Contents

[1. Introduction 2](#_Toc128993992)

[2. Walkthrough 2](#_Toc128993993)

[2.1 Code Resync 2](#_Toc128993994)

[2.2 Code Compile 2](#_Toc128993995)

[2.3 Deliverable creation 2](#_Toc128993996)

[2.3.1 Application 2](#_Toc128993997)

[2.3.2 FBL 8](#_Toc128993998)

[2.3.3 HSM 9](#_Toc128993999)

Figures

[Figure 1 - Change the Diagnostic Version if applicable 3](#_Toc129000060)

[Figure 2 - Select the new release binaries 4](#_Toc129000061)

[Figure 3 - Select the Encoded and Raw files 4](#_Toc129000062)

[Figure 4 - Select all the security relevant segments, all segments in our case 5](#_Toc129000063)

[Figure 5 – Start the Daimler Security Plugin 5](#_Toc129000064)

[Figure 6 - Select the key to be used for the Signature 5](#_Toc129000065)

[Figure 7 - After you hit CTRL+S, this window appears, it is expected, hit OK 6](#_Toc129000066)

[Figure 8 - Go to the Actions menu 6](#_Toc129000067)

[Figure 9 - Select the Application SMR-F Action and hit test 7](#_Toc129000068)

[Figure 10 - After hitting Test, this screen will appear that calls Trafo, select the Flash Ident and hit OK, this is a good opportunity to check the Part Number and Version directly in the name of the file 8](#_Toc129000069)

[Figure 11 - Configuration of the ODX for the FBL 8](#_Toc129000070)

[Figure 12 - The HSM Version to be changed 9](#_Toc129000071)

Tables

**No table of figures entries found.**

# Introduction

For each release there are a couple of steps that need to be followed in order to create the release bineries.

The current list of release bineries is:

* Application Container – Contains the application to be flashed via CAN (.smr-f file)
* FBL Updater Container – Contains the FBL Updater + New FBL (.smr-f file)
* HSM Updater Container – Contains the new HSM container (.smr-f file)

In order to create these files, we will walk through the whole process from the resync of the code, to compiling to releasing.

# Walkthrough

## Code Resync

The first step is to make sure you have the latest code from PTC.

Important paths to check:

* Makefiles -> *e:/MKSProjects/SBE/PP/DAIMLER\_MMA/Phase\_02/View\_Development/Tools/Build\_Env/Workspace/Build/Makefile/project.pj*
* Components -> *e:/MKSProjects/SBE/PP/DAIMLER\_MMA/Phase\_02/View\_Development/Components/project.pj*
* Build/Outputs/NVP -> *e:/MKSProjects/SBE/PP/DAIMLER\_MMA/Phase\_02/View\_Development/Tools/Build\_Env/Workspace/Outputs/nvp/project.pj*
* ODX -> *e:/MKSProjects/SBE/PP/DAIMLER\_MMA/Phase\_02/View\_Development/Tools/Odx/Workspace/Daimler\_1608/project.pj*

## Code Compile

After we are sure we have the right code, we need to compile all the components necessary for the release.

It’s important to note that there are a couple of postbuild scripts that will be called and are important for the release, for example:

* "S:\Tools\Vector\_configuration\Workspace\hsmupdater\Appl\PrepareFirmware.bat"

It’s important to build the FBL before the FBL Updater, since the newly built FBL will be used as an input for the output of the FBL Updater.

## Deliverable creation

After all the components have been compiled, it’s time to start the creation of the deliverable files.

### Application

The first step to prepare the release is to call the PrepareReleaseODX.py script -> *e:/MKSProjects/SBE/PP/DAIMLER\_MMA/Phase\_02/View\_Development/Tools/Odx/Workspace/Daimler\_1608/PrepareReleaseOdx.py*

This script will do the following things:

* Will copy the App and FBL binaries from the Outputs folder to the release folder
* Will change the hex address space to 0xAXXXX
* Will align the hex file to 32 bytes
* Will remove the Presence Pattern (that was introduced to be able to use the outputs without any additional changes with the debugger)
* Will create the Encoded and Compressed version of the binaries

The next step is to configure and call the replaceOdxVersions.py script -> *e:/MKSProjects/SBE/PP/DAIMLER\_MMA/Phase\_02/View\_Development/Tools/Odx/Workspace/Daimler\_1608/Supplier/replaceOdxVersions.py*

The important parts to configure in this script:

* Release numbers and versions for App:
  + Example when the part number is not changed:
    - oldAppPartNumber = "1749025500";
    - newAppPartNumber = "1749025500";
  + Example whem the version number is changed:
    - oldAppReleaseVer = "225000";
    - newAppReleaseVer = "230500";

Note: Please take the old values from the currently existing ODX files (application.odx-f) since the script does a search and replace, providing an incorrect old value means that the new versions will not be applied.

The next step is to run the release steps from ODX Studio:

Start by oppening the ODX Studio project -> *e:/MKSProjects/SBE/PP/DAIMLER\_MMA/Phase\_02/View\_Development/Tools/Odx/Workspace/Daimler\_1608/Supplier/Daimler\_Flashing\_support\_16\_08.odxstudio*

