MCAL Integration Package

Technical Reference

Basics and workflows

Version 1.01.00

Authors	Andrej Gazvoda, Günther Piehler, Roland Süß, Ingo Wuttke
Status	Released

Document Information

History

Author	Date	Version	Remarks
Roland Süß ; Ingo Wuttke	2015-02-27	1.00.00	Initial Ideas, usage as Application Note; Porting to Technical Reference template; adding detailed description about 3rd party tools etc.
Günther Piehler	2015-04-24	1.00.01	Review; small changes to increase understandability; Known Issue for missing config items added → released
Andrej Gazvoda; Roland Süß	2015-06-30	1.00.02	5.4 / 5.5 - Added known issues regarding EB tresos™ tool
Günther Piehler	2015-07-17	1.01.00	1.3 - Introduction of Mixed AUTOSAR use case 2 ff - Extend description of MCAL preparation (prerequisits) 3 - hint about recommended workflow added

Reference Documents

No.	Source	Title	Version
[1]	Vector	Product Information MICROSAR Vector SLP4	1.03.02
[2]	Vector	Catalog – Product Information MICROSAR – Chapter MCAL	V1.3 – 2015-02

Caution
We have configured the programs in accordance with your specifications in the questionnaire. Whereas the programs do support other configurations than the one specified in your questionnaire, Vector´s release of the programs delivered to your company is expressly restricted to the configuration you have specified in the questionnaire.

Contents

Illustrations

Tables

1 Introduction

The *MICROSAR MCAL Integration Package* covers the integration of a *3rd party MCAL* package into the Vector MICROSAR BSW stack. As part of this Vector performs a setup of the MCAL provided by the customer or the semiconductor vendor and supplements it with items needed for the integration with the MICROSAR BSW.

Typically the semiconductor vendor delivers the MCAL with its own configuration and generation tool chain. Vector provides solutions to deal with the dependencies and interfaces on embedded and on configuration level between the AUTOSAR BSW and the MCAL components (with AUTOSAR4.0.3 these configuration level based interfaces are about 10 parameters).

This document supports the user by launching the delivery, setting up a project and serving the tooling- and configuration-based interfaces between the Vector MICROSAR BSW and the 3rd party MCAL. The term “tooling” means Vector DaVinci Configurator on the one hand and a third party configuration and generation framework (EB tresos™, KPIT ECU Spectrum™ …) on the other.

The following use cases and corresponding workflows are explained and illustrated in detail later on:

1. *Single configuration tool usage*: Usage of the Vector DaVinci Configurator as exclusive configuration tool for both the MICROSAR BSW and the 3rd party MCAL.
2. *Mixed configuration tool usage*: Parallel and synchronized usage of the DaVinci Configurator and a 3rd party MCAL configuration tool.
3. *Split configuration tool usage*: Split configuration and code generation using DaVinci Configurator for the configuration/generation of the MICROSAR BSW and the 3rd party tool for configuration/generation of the MCAL.

In principal all three use cases are supported and the decision which one is used depends on user requirements as well as on the capabilities of the integrated MCAL. Detailed decision guidance and hints will be given in this document.

1.1 Responsibility

Vector does not take over any responsibility for the usage and functionality of the integrated 3rd party components. The *MCAL integration* into the MICROSAR SIP does not include a full functional test for the 3rd party components. It remains the responsibility of the customer to ensure freeness of defect and shortcomings as well as functional completeness of the 3rd party components.

1.2 Support requests

Vector will support you in all aspects of the *MCAL integration* into your MICROSAR SIP. In case of questions or problems please get in touch with your MICROSAR contact (e.g. as defined in the delivery description that is part of your SIP).
Since Vector is not the vendor of the integrated 3rd party components such support requests shall be directed to the 3rd party software vendor as they can provide you with first-hand information and support.

1.3 Mix between AUTOSAR specification versions

Sometimes it is necessary that there is a mixture of components released for different AUTOSAR versions. Most common case of mixture is the use of AUTOSAR 3.x related BSW (because OEMs have fixed their platform SW) with AUTOSAR4.0.x related MCALs. This is based on the fact that there are continuously new HW platforms available through the semiconductor vendors but not all devices will be supported with ASR4- as well as ASR3-based MCALs. In future there will be also the situation of mixture of AUTOSAR4.x based BSW with AUTOSAR3.x based MCALs. This is due to some OEM decisions to start with AUTOSAR4.x but willing to use some older microcontroller devices.
AUTOSAR3.x and AUTOSAR4.x based SW in general is not directly compatible. Nevertheless Vector supports you to be able to mix those components. There are several measures to take and Vector provides solution for this with the term *MCAL Integration Package (mixed ASR)*. There are some wrappers (on embedded as well as on configuration side) that are part of your SIP.

