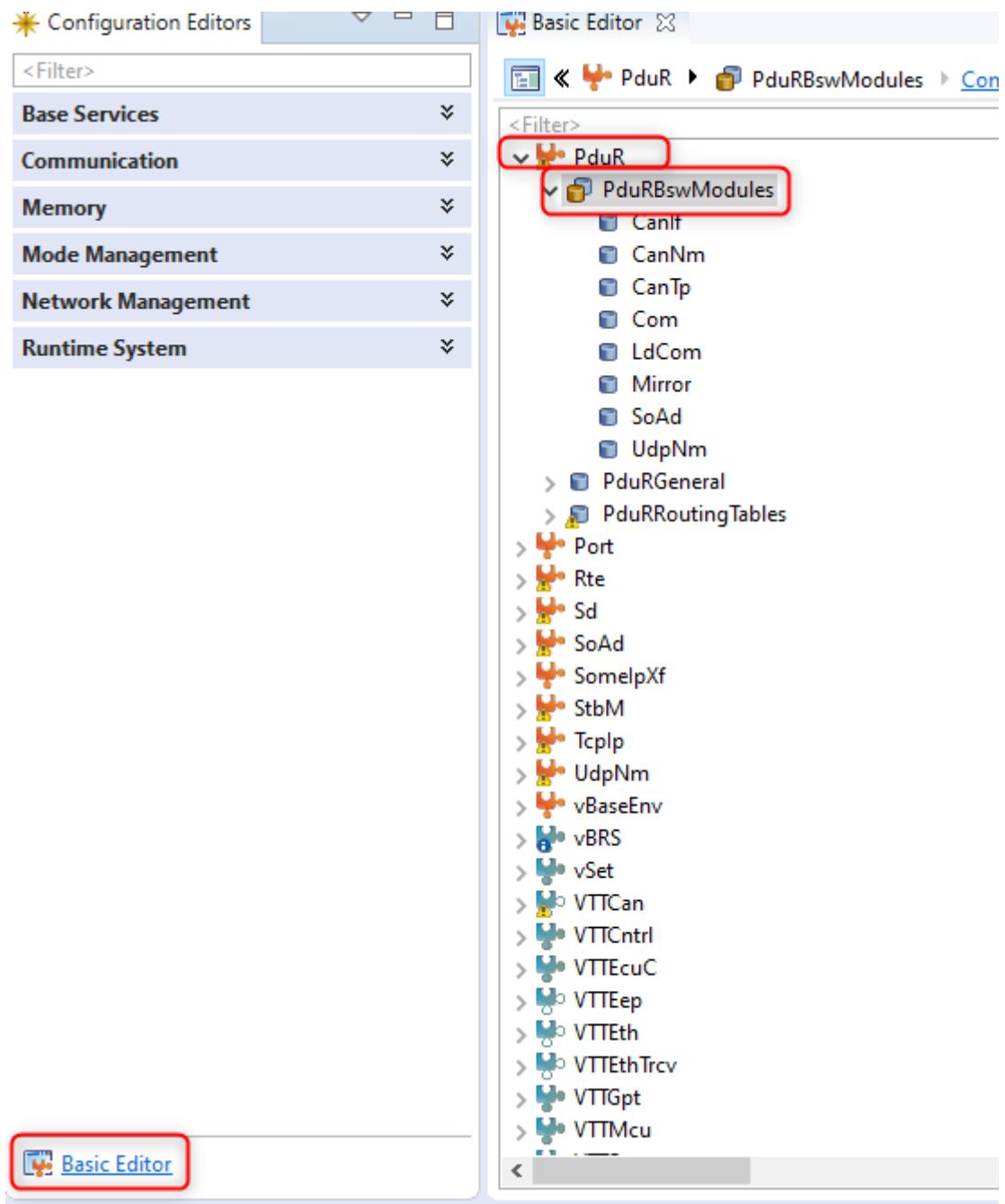
11.8 Troubleshooting

11.8.1 UL (Upper Layer) Indication related issues:

Some default configuration from Autosar Project don't provide any Tx Confirmation or Upper Module Configuration, hence some validation errors:

To modify this please refer to the following Path:

Basic Editor → PduR → PduRBswModules:



In here you can configure any missing Upper Layer or Tx Confirmation issues, like the image below:

Short Name:	Com
Bsw Module Ref:	/ActiveEcuC/Com
Cancel Receive:	<input type="checkbox"/> *
Cancel Transmit:	<input type="checkbox"/> *
Change Parameter Request Api:	<input type="checkbox"/> *
Communication Interface:	<input checked="" type="checkbox"/>
Lower Module:	<input type="checkbox"/> *
Os Application Ref:	<input type="checkbox"/>
Retransmission:	<input type="checkbox"/>
Transport Protocol:	<input type="checkbox"/> *
Triggertransmit:	<input checked="" type="checkbox"/>
Tx Confirmation:	<input checked="" type="checkbox"/>
Upper Module:	<input checked="" type="checkbox"/>
Use Tag:	<input type="checkbox"/> *

Short Name:	CanIf
Bsw Module Ref:	/ActiveEcuC/CanIf
Cancel Receive:	<input type="checkbox"/> *
Cancel Transmit:	<input type="checkbox"/> *
Change Parameter Request Api:	<input type="checkbox"/> *
Communication Interface:	<input checked="" type="checkbox"/>
Lower Module:	<input checked="" type="checkbox"/>
Os Application Ref:	
Retransmission:	<input type="checkbox"/>
Transport Protocol:	<input type="checkbox"/> *
Triggertransmit:	<input type="checkbox"/> *
Tx Confirmation:	<input checked="" type="checkbox"/>
Upper Module:	<input type="checkbox"/> *
Use Tag:	<input type="checkbox"/> *

12_10_Multiple Signal in CAN Message Configuration

⚠ Deviation Only

This is a deviation for when the CAN Message containing different Signals in the same Message. Hence this guide will only provide the extra steps that need to be done beside the other guides

- 8_Create a New Tx CAN Signal²
- 7_Create a New Rx CAN Signal³

- Prerequisites(see page 165)
- Steps(see page 165)
- Steps on Davinci Configurator(see page 165)
 - After creating a new Pdu (Step 1)(see page 165)
 - - Continue guide execution -(see page 167)
 - After creating a new ComSignals (Step 5 or Step 6)(see page 167)
 - - Continue guide execution -(see page 169)
 - After creating a new ComIPdus (Step 6 or 7)(see page 169)
 - - Continue guide execution -(see page 173)
- Troubleshooting(see page 173)

12.1 Prerequisites

- Make sure you have an up and running CAN configuration.
- Make sure to have a CANoe simulation or any other kind of verification tool.

12.2 Steps

12.3 Steps on Davinci Configurator

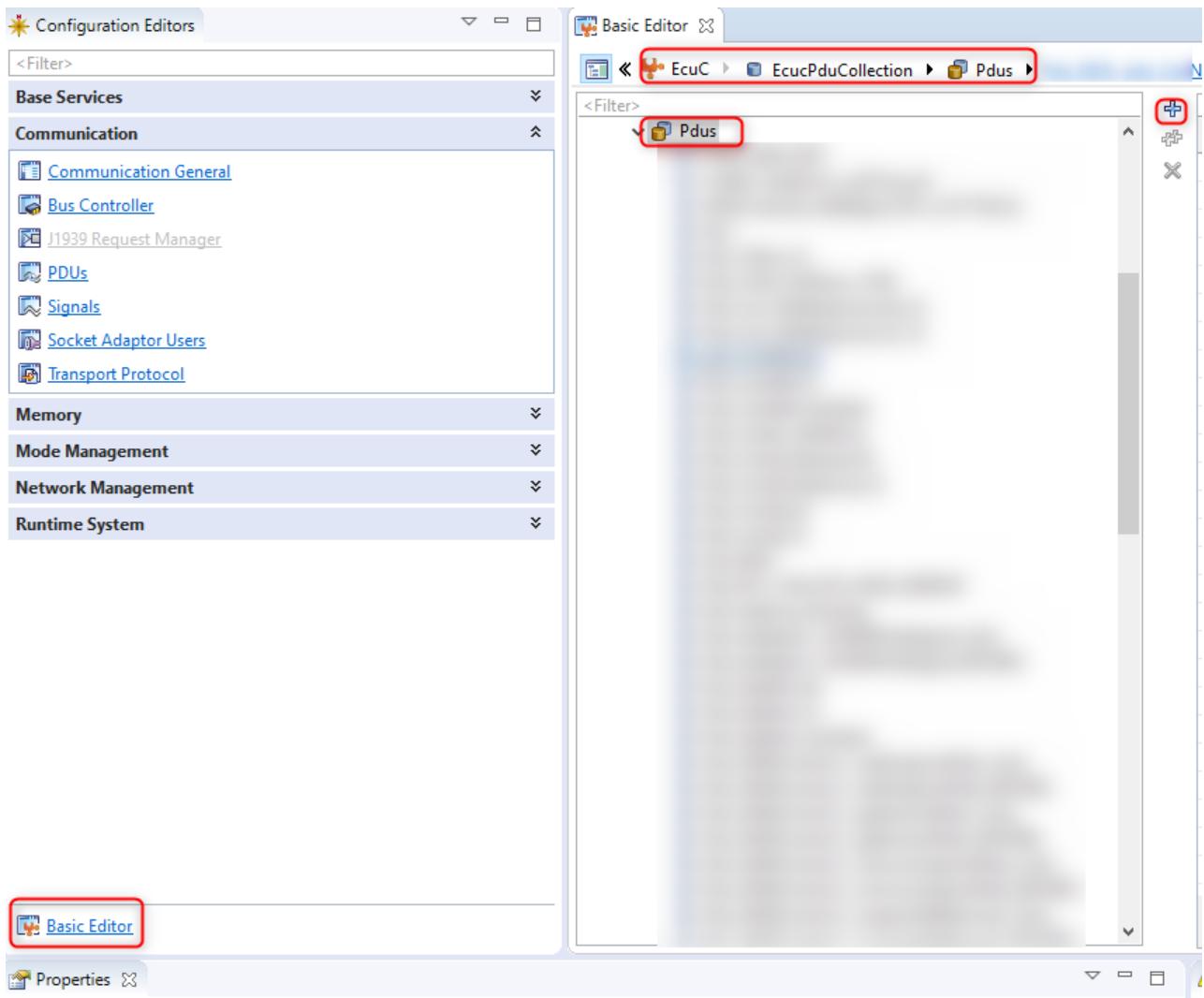
12.3.1 After creating a new Pdu (Step 1)

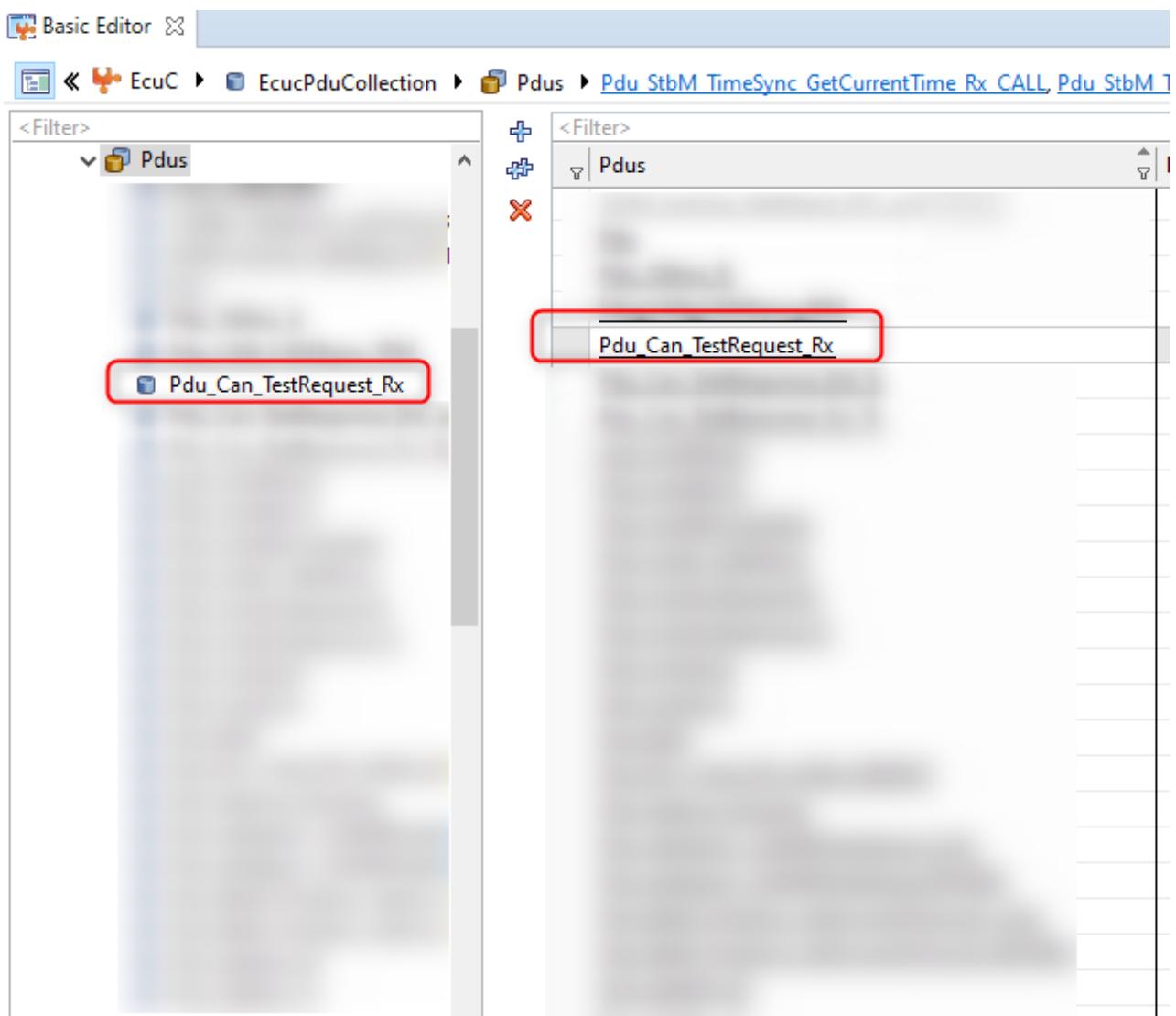
If the Message to be created has more than one Signal please create a single new Pdu for all of the necessary ones.

Go to Basic Editor → EcuC → EcucPduCollection → Pdu and Click on the "+" sign.

