Discover the Experience

EB tresos® Studio

new and noteworthy

product release 16.0

EB tresos® Studio

Elektrobit Automotive GmbH
Am Wolfsmantel 46
91058 Erlangen, Germany
Phone: +49 9131 7701 0
Fax: +49 9131 7701 6333
E-mail: info.automotive@elektrobit.com

Technical support

Europe	Japan	USA
Phone: +49 9131 7701 6060	Phone: +81 3 5577 6110	Phone: +1 888 346 3813

Support URL

http://automotive.elektrobit.com/support

Legal notice

Table of Contents

1. Preface

This document describes the highlight changes of EB tresos Studio and gives short examples of their application. This document does not document all changes in the current release but only the ones that need special attention.

A comprehensive list of the customer-visible EB tresos Studio changes can be found in the EB tresos Studio release notes document. The chapters of this document group the changes according to the EB tresos Studio version they are contained.

2. Interpretation of version information

The EB tresos *product line release* schedule contains two product line releases per year. Product line releases are named after the year of release and a variant represented by a letter. Examples for product line release version numbers are *2009.a*, *2009.b*, *2010.a*, etc.

The EB tresos product line version number can be found on the splash screen, the commandline and the **EB tresos details** dialog.

EB tresos Studio is released as part of the EB tresos product line. Each product within the product line is assigned an individual *product version number*.

The EB tresos Studio product version number can be found on the splash screen and the **EB tresos details** dialog next to the keyword *Studio*.

The EB tresos Studio product version number scheme is made of three parts: the major, the minor and the patch number. The numbers are separated by dots in the following naming scheme: `<major>.<minor>.<patch>`.

- An increment of the *major* version number indicates that the release contains major new features and changes compared to the previous major version.
- An increment of the *minor* version number indicates that the release contains minor new features and changes compared to the previous minor version.
- An increment of the *patch* version number indicates that the release fixes known problems.

There may be a *qualifier* additionally to the product version number. This *qualifier* is provided in brackets and displayed after the product version number. Examples for qualifiers are: *alpha*, *beta*, *customer-exclusive pre-release*, ...

- If the qualifier is *alpha* (e.g. *10.0.3 (alpha)*), the release is a pre-release. Pre-release means that not all planned features are implemented (i.e. not *feature complete*) and the pre-release just fulfills basic quality tests.
- If the qualifier is *beta* (e.g. *10.0.4 (beta)*), the release is a pre-release that implements all planned features but only fulfills basic quality tests.

Examples for spelled out EB tresos Studio product versions are:

- *EB tresos Studio 10.0.0 (alpha)*
- *EB tresos Studio 10.0.4 (beta)*
- *EB tresos Studio 10.0.6*
- *EB tresos Studio 10.1.0 (exclusive for customer xyz)*

3. Changes for release 16.0

3.1. AUTOSAR 4.1.3 support

EB tresos Studio now provides support for the relevant parts of the ECU configuration and system model for AUTOSAR release 4.1.3

Prior releases of EB tresos Studio supported AUTOSAR releases up to revision 4.1.2, which are still fully supported.

3.2. Preliminary AUTOSAR 4.2.1 support

EB tresos Studio already supports (parts of the) future AUTOSAR 4.2.1 system model. Since AUTOSAR 4.2.1 is not yet fully specified, this part of EB tresos Studio is subject to incompatible changes in future EB tresos Studio versions.

3.3. ECU Configuration search extended

3.3.1. ECU Configuration search now supports parameter comments

The **ECU Configuration Search** dialog now provides the possibility to search for parameter comments, see Figure 3.1, “The **ECU Configuration Search** dialog”. For example, it is possible to search for the following:

- Values in summary and description
- Comments in a special status, e.g. *todo*
- Comments written by a specific user

3.3.2. ECU Configuration search now supports config classes

The **ECU Configuration Search** dialog now provides the possibility to search for parameters of specific config classes, see Figure 3.1, “The **ECU Configuration Search** dialog”. You can limit the search to one or more config classes.

Each section in the **limit to** group is connected using logical AND operation. Inside each section, the chosen limitations are connected using logical OR operation.

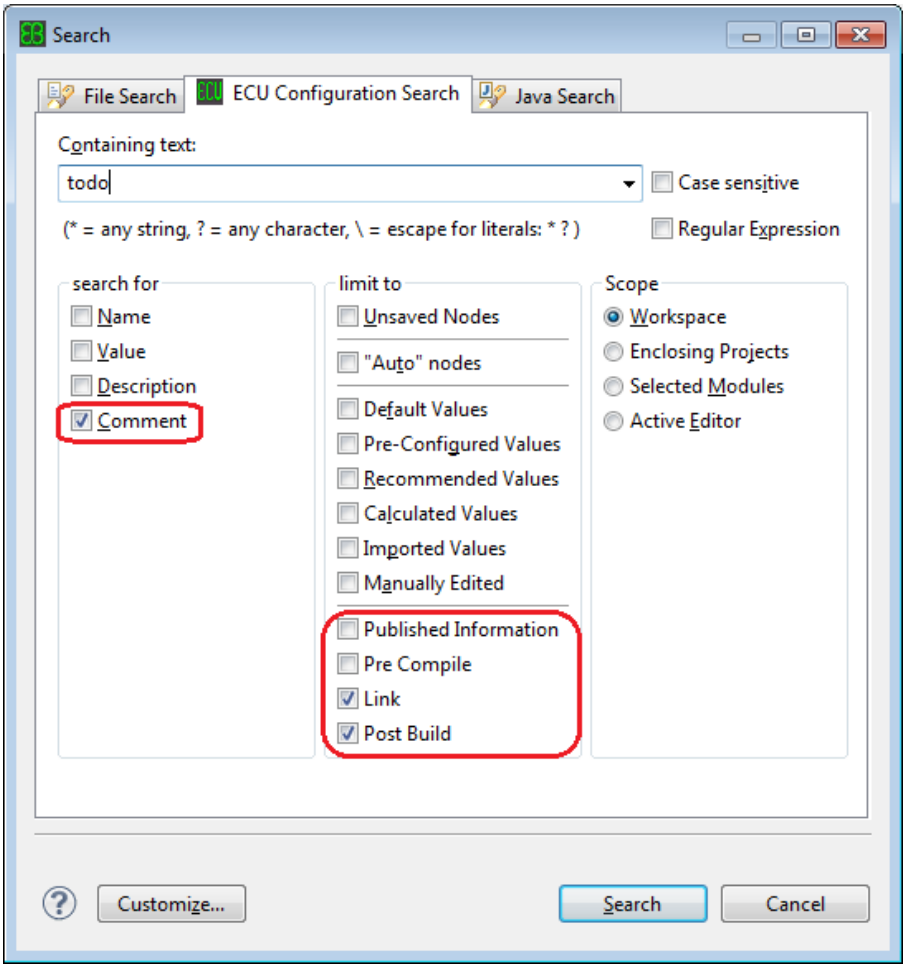


Figure 3.1. The **ECU Configuration Search** dialog

With these settings you can search for all parameters inside the workspace that have a parameter comment with text or status *todo* and have either config class *Link* or *Post Build*.

3.4. Enhancements for unattended wizards and workflows

3.4.1. Project-specific workflow files

Until now it was only possible to register workflows by adding an extension point to a plug-in or to register them via Java API. These workflows are available for all projects in the workspace.

Now it is possible to register a workflow for a single project. Follow these steps:

- Create a folder `.workflows` right beneath the `config` folder.
- Create a workflow `.xml` file and add your workflow description according to the specification in the EB tresos Studio developer's guide.
- Register the workflow. If the **Workflows** view is already open, either close and reopen the **Workflows** view or the project.

After you executed these steps, you can select the project-specific workflow in the **Select a workflow** dialog.

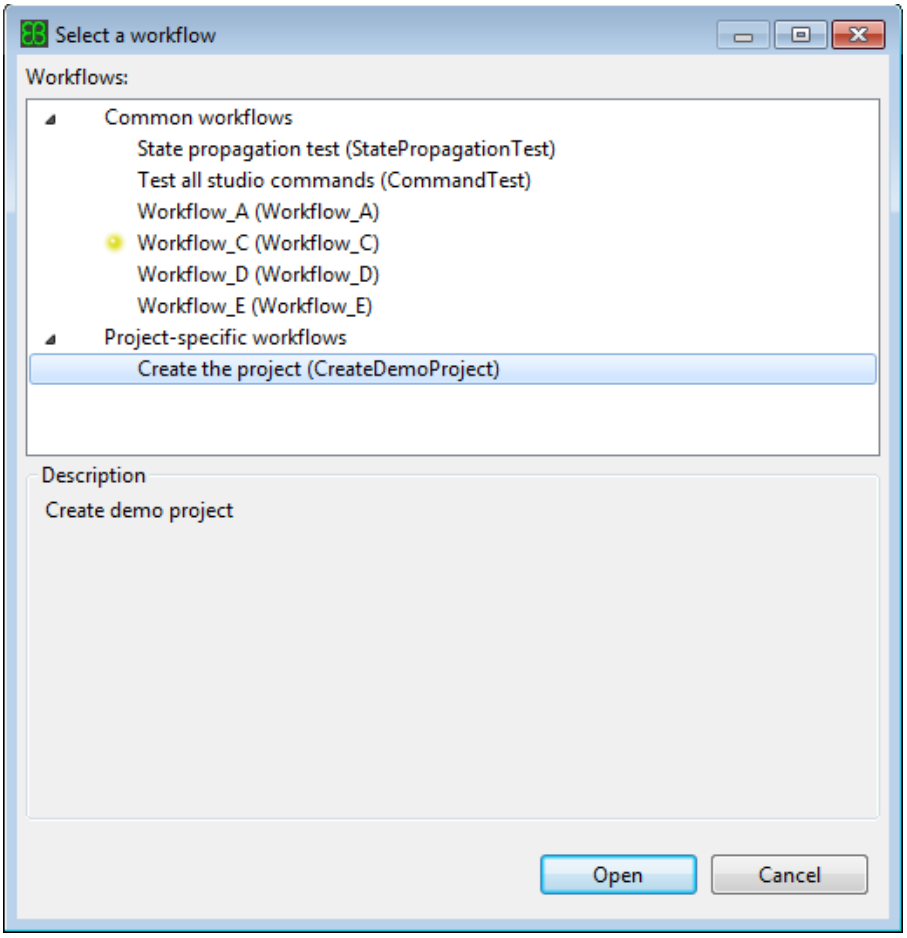


Figure 3.2. The **Select a workflow** dialog

NOTE **Availability of project-specific workflows**

A project-specific workflow is only available if you select the project from where it is registered. Accordingly for all other projects in your workspace the workflow will not be visible. If you need this workflow for another project too, copy the respective workflow description `.xml` file below a `.workflows` folder in the other project.

WARNING **Project-specific workflow IDs may not collide with IDs of common workflows**

The ID of a project-specific workflow may not exist within the commonly registered workflows. Else the registration will fail and an error is logged in the **Error Log** view. However, it is possible to register the same project-specific workflow description in several different projects.

3.4.2. New workflow command for ECU extract creation available

Until now the ECU extract creation was only performed after running a system description importer. Now EB tresos Studio supports a new workflow command to create an ECU extract for a configuration project.

Use the following command to configure and start ECU extract creation:

```
dreisoft.tresos.workflow.api.plugin.EcuExtractCreatorCommand()
```

This command will open the ECU extract creator configuration dialog where you can configure which settings shall be used for ECU extract creation.

Use the following command to skip the configuration dialog and therefore use the previously configured settings to run the ECU extract creation directly:

```
dreisoft.tresos.workflow.api.-
plugin.EcuExtractCreatorCommand(skipConfigDialog=true)
```

If the ECU extract creation was never configured before, an error appears.

3.4.3. Multiple instances for unattended wizards

Until now there was only one instance available for each registered unattended wizard. Therefore if a task (e.g. Calculate Handle IDs) was necessary in different configurations to support several use cases, it was necessary to change the configuration each time before running it.

The **Unattended Wizards** configuration dialog now provides the possibility to create several instances for each registered unattended wizard. When you open the **Unattended Wizards** configuration dialog for the first time,

it shows up as known before. Every installed and available unattended wizard is shown once in the list on the left part of the dialog.

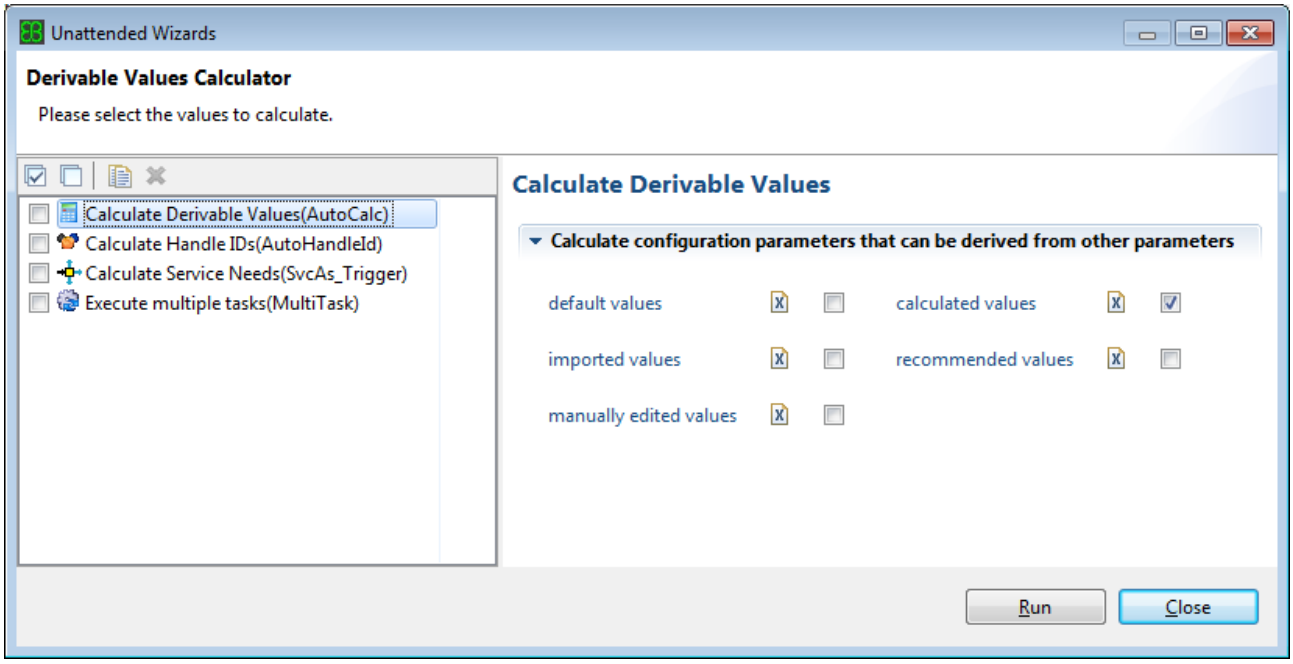


Figure 3.3. The **Unattended Wizards** configuration dialog

The configuration ID is shown in parentheses with which you can start the respective unattended wizard from the command line: `tresos_cmd.bat autoconfigure <projectName> <triggerId>` or via workflow command `AutoConfigureTriggerCommand`.

It is now possible to create another instance of an existing unattended wizard configuration by duplicating it. Just select one unattended wizard configuration and click the toolbar button **Copy the selected unattended wizard configuration** next to the **Uncheck all for manual run** button. A message box appears where you can choose a configuration ID for the new instance. The message box already provides a suggestion for the new name.

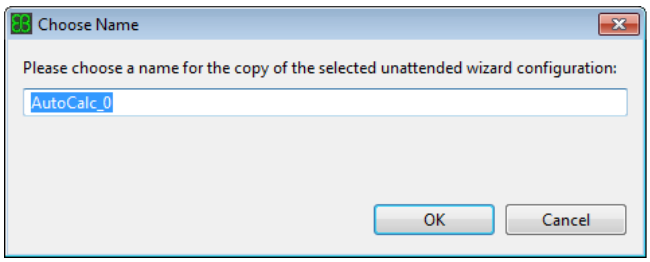


Figure 3.4. Message box with a suggested new name

The setting of the checkbox in front of the wizard configuration will not be copied.

It is possible to delete all instances which you have created on your own. The initial instances cannot be deleted at all!

3.4.4. Unattended wizard for executing multiple tasks

In the lifecycle of a project, it is sometimes necessary to perform a set of actions again and again. To be able to configure and run a set of actions repeatedly, a new unattended wizard is provided - the **Execute multiple tasks** wizard.

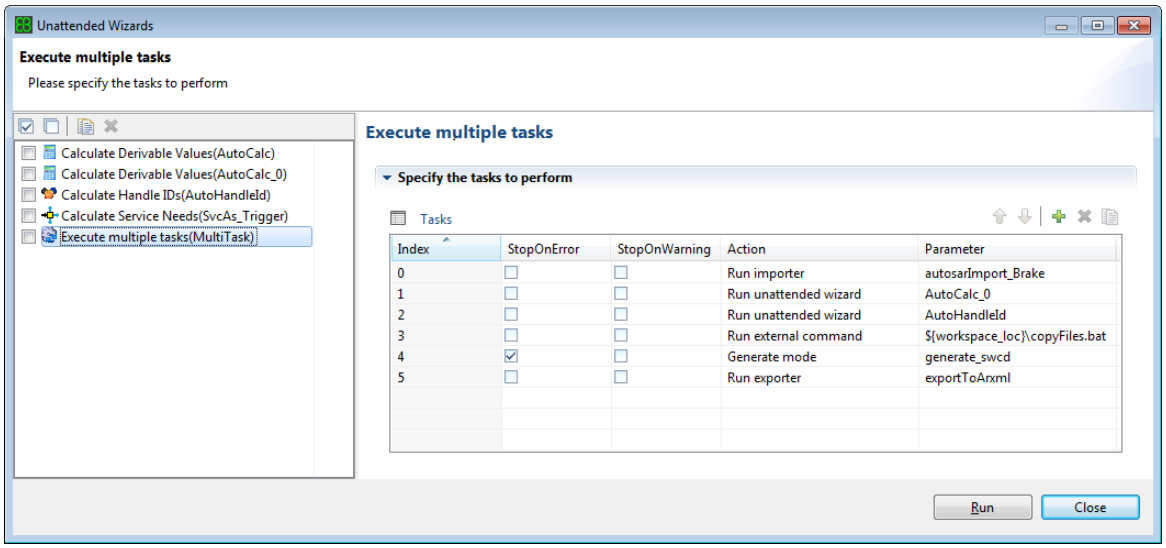


Figure 3.5. The **Execute multiple tasks** wizard

With the **Execute multiple tasks** wizard you can create and arrange a set of actions. The following actions are available:

- Run an unattended wizard
- Run an importer
- Run an exporter
- Run code generation
- Run an external command

The **Execute multiple tasks** wizard consists of a table, where you can add, delete, modify, and re-arrange actions. For each action a parameter has to be selected.

- To run an unattended wizard, select a triggerId from the list of available unattended wizard configurations.
- To run an importer, select the name of an importer which has to be available in the current project.
- To run an exporter, select the name of an exporter which has to be available in the current project.
- To run code generation, it is necessary to select the generation mode from the list.
- To run an external command, insert a command, e.g. to execute a batch file. If you press **CTRL+Space**, a content assist opens where you can choose variables, e.g. the workspace location (${workspace_loc}) to prevent using absolute paths.

With **StopOnError** and **StopOnWarning** it is possible to stop the execution of the whole **Execute multiple tasks** wizard if an action returns with errors and/or warnings. Thus, it is possible to prevent further execution of dependent tasks if an error occurs.

4. Changes for release 15.0

4.1. Path mapping support when importing configuration data from the AUTOSAR format

The import of ECU configuration files is now possible even if the configuration of the module is done outside of EB tresos Studio with a different package location and with any short name. This highly simplifies the data exchange with third-party tools.

Within the importer dialog for the existing **AUTOSAR importer** an additional tab **Path mapping** is shown. This is used to map the AUTOSAR `SHORT-NAME` paths and package structures from the input file to the ones used within the current project. It might be necessary to configure path mappings when:

- The configuration that shall be imported is not created using the same AUTOSAR schema file (`ECUC-MODULE-DEF`). In this case, all `DEFINITION-REF`s defined within the input files are wrong and must be mapped.
- The configuration that shall be imported is contained within a package structure which is different from the structure created by EB tresos Studio. In this case, the package structure must be mapped to the EB tresos Studio internal one.
- The `SHORT-NAME` of the configuration is different from the one that already exists in EB tresos Studio.
- The configuration that shall be imported contains references to another module configuration whose package structure is different from the structure created by EB tresos Studio or whose SHORT-NAME is different from the module configuration that already exists in EB tresos Studio.

EB tresos Studio tries to automatically calculate the path mappings for the current input file. Manual changes are only necessary if the target path cannot be calculated. For example if the input file contains references to another module configuration that is not contained in the file and whose paths need to be mapped.

You can also specify these mappings on the command line for legacy generate and legacy convert.

4.2. AUTOSAR 4.1.1 and 4.1.2 support

EB tresos Studio provides support for the relevant parts of the system model for AUTOSAR releases 4.1.1 and 4.1.2.

Prior releases supported AUTOSAR 4.0.3 which is still fully supported.

4.3. LDF 2.1 and 2.2 support

The EB tresos Studio LDF Importer is now able to process LDF 2.1 and 2.2 configuration files.

4.4. Eclipse Properties view support

EB tresos Studio now features a new Eclipse **Properties** view that you can use to replace the three proprietary **Element Outline** views: **Description**, **Information**, and **Errors**. All information shown in these views is now also available from the new **Properties** view. For further information, refer to the chapter `Properties view` in the EB tresos Studio User's guide.

The **Properties** view replaces the **Element Outline** views in the EB tresos Studio default perspective. The old **Element Outline** views are still avaialable. You can either use the old **Element Outline**views, the new **Properties** view or a combination of both as you prefer. For further instructions on how to do this, refer to section `Opening a view` in the EB tresos Studio User's guide.

5. Changes for release 14.2

5.1. Workflows view project-specific

In previous releases of EB tresos Studio the **Workflows** view displayed one globally stored workflow state. This was problematic when users performed one single workflow for several projects. They did not know anymore where they interrupted their work for the respective project.

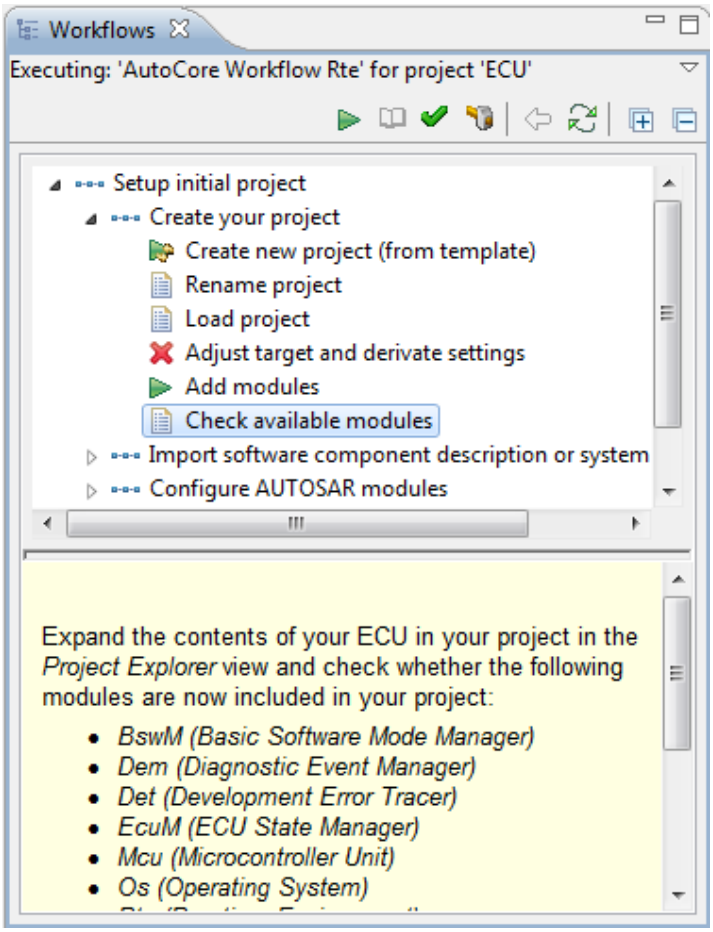
Now the **Workflows** view is project-specific. This means that for each project for which a workflow has been performed, a set of state information is stored. Therefore it is possible to work in parallel on several projects.

The workflow state shown in the workflow view depends on the current selection in the project explorer. The workflow view reacts on project selection changes. As soon as the selection changes in the project explorer, the workflow view displays the state for the selected project. If the selected project has no state information stored for the current workflowId and version, all steps are shown in state **Open** and by default the root step is selected.

If no project is selected, all steps are shown in state **Open** and by default the root step is selected. Users may use the workflow as usual, but no state information can be stored without a selected project.

It is also possible to export/archive a project including the workflow states. When users reimport the project, the state for the currently shown workflow is restored (if the workflow ID and version matches).

The **Workflow** view shows on top of the view which workflow is currently performed for which project.



The workflow description got extended. A new optional attribute **needsproject** marks a step (either a group or an action step) to require a project.

This option shall be used in workflows which start without having a project. In this case the workflow most probably starts with a step to create or import a project. The attribute **needsproject** marks the project specific steps of the workflow. If users have already created a new project before they use the workflow, the workflow view pre-selects the first step with the attribute **needsproject**=true. Users can immediately start to configure their projects then.

5.2. AUTOSAR 4.1.1 compliant post-build time support for Multiple Configuration Containers

Starting with release 14.2, EB tresos Studio implements the improved specification of AUTOSAR 4.1.1 with regard to post-build configuration time support.

The new concept changes the behavior of EB tresos Studio, after the project is switched into post-build configuration time. This especially applies to the semantics of parameter configuration classes and post-build changeable containers within Multiple Configuration Containers (MCCs).