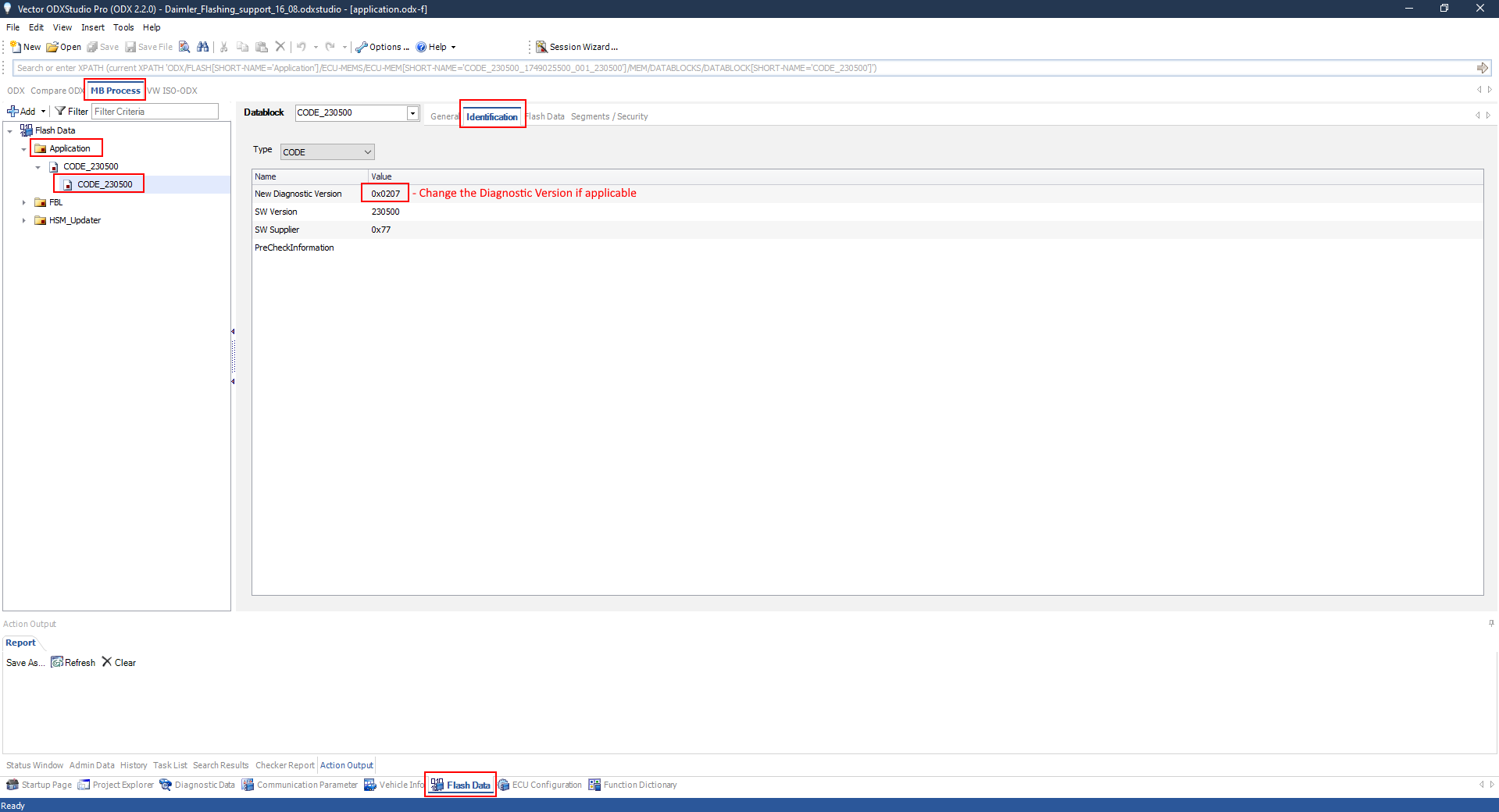


Figure 1 - Change the Diagnostic Version if applicable

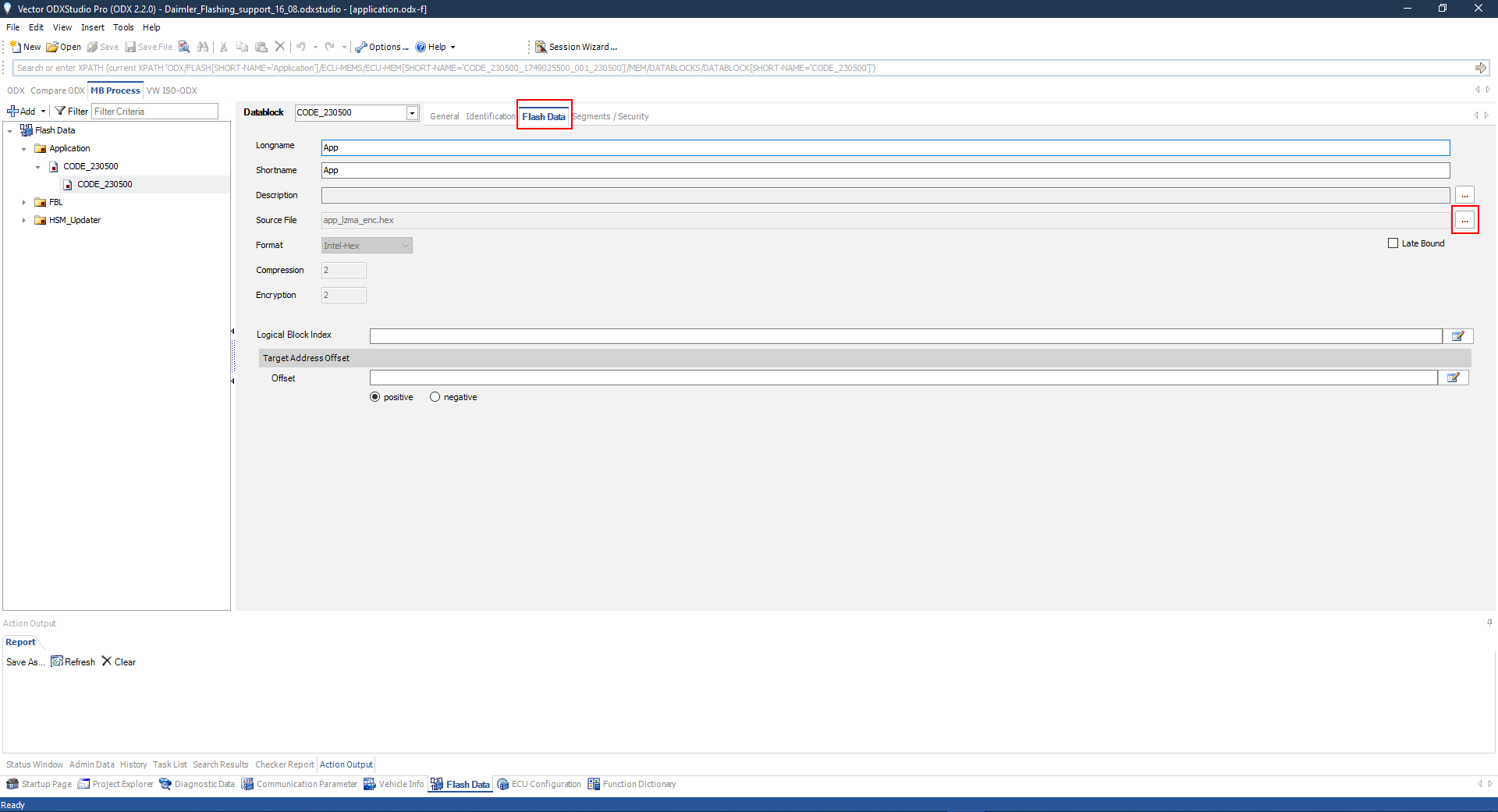


Figure 2 - Select the new release binaries

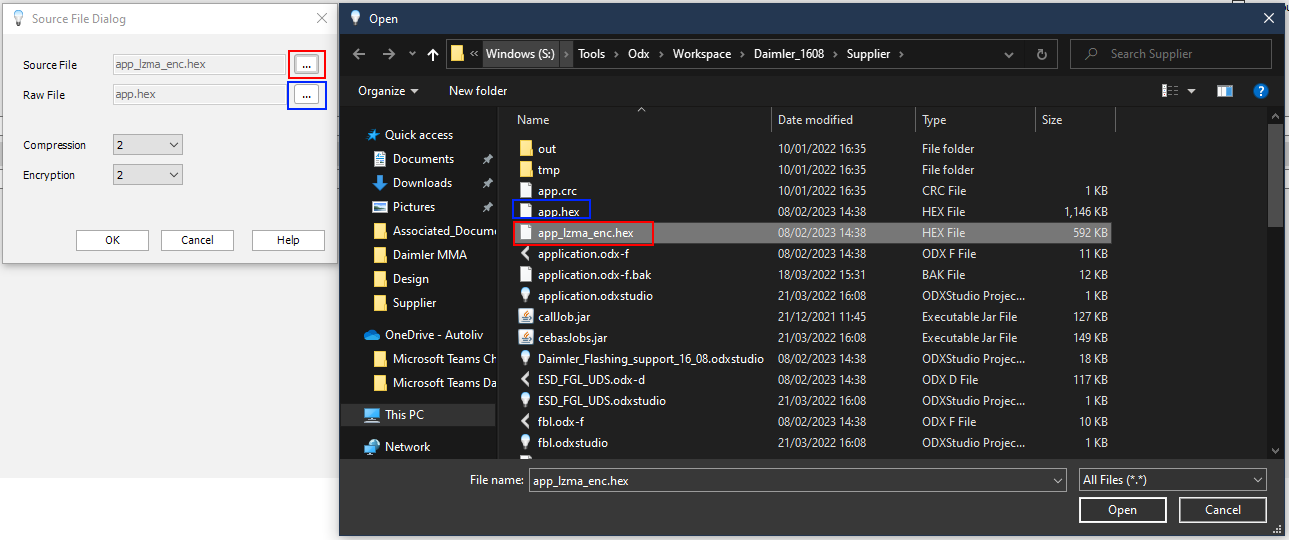