² https://luxproject.luxoft.com/confluence/display/FORDFNV4FO/8+_Create+a+New+Tx+CAN+Signal

³ https://luxproject.luxoft.com/confluence/display/FORDFNV4FO/7+_Create+a+New+Rx+CAN+Signal



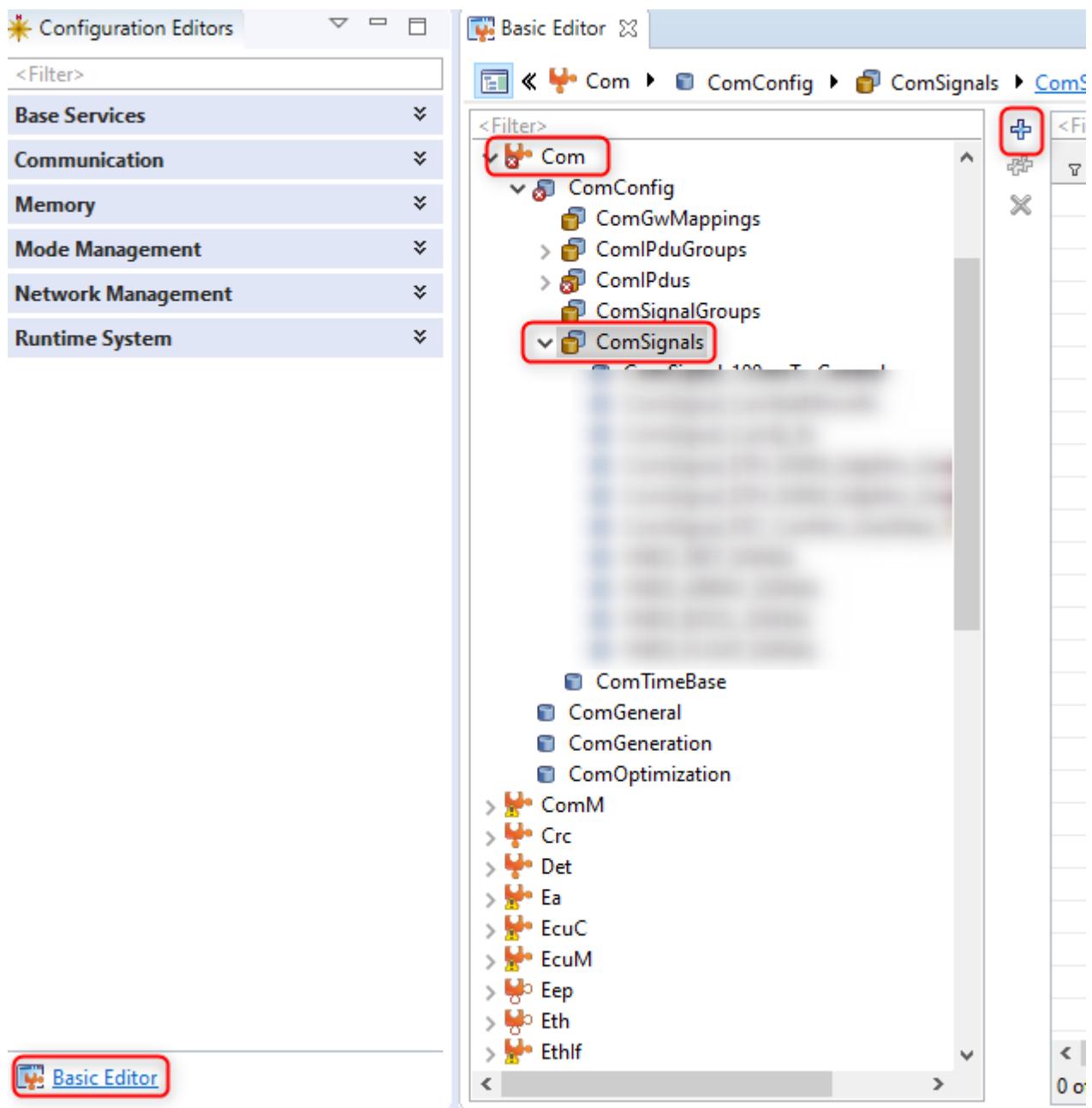


12.3.2 - Continue guide execution -

12.3.3 After creating a new ComSignals (Step 5 or Step 6)

Create one of each ComSignal for each Signal of the necessary for the message.

Go to Basic Editor → Com → ComConfig → ComSignals and Click the "+" sign



Please consider the Bit Position, Bit size and Endianess when adding them.

Top Screenshot: ComSignal_CAN_TestRequest_Rx Configuration

Short Name:	ComSignal_CAN_TestRequest_Rx
Bit Position:	0
Bit Size [Bit]:	32
Data Invalid Action:	
Error Notification:	
First Timeout [ms]:	
Handle Id:	1
Invalid Notification:	
Notification:	Rte_COMCb_ComSignal_CAN_TestRequest_Rx
Rx Data Timeout Action:	NONE
Rx Data Timeout Substitution Value:	
Signal Access:	ACCESS_NEEDED_BY_OTHER
Signal Data Invalid Value:	
Signal Endianness:	LITTLE_ENDIAN
Signal Init Value:	0
Signal Length [Byte]:	4

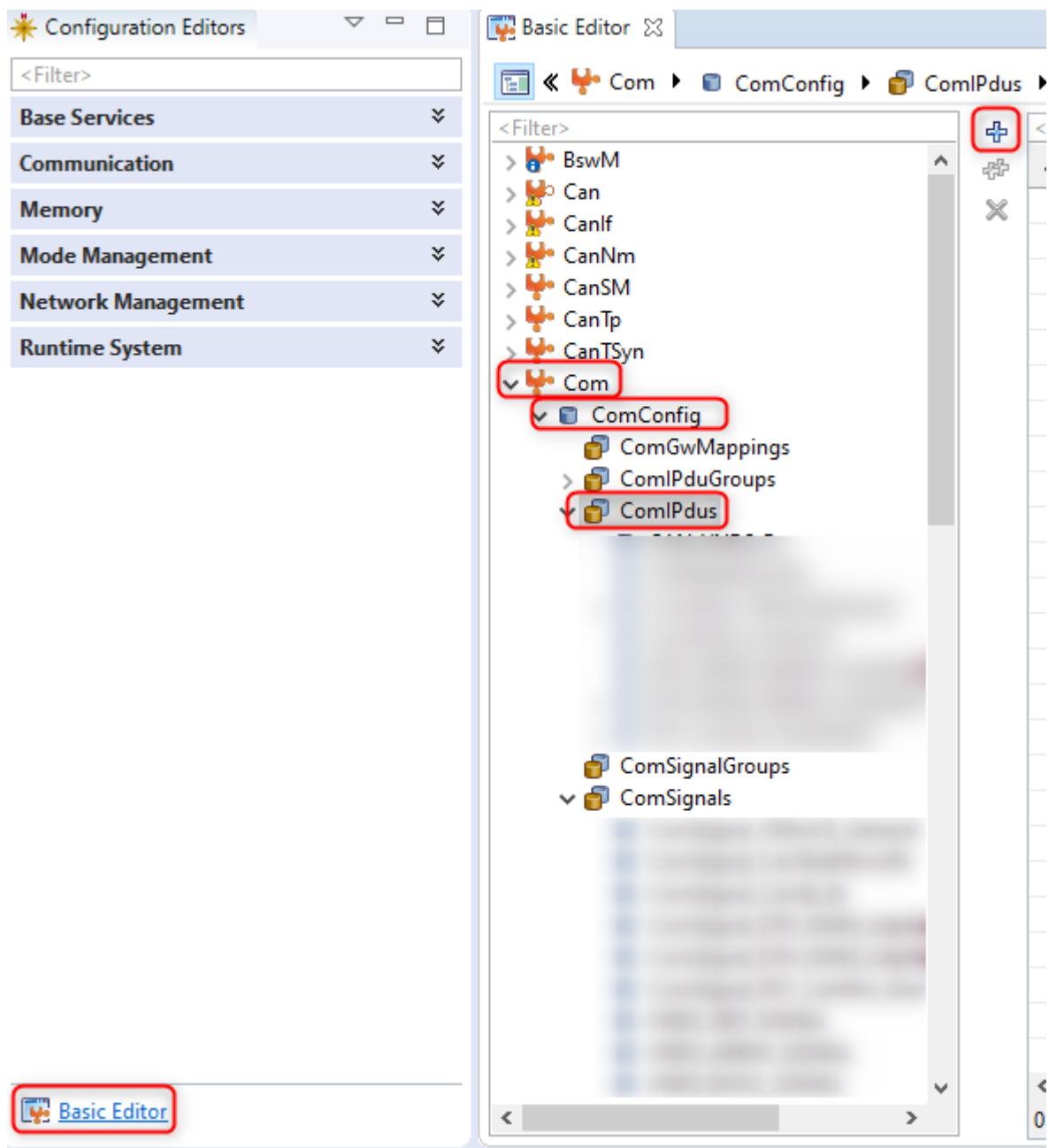
Bottom Screenshot: ComSignal_CAN_TestRequest_2_Rx Configuration

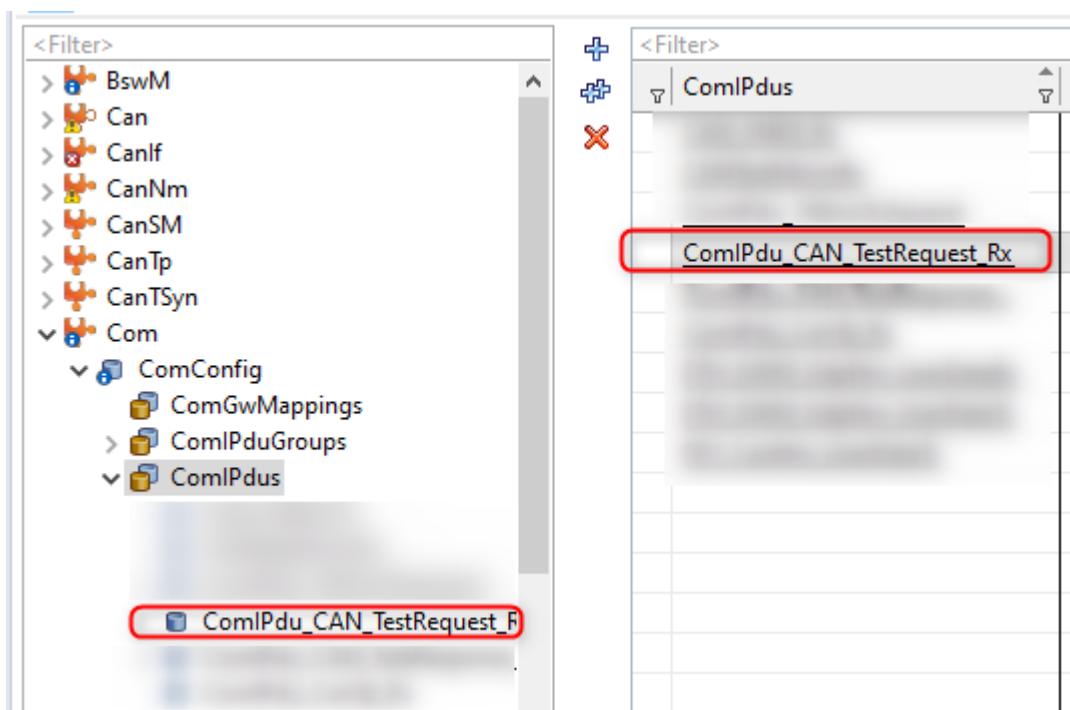
Short Name:	ComSignal_CAN_TestRequest_2_Rx
Bit Position:	32
Bit Size [Bit]:	32
Data Invalid Action:	
Error Notification:	
First Timeout [ms]:	
Handle Id:	0
Invalid Notification:	
Notification:	Rte_COMCb_ComSignal_CAN_TestRequest_Rx
Rx Data Timeout Action:	NONE
Rx Data Timeout Substitution Value:	
Signal Access:	ACCESS_NEEDED_BY_OTHER
Signal Data Invalid Value:	
Signal Endianness:	LITTLE_ENDIAN
Signal Init Value:	0
Signal Length [Byte]:	4

12.3.4 - Continue guide execution -

12.3.5 After creating a new ComIPdus (Step 6 or 7)

Go to Basic Editor → Com → ComConfig → ComIPdus and Click the "+" sign.





After the creating the ComIPdus with a meaningfull and unique name, configure the following fields:

- IPdu Direction: **RECEIVE**
- Pdu Id Ref: **(Use the previously one created on step 1)**
- IPdu Signal Ref: **(Use the previously one created on step 5)**

Short Name:	ComIPdu_CAN_TestRequest_Rx
IPdu Callout:	
IPdu Cancellation Support:	<input type="checkbox"/> *
IPdu Direction:	RECEIVE
IPdu Handle Id:	2
IPdu Signal Processing:	DEFERRED
IPdu Trigger Transmit Callout:	
IPdu Type:	NORMAL
Pdu Id Ref:	/ActiveEcuC/EcuC/EcucPduCollection/Pdu_Can_TestRequest_Rx