5.2.1. AUTOSAR 4.1.1: New requirements

The following AUTOSAR 4.1.1 requirements and constraints were taken into consideration:

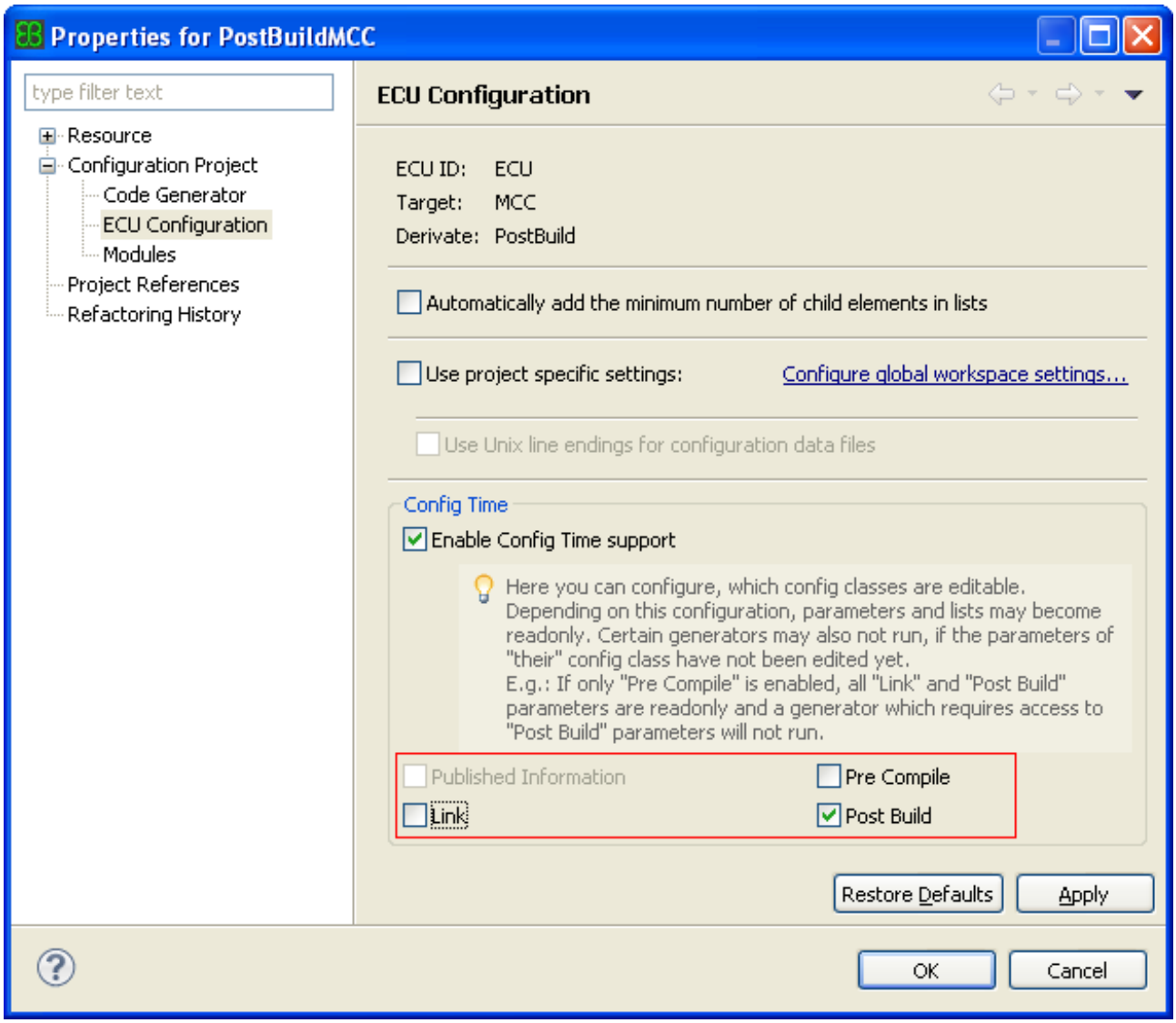
- [TPS_ECUC_02132] The postBuildChangeable attribute shall only be set to true for containers located within a multipleConfigurationContainer
- [TPS_ECUC_08000] Number of instances of a EcucContainerDef may change at post-build time if attribute postBuildChangeable is set to true
- [TPS_ECUC_08001] Configuration class of parameters and references within postBuildChangeable containers
- [TPS_ECUC_08002] Introduction of new EcucParamConfContainerDef instances in a post-build loadable configuration set
- [TPS_ECUC_08003] Usage of postBuildChangeable attribute is independent of aggregated subcontainers
- [TPS_ECUC_08004] Changing of values and multiplicities of EcucParameterValues at post-build time
- [constr_5500] Applicability of postBuildChangeable attribute
- [constr_5501] EcucParameterValues and EcucAbstractReferenceValues in EcucContainerValues that exist in multiple configuration sets
- [constr_5503] symbolicNameValue parameters in post-build configuration sets
- [constr_5504] Removing an instance of the EcucContainerDef in post-build time

5.2.2. New concept: Current Multiple Configuration Container Set

Additionally to the new AUTOSAR requirements, EB tresos Studio 14.2 introduces the concept of the *Current MCC Set*. This allows users to define sets of MCCs of a project in the Generic Configuration Editor of one specific post-build management configuration module. When you select one of those sets as the *Current MCC Set*, each referenced MCC is moved to the top of its MCC list, and thus becomes the default configuration container, which is used eg.

- when you import ECU configuration data with a System Description Importer, or
- when you start Guided Configuration wizards like e.g. **Unattended Wizards** or **Custom Module Editors**, or
- by code generators.

5.2.3. How it works: Theory of operation



Make sure to have the *Current MCC Set* configured in your ECU configuration project as described above.

After you change the **Config Time** of the project to **Post Build** only, the Configuration Verifier of EB tresos Studio applies additional checks according to the new AUTOSAR requirements. This can make new configuration problems appear in the **Problems View**, e.g. if parameters from different config sets of one module with configuration class **Pre Compile** or **Link Time** each have different values.

Another consequence is that during **Post Build** EB tresos Studio will not allow you to change or delete existing ECU configuration data any more, if the data is referenced by pre-build nodes (i.e. configuration classes **Pre Compile** or **Link**). This applies to containers, which have the schema attribute `postBuildChangeable` set to `true` as well as the contained pre-build configuration parameters.

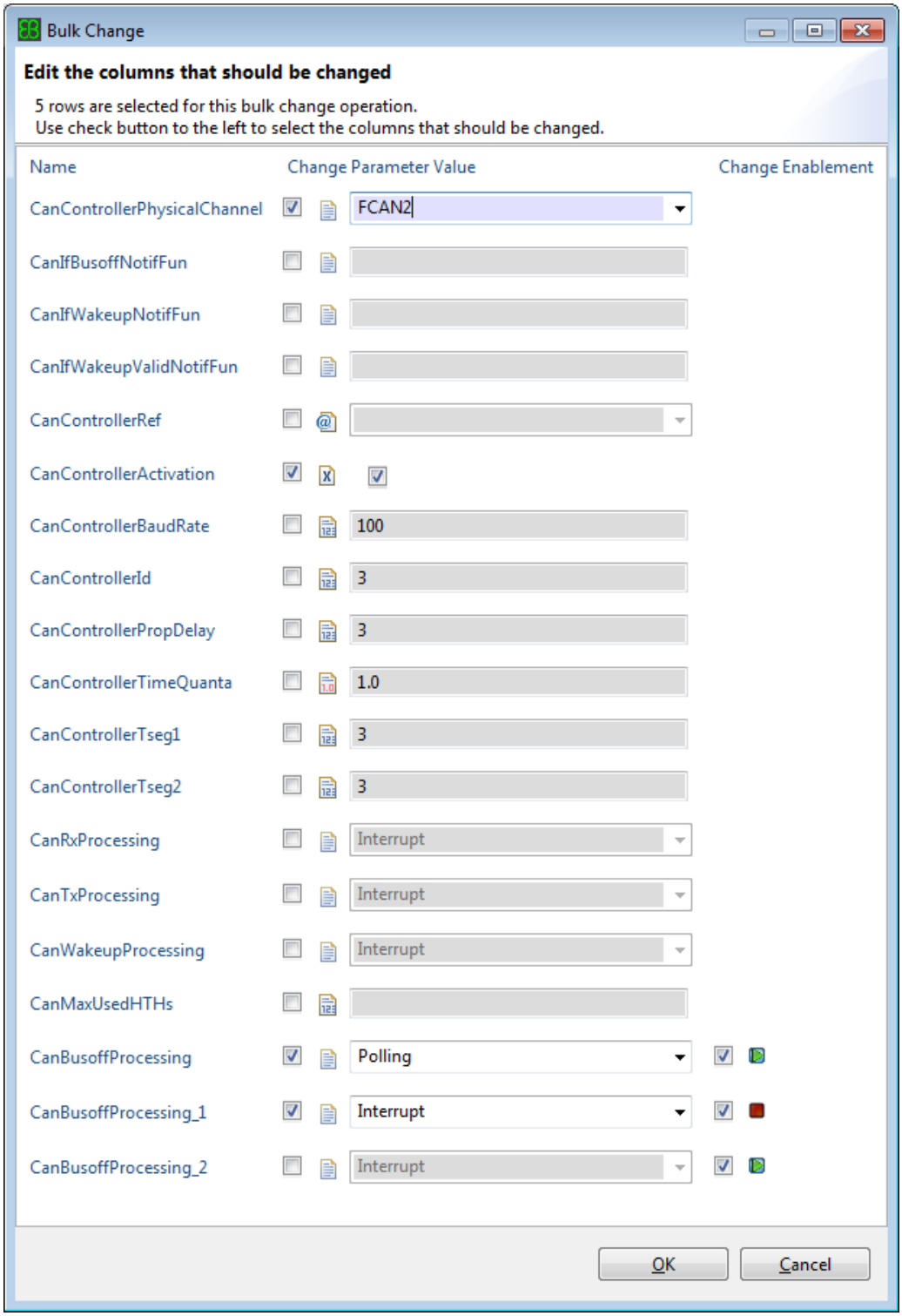
You may however still configure and delete newly created instances of post-build changeable containers and any contained parameters, even if those have a configuration class earlier than **Post Build** (i.e. **Pre Compile** or **Link**).

This way, EB tresos Studio prevents changes and deletions of configuration data which was likely taken into account when generating and compiling/linking the software for the ECU, which yet may require to be finally configured in a post-build scenario.

5.3. Bulk Change Editor enhancements

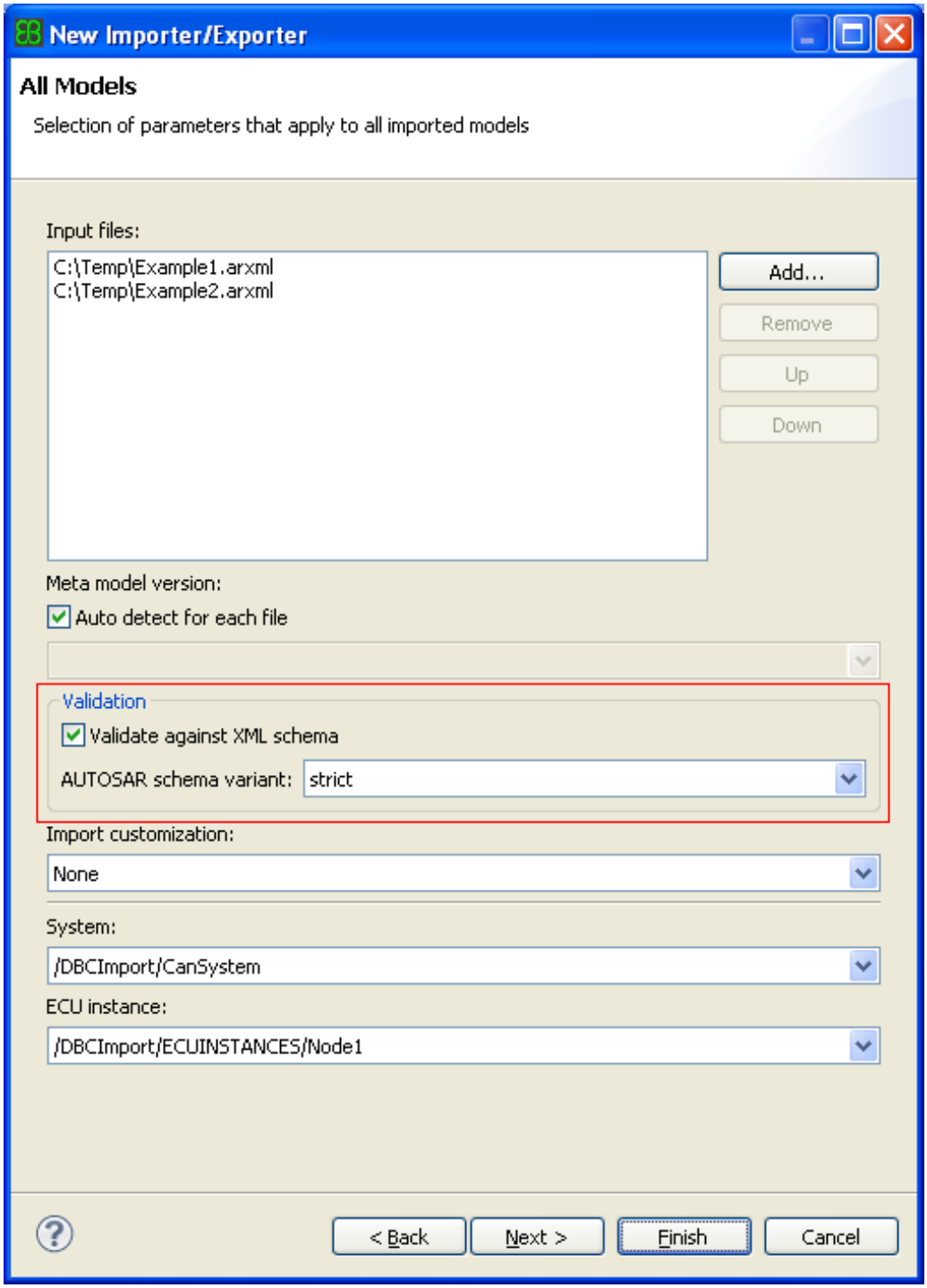
In previous releases of EB tresos Studio, the Bulk Change was possible only for non-optional configuration parameters, that were not in a subcontainer (list in list) and that were visible as column in the corresponding table. Now, Bulk Change also supports enabling and disabling of optional parameters. The Bulk Change of all parameters in the container including optional parameters is possible. The parameters inside optional containers are also visible in the Bulk Change Editor, however their modification is possible only if the parent container is enabled.

The changes in the Bulk Change Editor are also reflected in the GUI. The new GUI contains three sections, as visible in the image that follows. Firstly, there is the **Name** of the parameter. This is followed by **Change Parameter Value**, where the check box allows editing the value of the parameter. In **Change Enablement** section, the check box allows users to edit the optional state. These operations (changing the value or the enablement) can be done independently of each other.


Bulk Change
Edit the columns that should be changed
5 rows are selected for this bulk change operation.
Use check button to the left to select the columns that should be changed.
Name
Change Parameter Value
Change Enablement
CanControllerPhysicalChannel
FCAN2
CanIfBusoffNotifFun
CanIfWakeupNotifFun
CanIfWakeupValidNotifFun
CanControllerRef
CanControllerActivation
CanControllerBaudRate
100
CanControllerId
3
CanControllerPropDelay
3
CanControllerTimeQuanta
1.0
CanControllerTseg1
3
CanControllerTseg2
3
CanRxProcessing
Interrupt
CanTxProcessing
Interrupt
CanWakeupProcessing
Interrupt
CanMaxUsedHTHs
CanBusoffProcessing
Polling
CanBusoffProcessing_1
Interrupt
CanBusoffProcessing_2
Interrupt
OK
Cancel

5.4. Selectable AUTOSAR XML schema variant in System Description Importer

For AUTOSAR releases that come with different XML schema variants (e.g. a **standard** and a **strict** variant for AUTOSAR 4), you may now choose the schema variant that EB tresos Studio will use for validating the ARXML import files in the **System Description Importer**.



You may also choose the validation variant when you work with the EB tresos Studio command line by setting the system property `-DvalidationVariant=(strict|standard)`.

6. Changes for release 14.1

6.1. Performance improvements

The performance of EB tresos Studio has been improved in several ways:

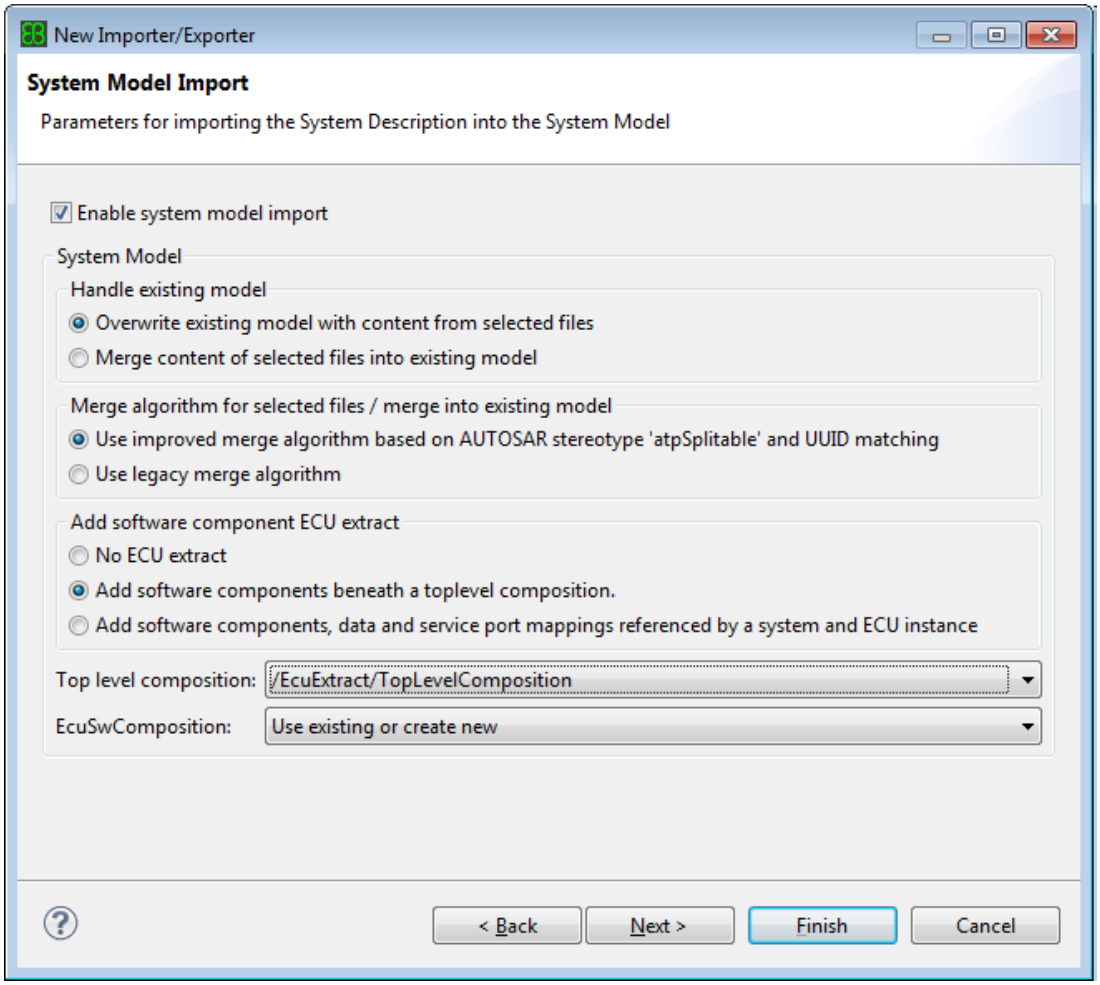
- ECU configuration nodes are now verified in parallel.
- In previous EB tresos Studio versions, the first importer/exporter has been selected by default when you opened the importer and exporter dialog. For importers with a huge amount of input files, the parsing of files takes up a lot of time. When you now open the importer and exporter dialog no entry is preselected.
- In the System Description Importer, the loading and parsing of files is now also speed up as the files are loaded in parallel.
- Additionally, in the System Description Importer, the performance has been improved by caching the already parsed files. In previous EB tresos Studio versions, the complete list of files was parsed again when you added or removed one file to or from the list. Now, only those files are parsed which are newly added or have been changed on disk.

6.2. System Description Importer improvements

Additional support to utilize the <atpSplitable> stereotype of aggregations during model merge has been added. This enhances the abilities of the model merger and allows you to merge more complex models.

6.2.1. New merge algorithm using AUTOSAR stereotype <atpSplitable>

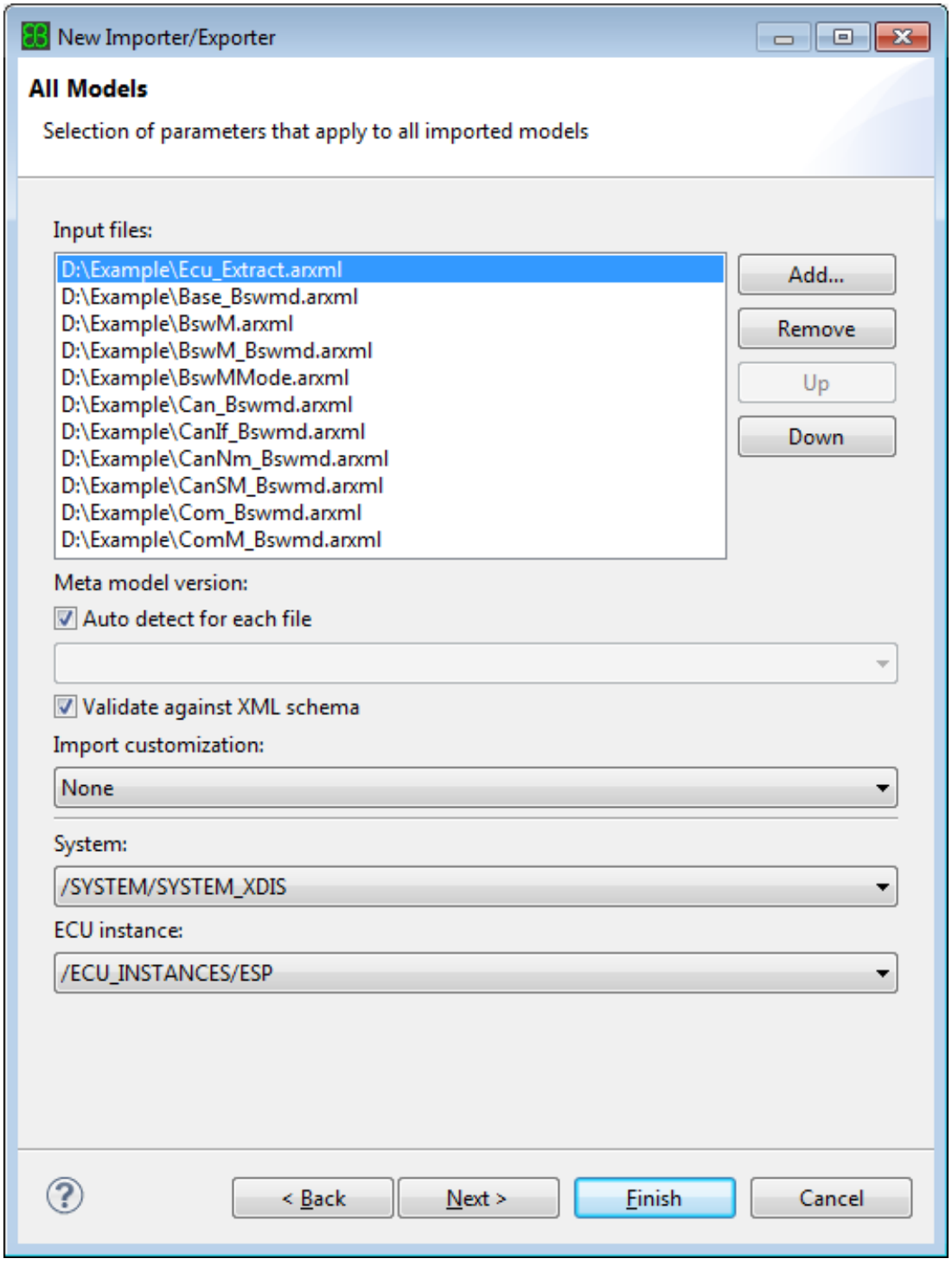
The AUTOSAR meta model stereotypes <splitable> (3.x) / <atpSplitable> (4.x) define which elements can be split up over several physical files. The new merge algorithm is based on this concept. In addition, the new merge algorithm uses UUID matching to find e.g. renamed elements.



To use the new merge algorithm, select **Use improved merge algorithm based on AUTOSAR stereotype 'atpSplitable' and UUID matching** in the System Model Import tab of the system description importer.

6.2.2. Specify import order of input files

In previous EB tresos Studio releases, it was only possible to adapt the order of the input files when you changed the file names. Now, two new buttons exist in the system description importer GUI to adjust the import order of the files. It is necessary to adjust the import order of the files if one system model element is available in several files. In this case you need to specify which file shall get the highest priority. The uppermost file inside the list has the highest priority.



With the **Up** and **Down** buttons on the right side of the **Input Files** text box, you can change the order of the list of files to be loaded.

6.3. Fibex 3.x files can be imported in 64 bit mode

The support for importing Fibex files in the 64 bit mode of EB tresos Studio has been added. Due to heap memory restrictions, the import of very large Fibex files (>20 MB) may fail in 32 bit mode. The 64 bit mode supports the import of considerably larger Fibex files.

