



Discover the Experience

EB tresos[®] Studio user's guide

product release 14.2





EB tresos® Studio user's guide

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

Phone: +49 9131 7701 6060

Japan

Phone: +81 3 5577 6110

USA

Phone: +1 888 346 3813

Support URL

[http://automotive.elektrobit.com/
support](http://automotive.elektrobit.com/support)

Legal notice

Confidential and proprietary information

ALL RIGHTS RESERVED. No part of this publication may be copied in any form, by photocopy, microfilm, retrieval system, or by any other means now known or hereafter invented without the prior written permission of Elektrobit Automotive GmbH.

ProOSEK®, tresos®, and street director® are registered trademarks of Elektrobit Automotive GmbH.

All brand names, trademarks and registered trademarks are property of their rightful owners and are used only for description.

© 2013 Elektrobit Automotive GmbH



Table of Contents

Begin here	25
1. If you are upgrading from a previous version	25
2. If you are using EB tresos Studio for the first time	25
3. If you want to configure AUTOSAR modules or your own modules with EB tresos Studio	26
4. If you want to extend the EB tresos Studio functionality	26
5. If you need help/more information	26
1. About this documentation	28
1.1. Introduction	28
1.2. Required knowledge and system environment	28
1.2.1. Required system environment	29
1.3. Interpretation of version information	29
1.3.1. Product version number	29
1.3.2. Qualifiers for basic software	30
1.4. Typography and style conventions	30
1.5. Naming conventions	32
1.5.1. AUTOSAR XML schema	32
1.5.2. Configuration parameter names	33
1.6. EB tresos Studio naming conventions	33
1.6.1. EB tresos Studio command shell	33
2. Safe and correct use of EB tresos Studio	34
2.1. Intended usage of EB tresos Studio	34
2.2. Possible misuse of EB tresos Studio	34
2.3. Target group and required knowledge	34
2.4. Quality standards compliance	34
3. EB tresos product line support	35
4. Background information	36
4.1. Overview	36
4.2. The AUTOSAR concept	36
4.3. What is EB tresos Studio?	37
4.3.1. The configuration and generation tool	37
4.3.2. Usage	38
4.3.3. Benefits	39
4.3.4. Features	39
4.4. File and directory structure	41
4.5. EB tresos Studio projects and workspaces	42
4.5.1. The Workspace	42
4.5.2. Projects	43
4.5.2.1. Configuration time of projects	43
4.5.3. File content types	44



4.5.3.1. Content types for XDM files	45
4.5.3.2. Content types for AUTOSAR module configuration files	45
4.5.3.3. Content types for AUTOSAR system description files	46
4.5.3.4. Content types for tresosDB files	47
4.5.3.5. Content types for DBC files	47
4.5.3.6. Content types for LDF files	48
4.5.3.7. Content types for Fibex files	48
4.6. 64 bit support	48
5. Using EB tresos Studio for the first time	49
5.1. Starting the graphical user interface	49
5.2. Selecting the workspace location	50
5.3. The welcome page	51
6. The graphical user interface (GUI)	54
6.1. Using the Workbench	54
6.1.1. Customizing the workbench	56
6.1.1.1. Moving views to a different location	56
6.1.1.2. Closing views, and minimizing or maximizing view stacks	57
6.1.1.3. Opening a view	58
6.1.2. Using perspectives	59
6.1.2.1. Saving user-defined perspectives	60
6.1.2.2. Opening perspectives	61
6.1.2.3. Switching between perspectives	62
6.1.2.4. Closing perspectives	62
6.1.2.5. Resetting perspectives	62
6.1.2.6. Deleting perspectives	63
6.2. Menu bar and tool bar	64
6.2.1. File menu	65
6.2.2. Edit menu	67
6.2.3. Search menu	67
6.2.4. Project menu	68
6.2.5. Window menu	69
6.2.6. Help menu	70
6.3. Views	72
6.3.1. Project Explorer view	72
6.3.1.1. Project and ECU configuration states	73
6.3.1.2. Layering and grouping of module configurations	74
6.3.1.3. Changing the visible layers	74
6.3.1.4. Collapsing the Project Explorer tree	79
6.3.1.5. Synchronizing the Project Explorer with an editor	80
6.3.1.6. Indicating different project states with colors	81
6.3.2. The Editor view	81
6.3.3. Node Outline view	84



6.3.3.1. Filtering the displayed items	84
6.3.3.2. Appearance of the displayed items	86
6.3.4. Element Outline views	86
6.3.4.1. Element Errors view	86
6.3.4.2. Element Information view	87
6.3.4.3. Element Description view	89
6.3.5. Error Log view	90
6.3.5.1. Changing the display settings of the Error Log view	92
6.3.5.2. Filtering Error Log messages by their severity level	92
6.3.5.3. Restricting the amount of shown Error Log messages and the backup logfile size	93
6.3.5.4. Copying Error Log entries into text documents	94
6.3.6. Problems view	94
6.3.6.1. Changing the display settings of the Problems View	96
6.3.6.2. Export the messages of the Problems View	97
6.3.6.3. Filtering Problems view messages by their severity level	97
6.3.6.4. Limit the number of entries shown in the Problems View	97
6.3.7. Workflows view	98
6.3.7.1. Selecting a workflow	100
6.3.7.2. Performing steps of a workflow	101
6.3.8. Sidebar view	102
6.4. Editors	103
6.4.1. The Generic Configuration Editor	103
6.4.1.1. Sections and parameters	103
6.4.1.2. References	108
6.4.1.3. Choices	110
6.4.1.4. Tables	111
6.4.1.5. Optional elements	114
6.5. Keyboard shortcuts	115
6.5.1. Shortcuts for the Workbench	115
6.5.2. Shortcuts for views	115
6.5.3. Shortcuts for editors and projects	115
6.5.4. Shortcuts for tables	116
7. Use cases	117
7.1. Overview	117
7.2. Changing the EB tresos Studio setup	118
7.2.1. Managing licenses	118
7.2.1.1. Adding a license	121
7.2.1.2. Deleting a license	122
7.2.1.3. Refreshing license information	122
7.2.1.4. Auto-installing licenses	122
7.2.2. Viewing setup information	123



7.2.2.1. Opening the About EB tresos dialog	123
7.2.2.2. Finding program startup parameters and system property settings	124
7.2.2.3. Opening the <i>EB tresos Details</i> dialog	125
7.2.2.4. Finding the location of the installation and the workspace	126
7.2.2.5. Finding all installed modules	126
7.2.2.6. Finding all installed plug-ins	127
7.2.2.7. Exporting a system report of your EB tresos Studio installation	128
7.2.3. Setting preferences	128
7.2.3.1. Setting general preferences	130
7.2.3.2. Setting Assistance dialog preferences	132
7.2.3.3. Setting preferences for parameter comments	132
7.2.3.4. Setting AUTOSAR preferences	133
7.2.3.5. Setting configuration preferences	134
7.2.3.6. Setting Error Log preferences	138
7.2.3.7. Setting License Server preferences	139
7.2.3.8. Setting colors and fonts	140
7.2.4. Working with multiple workspaces to store different projects and settings	143
7.2.4.1. Switching the workspace	144
7.2.4.2. Finding out which workspace directory is used	144
7.2.5. Placing an EB tresos Studio installation under version control	145
7.2.6. Adding new features and extensions to EB tresos Studio via external folders	145
7.3. Working with projects	146
7.3.1. Creating a new project	146
7.3.1.1. Step 1: Starting the Project Wizard	146
7.3.1.2. Step 2: Creating a new project	148
7.3.1.3. Step 3: Configuring project properties	150
7.3.1.4. Step 4: Defining module configurations	153
7.3.1.4.1. Adding module configurations to your project	154
7.3.1.4.2. Removing module configurations from your project	155
7.3.1.4.3. Enabling and disabling module configurations	157
7.3.1.4.4. Enabling and disabling code generation for module configurations	158
7.3.1.4.5. Naming <i>Module Configurations</i>	160
7.3.1.4.6. Choosing <i>Pre-Configurations</i>	161
7.3.1.4.7. Applying recommended configurations	164
7.3.1.4.8. Editing the configuration file settings	166
7.3.1.4.9. Upgrading module configurations	168
7.3.1.5. Step 5: Importing data into the project (optional)	169
7.3.1.6. Step 6: Finishing set-up of the new project	170
7.3.2. Saving a project	170
7.3.3. Renaming a project	171
7.3.4. Copying a project	171
7.3.5. Deleting a project	172



7.3.6. Closing and opening projects	172
7.3.6.1. Closing projects	172
7.3.6.2. Opening a project	175
7.3.6.3. Loading a project	175
7.3.6.4. Reloading a project	175
7.3.6.4.1. Reloading projects automatically	175
7.3.6.4.2. Reloading projects manually	176
7.3.7. Viewing and changing project properties	176
7.3.7.1. Listing the supported models of the project	178
7.3.7.2. Editing the Code Generator settings	179
7.3.7.3. Editing the ECU Configuration settings	181
7.3.7.4. Viewing information about the project's modules	182
7.3.7.4.1. Adding module configurations to your project	182
7.3.7.4.2. Removing module configurations from your project	184
7.3.7.4.3. Enabling and disabling module configurations	186
7.3.7.4.4. Enabling and disabling code generation for module configurations	187
7.3.7.4.5. Naming <i>Module Configurations</i>	189
7.3.7.4.6. Choosing <i>Pre-Configurations</i>	190
7.3.7.4.7. Applying recommended configurations	193
7.3.7.4.8. Editing the configuration file settings	195
7.3.7.4.9. Upgrading module configurations	197
7.4. Using team collaboration support	198
7.4.1. Overview	198
7.4.2. Working with CVS repositories	198
7.4.2.1. Registering your CVS repository location in EB tresos Studio	199
7.4.2.2. Sharing an EB tresos Studio project with other team members	201
7.4.2.3. Checking out a shared project from your CVS repository into your EB tresos Studio workspace	205
7.4.2.4. Updating, branching, and committing EB tresos Studio projects into your CVS repository	207
7.4.3. Working with SVN repositories	208
7.4.3.1. Registering your SVN repository location in EB tresos Studio	209
7.4.3.2. Sharing an EB tresos Studio project with other team members	210
7.4.3.3. Checking out a shared project from your SVN repository into your EB tresos Studio workspace	215
7.4.3.4. Updating, branching, and committing EB tresos Studio projects into your SVN repository	217
7.4.4. Synchronizing your project with your repository	220
7.4.4.1. Overview	220
7.4.4.2. Background information	220
7.4.4.2.1. The <i>Team Synchronizing</i> perspective	220
7.4.4.2.2. The <i>Synchronize</i> view	221



7.4.4.3. Opening the <i>Team Synchronizing</i> perspective	224
7.4.4.4. Viewing differences between your working copy and the base file in the repository	226
7.4.4.5. Synchronizing your working copy with changes in the repository	226
7.5. Differing files and file versions	228
7.5.1. Overview	228
7.5.2. Differing XDM files	228
7.5.2.1. Overview	229
7.5.2.2. Background information	229
7.5.2.3. Opening the compare editor	229
7.5.2.3.1. The <i>Compare</i> editor for XDM files	232
7.5.2.3.2. Tree table icons and overlay icons	234
7.5.2.4. Understanding the differing results	235
7.5.2.5. Navigating from difference to difference	236
7.5.2.5.1. Filtering	236
7.5.2.5.2. Expanding and collapsing the tree	236
7.5.2.5.3. Keyboard shortcuts	237
7.5.2.6. Merging differences	237
7.5.2.6.1. Applying differences	238
7.5.2.6.2. Rejecting changes	239
7.5.2.6.3. Undoing/Redoing operations	241
7.5.2.7. Limitations	241
7.5.2.7.1. Ignored differences	241
7.5.2.7.2. Schema support	241
7.6. Editing module configurations	241
7.6.1. Adding module configurations to your project	244
7.6.2. Removing module configurations from your project	245
7.6.3. Enabling and disabling module configurations	247
7.6.4. Enabling and disabling code generation for module configurations	251
7.6.5. Naming <i>Module Configurations</i>	253
7.6.6. Choosing <i>Pre-Configurations</i>	254
7.6.6.1. Setting the <i>Default Pre-Configuration</i>	254
7.6.6.2. Changing the preconfiguration of one specific module configuration	255
7.6.7. Applying recommended configurations	257
7.6.7.1. Setting the <i>Default Recommended Configuration</i>	257
7.6.7.2. Changing the recommended configuration of one specific module configuration	258
7.6.8. Editing the configuration file settings	260
7.6.8.1. Using existing configuration files	261
7.6.9. Upgrading module configurations	262
7.7. Upgrading projects and configurations	262
7.7.1. Converting a project to a new AUTOSAR version	263



7.7.2. Upgrading module configurations to new module versions	265
7.8. Editing parameters of a module configuration	267
7.8.1. Overview	267
7.8.2. Background information	268
7.8.3. Opening the ECU Configuration Editor	269
7.8.4. Setting the configuration variant of the module	270
7.8.5. Adding parameter comments	271
7.8.5.1. Adding and editing a parameter comment	272
7.8.5.2. Managing languages of parameter comments	274
7.8.5.3. Removing parameter comments	275
7.8.5.4. Generating a parameter comment report	276
7.8.5.4.1. Reading the parameter comment report	277
7.8.5.4.2. Customizing the DemoReportGenerator	278
7.8.6. Editing several table rows simultaneously	278
7.8.7. Autoconfiguring routine jobs	280
7.8.7.1. Configuring unattended wizards	280
7.8.7.2. Running unattended wizards	282
7.8.7.2.1. Running a single unattended wizard	282
7.8.7.2.2. Running multiple unattended wizards	284
7.8.7.2.3. Result feedback	286
7.8.7.3. Configuring the Calculate Derivable Values wizard	287
7.8.7.4. Configuring the Calculate Handle IDs wizard	288
7.8.8. Searching for configuration parameter names and their values	289
7.8.8.1. Setting search options	291
7.8.8.1.1. Defining general search options	292
7.8.8.1.2. Defining which kind of text is searched	292
7.8.8.1.3. Limiting the kind of parameters that are searched	293
7.8.8.1.4. Defining the scope of the search	294
7.8.8.2. Displaying search results	295
7.9. Generating code for projects	296
7.9.1. Verifying a project	296
7.9.2. Generating a project	298
7.9.3. Building a project	299
7.10. Importing and exporting	300
7.10.1. Exchanging a complete EB tresos Studio project with other installations	301
7.10.1.1. Importing a project	301
7.10.1.2. Exporting a project	305
7.10.1.2.1. Exporting a project as copy	306
7.10.1.2.2. Exporting a project as archive	307
7.10.2. Importing and exporting configuration data	308
7.10.2.1. Step-by-step guide	308
7.10.2.1.1. Step 1: Using the Create, manage and run im- and exporters window.	308



7.10.2.1.2. Step 2: Creating an export or import configuration	311
7.10.2.1.3. Step 3: Saving changes to an import or export configuration	312
7.10.2.1.4. Step 4: Importing or exporting project configuration data	313
7.10.2.2. Importing and exporting configuration data from and to the AUTOSAR format	314
7.10.2.2.1. How the import works	315
7.10.2.2.2. Importing and exporting configuration data	315
7.10.2.3. Importing AUTOSAR system descriptions	320
7.10.2.3.1. Overview	320
7.10.2.3.2. Background information	320
7.10.2.3.3. Importing system information	321
7.10.2.3.4. References	330
7.10.2.3.5. Variants tab	330
7.10.2.4. Exporting AUTOSAR system description	331
7.10.2.4.1. Overview	331
7.10.2.4.2. Background information	331
7.10.2.4.3. Exporting system information	332
7.10.2.5. Importing from the OIL format	335
7.10.2.5.1. Overview	335
7.10.2.5.2. OIL Importer and file format	335
7.10.2.5.3. Importing OIL configurations	335
7.10.2.5.4. Importing OIL files via the command line	337
7.10.2.6. Importing CAN configurations from the DBC format	338
7.10.2.6.1. Overview	338
7.10.2.6.2. Background information	339
7.10.2.6.3. Importing DBC configurations	340
7.10.2.6.4. References	345
7.10.2.7. Importing from the LDF format	359
7.10.2.7.1. Overview	359
7.10.2.7.2. LDF Importer and file format	359
7.10.2.7.3. Importing LDF configurations	359
7.10.2.7.4. References	364
7.10.2.8. Importing from the Fibex format	364
7.10.2.8.1. Overview	364
7.10.2.8.2. Background information	365
7.10.2.8.3. Importing Fibex configurations	365
7.10.2.9. Importing configurations from the TDB format	370
7.10.2.9.1. Overview	370
7.10.2.9.2. Background information	370
7.10.2.9.3. Importing TDB configurations	371
7.10.2.9.4. Mapping TDB parameters to AUTOSAR parameters	376
7.11. Working with the command line	376



7.11.1. General	377
7.11.1.1. Starting the commandline	377
7.11.1.2. The commandline syntax	377
7.11.1.3. System properties common to all commands	378
7.11.1.4. Specifying the workspace on the commandline	379
7.11.2. Commands for project-based code generation and configuration	380
7.11.2.1. Importing projects	380
7.11.2.1.1. Syntax	380
7.11.2.1.2. System properties	380
7.11.2.1.3. Parameters	381
7.11.2.1.4. Example	381
7.11.2.2. Deleting projects	381
7.11.2.2.1. Syntax	381
7.11.2.2.2. System properties	381
7.11.2.2.3. Parameters	382
7.11.2.2.4. Example	382
7.11.2.3. Executing importers configured for a project	382
7.11.2.3.1. Syntax	382
7.11.2.3.2. System properties	383
7.11.2.3.3. Parameters	383
7.11.2.3.4. Example	383
7.11.2.4. Listing importers configured for a project	383
7.11.2.4.1. Syntax	384
7.11.2.4.2. System properties	384
7.11.2.4.3. System properties	384
7.11.2.4.4. Parameters	384
7.11.2.4.5. Example	384
7.11.2.5. Verifying a project	384
7.11.2.5.1. Syntax	385
7.11.2.5.2. System properties	385
7.11.2.5.3. Parameters	385
7.11.2.5.4. Example	385
7.11.2.6. Generating code for a project	385
7.11.2.6.1. Syntax	385
7.11.2.6.2. System properties	386
7.11.2.6.3. Options	386
7.11.2.6.4. Parameters	386
7.11.2.6.5. Example	387
7.11.2.7. Listing custom-built generation modes	387
7.11.2.7.1. Syntax	387
7.11.2.7.2. System properties	387
7.11.2.7.3. Parameters	388



7.11.2.7.4. Example	388
7.11.2.8. Executing custom-built generation modes for projects	388
7.11.2.8.1. Syntax	388
7.11.2.8.2. System properties	388
7.11.2.8.3. Parameters	389
7.11.2.8.4. Example	389
7.11.2.9. Executing unattended wizards on projects	390
7.11.2.9.1. Syntax	390
7.11.2.9.2. System properties	390
7.11.2.9.3. Parameters	390
7.11.2.9.4. Example	391
7.11.2.9.5. Running the Calculate Default Values wizard on the command line	391
7.11.2.9.6. Running the Calculate Handle-Ids wizard on the command line	391
7.11.2.10. Listing all unattended wizards for a given project	391
7.11.2.10.1. Syntax	391
7.11.2.10.2. System properties	392
7.11.2.10.3. Parameters	392
7.11.2.10.4. Example	392
7.11.3. Commands for legacy code generation	392
7.11.3.1. Listing all modules available in legacy mode	393
7.11.3.1.1. Syntax	393
7.11.3.1.2. System properties	393
7.11.3.1.3. Example	393
7.11.3.2. Generating code	393
7.11.3.2.1. Syntax	393
7.11.3.2.2. System properties	394
7.11.3.2.3. Options	395
7.11.3.2.4. Parameters	397
7.11.3.2.5. Example	397
7.11.3.3. Verifying configurations	398
7.11.3.3.1. Syntax	398
7.11.3.3.2. System properties	398
7.11.3.3.3. Options	400
7.11.3.3.4. Parameters	401
7.11.3.3.5. Example	401
7.11.3.4. Listing all generation modes	402
7.11.3.4.1. Syntax	402
7.11.3.4.2. System properties	402
7.11.3.4.3. Example	402
7.11.3.5. Running generation modes	403
7.11.3.5.1. Syntax	403
7.11.3.5.2. System properties	403



7.11.3.5.3. Options	404
7.11.3.5.4. Parameters	406
7.11.3.5.5. Example	406
7.11.4. Commands for module development	406
7.11.4.1. Listing a data model	407
7.11.4.1.1. Syntax	407
7.11.4.1.2. System properties	407
7.11.4.1.3. Options	408
7.11.4.1.4. Parameters	408
7.11.4.1.5. Example	409
7.11.4.2. Creating vendor-specific module definitions (VSMD)	409
7.11.4.2.1. Syntax	409
7.11.4.2.2. System properties	410
7.11.4.2.3. Options	413
7.11.4.2.4. Parameters	414
7.11.4.2.5. Example	415
7.11.4.3. Checking vendor-specific against standard module definitions	415
7.11.4.3.1. Syntax	415
7.11.4.3.2. System properties	415
7.11.4.3.3. Parameters	416
7.11.4.3.4. Example	417
7.11.4.4. Listing all rules used for VSMD checking	417
7.11.4.4.1. Syntax	418
7.11.4.4.2. System properties	418
7.11.4.4.3. Example	418
7.11.4.5. Encrypting and signing modules	418
7.11.5. Commands for converting and merging files	418
7.11.5.1. Merging files and converting between different file formats	418
7.11.5.1.1. Syntax	419
7.11.5.1.2. System properties	419
7.11.5.1.3. Options	428
7.11.5.1.4. Parameters	429
7.11.5.1.5. Example	429
7.11.5.2. Using XDM files in different EB tresos Studio versions	430
7.11.5.2.1. Example	430
7.11.5.3. Upgrading the AUTOSAR compliance of XDM files	431
7.11.5.3.1. Syntax	431
7.11.5.3.2. System properties	431
7.11.5.3.3. Parameters	431
7.12. Variant Handling	432
7.12.1. General	432
7.12.1.1. Variant Handling general information	432

7.12.1.2. Resolving variants example: comandline	433
7.12.1.3. Resolving variants example: Im- and Export wizard	433
7.13. Post-build loadable configuration using Multiple Configuration Containers	434
7.13.1. General post-build loadable concept	434
7.13.2. Using Multiple Configuration Containers for post-build loadables	434
7.13.3. Defining valid sets of "Multiple Configuration Containers"	435
7.13.3.1. Generating code	436
7.13.3.2. Importing a system description	436
7.13.3.3. Guided Configuration wizards	436
7.13.4. Value Constraints	436
7.13.4.1. Readonly	436
7.13.4.2. Dependencies between parameters	437
8. References	438
8.1. Upgrading system models	438
8.1.1. Upgrading a system model from AUTOSAR 3.1.x to AUTOSAR 4.0.2	439
8.1.1.1. Topology	439
8.1.1.1.1. SYSTEM	439
8.1.1.1.2. ECU-INSTANCE	440
8.1.1.1.3. CAN-CLUSTER	440
8.1.1.1.4. CAN-COMMUNICATION-CONTROLLER	441
8.1.1.1.5. CAN-COMMUNICATION-CONNECTOR	442
8.1.1.1.6. CAN-PHYSICAL-CHANNEL	442
8.1.1.1.7. FLEXRAY-CLUSTER	443
8.1.1.1.8. FLEXRAY-COMMUNICATION-CONTROLLER	445
8.1.1.1.9. FLEXRAY-COMMUNICATION-CONNECTOR	446
8.1.1.1.10. FLEXRAY-PHYSICAL-CHANNEL	447
8.1.1.1.11. LIN-CLUSTER	448
8.1.1.1.12. LIN-MASTER	448
8.1.1.1.13. LIN-SLAVE	449
8.1.1.1.14. LIN-ERROR-RESPONSE	449
8.1.1.1.15. LIN-COMMUNICATION-CONNECTOR	450
8.1.1.1.16. LIN-PHYSICAL-CHANNEL	451
8.1.1.2. Communication	451
8.1.1.2.1. CAN-FRAME-TRIGGERING	451
8.1.1.2.2. FLEXRAY-FRAME-TRIGGERING	452
8.1.1.2.3. FLEXRAY-ABSOLUTELY-SCHEDULED-TIMING	452
8.1.1.2.4. LIN-FRAME-TRIGGERING	453
8.1.1.2.5. CAN-FRAME	453
8.1.1.2.6. FLEXRAY-FRAME	454
8.1.1.2.7. LIN-UNCONDITIONAL-FRAME	454
8.1.1.2.8. LIN-SPORADIC-FRAME	455
8.1.1.2.9. LIN-EVENT-TRIGGERED-FRAME	456

8.1.1.2.10. LIN-SCHEDULE-TABLE	456
8.1.1.2.11. ASSIGN-FRAME-ID	457
8.1.1.2.12. UNASSIGN-FRAME-ID	457
8.1.1.2.13. ASSIGN-NAD	457
8.1.1.2.14. FREE-FORMAT	458
8.1.1.2.15. PDU-TRIGGERING	458
8.1.1.2.16. I-SIGNAL-TRIGGERING	458
8.1.1.2.17. DCM-I-PDU	459
8.1.1.2.18. I-SIGNAL-I-PDU	459
8.1.1.2.19. MULTIPLEXED-I-PDU	460
8.1.1.2.20. DYNAMIC-PART	461
8.1.1.2.21. DYNAMIC-PART-ALTERNATIVE	461
8.1.1.2.22. STATIC-PART	462
8.1.1.2.23. SEGMENT-POSITION	462
8.1.1.2.24. N-PDU	463
8.1.1.2.25. NM-PDU	463
8.1.1.2.26. PDU-TO-FRAME-MAPPING	464
8.1.1.2.27. I-PDU-TIMING	464
8.1.1.2.28. I-SIGNAL-I-PDU-GROUP	465
8.1.1.2.29. I-SIGNAL-TO-I-PDU-MAPPING	465
8.1.1.2.30. I-SIGNAL	466
8.1.1.2.31. I-SIGNAL-GROUP	467
8.1.1.2.32. SYSTEM-SIGNAL	468
8.1.1.2.33. I-SIGNAL-PORT	468
8.1.1.2.34. FRAME-PORT	469
8.1.1.2.35. I-PDU-PORT	469
8.1.1.2.36. TRANSMISSION-MODE-DECLARATION	469
8.1.1.2.37. TRANSMISSION-MODE-CONDITION	470
8.1.1.2.38. TRANSMISSION-MODE-FALSE-TIMING	470
8.1.1.2.39. EVENT-CONTROLLED-TIMING	470
8.1.1.2.40. CYCLIC-TIMING	471
8.1.1.2.41. TIME-RANGE-TYPE	471
8.1.1.2.42. ABSOLUTE-TOLERANCE	471
8.1.1.2.43. RELATIVE-TOLERANCE	471
8.1.1.3. Gateway	472
8.1.1.3.1. FRAME-MAPPING	472
8.1.1.3.2. I-PDU-MAPPING	472
8.1.1.3.3. TARGET-I-PDU-REF	472
8.1.1.3.4. PDU-MAPPING-DEFAULT-VALUE	472
8.1.1.3.5. DEFAULT-VALUE-ELEMENT	473
8.1.1.3.6. I-SIGNAL-MAPPING	473
8.1.1.4. CAN Transport Layer	473