IPdu Group Ref

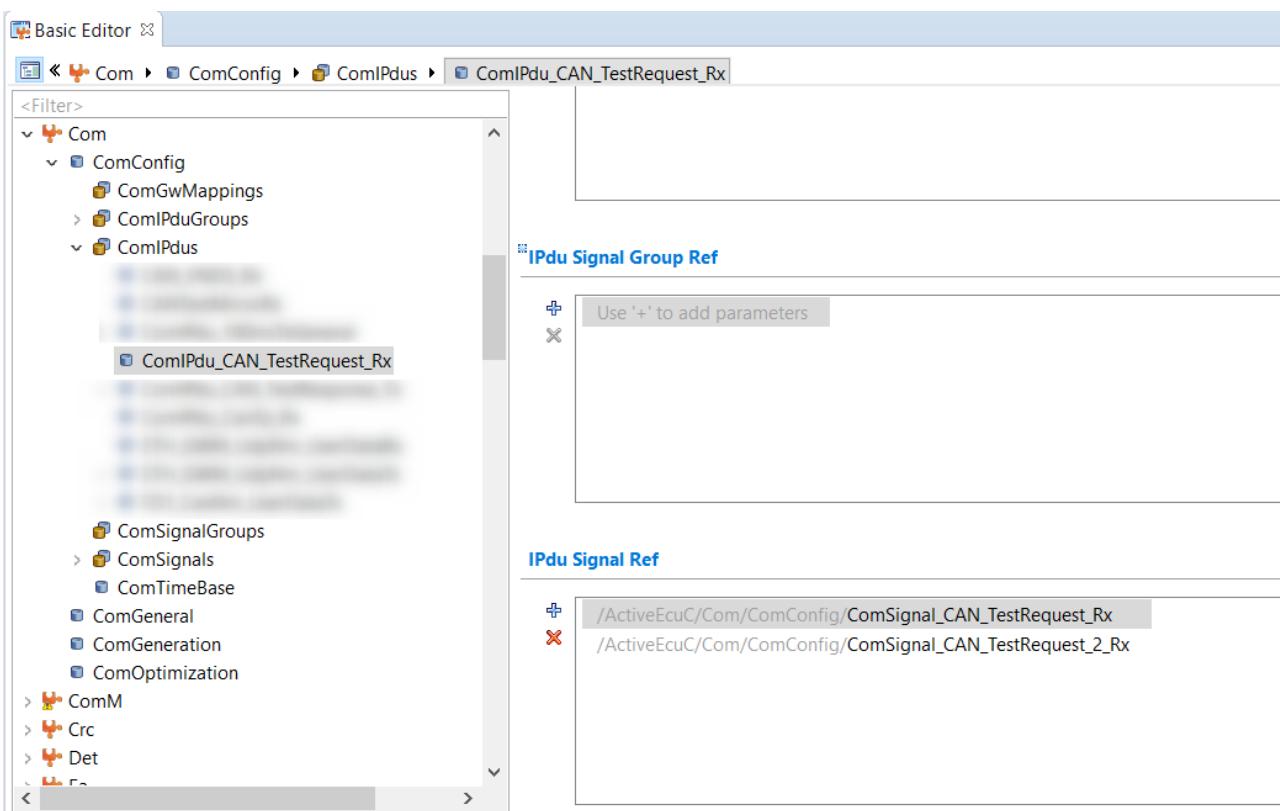
+ Use '+' to add parameters

X

IPdu Signal Group Ref

+ Use '+' to add parameters

X



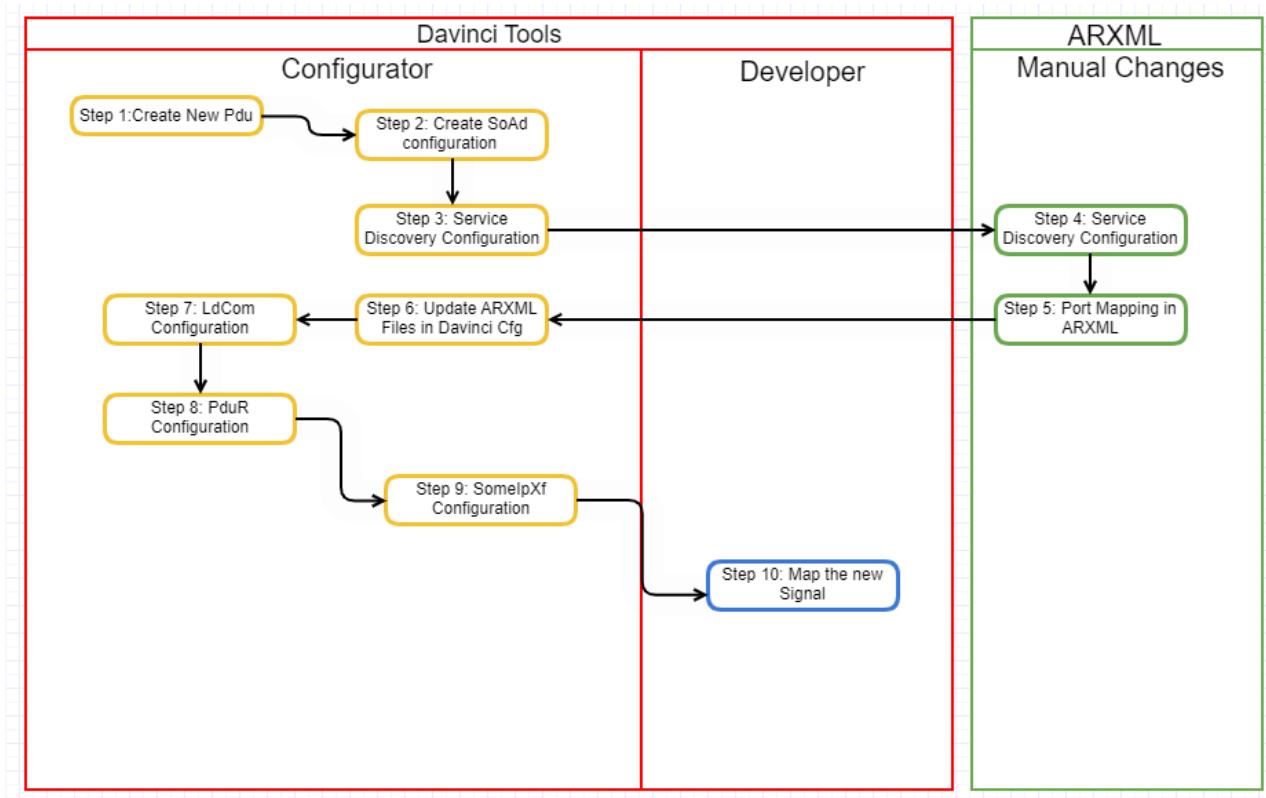
12.3.6 - Continue guide execution -

12.4 Troubleshooting

13_11_Create a New SOME/IP Signal

- Workflow Diagram(see page 174)
- Prerequisites(see page 175)
- Steps on Davinci Configurator(see page 175)
 - Step 1: Create a new Pdu(see page 175)
 - Step 2: Create SoAd configuration(see page 177)
 - Step 3: Service Discovery Configuration(see page 178)
- Steps to be performed directly on the ARXML File(see page 180)
 - Step 4: Create Signals in the ARXML(see page 180)
 - Step 5: Port Mapping on ARXML(see page 180)
- Steps on Davinci Configurator(see page 181)
 - Step 6: Update ARXML Files in Davinci Cfg(see page 181)
 - Step 7: LdCom Configuration(see page 182)
 - Step 8: PduR Configuration(see page 183)
 - Step 9: SomeipXf Configuration(SOME/IP Transformer):(see page 185)
- Steps on Davinci Developer(see page 186)
 - Step 10: Map the new Signal(see page 186)
- Troubleshooting(see page 186)

13.1 Workflow Diagram



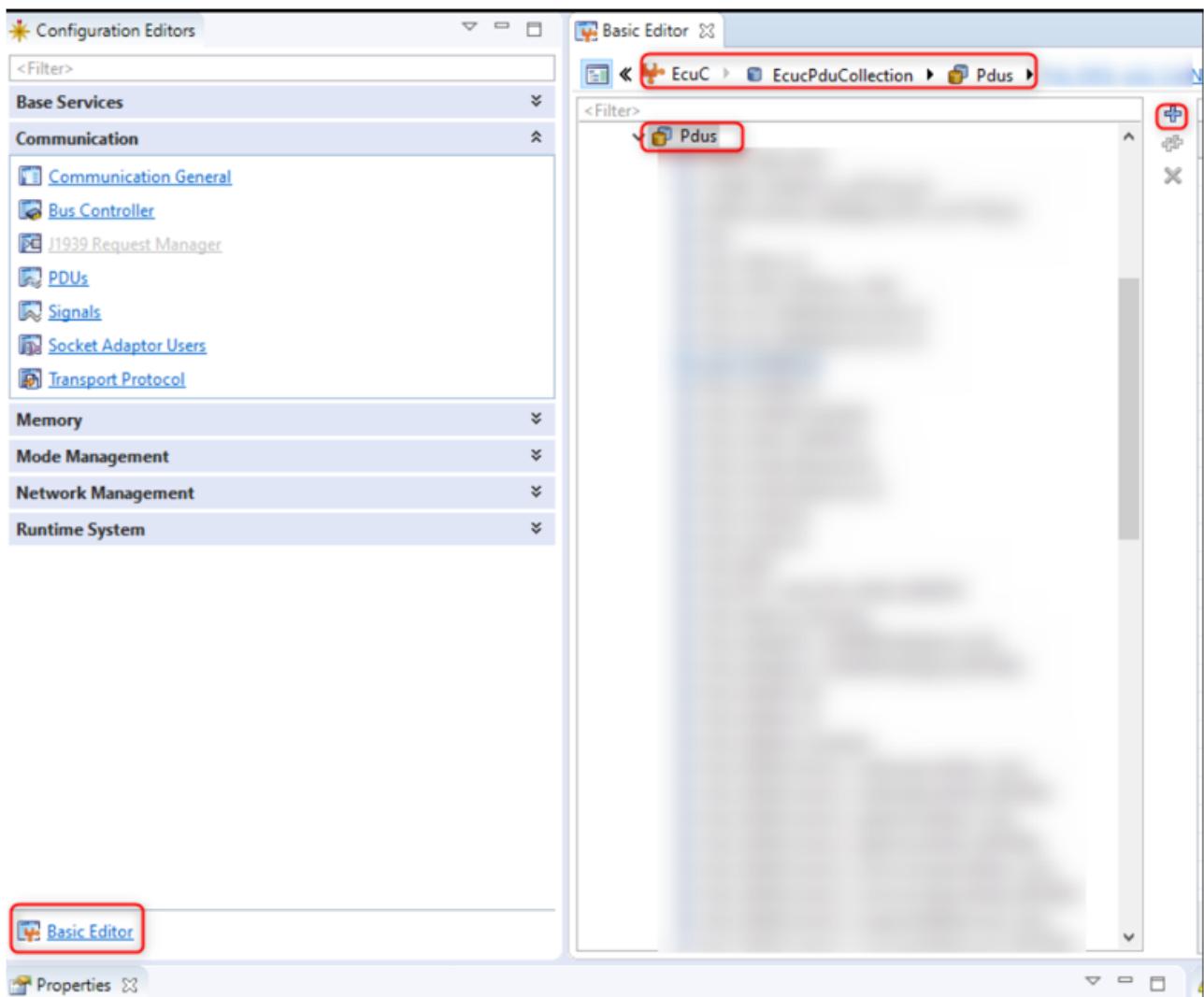
13.2 Prerequisites

1. Make sure you have an up an running ETH configuration.
2. Make sure you are able to transmit a any ETH Message (not the one to be configure in this recipe).
3. Make sure to have a CANoe simulation or any other kind of verification tool

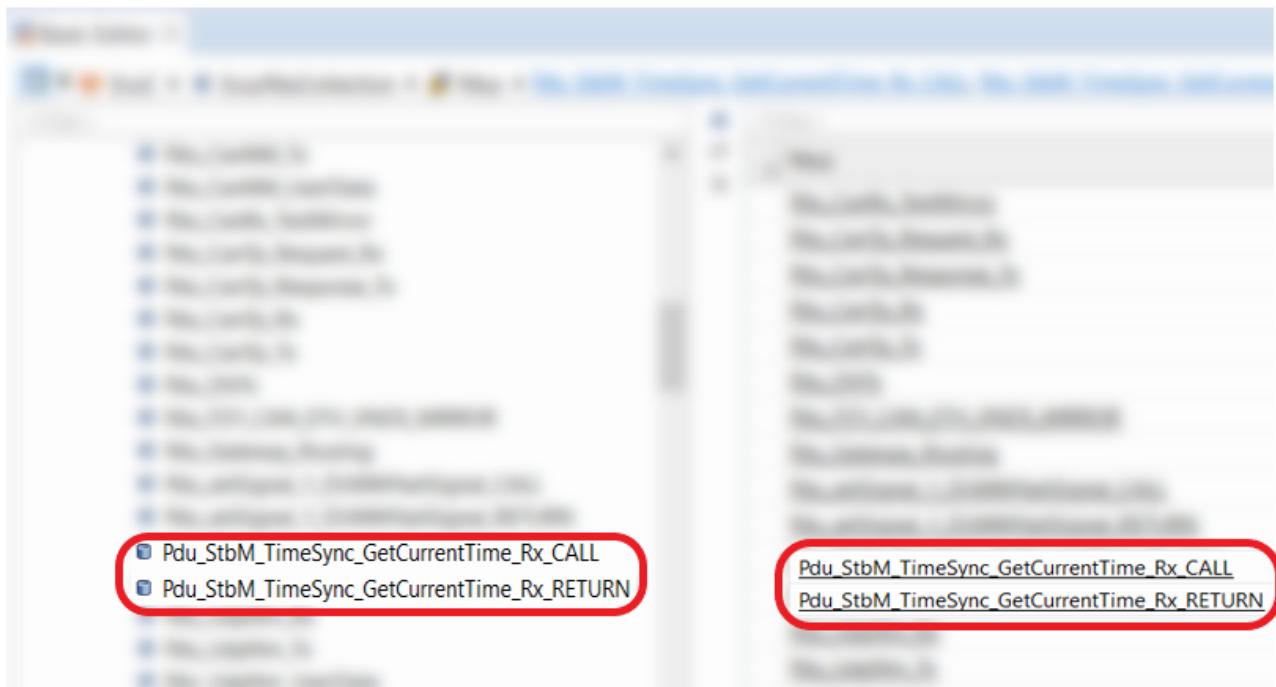
13.3 Steps on Davinci Configurator

13.3.1 Step 1: Create a new Pdu

Go to Basic Editor → EcuC → EcucPduCollection → Pdu and Clic on the "+" sign.



Need to create a CALL and a RETURN Pdu.

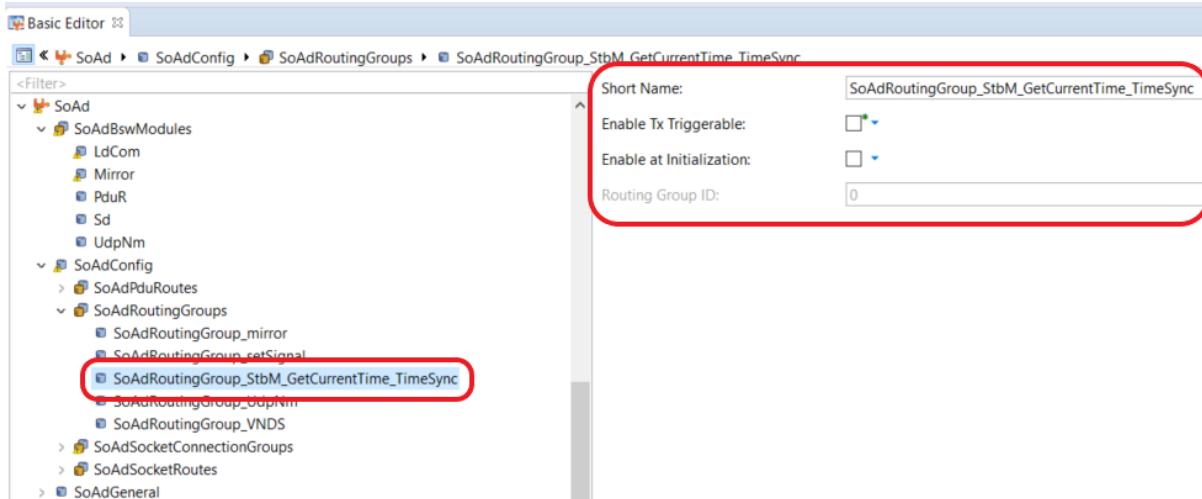


After creating the Pdu with a meaningful and unique names, configure the Data Length and any other field needed.

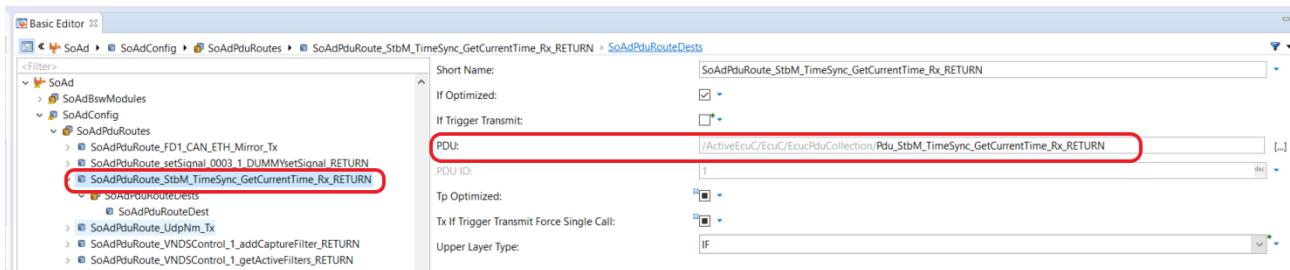
Pdu Name	Pdu Length [Byte]
Pdu_StbM_TimeSync_GetCurrentTime_Rx_CALL	32
Pdu_StbM_TimeSync_GetCurrentTime_Rx_RETURN	23

13.3.2 Step 2: Create SoAd configuration

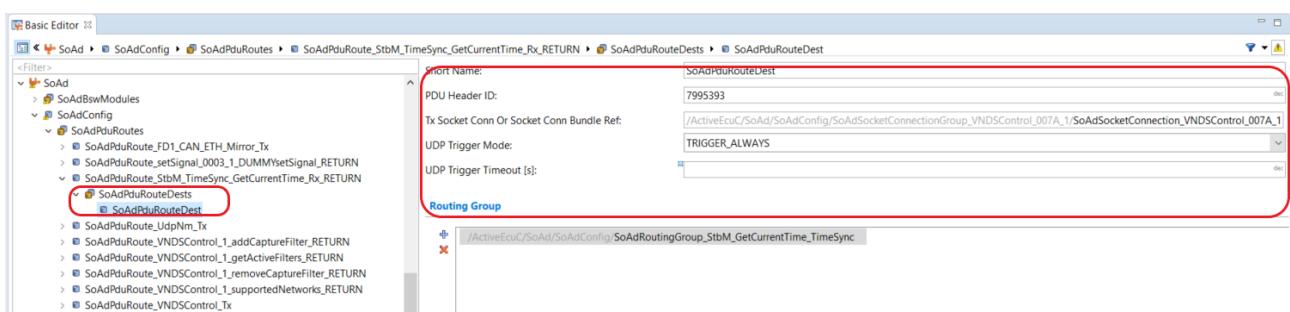
Create a specific routing group. This is required to forward PDUs from the upper layer to TCP Or UDP socket of the TCP/IP.