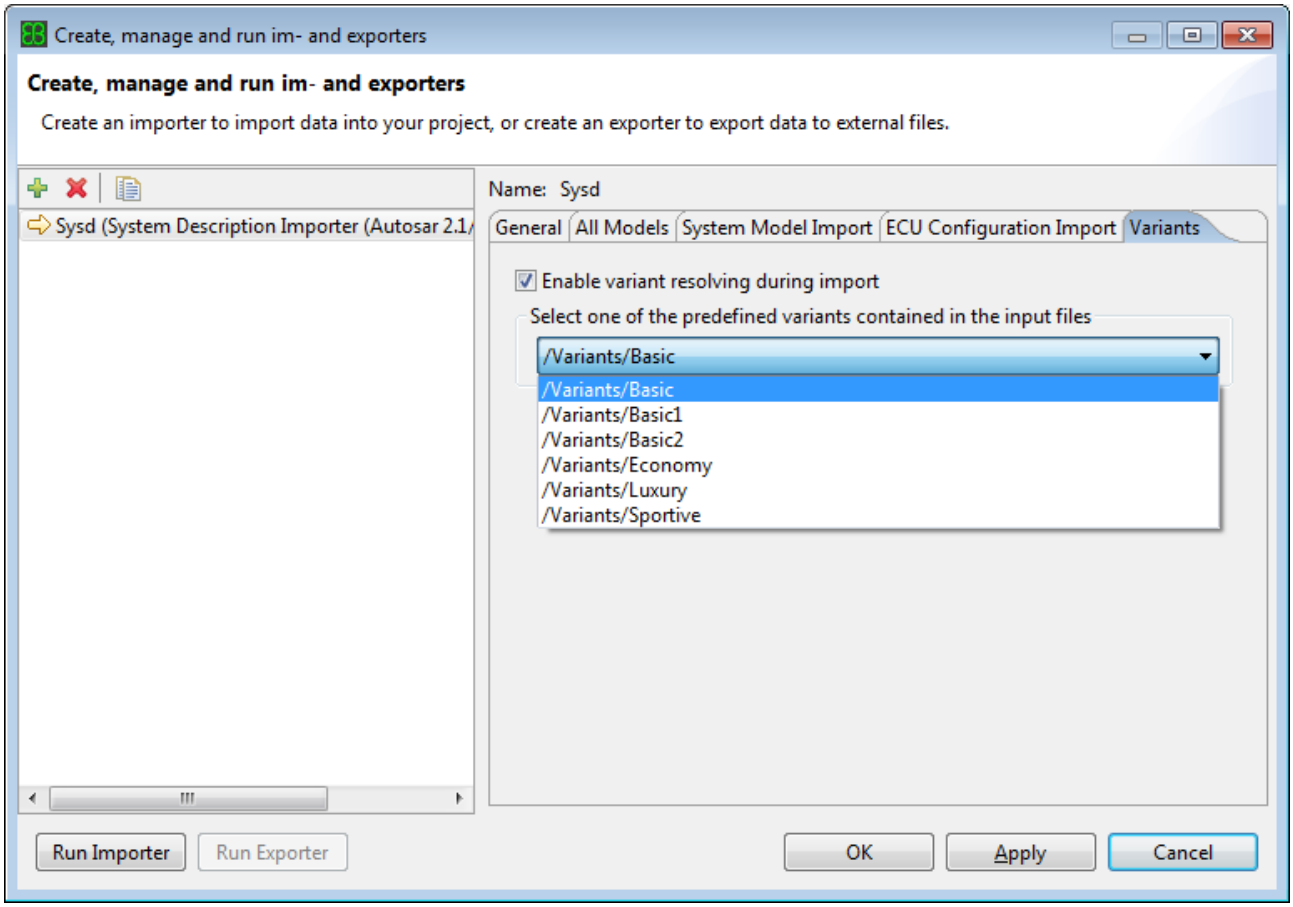
6.4. Support for importing Fibex 1.x and 2.0.x files has been removed

The importer support for the Fibex formats up to 2.0.x which became obsolete in the meantime has been removed. As a migration path, the EB tresos Studio releases 14.0 and earlier allow to convert such Fibex files to AUTOSAR system description files, which in turn can be used in EB tresos Studio releases 14.1 and newer. Please refer to the chapter "Commands for converting and merging files" in the EB tresos Studio user's guide for a description how you can convert Fibex files.

6.5. Variant handling

EB tresos Studio 14.1 introduces support for variant handling in AUTOSAR system description files. This allows to provide data for multiple variants of a system within one file, instead of providing multiple files with largely equal content. OEMs use this technique to represent product lines in AUTOSAR.

EB tresos Studio does not store the defined variants internally. Instead, you must select one of the defined variants upon import and all other variants are removed (AUTOSAR calls this variant resolving). This can be enabled on the variants tab in the System Description Importer. After variant resolving has been enabled, a list of predefined variants contained in the selected files is provided. You have to select one of those variants which will be resolved during import.



You can also enable variant resolving on the `tresos_cmd.bat` legacy command line by setting the system property `predefinedVariant` to the AUTOSAR path of the variant.

E.g. `tresos_cmd.bat -DpredefinedVariant="/Variants/Basic" legacy convert system.arxml@sysd resolved.arxml@sysd:4.0.3`
You can use this property with the following commands:

- `legacy convert`
- `legacy generate`
- `legacy make`
- `legacy verify`

7. Changes for release 14.0

7.1. Link support in the Details dialog of Error Log view and Problems view

The Details dialog of Error Log view and Problems view may provide links. The link will navigate you directly to the point of interest, currently to the node in the configuration editor.

7.2. Eclipse platform updated to version 3.8

The platform on which EB tresos Studio is developed has changed with this release. EB tresos Studio is now based on the Eclipse 3.8.0 platform.

7.3. EB tresos Studio 64 bit support

In 32 bit mode, the amount of useable memory is restricted and therefore in huge projects, this may lead to OutOfMemory errors even if there is physically enough memory available.

When running EB tresos Studio as a 64 bit application, this memory restriction does not exists. One precondition is, to use a 64 bit operating system - otherwise 64 bit applications cannot be executed at all. To run EB tresos Studio as a 64 bit application, you just have to use the `bin/tresos_gui_64.exe` for gui mode or the `bin/tresos_cmd_64.bat` for commandline mode.

7.4. Labels and icons for generator modes

EB tresos Studio can now show the menu items for specific generator modes using translateable labels and specific icons, instead of of just showing the symbolic name of the generator mode. For details, refer to the description of the new `modeMenuItem` element of the `dreisoft.tresos.generator.api.plugin.generator` extension point.

8. Changes for release 13.1

8.1. System description importer now can import files with different meta model versions

8.1.1. System description import behaviour in previous versions

In EB tresos Studio 11 it was possible to import files with different meta model versions (e.g. 3.1.2 and 3.1.4) in one single importer. The user had to select the meta model version with which the import should run. Then each file was parsed with the selected version and only the tags known by the selected meta model version are imported. Therefore the result differs according to the selected meta model version.

In EB tresos Studio 12 it was not possible to mix up files with different meta model versions.

The possibility to use one importer for each meta model version only worked for files without references between each other. Those open references get lost when importing the files in several importers.

8.1.2. New feature in EB tresos Studio 13.1

Now it is possible to import files with different meta model versions within one single importer. A new feature has been introduced. By default the system description importer will autodetect the meta model version of each file.

But the mixture is restricted by the meta model versions of the files and the System model version of the project.

- All files with meta model version up to 3.1 can be mixed and can be imported in System model (Autosar 3.1 and older)
- All files with meta model version 3.2.x can be mixed and can be imported in System model (Autosar 3.2 and newer)
- All files with meta model version 4.x can be mixed and can be imported in System model (Autosar 3.2 and newer)

When selecting, e.g. 3.1.2 and 3.1.4 files as input, then each file is parsed and transformed to a common version basis. So no data gets lost during the import.

If the meta model version cannot be autodetected, then one meta model version has to be selected from a drop-down list box. Then the importer works as known from EB tresos Studio version 11.

8.2. AUTOSAR 3.2.2 support

EB tresos Studio now supports AUTOSAR 3.2.2.

Prior releases supported AUTOSAR 3.2.1. Since AUTOSAR 3.2.1 is backward compatible, it is still possible to import AUTOSAR 3.2.1 files. When writing or exporting AUTOSAR 3.2 files, they will always be written as 3.2.2 files.

8.3. AUTOSAR 4.1.0 support

EB tresos Studio provides support for the Ethernet and TCP/IP relevant parts of the system model as of AUTOSAR 4.1.0 (snapshot from 2012-08-09, XSD document version 4.3.0, revision 0).

Prior releases supported AUTOSAR 4.0.3 which is still fully supported.

8.4. Subpackages support

It is now possible to define modules within any sub-package structure. This is e.g. required by AUTOSAR, when a vendor defines multiple complex device driver modules. For backwards-compatibility reasons, the usage of sub-packages for ecu module configurations is still not allowed.

9. Changes for release 13.0

9.1. AUTOSAR 3.2 support

The support for AUTOSAR revision 3.2.1 in EB tresos Studio includes the following features:

- You may now import AUTOSAR 3.2.1 system descriptions as ARXML files. EB tresos Studio provides the new content type `sysd:3.2.1` for that.

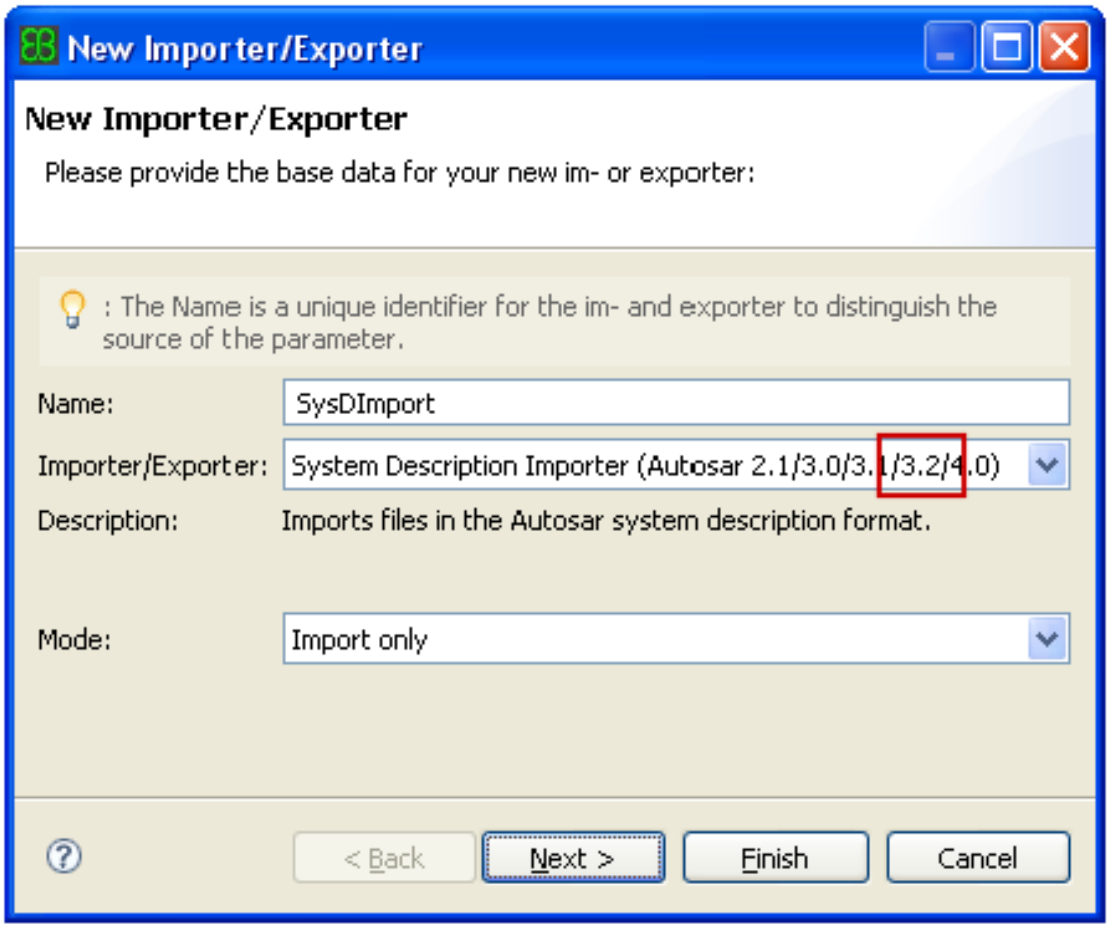


Figure 9.1. **System Description Importer** support for AUTOSAR 3.2

However, exporting into system description ARXML files is not supported.

- You may now import and export AUTOSAR 3.2.1 ECU configuration data from/into ARXML files. EB tresos Studio provides the new content type `asc:3.2` for that.

Figure 9.2. New content type **asc:3.2**

- The maximum `SHORT-NAME` length for AUTOSAR 3.2 projects is restricted to 128 characters, which conforms to the AUTOSAR specification. However, you may still disable the length check for AUTOSAR versions prior to 3.2.

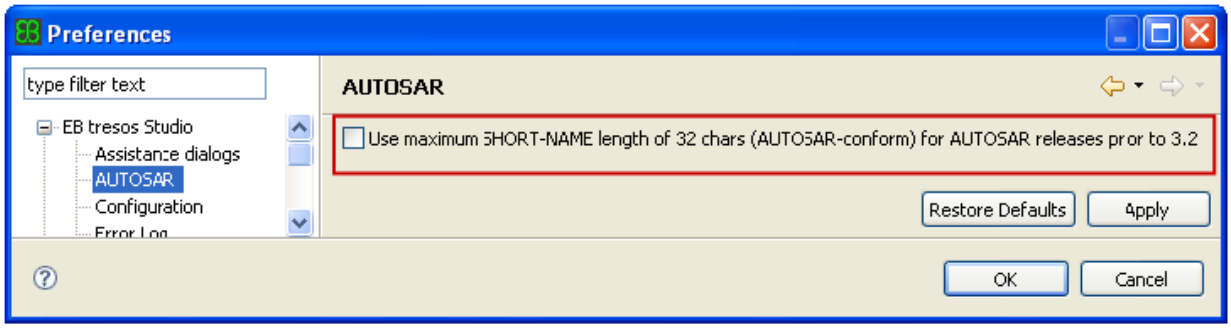


Figure 9.3. Preference for `SHORT-NAME` length handling in EB tresos Studio 13.0

- EB tresos Studio now provides a new VSMD rule set called `asc:3.2` for validating vendor-specific module definitions against the AUTOSAR 3.2.1 standard using the command line mode of EB tresos Studio.
- The AUTOSAR 3.2.1 standard module definitions are installed in the subdirectory `autosar\3.2` of your EB tresos Studio installation.

9.2. Configuration class support

AUTOSAR defines several stages of development, namely `PreCompile`, `Link`, and `PostBuild`. The parameter definition can define *when* a specific parameter must be set. This configuration time depends on the configuration variant selected for the module.

The configuration class support affects the EB tresos Studio user interface, the code generators, the API, the importers, and the exporters.

The configuration class support works as follows: EB tresos Studio allows to restrict the modifiability of parameters on configuration class basis in a configuration project. For most use cases, only the configuration class matching the current phase of development is editable, if you restrict parameter edits via configuration classes.

Each configuration time then requires different code generators, so the parameters can be processed from configuration time to configuration time. This is a new restriction to code generators: The code generators must not access parameters with later configuration classes, because these parameters might change their value in the future and the generator will not necessarily get executed another time. All existing generators that do not specify a configuration time are treated as `PostBuild` generators.

The API that is used to access the DataModel restricts the access to the parameters according to the configuration class settings of the configuration project. Write access is only permitted to parameters with an editable configuration class.

The AUTOSAR Im-/Exporter also supports configuration classes: Importers can now be limited to specific configuration classes. The importers also check whether parameters of a not editable configuration class shall be overwritten during the import. Furthermore, the importers now support merging the configurations or replacing the existing configuration. Exporters can now also be limited to specific configuration classes.

10. Changes for release 12.1

10.1. Team collaboration improvements

EB tresos Studio 12.1 provides the following improvements that help you work together on a project as a team:

10.1.1. The Compare editor for XDM files supports comparing and merging attributes now

In the previous version of EB tresos Studio, the **Compare** editor for XDM files did not consider differences in node attributes except for changes of the `ENABLE` parameter value. Now, the **Compare** editor for XDM files supports comparing and merging differences in arbitrary XDM attributes. Node attributes are displayed as the first children of the node.

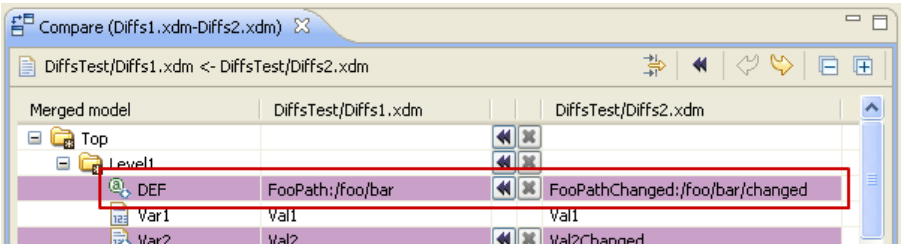


Figure 10.1. **Compare** editor for XDM files showing attribute differences

10.1.2. Automatic reload

In the previous version of EB tresos Studio, you had to manually trigger a reload of a loaded project whenever one of the relevant files had changed. This applied e.g. when you had updated any XDM files that store the module configurations via your version control system (e.g. CVS or SVN).

Now, EB tresos Studio detects changes on those files and triggers the reload automatically.

10.1.3. Warnings for ignored nodes

In previous versions of EB tresos Studio, the **Compare** editor for XDM files silently ignored nodes in the new file that did not match the structure of the original file during comparison.

Now it shows a warning dialog that lists all ignored nodes.

10.2. Generic Configuration Editor improvements

EB tresos Studio 12.1 provides the following improvements of the Generic Configuration Editor:

10.2.1. The user assistance for references has been extended

For references that do not have any referenceable nodes, the reference combo box stays empty, which means that you cannot select a reference. Now, EB tresos Studio provides the new context menu entry **List referenceable nodes**. To open the context menu with this new entry, click the type icon in front of the combo box:

Figure 10.2. The **List referenceable nodes** context menu entry

If you select the new entry **List referenceable nodes** in the context menu, EB tresos Studio performs the following actions:

- If there is exactly one referenceable node, EB tresos Studio jumps directly to this node.
- If there are more than one referenceable nodes, EB tresos Studio lists all referenceable nodes in the **Search** view. You may then jump to each node in the **Search** view either by double-clicking an entry of this list or by using the **Search** view context menu entry **show in editor**:

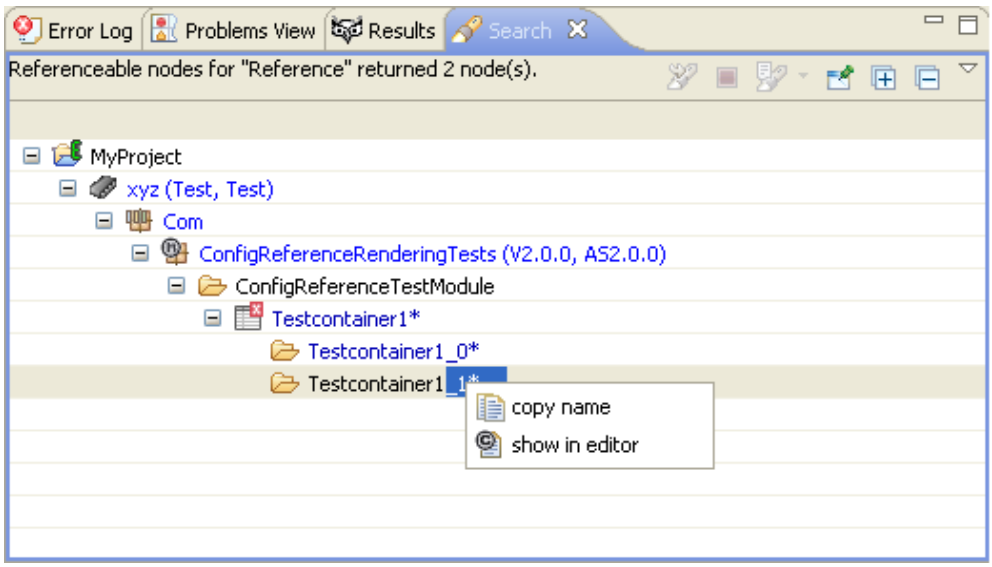


Figure 10.3. The **Search** view showing the list of referenceable nodes

After you have selected a value in the reference combo box from the list in the **Search** view, you can jump to this referenced node by using the context menu entry **Go to referenced node**.

NOTE

The context menu entries are not available for foreign references and for instance references

The two context menu entries **List referenceable nodes** and **Go to referenced node** are not available for foreign references and instance references.

10.2.2. Choices

The rendering of choices has been revised under the following aspects:

- The visualization of nested choices has been improved.
- The GUI is not crowded with inactive parameter configurations anymore, which are within subcontainers of choices that are currently not selected.
- The performance has been improved.
- Less GUI OS resources are used so that EB tresos Studio does not run out of GUI handles as soon anymore.

This was achieved by rendering each subcontainer of a choice element into a separate tab. To distinguish the currently selected choice, its tab title is rendered in bold font. You can still look at the disabled configuration parameters in the subcontainers of choices, which are currently not selected, by clicking on their tab title.

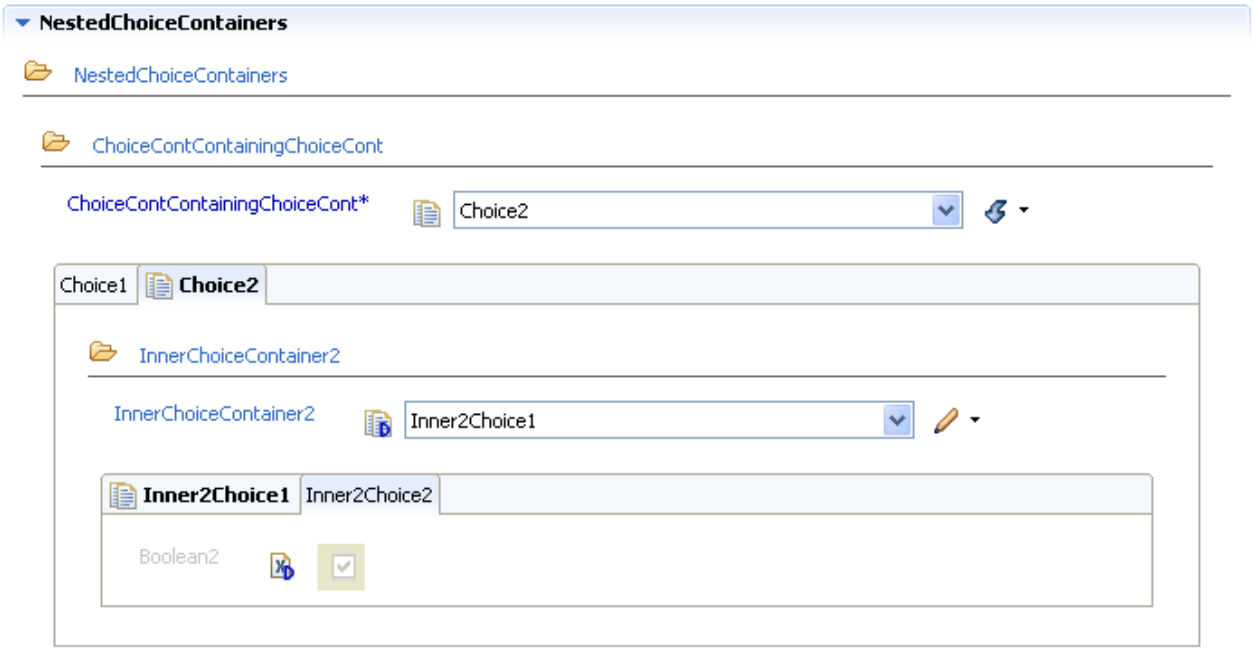


Figure 10.4. Choices displayed in tabs

10.3. Autosar 4.0.3 support

EB tresos Studio 12.1 now provides support for AUTOSAR 4.0.3 (xsd document version 4.2.1, revision 3) in parallel to the AUTOSAR 4.0.2. support which has been introduced with EB tresos Studio 12.0.

Please refer to the New and Noteworthy document of EB tresos Studio release 12.0 for a short description of the most important new features introduced with AUTOSAR 4.0 support.

It is now possible to create AUTOSAR 4.0.3 compliant modules, configure them within EB tresos Studio and import and export AUTOSAR 4.0.3 files.

11. Changes for release 12.0

11.1. AUTOSAR 4.0 Support

Support has been implemented for AUTOSAR 4.0.2. This chapter describes the implemented features, changes to EB tresos Studio and the limitations of the features.

11.1.1. Software Component Template

EB tresos Studio now supports the AUTOSAR 4.0 Software Component Template for projects with AUTOSAR 4.0 compliant modules and mixed projects.

11.1.2. ECU Configuration

EB tresos Studio now supports ECU Configuration Description and Parameter Definition according to the AUTOSAR 4.0 release (Specification of ECU Configuration 3.1.0 Rev 2).

It is now possible to create AUTOSAR 4.0 compliant modules and configure them within EB tresos Studio. ECU Configuration Descriptions using the AUTOSAR 4.0 file format can now be imported and exported from EB tresos Studio

The AUTOSAR features *variant handling*, *documentation support* and *calculation formulas* are currently not supported by EB tresos Studio.

11.1.3. New Project Wizard

The **New Project Wizard** has been changed to reflect the presence of the AUTOSAR 4.0 Software Component Template. Additionally, the **Release Version** drop-down list box has been moved to the first page since it affects the selection of the enabled models.