2 First Steps

During *MCAL integration* at Vector several supplements have been created and integrated into your delivery. A short overview of contributed parts shall be given.

2.1 Delivery structure

The following parts of the delivery (MICROSAR SIP) are relevant for the successful *MCAL integration*:

- `BSW\<MCAL_µC>`: Contains the makefiles (AUTOSAR makefiles) for all 3rd party MCAL components (typically all MCAL components are merged into 1 file).
- `BSWMD\<MCAL_µC>`: Contains configuration items for integration into the DaVinci Configurator. After execution of steps described under 2.2 the basic software definition files (VSMD)
- `DaVinciConfigurator\Generator\<MCAL_µC>`: Contains configuration items for integration into the DaVinci Configurator e.g. XML-preparation files as well as files containing the commands for execution of the 3rd party code generator plus optional convenience features like registration of Init-APIs towards the ECUM component recommended configurations etc.
- `ThirdParty\<MCAL_µC>\Supply`: Contains or currently must contain 3rd party MCAL and the tooling provided along with. The structure is defined by the MCAL supplier. Typically the supply folder contains:
  - MCAL implementation (C- and H files)
  - BSWMD description files according to AUTOSAR
  - Tooling for MCAL configuration and generation
  - MCAL related documentation

Caution
Your delivery (SIP) may not finally contain the integrated MCAL because of several reasons (license etc.). Currently it is recommended that the MCAL and its generation tooling, to be used with Vectors SIP, is directly installed into this above mentioned folder `ThirdParty\<MCAL_µC>\Supply` with exactly the default name of your MCALs supplier!

Note
Depending on the deployment of your MCAL supplier (keyword: 1 package for the whole MCAL, several packages for single functionalities like basic IO, memory, communication) you will have to install 1 or more packages into the `Supply` folder of your SIP.

Example with MCAL from 3rd party MCAL of Infineon:

```
ThirdParty\<MCAL_µC>\Supply
.\MC-ISAR_AS4XX_AURIX_TC27X_CA_PB_BASE_V100
.\MC-ISAR_AS4XX_AURIX_TC27X_CA_PB_MEM_V100
.\Tresos
```

> `ThirdParty\<MCAL_µC>\VectorIntegration`: Contains batch files / executables and scripts which support the user during project setup (refer to 2.2)

2.2 Starting up

There are 3 possible scenarios for your SIP which will be explained in detail.

2.2.1 MCAL delivered within Vector SIP

To figure out if this scenario is applied with your SIP just have a look into the above mentioned folders. When they are existent and contain the needed files (e.g. embedded source code, AUTOSAR description files etc.) you do not have to do any other preparation.

2.2.2 MCAL not contained within Vector SIP

With this scenario you have to do some preparations, e.g. install the MCAL etc.

Caution
In order to activate the MCAL and to use the workflows described later on some adaptions have to be made on MCAL packages.

To ease this procedure Vector provides a script to execute these steps automatically. This feature also supports the user during MCAL package updates received from the MCAL manufacturer.

Practical Procedure
Please execute the batch file located here:
`ThirdParty\<MCAL_µC>\VectorIntegration\Script_MCAL_Prepare.bat`
with the option `-prepare`.

If you are facing problems please refer to chapter 5.

In principle the following steps are executed by this batch file, for details please refer to the file itself:

> Prepare MCAL configuration tool by activation of plugins or similar steps

- Delete/Rename MCAL specific `Compiler_cfg.h` and `MemMap.h` within include paths (as those files are already contained within the Vector BSW)
- Copy BSWMD files from a location within the 3rd party folder structure for one specific derivative to `BSWMD\<MCAL_µC>`. If needed you will be asked which derivative shall be taken.
- Create some backup files to undo these actions later on when needed

2.2.3 MCAL Update needed

Usually within your development phase there will be updates of your 3rd party MCAL. Vector supports you for this use case and does some preparations within the MCAL Integration Package phase.

When you are receiving an update of the MCAL please clean up the MCAL Integration.

Practical Procedure

Execute the `Script_MCAL_Prepare.bat` with the option `-undo`.

Install the Update of the MCAL and execute the `Script_MCAL_Prepare.bat` with option `-prepare` again.

Caution

Take a deeper look at the update procedure!

Usually Vector prepares your SIP for the requested MCAL. On updates there may be several traps and pitfalls like:

- Installer path has changed because commonly the install path contains the version information of 3rd party product (e.g. MCAL_V300)
- No full update but only a partial update (either module point of view or file point of view – often named as hotfix)

Both changes have consequences!

- On updated installer path you would have to adapt the `Script_MCAL_Prepare.bat` as well as other artefacts like makefiles. Either request Vector for support or install the update of the product into the old folder (old == initial used for the SIP).
- When there are only partial updates available through your 3rd party MCAL supplier do not completely undo the MCAL Integration but only install the new artefacts into the Supply folder and execute the `Script_MCAL_Prepare.bat` with the `-prepare` option again.