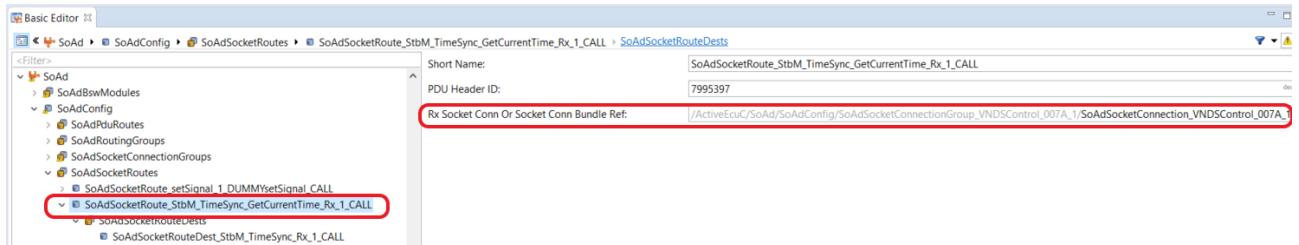
Create the Socket Adaptor PDUs Routes. To describe the path of a PDU from an upper layer of the SoAd to the socket in the TCP/IP For transmission(RETURN).



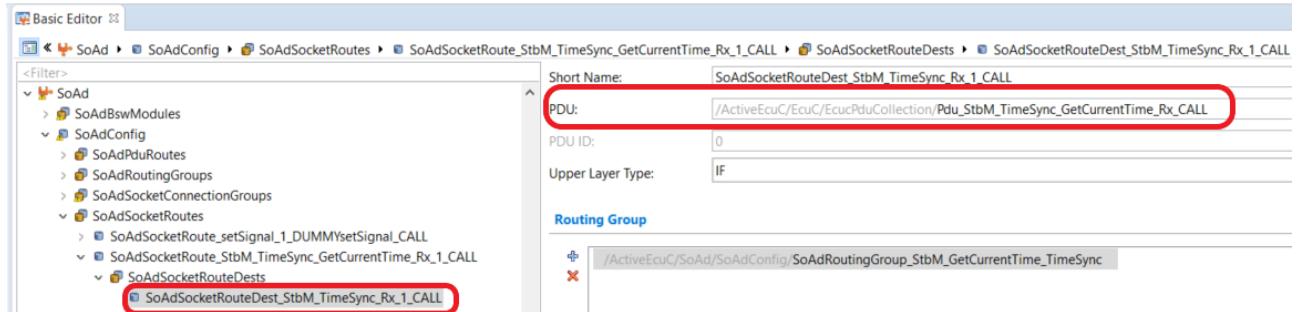
Specify the PDU route destination



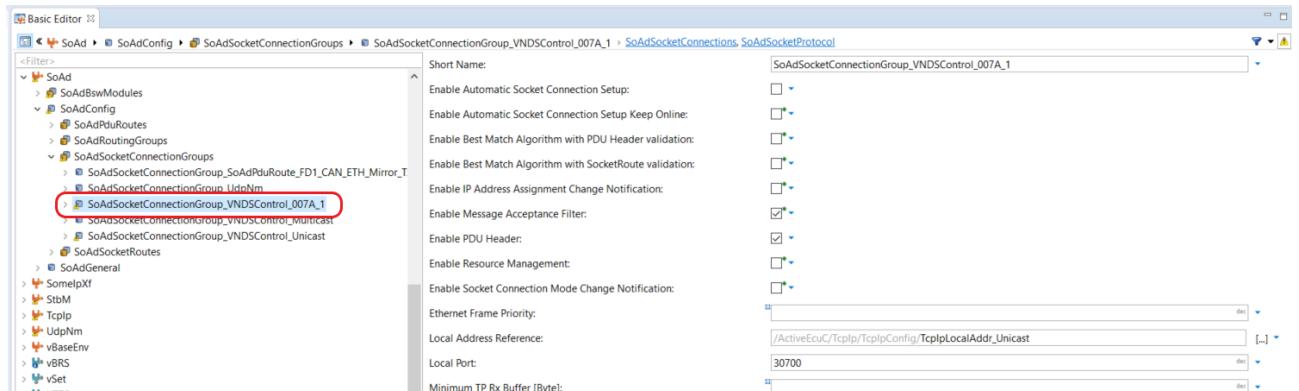
Create the Socket Adaptor Socket Route. To describe the path of a PDU from a socket to the TCP/IP for reception(CALL).



Specify the PDU route destination

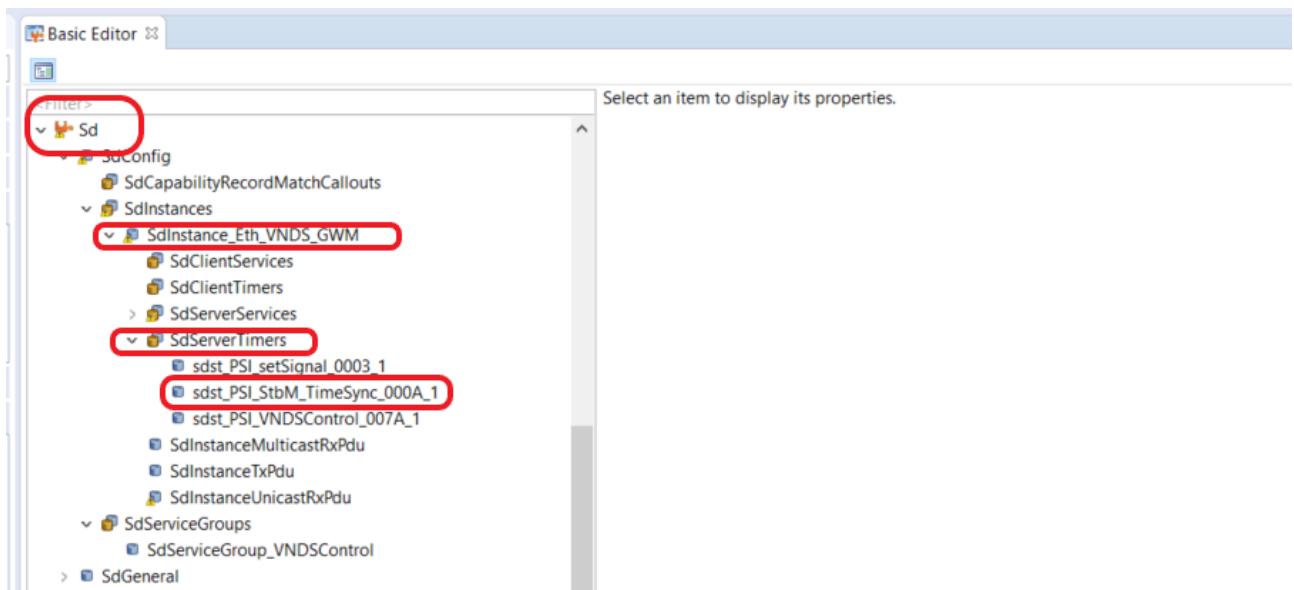


In this case data will be received/transmitted VNDSControl_007A_1

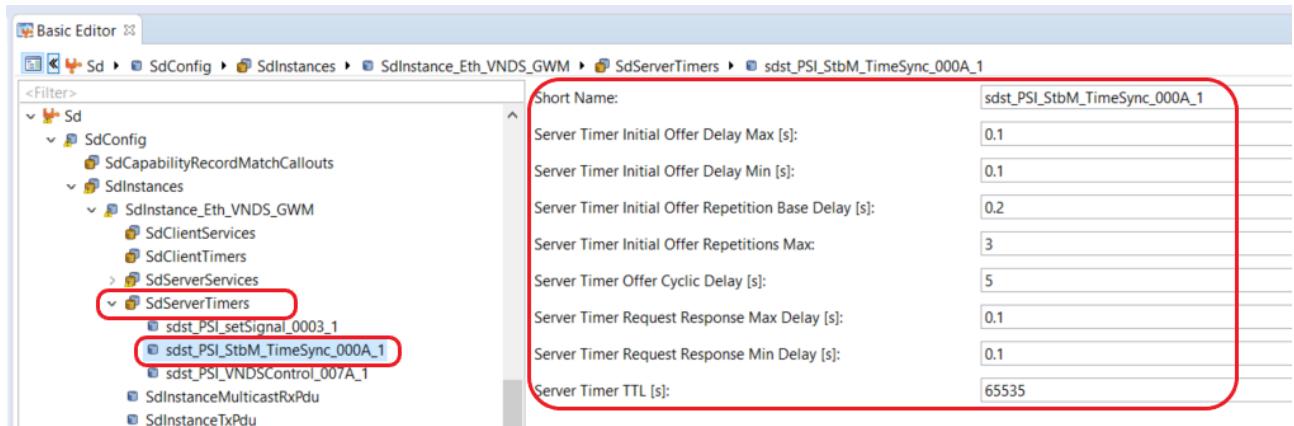


13.3.3 Step 3: Service Discovery Configuration

In Sd create the proper Timer used by the service Discovery module for server services.

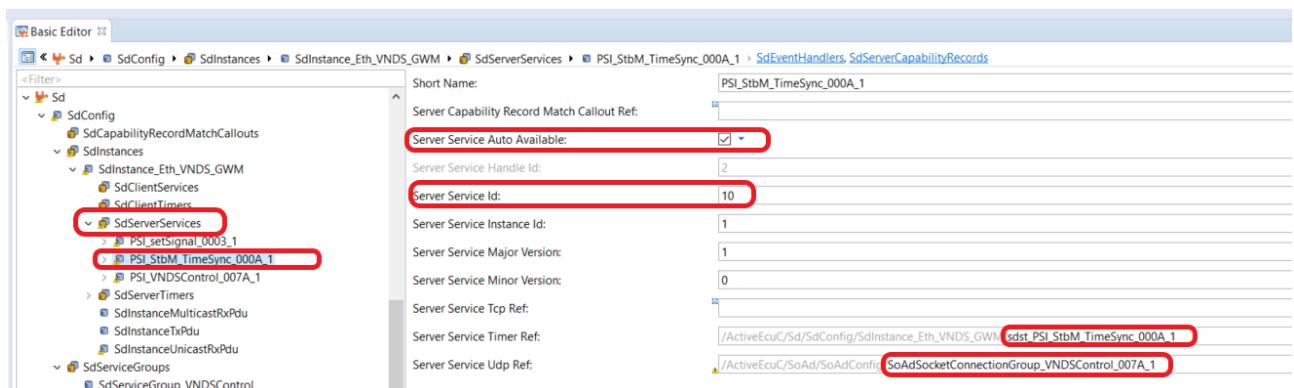


Configure the Sd timer



Configure the Service discovery Server to define the parameters contain by the Service

Enable Server Service Auto Available. This option makes the service to be available in the execution. And assign a service Id in this case Id is 0x0A.



13.4 Steps to be performed directly on the ARXML File

13.4.1 Step 4: Create Signals in the ARXML

Signals need to be added on TimeFollower_Systems.arxml manually.

Required Elements for CALL and RETURN:

- Signal
- System Signal
- Pdu

```
<AR-PACKAGE UUID="B162A5E39-17CB-4BD6-892E-E2498C428284">
<SHORT-><ELEMENTS>
  <ELEMENT>
    <SIGNAL UUID="958ea168-8c63-426d-8934-04c461744cf1">
      <SHORT-NAME>TimeFollower_1_StbM_TimeSync_GetCurrentTime_Rx_CALL</SHORT-NAME>
      <DATA-TRANSFORMATIONS>
        <DATA-TRANSFORMATION-REF-CONDITIONAL>
          <DATA-TRANSFORMATION-REF DEST="DATA-TRANSFORMATION"/>DataTransformationSets/DataTransformationSet/SomeIpTransformerBigEndian</DATA-TRANSFORMATION-REF>
        <DATA-TRANSFORMATION-REF-CONDITIONAL>
        <DATA-TRANSFORMATIONS>
          <DATA-TYPE-POLICY>TRANSFORMING-I-SIGNAL</DATA-TYPE-POLICY>
          <LENGTH>1</LENGTH>
        <SYSTEM-SIGNAL-REF DEST="SYSTEM-SIGNAL">/TimeFollower/Systems/Signals/sysSignal_TimeFollower_1_StbM_TimeSync_GetCurrentTime_Rx_CALL</SYSTEM-SIGNAL-REF>
      <TRANSFORMATION-I-SIGNAL-PROPS>
        <SOMEIP-TRANSFORMATION-I-SIGNAL-PROPS>
          <SOMEIP-TRANSFORMATION-I-SIGNAL-PROPS-VARIANTS>
            <SOMEIP-TRANSFORMATION-I-SIGNAL-PROPS-CONDITIONAL>
              <TRANSFORMER-REF DEST="TRANSFORMATION-TECHNOLOGY"/>DataTransformationSets/DataTransformationSet/SomeIpTransformer_BigEndian</TRANSFORMER-REF>
              <MESSAGE-TYPE>REQUEST</MESSAGE-TYPE>
            <SOMEIP-TRANSFORMATION-I-SIGNAL-PROPS-CONDITIONAL>
            <SOMEIP-TRANSFORMATION-I-SIGNAL-PROPS-VARIANTS>
            <SOMEIP-TRANSFORMATION-I-SIGNAL-PROPS>
          <TRANSFORMATION-I-SIGNAL-PROPS>
        </I-SIGNAL>
      <I-SIGNAL UUID="958ea168-8c63-426d-8934-04c461744cf1">
        <SHORT-NAME>TimeFollower_1_StbM_TimeSync_GetCurrentTime_Rx_RETURN</SHORT-NAME>
        <DATA-TRANSFORMATIONS>
          <DATA-TRANSFORMATION-REF-CONDITIONAL>
            <DATA-TRANSFORMATION-REF DEST="DATA-TRANSFORMATION"/>DataTransformationSets/DataTransformationSet/SomeIpTransformerBigEndian</DATA-TRANSFORMATION-REF>
          </DATA-TRANSFORMATION-REF-CONDITIONAL>
        <DATA-TRANSFORMATIONS>
          <DATA-TYPE-POLICY>TRANSFORMING-I-SIGNAL</DATA-TYPE-POLICY>
          <LENGTH>1</LENGTH>
        <SYSTEM-SIGNAL-REF DEST="SYSTEM-SIGNAL">/TimeFollower/Systems/Signals/sysSignal_TimeFollower_1_StbM_TimeSync_GetCurrentTime_Rx_RETURN</SYSTEM-SIGNAL-REF>
      <TRANSFORMATION-I-SIGNAL-PROPS>
        <SOMEIP-TRANSFORMATION-I-SIGNAL-PROPS>
          <SOMEIP-TRANSFORMATION-I-SIGNAL-PROPS-VARIANTS>
            <SOMEIP-TRANSFORMATION-I-SIGNAL-PROPS-CONDITIONAL>
              <TRANSFORMER-REF DEST="TRANSFORMATION-TECHNOLOGY"/>DataTransformationSets/DataTransformationSet/SomeIpTransformer_BigEndian</TRANSFORMER-REF>
              <MESSAGE-TYPE>RESPONSE</MESSAGE-TYPE>
            <SOMEIP-TRANSFORMATION-I-SIGNAL-PROPS-CONDITIONAL>
            <SOMEIP-TRANSFORMATION-I-SIGNAL-PROPS-VARIANTS>
            <SOMEIP-TRANSFORMATION-I-SIGNAL-PROPS>
          <TRANSFORMATION-I-SIGNAL-PROPS>
        </I-SIGNAL>
      <I-SIGNAL I-PDU UUID="61-a8afff-7a-4d72-8692-2f5e705d4538">
        <SHORT-NAME>pduTimeFollower_1_StbM_TimeSync_GetCurrentTime_Rx_CALL</SHORT-NAME>
        <LENGTH>0</LENGTH>
      <I-SIGNAL-TO-I-PDU-MAPPINGS>
        <I-SIGNAL-TO-I-PDU-MAPPING UUID="8038ED057-1C79-4FFA-8065-FE6608966DA3">
          <SHORT-NAME>TimeFollower_1_StbM_TimeSync_GetCurrentTime_Rx_CALL</SHORT-NAME>
          <I-SIGNAL-REF DEST="I-SIGNAL"/>/TimeFollower/Systems/Signals/TimeFollower_1_StbM_TimeSync_GetCurrentTime_Rx_CALL</I-SIGNAL-REF>
        <PACKING-BYTE-ORDER>OPQUEL</PACKING-BYTE-ORDER>
        <START-POSITION>0</START-POSITION>
        <TRANSFER-PROPERTY>TRIGGERED</TRANSFER-PROPERTY>
      </I-SIGNAL-TO-I-PDU-MAPPING>
    </I-SIGNAL>
  </ELEMENT>
</ELEMENTS>
```