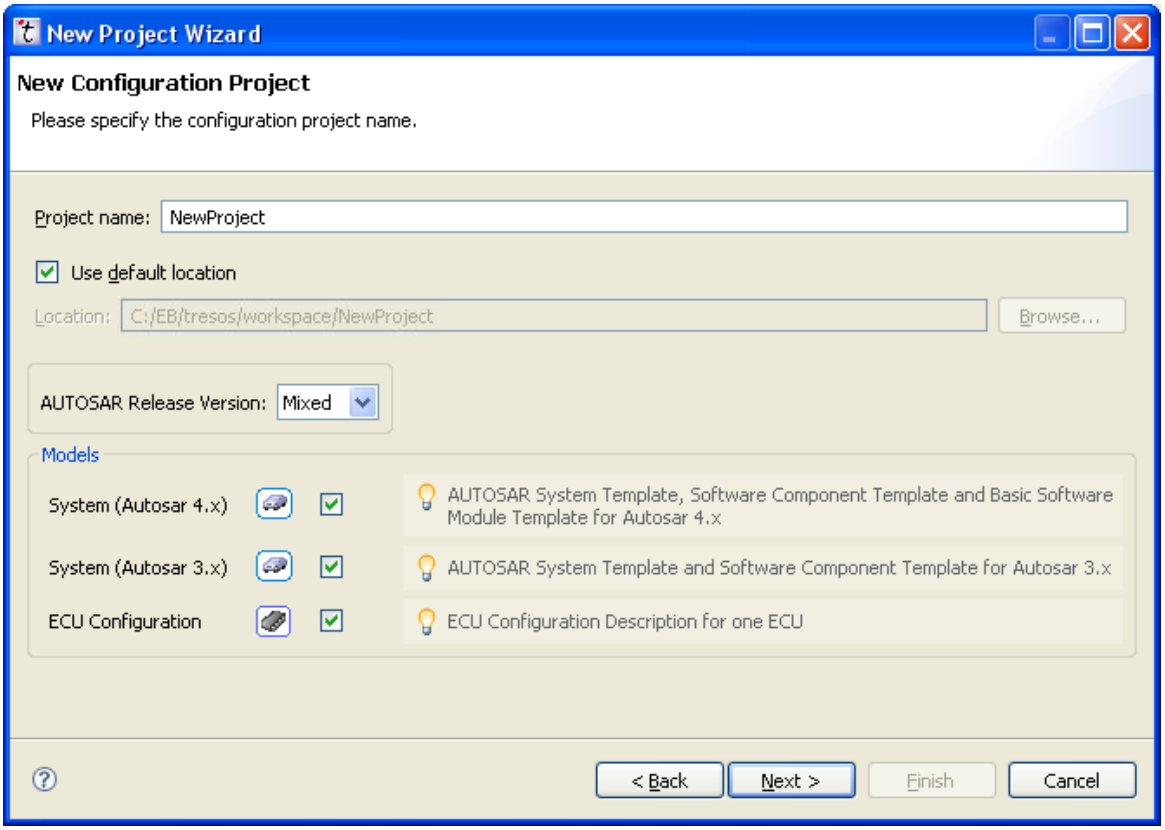


Figure 11.1. The **New Project Wizard**

Choosing **Mixed** as release version enables **System (Autosar 4.x)**, which provides the newly available Software Component Template 4.0.

11.1.4. System Description Importer AUTOSAR 4.0

EB tresos Studio is now able to import Software Component Templates via the System Description Importer.

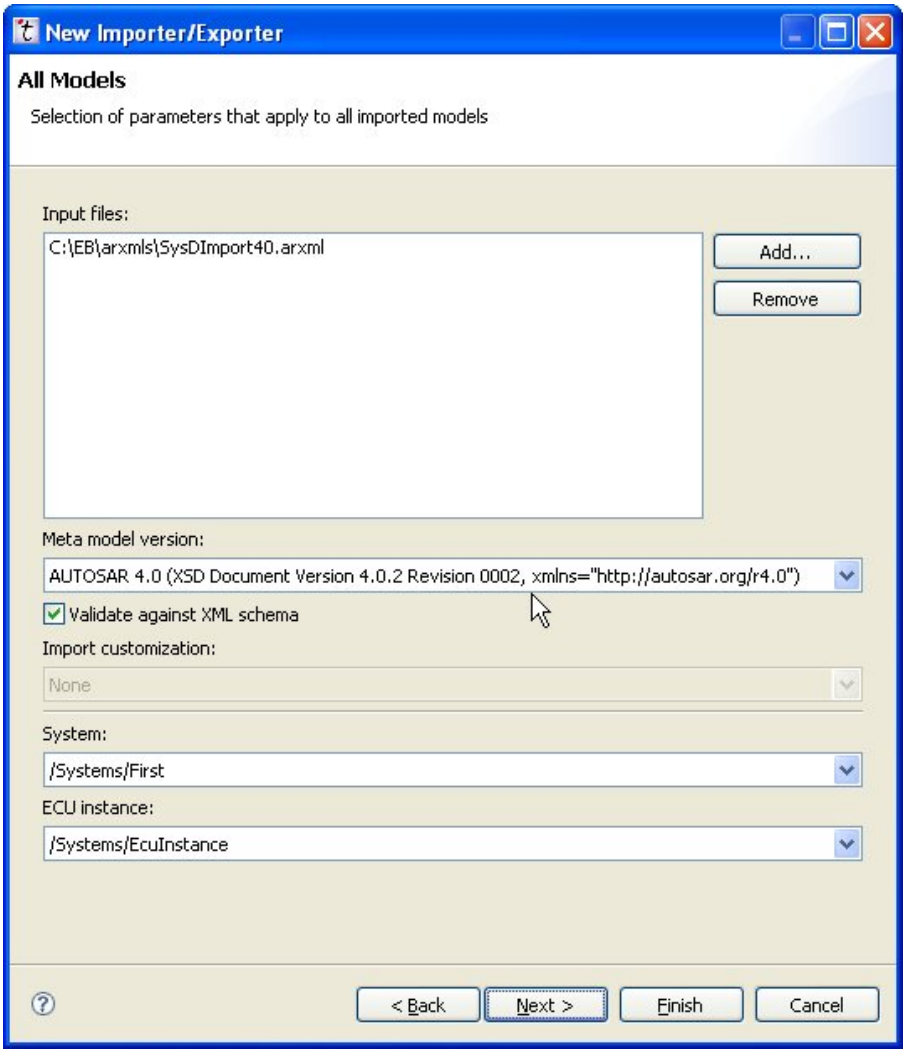


Figure 11.2. Importing a 4.0 Software Component Template

If the imported file contains AUTOSAR 4.0 documentation support or a system description template, this information is ignored.

Importing a file that contains AUTOSAR 4.0 variant handling or formula language tags completely fails because these features are not supported yet.

11.1.5. New XML tags

The AUTOSAR 4.0 format redefined most of the XML tags in XDM files. An AUTOSAR ECU configuration now looks as follows:

```
<AUTOSAR ...>
```

```
  <AR-PACKAGES>  <!-- Was TOP-LEVEL-PACKAGES -->
    <AR-PACKAGE>
      <SHORT-NAME>AUTOSAR</SHORT-NAME>
        <ELEMENTS>
          <ECUC-MODULE-DEF> <!-- Was MODULE-DEF -->
            <SHORT-NAME>Adc</SHORT-NAME>
             <CONTAINERS>
      <!-- Was PARAM-CONF-CONTAINER-DEF -->
               <ECUC-PARAM-CONF-CONTAINER-DEF>
                  <SHORT-NAME>AdcConfigSet</SHORT-NAME>
```

EB tresos Studio handles these tag renaming transparently. For existing node types, XDM files look as before:

```
<d:ctr type="AUTOSAR" factory="autosar"  ... >
  <d:lst type="TOP-LEVEL-PACKAGES">
    <d:ctr name="AUTOSAR" type="AR-PACKAGE">
        <d:lst type="ELEMENTS">
          <d:chc name="Adc" type="AR-ELEMENT" value="MODULE-DEF">
            <v:ctr type="MODULE-DEF">
               <v:lst name="AdcConfigSet"
      type="MULTIPLE-CONFIGURATION-CONTAINER">
                  <v:ctr name="AdcConfigSet"
            type="MULTIPLE-CONFIGURATION-CONTAINER">
```

11.1.6. The Standard Module Definition (StMD) is placed into a new subpackage

The Standard Module Definition (StMD) is now placed into a subpackage called `EcucDefs` of the AUTOSAR top-level package `/AUTOSAR/EcucDefs`. Vendor-specific module definitions (VSMDs) may not be placed into subpackages. Nevertheless EB tresos Studio is able to handle a StMD in a subpackage. The StMD is represented in the DataModel within the following structure:

```
<datamodel>
  <d:ctr type="AUTOSAR" factory="autosar" >
    <d:lst type="TOP-LEVEL-PACKAGES">
      <d:ctr name="AUTOSAR" type="AR-PACKAGE">
        <d:lst type="TOP-LEVEL-PACKAGES">
          <d:ctr name="EcucDefs" type="AR-PACKAGE">
            <d:lst type="ELEMENTS">
              <d:chc name="Adc" type="AR-ELEMENT" value="MODULE-DEF">
                 <v:ctr type="MODULE-DEF">
```

The new path structure of the STMD has the following two modules in EB tresos Studio:

- VSMD reference to the StMD:

```
<d:ref type="REFINED_MODULE_DEF" value="ASPath:/AUTOSAR/EcucDefs/Adc"/>
```

- Reference target definitions:

```
<v:ref name="OsAppScheduleTableRef" type="REFERENCE">
 <a:da name="REF"
       value="ASPathDataOfSchema:/AUTOSAR/EcucDefs/Adc/AdcConfigSet"/>
</v:ref>
```

The `vsmdcheck` command line command now checks whether all `DESTINATION-REF`s taken over from the StMD start with `/AUTOSAR/EcucDefs`.

11.1.7. Non-existing standard parameters

Parameters may not be dropped anymore when you derive a VSMD from a StMD. Instead, the `UPPER-MULTIPLICITY` must be set to `0`: `<UPPER-MULTIPLICITY>0</UPPER-MULTIPLICITY>`. The XDM format marks such parameters with a new attribute:

```
<v:var name="AdcClockSource" type="STRING">
  <a:a name="EXISTING" value="false"/>
</v:var>
```

Non-existing parameters do not appear in the GUI and do not produce a configuration parameter in the XDM.

The `vsmdcheck` command line command also checks this and reports an error if you omit a parameter.

11.1.8. Relative paths

AUTOSAR 4.0 allows to use relative paths. Relative paths do not start with a slash, e.g.: `AdcConfigSet`. Relative paths are resolved relative to a <REFERENCE-BASE> found upwards in the tree:

```
<AUTOSAR>
```

```
<AR-PACKAGES>
  <AR-PACKAGE>
    <SHORT-NAME>TopLevelPackage</SHORT-NAME>
    <REFERENCE-BASES>
      <REFERENCE-BASE>
        <SHORT-LABEL>base1</SHORT-LABEL>
        <IS-DEFAULT>true</IS-DEFAULT>
        <BASE-IS-THIS-PACKAGE>true</BASE-IS-THIS-PACKAGE>
        <PACKAGE-REF
      DEST="ECUC-MODULE-DEF">/AUTOSAR/EcucDefs/Adc</PACKAGE-REF>
      </REFERENCE-BASE>
    </REFERENCE-BASES>
```

EB tresos Studio resolves relative paths during parsing of AUTOSAR files and drops the `REFERENCE-BASE` information.

11.1.9. `UPPER-MULTIPLICITY-INFINITE` tag

The xml tag `UPPER-MULTIPLICITY` is now of type numerical and may not have the value `*` anymore. Instead, the tag `UPPER-MULTIPLICITY-INFINITE` shall be used which is of type Boolean. Using both tags (`UPPER-MULTIPLICITY` and `UPPER-MULTIPLICITY-INFINITE`) for one parameter is prohibited.

```
<UPPER-MULTIPLICITY-INFINITE>true</UPPER-MULTIPLICITY-INFINITE>
```

For the DataModel nothing has changed. `UPPER-MULTIPLICITY-INFINITE` is represented by omitting the `MAX` attribute.

11.1.10. Numbers and Boolean values

AUTOSAR now allows that integer values can be expressed in hexadecimal, octal, binary, and decimal format:

```
<VALUE>0x1f3b</VALUE>
 <VALUE>0b1011</VALUE>
 <VALUE>0735</VALUE> <!-- octal values start with '0' -->
```

Negative numbers can only be expressed in decimal format. The negative numbers are converted into the decimal format when they are imported into the DataModel. The DataModel does not store the number representation.

Float parameters have now some special values:

- Float parameters now also accept `NaN`, `INF` and `-INF` symbols to express the special values *Not a Number (NaN)*, *positive infinity (INF)* and *negative infinity (-INF)*.
- The DataModel and XDM files store float parameters as the values `NaN`, `Infinity`, `-Infinity`.
- The GUI allows to enter `NaN`, `Infinity`, `-Infinity`.

NOTE **String and numerical representation of float parameters**

`NaN`, `-INF`, `INF` must be treated in code templates as if using Java Public API. Both Java and XPath support these three values for numerical values (in case of XPath even for Boolean). The string representations for the values are: `NaN`, `Infinity`, `-Infinity`.

Boolean numbers are represented as `0`, `1` (number values) in AUTOSAR configurations. The DataModel still uses `true` and `false`.

11.1.11. `INVALID`, `WARNING` (and other) attributes with multiple expressions

AUTOSAR 4.0 introduces new restriction tags for strings (`MIN-LENGTH`, `MAX-LENGTH`, `REGULAR-EXPRESSION`) that together with `MIN` and `MAX` must be convertible from and to the AUTOSAR format even if there are XPath restrictions defined for the parameter. The DataModel therefore now allows to define multiple test expressions that are evaluated by different query languages in one attribute:

```
<a:a name="INVALID" type="Multi">

 <!-- String length -->
 <mt:length>
   <mt:tst expr="&gt;=6"/>
   <mt:tst expr="&lt;10"/>
 </mt:length>

 <!-- defining the AUTOSAR MIN/MAX range for a parameter -->
 <mt:range>
   <mt:tst expr="&lt;=255"/>
   <mt:tst expr="&gt;=0"/>
 </mt:range>

 <!-- regexp for strings -->
 <mt:regex>
   <mt:tst false="The value does not match the first regex."
           expr="H[ae]l{2}o.*"/>
```

```
   <mt:tst false="The value does not match the second regex."
           expr=".*(Du|Sie)!"/>
 </mt:regex>

 <!-- XPath expression -->
 <mt:xpath expr=". &gt;= 42"/>
</a:a>
```

11.1.12. AUTO parameters

The module developer may now define a parameter as AUTO. The user may choose whether to manually edit such a parameter or let the generator calculate its value.

AUTOSAR:

```
<ECUC-INTEGER-PARAM-DEF>
  <SHORT-NAME>example</SHORT-NAME>
  <WITH-AUTO>true</WITH-AUTO>
</ECUC-INTEGER-PARAM-DEF>
```

XDM:

```
<v:var name="example" type="INTEGER">
   <a:a name="WITH-AUTO" value="true"/>
</v:var>
```

AUTOSAR makes no statement whether the IS-AUTO tag must be taken over from StMD to VSMD or whether the VSMD may add an IS-AUTO to standard parameters.

In the configuration, the AUTO feature is represented by the two attributes IMPORT_INFO, and AUTO:

XDM:

```
<d:var name="example" type="STRING" value="Calculated">
  <a:a name="AUTO" value="true"/>
  <a:a name="IMPORTER_INFO" value="@CALC"/>
</d:var>
```

Autosar:

```
<ECUC-TEXTUAL-PARAM-VALUE>
  <DEFINITION-REF DEST="ECUC-FUNCTION-NAME-DEF">
      /TS_T16D4M2I0R0/Eep/EepInitConfiguration/EepJobEndNotification
  </DEFINITION-REF>
  <VALUE>Eep_VendorXY_JobEndNotification</VALUE>
  <IS-AUTO-VALUE>true</IS-AUTO-VALUE>
</ECUC-TEXTUAL-PARAM-VALUE>
```

Calculation: If set to Auto by the user the code generator of the module defining the parameter must calculate the parameter value with a code generator the runs in the pre generation step. The following Java Public API methods can be used to access Auto parameters:

```
dctxt.var.supportsAuto()
dctxt.var.isAuto()
dctxt.var.setAuto( true )
dctxt.var.setAutoValue( "Value" )
```

The user is now able to let the code generator calculate the parameter. A parameter that should be calculated by the code generator is represented in the following form in the XDM format:

```
node:isAuto()
node:setAutoValue('Value')
```

For Auto parameters the type icon left to the parameter value indicates that the value of the parameter:

- is set to be calculated by the code generator
- was calculated by the code generator

The button right (context-menu for tables) allows to switch between:

- set manually edited
- set to default value
- set to auto calculation

11.1.13. Variant tags

Although EB tresos Studio 12.0 does not provide Variant support some of the variant tags show up in AUTOSAR files as the AUTOSAR variant handling introduces intermediate tags:

```
<ECUC-FUNCTION-NAME-DEF>
  <ECUC-FUNCTION-NAME-DEF-VARIANTS>
    <ECUC-FUNCTION-NAME-DEF-CONDITIONAL>
      <DEFAULT-VALUE>FunctionNameDefault</DEFAULT-VALUE>
      <MAX-LENGTH>64</MAX-LENGTH>
      <MIN-LENGTH>16</MIN-LENGTH>
      <REGULAR-EXPRESSION>FunctionName[A-Za-z_-]+</REGULAR-EXPRESSION>
     </ECUC-FUNCTION-NAME-DEF-CONDITIONAL>
  </ECUC-FUNCTION-NAME-DEF-VARIANTS>
</ECUC-FUNCTION-NAME-DEF>
```

11.1.14. Parameter type `MultilineString`

AUTOSAR provides a new parameter type that is defined with the tag `ECUC-MULTILINE-STRING-PARAM-DEF`:

```
<ECUC-MULTILINE-STRING-PARAM-DEF>
  <SHORT-NAME>MultilineString</SHORT-NAME>
</ECUC-MULTILINE-STRING-PARAM-DEF>
```

The parameters are represented by schema nodes of type `MULTILINE-STRING`:

```
<v:var name="Example" type="MULTILINE-STRING"/>
```

If published information such as parameters have the type `MULTILINE-STRING_LABEL`:

```
<v:var name="Example" type="MULTILINE-STRING_LABEL"/>
```

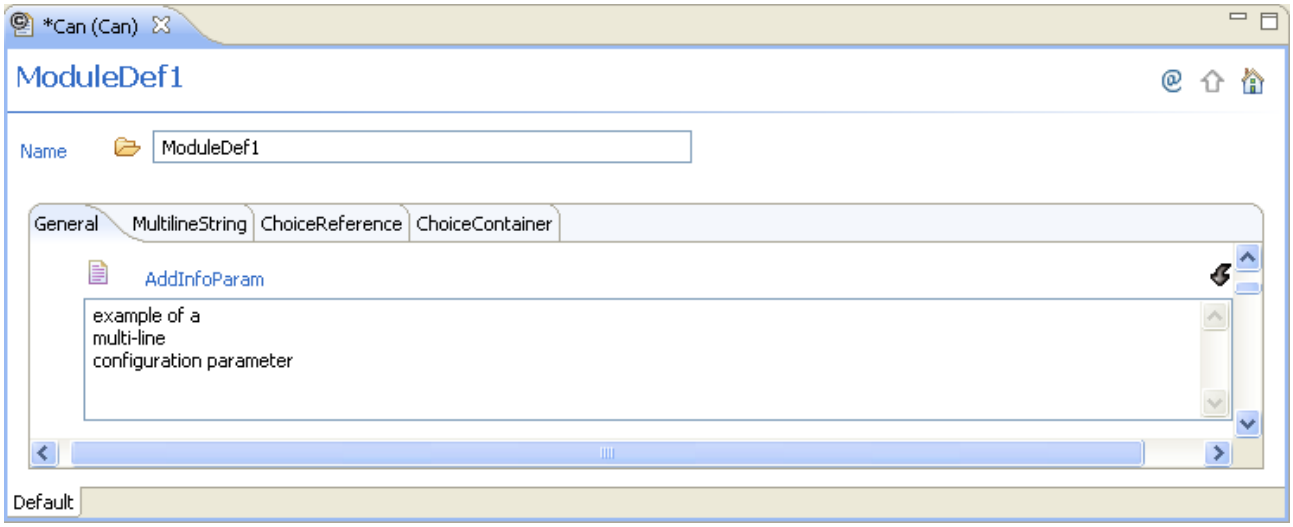
Values of such parameters are represented by data nodes of type `MULTILINE-STRING`:

```
<d:var name="Example" type="MULTILINE-STRING" value="Hallo&#10;World!&#10;"/>
```

Newlines are always represented by
 in the XDM format.

The GUI shows a corresponding multi-line widget for single value parameters and a dialog in tables:

As defined by AUTOSAR 4.0, it is now possible to use multi-line string parameters.



11.1.15. Parameter type `AddInfo`

A new parameter type has been introduced by AUTOSAR: “The parameter EcucAddInfoParamDef is used to specify the need for formated text in the ECU Configuration Value description”.

```
<ECUC-ADD-INFO-PARAM-DEF>
   <SHORT-NAME>DiagnosticTesterMessage</SHORT-NAME>
 </ECUC-ADD-INFO-PARAM-DEF>
```

```
<ECUC-ADD-INFO-PARAM-VALUE>
   <DEFINITION-REF DEST="ECUC-ADD-INFO-PARAM-DEF">
       /AUTOSAR/EcucDefs/Dcm/Dtc
   </DEFINITION-REF>
   <VALUE>
     &lt;P>
       <L-1 L="EN">Description of the Dtc 0815.</L-1>
```

```
    </P>
  </VALUE>
</ECUC-ADD-INFO-PARAM-VALUE>
```

The parameter value may contain complex rich text XML structures.

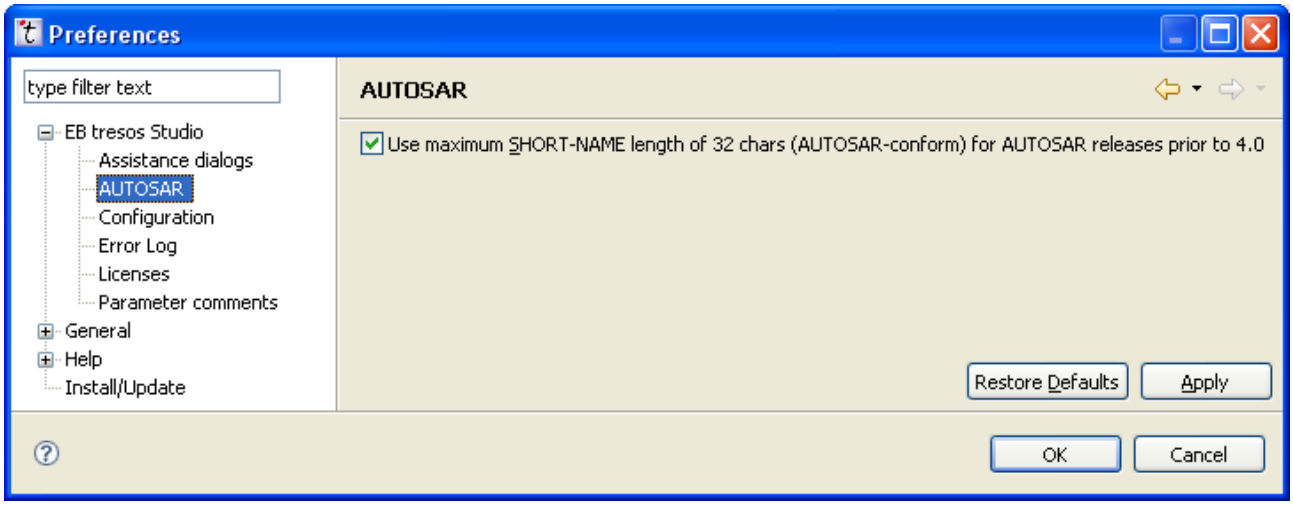
With EB tresos Studio 12.0 the richt text structures are not supported. All tags beneath `<VALUE>` are dropped during parsing of AUTOSAR files and only the plain text is used. The user may enter plain text using a multi-line textfield.

11.1.16. SHORT-NAME restrictions

In AUTOSAR 4.0, SHORT-NAMES now:

- may contain up to 128 characters
- may not contain two subsequent underscores '__-'
- clash with other SHORT-NAMES that only differ in case. However, paths are still case sensitive.