Figure 3 - Select the Encoded and Raw files

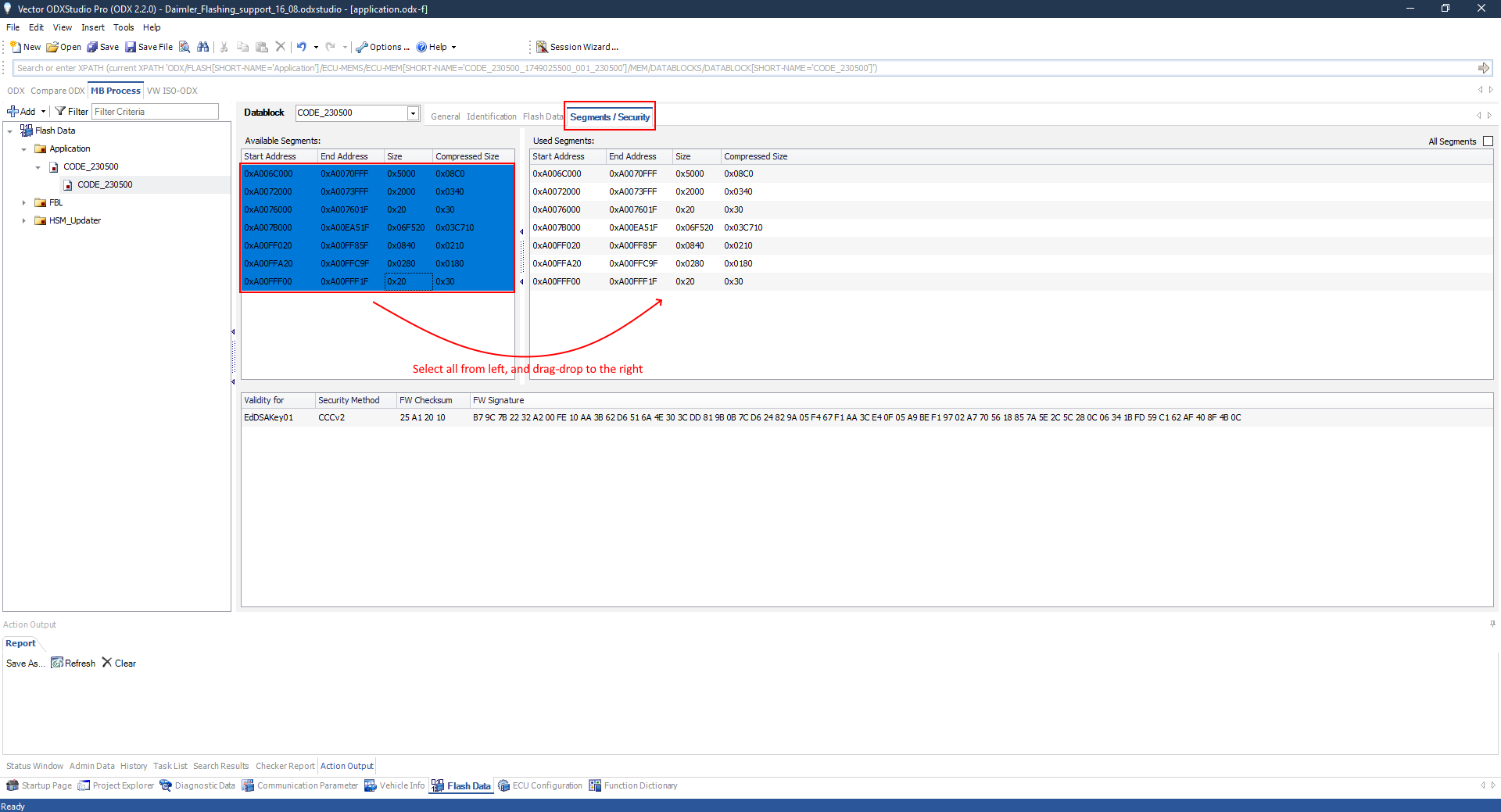


Figure 4 - Select all the security relevant segments, all segments in our case

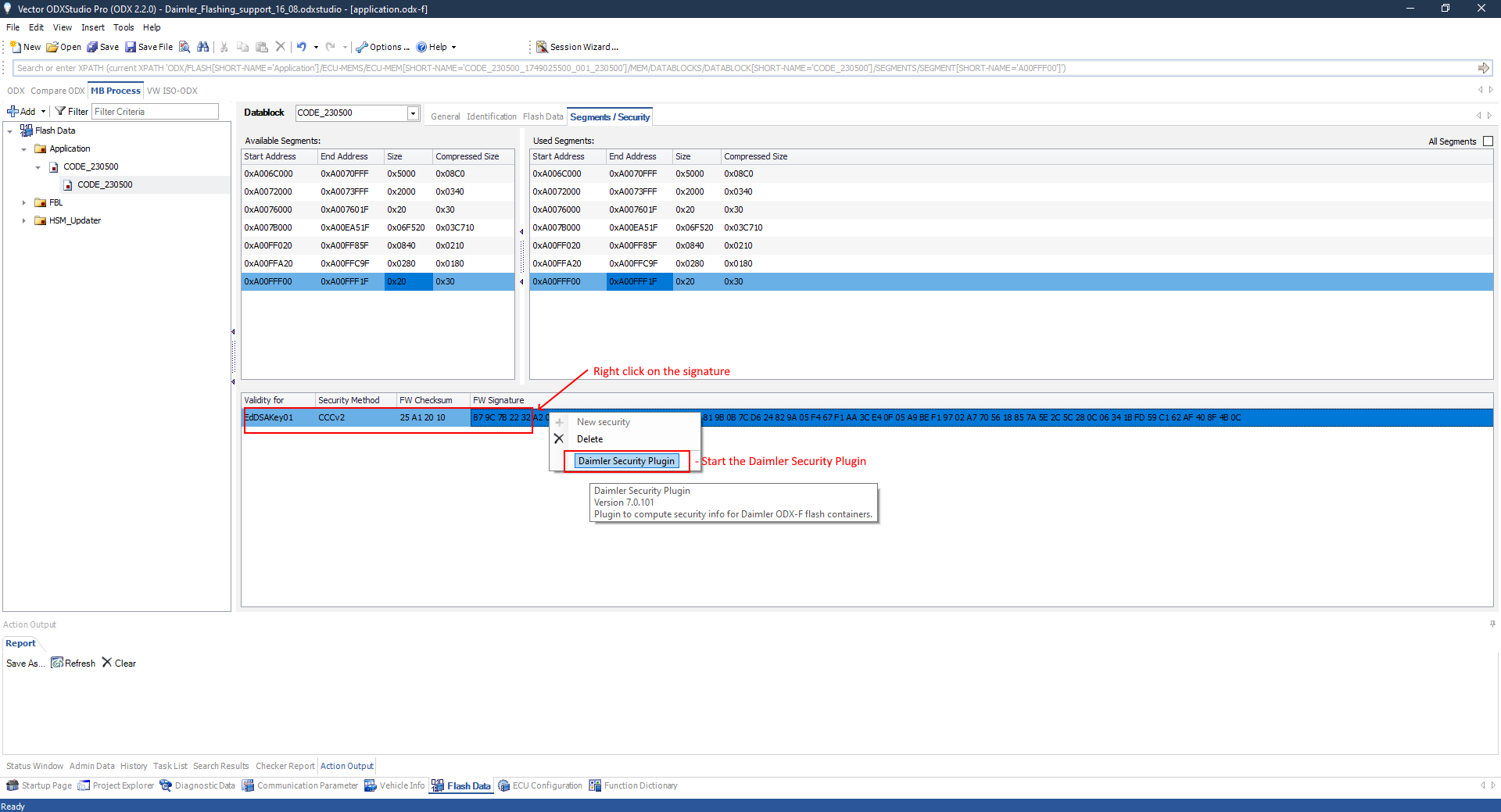


Figure 5 – Start the Daimler Security Plugin