13.4.2 Step 5: Port Mapping on ARXML

Open Gateway.arxml to add signal mapping into the mapping section.

```
<MAPPINGS>
  <SYSTEM-MAPPING UUID="462D4855-48B1-4F5E-B572-DC454E5329A7">
    <SHORT-NAME>Mapping</SHORT-NAME>
    <DATA-MAPPINGS>
```

```

<CLIENT-SERVER-TO-SIGNAL-MAPPING>
  <CALL-SIGNAL-REF DEST="SYSTEM-SIGNAL">/TimeFollower/Systems/Signals sysSignal_TimeFollower_1_StbM_TimeSync_GetCurrentTime_Rx_CALL /CALL-SIGNAL-REF>
  <CLIENT-SERVER-OPERATION-IREF>
    <CONTEXT-COMPOSITION-REF DEST="ROOT-SW-COMPOSITION-PROTOTYPE"/>/Systems/VNDS/EcuSwComposition /CONTEXT-COMPOSITION-REF>
    <CONTEXT-COMPONENT-REF DEST="SW-COMPONENT-PROTOTYPE"/>/ECUCompositionTypes/ECU_Composition_FNV4Time /CONTEXT-COMPONENT-REF>
    <CONTEXT-PORT-REF DEST="F-PORT-PROTOTYPE"/>/TimeFollower/FNV4_SWType/SOMEIP_Control /CONTEXT-PORT-REF>
    <TARGET-OPERATION-REF DEST="CLIENT-SERVER-OPERATION">/TimeFollower/Interfaces/FNV4_SOMEIP_Control SOMEIP_GetCurrentTime /TARGET-OPERATION-REF>
  <RETURN-SIGNAL-REF DEST="SYSTEM-SIGNAL">/TimeFollower/Systems/Signals sysSignal_TimeFollower_1_StbM_TimeSync_GetCurrentTime_Rx_RETURN /RETURN-SIGNAL-REF>
</CLIENT-SERVER-TO-SIGNAL-MAPPING>

```

Add the FIBEX-ELEMENT for PDUs and Signals.

```

<FIBEX-ELEMENT-REF-CONDITIONAL>
  <FIBEX-ELEMENT-REF DEST="I-SIGNAL-I-PDU">/TimeFollower/Systems/Signals/pduTimeFollower_1_StbM_TimeSync_GetCurrentTime_Rx_CALL</FIBEX-ELEMENT-REF>
</FIBEX-ELEMENT-REF-CONDITIONAL>
<FIBEX-ELEMENT-REF-CONDITIONAL>
  <FIBEX-ELEMENT-REF DEST="I-SIGNAL-I-PDU">/TimeFollower/Systems/Signals/pduTimeFollower_1_StbM_TimeSync_GetCurrentTime_Rx_RETURN</FIBEX-ELEMENT-REF>
</FIBEX-ELEMENT-REF-CONDITIONAL>
<FIBEX-ELEMENT-REF-CONDITIONAL>
  <FIBEX-ELEMENT-REF DEST="I-SIGNAL">/TimeFollower/Systems/Signals/TimeFollower_1_StbM_TimeSync_GetCurrentTime_Rx_CALL</FIBEX-ELEMENT-REF>
</FIBEX-ELEMENT-REF-CONDITIONAL>
<FIBEX-ELEMENT-REF-CONDITIONAL>
  <FIBEX-ELEMENT-REF DEST="I-SIGNAL">/TimeFollower/Systems/Signals/TimeFollower_1_StbM_TimeSync_GetCurrentTime_Rx_RETURN</FIBEX-ELEMENT-REF>
</FIBEX-ELEMENT-REF-CONDITIONAL>

```

Add PDUs triggering for CALL and RETURN.

```

<PDU-TRIGGERING UUID="D83352DF-B124-4506-AFAE-13BDB0A5B954">
  <SHORT-NAME>ETH_VNDS_PHY_ETH_VNDS_PHY_ETH_VNDS_ETH_VNDS_D</SHORT-NAME>
  <I-PDU-PORT-REFS>
    <I-PDU-PORT-REF DEST="I-PDU-PORT">/Topology/HardwareComponents/GWM/con_Eth_VNDS_GWM/ipduport_GWM_ETH_IN</I-PDU-PORT-REF>
  </I-PDU-PORT-REFS>
  <I-PDU-REF DEST="I-SIGNAL-I-PDU">/TimeFollower/Systems/Signals/pduTimeFollower_1_StbM_TimeSync_GetCurrentTime_Rx_CALL</I-PDU-REF>
</PDU-TRIGGERING>
<PDU-TRIGGERING UUID="2C41CF11-C815-46BA-AB5D-6491C57EF6DC">
  <SHORT-NAME>ETH_VNDS_PHY_ETH_VNDS_PHY_ETH_VNDS_ETH</SHORT-NAME>
  <I-PDU-PORT-REFS>
    <I-PDU-PORT-REF DEST="I-PDU-PORT">/Topology/HardwareComponents/GWM/con_Eth_VNDS_GWM/ipduport_GWM_ETH_OUT</I-PDU-PORT-REF>
  </I-PDU-PORT-REFS>
  <I-PDU-REF DEST="I-SIGNAL-I-PDU">/TimeFollower/Systems/Signals/pduTimeFollower_1_StbM_TimeSync_GetCurrentTime_Rx_RETURN</I-PDU-REF>
</PDU-TRIGGERING>

```

Add signals trigger

```

<I-SIGNAL-TRIGGERING UUID="FBBC95D2-CC33-47B7-8F1A-354D4A50C08F">
  <SHORT-NAME>signalTrig_StbM_TimeSync_GetCurrentTime_Rx_RETURN</SHORT-NAME>
  <I-SIGNAL-PORT-REFS>
    <I-SIGNAL-PORT-REF DEST="I-SIGNAL-PORT">/Topology/HardwareComponents/GWM/con_Eth_VNDS_GWM/isignalport_GWM_ETH_OUT</I-SIGNAL-PORT-REF>
  </I-SIGNAL-PORT-REFS>
  <I-SIGNAL-REF DEST="I-SIGNAL">/TimeFollower/Systems/Signals/TimeFollower_1_StbM_TimeSync_GetCurrentTime_Rx_RETURN</I-SIGNAL-REF>
</I-SIGNAL-TRIGGERING>
<I-SIGNAL-TRIGGERING UUID="9EAE1F96-18B9-4575-8AE1-758FB7BF9E25">
  <SHORT-NAME>signalTrig_StbM_TimeSync_GetCurrentTime_Rx_CALL</SHORT-NAME>
  <I-SIGNAL-PORT-REFS>
    <I-SIGNAL-PORT-REF DEST="I-SIGNAL-PORT">/Topology/HardwareComponents/GWM/con_Eth_VNDS_GWM/isignalport_GWM_ETH_IN</I-SIGNAL-PORT-REF>
  </I-SIGNAL-PORT-REFS>
  <I-SIGNAL-REF DEST="I-SIGNAL">/TimeFollower/Systems/Signals/TimeFollower_1_StbM_TimeSync_GetCurrentTime_Rx_CALL</I-SIGNAL-REF>
</I-SIGNAL-TRIGGERING>

```

13.5 Steps on Davinci Configurator

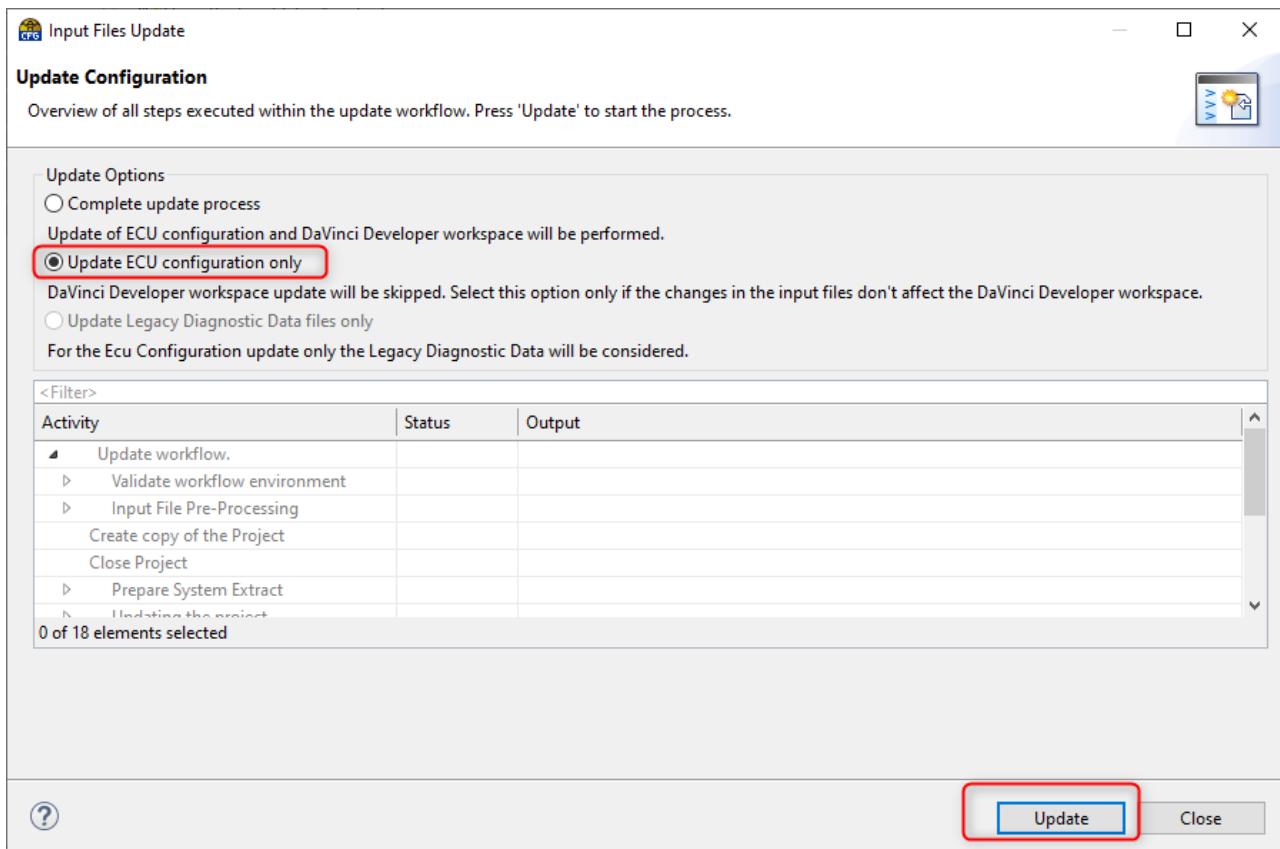
13.5.1 Step 6: Update ARXML Files in Davinci Cfg

⚠ Sync Needed

Please consider that after modifying (and saving) the ARXML file, an Update shall be need, please follow the recomended Updating process by clicking the Yellow ribbon in the Vector Davinci Configurator Tool.

Please make use to follow the Update show in the Following Images:

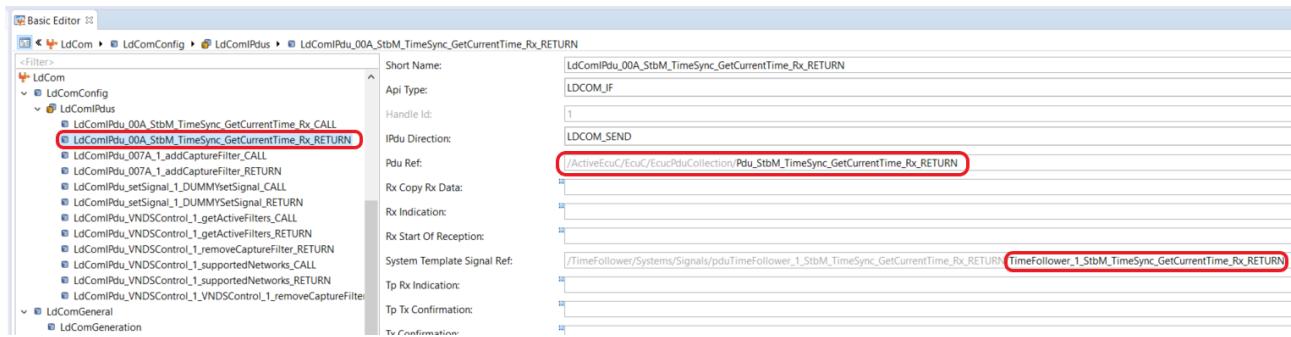




13.5.2 Step 7: LdCom Configuration

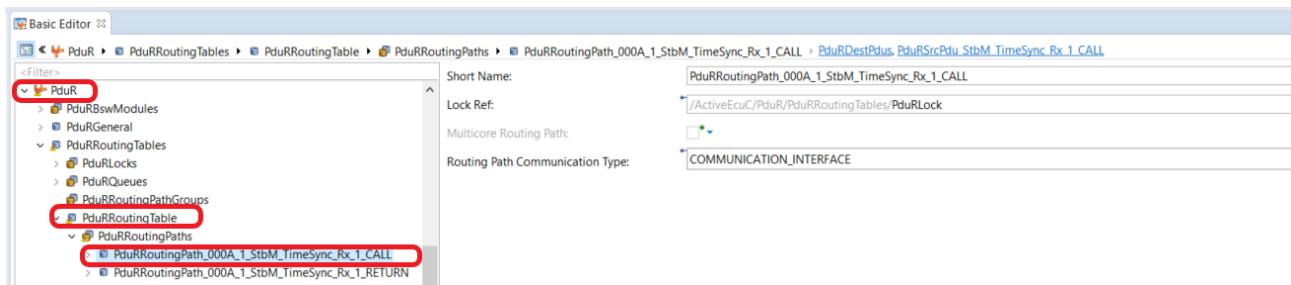
Configure LdCom IPdu CALL. Assign the corresponding Pdu and system signal.