Since AUTOSAR 4.0, the maximum possible length of the SHORT-NAME has bee changed from 32 to 128 characters. The preference to disable the length-check is now restricted to modules prior to AUTOSAR 4.0:



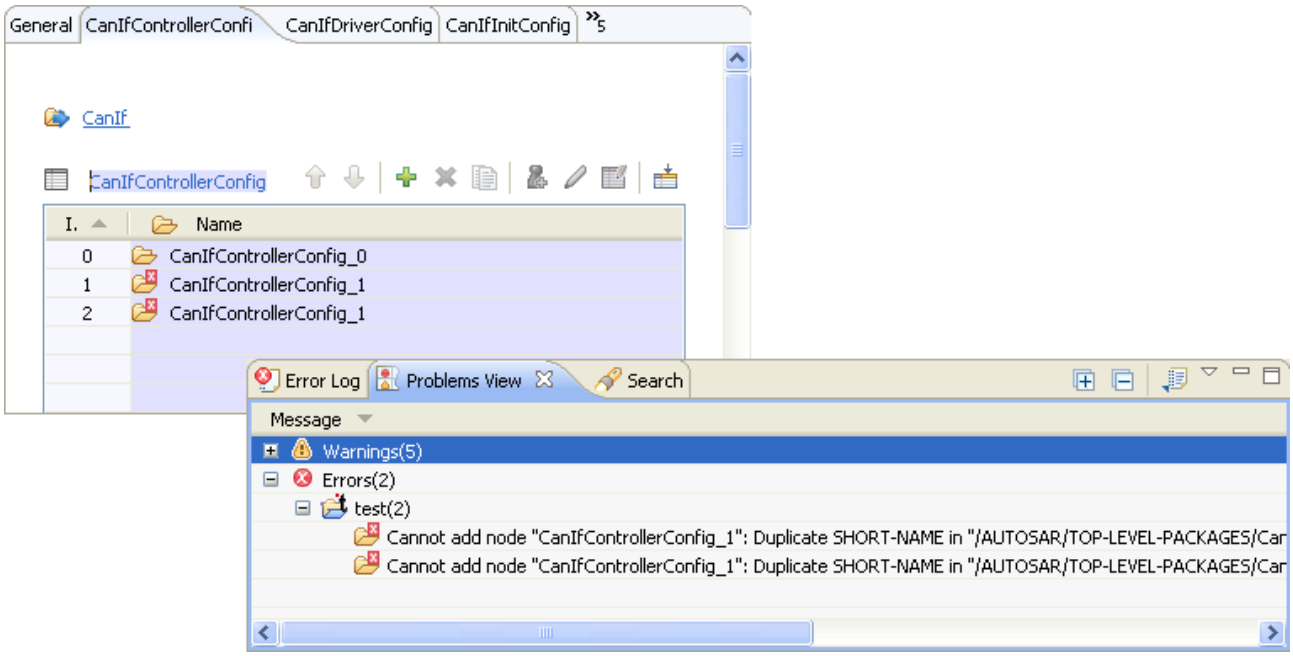
Because XDM is currently revision-independent, a new attribute has been introduced, which is mandatory for 4.0. modules. You must add this attribute to the `MODULE-DEF` node of all new (4.0) modules.

```
<v:ctr type="MODULE-DEF">
    <a:a name="RELEASE" value="asc:4.0"/>
```

The attribute may be omitted for older releases, but if defined, it must match the relVease version specified for the module within the `plugin.xml`:

```
<module id="myId"
        ...
    relVersionMajor="4"
    relVersionMinor="0"
        ...
</module>
```

Configurations may now contain invalid SHORT-NAMES - or in general invalid names. Whether or not a name is invalid may depend on the RELEASE attribute, but e.g. a duplicate name is also an invalid name. In such a case, an error is added to the **Problems** view and the user may change the name to a valid one:



11.1.17. Changes to the EB tresos Studio command line

The new content type `asc:4.0` is introduced for files containing AUTOSAR ECU Configuration Description or Parameter Definition. The content type `sysd:4.0.2` is introduced for files containing an AUTOSAR 4.-0 Software Component Template. According to this the new content type, `tdb:4.0` represents the internal storage format for system description data.

Examples:

- To convert a Parameter Definition file to xdm format, use the following command:

  ```
  legacy convert can.bmd@asc:4.0 can.xdm
  ```

- To convert a 4.0 Software Component Template file to the internal tresosDB format, use the following command:

legacy convert swc.arxml@sysd:4.0.2 out.tdb@tdb:4.0.

- To execute the RTE Generator using an AUTOSAR 3.1 module configuration together with a 4.0 Software Component Template, use the following command:

  legacy generate rte.epc@asc:3.1 swc.arxml@sysd:4.0.2.

11.1.18. Unsupported AUTOSAR 4.0 features

- Documentation support
  - AUTOSAR defines many rich text tags and attributes: `LONG-NAME`, `DESC`, `INTRODUCTION`, `ANNOTATIONS` with corresponding rich text XML tags below, which are ignored by EB tresos Studio.
  - Ignored XML tags: `ANNOTATIONS`, `DOCUMENTATION-BLOCK`, `DOCUMENTATION-BLOCK`, `INFORMAL-FORMULA`, `MIXED-CONTENT-FOR-OVERVIEW-PARAGRAPH`, `MIXED-CONTENT-FOR-LONG-NAME`.
- Variant handling

  Ignored XML tags: `ATTRIBUTE-VALUE-VARIATION-POINT`, `VARIATION-POINT`.
- Calculation formula for derived parameters

  AUTOSAR now defines a formal calculation language for derived parameters. This language can be used beneath the tag `DERIVATION`. As EB tresos Studio does not support the AUTOSAR-defined formal language, those values cannot be calculated via the AUTOSAR language.
- Parameters can be enabled or disabled depending on the value of other parameters with the tag `ECUC-COND`, similar to the `ENABLE` attribute. As EB tresos Studio does not support the AUTOSAR-defined calculation formula, this tag is not supported either.

11.2. Team collaboration support

EB tresos Studio now provides new features that help you work together on a project as a team. The following list informs you about the major features:

- You are now able to store EB tresos Studio project in version control systems via the Eclipse *Team* support. EB tresos Studio ships with support for the version control systems *CVS* and *SVN*. Other version control systems could be supported transparently by EB tresos Studio via the Eclipse APIs, but only CVS and SVN have been tested.
- You can now reload the project data from disk. To do so, click the **Reload Configuration** context menu item in the **Project Explorer** view. Reloading the project is required whenever the files on disk have changed,

e.g. when you have updated your SVN repository of the project. Currently, you can only reload a project manually.

- You can now compare XDM files with a graphical compare editor. The **Compare** editor for XDM files is registered in the Eclipse framework and opens up every time you compare XDM files. The editor displays the parameters in the XDM files as a tree. Different types of changes are highlighted with a different background color. Unlike plain text or XML compare editors, the **Compare** editor for XDM files takes the AUTOSAR information into account and e.g. also displays moved list items or enablement changes.

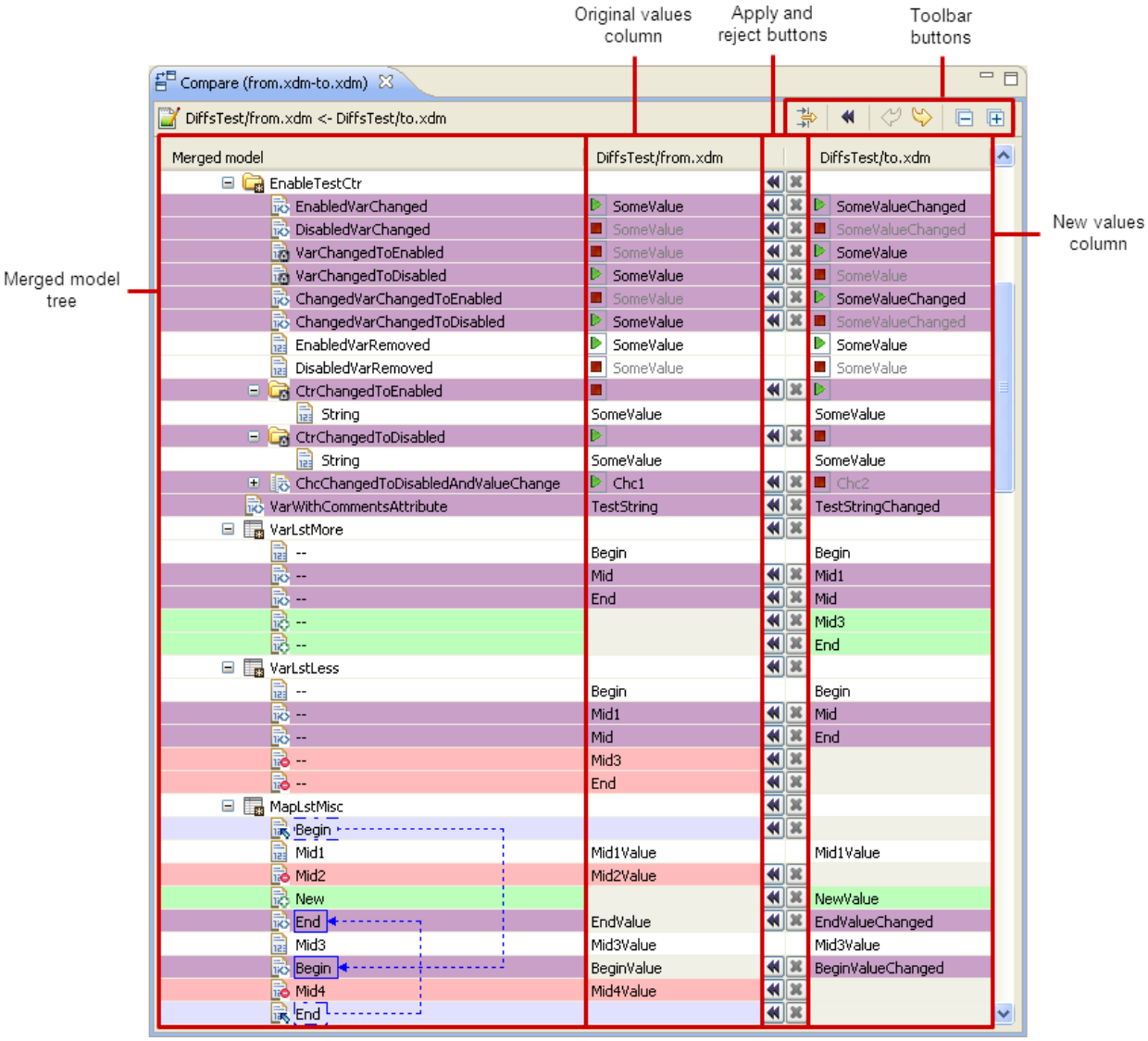


Figure 11.3. Example of the graphical **Compare** editor for XDM files

11.3. The EB tresos Studio documentation has been restructured

The EB tresos Studio documentation has been restructured and split into two parts: The first part is written for customers who want to use EB tresos Studio as a configuration and code generation tool. It is named `2.1_-Studio_documentation_users_guide.pdf`. The second part is written for customers who want to extend EB tresos Studio with their own functionality and who want to use the Public APIs. The second document is named `2.4_Studio_documentation_developers_guide.pdf`.

Both documents have been restructured to suit the target group. In particular the EB tresos Studio developer's guide has been restructured fundamentally to give you an easy access to the important topics and a fast overview of the content.

11.4. System Description Exporter

EB tresos Studio provides an exporter with which you may export the AUTOSAR system description (Sys-D) and the software component description (SWC-D) of a configuration project.

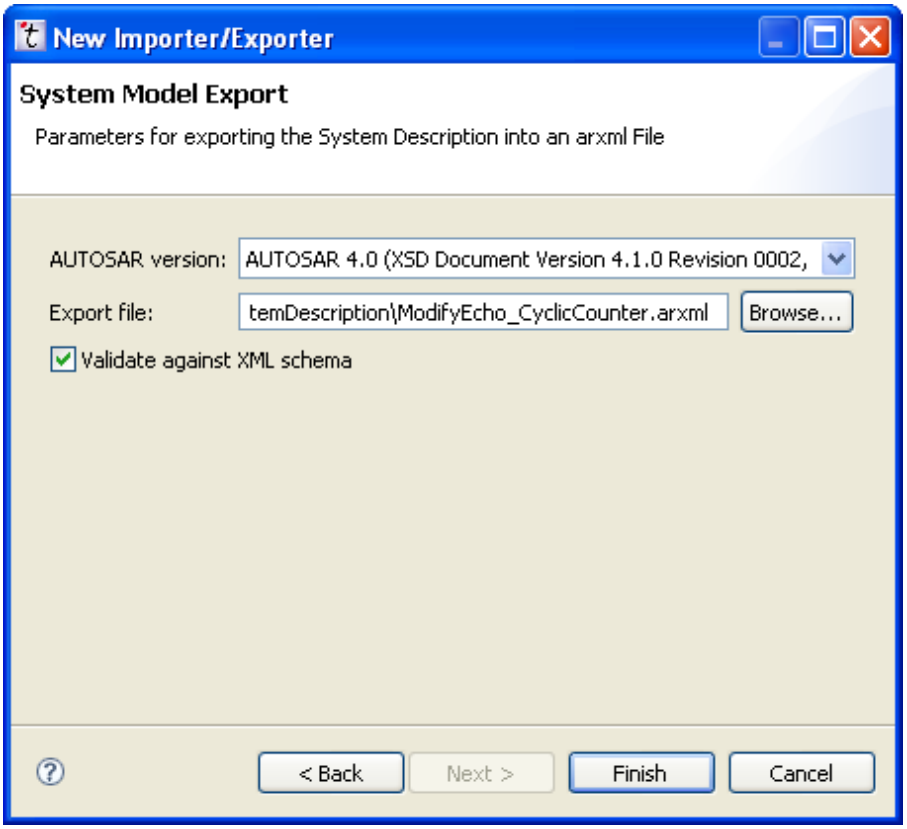


Figure 11.4. Creating a System Description Exporter

Supported AUTOSAR meta model formats for system model 1 are:

- AUTOSAR 3.0 (XSD Document Version 3.0.2 Revision 0003, xmlns="http://autosar.org/3.0.2")
- AUTOSAR 3.1 (XSD Document Version 3.1.0 Revision 0001, xmlns="http://autosar.org/3.1.0")
- AUTOSAR 3.1 (XSD Document Version 3.2.1 Revision 0003, xmlns="http://autosar.org/3.1.2")
- AUTOSAR 3.1 (XSD Document Version 3.3.0 Revision 0004, xmlns="http://autosar.org/3.1.4")
- AUTOSAR 3.1 (XSD Document Version 3.3.0 Revision 0004 (Daimler specific version 2), xmlns="http://autosar.org/3.1.4.DAI.2")
- AUTOSAR 3.1 (XSD Document Version 3.3.0 Revision 0004 (Daimler specific version 3), xmlns="http://autosar.org/3.1.4.DAI.3")
- AUTOSAR 3.1 (XSD Document Version 3.3.0 Revision 0004 (Daimler specific version 4), xmlns="http://autosar.org/3.1.4.DAI.4")

Supported AUTOSAR meta model formats for system model 2 are:

- AUTOSAR 4.0 (XSD Document Version 4.1.0 Revision 0002, xmlns="http://autosar.org/r4.0")

11.5. Improvements to the Public APIs

11.5.1. Drag and drop support for `SWTTree` and `SWTTable`

You can now implement drag and drop functionality for your guided configuration GUI for the classes `SWTTreeTable` (`dreisoft.tresos.guidedconfig.api.gui.form.SWTTreeTable`) and `SWTTable` (`dreisoft.tresos.guidedconfig.api.gui.form.SWTTable`).

11.5.2. Improvements to context updates

The update handling of the classes `ECUConfigContext`, `ModuleConfigContext`, and `SystemConfigcontext` has been improved. The correct context information is now preserved as long as possible, even if the user activates other views or editors.

11.5.3. The guided configuration editor can be closed automatically

If a guided configuration editor is open, this editor blocks all actions on the current project that could change this project. In previous versions of EB tresos Studio, these blocked actions were disabled and you had to close

the guided configuration editor manually to unlock these actions. Now, if you choose such a blocked action on the project, a confirmation dialog opens up and asks you whether the guided configuration editor shall be closed automatically.

Note that if the guided configuration editor has validation errors, automatic closing is not possible and you are informed about this. If you choose to not close the guided configuration editor or if automatic closing is not possible due to validation errors, the selected action is not performed.

11.5.4. New style constant `SHOW_READONLY_ROW_CONTENT`

A new style constant `SHOW_READONLY_ROW_CONTENT` has been introduced in the enumerated type `dreisoft.tresos.guidedconfig.api.gui.WidgetFactory.TreeTableFlags`. Any instance of `dreisoft.tresos.guidedconfig.api.gui.form.SWTTreeTable` created with the new style constant shows the content of all columns for read-only rows. Without this constant a tree only shows the value of the first column of those rows.

11.5.5. The DCtxt method `isExisting()` has been fixed and `isAccessible()` has been introduced

You can select nodes by XPath expressions that do not exist. If you use such nodes in XPath expressions and try to evaluate them in a DCtxt, the XPath engine implicitly assumes values for the nodes even if the nodes do not exist. There are two reasons why a node can be non-existing:

1. The node really does not exist, i.e. `EXISTING = false` in the schema file, or if you use wrong path arguments.
2. The node is disabled, i.e. optional disabled nodes or nodes whose enablement is calculated based on other parameters.

When you try to access the first type of non-existing nodes, the DCtxt always returns null. When you try to access disabled nodes, the return value depends on the option `ExistingNodesOnly`. You can query this option by calling

```
container.opt.getExistingNodesOnly()
```

If the option `ExistingNodesOnly` is set to `true`, only existing and enabled nodes can be accessed. The DCtxt returns `null` when you try to access a disabled node.

If the option `ExistingNodesOnly` is set to `false`, then you can also query disabled nodes.

There already exists a method to query if a node is existing. This method always returns `false` for disabled nodes, disregarding the state of `ExistingNodesOnly`:

```
ctxt.isExisting(String path)
```

In EB tresos Studio 12.0, the new method `isAccessible` is implemented, which returns `true` for disabled nodes, if the option `ExistingNodesOnly` is set to `false`.

```
ctxt.isAccessible(String path)
```

NOTE

The return values may vary

If the option `ExistingNodesOnly`is set to `false`, the return values from evaluating XPath expressions within DCtxt may differ from evaluating the same XPath expression within e.-g. template based code generator. This only occurs when you try to access disabled nodes in the XPath expression.

The corresponding XPath function to the DCtxt method `isAccessible()` is also introduced:

```
node:accessible()
```

The XPath function `node:accessible()` - used outside of the DCtxt - returns the same value as known from `node:exists()`.

12. Changes for release 11.0

12.1. Module-to-cluster assignment extension point

For an easier handling within the **Module Configurations** dialog and the **New Project Wizard** you can now assign modules to clusters. For improved flexibility for module providers and solution partners, you can also customize the module-to-cluster assignment by contributing to the new `modulecluster` extension point. In this extension point, you can define default module-to-cluster assignments for modules that are not automatically assigned to clusters and you can also override any existing module-to-cluster assignments. To do that, select the modules by their `categoryType` or `id` attributes. The `id` attribute even supports regular expressions.

The following shows an exemplary contribution:

```
<extension point="dreisoft.tresos.launcher2.api.plugin.modulecluster">
  <modulecluster>
    <default>
      <cluster name="Can Stack"/>
      <cluster name="Fr Stack"/>
      <module type="PduR"/>
      <module type="Com"/>
    </default>
    <default>
     <cluster name="Can Stack"/>
     <module type="CanIf"/>
    </default>
    <override>
     <cluster name="Can Stack"/>
     <module regExpModuleId="Can_TS_T16D4M[0-9]I[0-9]R[0-9]"/>
    </override>
  </modulecluster>
</extension>
```

12.2. Multiple perspectives support

EB tresos Studio now features support for multiple workbench perspectives. *Perspectives* are arrangements of views and of the editor area together with available menu and tool bar actions. There are two different kinds of perspectives: pre-defined ones, such as the **Studio** perspective, and user-defined ones.

To save a user-defined perspective:

- Rick-click on the customized perspective in the perspective bar.

  A context menu opens up.

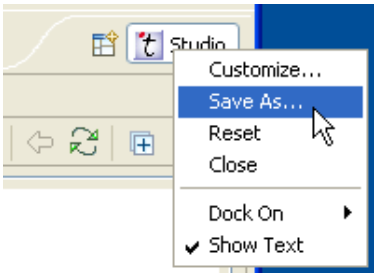


Figure 12.1. Saving a user-defined perspective

- Select **Save As...**.
- The **Save Perspective As...** dialog opens.

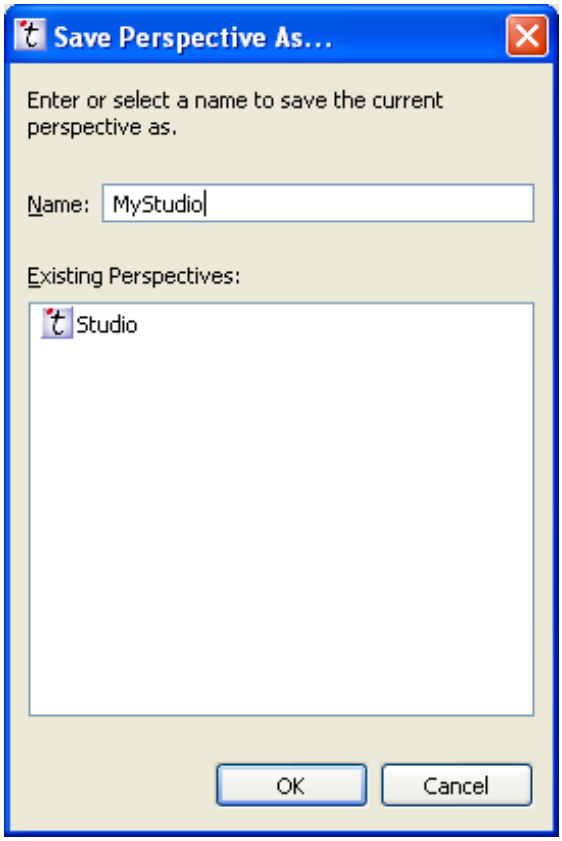


Figure 12.2. Entering a name for your user-defined perspective

- Enter a name and click **OK**.
- Your perspective is saved.

To open a **Perspective**:

- In the **Window** menu, select **Open Perspective**.

  A submenu opens up.

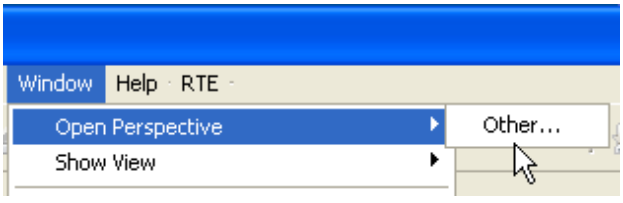


Figure 12.3. Opening a perspective

- Select one of the perspectives listed directly.

  The perspective opens up and is added to the perspective bar.
- If no perspective is listed, select **Other...** to open the **Open Perspective** dialog.

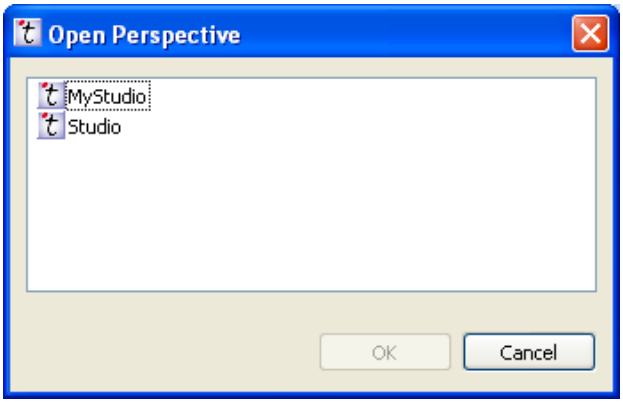


Figure 12.4. The **Open Perspective** dialog

- Select the perspective of your choice and click **OK**.
- The perspective opens up and is added to the perspective bar.

To switch between different open perspectives:

- Locate the perspective bar. By default, you find the perspective bar in the top right corner of the workbench.

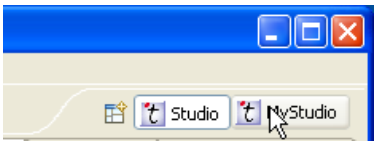


Figure 12.5. The perspective bar

- Select the perspective you want to switch to.

  The selected perspective opens up.

To delete a user-defined perspective:

- In the **Window** menu, select **Preferences...**
- The **Preferences** dialog opens up.

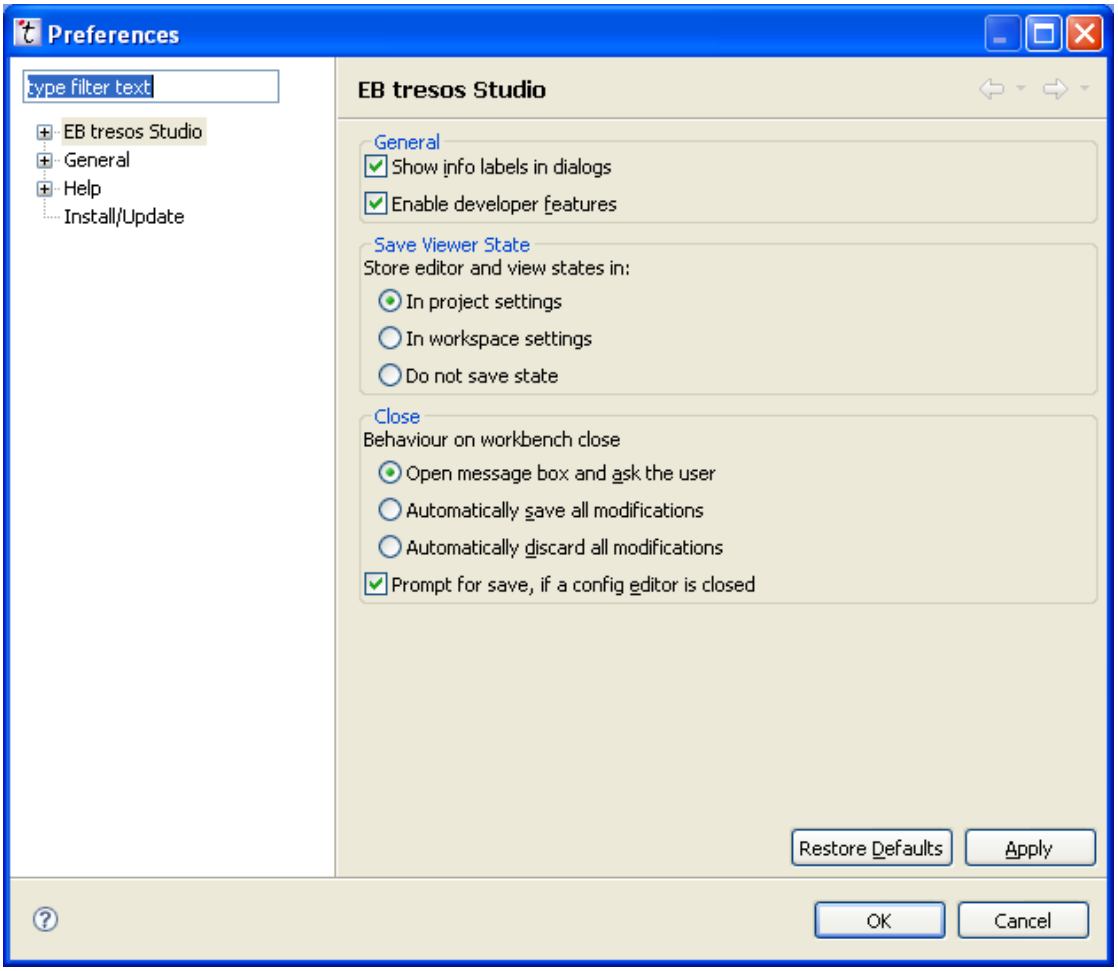


Figure 12.6. Deleting user-defined perspectives in the **Preferences** dialog

- On the left side, navigate to the **General/Perspectives** page.
- Locate the section **Available perspectives:**

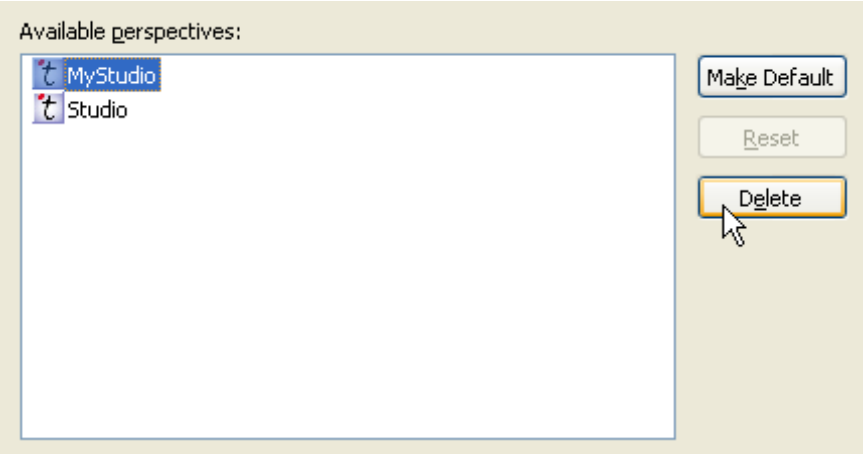


Figure 12.7. **Available perspectives:** preference

- Select the user-defined perspective that you want to delete and click **Delete**.
- Click **OK**.