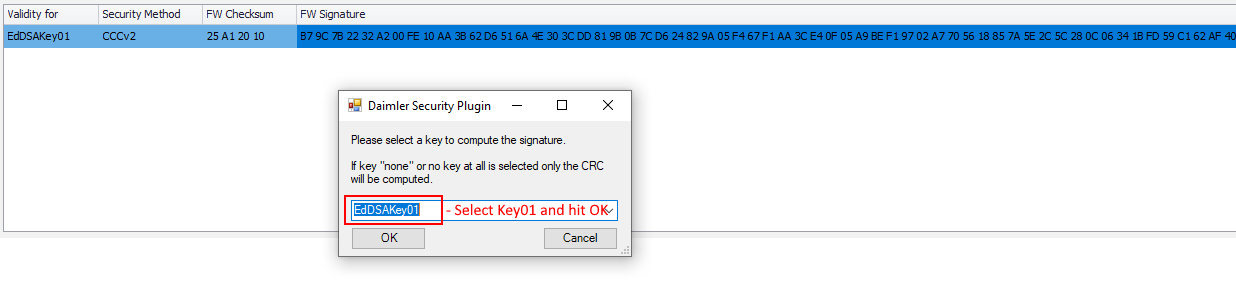


Figure 6 - Select the key to be used for the Signature

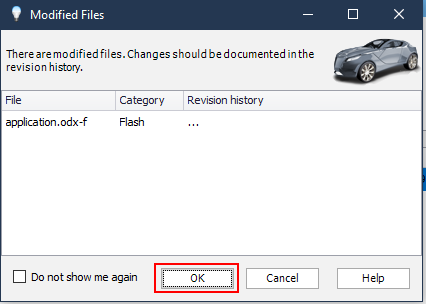


Figure 7 - After you hit CTRL+S, this window appears, it is expected, hit OK

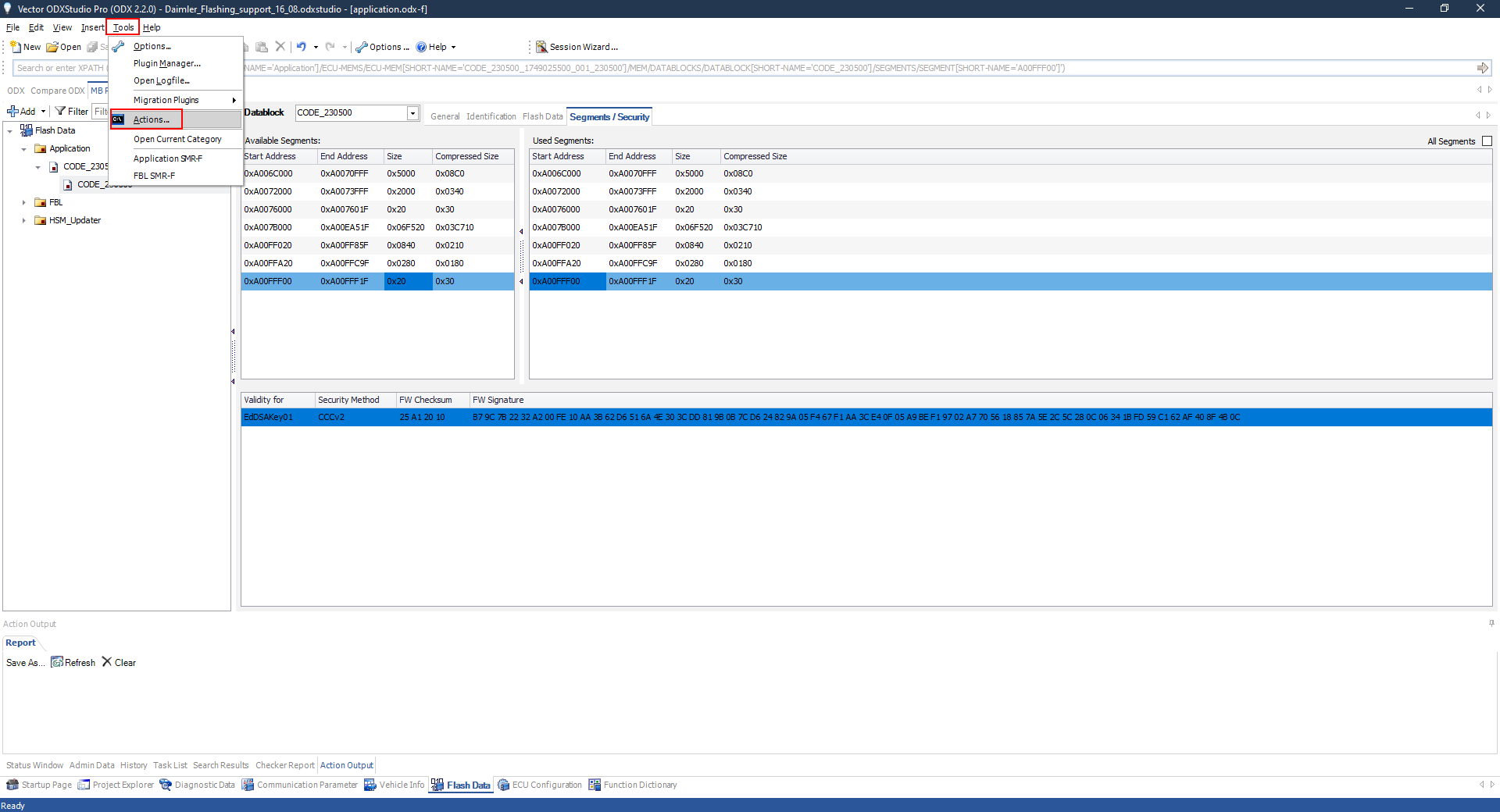


Figure 8 - Go to the Actions menu

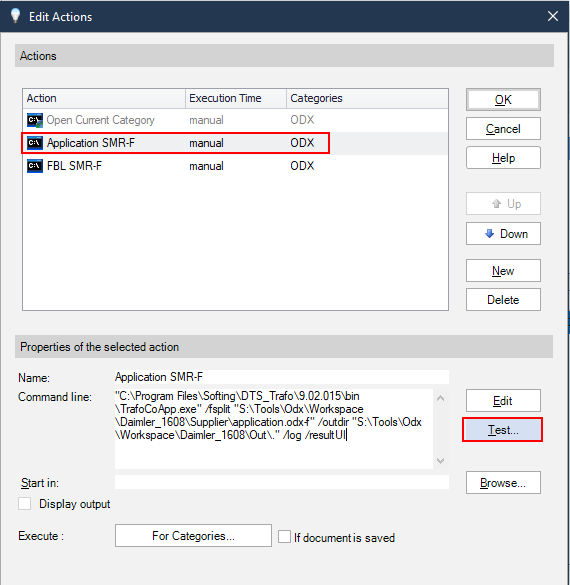


Figure 9 - Select the Application SMR-F Action and hit test

If the action is not available on your PC, just create a new one and add this to the Command line:

*"C:\Program Files\Softing\DTS\_Trafo\9.02.015\bin\TrafoCoApp.exe" /fsplit "S:\Tools\Odx\Workspace\Daimler\_1608\Supplier\application.odx-f" /outdir "S:\Tools\Odx\Workspace\Daimler\_1608\Out\." /log /resultUI*

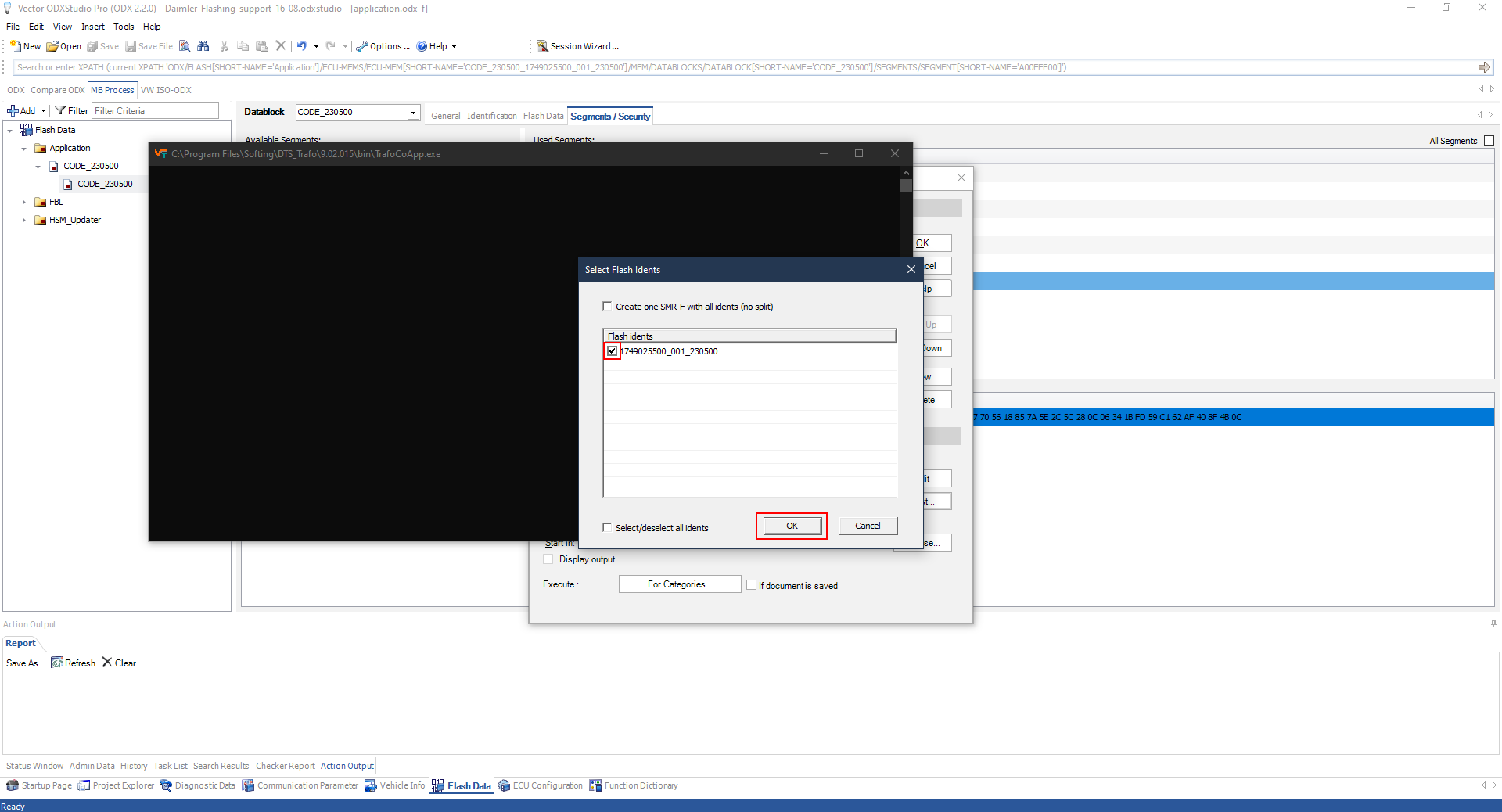


Figure 10 - After hitting Test, this screen will appear that calls Trafo, select the Flash Ident and hit OK, this is a good opportunity to check the Part Number and Version directly in the name of the file

The newly created SMR-F file will be found in: *S:\Tools\Odx\Workspace\Daimler\_1608\Out*

### FBL

Select the same steps as for the application, the only difference is related to naming conventions and paths, since they reflect the FBL part:

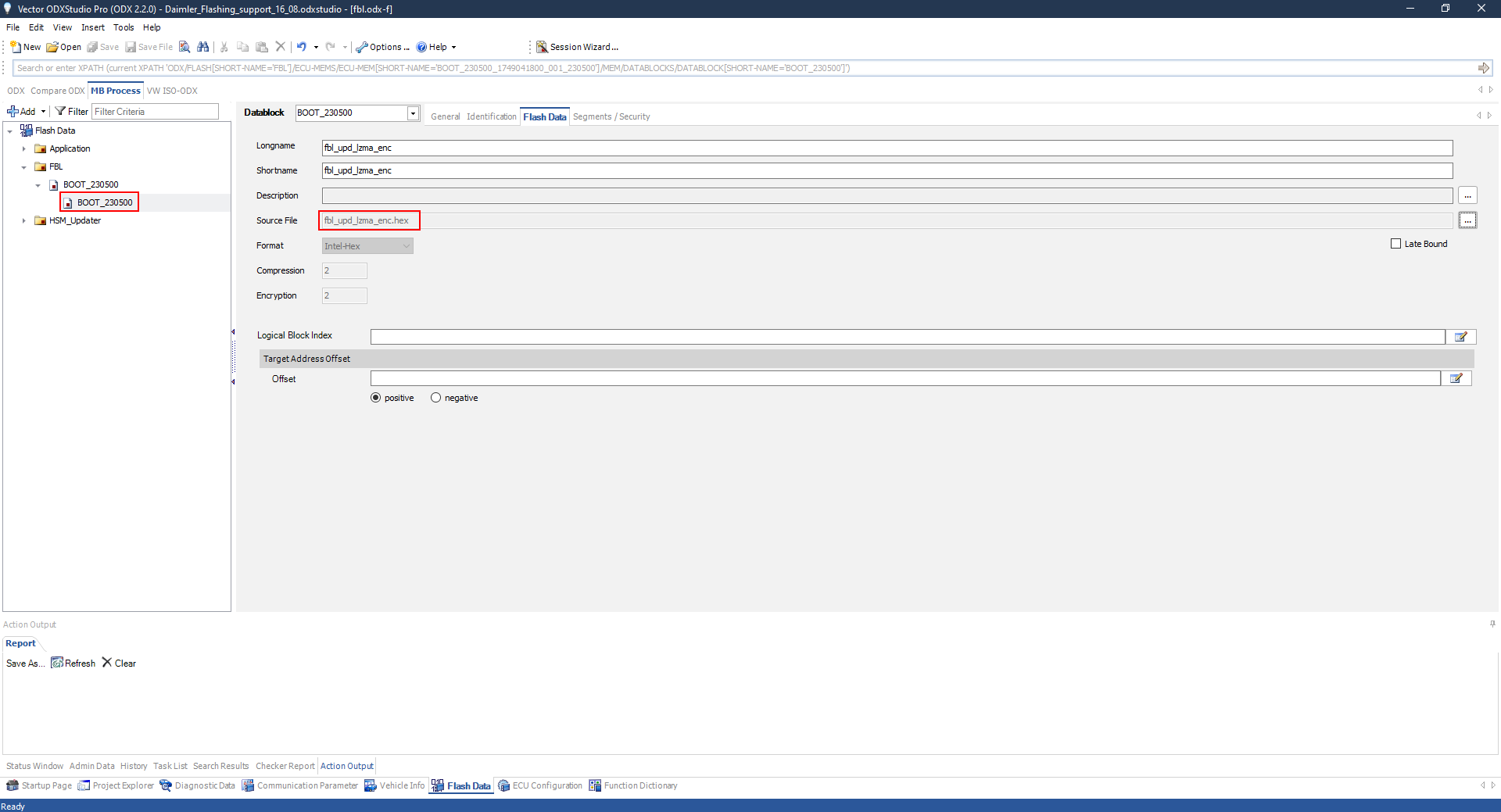


Figure 11 - Configuration of the ODX for the FBL

For the action, use this for the command line:

*"C:\Program Files\Softing\DTS\_Trafo\9.02.015\bin\TrafoCoApp.exe" /fsplit "S:\Tools\Odx\Workspace\Daimler\_1608\Supplier\fbl.odx-f" /outdir "S:\Tools\Odx\Workspace\Daimler\_1608\Out\." /log /resultUI*

### HSM

For the HSM, after the compilation, additional preparation needs to be done before we can generate the SMR-F file.

The first step is to change the Version of the HSM in the following script -> *e:/MKSProjects/SBE/PP/DAIMLER\_MMA/Phase\_02/View\_Development/Tools/Vector\_configuration/Workspace/hsmupdater/Appl/CreatevHsmUpdMultipleModuleContainerAndHeader.bat*

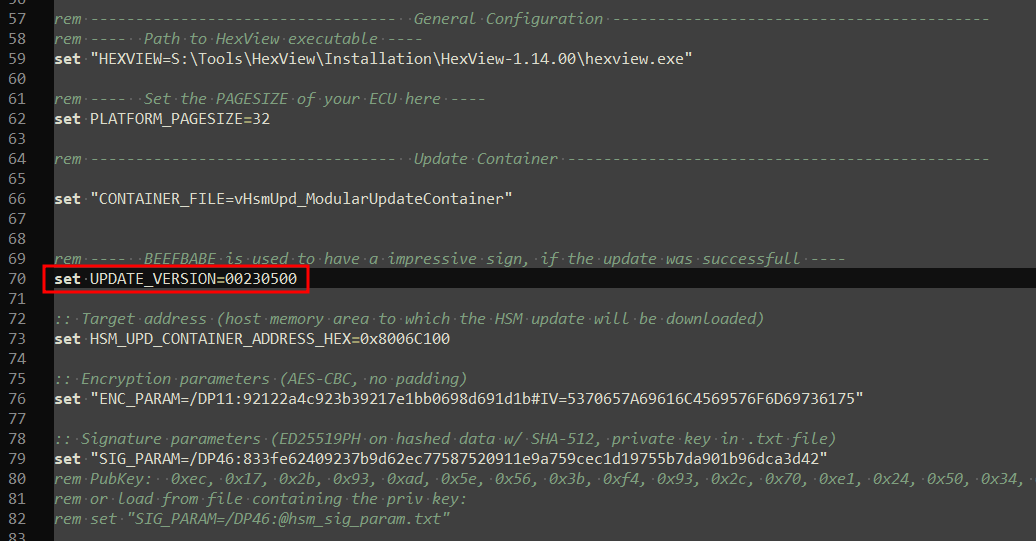


Figure 12 - The HSM Version to be changed

After the version has been changed, we need to run the PrepareRelease.py script -> *e:/MKSProjects/SBE/PP/DAIMLER\_MMA/Phase\_02/View\_Development/Tools/Vector\_configuration/Workspace/hsmupdater/Appl/PrepareRelease.py*

The script does the following things:

* Creates the update container by calling the previous script
* Merges the AppHeader with the main container
* Changes the hex address space to 0xAXXXX
* Alignes the hex file to 32 bytes