8.1.1.4.1. CAN-TP-CONFIG	473
8.1.1.4.2. CAN-TP-CONNECTION	474
8.1.1.5. FlexRay Transport Layer	476
8.1.1.5.1. FLEXRAY-TP-CONFIG	476
8.1.1.5.2. FLEXRAY-TP-CONNECTION	477
8.1.1.5.3. FLEXRAY-TP-CONNECTION-CONTROL	478
8.1.1.6. LIN Transport Layer	479
8.1.1.6.1. LIN-TP-CONFIG	479
8.1.1.6.2. LIN-TP-CONNECTION	480
8.1.1.7. Network Management	480
8.1.1.7.1. NM-CONFIG	480
8.1.1.7.2. NM-ECU	481
8.1.1.7.3. CAN-NM-CLUSTER	482
8.1.1.7.4. CAN-NM-NODE	483
8.1.1.7.5. CAN-NM-CLUSTER-COUPLING	484
8.1.1.7.6. FLEXRAY-NM-CLUSTER	484
8.1.1.7.7. FLEXRAY-NM-NODE	485
8.1.1.7.8. FLEXRAY-NM-CLUSTER-COUPLING	486
8.1.1.8. Software Component Description	486
8.1.1.8.1. COMPOSITION-SW-COMPONENT-TYPE	486
8.1.1.8.2. APPLICATION-SW-COMPONENT-TYPE	487
8.1.1.8.3. SERVICE-SW-COMPONENT-TYPE	487
8.1.1.8.4. COMPLEX-DEVICE-DRIVER-SW-COMPONENT-TYPE	488
8.1.1.8.5. ECU-ABSTRACTION-SW-COMPONENT-TYPE	489
8.1.1.8.6. PARAMETER-SW-COMPONENT-TYPE	489
8.1.1.8.7. R-PORT-PROTOTYPE	490
8.1.1.8.8. P-PORT-PROTOTYPE	490
8.1.1.8.9. CLIENT-SERVER-INTERFACE	491
8.1.1.8.10. ASSEMBLY-SW-CONNECTOR	491
8.1.1.8.11. DELEGATION-SW-CONNECTOR	491
8.1.1.8.12. SERVICE-CONNECTOR-PROTOTYPE	492
8.1.1.8.13. PARAMETER-INTERFACE	492
8.1.1.8.14. MODE-SWITCH-INTERFACE	492
8.1.1.8.15. MODE-GROUP	493
8.1.1.8.16. SW-COMPONENT-PROTOTYPE	493
8.1.1.8.17. VARIABLE-DATA-PROTOTYPE	493
8.1.1.8.18. ARGUMENT-DATA-PROTOTYPE	494
8.1.1.8.19. PARAMETER-DATA-PROTOTYPE	494
8.1.1.8.20. CLIENT-SERVER-OPERATION	495
8.1.1.8.21. APPLICATION-ERROR	495
8.1.1.8.22. NONQUEUED-RECEIVER-COM-SPEC	495
8.1.1.8.23. QUEUED-RECEIVER-COM-SPEC	496



8.1.1.8.24. CLIENT-COM-SPEC	496
8.1.1.8.25. PARAMETER-REQUIRE-COM-SPEC	497
8.1.1.8.26. QUEUED-SENDER-COM-SPEC	497
8.1.1.8.27. NONQUEUED-SENDER-COM-SPEC	498
8.1.1.8.28. SERVER-COM-SPEC	498
8.1.1.8.29. MODE-SWITCH-SENDER-COM-SPEC	498
8.1.1.8.30. PARAMETER-PROVIDE-COM-SPEC	499
8.1.1.8.31. TRANSMISSION-ACKNOWLEDGE	499
8.1.1.8.32. DATA-FILTER	499
8.1.1.8.33. VARIABLE-ACCESS	500
8.1.1.8.34. PARAMETER-ACCESS	501
8.1.1.8.35. RUNNABLE-ENTITY	502
8.1.1.8.36. WAIT-POINT	503
8.1.1.8.37. SWC-INTERNAL-BEHAVIOR	504
8.1.1.8.38. PORT-API-OPTION	505
8.1.1.8.39. DATA-SEND-COMPLETED-EVENT	506
8.1.1.8.40. DATA-RECEIVED-EVENT	506
8.1.1.8.41. DATA-RECEIVE-ERROR-EVENT	507
8.1.1.8.42. ASYNCHRONOUS-SERVER-CALL-RETURNS-EVENT	507
8.1.1.8.43. TIMING-EVENT	507
8.1.1.8.44. OPERATION-INVOKED-EVENT	508
8.1.1.8.45. SWC-MODE-SWITCH-EVENT	508
8.1.1.8.46. MODE-SWITCHED-ACK-EVENT	508
8.1.1.8.47. ASYNCHRONOUS-SERVER-CALL-POINT	509
8.1.1.8.48. SYNCHRONOUS-SERVER-CALL-POINT	509
8.1.1.8.49. NUMERICAL-VALUE-SPECIFICATION	509
8.1.1.8.50. TEXT-VALUE-SPECIFICATION	510
8.1.1.8.51. ARRAY-VALUE-SPECIFICATION	510
8.1.1.8.52. RECORD-VALUE-SPECIFICATION	511
8.1.1.8.53. CONSTANT-SPECIFICATION	511
8.1.1.8.54. CONSTANT-REFERENCE	511
8.1.1.8.55. UNIT	511
8.1.1.8.56. UNIT-GROUP	512
8.1.1.8.57. SW-BASE-TYPE	512
8.1.1.8.58. SW-DATA-DEF-PROPS	513
8.1.1.8.59. IMPLEMENTATION-DATA-TYPE	515
8.1.1.8.60. COMPU-METHOD	516
8.1.1.8.61. COMPU-SCALE	516
8.1.1.8.62. EXCLUSIVE-AREA	517
8.1.1.8.63. COM-MGR-USER-NEEDS	517
8.1.1.8.64. DIAGNOSTIC-COMMUNICATION-MANAGER-NEEDS	518
8.1.1.8.65. DIAGNOSTIC-EVENT-NEEDS	518



8.1.1.8.66. ECU-STATE-MGR-USER-NEEDS	518
8.1.1.8.67. FUNCTION-INHIBITION-NEEDS	519
8.1.1.8.68. NV-BLOCK-NEEDS	519
8.1.1.8.69. OBD-CONTROL-SERVICE-NEEDS	521
8.1.1.8.70. OBD-INFO-SERVICE-NEEDS	521
8.1.1.8.71. OBD-PID-SERVICE-NEEDS	522
8.1.1.8.72. OBD-RATIO-SERVICE-NEEDS	522
8.1.1.8.73. SUPERVISED-ENTITY-NEEDS	523
8.1.1.8.74. PER-INSTANCE-MEMORY	524
8.1.1.8.75. PER-INSTANCE-MEMORY-SIZE	524
8.1.1.8.76. SWC-IMPLEMENTATION	524
8.1.1.8.77. CODE	525
8.1.1.8.78. MODE-DECLARATION	526
8.1.1.8.79. MODE-DECLARATION-GROUP	526
8.1.1.8.80. DISABLED-MODE-IREF	526
8.1.1.8.81. MODE-SWITCHED-ACK	527
8.1.1.8.82. MODE-ACCESS-POINT	527
8.1.1.8.83. MODE-SWITCH-POINT	528
8.1.1.8.84. SW-ADDR-METHOD	528
8.1.1.8.85. MEMORY-SECTION	528
8.1.1.8.86. SYSTEM-MAPPING	528
8.1.1.8.87. SWC-TO-ECU-MAPPING	529
8.1.1.8.88. SWC-TO-IMPL-MAPPING	530
8.1.1.8.89. SENDER-RECEIVER-TO-SIGNAL-GROUP-MAPPING	530
8.1.1.8.90. SENDER-RECEIVER-TO-SIGNAL-MAPPING	530
8.1.1.8.91. CLIENT-SERVER-TO-SIGNAL-GROUP-MAPPING	531
8.1.1.8.92. SENDER-REC-RECORD-TYPE-MAPPING	532
8.1.1.8.93. SENDER-REC-RECORD-ELEMENT-MAPPING	532
8.1.1.8.94. SENDER-REC-ARRAY-TYPE-MAPPING	532
8.1.1.8.95. SENDER-REC-ARRAY-ELEMENT-MAPPING	533
8.1.1.8.96. END-TO-END-PROTECTION-SET	533
8.1.1.8.97. END-TO-END-PROTECTION	533
8.1.1.8.98. END-TO-END-PROTECTION-I-SIGNAL-I-PDU	534
8.1.1.8.99. END-TO-END-PROTECTION-VARIABLE-PROTOTYPE	534
8.1.2. Upgrading a system model from AUTOSAR 3.2.x to AUTOSAR 4.0.3	535
8.1.2.1. Topology	535
8.1.2.1.1. SYSTEM	535
8.1.2.1.2. ECU-INSTANCE	536
8.1.2.1.3. CAN-CLUSTER	537
8.1.2.1.4. CAN-COMMUNICATION-CONTROLLER	538
8.1.2.1.5. CAN-COMMUNICATION-CONNECTOR	539
8.1.2.1.6. CAN-PHYSICAL-CHANNEL	540



8.1.2.1.7. FLEXRAY-CLUSTER	540
8.1.2.1.8. FLEXRAY-COMMUNICATION-CONTROLLER	543
8.1.2.1.9. FLEXRAY-FIFO-CONFIGURATION	545
8.1.2.1.10. FLEXRAY-FIFO-RANGE	545
8.1.2.1.11. FLEXRAY-COMMUNICATION-CONNECTOR	546
8.1.2.1.12. FLEXRAY-PHYSICAL-CHANNEL	546
8.1.2.1.13. LIN-CLUSTER	547
8.1.2.1.14. LIN-MASTER	547
8.1.2.1.15. LIN-SLAVE	548
8.1.2.1.16. LIN-ERROR-RESPONSE	548
8.1.2.1.17. LIN-COMMUNICATION-CONNECTOR	548
8.1.2.1.18. LIN-PHYSICAL-CHANNEL	548
8.1.2.2. Communication	549
8.1.2.2.1. CAN-FRAME-TRIGGERING	549
8.1.2.2.2. FLEXRAY-FRAME-TRIGGERING	550
8.1.2.2.3. FLEXRAY-ABSOLUTELY-SCHEDULED-TIMING	550
8.1.2.2.4. LIN-FRAME-TRIGGERING	550
8.1.2.2.5. CAN-FRAME	551
8.1.2.2.6. FLEXRAY-FRAME	551
8.1.2.2.7. LIN-UNCONDITIONAL-FRAME	551
8.1.2.2.8. LIN-SPORADIC-FRAME	551
8.1.2.2.9. LIN-EVENT-TRIGGERED-FRAME	551
8.1.2.2.10. LIN-SCHEDULE-TABLE	551
8.1.2.2.11. ASSIGN-FRAME-ID	551
8.1.2.2.12. UNASSIGN-FRAME-ID	552
8.1.2.2.13. ASSIGN-NAD	552
8.1.2.2.14. FREE-FORMAT	552
8.1.2.2.15. PDU-TRIGGERING	552
8.1.2.2.16. I-SIGNAL-TRIGGERING	552
8.1.2.2.17. DCM-I-PDU	552
8.1.2.2.18. I-SIGNAL-I-PDU	552
8.1.2.2.19. MULTIPLEXED-I-PDU	552
8.1.2.2.20. DYNAMIC-PART	553
8.1.2.2.21. DYNAMIC-PART-ALTERNATIVE	553
8.1.2.2.22. STATIC-PART	554
8.1.2.2.23. SEGMENT-POSITION	554
8.1.2.2.24. N-PDU	554
8.1.2.2.25. NM-PDU	555
8.1.2.2.26. USER-DEFINED-PDU	555
8.1.2.2.27. USER-DEFINED-I-PDU	555
8.1.2.2.28. PDU-TO-FRAME-MAPPING	556
8.1.2.2.29. I-PDU-TIMING	556

8.1.2.2.30. I-SIGNAL-I-PDU-GROUP	556
8.1.2.2.31. PDUR-I-PDU-GROUP	556
8.1.2.2.32. I-SIGNAL-TO-I-PDU-MAPPING	557
8.1.2.2.33. I-SIGNAL	557
8.1.2.2.34. I-SIGNAL-GROUP	557
8.1.2.2.35. SYSTEM-SIGNAL	557
8.1.2.2.36. I-SIGNAL-PORT	557
8.1.2.2.37. FRAME-PORT	557
8.1.2.2.38. I-PDU-PORT	558
8.1.2.2.39. TRANSMISSION-MODE-DECLARATION	558
8.1.2.2.40. TRANSMISSION-MODE-CONDITION	558
8.1.2.2.41. TRANSMISSION-MODE-FALSE-TIMING	558
8.1.2.2.42. EVENT-CONTROLLED-TIMING	558
8.1.2.2.43. CYCLIC-TIMING	558
8.1.2.2.44. TIME-RANGE-TYPE	559
8.1.2.2.45. ABSOLUTE-TOLERANCE	559
8.1.2.2.46. RELATIVE-TOLERANCE	559
8.1.2.3. Gateway	559
8.1.2.3.1. FRAME-MAPPING	559
8.1.2.3.2. I-PDU-MAPPING	559
8.1.2.3.3. TARGET-I-PDU-REF	559
8.1.2.3.4. PDU-MAPPING-DEFAULT-VALUE	560
8.1.2.3.5. DEFAULT-VALUE-ELEMENT	560
8.1.2.3.6. I-SIGNAL-MAPPING	560
8.1.2.4. CAN Transport Layer	560
8.1.2.4.1. CAN-TP-CONFIG	560
8.1.2.4.2. CAN-TP-NODE	561
8.1.2.4.3. CAN-TP-ADDRESS	562
8.1.2.4.4. CAN-TP-CHANNEL	562
8.1.2.4.5. CAN-TP-CONNECTION	562
8.1.2.4.6. Second CAN-TP-CONNECTION for FULL-DUPLEX-MODE	563
8.1.2.5. FlexRay ISO Transport Layer	564
8.1.2.5.1. FLEXRAY-TP-CONFIG	564
8.1.2.5.2. FLEXRAY-TP-ECU	565
8.1.2.5.3. FLEXRAY-TP-NODE	566
8.1.2.5.4. TP-ADDRESS	566
8.1.2.5.5. FLEXRAY-TP-CONNECTION	566
8.1.2.5.6. FLEXRAY-TP-CONNECTION-CONTROL	567
8.1.2.5.7. FLEXRAY-TP-PDU-POOL	567
8.1.2.6. LIN Transport Layer	568
8.1.2.6.1. LIN-TP-CONFIG	568
8.1.2.7. Network Management	569



8.1.2.7.1. NM-CONFIG	569
8.1.2.7.2. NM-ECU	570
8.1.2.7.3. CAN-NM-CLUSTER	570
8.1.2.7.4. CAN-NM-NODE	571
8.1.2.7.5. CAN-NM-CLUSTER-COUPING	572
8.1.2.7.6. FLEXRAY-NM-CLUSTER	572
8.1.2.7.7. FLEXRAY-NM-NODE	573
8.1.2.7.8. FLEXRAY-NM-CLUSTER-COUPING	574
8.1.2.8. Software Component Description	574
8.1.2.8.1. COMPOSITION-SW-COMPONENT-TYPE	574
8.1.2.8.2. APPLICATION-SW-COMPONENT-TYPE	574
8.1.2.8.3. SERVICE-SW-COMPONENT-TYPE	574
8.1.2.8.4. COMPLEX-DEVICE-DRIVER-SW-COMPONENT-TYPE	575
8.1.2.8.5. ECU-ABSTRACTION-SW-COMPONENT-TYPE	576
8.1.2.8.6. PARAMETER-SW-COMPONENT-TYPE	576
8.1.2.8.7. R-PORT-PROTOTYPE	577
8.1.2.8.8. P-PORT-PROTOTYPE	577
8.1.2.8.9. CLIENT-SERVER-INTERFACE	577
8.1.2.8.10. DELEGATION-SW-CONNECTOR	577
8.1.2.8.11. ASSEMBLY-SW-CONNECTOR	577
8.1.2.8.12. SERVICE-CONNECTOR-PROTOTYPE	577
8.1.2.8.13. PARAMETER-INTERFACE	577
8.1.2.8.14. MODE-SWITCH-INTERFACE	578
8.1.2.8.15. MODE-GROUP	578
8.1.2.8.16. SW-COMPONENT-PROTOTYPE	578
8.1.2.8.17. VARIABLE-DATA-PROTOTYPE	578
8.1.2.8.18. ARGUMENT-DATA-PROTOTYPE	578
8.1.2.8.19. PARAMETER-DATA-PROTOTYPE	578
8.1.2.8.20. CLIENT-SERVER-OPERATION	578
8.1.2.8.21. APPLICATION-ERROR	579
8.1.2.8.22. NONQUEUED-RECEIVER-COM-SPEC	579
8.1.2.8.23. QUEUED-RECEIVER-COM-SPEC	580
8.1.2.8.24. CLIENT-COM-SPEC	580
8.1.2.8.25. PARAMETER-REQUIRE-COM-SPEC	580
8.1.2.8.26. QUEUED-SENDER-COM-SPEC	580
8.1.2.8.27. NONQUEUED-SENDER-COM-SPEC	581
8.1.2.8.28. SERVER-COM-SPEC	581
8.1.2.8.29. MODE-SWITCH-SENDER-COM-SPEC	581
8.1.2.8.30. PARAMETER-PROVIDE-COM-SPEC	582
8.1.2.8.31. TRANSMISSION-ACKNOWLEDGE	582
8.1.2.8.32. DATA-FILTER	582
8.1.2.8.33. VARIABLE-ACCESS	583



8.1.2.8.34. PARAMETER-ACCESS	583
8.1.2.8.35. RUNNABLE-ENTITY	583
8.1.2.8.36. WAIT-POINT	584
8.1.2.8.37. SWC-INTERNAL-BEHAVIOR	584
8.1.2.8.38. PORT-API-OPTION	585
8.1.2.8.39. DATA-SEND-COMPLETED-EVENT	585
8.1.2.8.40. DATA-RECEIVED-EVENT	585
8.1.2.8.41. DATA-RECEIVE-ERROR-EVENT	585
8.1.2.8.42. ASYNCHRONOUS-SERVER-CALL-RETURNS-EVENT	585
8.1.2.8.43. TIMING-EVENT	585
8.1.2.8.44. OPERATION-INVOKED-EVENT	585
8.1.2.8.45. SWC-MODE-SWITCH-EVENT	585
8.1.2.8.46. MODE-SWITCHED-ACK-EVENT	586
8.1.2.8.47. ASYNCHRONOUS-SERVER-CALL-POINT	586
8.1.2.8.48. SYNCHRONOUS-SERVER-CALL-POINT	586
8.1.2.8.49. NUMERICAL-VALUE-SPECIFICATION	586
8.1.2.8.50. TEXT-VALUE-SPECIFICATION	586
8.1.2.8.51. ARRAY-VALUE-SPECIFICATION	586
8.1.2.8.52. RECORD-VALUE-SPECIFICATION	586
8.1.2.8.53. CONSTANT-SPECIFICATION	586
8.1.2.8.54. CONSTANT-REFERENCE	587
8.1.2.8.55. UNIT	587
8.1.2.8.56. UNIT-GROUP	587
8.1.2.8.57. PHYSICAL-DIMENSION	587
8.1.2.8.58. SW-BASE-TYPE	588
8.1.2.8.59. SW-DATA-DEF-PROPS	589
8.1.2.8.60. DATA-CONSTR	590
8.1.2.8.61. DATA-CONSTR-RULE	591
8.1.2.8.62. PHYS-CONSTRS	591
8.1.2.8.63. INTERNAL-CONSTRS	592
8.1.2.8.64. IMPLEMENTATION-DATA-TYPE	592
8.1.2.8.65. COMPU-METHOD	592
8.1.2.8.66. COMPU-SCALE	592
8.1.2.8.67. EXCLUSIVE-AREA	593
8.1.2.8.68. COM-MGR-USER-NEEDS	593
8.1.2.8.69. DIAGNOSTIC-COMMUNICATION-MANAGER-NEEDS	593
8.1.2.8.70. DIAGNOSTIC-EVENT-NEEDS	593
8.1.2.8.71. ECU-STATE-MGR-USER-NEEDS	593
8.1.2.8.72. FUNCTION-INHIBITION-NEEDS	593
8.1.2.8.73. NV-BLOCK-NEEDS	593
8.1.2.8.74. OBD-CONTROL-SERVICE-NEEDS	594
8.1.2.8.75. OBD-PID-SERVICE-NEEDS	594



8.1.2.8.76. OBD-RATIO-SERVICE-NEEDS	594
8.1.2.8.77. SUPERVISED-ENTITY-NEEDS	594
8.1.2.8.78. PER-INSTANCE-MEMORY	594
8.1.2.8.79. PER-INSTANCE-MEMORY-SIZE	595
8.1.2.8.80. SWC-IMPLEMENTATION	595
8.1.2.8.81. CODE	595
8.1.2.8.82. MODE-DECLARATION	595
8.1.2.8.83. MODE-DECLARATION-GROUP	595
8.1.2.8.84. DISABLED-MODE-IREF	595
8.1.2.8.85. MODE-SWITCHED-ACK	595
8.1.2.8.86. MODE-ACCESS-POINT	595
8.1.2.8.87. MODE-SWITCH-POINT	596
8.1.2.8.88. SW-ADDR-METHOD	596
8.1.2.8.89. MEMORY-SECTION	596
8.1.2.8.90. SYSTEM-MAPPING	596
8.1.2.8.91. SWC-TO-ECU-MAPPING	596
8.1.2.8.92. SWC-TO-IMPL-MAPPING	597
8.1.2.8.93. SENDER-RECEIVER-TO-SIGNAL-GROUP-MAPPING	597
8.1.2.8.94. SENDER-RECEIVER-TO-SIGNAL-MAPPING	597
8.1.2.8.95. CLIENT-SERVER-TO-SIGNAL-GROUP-MAPPING	597
8.1.2.8.96. SENDER-REC-RECORD-TYPE-MAPPING	597
8.1.2.8.97. SENDER-REC-RECORD-ELEMENT-MAPPING	597
8.1.2.8.98. SENDER-REC-ARRAY-TYPE-MAPPING	597
8.1.2.8.99. SENDER-REC-ARRAY-ELEMENT-MAPPING	597
8.1.2.8.100. PNC-MAPPING	598
8.1.2.8.101. END-TO-END-PROTECTION-SET	598
8.1.2.8.102. END-TO-END-PROTECTION	598
8.1.2.8.103. END-TO-END-PROTECTION-I-SIGNAL-I-PDU	598
8.1.2.8.104. END-TO-END-PROTECTION-VARIABLE-PROTOTYPE	598
8.1.3. Upgrading a system model from AUTOSAR 3.2.2 to AUTOSAR 4.0.3	598
8.1.3.1. Topology	599
8.1.3.1.1. CAN-CLUSTER	599
8.1.3.1.2. CAN-COMMUNICATION-CONNECTOR	600
8.1.3.1.3. FLEXRAY-CLUSTER	601
8.1.3.1.4. FLEXRAY-COMMUNICATION-CONNECTOR	603
8.1.3.1.5. LIN-CLUSTER	604
8.1.3.1.6. LIN-COMMUNICATION-CONNECTOR	605
8.1.3.1.7. LIN-SLAVE-CONFIG	605
8.1.3.2. FlexRay ISO Transport Layer	606
8.1.3.2.1. FLEXRAY-TP-ECU	606
8.1.3.3. Network Management	607
8.1.3.3.1. NM-COORDINATOR	607



8.1.3.4. Software Component Description	607
8.1.3.4.1. APPLICATION-SW-COMPONENT-TYPE	607
8.1.3.4.2. SERVICE-SW-COMPONENT-TYPE	608
8.1.3.4.3. COMPLEX-DEVICE-DRIVER-SW-COMPONENT-TYPE	608
8.1.3.4.4. ECU-ABSTRACTION-SW-COMPONENT-TYPE	609
8.1.3.4.5. NV-BLOCK-SW-COMPONENT-TYPE	610
8.1.3.4.6. NV-BLOCK-DESCRIPTOR	610
8.1.3.4.7. NV-BLOCK-DATA-MAPPING	611
8.1.3.4.8. ROLE-BASED-PORT-ASSIGNMENT	611
8.1.3.4.9. P-PORT-PROTOTYPE	612
8.1.3.4.10. NONQUEUED-RECEIVER-COM-SPEC	612
8.1.3.4.11. QUEUED-RECEIVER-COM-SPEC	613
8.1.3.4.12. NV-PROVIDE-COM-SPEC	614
8.1.3.4.13. RUNNABLE-ENTITY	614
8.1.3.4.14. R-PORT-PROTOTYPE	616
8.1.3.4.15. NV-REQUIRE-COM-SPEC	616
8.1.3.4.16. NV-DATA-INTERFACE	616
8.1.3.4.17. COMPU-SCALE	617
8.1.3.4.18. NV-BLOCK-NEEDS	617
8.1.3.4.19. END-TO-END-PROTECTION	619
8.1.3.5. BSW Module Description	619
8.1.3.5.1. BSW-MODULE-DESCRIPTION	619
8.1.3.5.2. BSW-MODULE-ENTRY	620
8.1.3.5.3. SW-SERVICE-ARG	621
8.1.3.5.4. RETURN-TYPE	621
8.1.3.5.5. BSW-MODULE-DEPENDENCY	622
8.1.3.5.6. BSW-INTERNAL-BEHAVIOR	622
8.1.3.5.7. BSW-INTERRUPT-ENTITY	623
8.1.3.5.8. BSW-SCHEDULABLE-ENTITY	624
8.1.3.5.9. BSW-CALLED-ENTITY	625
8.1.3.5.10. BSW-TIMING-EVENT	626
8.1.3.5.11. BSW-INTERNAL-TRIGGER-OCCURRED-EVENT	626
8.1.3.5.12. BSW-IMPLEMENTATION	627
8.1.3.5.13. SWC-BSW-MAPPING	629
8.1.4. AUTOSAR 4.0 platform data types	629
Bibliography	631
A. Third party license	632
Index	636

Begin here

1. If you are upgrading from a previous version

- ▶ What's new in this EB tresos Studio version?

Refer to the document EB tresos Studio new and noteworthy, located in your EB tresos Studio <INSTALL_PATH>/doc directory.

- ▶ Which known problems, fixed problems, incompatibilities to previous releases, limitations and restrictions have been reported for the current EB tresos Studio version?

Refer to the EB tresos Studio release notes, located in your EB tresos Studio <INSTALL_PATH>/doc directory.

2. If you are using EB tresos Studio for the first time

If you are a first time user of EB tresos Studio, you may want to get familiar with some of the concepts behind AUTOSAR at [Section 4.2, “The AUTOSAR concept”](#).

- ▶ What is EB tresos Studio?

The best way to find out more about EB tresos Studio is to read through [Chapter 4, Background information](#). In this chapter you find basic information about file formats and basic concepts within EB tresos Studio. You will need this knowledge to understand the rest of the user's guide.

- ▶ How do I start working with EB tresos Studio?

The chapter [Chapter 5, Using EB tresos Studio for the first time](#) covers the very first start of EB tresos Studio.

- ▶ What are all these GUI elements for?

To get to know the EB tresos Studio GUI, refer to [Chapter 6, The graphical user interface \(GUI\)](#).

- ▶ Can I customize the EB tresos Studio GUI for my needs?

Yes, you can. Information on how to do that is available at [Section 7.2, “Changing the EB tresos Studio setup”](#).