3 Workflow

As mentioned in the introduction there are three reasonable workflows the user can choose from. Which one leads to the highest benefit-cost ratio depends on user requirements / preference on the one hand and on the level auf AUTOSAR conformance of the MCAL on the other hand. In the following all three workflows are introduced and their corresponding prerequisites and advantages/disadvantages are listed for guidance.

Note
Your SIP contains a Release Note (*ReleaseNotes_3rdPartyMCAL_VectorIntegration.pdf*) which describes the recommended / supported workflow.

3.1 Single configuration tool usage

After execution of the first steps described in 2.2 the user is able to use Vector DaVinci Configurator as the exclusive configuration tool for both the MICROSAR BSW as well as the 3rd party MCAL. The configuration of the 3rd party components is based on the MCAL AUTOSAR component definition files (BSWMD) which are (have to be) included in the original semiconductor vendor packages.

You can activate/add the 3rd party components to your configuration in the same way you would activate/add the MICROSAR BSW components.

Vector has integrated the commands for the external code generators so that you will be able to start the MCAL code generation within the Configurator GUI by using this workflow. DaVinci Configurator is used to call the MCAL generator and also displays warnings, errors, and other information.

Prerequisite	
AUTOSAR component definition files provided by MCAL manufacturer must be completely AUTOSAR conform and consistent to possibly used proprietary file formats (used internally by the 3rd party tool).	
Advantages	**Disadvantages**
Only one configuration tool and one configuration file to be used / no redundant data	No usage of convenience features which are possibly provided by MCAL manufacturer but not transferred/modeled within the AUTOSAR component definition files
All dependencies between BSW and MCAL components can be solved easily	

Table 3-1 Guidance for single configuration tool usage

3.2 Mixed configuration tool usage

The configuration tooling respectively the internal (proprietary) description files of some 3rd party MCAL contains internal logic which supports the user during configuration steps. Examples are

- Parameter A is only needed if Parameter B is set to value “X”
- Parameter C is calculated automatically based on A and B
- During component instantiation a list of containers is created automatically with reasonable and different values

These mechanisms are not completely specified by the AUTOSAR standard and not always modeled correctly within the AUTOSAR component definition files provided. The fact is that they are not feasible useable outside the original MCAL configuration tool. As a result and depending on the level of feature convenience of the MCAL tool it is advisable to at least use it for the project setup as it will speed up the process in this phase.

By doing so the user has to deal with two different tools during ECUC setup. But in a later phase it will be easier to use a *one tool solution* as described in 3.1. Thus the user will have to deal with a tool transition. To ease this procedure Vector has introduced some tool features.

The diagram below shows a workflow to use both the features of the MCAL tool and the convenience of a solution which is based on a single tool.