  The selected perspective is deleted.

For more details on how to work with workbench perspectives, consult the EB tresos Studio user's guide.

12.3. Comments for configuration parameters

EB tresos Studio now supports to add comments for configuration parameters. With the new dialog **Comments for <parameter>** you can add comments of a parameter and edit existing comments. It is possible to add comments for different languages.

The figure below shows the **Comments for <parameter>** dialog. In this dialog, you may choose the language in which the comment shall be written, add a summary, description and set a status for the comment.

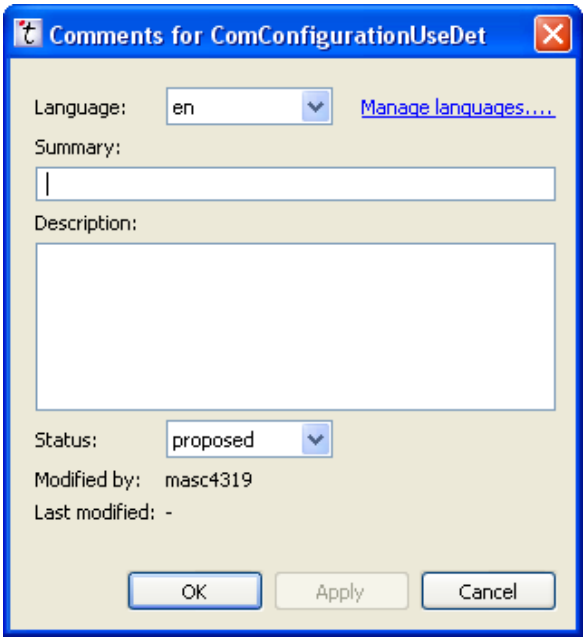


Figure 12.8. The **Comments for <parameter>** dialog

12.4. Changes to assistance dialogs

12.4.1. Dynamic editing in tables and trees

The support for direct editing in table or tree widgets (, i.e. *cell editing*) in the Guided Configuration API has been extended. In prior EB tresos Studio versions the custom components for the cell editing were limited in two ways: Firstly, one column could only have one kind of cell editor. Secondly, an existing cell editor could not be changed on-the-fly. Both limitations have been eliminated. You may now use different cell editors in different rows of the same column, and you can also change these editors at runtime. For example, that way you can change the editor type of a specific table cell depending on the user's input in another cell.

12.4.2. You can now place guided configuration widgets on non-guided configuration dialogs or views

You can now place the widgets provided by the guided configuration framework outside of a guided configuration page, e.g. on an external dialog or view. To create these widgets, use the `WidgetFactory` class as usual.

If the dialog or view is closed, widgets and their controllers are disposed correctly and the widgets do not not receive events. If the dialog or view is opened again, the widgets are re-instantiated.

Note that custom widgets used in that way need to unregister their controller from the `WidgetFactory` themselves when they are disposed.

12.5. Changes to Workflow API

12.5.1. You can now turn off the automatic selection of the next step in the Workflows view

By default each step in the **Workflows** view is automatically marked with a **Finished** marker after you have performed an action command and the next step is automatically selected. If you do not want to automatically navigate to the next step after running a command, you may now turn off this behavior by setting the new optional attribute `autoadvance` of the `command` tag to `false` in your workflow description file:

```
<command description="Open assistance dialog"
  serialization="dreisoft.tresos.guidedconfig.api.plugin.
                          SidebarTriggerCommand(triggerType=GCDemo)"
  autoadvance="false"/>
```

This may be useful if you had opened an editor via a command from the **Workflows** view. Prior to adding this feature, the **Workflows** view automatically navigated to the next step. As a result, the description of this action step in the **Workflows** view would not be available anymore as the **Workflows** view already displayed the next step.

12.6. AUTOSAR system model version 3.1.4 is now supported

The AUTOSAR system model support has been enhanced. EB tresos Studio now supports the AUTOSAR system description version 3.1.4 and the Daimler-specific enhancements 3.1.4.DAI2, 3.1.4.DAI.3 and 3.1.4.-DAI.4 in addition to the already supported versions. You can now import AUTOSAR system description files of these versions into your projects and convert them into the TresosDB file format with the `legacy convert` command on the command line.

12.7. Resource Public API

EB tresos Studio now provides a Resource Public API with which you may load and store ECU configuration files from all file formats supported by EB tresos Studio.

With the Resource Public API you can accomplish similar tasks by writing Java code as on the command line with the `legacy convert` command. For example, the following code fragment converts an `XDM` file into an AUTOSAR ECU configuration file:

```
APIOperationStatus status = APIOperationStatus.getOkStatus();
RCtxt rctxt = RCtxt.create( new Location( new File( "Dem.xdm" ) ) );
rctxt.load( EcuConfigContext.getInstance().getDCtxt(), null, status );
rctxt.setLocation( new Location( new File( "Dem.arxml" ),
ImporterConstants.CONTENT_TYPE_ASC + ":3.1" ) );
rctxt.save( EcuConfigContext.getInstance().getDCtxt(), null, status );
```

The following code fragment imports an AUTOSAR system description file into the ECU configuration of a project:

```
APIOperationStatus status = APIOperationStatus.getOkStatus();
RCtxt rctxt = RCtxt.create(
  new Location( new File( "SystemDescription.arxml" ),
                ImporterConstants.CONTENT_TYPE_SYSD ) );
Map<String,String> options = new HashMap<String,String>();
options.put( DCtxt.MERGE_SOURCE, "system" );
options.put( ImporterConstants.OPTION_SYSTEM, "/System/SystemESP" );
options.put( ImporterConstants.ECU_INSTANCE, "/System/EcuESP" );
options.put( ImporterConstants.OPTION_PROJECT, "myProject" );
options.put( ImporterConstants.OPTION_MODULE( "EcuC" ), "EcuC" );
options.put( ImporterConstants.OPTION_MODULE( "PduR" ), "PduR" );
options.put( ImporterConstants.OPTION_BUFFER_ASSIGNMENT,
             ImporterConstants.VAL_BUFFER_ASSIGNMENT_TRUE );
options.put( ImporterConstants.OPTION_INSTANCE_HANDLING,
             ImporterConstants.VAL_INSTANCE_SUFFIX_TRUE );
rctxt.load( _ecu_config, options, status );
EcuConfigContext.getInstance().getDCtxt().merge( rctxt, options, status );
```

Further information on the Resource Public API is available in the EB tresos Studio developer's guide.

13. Changes for release 10.0

13.1. Changes to assistance dialogs

13.1.1. Results view

The results of assistance dialogs now are displayed in the **Results** view instead of in a modal dialog. This concerns the following types of assistance dialogs:

- dialogs opened from the **Sidebar** view
- unattended wizards started with the tool bar button  .

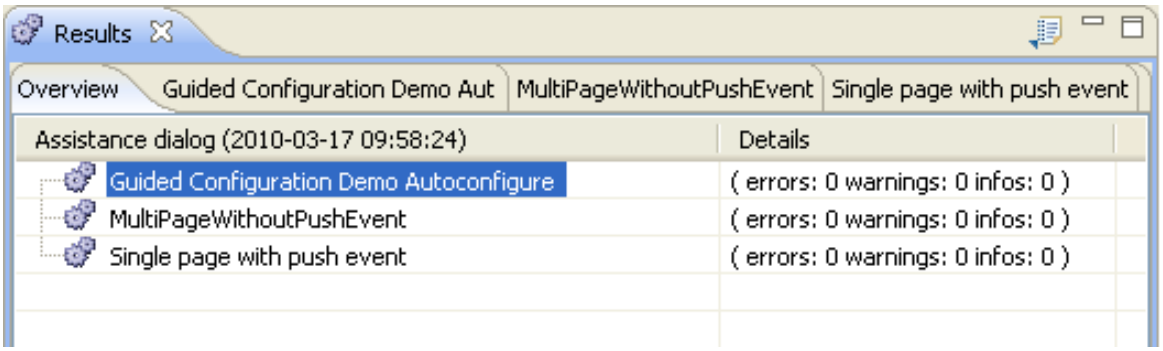


Figure 13.1. The **Results** view

The **Results** view displays a summary of all executed wizards in its **Overview** tab and contains one tab for every wizard executed. You can export the results by clicking the  button in the tool bar of the **Results** view.

After a dialog or unattended wizard has been executed, an information dialog is displayed to inform you about the results in the **Results** view.

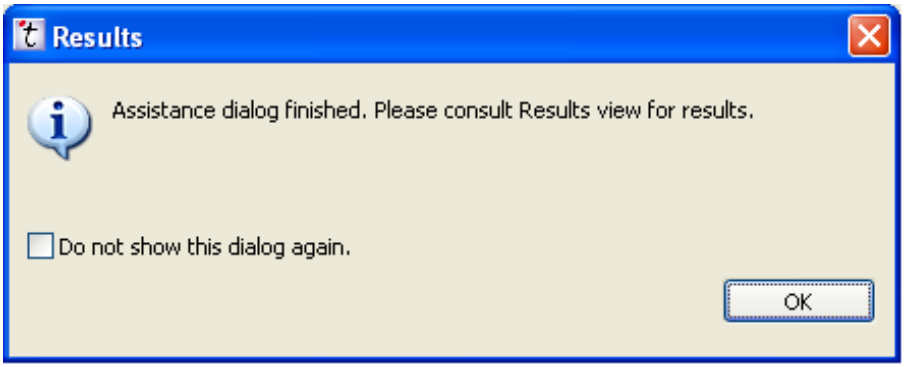


Figure 13.2. Results view info dialog

You may disable this information dialog by selecting the **Do not show this dialog again** check box.

To reset this setting:

- In the **Window** menu, select **Preferences...**.

  The **Preferences** dialog opens up.
- Select **EB tresos Studio/Assistance dialogs** in the tree view:

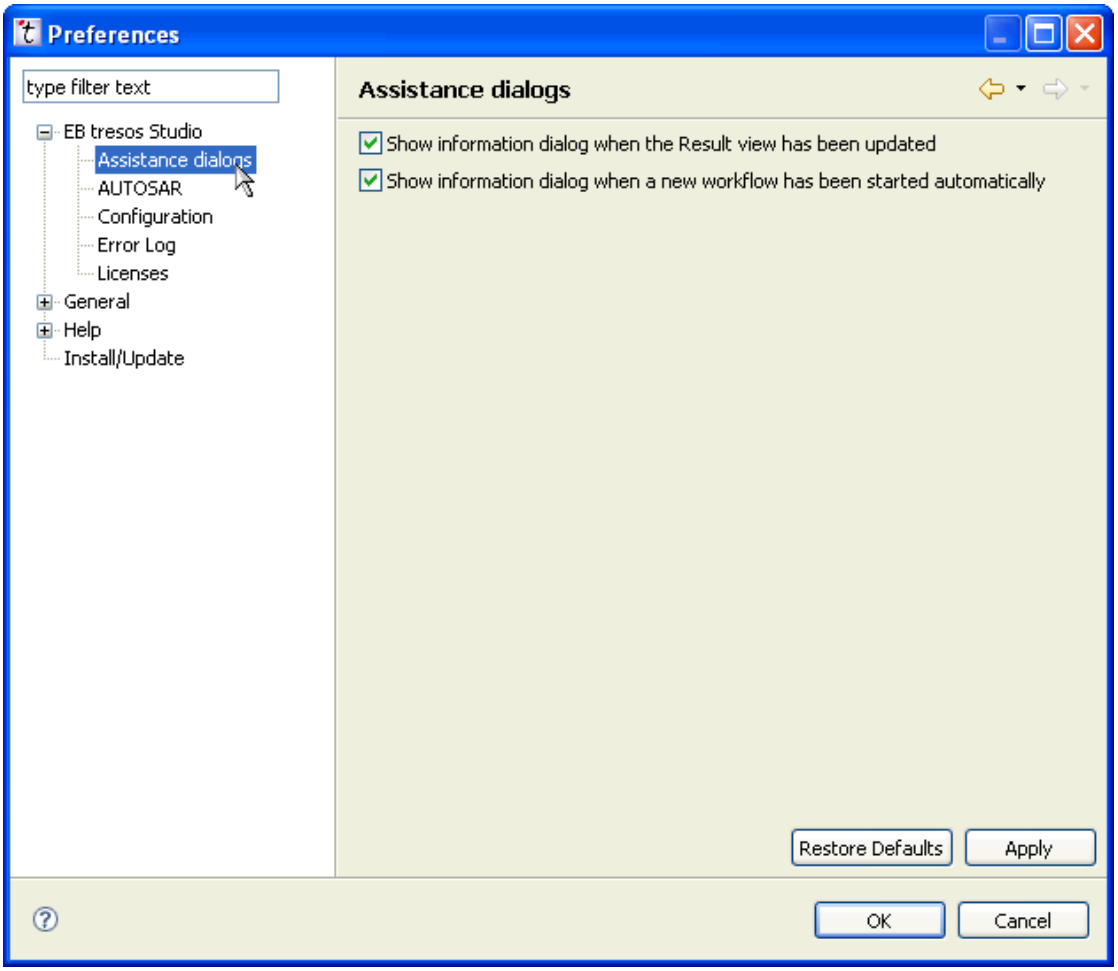


Figure 13.3. Changing the preferences of assistance dialogs

- Select **Show information dialog when the Result view has been updated**.
- Click **Apply** and **OK**.

Assistance dialogs of the type *editor* dot not display their results in the **Results** view, but write their results to the **Error Log** view now. This concerns the following editors:

- custom module editors opened from the **Project Explorer** view
- stand-alone editors opened from the **Sidebar** view

13.1.2. Improvements of the SWTTable widget

The `SWTTable` API class has been enhanced by the following features. These features apply to assistance dialogs that use the widget without modifications:

- Cell editor for Boolean (check box) cells

In previous EB tresos Studio versions you could only possible change the value of a Boolean cell with a mouse click anywhere in the cell. Now, you can only edit the value, if you click exactly at the check box. If you click anywhere else in the cell or if you move from one cell to the Boolean cell with the **TAB** key, a cell editor opens up. The check box control of this cell editor differs slightly from the normally displayed check box so you can identify it easily. When the cell editor opens up, it displays the current value of the Boolean cell (not toggled). The check box value can then be changed by pressing the **SPACEBAR** key or by clicking it with the mouse. Changing the value does not close the editor. You can apply the value by pressing the **ENTER** key or by closing the cell editor (e.g. with the **TAB** key); if you press **ESC**, the editor closes without saving your modifications.

- Improved keyboard support
  - The **INSERT** key: Executes the **Add new element with default values** action, if it is available in the table's tool bar.
  - The **DELETE** key: Executes the **Remove selected elements** action if it is available in the table's tool bar.
  - The **F2** key: Opens the cell editor of the first editable cell of the currently selected row.
  - The **CTRL-A** key: Selects all rows in the table.
  - The **ESC** key: Clears all rows in the table.
  - The **TAB** key: With the check box cell editor it is now possible to completely edit a table with the keyboard. Accessing a Boolean cell with the **TAB** key did not work in previous releases.
- Improved layout

  The table and the GUI tree widgets now resize their visible area according to the free space around the GUI widgets. If you resize editor windows or dialogs, the trees and tables now resize themselves as well.

13.1.3. Autoconfiguration wizards are now called unattended wizards and have been improved

The autoconfiguration wizards have been renamed to *unattended wizard*s and the **Autoconfigure Wizard Configuration** dialog has been renamed to **Unattended Wizards** dialog. The name of the selected wizard is now displayed in the title area of this dialog:

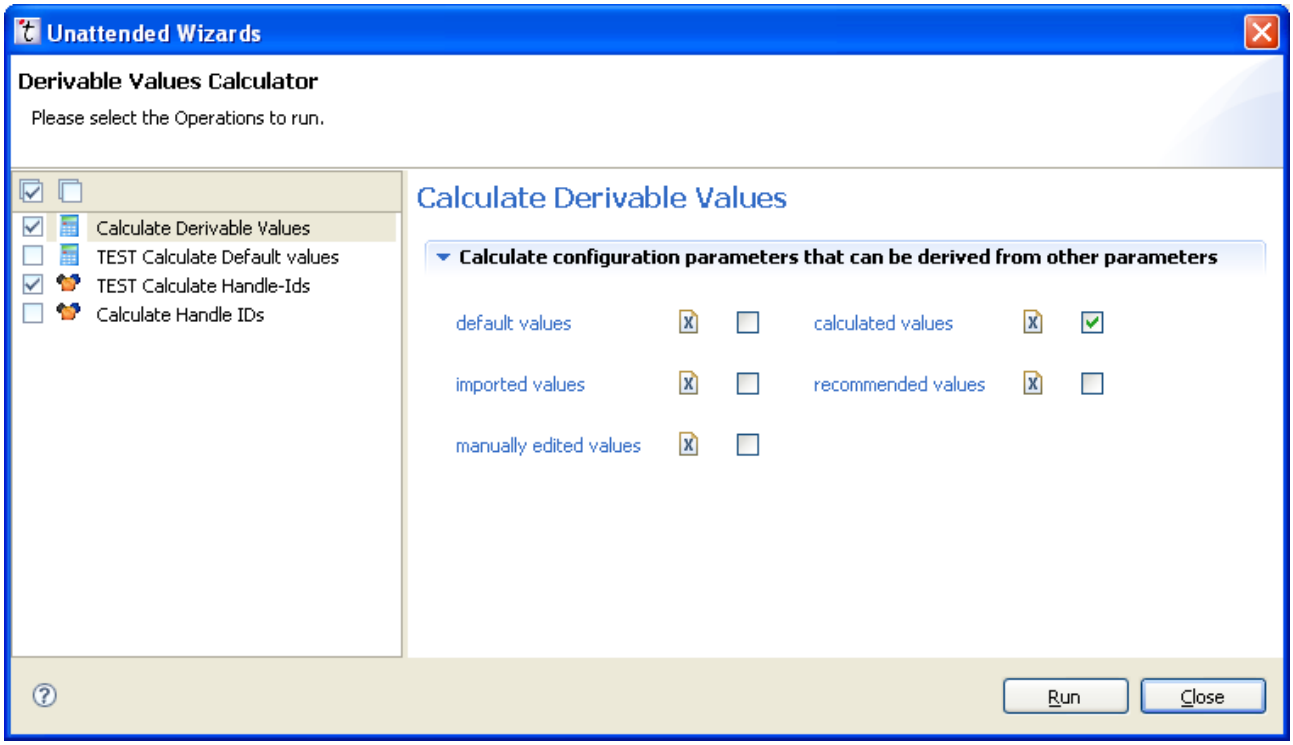


Figure 13.4. The **Unattended Wizards** dialog in multiple page mode

For the sake of simplicity only two buttons are available now:

Run
If you click the **Run** button, only the currently selected unattended wizard is executed.

Close
If you click the **Close** button, the configuration of the unattended wizards is saved and the **Unattended Wizards** dialog closes.

The former **Run checked** button has been removed. You can still mark wizards in the **Unattended Wizards** dialog so that you can start them later manually in this dialog. To start an unattended wizard manually:

- Either click the tool bar button .
- Or select the new menu item **Run all marked**:

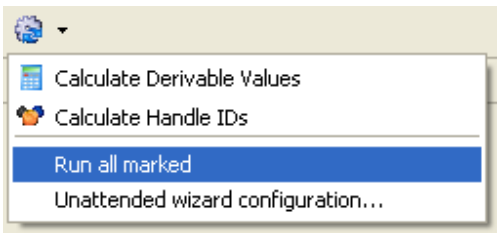


Figure 13.5. The **Run all marked** menu item

EB tresos Studio comes with a new view, called the **Workflow** view. If you are writing your own workflow you can open the **Unattended Wizards** dialog by using the `dreisoft.tresos.guidedconfig.api.plugin.AutoConfigureDialogCommand` command handler. You can use parameters to filter the unattended wizards shown in the dialog. If only one unattended wizard is shown in the **Unattended Wizards** dialog, the dialog is opened in *single page mode*:

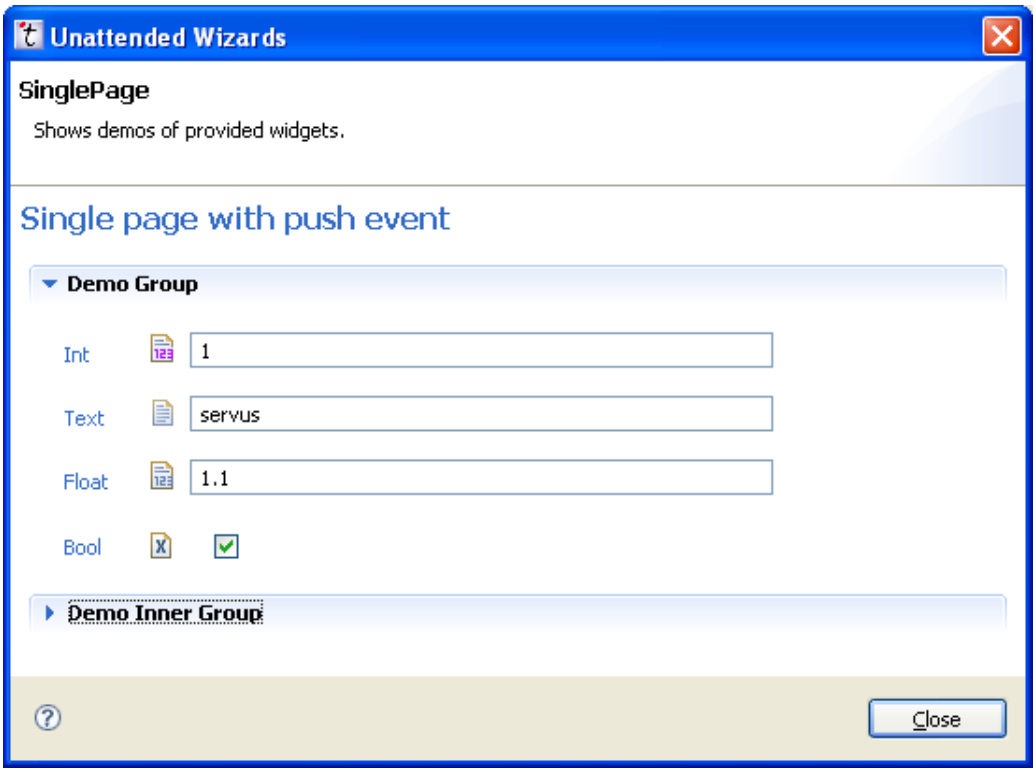


Figure 13.6. The **Unattended Wizards** dialog in single page mode without the **Run** button

The **Unattended Wizards** dialog in single page mode has a simplified layout that does not include the list of wizards on the left and does not contain the **Run** button.

13.1.3.1. Listing the available unattended wizards in the command line

You can use a new command in the EB tresos Studio command line to list the IDs of all available unattended wizards that can be executed via the command line. The command is called **listautoconfigure**.

For example, if your project is called `myproject`, you can now call EB tresos Studio with the command **tresos_cmd.bat listautoconfigure project** and all available unattended wizards will be listed in the command line. To run one of these wizards, use the **autoconfigure** command on the command line.

13.2. Workflow API

The EB tresos Studio Public API has been extended by a *Workflow API*. The Workflow API provides the possibility to enhance EB tresos Studio with workflows, which guide the user through the tool. A workflow consists of a list of steps the user has to perform to accomplish a certain task, e.g. to configure a Com stack from scratch.

A workflow is described by a workflow description file written in XML. The workflow schema can be found at:

```
http://www.tresos.de/_projects/tresos/workflow_1_0.xsd
```

A separate view, called the **Workflows** view, displays one workflow at a time and provides functionalities to process the tasks.