3. If you want to configure AUTOSAR modules or your own modules with EB tresos Studio

[Chapter 7, Use cases](#) is the place where you find instructions for:

- ▶ Starting a new project ([Section 7.3, “Working with projects”](#)).
- ▶ Using EB tresos Studio to edit an ECU configuration ([Section 7.8, “Editing parameters of a module configuration”](#)).
- ▶ Using the importing functions to import DBC, LDF, Fibex, or AUTOSAR files ([Section 7.10, “Importing and exporting”](#)).
- ▶ Generating source code out of your ECU configuration ([Section 7.9, “Generating code for projects”](#)).
- ▶ Using the EB tresos Studio tool on the command line ([Section 7.11, “Working with the command line”](#)).

4. If you want to extend the EB tresos Studio functionality

Refer to the EB tresos Studio developer's guide, if you want to extend the EB tresos Studio functionalities and find out how to:

- ▶ work with Eclipse,
- ▶ develop your own modules,
- ▶ work with configuration models,
- ▶ create your own code generator, or
- ▶ extend the EB tresos Studio user interface.

The EB tresos Studio developer's guide is delivered as a separate document, which is located at `$TRESOS_BASE\doc\2.0_EB_tresos_Studio`. `$TRESOS_BASE` is the directory into which you have installed EB tresos Studio.

5. If you need help/more information

- ▶ [Chapter 3, EB tresos product line support](#)

Receive technical support via email or phone from the [Chapter 3, EB tresos product line support](#) hotline.

► [Chapter 1, About this documentation](#)

Find out about:

- Required knowledge, tools, and system environment
- Typographical and style conventions used throughout this documentation. Defines usage of special fonts in the documentation.
- Naming conventions used in this documentation. Defines usage of special names and small/capital lettering in the documentation.

► [Index](#)

You cannot find what you are looking for? Try the [Index](#) list (alphabetically sorted).

► [EB tresos glossary](#)

You do not understand what a word or abbreviations means? Find out in the [EB tresos glossary](#).

► [Bibliography](#)

Would you like to read more detailed information? Find bibliographic references and further reading suggestions in the [Bibliography](#).

1. About this documentation

1.1. Introduction

Welcome to the EB tresos Studio documentation.

This chapter [About this documentation](#) provides you with

- ▶ Required knowledge, tools, and system environment
- ▶ Typographical and style conventions used throughout this documentation. Defines usage of special fonts in the documentation.
- ▶ Naming conventions used in this documentation. Defines usage of special names and small/capital lettering in the documentation.

NOTE

Please note that there is a separate documentation for the EB tresos AutoCore, if you also purchased and installed the EB tresos AutoCore modules.

1.2. Required knowledge and system environment

1.2.1. Required knowledge

If you want to configure an AUTOSAR stack with EB tresos Studio, you need to know the following:

- ▶ the EB tresos Studio user's guide
- ▶ the documentation of any other products from the EB tresos product line that you purchased along with EB tresos Studio
- ▶ the documentation of any MCAL products that you purchased along with EB tresos Studio

If you want to extend the EB tresos Studio functionality, you need to know the following:

- ▶ the EB tresos Studio developer's guide
- ▶ Java programming language
- ▶ basic knowledge about developing Eclipse plug-ins

1.2.1. Required system environment

Item	Requirement
operating system	Microsoft Windows XP/ Windows 7 (32-Bit or 64-Bit)
processor	Intel Pentium or compatible, 1 GHz
memory	500 MB
drives	CD-Rom drive for the installation CD
connectors	USB port, if dongled licenses are used
network	Network connection, if network licenses are used

Table 1.1. Minimum system requirements

1.3. Interpretation of version information

1.3.1. Product version number

Each product within the product line is assigned an individual *product version number*.

The product version number scheme is made of three parts:

1. The major number

An increment of the *major* version number indicates that the release contains major new features and changes compared to the previous major version.

2. The minor number

An increment of the *minor* version number indicates that the release contains minor new features and changes compared to the previous minor version.

3. The patch number

An increment of the *patch* version number indicates that the release fixes known problems.

The numbers are separated by dots in the following naming scheme: <major>. <minor>. <patch>.

Examples for spelled out product version numbers are:

- ▶ EB tresos AutoCore OS 4.0
- ▶ EB tresos AutoCore OS 4.1
- ▶ EB tresos AutoCore Generic 4.3

- ▶ EB tresos AutoCore Generic 4.4

You can find the product version number for example on the documentation cover and the release notes cover.

Within GUIs, e.g. in EB tresos Studio, you can find the product version number on the splash screen and the *EB tresos details* dialog next to the keyword *Studio*.

There may be a *qualifier* additionally to the product version number. This *qualifier* is provided in brackets and displayed after the product version number.

1.3.2. Qualifiers for basic software

Basic software of the EB tresos product line are the products: EB tresos AutoCore Generic and EB tresos AutoCore OS.

The qualifier for basic software is provided in the quality statement (Q statement), which is part of each delivery. In the quality statement the application area is documented based on the quality level of the delivery. The following quality levels are defined for basic software:

- ▶ Prototype Level
- ▶ Development Level
- ▶ Mass Production Level

For details on these levels, please refer to the quality statement of your delivery.

The quality statement is provided in parallel to the installation file of the basic software and on the command server (EB Command) in the naming scheme: `EB_tresos_AutoCore-quality_statement-<RelName>-<Target>-B<BuildNr>(_<Suffix>).doc`, e.g. `EB_tresos_AutoCore-quality_statement-ACG-6.4-WIN32X86-B64208.doc`

1.4. Typography and style conventions

Throughout the documentation you will see that words and phrases are displayed in bold or italic font, or in Monospace font. To find out what these conventions mean, please consult the following table. All default text is written in Arial Regular font without any markup.

Convention	Item is used	Example
Arial italics	to define new terms	The <i>basic building blocks</i> of a configuration are module configurations.
Arial italics	to emphasize	If your project's release version is mixed, all content types are available. It is thus called <i>mixed version</i> .

Convention	Item is used	Example
Arial italics	to indicate that a term is explained in the glossary	...exchanges <i>protocol data units (PDUs)</i> with its peer instance of other ECUs.
Arial boldface	for menus and submenus	Choose the Options menu.
Arial boldface	for buttons	Select OK .
Arial boldface	for keyboard keys	Press the Enter key
Arial boldface	for keyboard combination of keys	Press Ctrl+Alt+Delete
Arial boldface	for commands	Convert the XDM file to the newer version by using the legacy convert command.
Monospace font (Courier)	for file and folder names, also for chapter names	Put your script in the <code>function_name/abc-folder</code>
Monospace font (Courier)	for code	<code>for (i=0; i<5; i++) { /* now use i */ }</code>
Monospace font (Courier)	for function names, methods, or routines	The <code>cos</code> function finds the cosine of each array element. Syntax line example is <code>MLGetVar ML_var_name</code>
Monospace font (Courier)	for user input/indicates variable text	Enter a three-digit prefix in the menu line.
Square brackets []	for optional parameters; for command syntax with optional parameters	<code>insertBefore [<opt>]</code>
Curly brackets {}	for mandatory parameters; for command syntax with mandatory parameters (in curly brackets)	<code>insertBefore {<file>}</code>
Three dots ...	for further parameters	<code>insertBefore [<opt>...]</code>
A vertical bar	to separate parameters in a list from which one parameters must be chosen or used; for command syntax, indicates a choice of parameters	<code>allowinvalidmarkup {on off}</code>

Convention	Item is used	Example
Warning	to show information vital for the success of your configuration	WARNING  This is a warning This is what a warning looks like.
Notice	to give additional important information on the subject	NOTE  This is a notice This is what a notice looks like.
Tip	to provide helpful hints and tips	TIP  This is a tip This is what a tip looks like.

1.5. Naming conventions

The naming conventions listed below will enable you to understand meanings of words and word groups that may otherwise be confusing without explanation. This chapter is useful as a reference point if you are unsure what a specific spelling (e.g. capital letters in parameters, a word in boldface that appears on a screen) means. Browse through to see the conventions and return if you need an explanation for a phenomenon you do not understand.

1.5.1. AUTOSAR XML schema

Parameters spelled in capital letters (e.g. RECOMMENDED-CONFIGURATION-REF and BSW-MODULE-DESCRIPTION) refer to the AUTOSAR XML schema.

XML tags are presented as specified in AUTOSAR Model Persistence Rules for XML V2.1.2 R3.0 Rev 0001, specifically in chapter 3.6 XML names [5]:

- ▶ All XML elements, XML attributes, XML groups and XML types used in the AUTOSAR XML schema are written in upper case letters.
- ▶ In order to increase the readability of the XML names, hyphens are inserted in the XML names which separate the parts of the names.

1.5.2. Configuration parameter names

All configuration parameter names used in EB tresos Studio could be used for code generation and therefore potentially conflict with names used in the EB tresos AutoCore.

WARNING To avoid naming conflicts, do not use any of the following names as configuration parameter name:



- ▶ any C keywords
- ▶ name of AUTOSAR modules, e.g. Rte, Os, Com.

1.6. EB tresos Studio naming conventions

1.6.1. EB tresos Studio command shell

System commands and properties in the EB tresos Studio command shell are spelled with mixed lower- and upper-case letters, e.g. -DmergeConfigs.

Where the AUTOSAR XML schema uses hyphens (–) to separate parameter parts, EB tresos Studio uses the camelback case to separate different parts of the parameters.

Define the SHORT-NAME of the parameter ECU-PARAMETER. Example: -DecuParamDefName=<name>

Example 1.1. Example instructions that use the AUTOSAR XML schema and the EB tresos Studio spelling convention

2. Safe and correct use of EB tresos Studio

2.1. Intended usage of EB tresos Studio

EB tresos Studio is intended to be used for configuring and generating EB tresos OsekCore and ECU basic software compliant to the AUTOSAR standard (see www.autosar.org).

2.2. Possible misuse of EB tresos Studio

- ▶ If you use the product in non-automotive projects or in automotive projects that are not based on OSEK or AUTOSAR technology (see www.autosar.org), the product and its technology may not conform to the requirements of your application. Elektrobit Automotive GmbH is not liable for such misuse.

2.3. Target group and required knowledge

- ▶ Automotive software engineers
- ▶ Programming skills and experience in programming AUTOSAR- and OSEK-compliant ECUs

2.4. Quality standards compliance

EB tresos Studio has been developed following processes that have been assessed and awarded ISO 9001:2008. These processes are based on Automotive SPICE. Should it be necessary to use any part of EB tresos Studio for a safety relevant project, contact EB first.

3. EB tresos product line support

Europe

Phone: +49 9131 7701 6060

Japan

Phone: +81 3 5577 6110

USA

Phone: +1 888 346 3813

Support URL

<http://automotive.elektrobit.com/support>

4. Background information

4.1. Overview

The chapter **Background information** is for first-time users of EB tresos Studio and those, who would like

- ▶ to update their knowledge of the basic concepts of AUTOSAR and
- ▶ to find out some of the advantages EB tresos Studio has to offer.

In each of the following chapters you will also find sections called **Background information**. These provide a conceptual context for the instructions and references that follow in later sections.

While **Background information** chapters may help advanced users of EB tresos Studio to figure out why some things are the way they are, they are primarily aimed at newcomers to AUTOSAR and EB tresos Studio and those, not very familiar with the graphical user interface.

4.2. The AUTOSAR concept

AUTOSAR (*Automotive Open System Architecture*) is a partnership of

- ▶ automotive,
- ▶ electronics,
- ▶ semiconductor,
- ▶ hard- and software

companies that work toward an open industry standard for automotive embedded architectures.

Software adhering to this industry standard allows you

- ▶ to manage the complicated interaction within automotive Electrical/Electronic (E/E) architectures,
- ▶ to make product upgrades and updates easier and more flexible,
- ▶ to re-use software within a product line and across product lines,
- ▶ to improve E/E systems' quality and reliability,
- ▶ and to find software errors in the early phases of design. Electronic control units (ECU) are the platforms on which the functions of the application (AUTOSAR application software components) are executed.

AUTOSAR functionally separates an ECU's application from the ECU's basic functionalities. An ECU's basic functionalities are

- ▶ the operating system,
- ▶ communication systems,
- ▶ and access to peripherals.

These basic functionalities are implemented in what AUTOSAR calls Basic Software (BSW). All data exchange between application components and the BSW are routed via an abstraction layer. This abstraction layer is the Run-Time Environment (Rte), which hides the BSW layers from the application components. The Rte implements the Virtual Function Bus (VFB), which is the abstract communication layer via which the application components actually exchange signals/data. Thus application components exchange data without knowing the path this data is going to take within or between ECUs.

The BSW includes drivers to access peripherals and communication, and the operating system. Application engineers no longer need to or indeed may access the BSW's resources directly but route their call via defined interfaces in the Rte. In theory this also enables to move functions from one ECU to another. In practice this is difficult because of signal transmission times and delays.

4.3. What is EB tresos Studio?

4.3.1. The configuration and generation tool

Since AUTOSAR relies on generating code according to user requirements, a configuration tool such as EB tresos Studio is needed to configure standard software components and to generate specific ECUs.

The tool follows the AUTOSAR methodology, but can be used for much more than just basic software modules of this standard.

The open concept of its code generator engines, parameter verification language, data model access and GUI allows you to seamlessly integrate EB tresos Studio into existing tool-chains and workflows.

It helps you to deploy standard software components within your organization in an efficient and easy-to-use manner.

The main concepts of EB tresos Studio are to follow the AUTOSAR methodology, provide open and documented interfaces, help to avoid user-errors and create an easy-to-use tool environment.

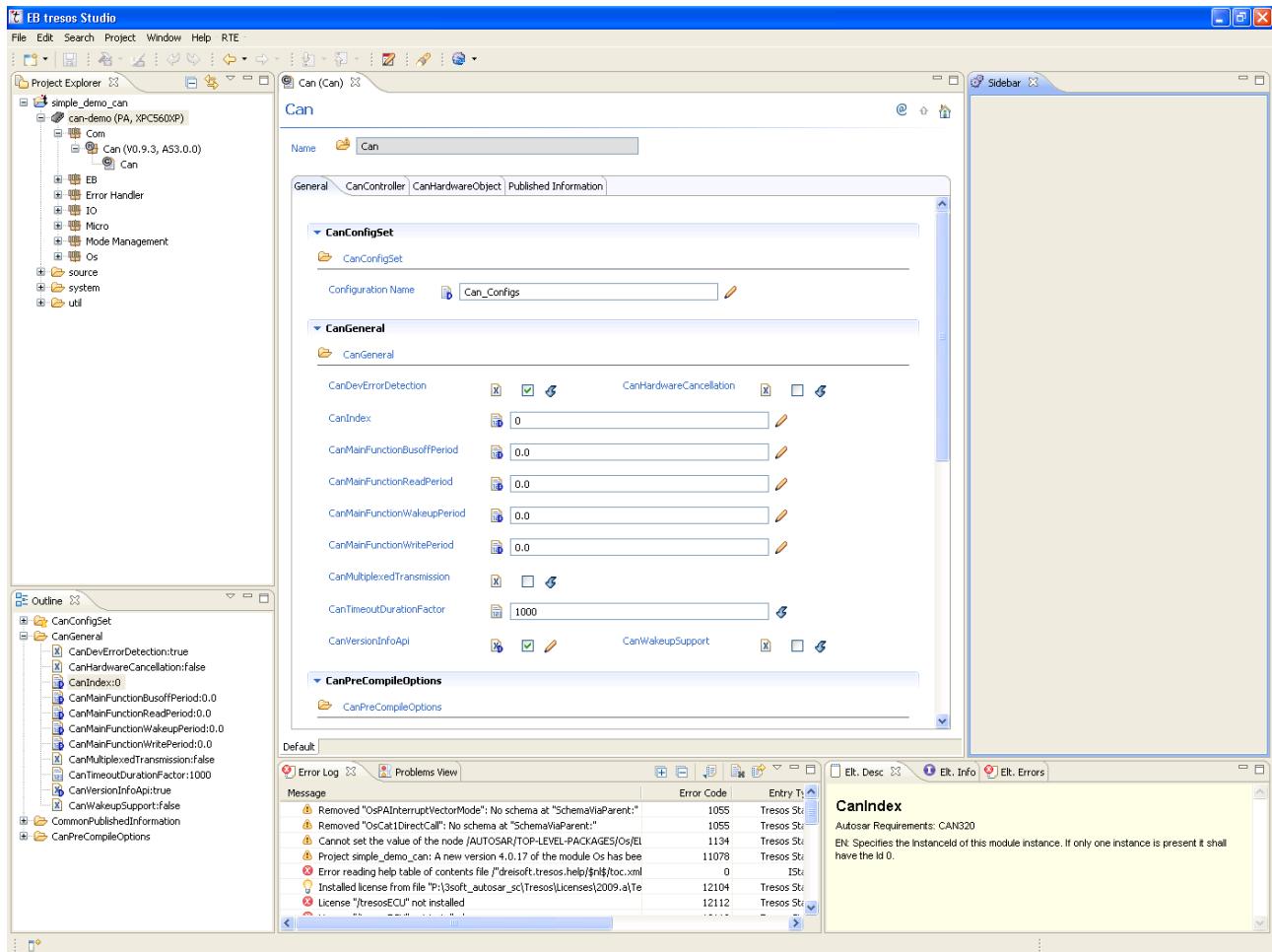


Figure 4.1. The main window of EB tresos Studio

4.3.2. Usage

Typical use scenarios for EB tresos Studio are:

- ▶ OEMs use EB tresos Studio for pre-integration of ECU standard-core software and deployment among their suppliers.
- ▶ Tier-one companies use EB tresos Studio for specification and code generation of basic software for their production ECU projects.
- ▶ Semiconductor vendors have chosen EB tresos Studio as the tool for the deployment of microcontroller drivers among their customers. Of course, EB uses EB tresos Studio as the configuration editor for all their EB tresos AutoCore products.

4.3.3. Benefits

You can immediately experience the benefits of EB tresos Studio's AUTOSAR methodology, when you work with the software tool:

- ▶ User-friendly configuration of basic software modules
- ▶ Competent implementation of the AUTOSAR methodology
- ▶ Quick familiarization
- ▶ Open and documented interfaces
- ▶ Extendibility and interoperability

4.3.4. Features

EB tresos Studio offers a wealth of features, which simplify your job of configuring software modules for an *ECU*.

Furthermore, the tool offers interfaces and methods for tool-chain integration and customization:

- ▶ Generic configuration editor according to AUTOSAR with many add-on features and improvements.
- ▶ Easy navigation within the ECU configuration data structure via
 - ▶ tabbed editors,
 - ▶ node views,
 - ▶ and clickable parameter references.
- ▶ Eclipse-based graphical user-interface, customizable via Java plug-ins.
- ▶ Validation of configuration parameters and immediate user feedback of detected issues.
- ▶ Importers for legacy formats such as
 - ▶ *FIBEX*,
 - ▶ *DBC*,
 - ▶ and *LDF*
- ▶ Integration of company-specific software modules due to an open Java API and code generator engines
- ▶ Interoperability with system architecture tools such as dSPACE SystemDesk.

In detail, these features include:

Eclipse based framework

Using Eclipse 3.x as the foundation technology ensures major extensibility and support for a fast growing set of software development related tooling.

Configuration capabilities

EB tresos Studio provides support for configuration editors that allow to edit the configuration of an embedded software module.

Generic configurator

EB tresos Studio provides a generic configuration editor that uses the definition of configuration parameters and automatically displays a dialog-based graphical user interface (GUI).

Custom configurators

EB tresos Studio can handle multiple editors per module. Editors can be specifically implemented in Java for a specific module.

Code generator capabilities

EB tresos Studio provides support for code generators that allow to transform a configuration into source code that can be compiled for an embedded target platform.

Commandline interface

EB tresos Studio provides a commandline interface for code-generation to run in batch process.

Multiple generators

Each module may provide its own or even multiple generators.

Template-based generator

EB tresos Studio provides a code-template based generic generator engine.

Project-based configuration

EB tresos Studio is capable of managing multiple configuration projects where each project may represent the configuration for a whole ECU.

Multiple modules

A project is capable to host multiple modules that share a common configuration database.

Multiple configurations

A module can be configured multiple times inside one project.¹

Cross module dependencies

Cross module-dependencies within the configuration of one ECU can be expressed via XPath-expressions and are evaluated constantly during configuration.

GUI hints

A parameter definition of a module can contain GUI hints such as an enable/disable expression, a HTML-description or layout hints to allow the generic configurator provide a more useable interface.

List in the configuration

Lists of elements in the configuration are shown as a table with multiple sortable columns.

Navigatable references

From references in the configuration that point to loaded parameters the GUI can be navigated easily to the referenced element.

¹Not every module may support this option.

Importer/Exporter

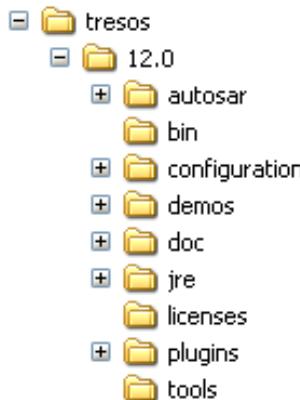
EB tresos Studio provides a generic importer/exporter interface that allows to add importers and exporters for foreign formats such as OIL, DBC, DIL, Fibex and annotates imported data so that a re-import is possible.

ECU Configuration

EB tresos Studio allows to configure, import and export ECU Configuration modules compatible to the AUTOSAR release versions 2.0, 2.1, 3.0, 3.1, 3.2, 4.0.2 and 4.0.3.

4.4. File and directory structure

A new EB tresos Studio installation has the following directory structure:



Directory	Description
autosar	contains the AUTOSAR ECU configuration parameter definition for each supported AUTOSAR version and a Perl script to generate EPD files from each parameter definition
bin	contains the executable files to start EB tresos Studio as a 32 bit application (<code>tresos_gui.exe</code> and <code>tresos_cmd.bat</code>) or as a 64 bit application (<code>tresos_gui_64.exe</code> and <code>tresos_cmd_64.bat</code>)
configuration	run-time configuration information for EB tresos Studio, for example the plug-in cache
demos	contains demo plug-ins for developers; the demos showcase the EB tresos Studio API
doc	program documentation, e.g. the EB tresos Studio user's guide
jre	Java run-time environment; this is required to run EB tresos Studio
jre64	Java run-time environment; this is required to run EB tresos Studio as a 64 bit application.
licenses	license agreements of third-party products

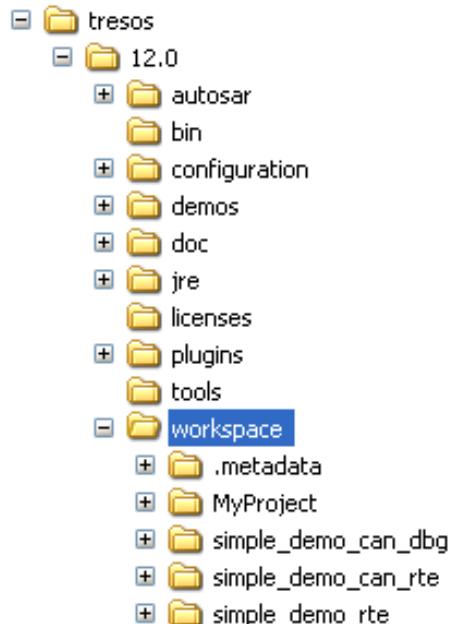
Directory	Description
plugin	plug-ins and modules, which are part of the EB tresos Studio installation
tools	contains tools for developers that use the EB tresos Studio API

4.5. EB tresos Studio projects and workspaces

The EB tresos Studio tool enables you to work with multiple projects rather than a single configuration. Projects are arranged in a workspace.

4.5.1. The Workspace

A workspace is a folder by default in the installation tree of your EB tresos Studio installation. If the workspace folder is not present, it is automatically created upon the first start of the program. It is possible to have several workspaces managing different projects (see [Section 7.2.4.1, “Switching the workspace”](#)).



Besides projects the workspace stores non project specific preferences and also log files in a meta data area.

EB tresos Studio always operates on one workspace which is initially selected on tool startup (see [Section 7.2.4.1, “Switching the workspace”](#)). The workspace can be switched via the **File** menu, EB tresos Studio restarts in this case with the new workspace open.

4.5.2. Projects

ECU configurations, system descriptions and also companion files like the source code of an application can be stored in a project. A project is represented as a folder with some hidden files describing the project meta data. The project folder normally resides beneath the workspace folder but can also be placed somewhere else in the file system. Projects can be imported into a workspace after which they show up in the EB tresos Studio GUI.

An EB tresos Studio project can host an ECU configuration consisting of module configurations in the form of XML files or a system description model in the form of a database or both. ECU configurations and system descriptions are stored in a folder `config` beneath the top level folder of the project. Arbitrary files can be created or imported into a project folder via the GUI of EB tresos Studio

4.5.2.1. Configuration time of projects

The *configuration time* is the phase or the process step of your project during the development of AUTOSAR basic software (BSW) modules. It defines which parts of a module must be configured in which development phase. AUTOSAR defines the following four configuration classes:

Configuration class	Description
PublishedInformation	Specifies which information is fixed before the precompile stage.
PreCompile	Specifies that the module is configured before compiling the source code.
Link	Specifies that the module is configured during link time. This means the object code of the module receives parts of its configuration from another object code file.
PostBuild	Specifies that the module gets the parameters of its configuration either by downloading a separate file to the ECU memory (<code>POST-BUILD-LOADABLE</code>) or by defining multiple configuration sets and choosing one during boot-time (<code>POST-BUILD-SELECTABLE</code>).

For more detailed information on AUTOSAR configuration classes, refer to [4], chapter 2.3.2 Introduction To Configuration Classes.

The order of the configuration times in the configuration process is: PublishedInformation -> PreCompile -> Link -> PostBuild.

You may define the configuration time of your project. For instructions on defining the configuration time of your project, refer to [Section 7.3.7.3, “Editing the ECU Configuration settings”](#). The configuration time of the project defines which parameters and references of the module configuration may be edited in the following way: Each parameter and each reference of an ECU configuration belongs to exactly one configuration class. The configuration class of a parameter or a reference depends on the configuration variant of the module defi-

nition: Each module defines the configuration variants it supports. While configuring a module, you must select one of the configuration variants of the module. Instructions for setting the configuration variant is available at [Section 7.8.4, “Setting the configuration variant of the module”](#). The selected configuration variant is then called *implementation configuration variant*. If the implementation configuration variant of a module is set, the configuration class of all parameters and references of the module are uniquely determined.

The procedure is as follows:

- ▶ Select the configuration variant of your modules (see [Section 7.8.4, “Setting the configuration variant of the module”](#)) to set the configuration class of the parameters and references.
- ▶ Select the configuration time of your project (see [Section 7.3.7.3, “Editing the ECU Configuration settings”](#)) to limit the access to the parameters and references of your configuration process step.

4.5.3. File content types

Content types define the content of files more precisely than the file extension. This influences how the information read from files is interpreted or how the information is saved to the files. In EB tresos Studio, content types are used in two locations: when using the AUTOSAR importer or exporter (see [Section 7.10.2.2, “Importing and exporting configuration data from and to the AUTOSAR format”](#)) and when using the command line (see [Section 7.11, “Working with the command line”](#)).

The content type of a file defines how a file is validated, e.g. against an XSD schema. The content type also determines how new files are written. Every input and output file for each command on the command line may be equipped with a content type.

By default, EB tresos Studio knows the following content types:

- ▶ Files containing module configuration data:
 - ▶ `xdm` for the XDM file format
 - ▶ `asc` for AUTOSAR module configuration files
- ▶ Files containing system description data:
 - ▶ `sysd` for AUTOSAR system description files
 - ▶ `t_db` for tresosDB system description files
 - ▶ `dbc` for the DBC file format
 - ▶ `ldf` for the LDF file format
 - ▶ `fibex` for Fibex files

In addition, parsers contained in other plugins or modules that you may have installed may define their own content types.