Configure LdCom IPdu RETURN. Assign the corresponding Pdu and system signal.

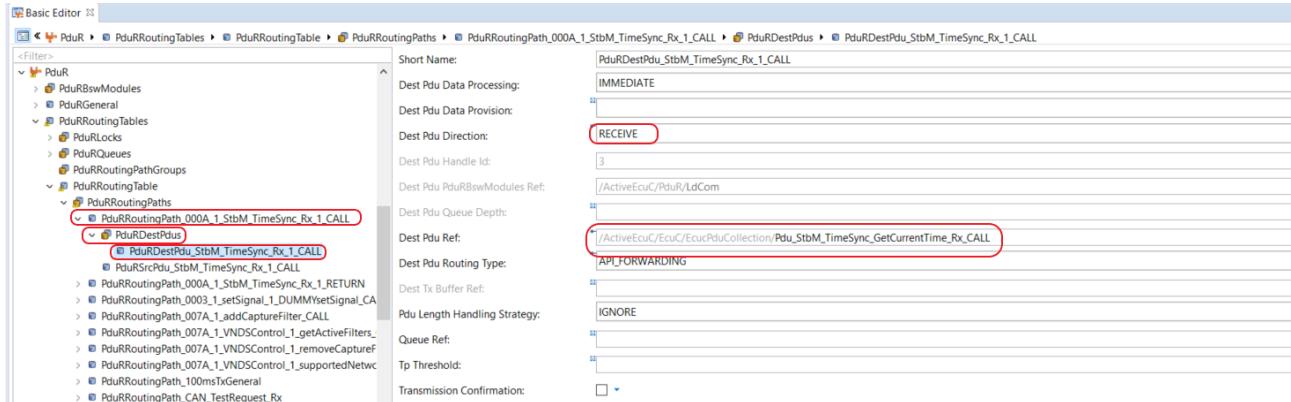


13.5.3 Step 8: PduR Configuration

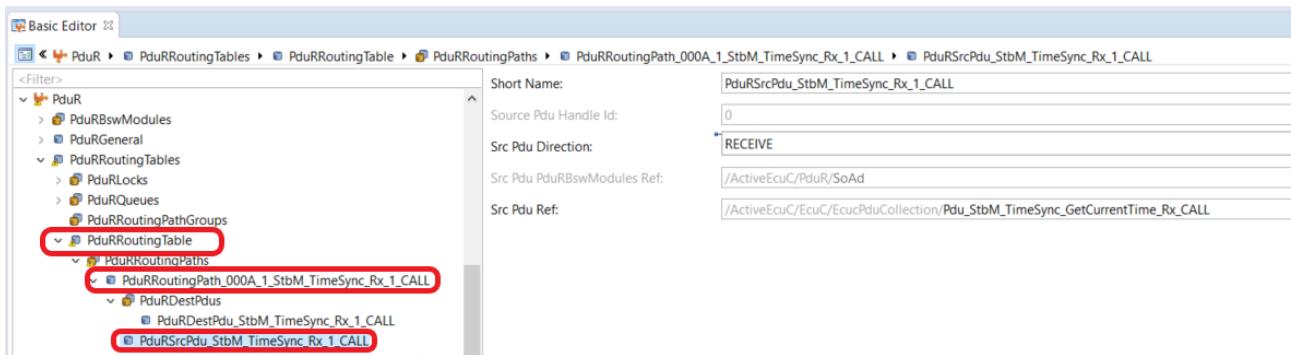
Create PduR for CALL



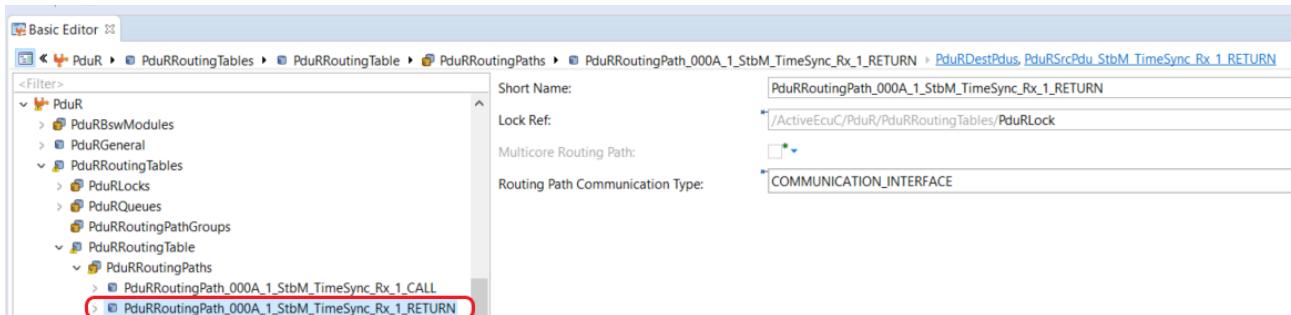
Configure PduR CALL destination.



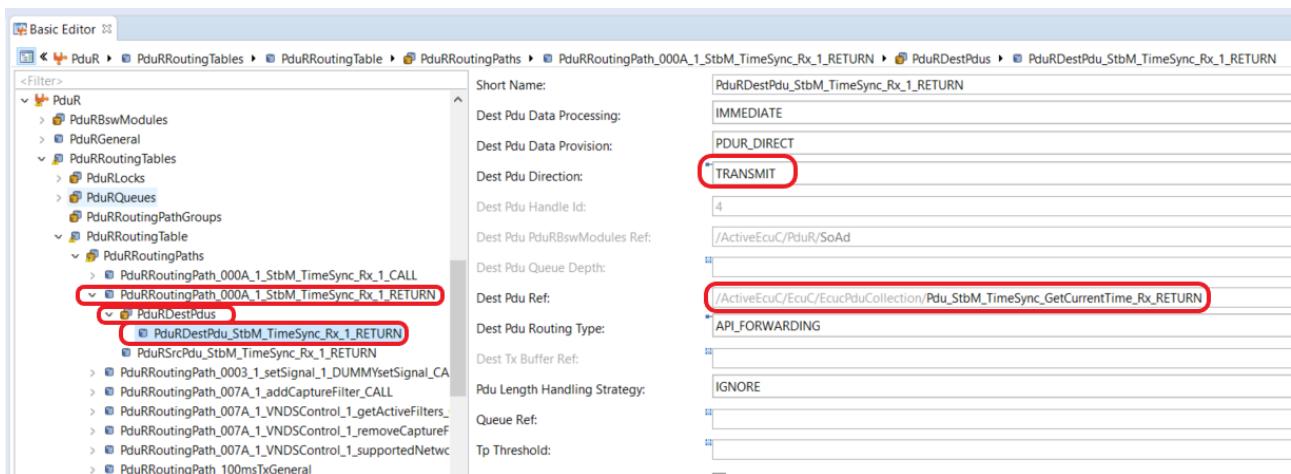
Configure PduR CALL Source.



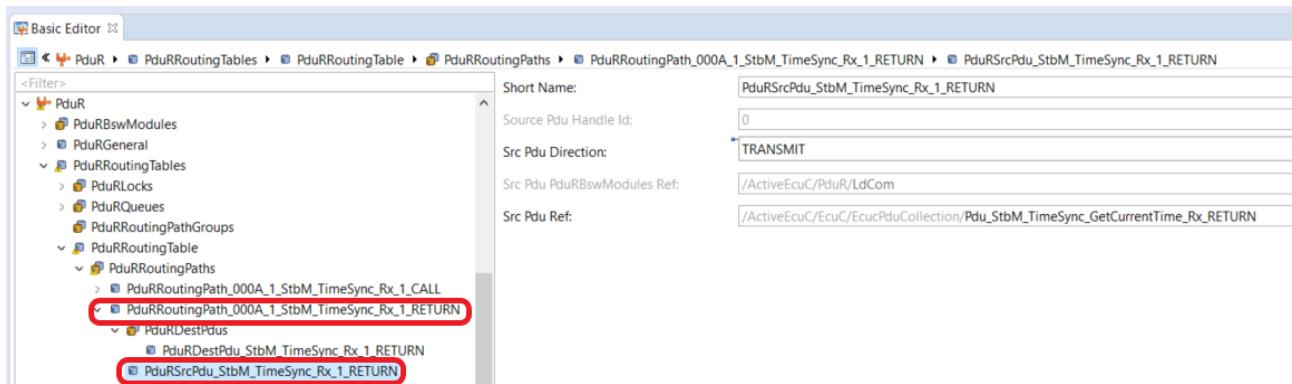
Create PduR for RETURN



Configure PduR RETURN destination.

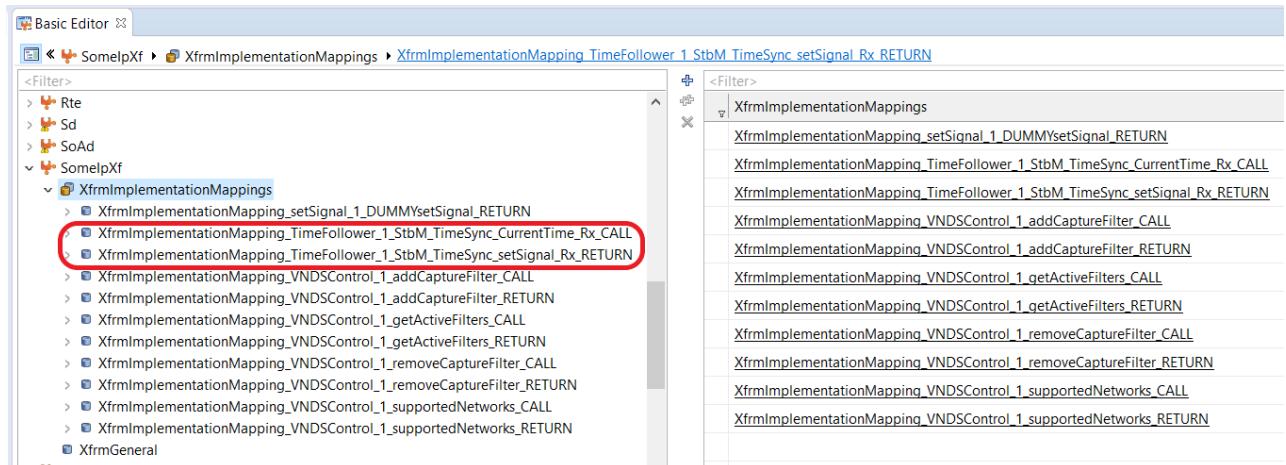


Configure PduR RETURN Source.

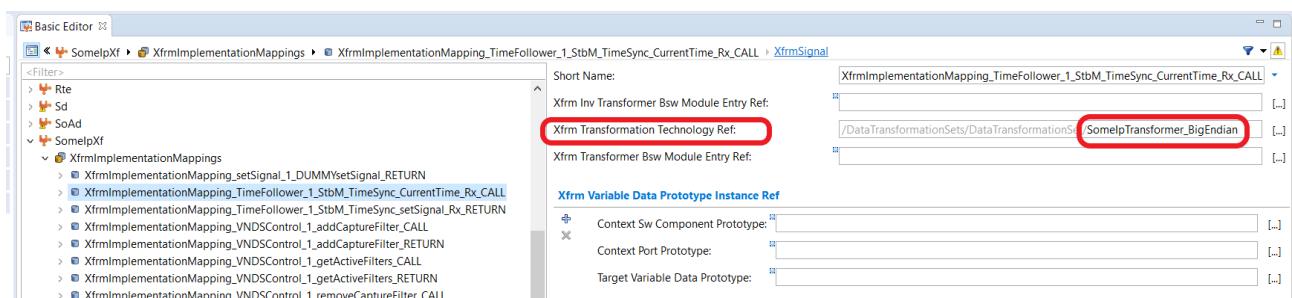


13.5.4 Step 9: SomelpXf Configuration(SOME/IP Transformer):

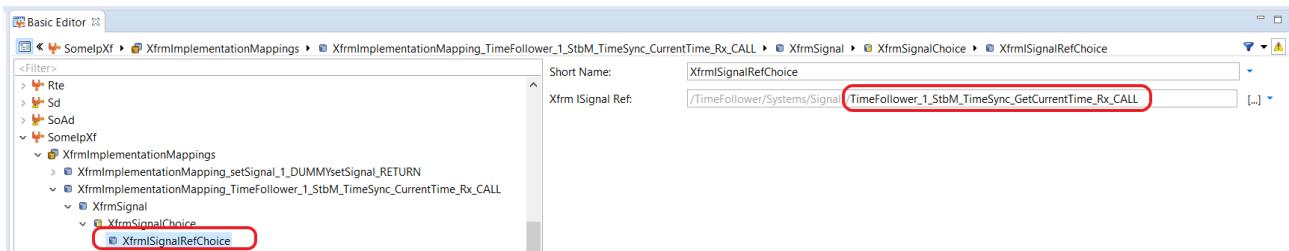
Create an Xfrm Implementation mapping for CALL and RETURN.



Update Transformation Technology Ref to BigEndian.



Map the corresponding Signal into the Signal reference choice.



This steps need to be done for CALL and RETURN signals.