The ouput of the script is the following file -> *vHsmUpd\_ModularUpdateContainer.hex*

Copy this file to -> *S:\Tools\Odx\Workspace\Daimler\_1608\Supplier*

Rename it to -> *hsm\_upd\_container.hex*

Now we go to ODX Studio in order to generate the SMR-F file:

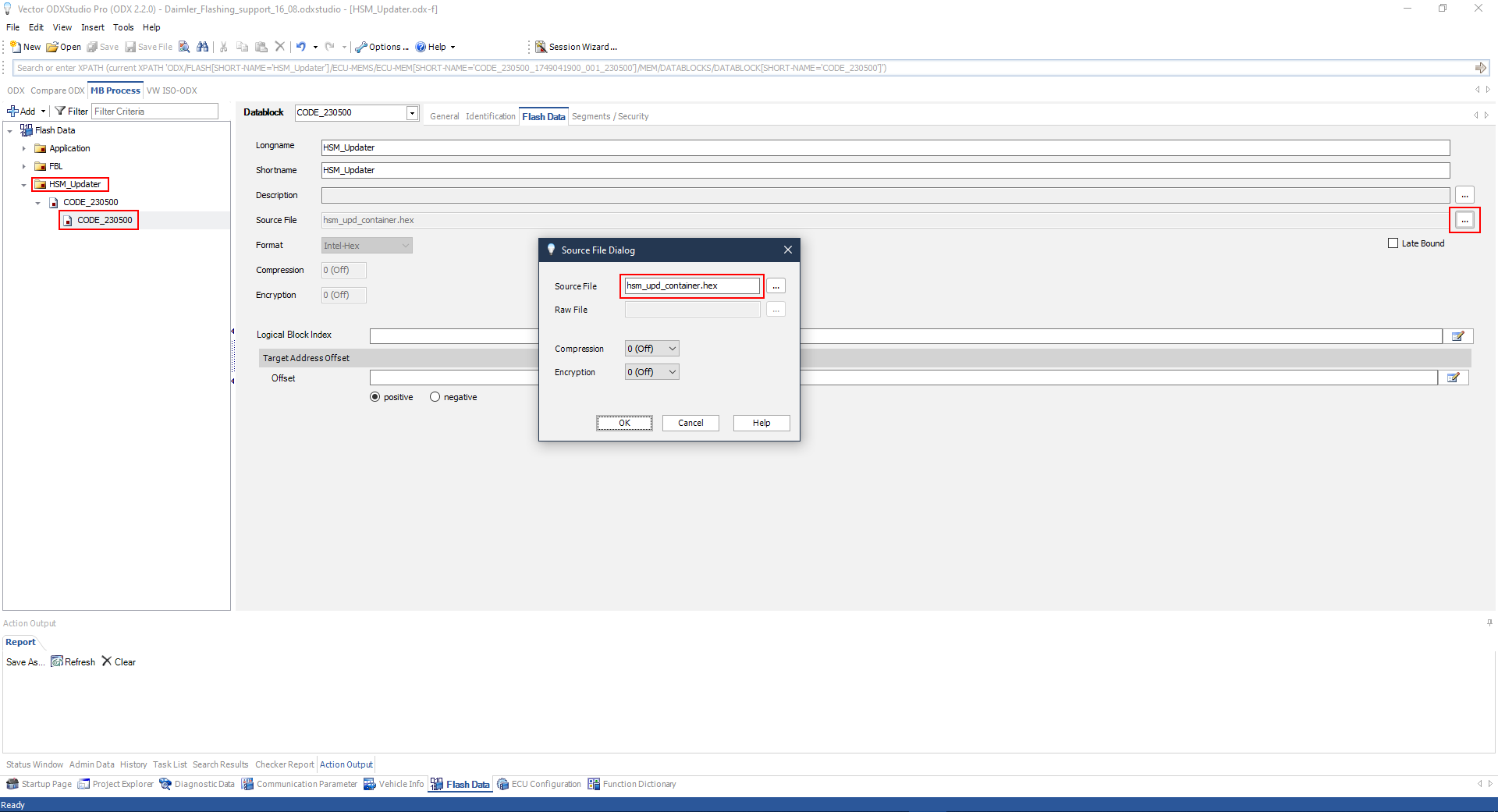


Figure 13 - Select the Update Container

Note: The HSM Container has a different handling for encryption, therefore we don’t do any encryption from ODX, so the Segments/Security step is skipped.

After the correct Container has been selected, we only need to create the file from the Actions menu.

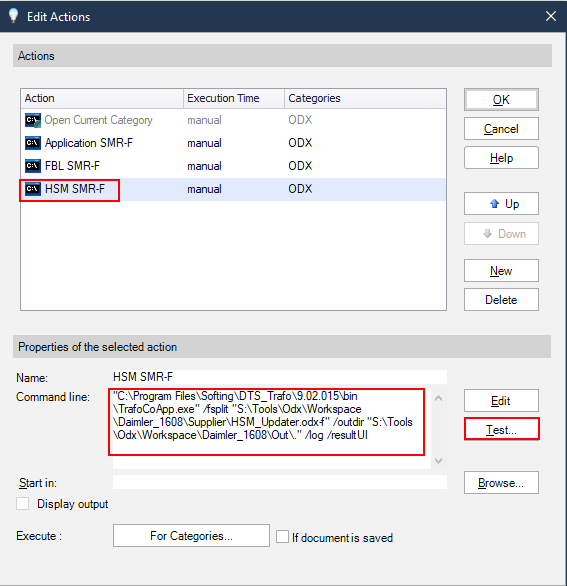
**

Figure 14 - Action for the HSM SMR-F Creation

For the action, use this command line:

*"C:\Program Files\Softing\DTS\_Trafo\9.02.015\bin\TrafoCoApp.exe" /fsplit "S:\Tools\Odx\Workspace\Daimler\_1608\Supplier\HSM\_Updater.odx-f" /outdir "S:\Tools\Odx\Workspace\Daimler\_1608\Out\." /log /resultUI*

# Flashing