4.5.3.1. Content types for XDM files

Different versions of EB tresos Studio may use different versions of the XDM file format. Newer versions of EB tresos Studio are always able to read the older file versions. The following content types are available:

Content type	Description
xdm	refers to the latest version of the XDM file format (currently 3.0)
xdm:2.0	refers to an older version of the XDM format which was used in versions prior to and including EB tresos Studio 2007.b
xdm:3.0	refers to the version of the XDM format that has been used since EB tresos Studio 2008.a.

4.5.3.2. Content types for AUTOSAR module configuration files

EB tresos Studio enables importing or exporting project configuration data from and to the AUTOSAR format. To do so, you need to be aware of the AUTOSAR content type of the file to be imported or exported. The term content type describes the fact that the AUTOSAR specification has been released in several versions already. These content type versions are validated against different XSD schemas and their characteristics differ from each other.

The table below lists all AUTOSAR formats supported. To exchange files between different tools, use a format that can be checked by both tools when they use the same AUTOSAR XSD file.

Content type	Description
asc	Default content type.
asc:2.0	Imports/Exports files using the AUTOSAR XSD version 2.0.
asc:spec2.0	Imports/Exports files using the AUTOSAR XSD version 2.0. Writes files as specified in [3].

Differences to asc:2.0:

- ▶ Empty values and non-configured values are allowed. This also applies to integer values. This means that empty value and value-ref tags may occur, e.g. <value></value>! However, <value/> may not occur.
- ▶ For configuration classes, the values are stored/accepted as defined within the specification: PreCompile, Link, PostBuild, PostBuildLoadable, and PostBuildSelectable.
- ▶ The order of XML elements is not specified.

Content type	Description
asc:2.1	Imports/Exports files using the AUTOSAR XSD version 2.1. This format enables exchanging files between tools if the tools use the same AUTOSAR XSD file format and can check the files' validity.
asc:3.0	Imports/Exports files using the AUTOSAR XSD version 3.0.
asc:3.1	Imports/Exports files using the AUTOSAR XSD version 3.1.
asc:3.2	Imports/Exports files using the AUTOSAR XSD version 3.2.
asc:4.0.2	Imports/Exports files using the AUTOSAR XSD version 4.0.2.
asc:4.0.3	Imports/Exports files using the AUTOSAR XSD version 4.0.3.

4.5.3.3. Content types for AUTOSAR system description files

EB tresos Studio can import system description data from AUTOSAR system description files and convert them into a tresosDB file. For details about the file format, refer to [Section 4.5.3.4, “Content types for tresosDB files”](#).

The AUTOSAR specification has been released in several versions already. EB tresos Studio support the following ones:

Content type	Description
sysd	Default content type, EB tresos Studio will try to automatically detect the right version
sysd:2.1.2	Imports files using the AUTOSAR XSD version 2.1 (XSD Document Version 2.1.3 Revision 0017, xmlns="http://autosar.org/2.1.2")
sysd:2.1.4	Imports files using the AUTOSAR XSD version 2.1 (XSD Document Version 2.1.4 Revision 0018, xmlns="http://autosar.org/2.1.4")
sysd:3.0.2	Imports files using the AUTOSAR XSD version 3.0 (XSD Document Version 3.0.2 Revision 0003, xmlns="http://autosar.org/3.0.2")
sysd:3.1.0	Imports files using the AUTOSAR XSD version 3.1 (XSD Document Version 3.1.0 Revision 0001, xmlns="http://autosar.org/3.1.0")
sysd:3.1.2	Imports files using the AUTOSAR XSD version 3.1 (XSD Document Version 3.2.1 Revision 0003, xmlns="http://autosar.org/3.1.2")
sysd:3.1.4	Imports files using the AUTOSAR XSD version 3.1 (XSD Document Version 3.3.0 Revision 0004, xmlns="http://autosar.org/3.1.4")
sysd:3.1.4.DAI.2	Imports files using the AUTOSAR XSD version 3.1 (XSD Document Version 3.3.0 Revision 0004 (Daimler specific version 2), xmlns="http://autosar.org/3.1.4.DAI.2")

Content type	Description
sysd:3.1.4.DAI.3	Imports files using the AUTOSAR XSD version 3.1 (XSD Document Version 3.3.0 Revision 0004 (Daimler specific version 3), xmlns="http://autosar.org/3.1.4.DAI.3")
sysd:3.1.4.DAI.4	Imports files using the AUTOSAR XSD version 3.1 (XSD Document Version 3.3.0 Revision 0004 (Daimler specific version 4), xmlns="http://autosar.org/3.1.4.DAI.4")
sysd:3.2.1	Imports files using the AUTOSAR XSD version 3.2 (XSD Document Version 3.5.0 Revision 0001, xmlns="http://autosar.org/3.2.1")
sysd:4.0.2	Imports files using the AUTOSAR XSD version 4.0.2 (XSD Document Version 4.1.0 Revision 0002, xmlns="http://autosar.org/r4.0")
sysd:4.0.3	Imports files using the AUTOSAR XSD version 4.0.3 (XSD Document Version 4.2.1 Revision 0003, xmlns="http://autosar.org/r4.0")
sysd:4.1.0	Imports files using the AUTOSAR XSD version 4.1.0 (snapshot 2012-08-09, XSD Document Version 4.3.0 Revision 0000, xmlns="http://autosar.org/r4.0")

4.5.3.4. Content types for tresosDB files

The tresosDB format is the internal storage format for system description data. The tresosDB format is also used to exchange system description data with other tools.

There is no need to specify a version number when importing or exporting tresosDB files. During import, the version of the file is automatically detected and the data is converted to the most current version. Thus it is only possible to export a tresosDB file of the most current version.

Content type	Description
tdb	Default content type for a tresosDB file, at the moment tdb:3.1
tdb:3.1	Content type for a tresosDB file representing Autosar 3.1.

4.5.3.5. Content types for DBC files

DBC files can be imported by EB tresos Studio and converted into a tresosDB ([Section 4.5.3.4, “Content types for tresosDB files”](#))

It is not necessary to distinguish different versions of CANdb files, thus it is not necessary to provide a version number.

Content type	Description
dbc	DBC (CANdb) file

4.5.3.6. Content types for LDF files

LDF files can be imported by EB tresos Studio and converted into a tresosDB file ([Section 4.5.3.4, “Content types for tresosDB files”](#)).

EB tresos Studio supports the following LDF file versions:

Content type	Description
1df	Default version for LDF files, at the moment 1df:2.0
1df:2.0	LDF version 2.0

4.5.3.7. Content types for Fibex files

Fibex files can be imported by EB tresos Studio and converted into a tresosDB file ([Section 4.5.3.4, “Content types for tresosDB files”](#)).

While EB tresos Studio supports different versions of the Fibex format (see [Section 7.10.2.8, “Importing from the Fibex format”](#)), it is not necessary to specify one with the content type. The parser automatically detects, which supported version is used in the files.

Content type	Description
fibex	Fibex file, version is auto-detected

4.6. 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.

5. Using EB tresos Studio for the first time

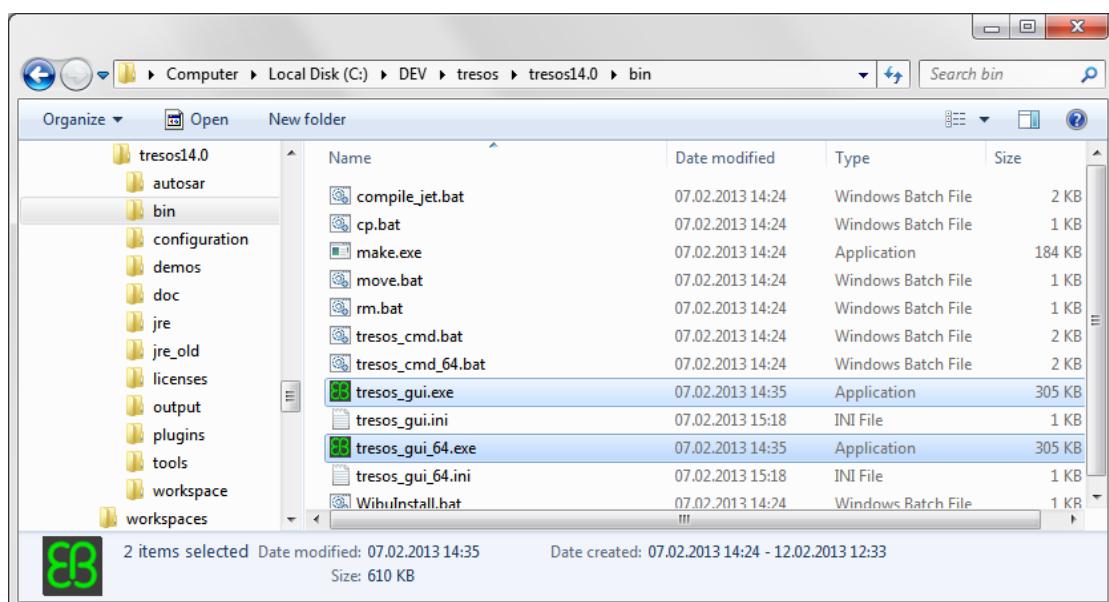
5.1. Starting the graphical user interface

By default, the EB tresos Studio installer creates an icon on your desktop and adds an entry in the Windows Start menu. Both are labeled *tresos Studio*. You can use either to start EB tresos Studio. The icon looks like this:



If you do not see such an icon:

- ▶ Locate the root directory of the EB tresos Studio installation `$TRESOS_BASE`.



- ▶ In the `$TRESOS_BASE`, locate `bin\tresos_gui.exe`.
- ▶ Double-click `bin\tresos_gui.exe` or `tresos_gui_64.exe`.

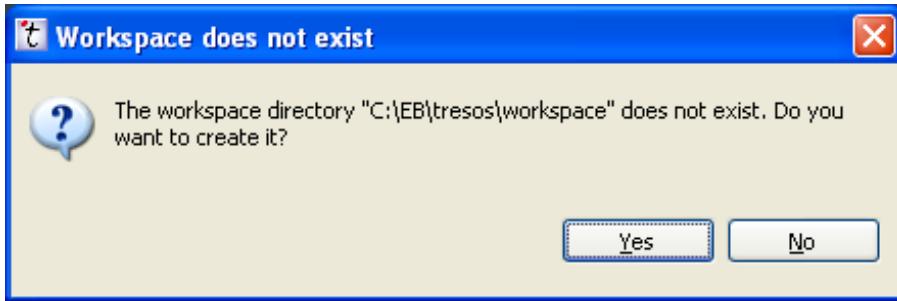
The GUI is being started.

Alternatively you can also double-click the `bin\tresos_gui_64.exe` to start EB tresos Studio as an 64bit application.



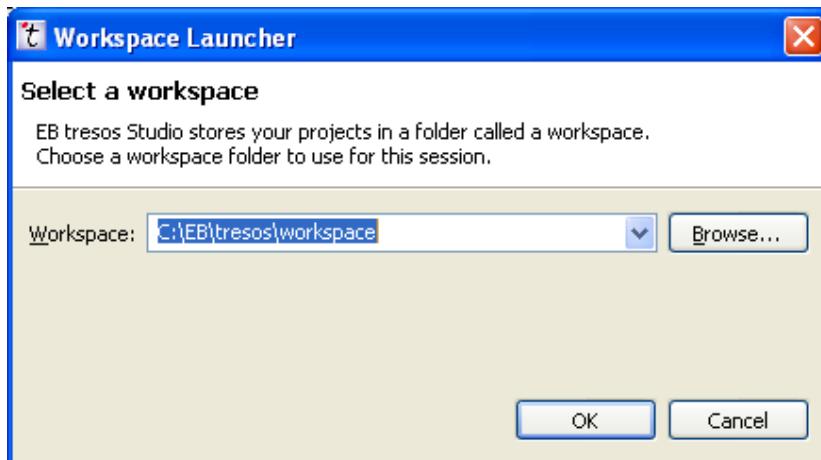
5.2. Selecting the workspace location

The workspace is a directory where projects and settings are stored. If you start EB tresos Studio for the first time, you are asked whether you want to create the default workspace. The default workspace is the directory `workspace` within the `$TRESOS_BASE` directory. `$TRESOS_BASE` is the location at which you installed EB tresos Studio.



To select your workspace location:

- ▶ You can either accept the default workspace by pressing **Yes** or open a workspace selection dialog by pressing **No**.
 - ▶ If you press **Yes**, the default workspace is created at `$TRESOS_BASE/workspace`.
 - ▶ If you press **No**, the following dialog opens up:



- ▶ If you want to create your own workspace path, insert a path to a workspace of your choosing and press **OK**.
- After pressing **OK**, the selected workspace is created and the EB tresos Studio main window appears.
- ▶ If you want to exit the startup process, press the **Cancel** button.

TIP**How to change your workspace at a later time**

Once a workspace is created EB tresos Studio opens this workspace at startup. You can change the workspace afterwards with the *Switch Workspace* action under the *File* menu.

TIP**If your workspace does not exist or is being used**

Any time the workspace does not exist or is already in use by another instance of EB tresos Studio you will be asked for creating or changing the workspace at startup.

5.3. The welcome page

The graphical user interface starts up with a *Welcome* view on the first launch:



On the Welcome view you can find the version of the EB tresos Studio release you are using and the name of the license (if a license is already installed). The view also displays some hints for first-time users and provides shortcuts to common functions, such as creating a new project or opening the license dialog.

To start working with EB tresos Studio, close the Welcome view. To do so, click on the **Close this page** link at the bottom of the page, or click the **x** icon next to the *Welcome* title.

If you want to open the view again, choose the *Welcome* item from the **Help** menu.



After closing the view, you see the EB tresos Studio main window. The contents of this window are explained in [Chapter 6, The graphical user interface \(GUI\)](#).

6. The graphical user interface (GUI)

This chapter is going to introduce you to the main window of EB tresos Studio, also called the *workbench*. You will get to know how the workbench is organized and what functions are available through the menus and the tool bars.

6.1. Using the Workbench

The graphical user interface (GUI) of EB tresos Studio consists of views that are arranged in the workbench, e.g. the *Project Explorer*. An arrangement of views is called a *perspective*. Perspectives are either pre-defined or user-defined.

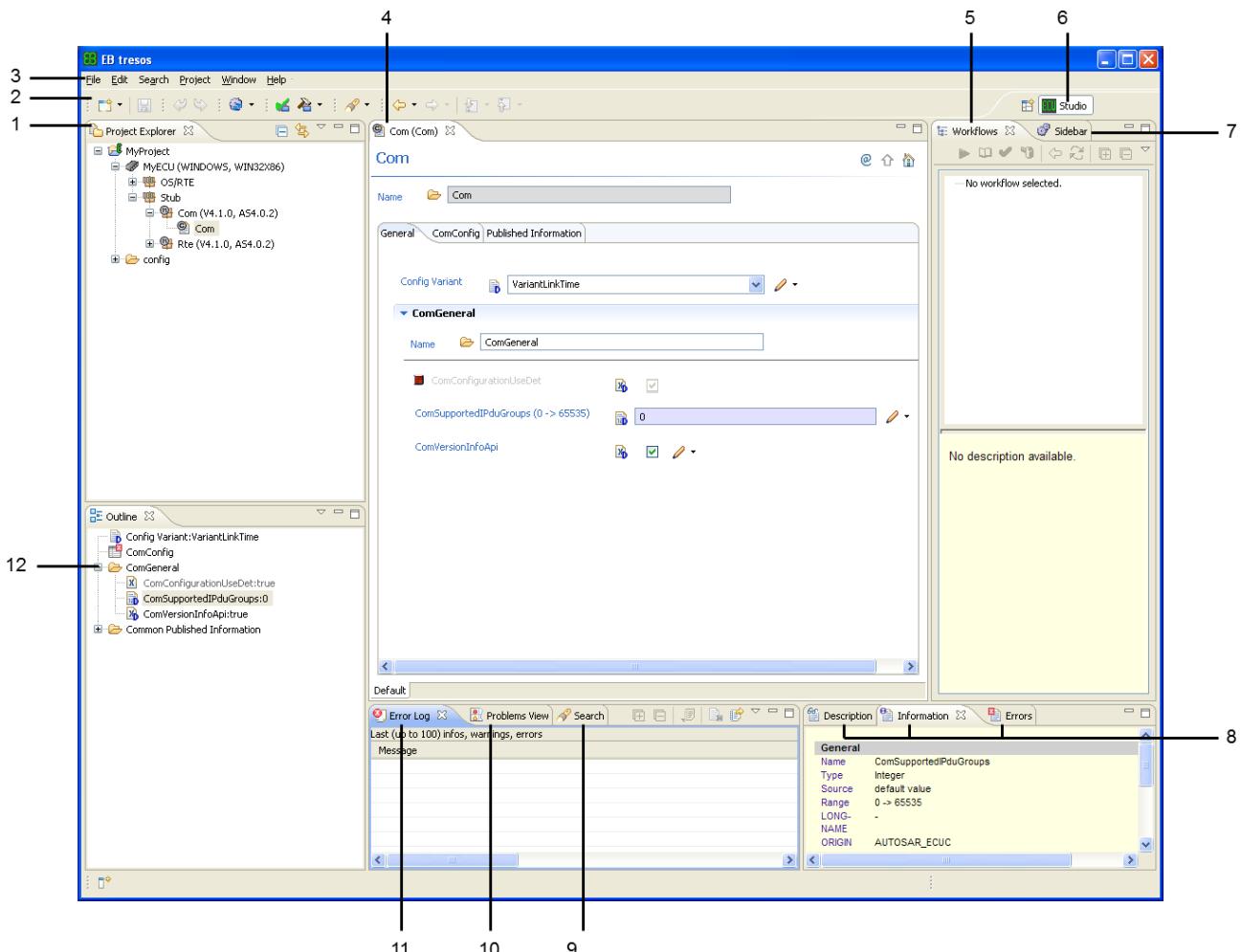


Figure 6.1. The EB tresos Studio Workbench



Number	Element	Description
1	<i>Project Explorer</i>	The <i>Project Explorer</i> displays all EB tresos Studio configuration projects in a workspace (see Section 4.5, “EB tresos Studio projects and workspaces”) and allows you to edit projects and open editors. The default location of the <i>Project Explorer</i> view is on the left of the screen.
2	Tool bar	The tool bar is located in the upper left corner of the screen.
3	Menu bar	The menu bar is located in the upper left corner of the screen. In addition to the default menu items, there may additional menu items present. These are contributed by the additional plug-ins and modules you have installed and should be documented there.
4	<i>Editor</i> view	The <i>Editor</i> view is in the center of the workbench. It contains editors, with which you may edit the data of a project.
5	<i>Workflows</i> view	The <i>Workflows</i> view displays workflows which guide you to accomplish a certain task, e.g. to configure a Com stack. The default location is in the upper right corner of the screen.
6	Perspectives bar	The perspectives bar in the upper right corner of the screen is an element for handling workbench perspectives, i.e. opening, closing, switching, customizing, saving and resetting perspectives.
7	<i>Sidebar</i> view	The <i>Sidebar</i> view displays assistance dialogs for your current task. Per default the <i>Sidebar</i> view is located in the upper right corner of the screen, as a tab behind the <i>Workflows</i> view.
8	<i>Element Outline</i>	The <i>Element Outline</i> is on the right bottom of the workbench; it consists of multiple views that show information about the parameter currently selected in the Generic Configuration Editor.
9	<i>Search</i> view	The <i>Search</i> view enables searching and finding files, specific configuration parameters, or text strings.
10	<i>Problems View</i> view	The <i>Problems</i> view displays the messages found in the configuration of a loaded project. The default location of the <i>Problems</i> view is on the bottom of the screen.
11	<i>Error Log</i> view	The <i>Error Log</i> view is the place in the GUI where all errors are logged. The <i>Error Log</i> receives error messages from EB tresos Studio components and from other Eclipse components. The list of messages in the <i>Error Log</i> is preserved in between the sessions of EB tresos Studio.

Number	Element	Description
		The default location of the <i>Error Log</i> view is on the bottom of the screen.
12	<i>Node Outline</i> view	The <i>Node Outline</i> view on the left bottom of the workbench, which shows a tree view of the editable configuration parameters that you may edit in the <i>Editor</i> view.

6.1.1. Customizing the workbench

When you open EB tresos Studio for the first time, it opens up with a default layout for the views, namely the *Studio perspective* (see also [Section 6.1.2, “Using perspectives”](#)). The Studio perspective composes the views you need to edit ECU configuration projects. The layout is arranged for medium sized screens. You might want to change this layout so that it better suits your task. To do so, you may rearrange the perspective via drag and drop, stack the views as tabs over each other or maximize, minimize and close the views.

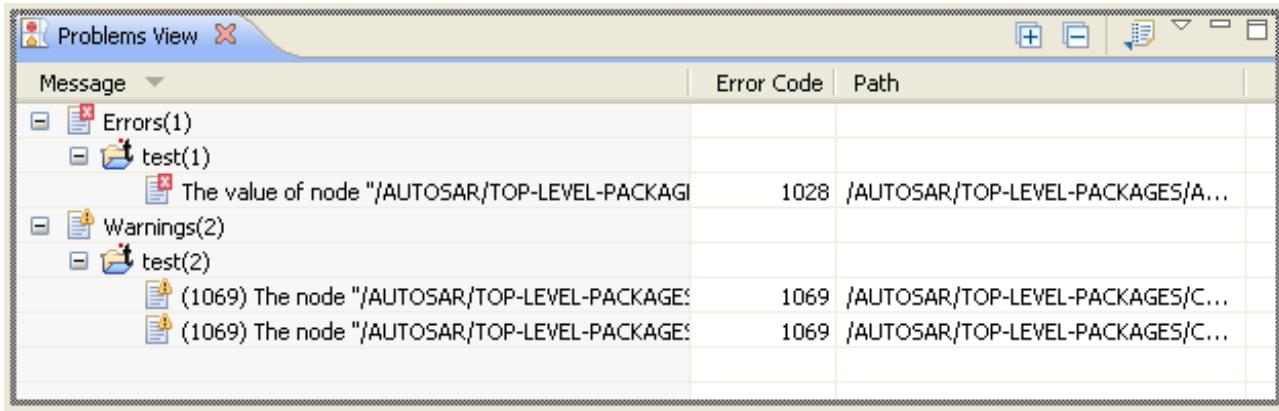
NOTE

Not all views are visible

Some views are not visible by default, but they appear when they are needed. The *Search* view, for example, is usually not visible. Only if you have started a search, the view appears and displays the search results to you.

6.1.1.1. Moving views to a different location

Every view has a title area. If you click this area, keep the mouse button pressed and move around you can drag the view to a different location. The location to which the view will be placed when you release the mouse button, is indicated by a gray rectangle.



If you drop the view at a place where already another view is present, the views will be stacked. This means that there is always one view visible and the other view is behind it. The title areas of the views then work like tabs and you can move the other view to the foreground by clicking on its title area.

TIP**You may move a whole stack of views**

In the default window layout, the *Element Outline* views (see [Section 6.3.4, “Element Outline views”](#)) are an example for stacked views. You can also move a whole stack of views. To do that, drag the view stack at the empty area next to rightmost view title and place it to the location of your choice.

If you drop a view outside of the workbench, it is detached from the workbench and becomes a window of its own. To reverse this, drag it at its title area back into the workbench.

For editors, some extra rules apply:

- ▶ You cannot stack editors together with standard views and vice versa.
- ▶ Editors can only reside in the *Editor* view.
- ▶ You cannot detach editors from the workbench.

6.1.1.2. Closing views, and minimizing or maximizing view stacks



Number	Button	Description
1	Close	Closes the view so that it disappears completely. If you want to re-open the view again, continue reading at Section 6.1.1.3, “Opening a view” .
2	Minimize	Minimizes this view or view stack. Minimized views occupy a small space at the border of the window. In this place only the icons of the minimized view stack are shown and you can click them to restore the views again. This is useful if you temporarily want to hide a view, e. g. to have more space for other views.
3	Maximize	Maximizes the view. The view will occupy most of the workbench space. This is useful if you temporarily want to focus on one view only. E. g. sometimes you might want to maximize the <i>Problems</i> view in order to see the complete message text of all configuration problems.

To restore maximized or minimized views or view stacks, click on the **Restore** button . The **Restore** button is located either near the icons of the minimized view, or as a replacement of the **Maximize** button, if the view has been maximized.

TIP**Double-click the title area of a view to restore or maximize it**

To maximize a view, or to restore a maximized view, you can also double-click its title area.

6.1.1.3. Opening a view

Sometimes you might want to open a new view or a view you have closed before.

To open a view:

- ▶ In the **Window** menu, select *Show View*.
- ▶ A submenu opens up.

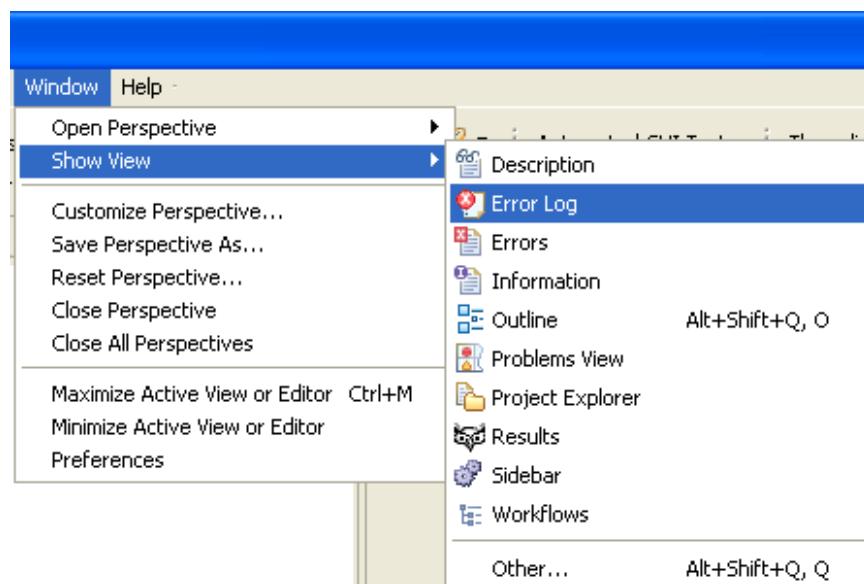


Figure 6.2. Opening a view that is not displayed in the workbench

- ▶ Select one of the views listed or select *Other...* to open the *Show View* dialog.

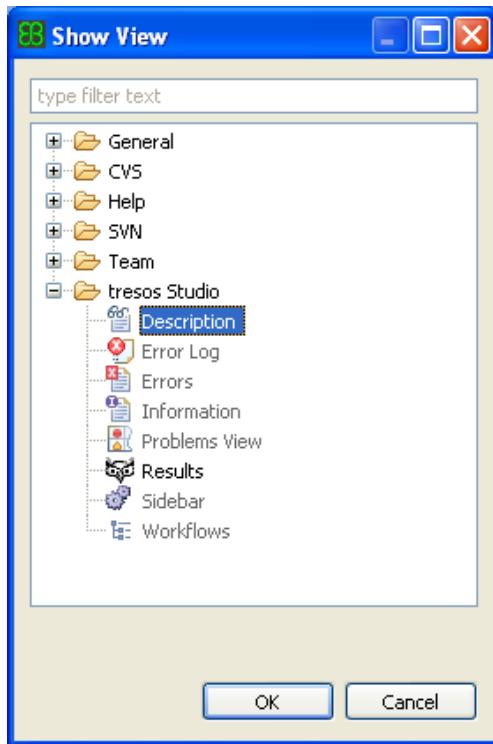


Figure 6.3. Selecting a view to open

- ▶ Select one or more views to open and click **OK**.