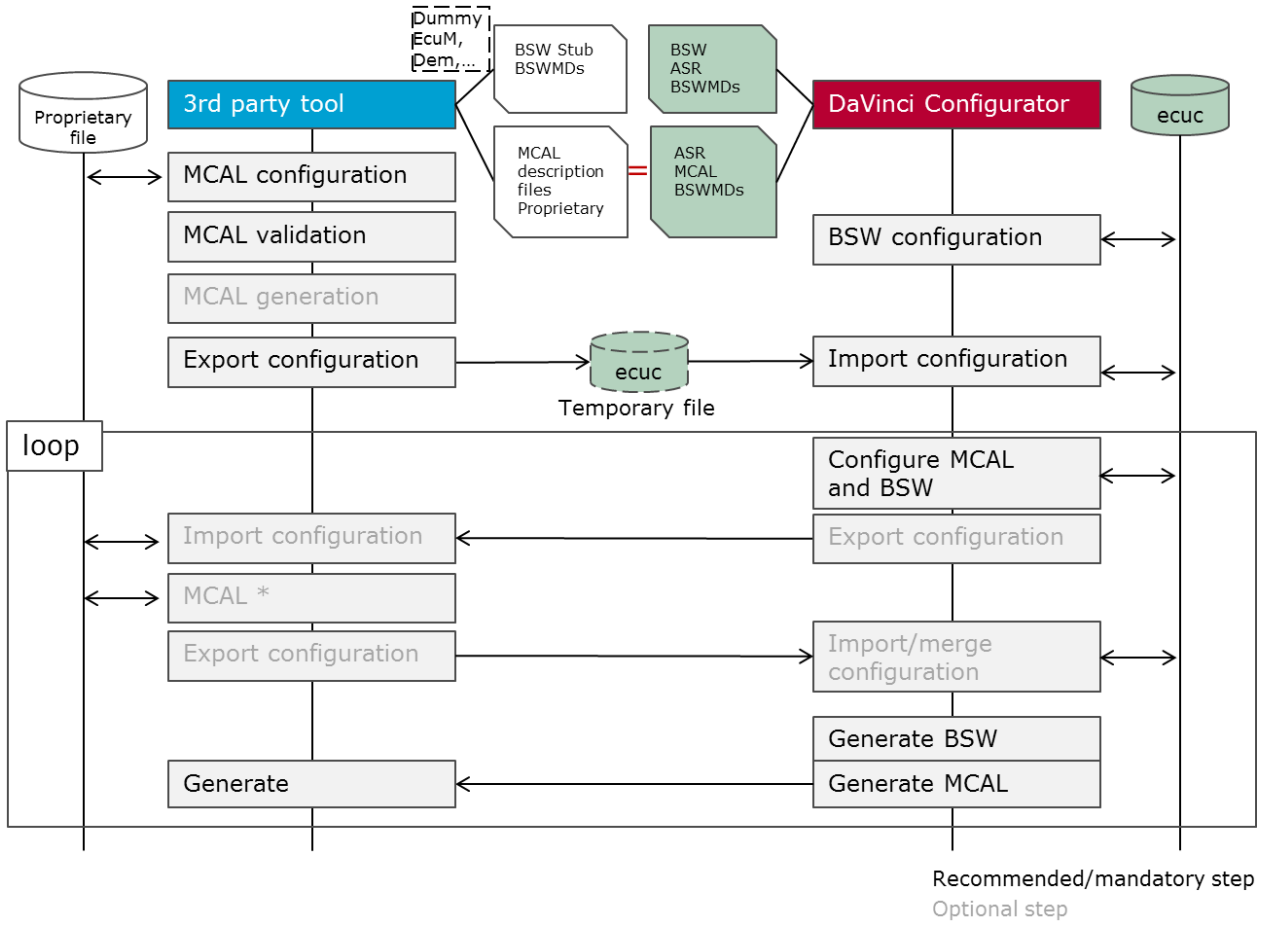


Figure 3-1 Configuration workflow – Mixed configuration tool usage

Prerequisite	
AUTOSAR component definition files provided by MCAL manufacturer must be completely AUTOSAR conform and as consistent as possible to used proprietary files	
Advantages	**Disadvantages**
Usage of convenience features of both tools	Additional tool knowledge is needed
Dependencies between BSW and MCAL components can be solved by data synchronization (export/import)	Overhead due to export/import workflow
Switch to single configuration tool and configuration file possible	

Table 3-2 Guidance for mixed configuration tool mode

Practical Procedure (startup)

1. Create a DaVinci project
2. Activate BSW components needed, but no MCAL components
3. Create a project within the 3rd party configuration tool (see 4.2)
4. Configure all MCAL components needed within the 3rd party configuration tool
5. Export all MCAL components from the 3rd party tool to an AUTOSAR conform ECUC file
6. Import this file in DaVinci Configurator

3.3 Split configuration tool usage

This workflow is required if the third party configuration tool is not completely AUTOSAR compliant as it e.g. violates the AUTOSAR standard parameter definition or does not support a lossless ECUC roundtrip with DaVinci Configurator or other AUTOSAR tools.

As the number of interfaces respectively the number of exchanged parameters between the AUTOSAR BSW and the MCAL components is comparatively small it may also be a solution to use both tools permanently in parallel. The implication is that the code generation will be triggered from the tool in which the configuration has been made.

In order to integrate both parts on embedded side there is still the need to share some parameters between both configuration tools. This can either be achieved by manual replication (using configuration stubs, e.g. DEM stub of 3rd party MCAL, MCU stub of Vector) or by using the import export mechanisms introduced in 3.2.

Prerequisite	
AUTOSAR component definition files provided by MCAL manufacturer must be available but must be matching to proprietary formats only partly (at interfaces between BSW and MCAL).	
Advantages	**Disadvantages**
Relatively tolerant workflow in case that AUTOSAR component definition files and/or 3rd party tools are not completely AUTOSAR conform	Dependencies between BSW and MCAL components must be solved manually
	Additional tool knowledge is needed
	Overhead due to 2 tool workflow and redundant configuration data → manual synchronization needed

Table 3-3 Guidance for split configuration tool usage

4 Configuration tools

In this chapter some hints for the configuration tools included in your package are given.

Note
Most common tooling used by 3rd party MCAL supplier is EB tresos™ therefore the usage is explained extensively.

4.1 Vector DaVinci Configurator

Use DaVinci Configurator to configure your MICROSAR BSW using a DPA project.
Please refer to the startup manual included in the delivery. In addition the DaVinci Configurator provides user support via the help menu.

4.2 EB tresos™

The program itself is located in the folder beneath: `ThirdParty\<MCAL_µC>\Supply\`

To start the tool the user must execute tresos_gui.exe which can be found usually beneath `Tresos\bin`.