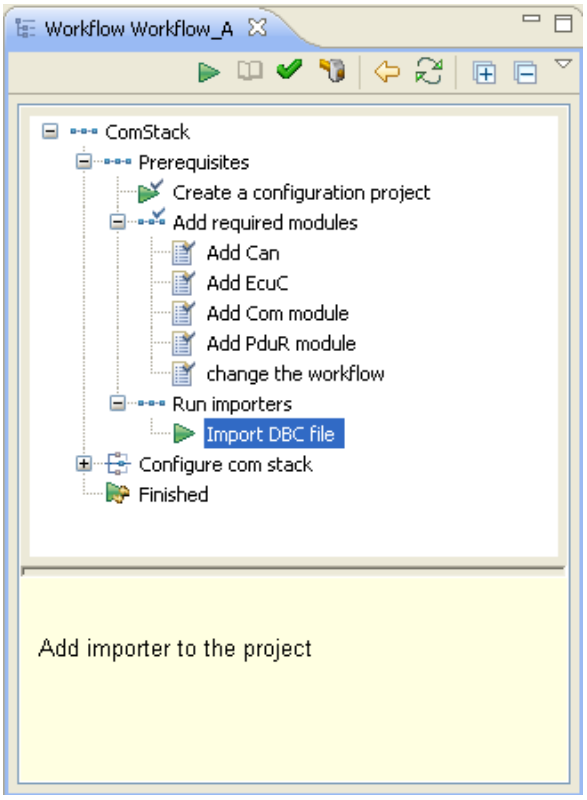


Figure 13.7. The **Workflows** view

In the **Workflows** view the user can go through the workflow step by step and perform the actions described for each step. The actions may be only descriptions of what to do or may execute commands, either Eclipse commands or EB tresos Studio specific commands.

The Workflow API provides a collection of EB tresos Studio specific commands, e.g. to open the importer/exporter dialog or to run an unattended wizard. These commands can be used in addition to the commands provided by the Eclipse framework, e.g. to create a new file or to open a message dialog.

13.2.1. Working with the Workflows view

The **Workflows** view is displayed by default when you open EB tresos Studio for the first time. If the **Workflows** view is not visible in EB tresos Studio, you can open it by selecting **Show View** in the **Window** menu. In the menu that opens up, select **Workflows**.

A more detailed description about the **Workflows** view is available in the EB tresos Studio user's guide in chapter *The GUI main window*.

13.2.2. Writing your own workflow descriptions

A detailed description about the Workflow API is available in the EB tresos Studio developer's guide, chapter *Workflow API*. This chapter explains

- the XML schema for the workflow description
- how to register a new workflow
- how to use workflow hyperlinks
- the functionality of the **Workflows** view.

There also exists a demo plug-in named *WorkflowDemo1*. This demo is located at `$TRESOS_BASE/demos/`.

The demo workflow shows all EB tresos Studio specific commands and how workflows can be linked together. You find a description of the demo plug-in in the EB tresos Studio developer's guide in chapter *Workflow API*.

13.3. Module configurations

The following sections describe the changes in EB tresos Studio concerning the handling of module configurations in Configuration Projects.

13.3.1. Clustering of modules

Modules often belong to or depend on each other. To reflect this, contributors of the module extension point can now assign modules to clusters using the `cluster` element:

```
<extension point="dreisoft.tresos.launcher2.plugin.module" id=... >

  <module id=... >
    ...
    <cluster
```

```
            name="Cluster One">
        </cluster>
        <cluster
            name="Cluster Three">
        </cluster>
    </module>
</extension>
```

Modules can belong to zero, one or several clusters.

13.3.2. Improved user interface for handling module configurations

The user interface (UI) for handling the module configurations of a configuration project in EB tresos Studio has been redesigned for improved usability. This regards the **New Project Wizard** that is used to create new ECU configuration projects, as well as the **Module Configurations** dialog and the **Properties** dialog, both used to edit existing projects.

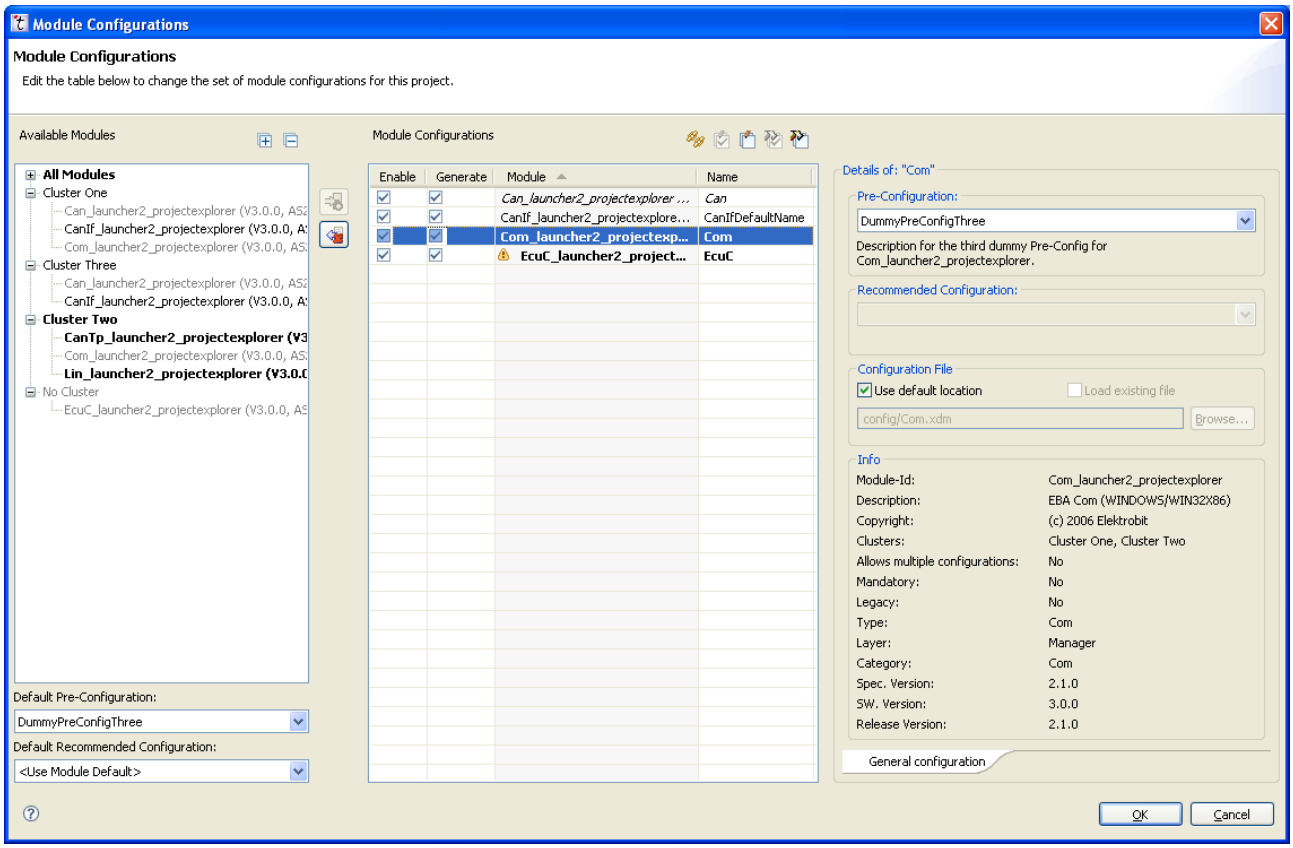


Figure 13.8. The **Module Configurations** dialog

The following sections provide further details about the new UI features.

13.3.2.1. Module presentation

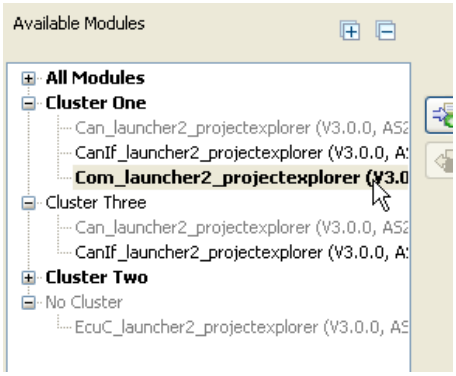


Figure 13.9. The **Available Modules** tree

Available modules are now clearly presented in a tree widget, grouped by their clusters.

Now modules are displayed in different fonts in the **Available Modules** tree so that you can recognize easily which modules and clusters already have been added to your project and which modules are still available.

Modules for which no module configuration has been added to your project are displayed in bold font. Modules for which no (more) configuration can be added are grayed-out.

The **Info** panel on the right side of the **Module Configurations** dialog displays information about the selected module such as the clusters the module belongs to. To open the information in the **Info** panel, select the module of interest in the **Available Modules** tree.

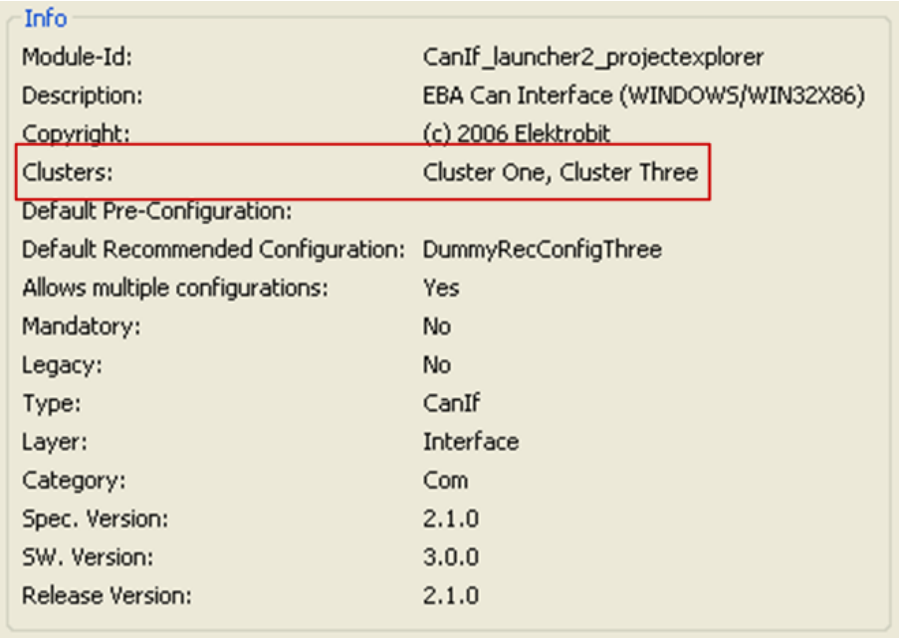


Figure 13.10. Viewing module specific information

13.3.2.2. Adding module configurations to your project

You can now add several module configurations at once to your project. To add module configurations to your project:

- In the **Available Modules** tree, select all modules for which you want to add module configurations to your project.
- Click the **Add** button  .

Module configurations with default settings are added to your project for all selected modules. The module configurations are now displayed in the **Module Configurations** table:

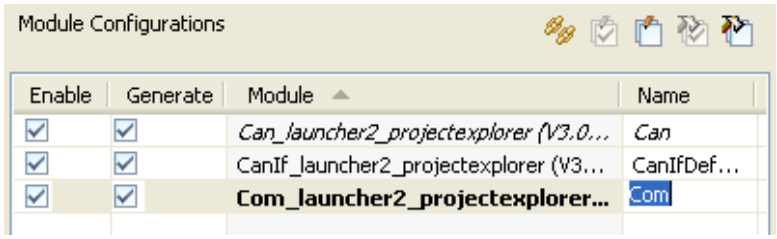


Figure 13.11. The **Module Configurations** table

In the **Module Configurations** table, newly added modules are displayed in a bold font, mandatory ones in italic.

You can modify the default settings of new module configurations on a per module configuration basis, either directly in the **Module Configurations**table or in the **Details** area, which is located at the right side of the **Module Configurations** dialog.

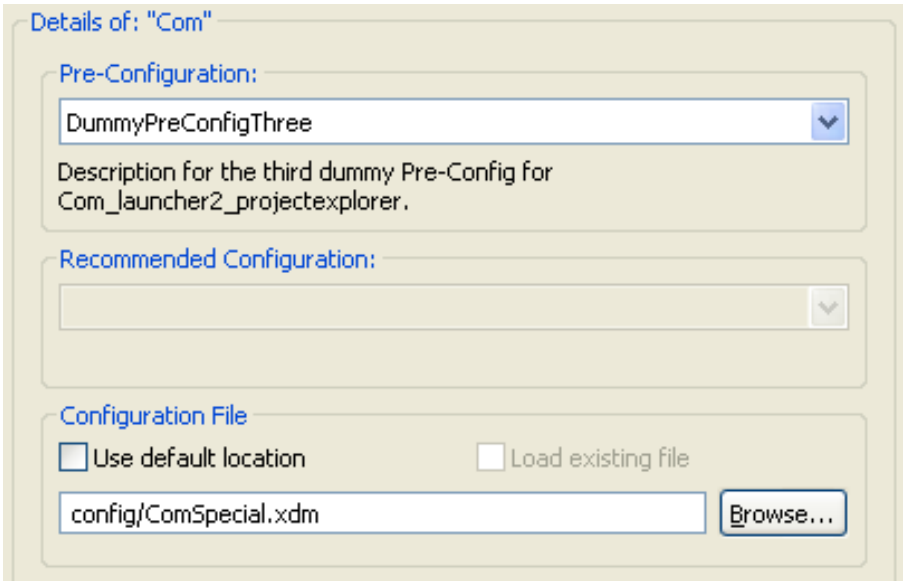


Figure 13.12. Editing the settings of a new module configuration

13.3.2.3. Enabling and disabling module configurations and code generation for module configurations

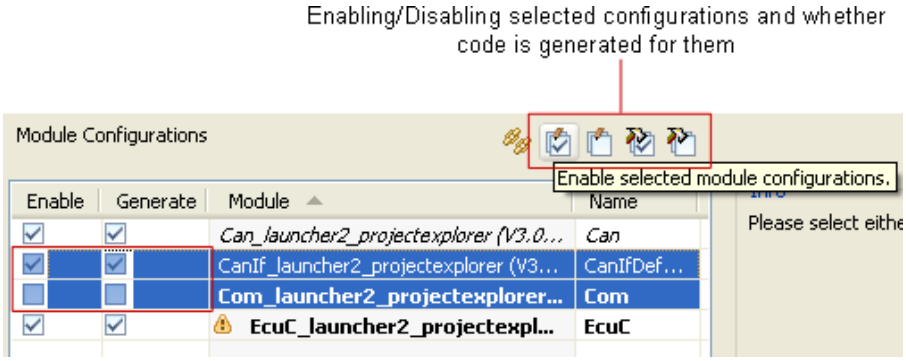


Figure 13.13. Tool bar of the **Module Configurations** table

In the **Module Configurations** table, you can now select or clear **Enable** and **Generate** check boxes for all selected module configurations at once. To select or clear the **Enable** or **Generate** check boxes, use the buttons in the tool bar of the **Module Configurations** table.

Note that the tool bar also contains the former **Upgrade** button.

13.3.2.4. Default preconfigurations and recommended configurations

Modules might provide preconfigurations to be applied or recommended configurations for you to choose from. To edit the preconfigurations and recommended configurations, the **Module Configurations** dialog provides the **Default configuration settings** area:

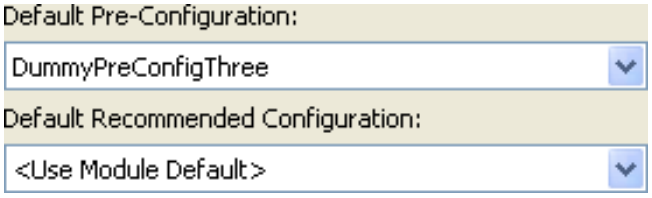


Figure 13.14. Editing the default configuration settings of the module configurations

In the **Default configuration setting** area, you can set the default preconfiguration and recommended configuration which are used as a default values for all new module configurations. To support a more consistent selection, these preconfiguration and recommended settings are persisted. You can also edit the preconfiguration or the recommended configuration of an individual module configuration.

NOTE

The value set is a union set of all preconfigurations and recommended configurations provided by all modules

The configuration names to choose from for both choices derive from the union set of all names provided by all modules that match the project's target architecture and AUTOSAR release version. If the chosen name is not valid for a specific module, the module's default is taken instead when you add it to your project. If the module does not provide such a default, no recommended configuration is applied.

13.4. The Node Outline view can now display parameter labels

You can now control whether the **Node Outline** view displays the parameter names or the parameter labels of the module configuration. The parameter names are identical to the AUTOSAR `SHORT-NAMES`, but modules can provide their own labels instead of the parameter names. The labels are usually more meaningful than the abbreviated `SHORT-NAMES`.

You can change this display setting in the menu of the **Node Outline** view. To change how nodes are displayed in the **Node Outline** view:

- Open the menu of the **Node Outline** view.
- Select **Show**.

A submenu opens up.

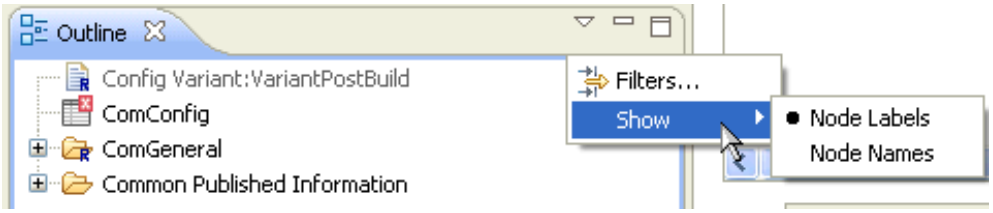


Figure 13.15. Changing display settings in the **Node Outline** view

- Select **Node Labels** or **Node Names**.

Note that the Generic Configuration Editor always displays the parameter labels. Now you can choose whether the **Node Outline** view shall display the labels used in the Generic Configuration Editor or the names used in the AUTOSAR files.

13.5. New XPath function: `num:f`

The XPath function `num:f` is now available. This function converts any number into a string that holds the same number in floating-point format. In contrast to other conversion functions, this function never represents the result in scientific notation.

For example, if you execute `num:f('1.2E3')`, the result will be `'1200.0'` instead of `'1.2E3'`.

13.6. You can now import into `CanTp`, `CanNm`, `FrTp`, `FrNm` configurations

EB tresos Studio 10.0 supports the configuration of AUTOSAR 3.0 and 3.1 `CanTp`, `CanNm`, `FrTp` and `FrNm` modules when importing Fibex 3.0/3.1 and system description files.

14. Changes for release 2010.a

14.1. AUTOSAR 3.1

EB tresos Studio now supports AUTOSAR 3.1.

14.2. New features for the commandline

14.2.1. Improved support for XDM-to-AUTOSAR conversion

In EB tresos Studio product release 2010.a the XDM-to-AUTOSAR conversion has been improved. The following features are now supported:

1. It is possible to use XPath expressions in XDM files to calculate default values, parameter ranges and other aspects of configuration parameters.
   - Using XPath expressions in conjunction with the ECU Resource Manager makes it possible to extract hardware specifics from the parameter definition file and place them into separate property files.
   - Additionally, XPath expressions also make it possible to calculate values that depend on the rest of the configuration.

     While values in the XDM format can be calculated that way, the AUTOSAR formats required fixed values for all aspects of configuration parameters. When you convert XDM parameter definition files to the AUTOSAR format, ensure that values calculated via XPath expressions can be converted into fixed values in the AUTOSAR formats. Not all XPath expressions can be converted into fixed values.
2. With EB tresos Studio 2010.a the functionality to convert XPath expressions from parameter definitions has been enhanced. This makes it possible to convert expressions that:
   - Access the ECU Resource Manager, even if the ECU Resource Manager property files depend on a subderivative, compiler or on other parameters selected in the ECU configuration.
   - Convert XPath expressions that access the ECU configuration by providing a fallback value in case an ECU configuration is not available.
3. The legacy convert commandline has been enhanced: it is now possible to specify all parameters needed for selecting the ECU Resource Manager properties files, which provide hardware-specific values for the XDM-to-AUTOSAR format conversion. (XDM parameter definition files are converted to AUTOSAR format via the legacy convert commandline.) For example, if the property file registration depends on the setting of a *subderivative* parameter, the following parameter specification is now possible:

```
<extension point="dreisoft.tresos.autosar2.api.plugin.ecuresource">
 <ecuresource
   target="x86"
   derivate="808x"
   file="resource/8088.properties">
   <parameter name="subderivative" value="8088"/>
 </ecuresource>
 </extension>
```

You may select the subderivative parameter on the commandline in the following way:

```
tresos_cmd.bat -Dtarget=x86 -Dderivate=808x -Dsubderivative=8088
               legacy convert TSR.xdm TSR.bmd@asc:3.0
```

4. Since EB tresos Studio 2009.a, ECU Resource Manager property files can be bound to modules of a project.

   - To select the modules for conversion, select the modules with the commandline option `EcuResourceModuleIds` in the following way:

     ```
     tresos_cmd.bat -Dtarget=x86 -Dderivate=808x -Dsubderivative=8088
                    -DEcuResourceModuleIds=TSR
                    legacy convert TSR.xdm TSR.bmd@asc:3.0
     ```

     If the option is not set, i.e. you do not select any specific modules, all installed modules are taken into account.

   - When you convert parameter definition files, it is not possible to access configuration values. To allow conversion of XPath expressions that access the configuration, the XPath function `node:fallback()` has been introduced.

     For example, the following expression evaluates to the XPath expression `../@index`, if the configuration can be accessed.:

     ```
     node:fallback( ../@index, 5 )
     ```

     When a parameter definition is converted and therefore the ECU configuration is not accessible, the value `5` is returned.

5. AUTOSAR requires a minimum and a maximum value for all integer and float parameter definitions. To provide this value range, the XDM format uses the `INVALID` attribute of the type `Range`. If you use an `INVALID` attribute of the type `XPath` for an integer or float parameter and therefore an `INVALID` attribute of the type `Range` cannot be used during XDM-to-AUTOSAR conversion,
   - Reasonable defaults are written to the AUTOSAR file; these defaults cover the maximum range of a signed value,
   - and a warning is printed on the commandline.
6. When you convert XDM parameter definition files to the AUTOSAR format, XPath expressions are only evaluated if the commandline option `-DwriteXPathAttributes` is set to `true`. This option `-DwriteXPathAttributes` is set to `true` per default in EB tresos Studio 2010.a. If an XPath expression cannot be evaluated, the XDM-to-AUTOSAR conversion fails with an error.

14.2.2. Importing projects via commandline

To access projects in order to generate code or to modify the ECU configuration, a project must be part of the current workspace. To get a project into the workspace, either create it in the workspace or import it into the workspace.

To support batch processing, the commandline now allows to import projects from the file system into the current workspace.

To import a project into the workspace, use the following command:

```
tresos_cmd.bat [<system_property>...] [-data <workspace>]
               importProject [-c] <project path>...
```

14.2.3. Deleting projects via commandline

To support batch processing the commandline now allows to remove projects from the current workspace.

To remove a project from the workspace, use the following command:

```
tresos_cmd.bat [<system_property>...] [-data <workspace>]
               deleteProject [-d]<project>...
```

14.2.4. Executing importers via commandline

You may configure importers for each project via the GUI. These importers enable importing or exporting configuration data to or from ECU configuration formats and system formats, such as DBC, LDF, etc. You may now execute these importers from the commandline.

Importers can execute on the commandline in

- import mode
- export or import mode
- export mode

.

To execute an importer in import mode use:

```
tresos_cmd.bat [<system_propert<>...] [-data <workspace>]
import <project> <importer>...
```

To execute an importer in export mode use:

```
tresos_cmd.bat [<system_propert<>...] [-data <workspace>]
export <project> <importer>...
```

14.2.5. Conversion of system configuration

The `legacy convert` commandline command now allows to convert system description, tresosDB, DBC, LDF and Fibex files to a tresosDB file, as shown in the following code example:

```
tresos_cmd.bat legacy convert system.arxml@sysd system.dbc system.tdb
```

14.3. New importer features

14.3.1. IpduM support

EB tresos Studio 2010.a supports the configuration of AUTOSAR 3.0 and 2.1 `IpduM` modules via Com import of DBC, Fibex 3.0/3.1, and system description files.

The DBC Importer, the Fibex Importer and the tresosDB Iimporter support the configuraton of multiple static parts and bit range relocation. (Bit range relocation means bit ranges can be shifted between multiplexer PDUs and their related sub-PDUs in order to save RAM).

The System Description Importer does not support bit range relocation and supports just one static part per multiplexed PDU. This limitation is due to shortcomings of the AUTOSAR 3.0 system template.

For a detailed description on how the Com Importer processes multiplexer PDUs, please refer to the EB tresos Studio user's guide.

14.3.2. Instance handling

In EB tresos Studio 2010.a the System Description Importer, the tresosDB Importer and the Fibex Importer generate instances of frames, PDUs and signals in a different way than they did in previous releases: The `FrIf` module is now configured to perform a fan-out and fan-in of frame and PDU instances, thus reducing the number of PDUs (and signals contained therein) in the `PduR` and `Com` modules. For detailed information, please refer to the EB tresos Studio user's guide.

14.3.3. CanTp and CanNM support

The System Description Importer, the tresosDB Importer and the DBC Importer now allow to import ECU configuration parameters for the modules `CanTp` and `CanNM`.

14.4. Usability improvements

14.4.1. Wizard result tree

Wizards such as the **Derivable Values Calculator** or the **Handle ID Calculator** now display a tree with all changed parameters. Clicking on a tree item opens a configuration editor that shows the changed parameter.