Note that views that are already open, are displayed in gray font in the *Show View* dialog.

- ▶ The selected views open at their last position.

6.1.2. Using perspectives

Tightly related to workbench customizing (see [Section 6.1.1, “Customizing the workbench”](#)) is the use of workbench perspectives. Perspectives are named arrangements of views and editors, together with menu and tool bar actions.

There are pre-defined perspectives, such as the Studio perspective, and user-defined perspectives, in which you may store your customized workbench layouts.

NOTE**Only one perspective can be active**

While there can be more perspectives opened at a time, only one perspective can be active.

6.1.2.1. Saving user-defined perspectives

You can store custom layouts of your workbench into named perspectives.

To save the current perspective layout into your user-defined perspective:

- ▶ In the **Window** menu, select *Save Perspective As...*
- ▶ The *Save Perspective As...* dialog opens.

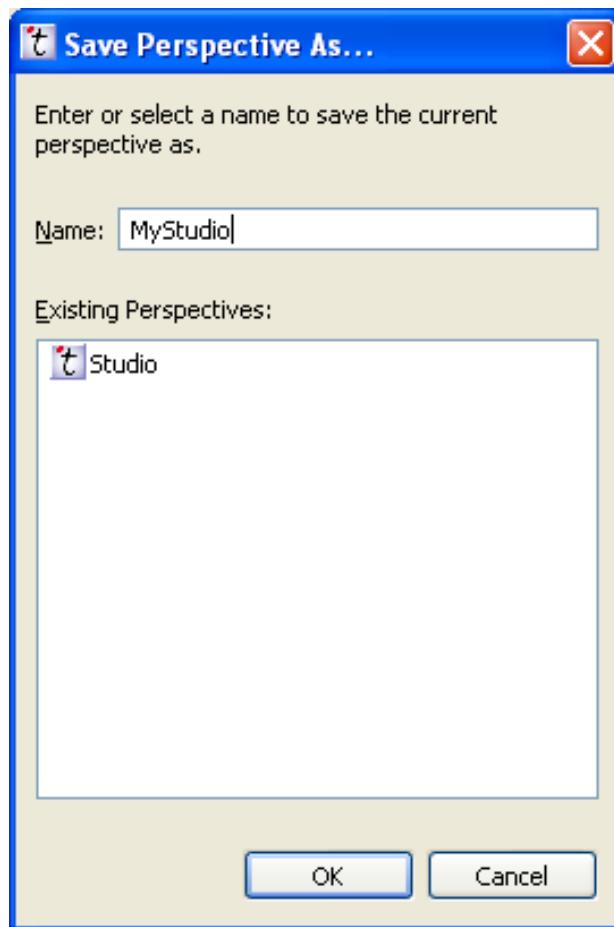


Figure 6.4. Saving your customized perspective

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

-
- WARNING** **Do not store your perspective with the name of a pre-defined perspective**
-  It is technically feasible to save your customized perspective with the name of an existing pre-defined perspective. However, this is not recommended, as you then won't be able to restore the pre-defined perspective anymore.
-

6.1.2.2. Opening perspectives

To open a perspective:

- ▶ In the **Window** menu, select **Open Perspective** or click on the **Open Perspective** button  in the perspective bar.
- ▶ A context menu opens up.



Figure 6.5. Opening a perspective

- ▶ Select one of the perspectives listed. If the desired perspective is not listed in the context menu, select **Other....**

The *Open Perspective* dialog opens up.

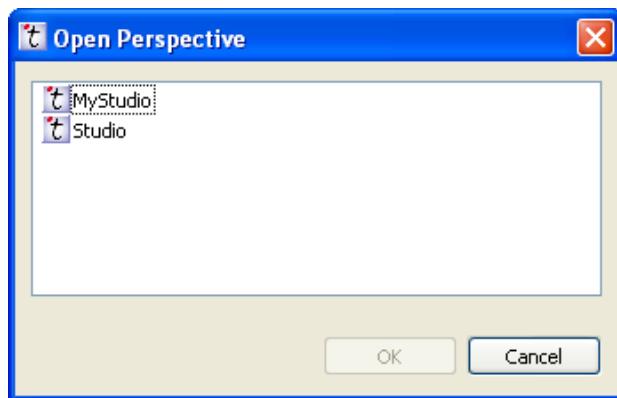


Figure 6.6. Selecting a perspective to open

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

6.1.2.3. Switching between perspectives

To change the currently active perspective to another open one:

- ▶ Click on the perspective icon or name you want to switch to in the perspective bar.



Figure 6.7. Switching perspectives

- ▶ The currently active perspective is changed.

6.1.2.4. Closing perspectives

To close the current perspective:

- ▶ In the **Window** menu, select *Close Perspective*.
- ▶ The current perspective closes.

To close any open perspective:

- ▶ In the perspective bar, right-click on the icon of the perspective you want to close.

A context menu opens up.

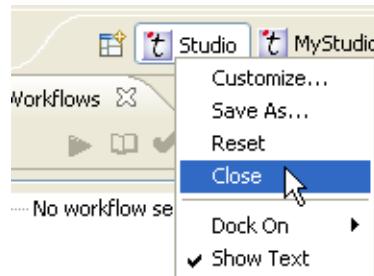


Figure 6.8. Closing perspectives

- ▶ The selected perspective closes.

If another perspective is open, it becomes activated. If you close the last open perspective, the workbench is emptied.

6.1.2.5. Resetting perspectives

You may reset a perspective to its default state, i.e. when you want to undo the last changes you made in the layout.

To reset the current perspective:

- ▶ In the **Window** menu, select *Reset Perspective* or right-click on the current perspective in the perspective bar.

A context menu opens up.

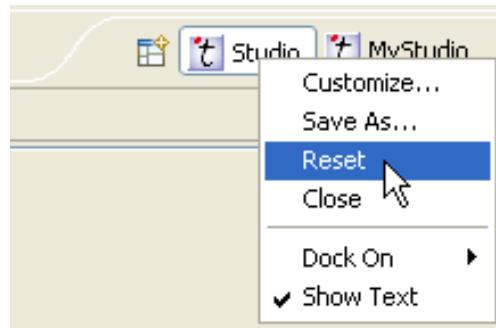


Figure 6.9. Resetting perspectives

- ▶ In the context menu, select *Reset*.
- ▶ The current perspective is reset, either to its pre-defined or to its last saved state.

6.1.2.6. Deleting perspectives

You may also delete your user-defined perspectives. To delete a user-defined perspective:

- ▶ In the **Window** menu, select *Preferences...*
- ▶ The *Preferences* dialog opens up.
- ▶ In the left tree, navigate to the *General/Perspectives* page.
- ▶ Locate the section *Available perspectives*:

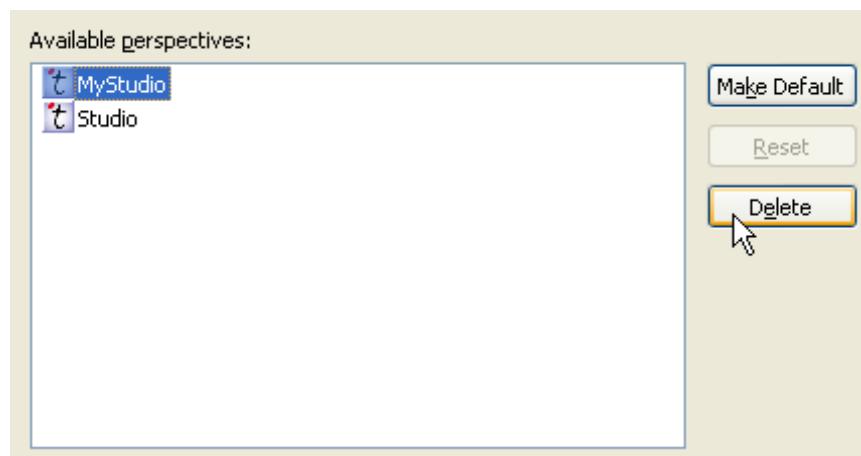
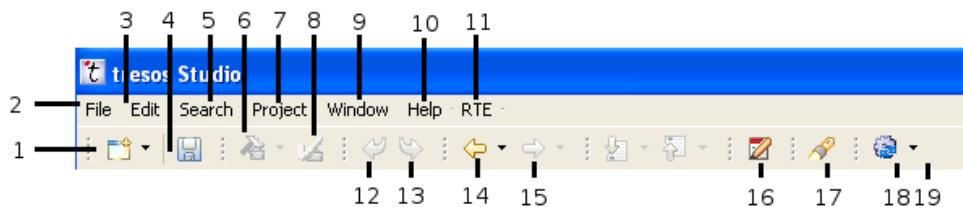


Figure 6.10. Deleting perspectives in the *Preferences* dialog

- ▶ Select the user-defined perspectives you want to delete and click **Delete**.
- ▶ Click **OK** in the *Preferences* dialog.
- ▶ The selected user-defined perspectives are deleted.

6.2. Menu bar and tool bar

The menu bar and the tool bar are located in the upper left corner of the screen:

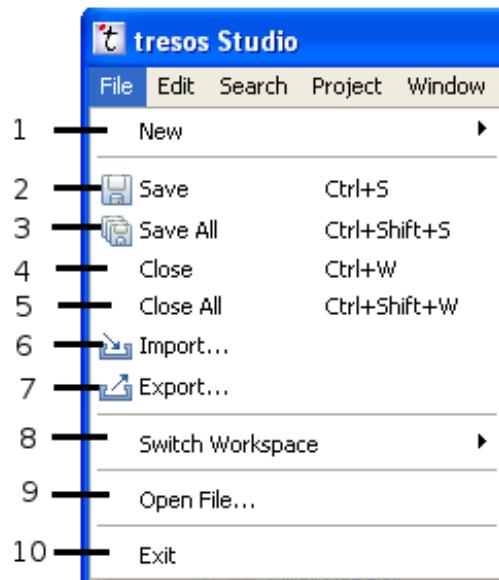


Number	Menu	Description
1	New project wizard icon	Select to open a new project wizard.
2	File menu	Select for project unspecific tasks.
3	Edit menu	Select for tasks related to editing project data.
4	Save configuration project icon	Select to save the configuration project of the currently active editor (see Section 7.3.2, “Saving a project”).
5	Search menu	Select to search for and to find files, specific configuration parameters, or text strings
6	Generate code for the currently selected project icon	Select to start generating code, see details in Section 7.9, “Generating code for projects” .
7	Project menu	Select to verify and generate code, see details in Section 7.9, “Generating code for projects” or to configure and run unattended wizards, see details in Section 7.8.7, “Autoconfiguring routine jobs” .
8	Verify selected project icon	Select to start verifying code, see details in Section 7.9.1, “Verifying a project”
9	Window menu	Select to set preferences and to choose perspectives and views, see details in Section 7.3.6, “Closing and opening projects” , Section 6.1.1, “Customizing the workbench” and Section 6.1.2, “Using perspectives” .
10	Help menu	Select to access all global information and help functions for EB tresos Studio, see details in Section 7.2.2, “Viewing setup information” .

Number	Menu	Description
11	<i>RTE</i> menu	This item is only available if you have an <i>Rte</i> module installed. It enables you to start the <i>Contract Phase</i> . You will find more information about this item in the user manual of the <i>Rte</i> module.
12	<i>Undo</i> icon	Select to undo the last edit. The button is active only if undoable change options are available.
13	<i>Redo</i> icon	Select to redo the last undone edit. The button is active only if redo change options are available.
14	<i>Back</i> icon	Select to move to the last editing position.
15	<i>Forward</i> icon	Select to move forward to the next editing position. This option is only available if the Back button or the corresponding menu was selected previously.
16	<i>RTE Contract Phase generation</i> icon	This item is only available if you have an <i>Rte</i> module installed. Select to open the <i>Please select Software Component Description files</i> window to start the RTE Contract Phase. You will find more information about this item in the user manual of the <i>Rte</i> module.
17	<i>Search</i> icon	Select to open the <i>Search</i> window default page, which is currently the <i>File Search</i> page.
18	<i>Run all unattended wizards marked with a '*' icon</i>	Select to run all <i>Unattended Wizard(s)</i> for a specified project, which are selected to run manually.
19	<i>Arrow down</i> icon	Select to configure or run unattended wizards, see details in Section 7.8.7, “Autoconfiguring routine jobs” .

6.2.1. File menu

The *File* menu is located on the far left of the EB tresos Studio main window.

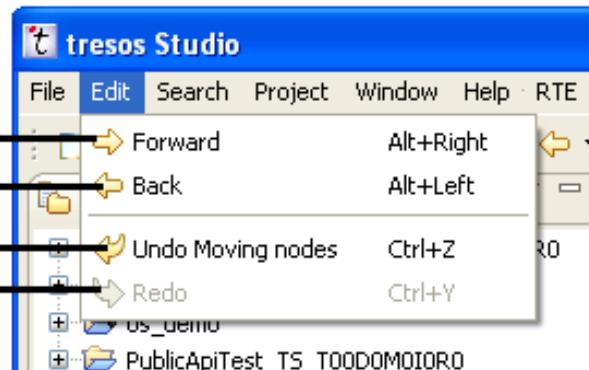


Number	Menu	Description
1	New	Select <i>Project</i> in the sub menu to create a new EB tresos Studio project Section 7.3.1, “Creating a new project” .
2	Save	Select to save the content of the currently active editor. If the editor is a configuration editor, the associated configuration project is saved Section 7.3.2, “Saving a project” .
3	Save all	Select to save the content of all open editors.
4	Close	Select to close the currently active editor. If you have modified any content since saving, a pop-up window is going to ask you to save unsaved content (see Section 7.3.6.1, “Closing projects”).
5	Close all	Select to close all open editors. If content of one or more editors is unsaved, a pop-up window is going to ask you to save unsaved content.
6	Import	Select to import projects (see Section 7.10.1.1, “Importing a project”).
7	Export	Select to export projects (see Section 7.10.1.2, “Exporting a project”).
8	Switch Workspace	Select to select a new workspace directory. If you select a new workspace directory, EB tresos Studio shuts down and restarts with the new workspace. See the concept of workspaces Section 4.5, “EB tresos Studio projects and workspaces” .

Number	Menu	Description
9	<i>Open File</i>	Select to open a file.
10	<i>Exit</i>	Select to close EB tresos Studio. A pop-up window is going to ask you if you want to save any changes before closing the tool.

6.2.2. Edit menu

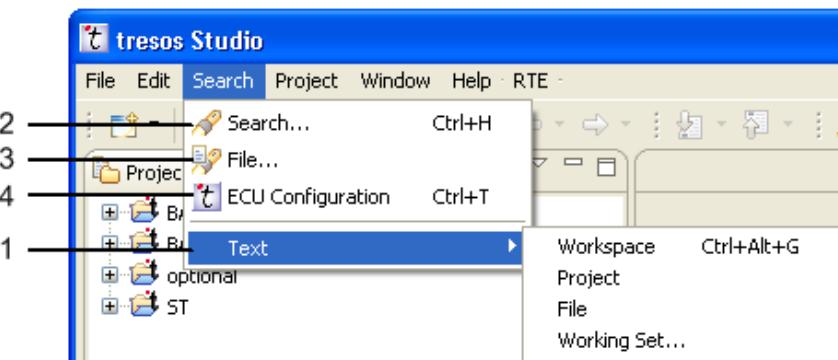
The *Edit* menu is the second drop down menu from the left. The menu items in this menu are active only if you are working with an editor.



Number	Menu	Description
1	<i>Forward</i>	Select to move to the next editing position in the editor.
2	<i>Back</i>	Select to move to the previous editing position you worked on in the editor.
3	<i>Undo ...</i>	Select to undo the last change. Instead of "..." the name of the action that would be undone is displayed.
4	<i>Redo ...</i>	Select to redo the last change. Instead of "..." the name of the action that would be redone is displayed.

6.2.3. Search menu

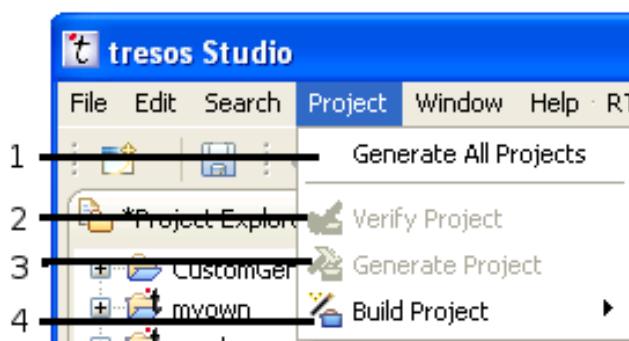
The *Search* menu enables you to search for and to find files, specific configuration parameters, or text strings.



Number	Menu	Description
1	<i>Text</i>	Select to search for text strings. As search-pattern, the currently selected text is used. This menu option will open a submenu with different search scope options.
2	<i>Search</i>	Select to open the <i>Search</i> window, which is a dialog with a tab for every different type of search. If this dialog is invoked from a configuration editor, the <i>ECU Configuration Search</i> is active directly when the dialog is opened.
3	<i>File</i>	Select to open the file search. Within the file search you may search for files with a specific name or content.
4	<i>ECU Configuration</i>	Select to open the configuration parameter search within open EB tresos Studio projects. Instead of clicking the menu item, an alternate way of opening the dialog is to press Ctrl+T .

6.2.4. Project menu

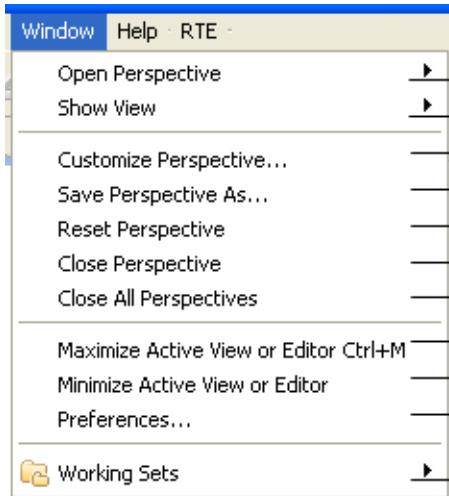
The *Project* menu is the fourth drop down menu from the left.



Number	Menu	Description
1	<i>Generate All Projects</i>	Select to generate the code for all projects Section 7.9.2, “Generating a project” .
2	<i>Verify Project</i>	Select to verify the project currently selected in the Project Explorer Section 7.9.1, “Verifying a project” . If no project is selected in the <i>Project Explorer</i> , this entry is grayed out.
3	<i>Generate Project</i>	Select to generate the project that is currently selected in the <i>Project Explorer</i> (see Section 7.9.2, “Generating a project”). If no project is selected in the <i>Project Explorer</i> , this entry is grayed out.
4	<i>Build Project</i>	Select to open a submenu that contains a list of generation modes Section 7.9.3, “Building a project” . The available generation modes are determined by the modules of the currently selected project. Please refer to the documentation of your installed modules. If no project is selected in the <i>Project Explorer</i> , this entry is grayed out.

6.2.5. Window menu

The *Window* menu is the fifth drop down menu from the left.

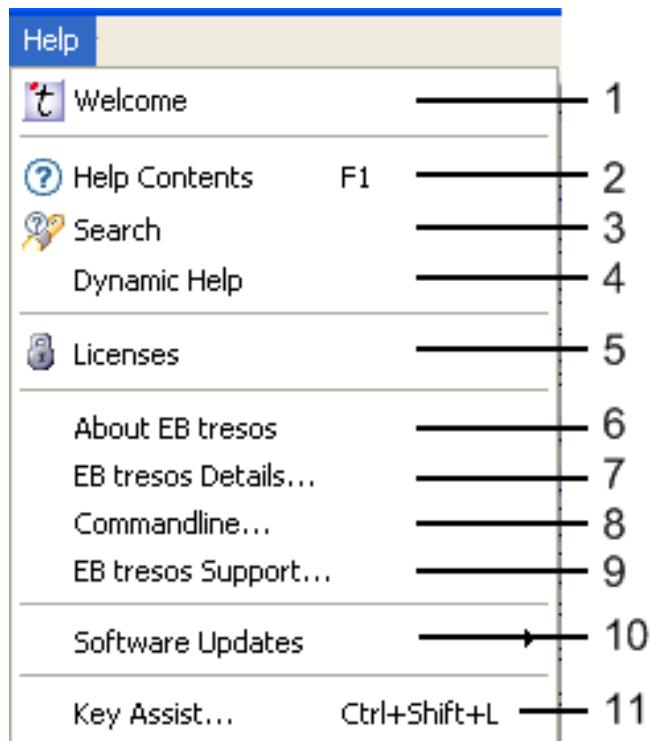


Number	Menu	Description
1	<i>Open Perspective</i>	Select <i>Open Perspective</i> , then select <i>Other..</i> to open a perspective. A perspective is an arrangement of views. See also Section 6.1.2.2, “Opening perspectives” .

Number	Menu	Description
2	<i>Show View</i>	Select <i>Show View</i> , then select <i>Other</i> to reopen a view. A view in this context is a part of a window such as the <i>Project Explorer</i> .
3	<i>Customize Perspective...</i>	Currently not used by EB tresos Studio.
4	<i>Save Perspective As...</i>	Select to store the current workbench layout as a named perspective so that you can restore the layout later on. See also Section 6.1.2.1, “Saving user-defined perspectives” .
5	<i>Reset Perspective</i>	Select to reset the current perspective to its pre-defined or saved state. See also Section 6.1.2.5, “Resetting perspectives” .
6	<i>Close Perspective</i>	Select to close the current perspective. If you close the last open perspective, the workbench is emptied. See also Section 6.1.2.4, “Closing perspectives” .
7	<i>Close All Perspectives</i>	Select to close all perspectives and empty the workbench.
8	<i>Maximize Active View or Editor</i>	Select to enlarge the currently selected view to its maximum. This means that all other views will be minimized and may disappear from view.
9	<i>Minimize Active View or Editor</i>	Select to minimize the currently active view. This means that the active view may disappear from view and just the other views are displayed.
10	<i>Preferences</i>	Select to change the EB tresos Studio preferences Section 7.2.3, “Setting preferences” .
11	<i>Working Sets</i>	Currently not used by EB tresos Studio.

6.2.6. Help menu

The *Help* menu provides information about the tool, software updates, and licenses.



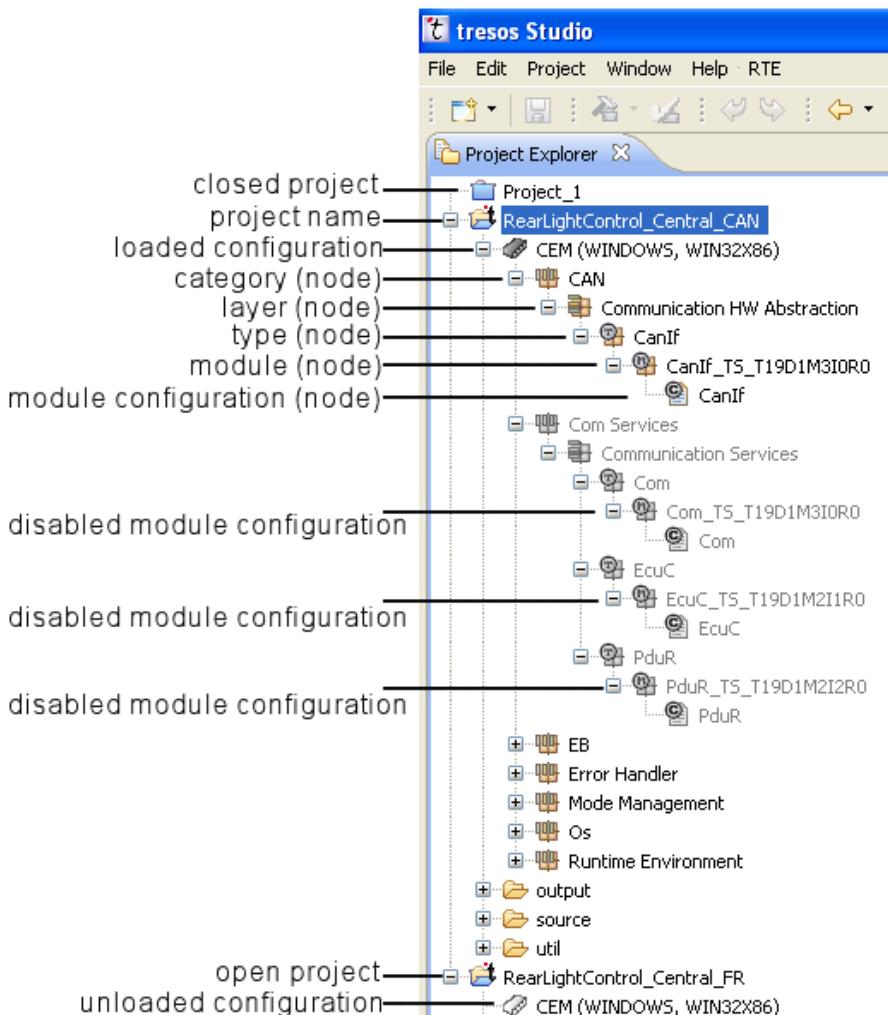
Number	Menu option	Description
1	<i>Welcome</i>	Select to reopen the welcome screen.
2	<i>Help Contents</i>	Select to open the help window and browse the online help function of EB tresos Studio.
3	<i>Search</i>	Select to search the online help of EB tresos Studio.
4	<i>Dynamic Help</i>	Currently not supported in EB tresos Studio.
5	<i>Licenses</i>	Select to open the license dialog (see Section 7.2.1, “Managing licenses”).
6	<i>About EB tresos</i>	Select to access information about the EB tresos home page and support page.
7	<i>EB tresos Details</i>	Select to access detailed information about EB tresos Studio, to list the variant, the release version, the build number, the install location, the workspace location, all features, all modules and all bundles.
8	<i>Command line</i>	Select to browse the EB tresos Studio command line options and view the command line options help messages.
9	<i>EB tresos Support...</i>	Opens the EB tresos support web page in an Internet browser.
10	<i>Software Updates</i>	Currently not supported in EB tresos Studio.

Number	Menu option	Description
11	Key Assist	Currently not supported in EB tresos Studio.

6.3. Views

6.3.1. Project Explorer view

The *Project Explorer* is the main utility to work with EB tresos Studio projects. The *Project Explorer* displays all EB tresos Studio configuration projects in a workspace and allows you to open editors. The project explorer displays a tree. The tree starts with the projects on the first level. The second level contains the ECU configuration node. Beneath that level, the tree continues with several layers down to the module configurations. All items of the tree are represented by different icons. The items may change their state which is also reflected by their icon. The following picture gives you a comprehensive overview:



NOTE

A project which contains an ECU configuration usually is called a *configuration project* for short.



6.3.1.1. Project and ECU configuration states

Projects and ECU configurations can be in different states each. A project can be either open or closed. An ECU configuration can either be loaded or unloaded.

The following table gives an overview over the different project states and explains their use:

Project State	Icon	Description
open		In this state the <i>Project Explorer</i> displays the ECU configuration node for this project. This is the default state for projects after start-up.
closed		In a closed project you cannot edit any configurations, nor can you generate any code. The project consumes the least possible resources.

The following table gives an overview over the different ECU configuration states and explains their use:

ECU Configuration State	Icon	Description
unloaded		In an unloaded ECU configuration you can view the configuration's properties such as the ECU Id or the target and derivate values, but you cannot edit them. The ECU configuration data is not loaded and therefore you can neither edit configurations, nor can you generate any code in this state. The benefit of this state is its very low resource consumption. This state is the default state for projects when EB tresos Studio is started.
loaded		A loaded ECU configuration is fully functional. Loaded configurations can be edited, modules can be added and code can be generated. Beneath the ECU configuration node, a hierarchical configuration tree displays configurations grouped according to categories. Newly created projects are in this state.