A useful documentation when you are struggling with a task can be found within the tresos folder structure by searching for a document named "*Studio_documentation_users_guide*".

4.2.1 Setting up a new Configuration Project

> To create a new project the Project Wizard must be started by to selecting File → New → Configuration Project.

> The project name as well as the workspace location plus the relevant AUTOSAR Release Version can now be chosen.

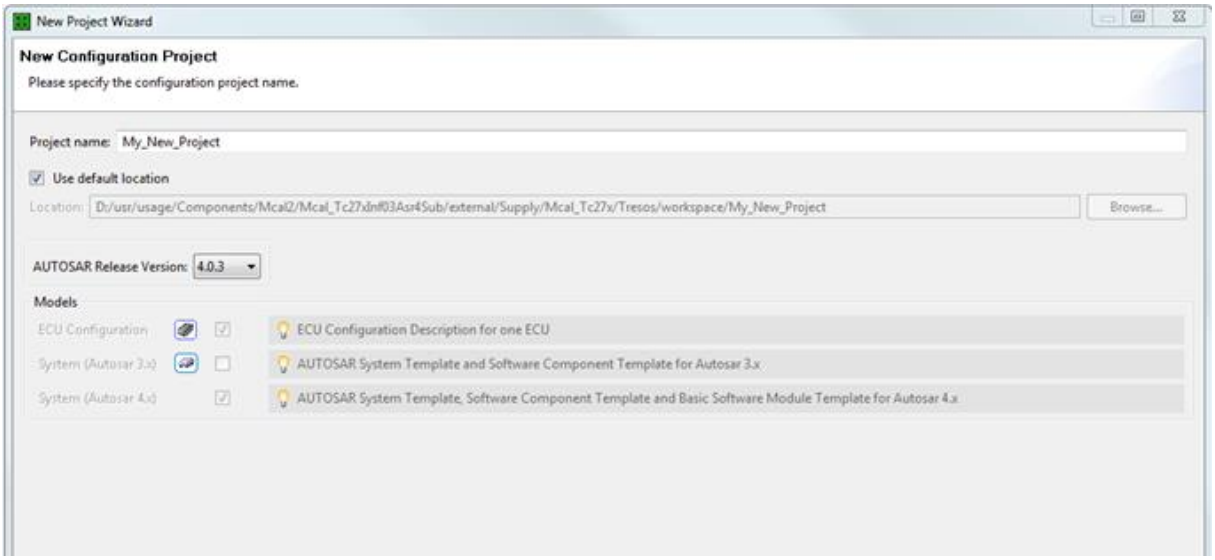


Figure 4-1 New Configuration Project

> Hint: The workspace location can be chosen freely the path for code generation output can be set in the next step.

4.2.2 Project Details

In the next step the wizard request the following data:

> ECU ID: fill in a symbolic name

> Target: Select your microcontroller target

> Generation Path: please choose the settings according to the settings within the DaVinci Configurator to keep consistency and avoid problems regarding make process

> The remaining choices can be selected such as in the figure below

> We recommend to activate the check box to ‘Automatically add the minimum number of child elements in lists’