13.6 Steps on Davinci Developer

13.6.1 Step 10: Map the new Signal

After Step 9 the Mapping shall be done automatically, please verify that the mapping looks like the following image:

	Network Signal	Transform	Message	Network	Direction	Port	DEP/Operation/Trigger	Prede
10101	~sysSignal_VNDSControl_1_add...	SomeIP	ETH_VNDS_PHY_ETH_VNDS...	ETH_VNDS	Rx-Signal	P_122_1_VNDSControl	addCaptureFilter	-
	~sysSignal_VNDSControl_1_add...	SomeIP	ETH_VNDS_PHY_ETH_VNDS...	ETH_VNDS	Tx-Signal	P_122_1_VNDSControl	addCaptureFilter	-
	~sysSignal_VNDSControl_1_get...	SomeIP	ETH_VNDS_PHY_ETH_VNDS...	ETH_VNDS	Rx-Signal	P_122_1_VNDSControl	getActiveFilters	-
	~sysSignal_VNDSControl_1_get...	SomeIP	ETH_VNDS_PHY_ETH_VNDS...	ETH_VNDS	Tx-Signal	P_122_1_VNDSControl	getActiveFilters	-
	~sysSignal_VNDSControl_1_re...	SomeIP	ETH_VNDS_PHY_ETH_VNDS...	ETH_VNDS	Rx-Signal	P_122_1_VNDSControl	removeCaptureFilter	-
	~sysSignal_VNDSControl_1_re...	SomeIP	ETH_VNDS_PHY_ETH_VNDS...	ETH_VNDS	Tx-Signal	P_122_1_VNDSControl	removeCaptureFilter	-
	~sysSignal_VNDSControl_1_sup...	SomeIP	ETH_VNDS_PHY_ETH_VNDS...	ETH_VNDS	Rx-Signal	P_122_1_VNDSControl	supportedNetworks	-
	~sysSignal_VNDSControl_1_sup...	SomeIP	ETH_VNDS_PHY_ETH_VNDS...	ETH_VNDS	Tx-Signal	P_122_1_VNDSControl	supportedNetworks	-
	~sysSignal_Data_Request_TFM...	No	FD1_CAN_FD1_CAN_FD1_C...	FD1_CAN	Rx-Signal	Test_Option_Rq	Data_Resp	-
	~sysSignal_Data_Reqo_TFM_Tx	No	FD1_CAN FD1_CAN FD1_C...	FD1_CAN	Tx-Signal	Data_Resp	SOMEIP_GetCurrentTime	-
	~sysSignal_TimeFollower_1_St...	SomeIP	ETH_VNDS_PHY_ETH_VNDS...	ETH_VNDS	Rx-Signal	S_FNV4_SOMEIP_Control	SOMEIP_GetCurrentTime	-
	~sysSignal_TimeFollower_1_St...	SomeIP	ETH_VNDS_PHY_ETH_VNDS...	ETH_VNDS	Tx-Signal	S_FNV4_SOMEIP_Control	SOMEIP_GetCurrentTime	-

13.7 Troubleshooting

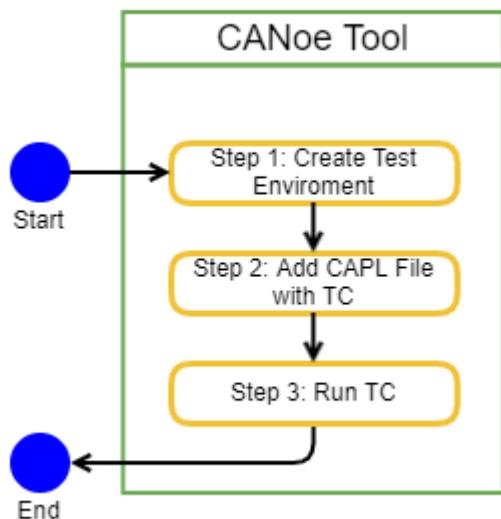
14 12_Adde new Test Enviroment to CANoe

① Vector CANoe

This Recipe is related to create Test Enviroment for Automated Test in CANoe Tools, which is from th same provider that DaVinci Tools, Vector

- [Workflow Diagram](#)(see page 187)
- [Prerequisites](#)(see page 187)
- [Steps](#)(see page 188)
- [Steps to be performed in Vector CANoe tool](#)(see page 188)
 - Step 1: Create new Test Enviroment(see page 188)
 - Step 2: Insert New Testcase File(see page 189)
 - Step 3: Run the TestCase(see page 191)
- [Troubleshooting](#)(see page 195)

14.1 Workflow Diagram



14.2 Prerequisites

1. Have a project up and running (having the input file and every previous detail).
2. Having a CANoe Setup configure for your project.

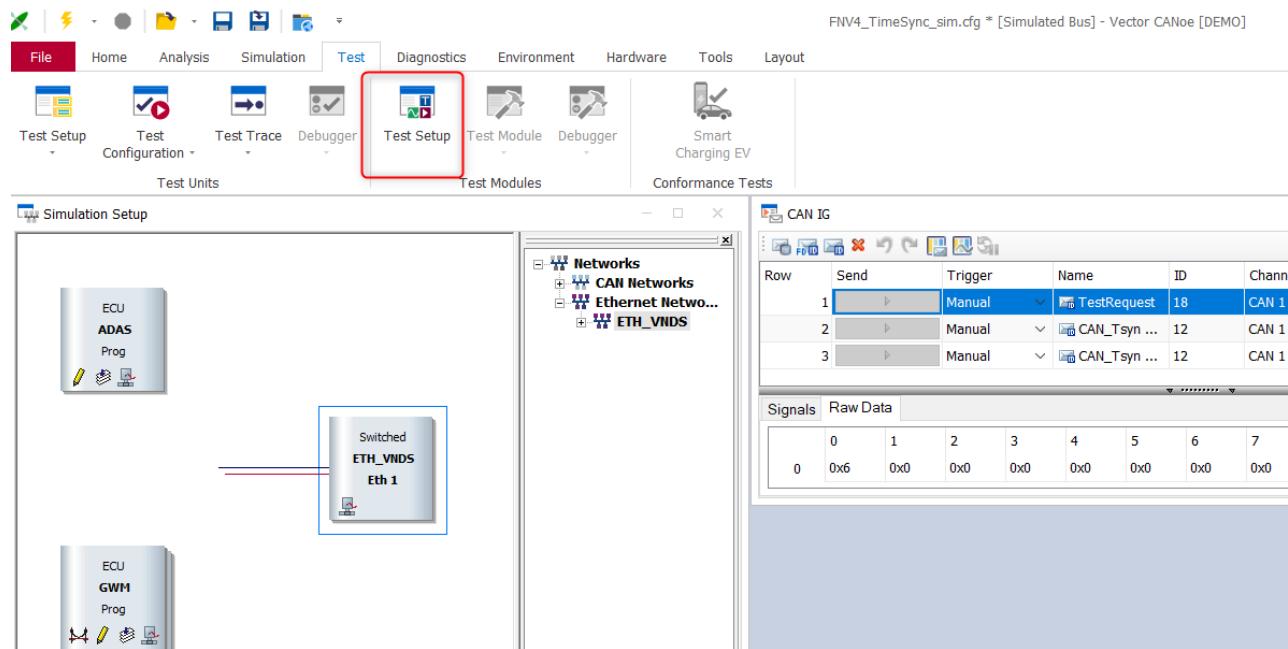
14.3 Steps

14.4 Steps to be performed in Vector CANoe tool

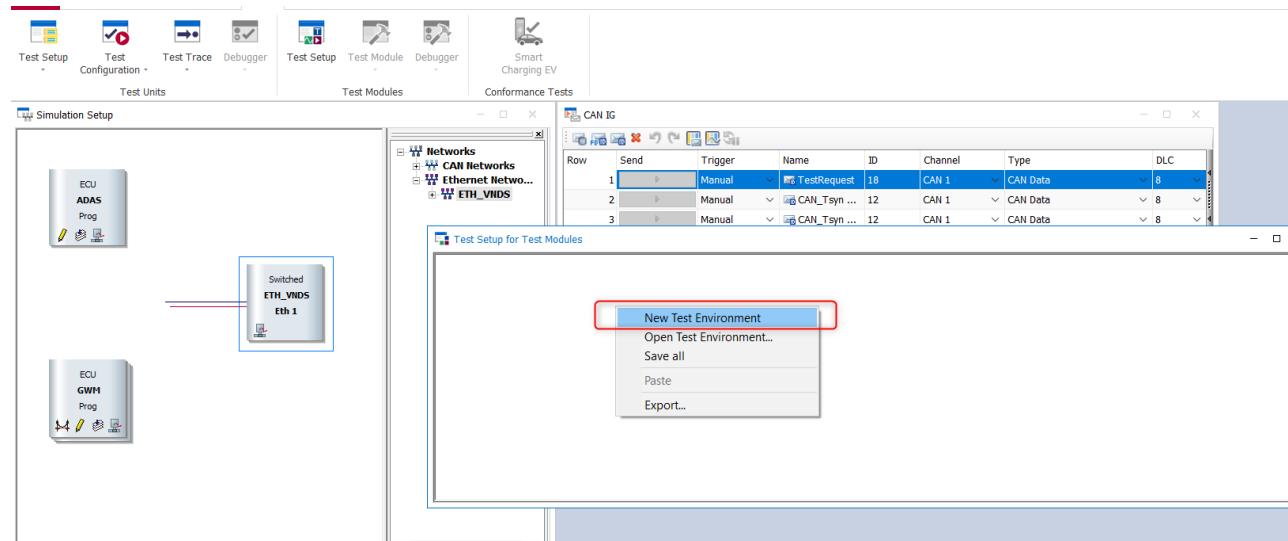
14.4.1 Step 1: Create new Test Environment

To create new Test Environment please follow the next Path:

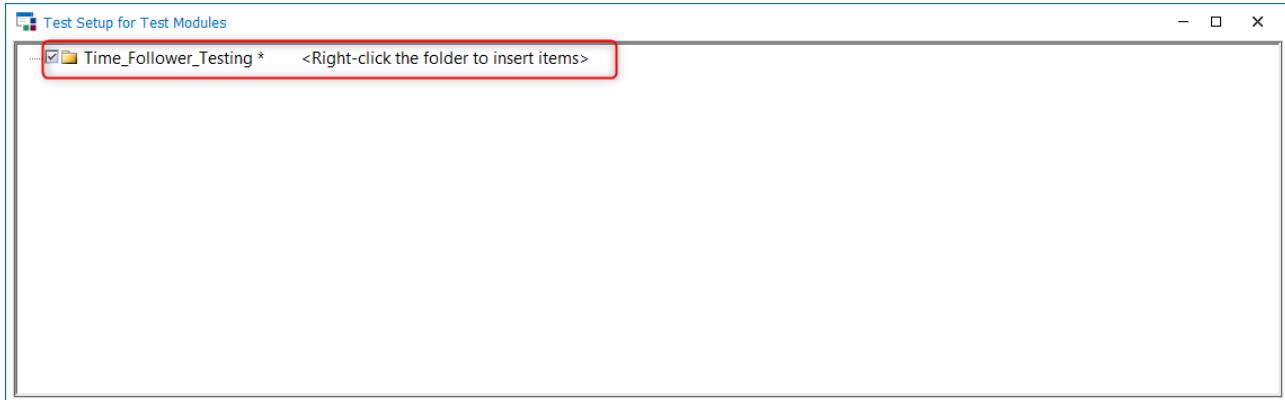
Open <project CANoe Simulation> → Test Tab → Click "Test Setup".



A New window called "Test Setup for Test Module" shall appear, then right click and Click "New Test Environment".



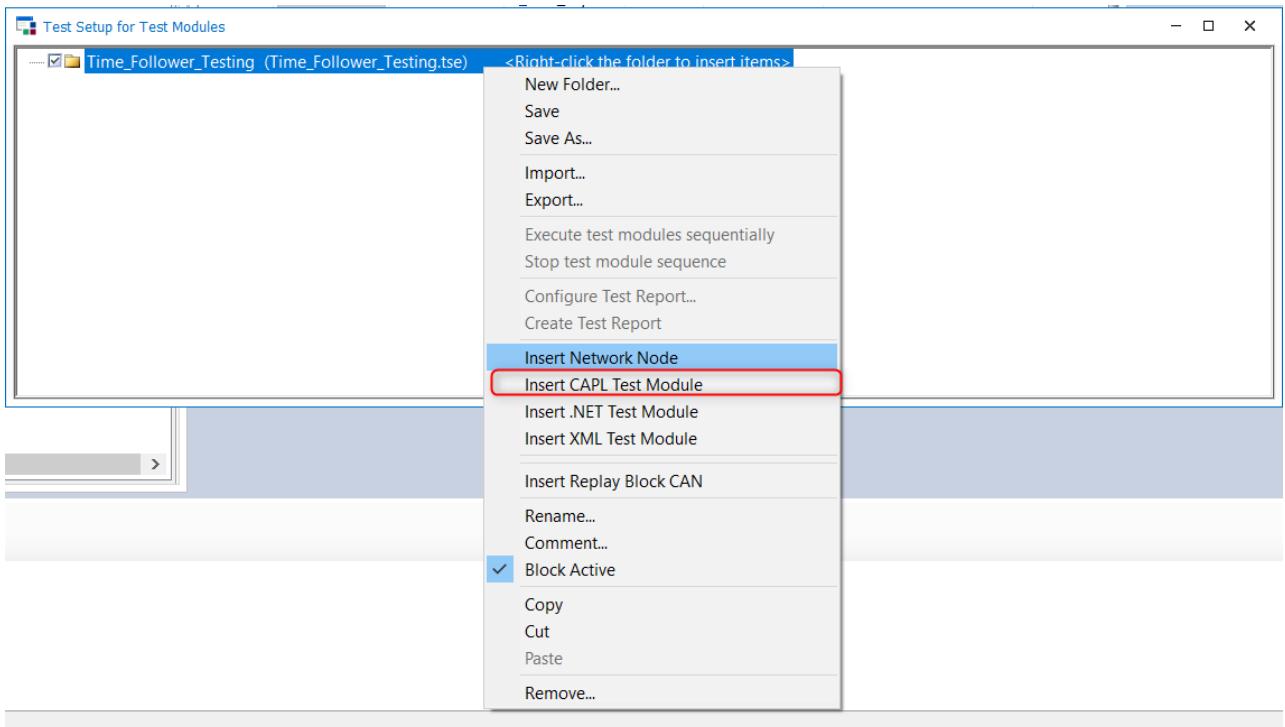
Rename your new Environment and save it in expected path.



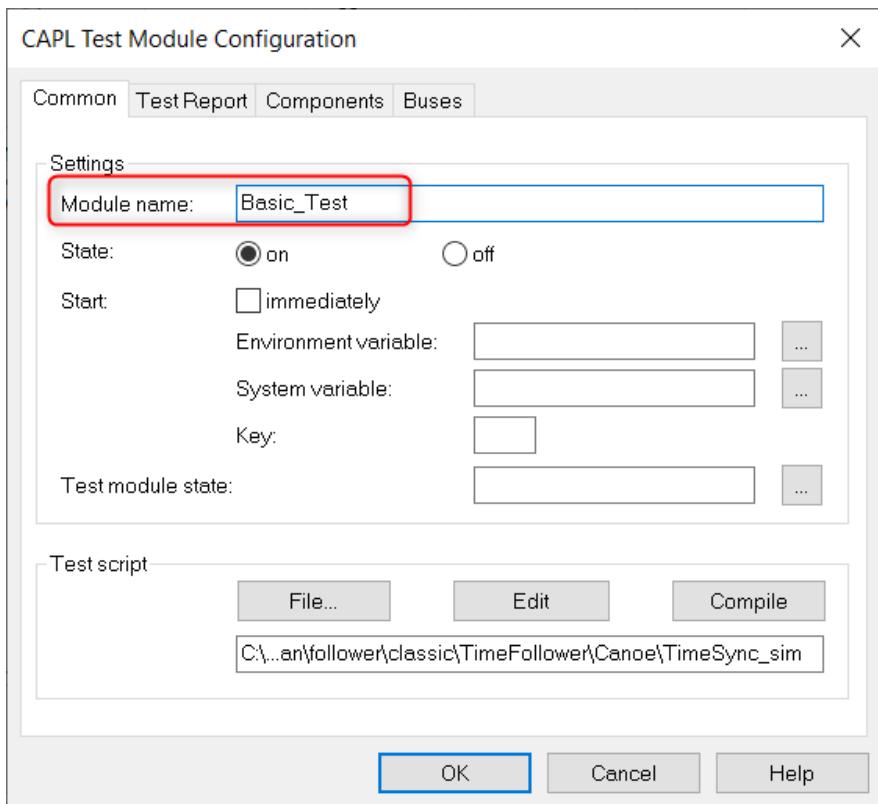
14.4.2 Step 2: Insert New Testcase File

To add your Testcase file (.can or .cin extension):

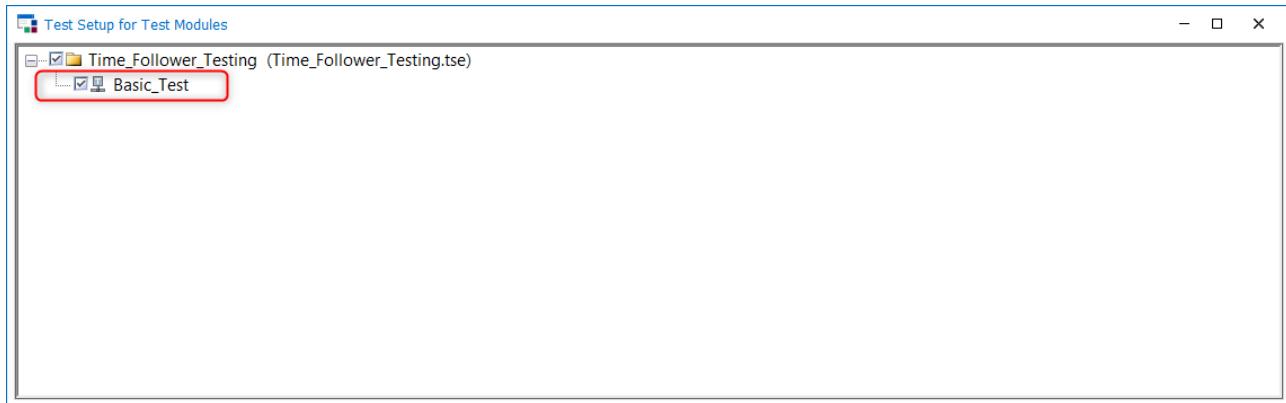
Right click in the Test Environment and select Insert CAPL Test Module



This will add the Test case file, now right click → Configuration, and Rename it with a meaningful name.