6.3.1.2. Layering and grouping of module configurations

The *Project Explorer* groups module configurations according to the modules they belong to. The modules themselves are grouped into hierarchical layers of categories.

Layer name	Icon	Description
Loaded configuration		Represents the loaded configuration of a project. icon in the shape of a chip
Category node/layer		Provides a horizontal grouping of modules within an operating system stack. icon contains three vertically ordered bars
Layer node/layer		Provides a vertical grouping of modules within an operating system stack. icon contains three horizontally ordered bars
Type node/layer		Describes the functionality of the module, e.g. Can for a CAN driver. icon with a <i>T</i> on the folder
Module node/layer		Is the actual implementation of a module, such as a CAN driver for a specific target and derivate. icon with an <i>M</i> on the folder
Module configuration node/layer		Is the configuration for an actual implementation of a module (see module node); the leaf node represents either: <ul style="list-style-type: none">▶ a module configuration or▶ an editor if the module provides multiple editors for configuration.

6.3.1.3. Changing the visible layers

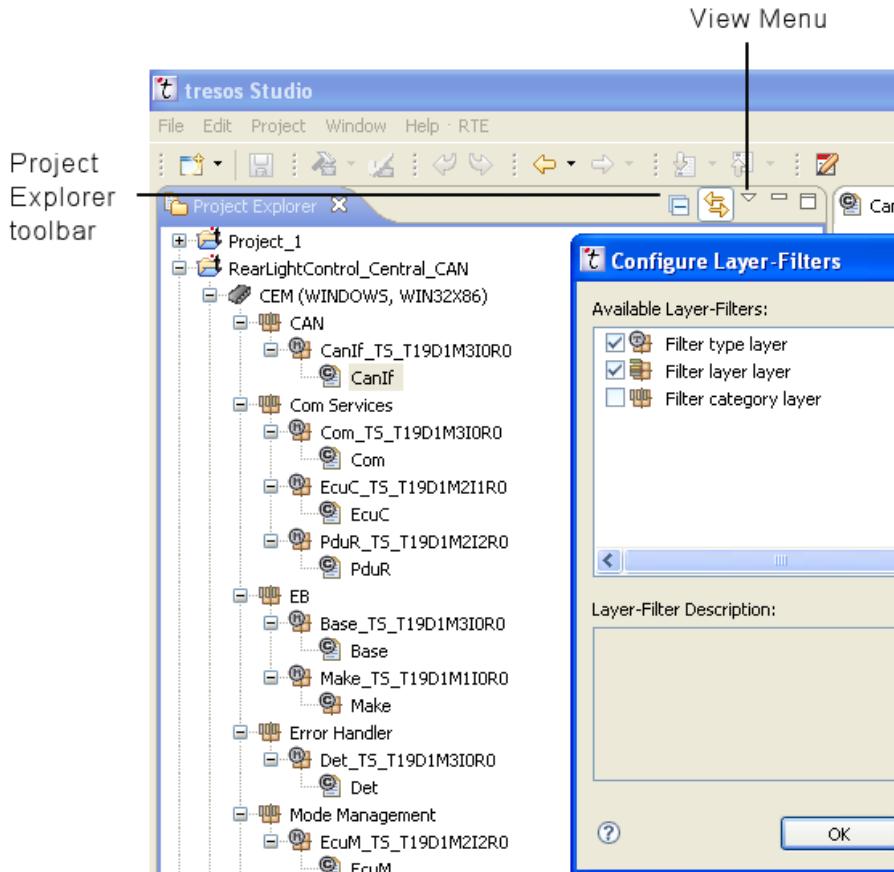
You can customize the layer grouping of the *Project Explorer* by selecting which layers shall be displayed. To change this:

- ▶ Select the **View Menu** button in the Project Explorer tool bar.

The **View Menu** menu opens up.

- ▶ Select the *Layer Filters* menu option.

The *Configure Layer Filters* window opens up.



If a layer is disabled, it is excluded from the tree. The content beneath the hidden nodes however, is not hidden. Instead, the content of the hidden nodes is displayed one level higher than it normally would have been. Nodes are grouped together if their parents are the same and if they share the same label.

The *Configure Layer Filters* window provides check boxes to disable layers.

To disable the *category* layer

- ▶ Select the box *Filter category layer*

To disable the *layer* layer

- ▶ Select the box *Filter layer layer*

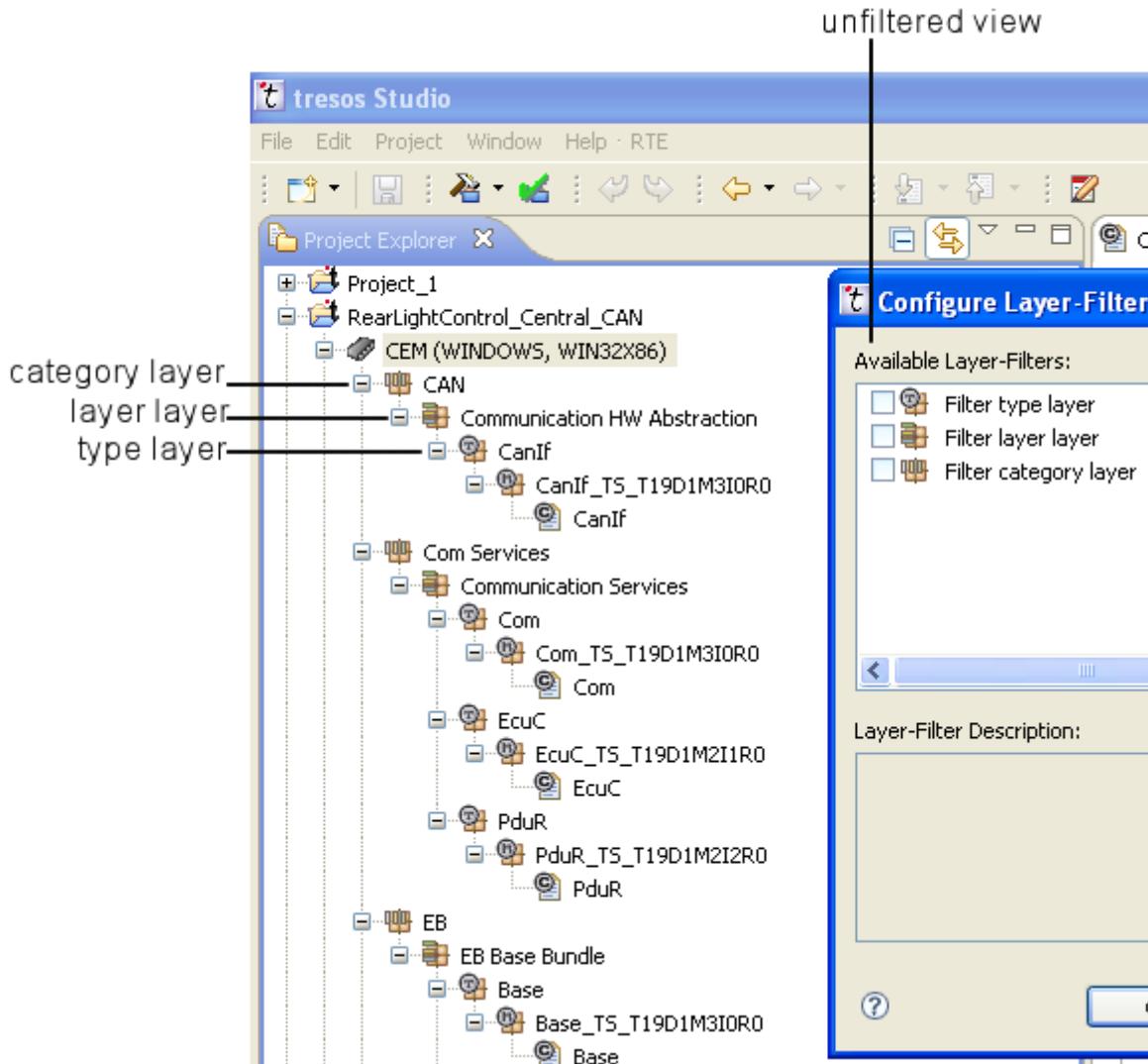
To disable the *type* layer

- ▶ Select the box *Filter type layer*

Example: Unfiltered view

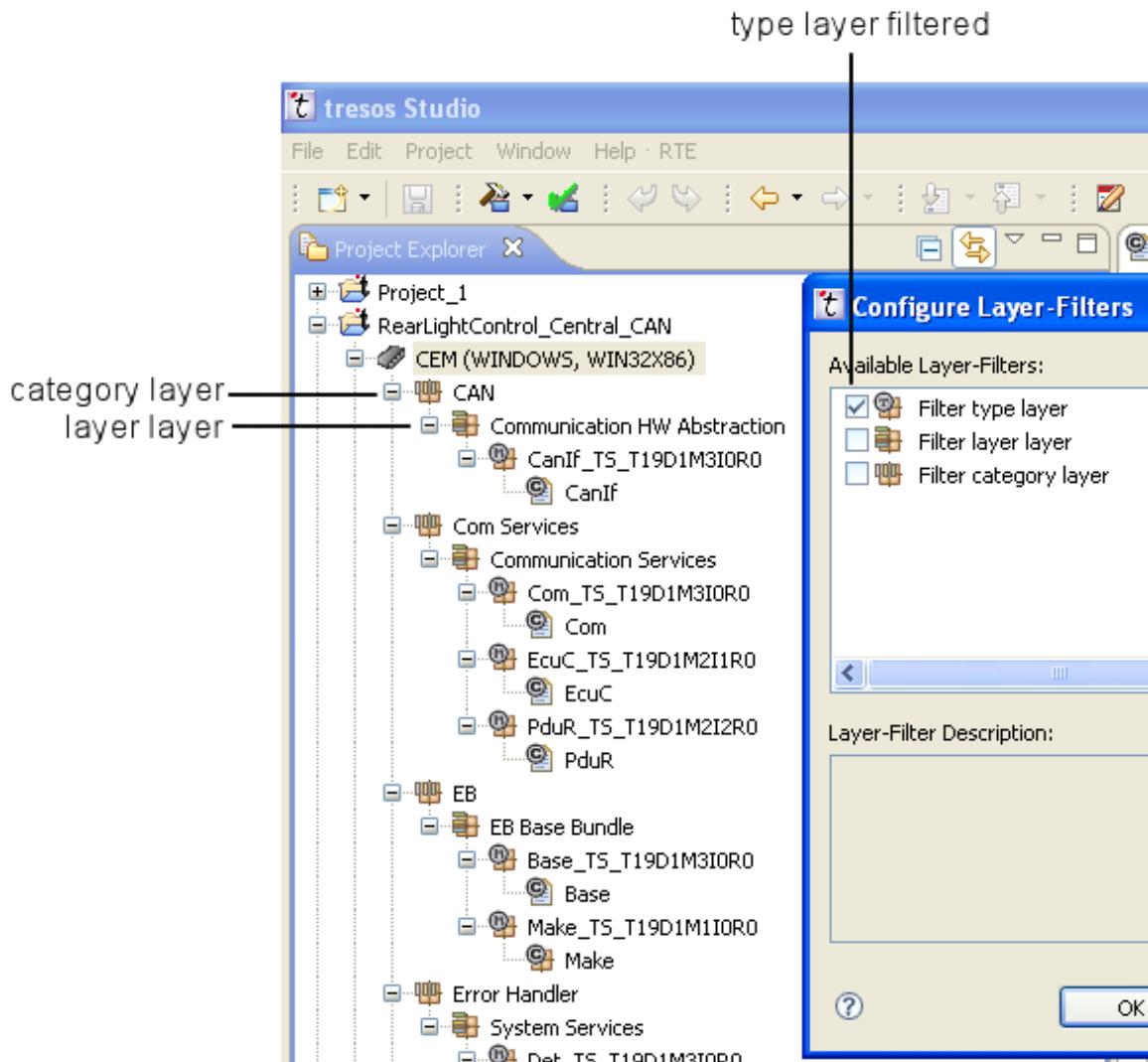
- ▶ The unfiltered view shows

- ▶ the category layer
 - ▶ the layer layer
 - ▶ the type layer
- as well as the module layer and the module configuration layer



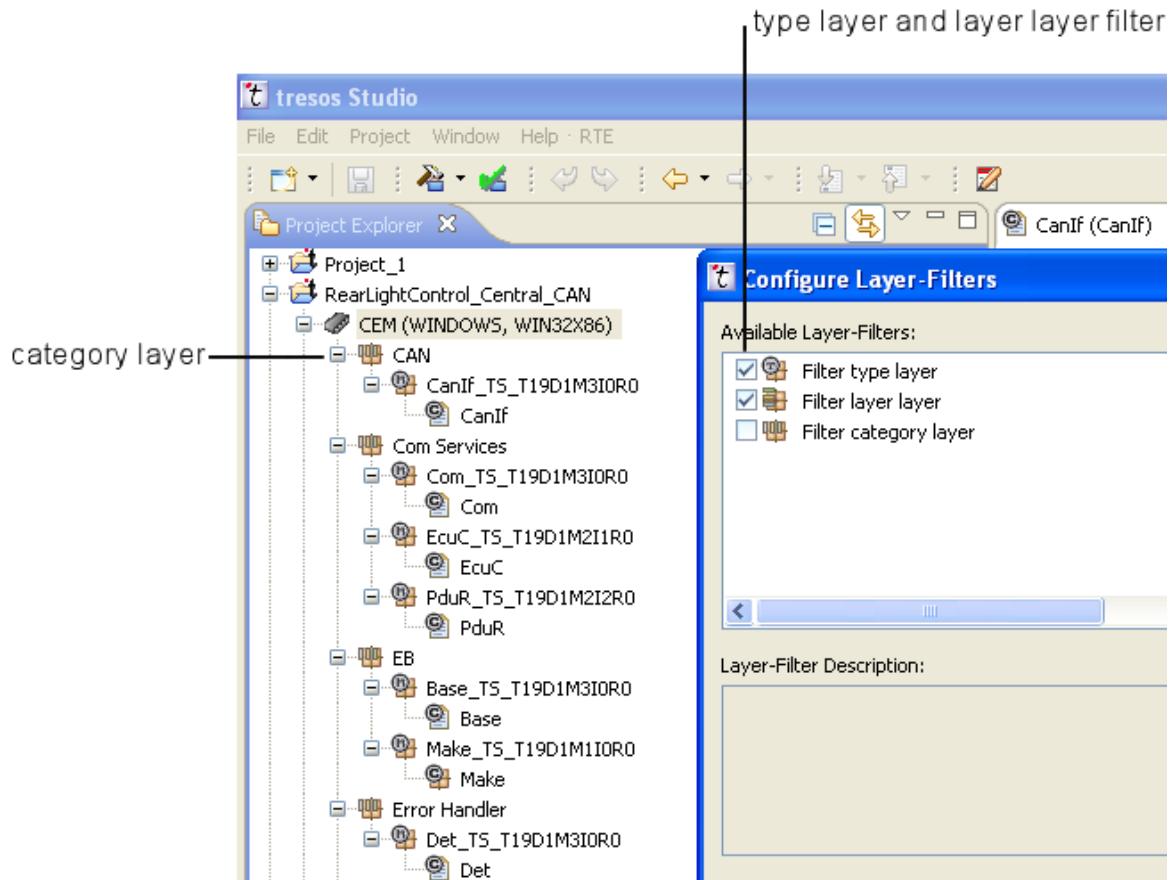
Example: Type layer filtered out

- ▶ If the type layer is filtered out, the *Project Explorer* shows
 - ▶ the category layer
 - ▶ the layer layer
 - ▶ as well as the module layer and the module configuration layer



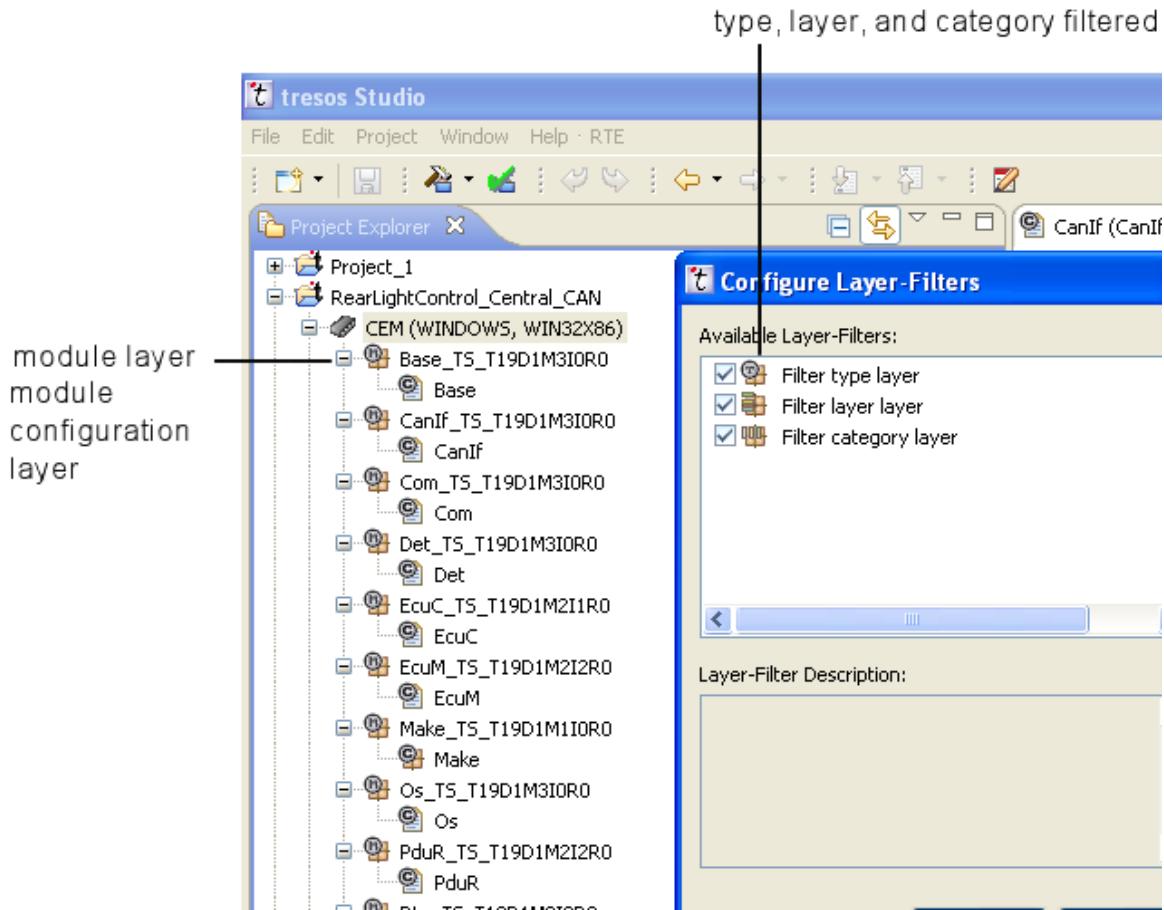
Example: Type and layer layer filtered out

- ▶ If the type and layer layer is filtered out, the *Project Explorer* shows
 - ▶ the category layer
 - ▶ as well as the module layer and the module configuration layer



Example: Type, layer, and category layer filtered out

- ▶ If the type, layer, and category layer are filtered out, the *Project Explorer* shows the only module layer and the module configuration layer.

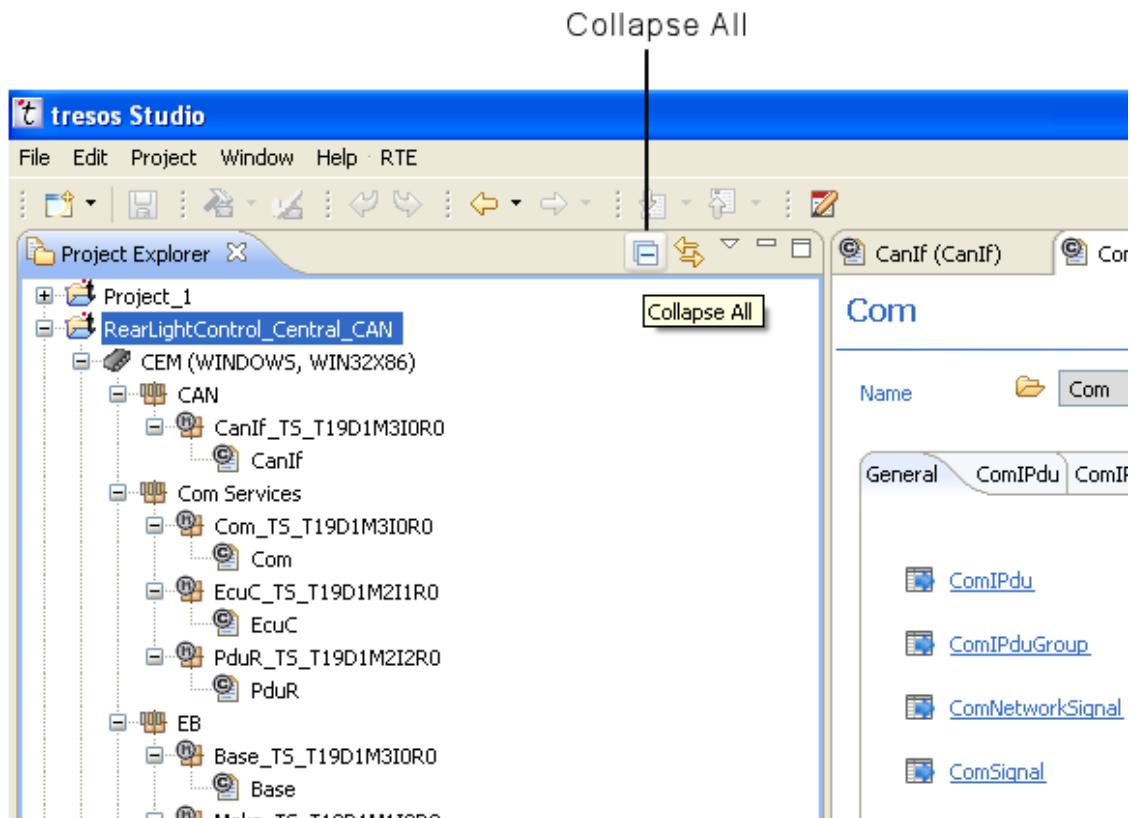


6.3.1.4. Collapsing the Project Explorer tree

To collapse the whole project tree inside the *Project Explorer* so that only the project names are visible:

- ▶ Select the *Collapse All* menu option from the *Project Explorer* tool bar.

Only project names are visible now.

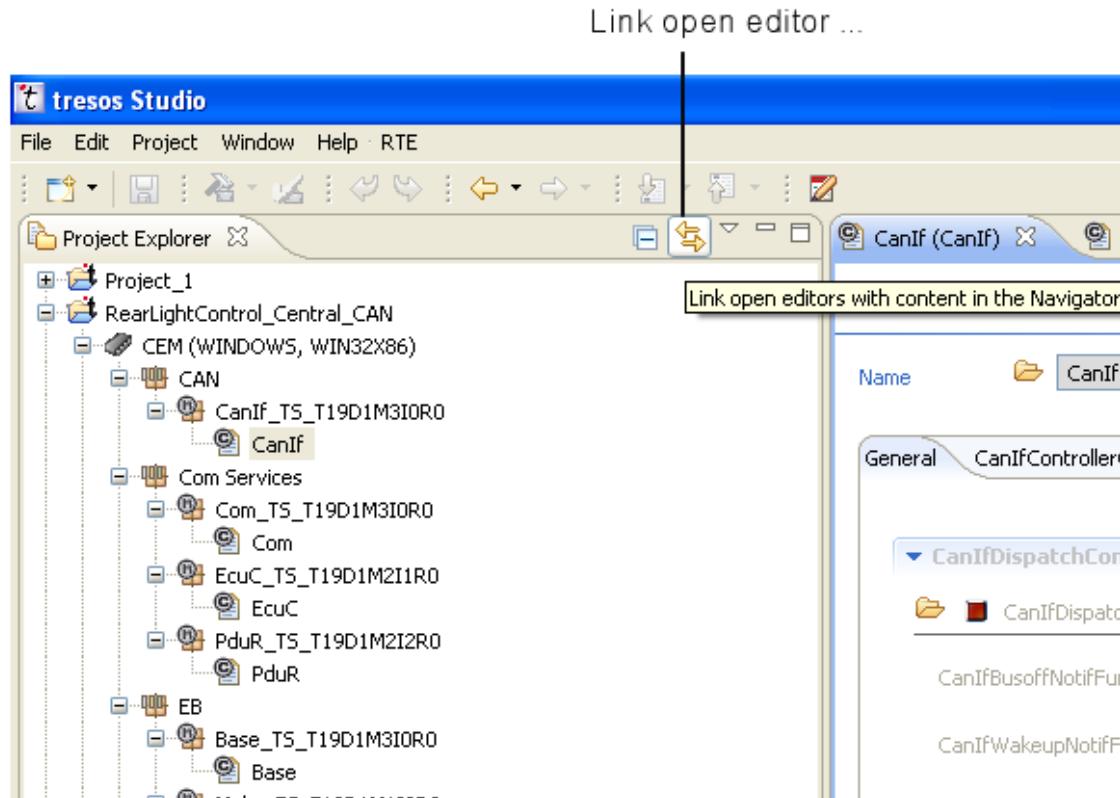


6.3.1.5. Synchronizing the Project Explorer with an editor

To synchronize the *Project Explorer* view with the active editor on the right:

- ▶ Select the **Link open editors with content in the Navigator** toggle switch.

The *Project Explorer* expands a module configuration and scrolls it into view for which ever corresponding editor is selected.



6.3.1.6. Indicating different project states with colors

The *Project Explorer* uses colors to indicate the different states that the module configuration can be in. To change colors of any or all meanings, please refer to [Section 7.2.3.8, “Setting colors and fonts”](#).

Color	Meaning
Black	default color for nodes
Blue	<ul style="list-style-type: none"> ▶ default color for modified and not yet saved module configurations ▶ used in layers in helping to find the modified module configuration if the <i>Project Explorer</i> tree is collapsed ▶ used in editors to highlight modified parameters
Gray	<ul style="list-style-type: none"> ▶ default color for disabled module configurations ▶ if one or more module configurations are disabled, the whole subtree of the disabled module configurations is shown as disabled in gray

6.3.2. The Editor view

The *Editor* view is the main utility to edit a module configuration and the parameters involved in this module. The *Editor* view opens the Generic Configuration Editor, which corresponds to the module you want to edit. Thus, the *Editor* view may look a little different, i. e. have different tabs, for each module. The following image gives you an overview of the main elements of the *Editor* view. A more detailed description of the *Editor* view and instructions for working with the *Editor* view are available at [Section 7.8, “Editing parameters of a module configuration”](#).