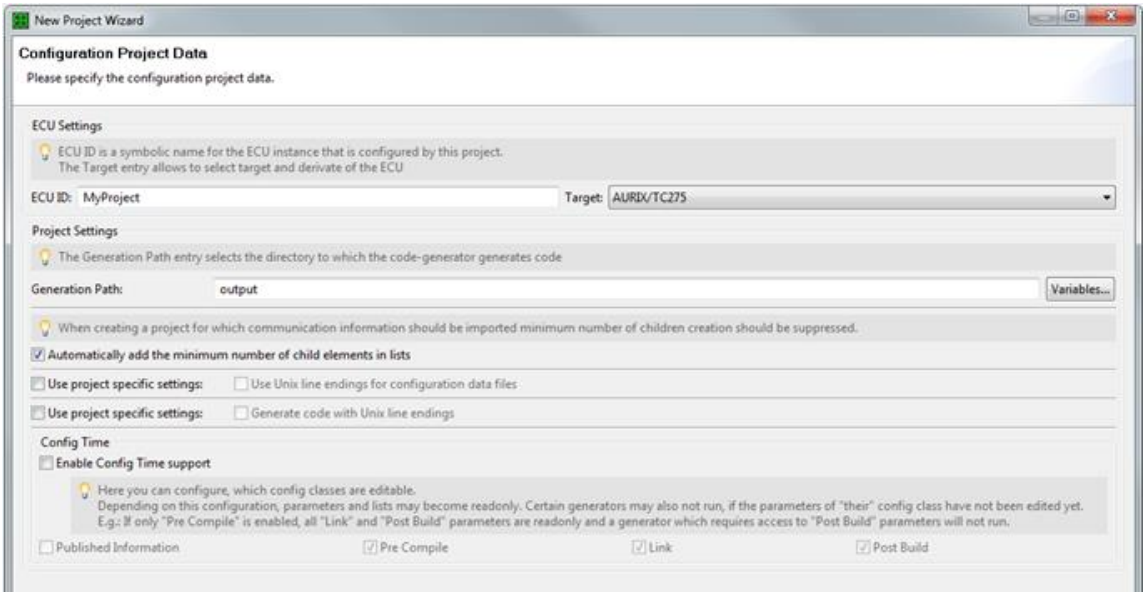


Figure 4-2 Configuration Project Data

> Hint: Several project settings can be changed after creation of the project by editing the file `preferences.xdm` located within the created workspace folder; e.g. adapt the generation path by changing the default value ‘output’ to a relative or absolute path (`<d:var name=”GenerationPath” value=”output”/>`)

4.2.3 Selection of components

Now choose the components, which should be added to the project. In general, every MCAL component should be selected. In that case, use STRG + A to select all and click on the tagged button.

Attention: Some additional “dummy” components (stubs) might be necessary in order to configure the MCAL components e.g. Base, Resource, Dem or EcuM. Those components are only used to provide parameters and references needed by the MCAL components such as *Dem Events* and *EcuM run modes*, please refer to 4.3.

The window should now be similar to this:

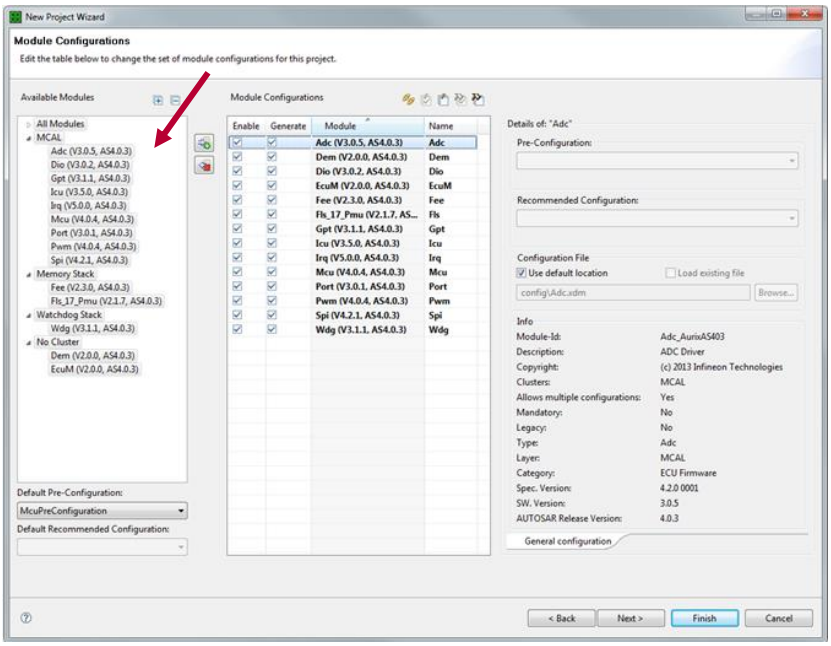
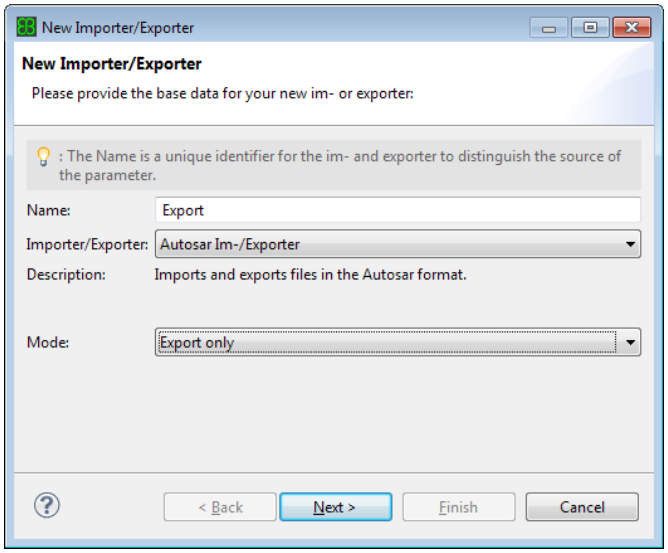


Figure 4-3 Component Configurations

4.2.4 Creation of importers and exporters

Now the user has the possibility to create importer and/or exporter to create a configuration file in an AUTOSAR standardized format out of the proprietary tresos™ configuration files. This functionality is needed to enable the workflows described in chapter 3.2 and 3.3.