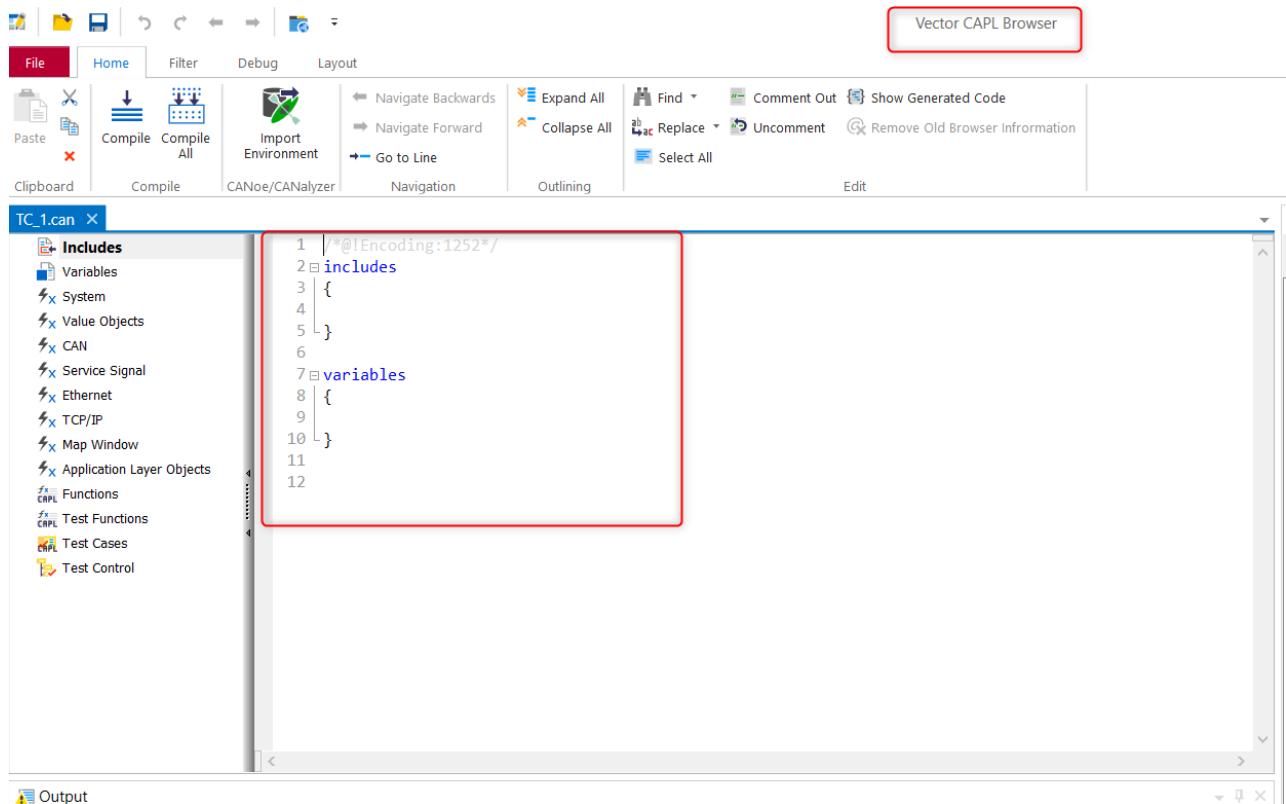
The Test Environment shall look like:



Now in order to fill the Testcase file

Right Click → Edit, this will open a Pop-up window to chose the proper path for your Test Case file, after saving the file the CAPL Browser Tool will open.

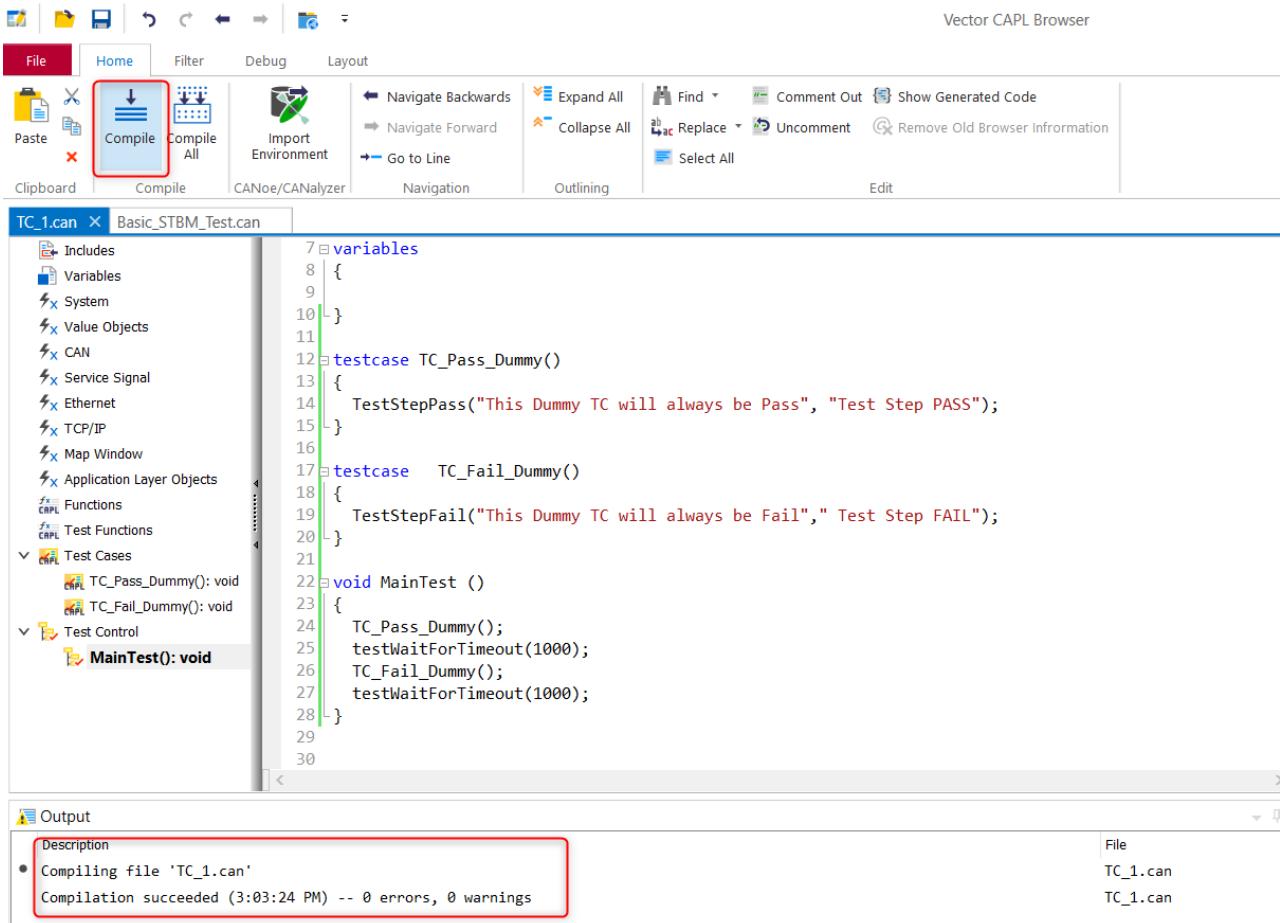
The created file is empty with the minimum structure to be compile



Now fill the .can File with the necessary test steps and validation.

14.4.3 Step 3: Run the TestCase

After having the Test Case file done, please Compile it before closing the CAPL Browser Tool.



The screenshot shows the Vector CANPL Browser interface. The 'File' tab is selected. In the toolbar, the 'Compile' button (represented by a downward arrow icon) is highlighted with a red box. The main window displays a code editor for 'TC_1.can' containing C code for test cases. Below the code editor is an 'Output' window showing compilation results.

```

7  variables
8  {
9  }
10 }
11
12 testcase TC_Pass_Dummy()
13 {
14     TestStepPass("This Dummy TC will always be Pass", "Test Step PASS");
15 }
16
17 testcase TC_Fail_Dummy()
18 {
19     TestStepFail("This Dummy TC will always be Fail", "Test Step FAIL");
20 }
21
22 void MainTest ()
23 {
24     TC_Pass_Dummy();
25     testWaitForTimeout(1000);
26     TC_Fail_Dummy();
27     testWaitForTimeout(1000);
28 }
29
30

```

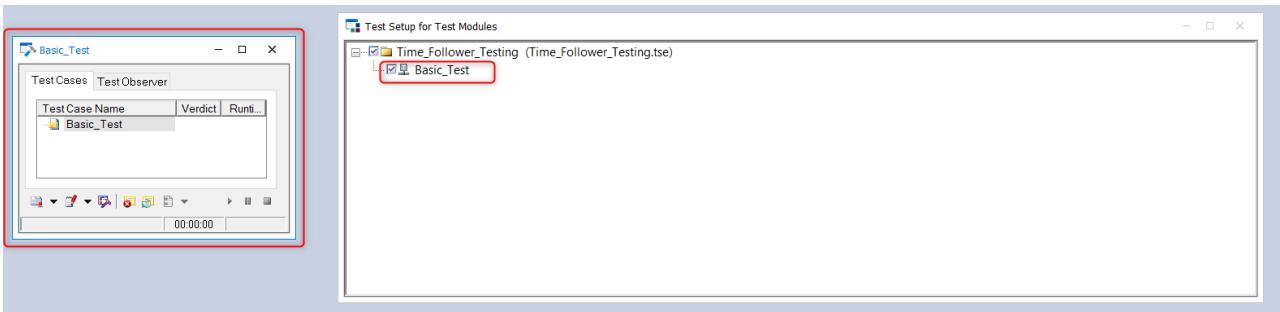
Output

- Compiling file 'TC_1.can'

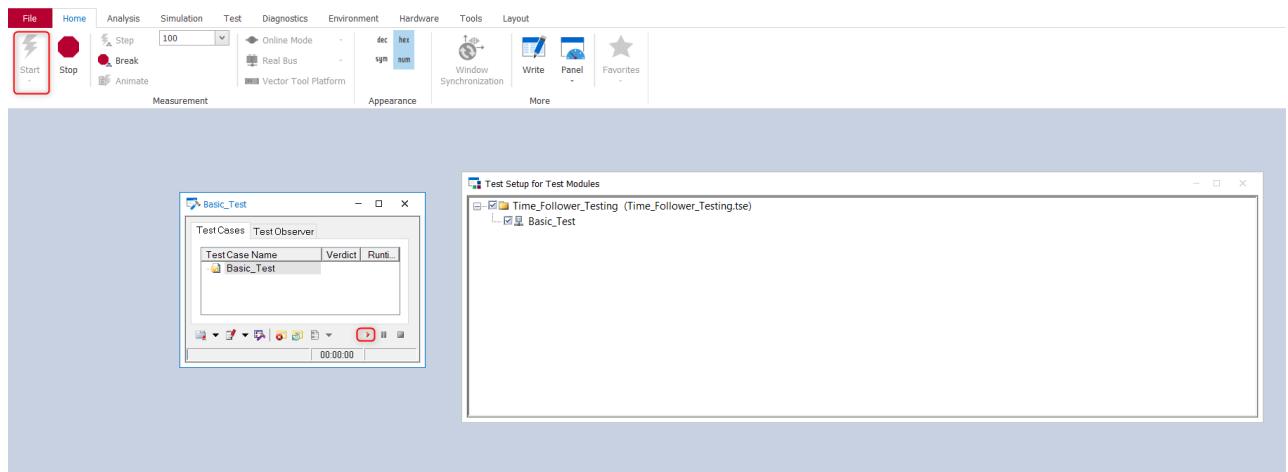
Compilation succeeded (3:03:24 PM) -- 0 errors, 0 warnings

Now return to the Test Environment Window in the CANoe Simulation, and Doble click on the Test Case File.

This will bring up the Execution Window.



Now start the project simulation and hit the small play button on the TestCase file Execution Window.



That will execute the TestCases on your TestCases File. Some extra information can be seen in the Test Observer tab.

Basic_Test

Test Cases Test Observer

Test Case Name	Verdict	Runtime
Basic_Test		
TC_Pass_Dummy	✓	0.009 s
TC_Fail_Dummy	✗	0.003 s

executed: 2 00:00:03 Failed: 1

The screenshot shows a software interface titled 'Basic_Test'. At the top, there are two tabs: 'Test Cases' (which is selected) and 'Test Observer'. Below the tabs is a table with three columns: 'Test Case Name', 'Verdict', and 'Runtime'. The table has four rows. The first row is collapsed under the heading 'Basic_Test'. The second row contains 'TC_Pass_Dummy' with a green checkmark and '0.009 s' runtime. The third row contains 'TC_Fail_Dummy' with a red X and '0.003 s' runtime. At the bottom of the window, there is a status bar with the text 'executed: 2', '00:00:03', and 'Failed: 1'. The overall interface is clean and professional, typical of automotive development tools.

Basic_Test

Test Cases Test Observer

Action	Status
① TC_Pass_Dummy ⚠ This is only a DEMO variant of CANoe!	start test case
✓ Test Step PASS	warning
✓ TC_Pass_Dummy	pass
② TC_Fail_Dummy ⚠ This is only a DEMO variant of CANoe!	start test case
✗ Test Step FAIL	warning
✗ TC_Fail_Dummy	fail

Test Case:
Test Function:

executed: 2 00:00:03 Failed: 1

The screenshot shows a software interface titled 'Basic_Test'. At the top, there are two tabs: 'Test Cases' (which is selected) and 'Test Observer'. The main area displays a table of test actions and their corresponding statuses. The table has two columns: 'Action' and 'Status'. The 'Action' column lists several entries, each with a small icon indicating its status: a blue info icon for 'TC_Pass_Dummy', a green checkmark for 'Test Step PASS', another green checkmark for 'TC_Pass_Dummy', a blue info icon for 'TC_Fail_Dummy', a red asterisk for 'Test Step FAIL', and a red asterisk for 'TC_Fail_Dummy'. The 'Status' column provides more detail: 'start test case' (blue), 'warning' (yellow), 'pass' (green), 'start test case' (blue), 'warning' (yellow), and 'fail' (red). Below the table, there are fields for 'Test Case:' and 'Test Function:', both currently empty. At the bottom, there is a toolbar with various icons and a status bar showing 'executed: 2', '00:00:03', and 'Failed: 1'.

14.5 Troubleshooting