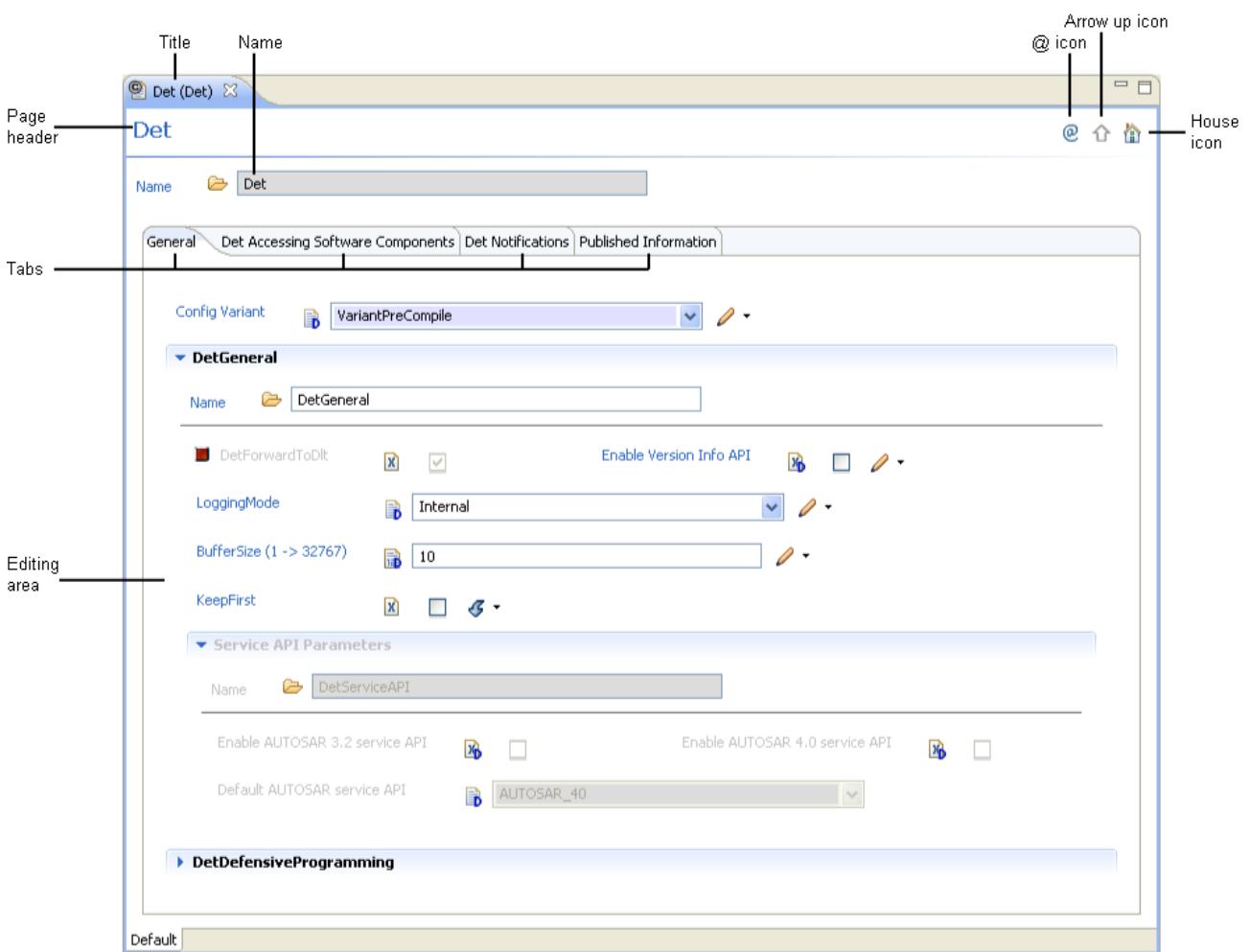


Figure 6.11. The GUI elements of the Editor view: page header, name, title and tabs

Name	Explanation
Editing area	Shows the current configuration parameters. This area displays one page/tab at a time. Pages are organized hierarchically in the form of a tree: The editor is the home page and all other pages are arranged below this home page.

Name	Explanation
	<p>NOTE The AUTOSAR equivalent of a page  The equivalent of a page in an AUTOSAR module is a CONTAINER or a MODULE-CONFIGURATION.</p>
Tabs	<p>Display the parameter in the editor. Only one tab at a time is displayed.</p> <p>NOTE How tabs are organized  ▶ The General tab contains all containers, parameters, and references that appear exactly once. In AUTOSAR modules those elements are:</p> <ul style="list-style-type: none"> ▶ containers: CONTAINER ▶ parameters: *-VALUE ▶ references: REFERENCE-VALUE ▶ elements that appear exactly once: LOWER-MULTIPLICITY==UPPER-MULTIPLICITY==1 <p>▶ All elements that appear more often are displayed as a tab of their own.</p> <p>▶ The AUTOSAR INSTANCE-REFERENCE-VALUE is also displayed in its own tab, because an instance reference contains a table with context references.</p>
Page header	Shows the title of the page on the left.
Title	<p>Shows the title text with the module name, the module configuration name in brackets, and a close button.</p> <p>NOTE The AUTOSAR module name is the title  The equivalent of a title in an AUTOSAR module is the name of the CONTAINER-DEF or MODULE-DEF.</p> <p>TIP Point the mouse over the title for tool tips  To display a tool tip with the module configuration name, the module-ID, and the project the editor belongs to, place your mouse pointer over the title.</p>
Name	Displays the name of the element in the configuration that is represented by the page. If the element is an AUTOSAR module, the SHORT-NAME is displayed.

Name	Explanation
	<ul style="list-style-type: none"> ▶ If the page represents a MODULE-CONFIGURATION element, you cannot change the name. ▶ If the page represents a CONTAINER, you may change the name.
@ @ icon	Select to enter an XPath path to a configuration parameter. The editor navigates to this parameter.
⬆ Arrow up icon	Select to navigate to one page up in the page hierarchy. The editor navigates to the page that hosts the current page.
🏠 House icon	Select to navigate to the home page of this module.

A more detailed description of the *Editor* view and instructions for working with the *Editor* view are available at [Section 7.8, “Editing parameters of a module configuration”](#).

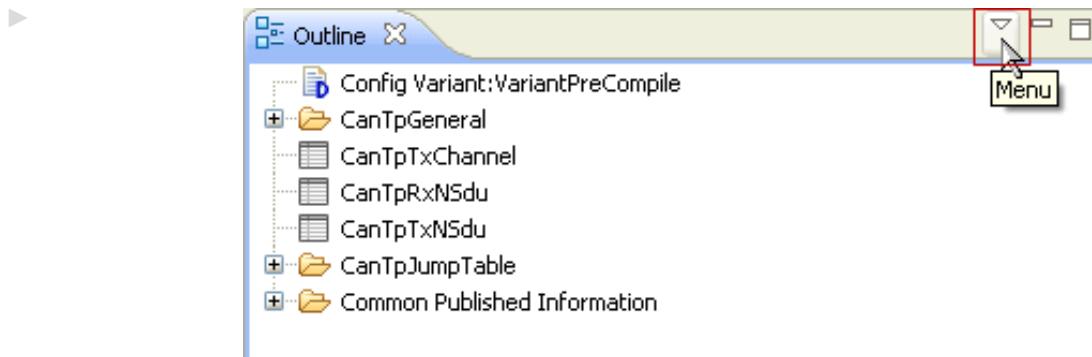
6.3.3. Node Outline view

The *Node Outline* is a tree view outline of the editor contents. The *Node Outline* shows the editor contents in a hierarchical structure. Its main purpose is to navigate between parameters. The *Node Outline* displays the same icons as the *Editor* view for parameters.



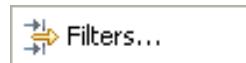
6.3.3.1. Filtering the displayed items

To filter the displayed items in the *Node Outline*:

Figure 6.12. Opening the *Node Outline* menu

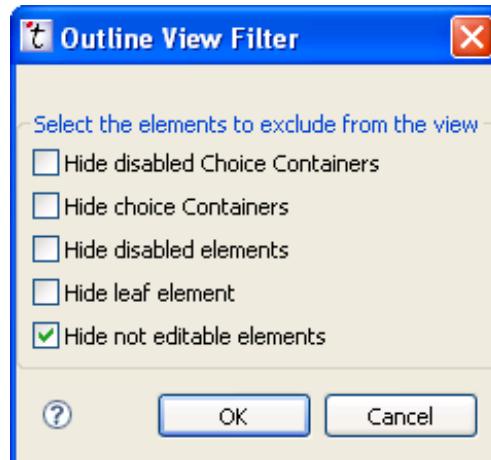
Open the **Menu** symbolized by a small arrow down in the *Node Outline* view tool bar.

A context menu opens up.

Figure 6.13. Opening the *Node Outline Filter* menu

► Select *Filters*....

The *Outline View Filter* window opens up.

Figure 6.14. The *Outline View Filter* window

► Select the check boxes according to the results you wish to obtain:

- To display just the active elements of a choice, select **Hide disabled Choice Containers**.
- To hide containers inside a choice element, select **Hide choice Containers**.
- To hide grayed out elements, select **Hide disabled elements**.



- ▶ To hide parameters and references that show just the structure of the pages in the Generic Configuration Editor, select **Hide leaf element**.
- ▶ To hide elements that cannot be changed in the Generic Configuration Editor such as AUTOSAR CommonPublishedInformation, select **Hide not editable elements**.
- ▶ Click **OK**.

Your filter settings are saved and the items displayed in the *Node Outline* view are displayed according to these settings.

6.3.3.2. Appearance of the displayed items

To control whether the node names or the node lable shall be shown for the items in the *Node Outline*:

- ▶ Select the *Menu* symbolized by a small arrow down.
 - A menu opens up.
- ▶ Select *Show*.
 - A sub-menu with two options opens up.
- ▶ Select the option *Node Names* to display the names of the items shown in the *Node Outline* view. The names of the items in the view are identical to the names of the nodes in the Data Model underlying the editor representation.
- ▶ Select the option *Node Labels* to display the lables of the items. If an item does not have a label, the name of the item is shown instead. Labels usually provide a more meaningful description of the items.

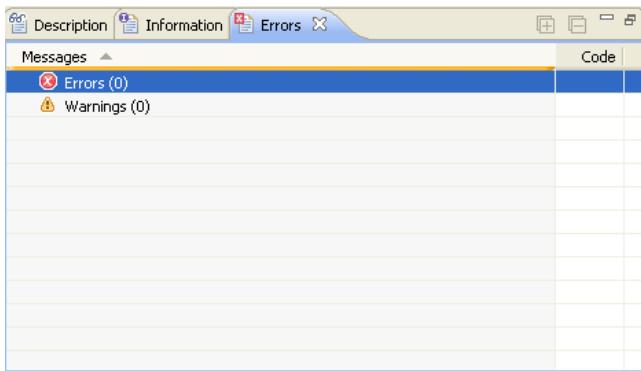
The state the *Node Outline* view is currently in can be checked by opening the menu: a check mark is drawn before the item currently active. The setting is persistent and will be remembered if you close the tool.

6.3.4. Element Outline views

The *Element Outline* consists of three views that display information about the parameter currently selected in the editor. The three views are:

- ▶ the *Element Errors* view
- ▶ the *Element Information* view
- ▶ the *Element Description* view

6.3.4.1. Element Errors view

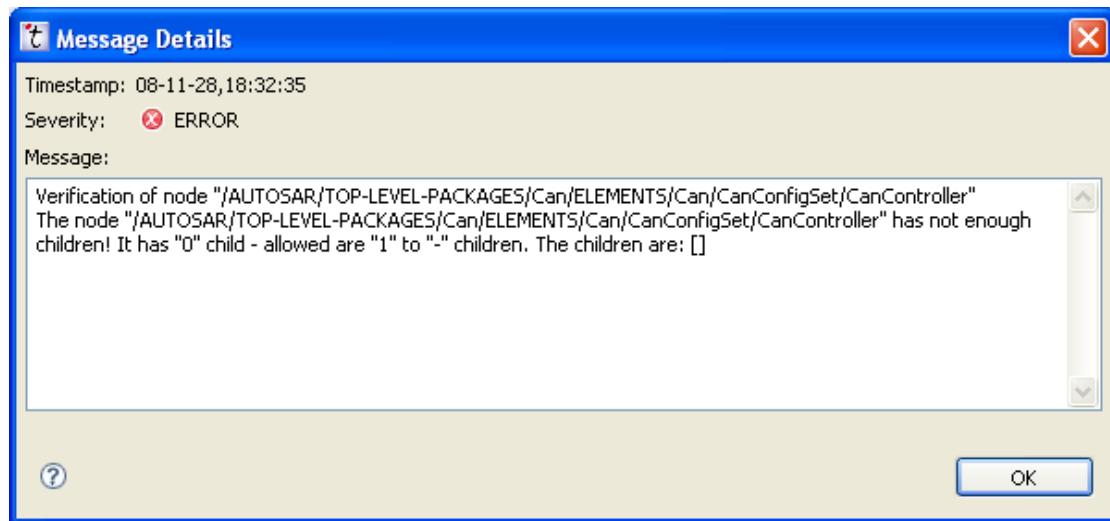


The *Element Errors* view displays messages and errors just as the *Problems* view (see [Section 6.3.6, “Problems view”](#)) but only for the parameter currently selected in the editor. The error code provided is a tool-internal code used by `EB tresos Studio` developers to locate an error in the source code of the tool.

To see detailed message information:

- ▶ Double-click on the message in the *Element Errors* view.

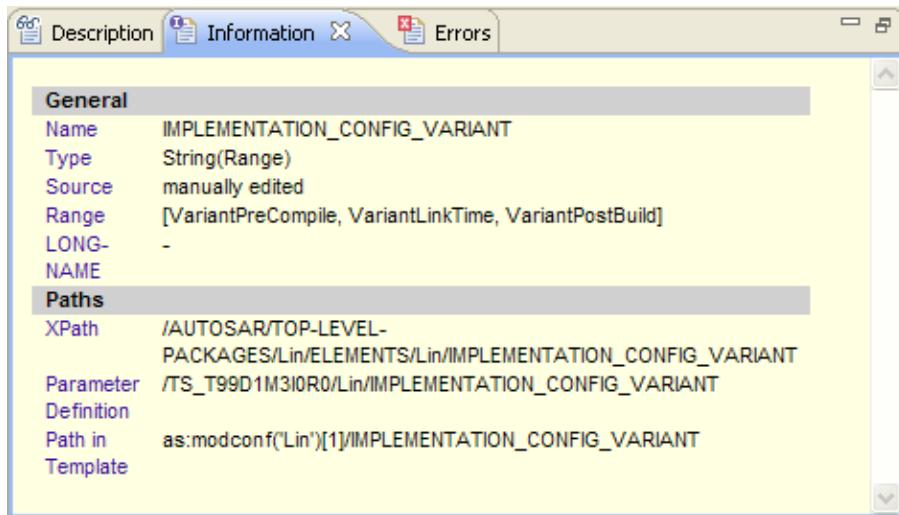
The *Message Details* window opens up.



The window displays the following information:

- ▶ *Timestamp*: the time when the error was reported
- ▶ *Severity*: the message is an error or a warning
- ▶ *Message*: the message information

6.3.4.2. Element Information view



The *Element Information* view shows parameter meta information of the parameter currently selected in the editor. This meta information is derived from the AUTOSAR parameter definition and may contain the following information:

- ▶ *Name*: the name of the element
- ▶ *Type*: the element type (integer, floating point unit, Boolean, string, etc.)
- ▶ *Source*: what the source of the parameter is (e.g. *manually edited* or *default value*).
- ▶ *SHORT-NAME*: the AUTOSAR SHORT-NAME of the parameter
- ▶ *LONG-NAME*: the AUTOSAR LONG-NAME of the parameter
- ▶ *Min*: the minimum number of entries in a table
- ▶ *Max*: the maximum number of entries in a table
- ▶ *Range*: the valid values for this parameter
- ▶ *ORIGIN*: the AUTOSAR ORIGIN of the parameter
- ▶ *UUID*: the AUTOSAR UUID of the parameter
- ▶ *XPath*: the XPath path expression via which it is possible to navigate to the parameter
- ▶ *ASPPath*: The AUTOSAR path expression of the parameter. For reference parameters, the path describing all referenceable nodes is shown here. This path is also shown, if there is no referenceable node available (e. g. not yet configured). It may contain a prefix such as `ASPathDataOfSchema:`, which means, that all configured nodes for a specific schema-node are referenceable. In this case, the AUTOSAR SHORT-NAME-path of the schema-node is the last part of this path.
- ▶ *Parameter Definition*: displays the path to the AUTOSAR parameter definition element
- ▶ *Path to referenceable nodes*: the path to the schema node whose data nodes may be referenced by this reference

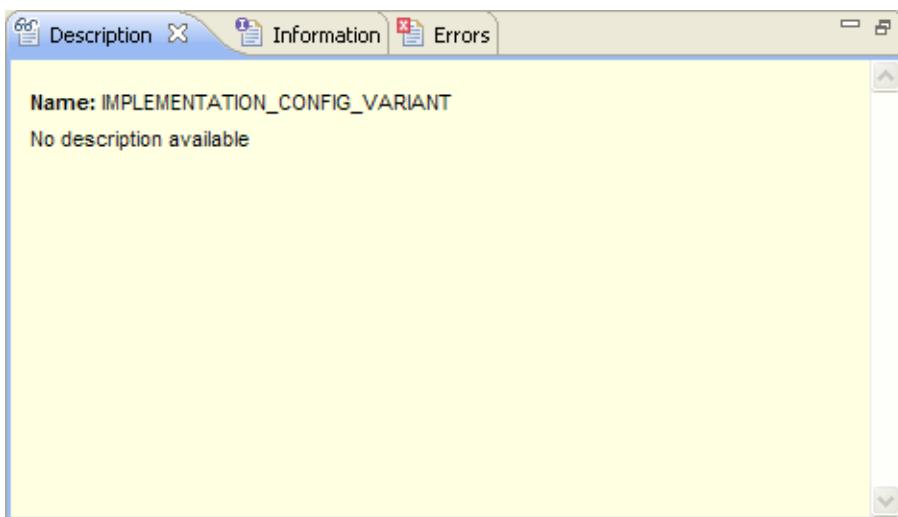
- ▶ *Multiple Configuration*: displays *true* if the selected element is an AUTOSAR multiple configuration container
- ▶ *Config Class*: the AUTOSAR CONFIG-CLASS of the parameter
- ▶ *Config Variant*: the AUTOSAR CONFIG-VARIANT of the parameter
- ▶ *Category*: the AUTOSAR CATEGORY of the parameter
- ▶ *Post Build*: if the AUTOSAR attribute POSTBUILDCHANGEABLE is set to true, *true* is displayed here
- ▶ *Affection Kind*: the AUTOSAR AFFECTION-KIND of the parameter
- ▶ *Affected*: the AUTOSAR AFFECTED of the parameter
- ▶ *Path in Template*: When referencing a parameter within a code template of the template-based code generator, the displayed path may be used. The path is only displayed, if the preference *Enable developer features* is enabled.

NOTE

Not all information is available for every configuration parameter. E. g. the *ASPath* is omitted if the parameter is not referenceable by an AUTOSAR path.



6.3.4.3. Element Description view



The *Element Description* view describes the parameter selected in the editor and gives configuration hints, references other parameters, etc. The description may contain hyperlinks to related parameters (not shown in the screenshot).

To navigate to the related parameter:

- ▶ Select the linked parameter from the *Element Description* view.

This opens up information about the parameter in the *Element Description* view.

6.3.5. Error Log view

The *Error Log* view is where all logged errors are visible in the GUI. The *Error Log* receives error messages from EB tresos Studio components and from other framework components. By default the list of messages in the *Error Log* is preserved in between the sessions of EB tresos Studio, thus the *Error Log* may start up filled with messages.

The line above the header of the shown table contains information about entries which are filtered out.

TIP

Section [Section 7.2.3.6, “Setting Error Log preferences”](#) describes ways to filter out messages and to customize the behavior of the *Error Log*.



The default location of the *Error Log* view is on bottom of the screen in the middle.

1	2	3	4	5
Message	Err...	Entry Type	Origin	Timestamp
☒ The bundle org.eclipse.ant.core could not be properly started. P	18003	Tresos Status	Tresos	08-09-26,13:58:22
☒ A handler conflict occurred. This may disable some commands.	0	IStatus	org.eclipse.ui.workbench	08-09-26,13:58:21
☒ Exception occured: org.eclipse.core.runtime.CoreException.		Exception	Java	08-09-26,13:58:20
☒ Exception occured: org.eclipse.core.runtime.CoreException.		Exception	Java	08-09-26,13:58:19
☒ Exception occured: org.eclipse.core.runtime.CoreException.		Exception	Java	08-09-26,13:58:19
☒ Exception occured: org.eclipse.core.runtime.CoreException.		Exception	Java	08-09-26,13:58:19
☒ Exception occured: org.eclipse.core.runtime.CoreException.		Exception	Java	08-09-26,13:58:19
☒ Exception occured: java.lang.ClassCastException.		Exception	Java	08-09-26,13:58:19
☒ Exception occured: java.lang.IllegalStateException.		Exception	Java	08-09-26,13:58:19
⚠ Project RearLightControl_Central_CAN: A new version 3.0.0 of t	11078	Tresos Status	Tresos	08-09-23,11:32:52
⚠ Project RearLightControl_Central_CAN: A new version 3.0.1 of t	11078	Tresos Status	Tresos	08-09-23,11:32:52
⚠ Added "/AUTOSAR/TOP-LEVEL-PACKAGES/NvM/ELEMENTS/NvM/	1648	Tresos Status	Tresos	08-09-23,11:32:51
⚠ Added "/AUTOSAR/TOP-LEVEL-PACKAGES/NvM/ELEMENTS/NvM/	1648	Tresos Status	Tresos	08-09-23,11:32:51
⚠ Cannot set the value of the node /AUTOSAR/TOP-LEVEL-PACKA	1134	Tresos Status	Tresos	08-09-23,11:32:51

The *Error Log* displays a list of log entries, which can have the following properties:

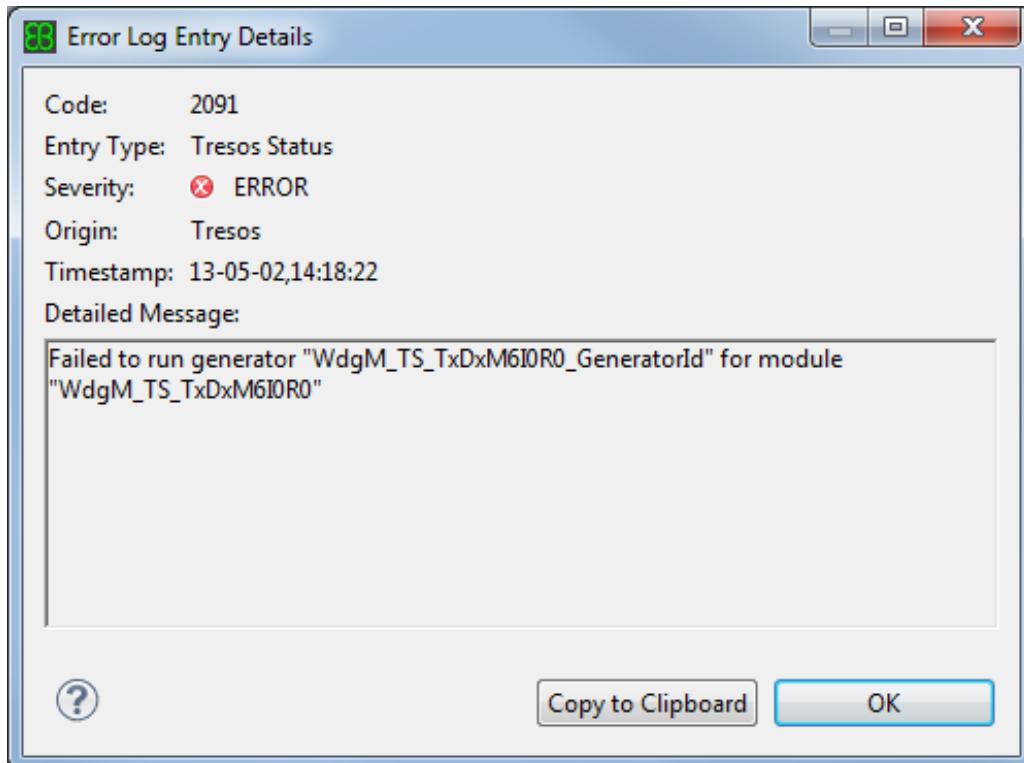
Number	Information type	Explanation
1	Message	<p>Indicates the actual error message. The type of message and its severity is indicated by the icon in front of the message.</p> <p>▶ ☒ indicates an error message (something did not go as expected and it had to be aborted)</p>

Number	Information type	Explanation
		<ul style="list-style-type: none"> ▶ indicates a warning message (something did not go as expected, but it could continue) ▶ indicates an info message (just a notification or feedback for the user)
2	Error Code	Indicates the error number, which is used by EB tresos Studio developers to identify the error in the software code.
3	Entry Type	Indicates the error type, e.g. an exception, the EB tresos status, etc.
4	Origin	Indicates the component that logged the message.
5	Timestamp	Indicates the day and time the message was produced.

To find out more detailed information about an error:

- ▶ Double-click on the entry you want to learn more about in the Error Log.

The *Error Log Entry Details* window opens up.



Additionally to the *Error Log* view information, the window displays a detailed message text. Links may be available and will navigate you directly to the point of interest.

6.3.5.1. Changing the display settings of the Error Log view

To display the message hierarchy of all log entries into messages and submessages:

- ▶ Select the  icon.

The view of all messages is expanded and all messages with submessages are displayed.

To show just the top-level messages instead of all submessages:

- ▶ Select the  icon.

The message view collapses and displays just top-level messages.

To write a message to a text file:

- ▶ Select the  icon.

A dialog window opens up, which enables you to save the message to your computer. This may be useful, e.g. if you want to contact EB tresos Studio support with a problem connected with the message.

To clear the Error Log view:

- ▶ Select the  icon.

All messages disappear from the Error Log view. The messages are not lost but can be restored with the restore button.

To restore messages to the Error Log view:

- ▶ Select the  icon.

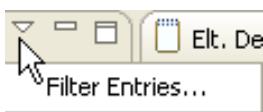
All messages you previously cleared are restored to the Error Log view.

6.3.5.2. Filtering Error Log messages by their severity level

You may filter out messages depending on their severity level.

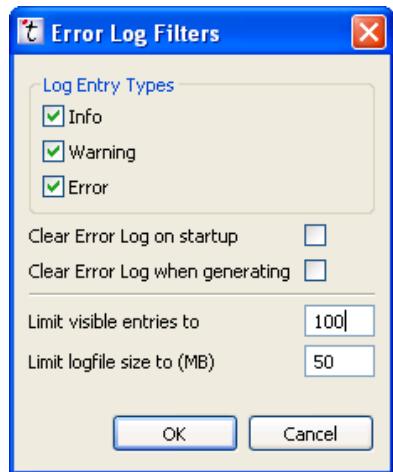
- ▶ Select the  icon.

A context menu opens up.



- ▶ Select *Filter Entries*.

The *Error Log Filters* window opens up.



- ▶ Select the **Log Entry Types** you want the Error Log view to display.
- ▶ Select **OK**.

The Error Log view displays only messages that you selected in the Error Log Filters window.

6.3.5.3. Restricting the amount of shown Error Log messages and the backup logfile size

You may restrict the amount of shown Error Log messages and the size of the backup logfiles.

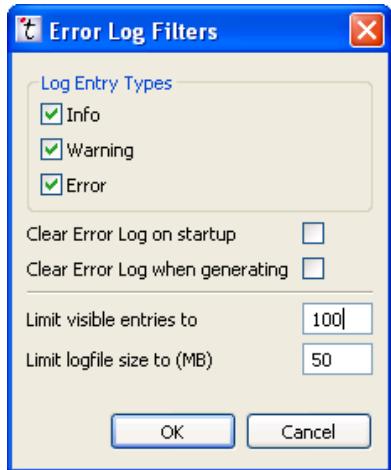
- ▶ Select the ▾ icon.