Please Click on + to add an Im-/Exporter and use the following settings:



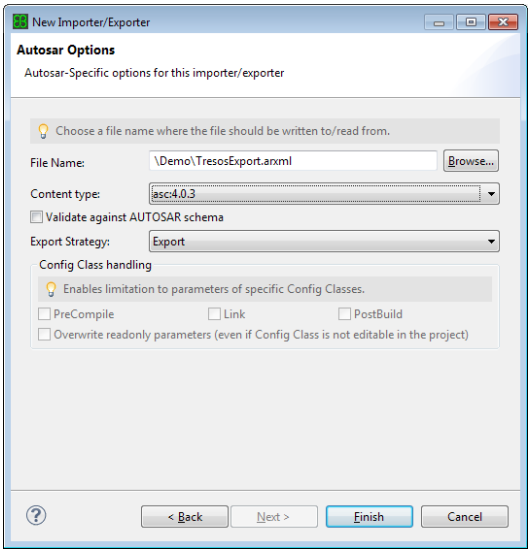


Figure 4-4 Create an exporter (step 1)
Figure 4-5 Create an exporter (step 2 – AUTOSAR options)

The importer can be configured in a similar manner.

If you don't create the im-/exporters during setup you can do this later by using the context menu on the corresponding project.

4.2.5 Configure the MCAL components

Now the wizard is finished and the MCAL itself can be configured.

4.2.6 Generation of the 3rd party MCAL

If everything is configured, click on the generation button. Tresos will now generate your project (note: there will be a validation phase triggered before to ensure that configuration is formal correct).

Figure 4-6 Generate Button

4.3 Configuration hints for parallel usage of DaVinci Configurator and EB tresos™

One important thing to consider is that the configurations for Dem and EcuM must be consistent between DaVinci Configurator and tresos as these are interface components provided by Vector that also have impact on the 3rd party software. Affected configuration entities are `Dem/DemConfigSet/DemEventParameter` and `EcuM/EcuMConfiguration/EcuMCommonConfiguration/EcuMWakeupSource`.

The following chapters handle the configuration direction from upper layer to lower layer like mentioned above, but please are aware that the direction might also be inverse, i.e. for configuration of *Dio Ports*, *Mcu ConfigSets*, *Mcu ClockReferences* or *Fee Blocks*. If you use the workflows described in 3.1 and 3.2 this is done by export and import features.

4.3.1 Vector DaVinci Configurator 5

You can find the configuration of e.g. DemEventParameter in the Basic Editor.

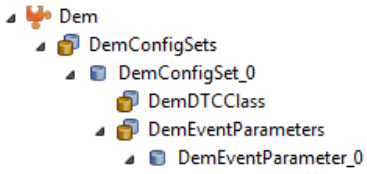


Figure 4-7 DEM Path using DaVinci Configurator 5

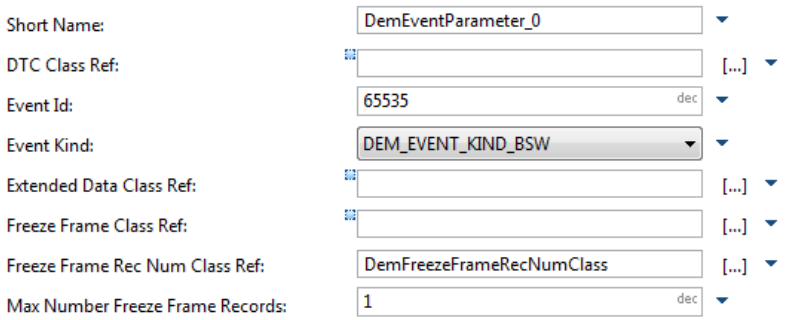


Figure 4-8 Settings within DaVinci Configurator 5 Pro

4.3.2 EB tresos™

> One fast way to change the configuration for DEM in tresos™ is with the Outline window.

> To do so you, double-click on DEM. Now, if you have Outline opened, you will see the folder structure of DEM.

> Navigate to `DemEventParameter_0`, select the same `Event ID` and `Event Kind` as in the DaVinci Configurator Pro.