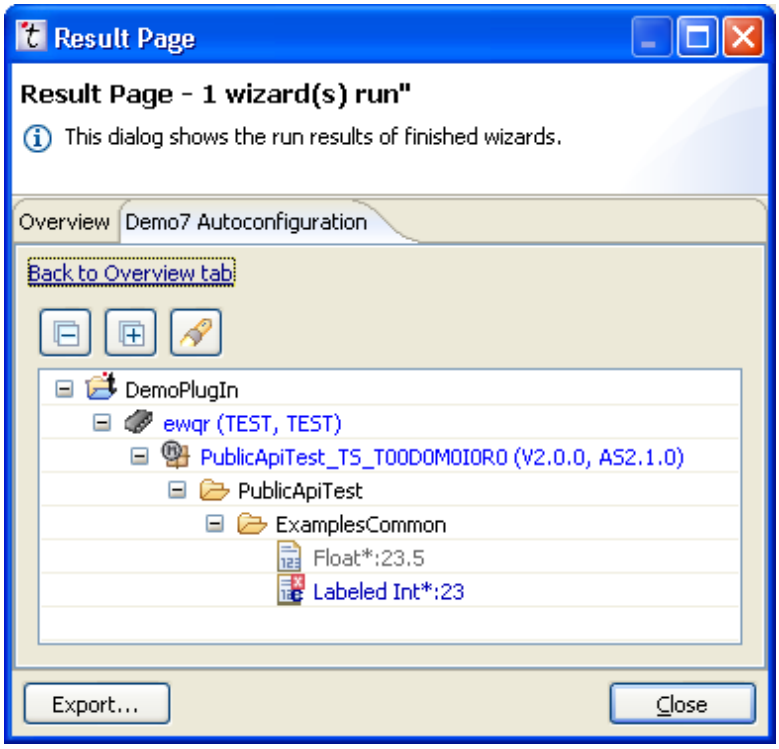


Figure 14.1. Wizard result tree

14.4.2. More information in the Elt. Info view

The **Elt. Info** view, which provides information about the parameter that you are editing, now displays the following additional information:

- For references: path in the parameter definition where referenceable nodes are located.
- For list entries: name and SHORT-NAME.
- For enumaration parameters: valid values.

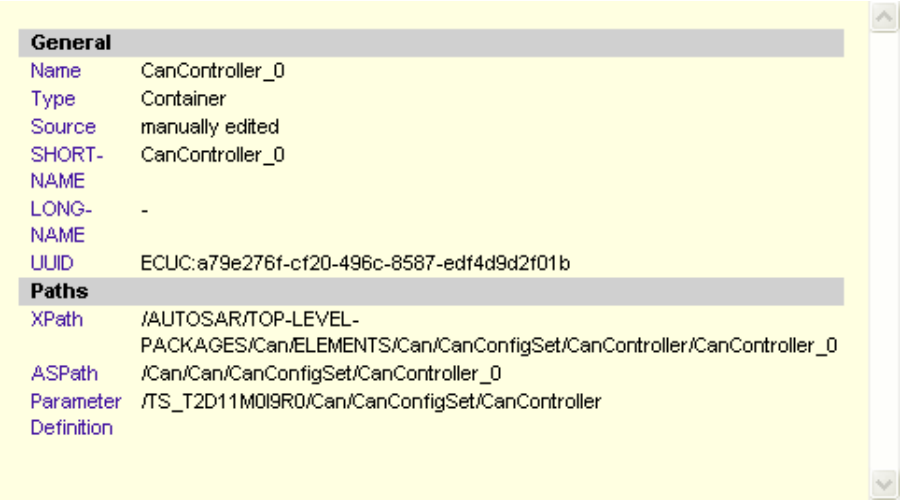


Figure 14.2. Elt. Info view

14.5. Miscellaneous new features

14.5.1. Auto-installing licenses

EB tresos Studio searches for license files in both the `bin` and the `configuration` directories under `$TRESOS_BASE` once at startup and at any time the license information is refreshed manually in the license dialog. It is therefore possible to install a license file by simply copying it to one of these directories. Please read more about this feature in the EB tresos Studio user's guide.

14.5.2. Workspace check on startup

On the first launch of EB tresos Studio you are asked whether to create the default workspace or to select another existing one. The same happens again, whenever the previously used workspace location does not exist anymore when you start EB tresos Studio.

If you start EB tresos Studio while the workspace location is already in use by a running program, you may now select another workspace. Formerly, the startup process exited if the workspace was already being used.

14.5.3. License server via environment variable

If you are using FLEXnet floating network licenses with EB tresos Studio 2010.a, you may now configure the license server via the environment variable `TRESOS_LICENSE_FILE`. It is possible to set environment variables via the **System** tool in the Microsoft Windows Control Panel, as in the following image:

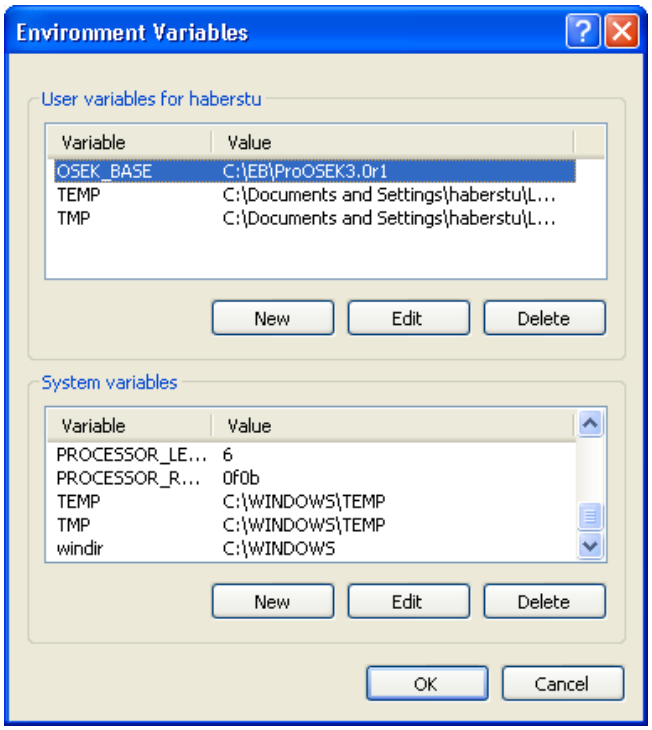


Figure 14.3. Environment variables

The environment variable `TRESOS_LICENSE_FILE` stores the host name of the FLEXnet server.

If you select the license server via the environment variable, you must not configure license settings in the EB tresos Studio preferences. Leave the default settings as in the following image:

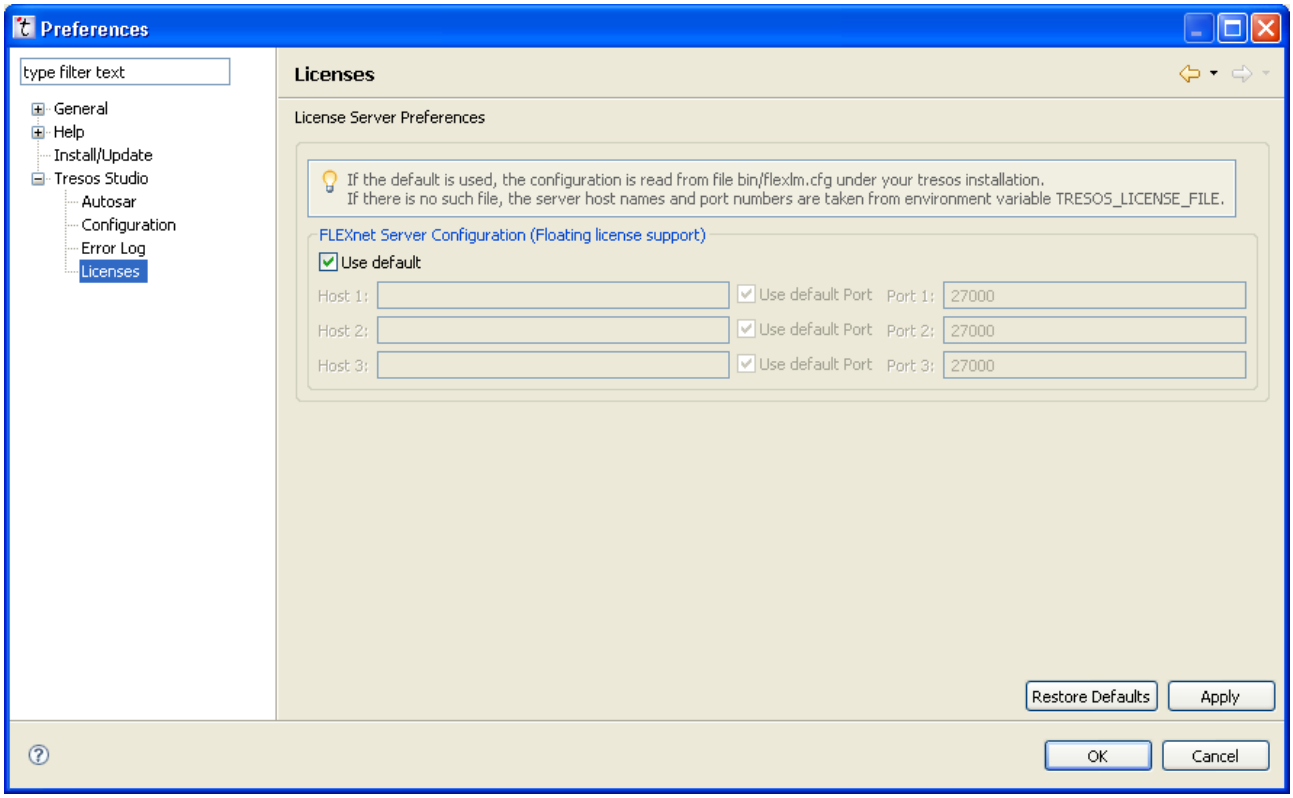


Figure 14.4. Environment variables

14.5.4. Customer-specific default preferences

EB tresos Studio lets you set many preferences such as the maximum number of entries displayed in the **Error Log** view; these entries are stored in the `workspace` folder.

To use the customer-specific preset for these preferences, you need to get an adapted version of the `defaultPreferences.properties` and install it with EB tresos Studio. Install the file in the following directory:

```
$TRESOS_BASE/configuration/defaultPreferences.properties
```

The file is a Java properties file, which defines values for the following preferences:

```
#
# General Preferences
#

# setting to show info labels in dialogs.
```

```
  # valid values: true, false
  showInfoLabels: true

  # setting to enable developer features.
  # valid values: true, false
  enableDeveloperFeatures: false

  # setting to save on exit.
  # valid values: save, nosave, ask
  actionOnExit: ask
```

14.6. New features for developers

14.6.1. Exit code of external generators

The new optional parameter `evalExitCode` has been introduced at the generator extension point. With it you can configure EB tresos Studio to evaluate the exit code of an external generator. In previous releases the exit code of external generators was always ignored.

Valid values for the parameter `evalExitCode` are `true` and `false`; `false` is the default value if the parameter is not specified.

If the parameter `evalExitCode` is `true`, EB tresos Studio evaluates the exit code of the external generator after the external generator finishes. Any exit code different from `0` (zero) is interpreted as abnormal termination of the external generator. In this case an error is generated.

For the definition of the new optional parameter, refer to the code snippet of a `plugin.xml`:

```
  <extension point="dreisoft.tresos.launcher2.plugin.generator"
             id="Generator_Det_TS_T16D4M2I0R0"
             modes="myMode"
             name="Det TemplateGenerator">

    <generator id="ExtGenDet"
               moduleId="Det_TS_T16D4M2I0R0"
               class="dreisoft.tresos.autosar2.generator.ExternalGenerator">
      <parameter name="commandline"
                 value="det.exe -f ${configFiles} -o ${outputDir}"/>
...
      <parameter name="evalExitCode" value="true"/>
```

```
    </generator>
  </extension>
```

14.6.2. New XPath functions

The following new XPath functions are available now:

- `boolean node:contains( node-set, node )`

  Returns `true`, if the given node-set contains the given node.

- `boolean node:containsValue( node-set, node )`

  Returns `true`, if at least one of the given node have the given value.

- `int node:pos( node )`

  Returns the position of the given node within its parent (starting with `0`). Returns `-1` if the given node is invalid.

- `node node:when( expression, [trueValue[, falseValue]] )`

  Depending on the boolean result of the given expression, the given `trueValue` or `falseValue` is returned. The default trueValue is boolean `true`. The default falseValue is boolean `false`.

- `node-set node:foreach( node-set, varName, expression )`

  Executes the given expression once for each element of the given node-set. The value of the current node is stored in the variable with the given name. A list is returned that contains the result of the expression for each loop.

- `node-set node:range( node )`

  Returns the possible values of the given node, if the values are available as a fixed list (as is the case e.g. for enumerations).

- `boolean text:contains( node-set, value )`

  If the given list of elements contains the given value, the function returns `true`.

- `boolean text:contains( string, value )`

  If the given string contains the given value, the function returns `true`.

- `int text:indexOf( string, value )`

  Returns the position (starting with `0`) of the given value within the given string. If the given value is not contained, the function returns `-1`.

- `int text:lastIndexOf( string, value )`

Returns the position (starting with `0`) of the last occurence of the given value within the given string. If the given value is not contained, `-1` is returned.

- `number num:min( node-set )`

  Returns the smallest value of the given nodes

- `number num:max( node-set )`

  Returns the largest value of the given nodes.

15. Changes for release 2009.a-Final

15.1. Fibex Importer

EB tresos Studio now includes an improved Fibex Importer. This importer is able to import configurations from additional Fibex formats.

The following Fibex formats are now supported:

- 1.1.5a
- 1.2.0a
- 2.0.0b
- 2.0.0d
- 2.0.1
- 3.0.0 (new)
- 3.1.0 (new)

You are able to import data into the following ECU configuration modules:

- `Com` (`Communication`)
- `PduR` (`PDU Router`)
- `CanIf` (`CAN Interface`)
- `Can` (`CAN Driver`)
- `FrIf` (`FlexRay Interface`)
- `Fr` (`FlexRay Driver`)
- `EcuC` (`EcuC Module`)

For details, please refer to the EB tresos Studio documentation, chapter EB tresos Studio user's guide/Importing and exporting.

15.2. Autosar 3.0 Lin/LinIf support in the Com Importers

Autosar 3.0 `Lin` and `LinIf` configurations can now be imported from LDF, TDB, and Autosar system description files. For details, please refer to the EB tresos Studio documentation.

15.3. Extensions to the Guided Configuration API

The Guided Configuration API has been extended and improved. The following description focusses on the major changes, not on the small features that have been added and existing classes that have been changed. You can find more information in the EB tresos Studio documentation, chapter EB tresos Studio developer's guide/Guided Configuration API and in the Javadoc. The new methods and classes can be recognized by their *Since 2009.a* comment in the Javadoc. The Guided Configuration chapter in the EB tresos Studio developer's guide has been rewritten and updated with the new API.

The major changes include:

15.3.1. New GUI widgets

Tree table widgets are now avaible. As all standard widgets, this widget can be instantiated by using the `WidgetFactory`. The widget is similar to a table. Unlike the table widget the first column of the tree table consists of a tree. This makes the widget suitable for representing hierarchical data.

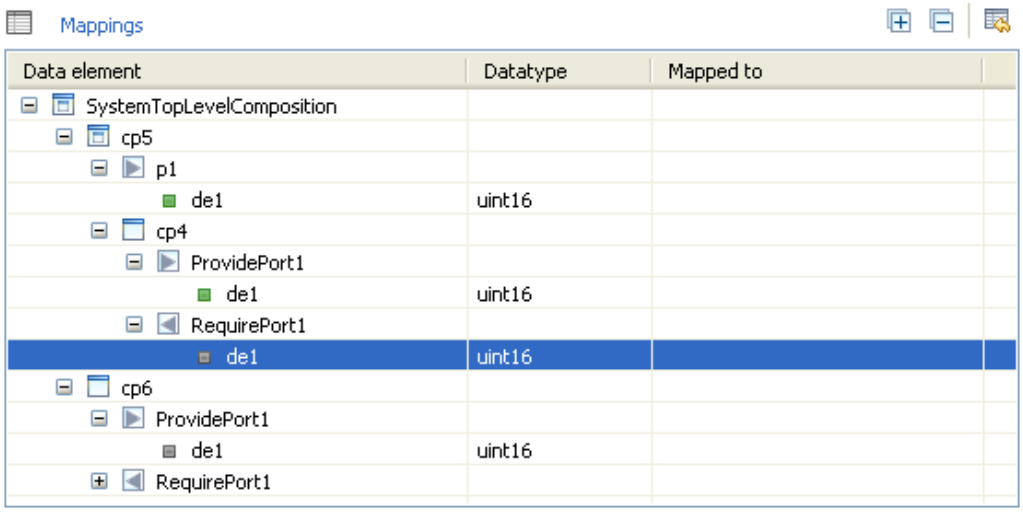


Figure 15.1. Guided configuration tree table

Button widgets are now supported. As all standard widgets, this widget can be instantiated by using the `WidgetFactory`. You can choose between a push button and a toggle button. You can register a selection changed listener to execute actions when the button is pressed.

Figure 15.2. Guided configuration button

Browser widgets are now available. The browser widget can be used to display websites or read-only text areas. As all standard widgets, this widget can be instantiated by using the `WidgetFactory`.

Figure 15.3. Guided configuration browser

15.3.2. GUI descriptions via XML

Instead of creating a subclass of `AbstractPage` and writing Java code, it is now possible to describe the appearance of a page by using XML code. This API is called *XForms*.

For example, you can create a simple toggle button with an icon with the following XML snippet:

```
<Button widgetId="button" icon="icons/sample.gif" flag="TOGGLE"/>
```

15.3.3. Improved Mementos

By calling the method `Memento.lock()` you are now able to lock Mementos. This is useful if you access a Memento from different threads concurrently. Release the lock on a Memento by calling the `Memento.unlock()` method.

You can now group several changes to a single Memento into a transaction. This means you may execute several changes without any events being sent. Furthermore, it is possbile to undo all changes from a transaction with a single undo call. To start a transaction, use `MementoOperationHandler.beginTransaction()`. To save the transaction, use `MementoOperationHandler.commitTransaction()`. To abort a transaction, use `MementoOperationHandler.rollbackTransaction()`.

15.3.4. Improvements to existing widgets

The controller classes for the standard widgets are now part of the Public API. The controllers connect the GUI widgets to the data represented. The base class for all controllers is the class `AbstractWidgetController`. Please refer to the Javadoc of this class for more information.

You can also use the controllers to get notified of widget selection events. If you want to add a selection listener to a widget, call the method `AbstractWidgetController.addSelectionEventListener()` on the widget's controller.

15.3.5. Incompatible API changes

There have been incompatible changes of the Public API between EB tresos Studio 2008.b and 2009.a:

The `ECUConfigContext` API class in the Java package `dreisoft.tresos.guidedconfig.api.context` changed between EB tresos Studio 2008.b and EB tresos Studio 2009.a. The attribute `relVersion` now returns an instance of the Java class `ProjectVersion` instead of `ModuleVersion`. Furthermore, the attribute `generationPath` now returns an instance of the class `java.io.File` instead of `java.lang.String`.

In the `plugin.xml` file the equations that control the visibility of wizards changed between EB tresos Studio 2008.b and EB tresos Studio 2009.a. The changes concern how to make wizards visible in the **Sidebar** view and the **Autoconfigure wizard** submenu in the main window toolbar. To make a wizard visible only if a project has an ECU configuration, set the property `ECUConfigContext` to `true` in the visibility equation of the extension-point of the trigger that registers the wizard.

16. Changes for release 2009.a-Beta.1

16.1. System Description Importer

EB tresos Studio now provides an importer that allows to import Autosar system description and software component description files.

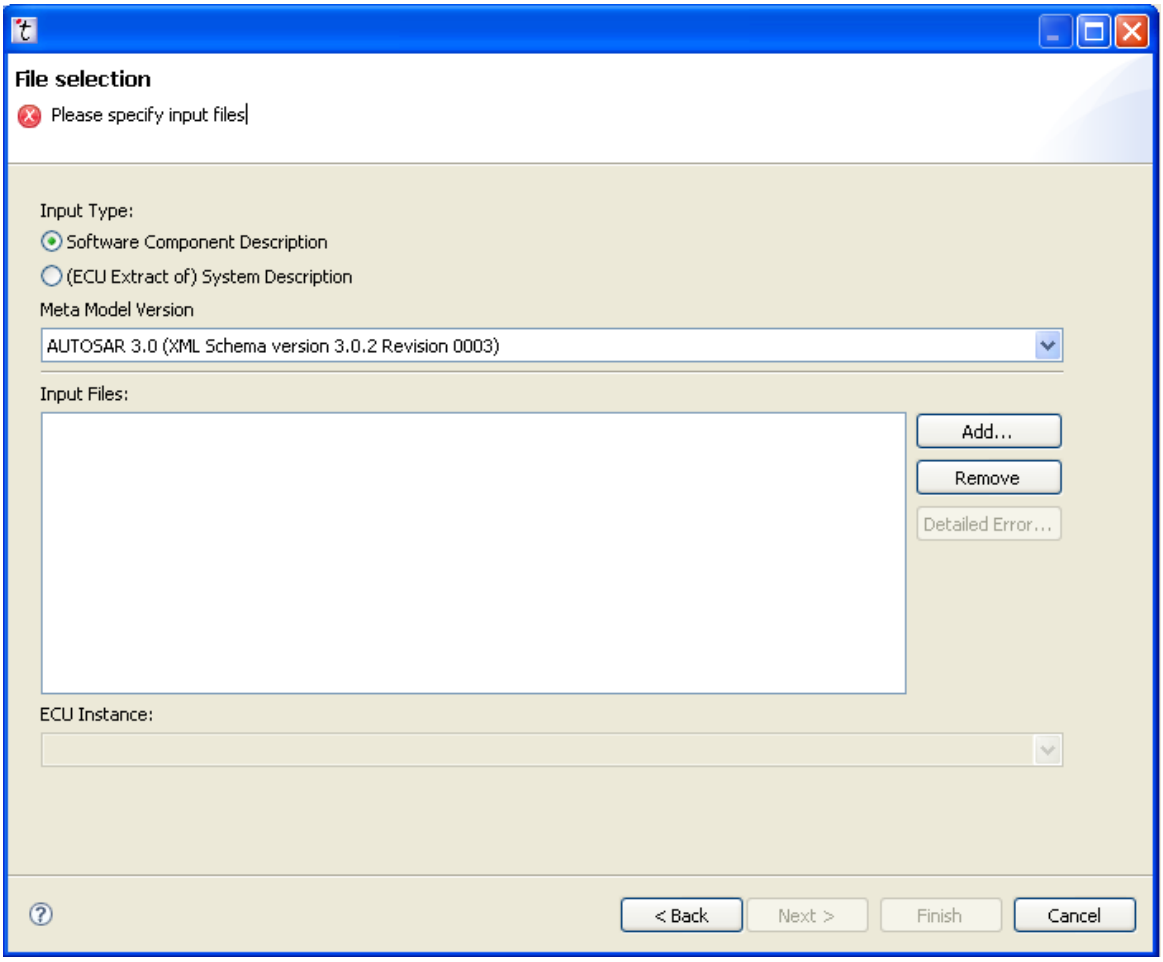


Figure 16.1. The file selection page of the system description importer

System information can be imported either into the system model of the project or into the ECU configuration.

The following ECU configuration modules can be configured by the importer:

- `Com` (`Communication`)
- `PduR` (`PDU Router`)

- `CanIf` (`Can Interface`)
- `Can` (`CAN Driver`)
- `FrIf` (`FlexRay Interface`)
- `Fr` (`FlexRay Driver`)
- `EcuC` (`EcuC Module`)

Supported Autosar meta-model formats are:

- 2.1.3 (Revision 017)
- 2.1.4 (Revision 018)
- 3.0.2 (Revision 003)
- 3.1.0 (Revision 001)

The system description importer can be called from the **Create, manage and run im- and exporters** dialog window. Please refer to the importer chapter of the EB tresos Studio user's guide for details.

16.2. System support in the Com Importers

The FIBEX, DBC and LDF importers have been extended to support the import of system information. Therefore, the GUI has been changed. There is now an *ECU* and a *System* page available. The import of ECU configuration data and system data can be turned on and off separately.

The importer chapters of the EB tresos Studio user's guide have been updated. Please refer to these chapters for more information.

17. Changes for release 2009.a-Alpha.2

No relevant new features.

18. Changes for release 2009.a-Alpha.1

18.1. Profiler for template-based code generators

A basic profiling-mechanism is now available for the template-based code generator. It can be enabled via the preferences or the commandline option "-DenableProfiler=true".

If enabled, a profiler-file is created beneath each generated file. Its name is the same as the generated file with the additional extension ".profile. It contains runtime-informations for each executed command to easily find the command(s) which causes the performance problem.

Example:

```
Sum for this file:
82.780ms
99.79%  (of complete generation process)

LineNumber: number of executions, overall time in percent, overall time
-------------------------------------------------------------------------

1380     : 264192x 24.48% 20.307ms (VAR|DEF)
1402     : 266256x 22.32% 18.522ms (VAR|DEF)
1345-1347:  73728x  5.82%  4.832ms (IF)
1315-1321:  73728x  5.46%  4.535ms (IF)
1465     :   6192x  5.38%  4.468ms (VAR|DEF)
```

18.2. Keyboard support within tables in the generic configuration editor

The generic configuration editor now supports keyboard shortcuts within table controls. These shortcuts are enabled if the table control has the focus and is enabled.

Key	Description
Ins	Adds a single new row at the end of the table.
Del	Removes the currently selected rows.
Ctrl-A	Selects all data in the table.

Key	Description
F2	Opens the cell editor of the first editable column of the currently selected row. Normally, this will be the name cell.
Spacebar	Steps into the currently selected row and opens the detail editor page for this row.

Table 18.1. Keyboard shortcuts within tables

18.3. New command line option "preserveConfiguration"

A new command line option `preserveConfiguration` has been introduced for command `legacy convert`. If it's set to `true`, then the imported configuration won't be changed by adding missing configuration parameters or default values. This option is mainly intended to be used when converting a pre-configuration file where the configured values may not be changed.

Usage: `-DpreserveConfiguration=(true|false)`