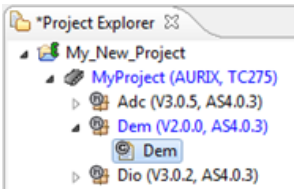


Figure 4-10 DEM-Path within tresos™

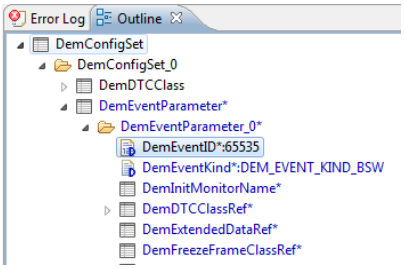


Figure 4-9 DEM-Path in the Outline window of tresos™

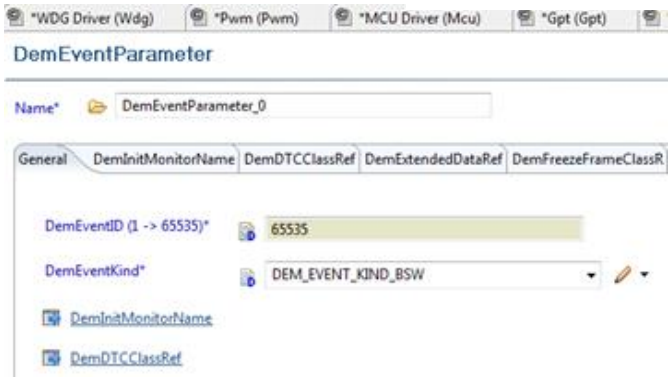


Figure 4-11 Settings for DEM within tresos™

5 Known Issues

This chapter describes know issues which are related to MCAL integration package but are not fixable by Vector without support of 3rd parties. Vector is working on solutions in coordination with the other involved parties.

5.1 MCAL and SIP storage location using Vector Makesupport

Currently the 3rd party MCAL as well as the SIP has to be stored at the same local drive (e.g. `D:\`) because of the Build-/Makeprocess.

Typically the 3rd party MCAL installer comes along with a default drive which does not fit to the drive the SIP is installed. Please be aware of this restriction during MCAL installation process (Clarification: MCAL installation process is something different as the MCAL integration steps described in former chapter 2.2)

5.2 Long path names

Caution
The 3rd party deliverables are currently stored / copied into the Vector delivery structure and may use very long path names. The Vector delivery structure itself usually resides some folders below the root directory. It may happen that some Microsoft Windows™ tools are not able to access those files anymore due to resulting path names which are too long.

Currently Vector is working on a solution for this problem.
As long as there is no final fix we recommend to install the Vector SIP as well as the 3rd party MCAL as near to the root-directory as possible (e.g. `D:\<ECU-Acronym>` like `D:\BCM`)

5.3 Missing configuration items within imported configuration

Sometimes it happens that 3rd party tools do not export all configuration items / information into AUTOSAR standard format of ECUC.

Currently there is no workaround available for this. The user will be informed within the import process into DaVinci Configurator. After importing there must be manual editing of the configuration to add all of those “missed information” (Note: mainly these are not used by 3rd party tooling but relevant for consistent configuration with components BSWMDs).

Note
Due to the fact that 3rd party supply must be AUTOSAR conform please ask your MCAL vendor to provide an update of the corresponding artefacts.

5.4 Configuration Export with EB tresos™ Version 13.0.0

After you have exported the EB tresos™ configuration, you have to close the project before you proceed the export the next time. If not, the export will be faulty and will cause problems during import into DaVinci Configurator.

5.5 Error messages regarding CommonPublishedInformation with EB tresos™

After setting up a MCAL project with EB tresos™ error messages might occur for each parameter of container CommonPublishedInformation like follows:

```
Invalid value for node "/AUTOSAR/TOP-LEVEL-
PACKAGES/Wdg/ELEMENTS/Wdg/CommonPublishedInformation/VendorId
": Value "" is no number
```

Closing EB tresos™ tool once and opening again resolves the problem. This issue affects all MCAL modules.

6 Glossary and Abbreviations

6.1 Glossary

Term	Description
3rd party components / MCAL	BSW components that have been provided by a company other than Vector. Vector may have integrated the software within the SIP but does not take over any responsibility with regard to the functionality of these components.
DaVinci Configurator	Vectors configuration and generation tool of the MICROSAR components

6.2 Abbreviations

Abbreviation	Description
AUTOSAR	Automotive Open System Architecture
BSW	Basic Software
BSWMD	Basic Software Module Description; AUTOSAR standard file
DEM	Diagnostic Event Manager; AUTOSAR BSW component
DPA	DaVinci Project Assistant
ECU	Electronic Control Unit
ECUC	ECU Configuration; configuration of all BSW as well as MCAL components (Note: AUTOSAR also supports ECUC splits containing just a subset of components with its configuration)
ECUM	ECU State Manager; AUTOSAR BSW component
(G)UI	(graphical) User Interface
MCAL	Microcontroller Abstraction layer
MICROSAR	Vectors solution / brand for AUTOSAR BSW
SIP	Software Integration Package (as provided by Vector); MICROSAR delivery containing customers' selection of MICROSAR BSW components including all parts of *MCAL Integration*.
StMD	Standard Module Definition also known as BSWMD
VSMD	Vendor Specific Module Definition (within AUTOSAR methodology it is derived from StMD)

7 Contact

Visit our website for more information on

- News
- Products
- Demo software
- Support
- Training data
- Addresses

www.vector.com