A context menu opens up.



- ▶ Select *Filter Entries*.

The *Error Log Filters* window opens up.



- ▶ Enter the amount of Error Log Entries you want to see at most.
- ▶ Enter the maximum size for backup logfiles.
- ▶ Select **OK**.

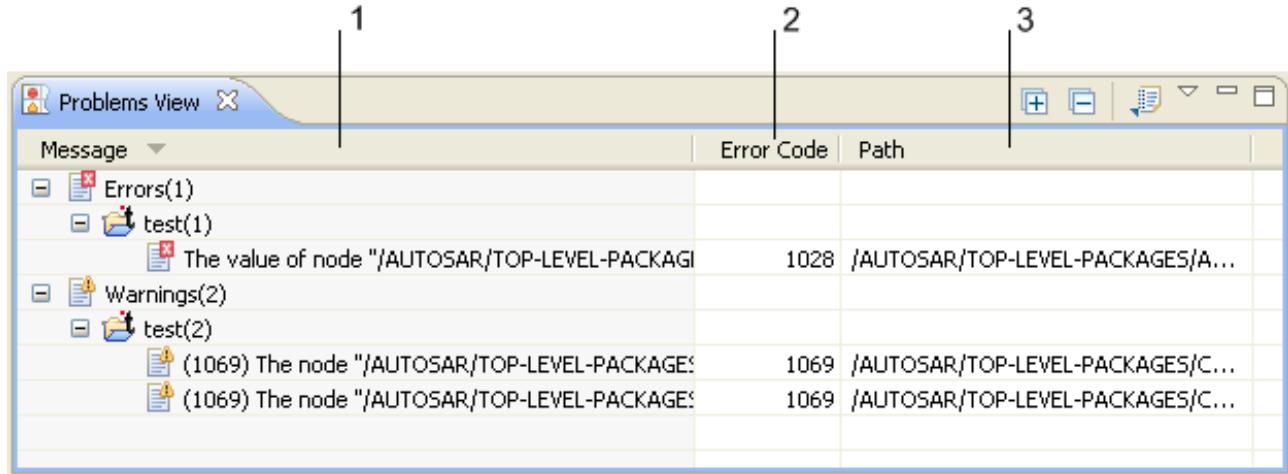
The Error Log view displays at most the number of messages you configured. All backup logfiles created in the future will not be bigger than the configured value.

6.3.5.4. Copying Error Log entries into text documents

You can copy an Error Log entry as plain text to the Windows clipboard and paste it in any text application. This is useful e. g. if you want to contact EB support because of an error message.

To do so, select the Error Log entry you want to copy and use either keyboard shortcut **Ctrl+C** or right-click and select **Copy** in the context menu. A text representation (including more information than you can see in the GUI) of the Error Log entry is now inside the Windows clipboard and can be pasted into other applications.

6.3.6. Problems view



While you edit your configuration, EB tresos Studio constantly checks the configuration for configuration errors and warnings in the background. This background operation is called the *verifier* thread. If a problem in the configuration is found, the verifier adds an entry to the *Problems* view.

The items in the *Problems* view are grouped by the project names and by the type of problem. Possible types are *error*, *warning* and *info*.

To display all levels of errors and warnings available:

- ▶ Open up the top-level tree of the *Problems* view.

The number in brackets behind the group entries indicates the number of messages in the respective category.

Number	Information type	Explanation
1	Message	Indicates the message text. The message can be either a warning or an error.
2	Error Code	Indicates the error number, which is used by EB tresos Studio developers to identify the error in the software code.
3	Path	Indicates the XPath term in the configuration model to the parameter that issued a warning or error.

To edit the parameter that has issued a message:

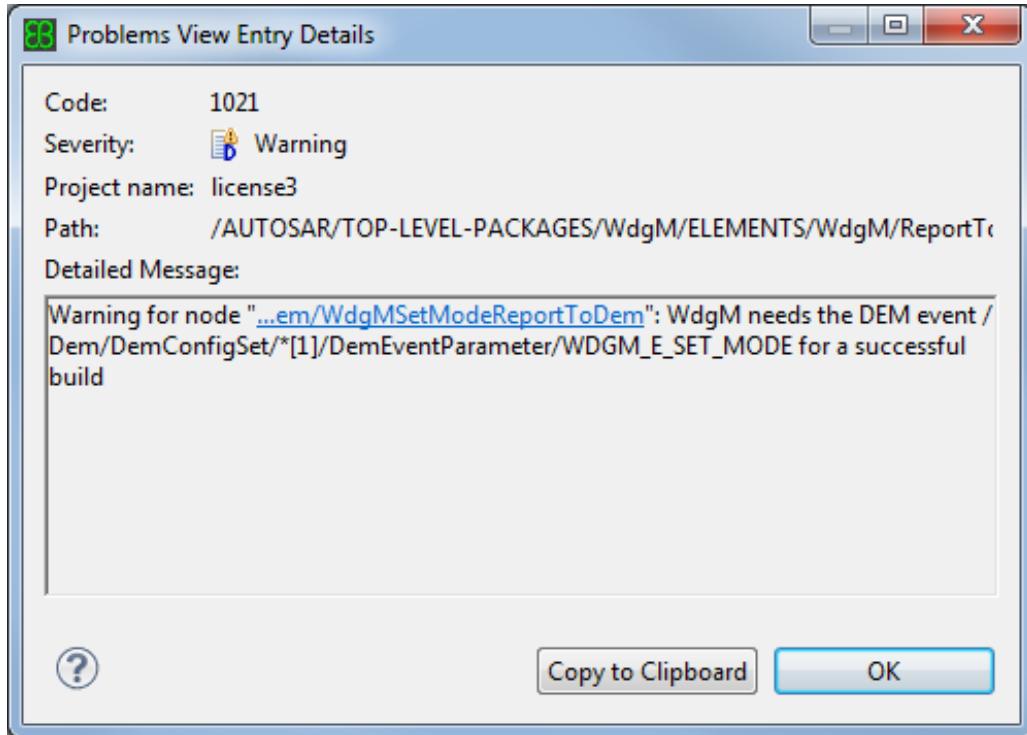
- ▶ Single-click on the entry in the *Problems* view.

The editor of the respective parameter opens up directly at the position where the problem is and you may edit the parameter.

To find out more detailed information about the entry:

- ▶ Double-click on the entry in the *Problems* view.

The *Problems View Entry Details* window opens up.



Additionally to the *Problems* view entry this window shows:

- ▶ the project name in which the message occurred
- ▶ the detailed message text. Links may be available and will navigate you directly to the point of interest.

6.3.6.1. Changing the display settings of the Problems View

To display the message hierarchy of all log entries into messages and submessages:

- ▶ Select the  icon.

The view of all messages is expanded and all messages with submessages are displayed.

To show just the top-level messages instead of all submessages:

- ▶ Select the  icon.

The message view collapses and displays just top-level messages.

6.3.6.2. Export the messages of the Problems View

To write a message to a text file:

- ▶ Select the  icon.

A dialog window opens up, which enables you to save the message to your computer. This may be useful, e.g. if you want to contact EB tresos Studio support with a problem connected with the message.

6.3.6.3. Filtering Problems view messages by their severity level

You may filter out messages depending on their severity level.

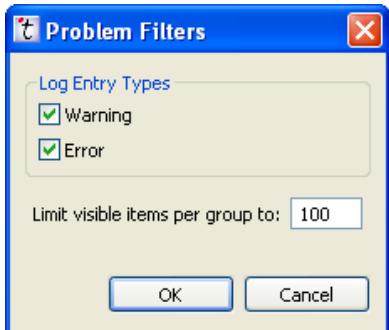
- ▶ Select the  icon.

A context menu opens up.



- ▶ Select *Filter Entries*.

The *Problem Filters* window opens up.



- ▶ Select the *Log Entry Types* you want the *Problems* view to display.
- ▶ Select **OK**.

The *Problems* view displays only the types that you selected in the *Problem Filters* window.

6.3.6.4. Limit the number of entries shown in the Problems View

If you have many entries in the *Problems* view and experience the GUI slowing down, you can restrict the number of entries shown in the *Problems* to an upper bound:

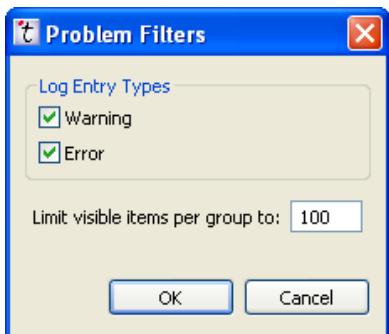
- ▶ Select the icon.

A context menu opens up.



- ▶ Select *Filter Entries*.

The *Problem Filters* window opens up.



- ▶ Enter a positive number in the entry **Limit visible items per group to**.
- ▶ Select **OK**.

The *Problems* view restricts the number of shown entries.

6.3.7. Workflows view

The *Workflows* view displays one workflow at a time. A workflow consists of instructions that help you to accomplish a certain task. The steps to proceed are structured as a tree, through which you may go and perform each single step.

The *Workflows* view consists of three parts:

tool bar

The tool bar provides buttons to perform the actions of the workflow.

step area

The *step area* displays the workflow structured as a tree. The step area may contain action steps and group steps. For further information on action steps and group steps, refer to [Section 6.3.7.2, “Performing steps of a workflow”](#).

description area

The *description area* describes the step currently selected.

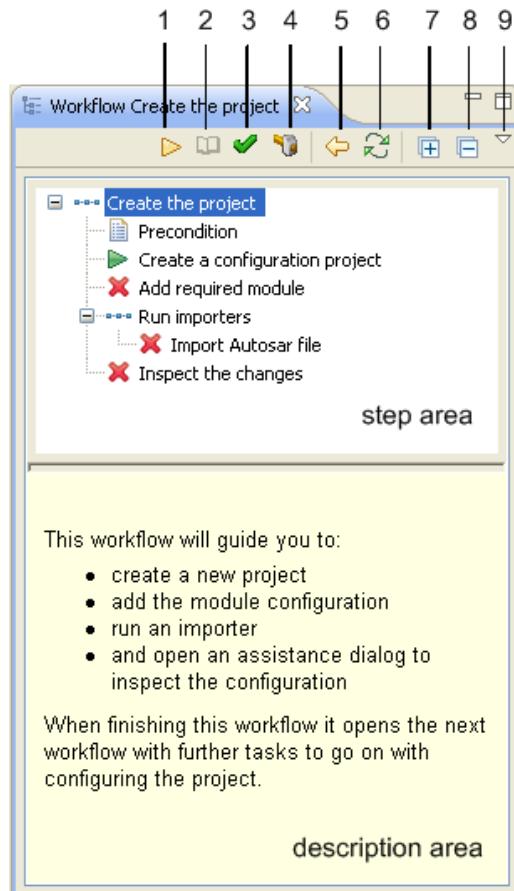


Figure 6.15. Workflows view

In the tool bar of the *Workflows* view, the following buttons are available:

Number	Action	Description
1	Run/Jump	<p>This action shows different icons depending on the current selection in the workflow tree:</p> <p>▶ Jump button to jump to the next step. This action is shown for all group steps and for all unavailable action steps (marked with ✗)</p> <p>▶ Run button to execute the selected operation. This action is shown for all available action steps.</p> <p>If you click the Run button, the default action of the selected step is performed and the next step is selected. If the action is not repeatable, it is marked as finished afterwards. If the step currently selected only has a description or is a group step, this action only selects the next step.</p>

Number	Action	Description
2	Configure	The Configure button is only enabled, if the currently selected step is an action step with a configuration command. Clicking the Configure button executes the configuration command associated with this step.
3	Finish	The Finish button marks the step currently selected as finished. If a group step is selected, all sub-steps are marked as finished, too. If a step is repeatable, the only way to mark it as finished is to click the Finish button.
4	To do	If you cannot perform a step right away, you may mark this step with a To do marker. You can thus see which steps still need to be performed so that you won't forget them.
5	Back	The Back button allows you to go back to the last step performed. Reverting the last step performed is only possible, if you had used the tool bar buttons to perform the steps. If you use the cursor keys to select steps, you cannot use the Back button to revert the selection change.
6	Reset	With the Reset button, you may reset all step states of the complete workflow tree.
7	Expand all	Expands all nodes of the workflow tree.
8	Collapse all	Collapses all nodes of the workflow tree.
9	Workflow selection menu	In the workflow selection menu, you can select the last three workflows used. You may also open the <i>workflow selection</i> dialog to choose a workflow from the list of all registered workflows.

6.3.7.1. Selecting a workflow

In the **workflow selection** menu, you can select one of the last three workflows used or open a dialog in which all registered workflows are displayed.

To select one of the last three workflows used:

- ▶ Click the  button in the tool bar.

The **workflow selection** menu opens up.

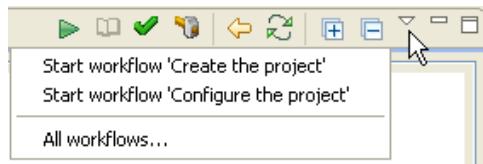


Figure 6.16. Opening the workflow selection menu

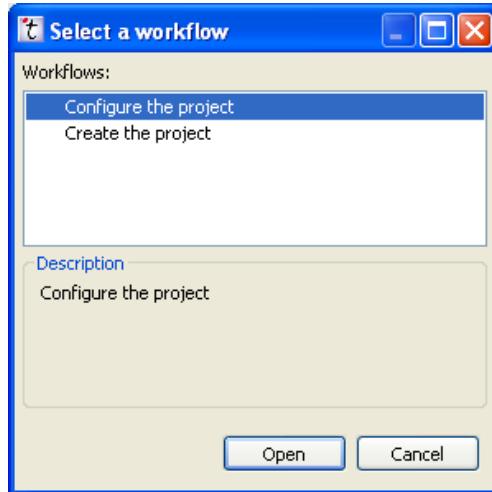
- ▶ Select one of the listed workflows.

The selected workflow is opened in the *Workflows* view.

To open a workflow from the list of all registered workflows:

- ▶ In the **workflow selection** menu, select the item *All workflows....*

The *Select a workflow* dialog opens up.

Figure 6.17. The *Select a workflow* dialog

- ▶ Select the workflow you would like to open in the *Workflows* view. Note that only one workflow can be selected at a time.

The selected workflow is displayed in the *Workflows* view.

6.3.7.2. Performing steps of a workflow

The *step area* of the *Workflows* view displays the steps that are necessary to complete the task. These steps may be either *group* steps or *action* steps. Group steps are steps that contain other steps, action steps are steps that contain commands that can be performed when you process the workflow.

The group steps are displayed as follows:

Icon	Description
	This icon represents a group step of the type <i>choice</i> . This means that only one of the steps must be performed to finish the task.
	This icon represents a group step of the type <i>sequence</i> . This means that all steps must be executed successively to finish the task.

The action steps are displayed as follows:

Icon	Description
	This icon represents an action step whose command can be performed either by clicking the Run button or by double-clicking the step itself.
	This icon represents an action step that provides only a description.

The state of each step is displayed with the following icons and overlay icons:

Icon	Description
	This icon is displayed, if the step is not available for the current selection in the <i>Project Explorer</i> view.
	You can mark each step with a To do marker by using the tool bar button To do . The selected step and all steps on the path upwards to the root node of the workflow tree are then marked as <i>To do</i> . This helps you to remember steps that haven't been completed yet, but must not be forgotten.
	This overlay icon informs you that this step already has been finished. If you go through the workflow tree with the Run button, a finished step will be skipped.
	This overlay icon is displayed, if the step is a repeatable step. Repeatable steps are not automatically marked as finished after running the step. The only way to mark a repeatable step as finished is to click the Finish button in the tool bar.

6.3.8. Sidebar view

The *Sidebar* view displays assistance dialogs available for the elements currently selected, e.g. the CAN Buffer Assignment Editor and the Frif Joblist Editor. The content of the *Sidebar* view is split into categories, e.g. *OS*, *Debugging*. These categories may be organized in a hierarchical manner, which is displayed as a tree in the *Sidebar* view. Which assistance dialogs are available depend on the context, i.e. on the opened editors, the selected items in the project explorer, the module configurations of your project etc.

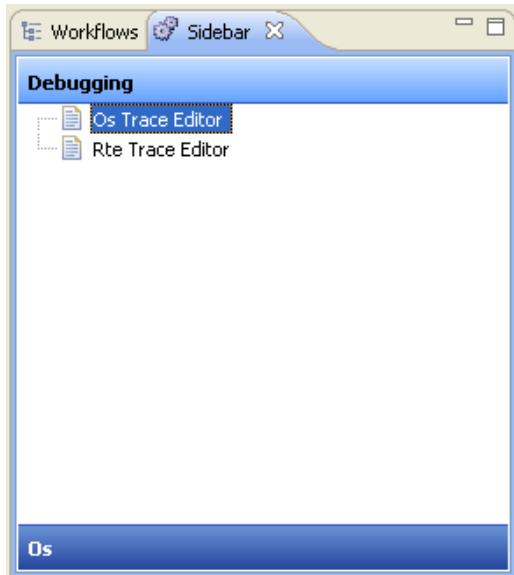


Figure 6.18. The *Sidebar* view

You can open these assistance dialogs by double-clicking them in the *Sidebar* view. You find detailed information on the assistance dialogs in the documentation of the respective modules.

6.4. Editors

6.4.1. The Generic Configuration Editor

The Generic Configuration Editor is the central tool for editing ECU configurations. The content of this editor is rendered out of the schema of the module to which a configuration belongs. The content below the header depends on what you are currently viewing. E.g. if you jump into a subcontainer, the editor will show the content of the subcontainer. You can use the *Node Outline* view to see your current position and to navigate inside the editor.

6.4.1.1. Sections and parameters

The following image shows you how parameters are displayed in the Generic Configuration Editor:

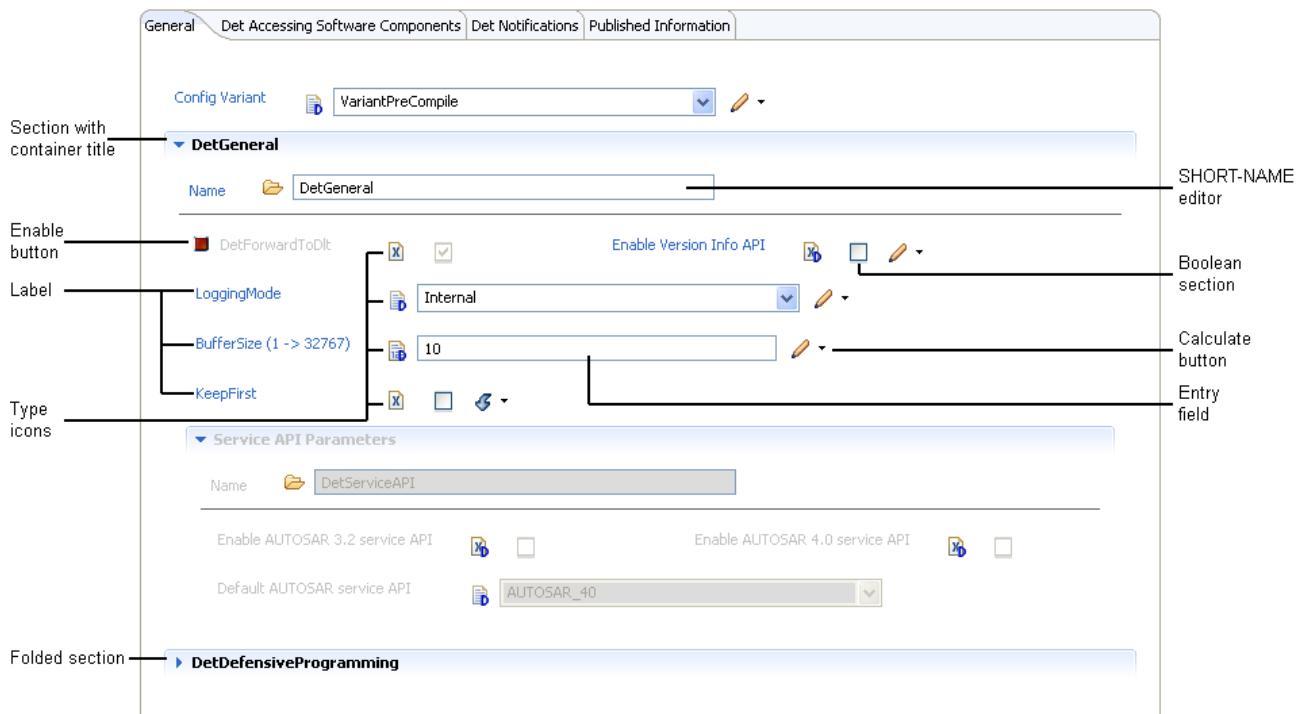


Figure 6.19. The GUI elements of the Generic Configuration Editor: sections, labels, type icons

Name	Explanation
Section with container title	Parameters and references in the General tab are sorted into sections. For AUTOSAR any section represents a CONTAINER that appears only once inside the element.
SHORT-NAME editor	In this entry field you can edit the SHORT-NAME of an AUTOSAR CONTAINER. The default value is the SHORT-NAME of the corresponding container in the AUTOSAR module definition.
Boolean section	In the Boolean section, you may edit the Boolean value of a parameter. <ul style="list-style-type: none"> ▶ To set the Boolean value to <code>true</code>, check the check box. ▶ To set the Boolean value to <code>false</code>, uncheck the check box.
Calculate button	Selecting the Calculate button initiates specified parameter calculations. The appearance of the icon depends on the current state of the parameter: <ul style="list-style-type: none"> ▶ <p>The current value of the element is not edited manually but comes from another source (e.g. is the default value or was imported). Clicking this button marks the element as manually edited. With this operation you accept the value. Operations such as running unattended wizards or importers that</p>

Name	Explanation
	<p>would normally overwrite the default, calculated or imported values do not change the element any more.</p> <ul style="list-style-type: none"> ▶ The small calculator icon indicates that the value of the element depends on the value of other elements and that you can let EB tresos Studio calculate its value. The current value of the element is not edited manually but comes from another source (e.g. is the default value or was imported). Clicking this button marks the element as manually edited. With this operation you accept the value. Operations like running unattended wizards or importers that would overwrite default, calculated or imported values do not change the element any more. ▶ The current value of the element was edited manually. Clicking the button recalculates the element value and marks it as a calculated default value. ▶ The current value of the element was edited manually. Clicking the button sets the value of the element to its default. ▶ The current value of the element was edited manually and there is no default value to set. Clicking the button clears the value of the element. ▶ The value of this integer element may be calculated automatically by the generator. If you click this button, a context menu opens up in which you may choose whether the value shall be calculated automatically or whether you want to calculate this value manually:
Entry field	

Name	Explanation
	<p>The entry field next to the type icon shows a parameter value you may change. The entry field is enabled or disabled, depending on the configuration class of the parameter and on the configuration time of the project. Depending on the parameter type, the entry field is displayed either as text box, as check box or as combo box:</p> <ul style="list-style-type: none"> ▶ Arbitrary text strings are displayed as text boxes or as combo boxes if they have a predefined value list. No context menu is provided. ▶ Boolean values are displayed as check boxes. ▶ Integers or floating point numbers are displayed as text boxes or - if the editor provides a predefined value list - as combo boxes. A context menu enables selecting the display formats: <ul style="list-style-type: none"> ▶ Autodisplay as <code>HEX</code> ▶ Autodisplay as <code>BIN</code> ▶ Autodisplay as <code>DEC</code> ▶ Autodisplay as <code>OCT</code> <p>The display format does not affect the format in which the number is stored in the configuration.</p> <p>To place separators (e.g. periods or spaces) into the displayed number, select <i>Show separators</i> from the context menu.</p> <p>You may enter integers in the following number formats:</p> <ul style="list-style-type: none"> ▶ Hexadecimal digits: If the number starts with a <code>0x</code>, it is interpreted as a hex number. ▶ Binary digits: If the number starts with a <code>b</code>, it is interpreted as binary number. ▶ Octal digits: If the number starts with a <code>\0</code> it is interpreted as an octal number. ▶ other digits: All other numbers are interpreted as decimal numbers.
Folded section	<p>For a better overview you can fold sections to temporarily hide their content.</p> <ul style="list-style-type: none"> ▶ To fold a section, click the container title. The section folds and the blue arrow before the container title becomes horizontal. ▶ To unfold a folded section, click again the container title. The section unfolds and the blue arrow before the container title becomes vertical.

Name	Explanation
Type icons	<p>The type icon indicates the type of parameter. The following list explains the meaning of all type icons:</p> <ul style="list-style-type: none"> ▶  You may enter arbitrary text strings. ▶  You may enter integer or floating point numbers. ▶  You may enter Boolean values. <p>There are some overlay icons that might be displayed over the type icon. The following list explains the meaning of all overlay icons:</p> <ul style="list-style-type: none"> ▶  Indicates a wrong parameter. ▶  Indicates an editor warning. ▶  The element is currently set to its default value. ▶  The element value is calculated. ▶  The parameter is set to a preconfigured value. ▶  The parameter is set to a recommended configured value. ▶  The parameter is calculated automatically by the generator. ▶  The element value was imported. ▶  The parameter has a comment. <p>This parameter comment is displayed as a tool tip and shows a summary, status and the default language of the comment.</p> <ul style="list-style-type: none"> ▶ Depending on the parameter, the icons might host a context menu with further functionality.
Label	The label describes the element. In the majority of cases it indicates the parameter name. In the case of an AUTOSAR module, the label indicates the SHORT-NAME. It might also be named differently, e.g. to indicate the functionality of the element.

Name	Explanation
	<p>NOTE Labels with a star indicate unsaved values</p>  <p>If the parameter contains an unsaved value, the label is displayed with a star and turns into darker blue.</p>
Enable button	This button is for enabling or disabling optional elements. For details see Section 6.4.1.5, “Optional elements” .

6.4.1.2. References

References are elements that refer to parameters. AUTOSAR distinguishes between REFERENCE-VALUE and INSTANCE-REFERENCE-VALUE:

- ▶ REFERENCE-VALUE (simple reference) references a single parameter with a reference path.
- ▶ INSTANCE-REFERENCE-VALUES (instance reference) need a target reference path and various context reference paths to reference a parameter.

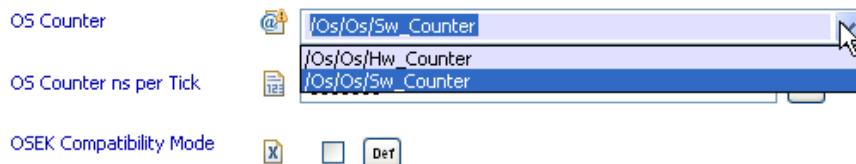


Figure 6.20. The GUI elements of the Editor view: references

In the entry field of a simple reference, you may enter:

- ▶ a path expression, or
- ▶ an AUTOSAR path, or
- ▶ select an AUTOSAR path from the paths determined in the parameter definition.

NOTE**A warning icon indicates an unverified path of the reference**

If the reference element is not part of the loaded configuration and thus EB tresos Studio cannot verify the path, the element is marked with a warning icon.

TIP**Navigating with context menus**

You may navigate to the editor of the referenced parameter via the context menu 'Go to referenced node' of the type icon. This is possible only if the parameter is part of the configuration project.

You can also navigate to the referenceable nodes e.g. to add some more entries to a list. This is possible via the context menu 'List referenceable nodes' of the type icon. If there is exactly one referenceable node, then the editor navigates directly to this node. If there are more referenceable nodes, then a list of nodes is shown in the Search view.

AUTOSAR instance references consist of a target reference and a list of context references.

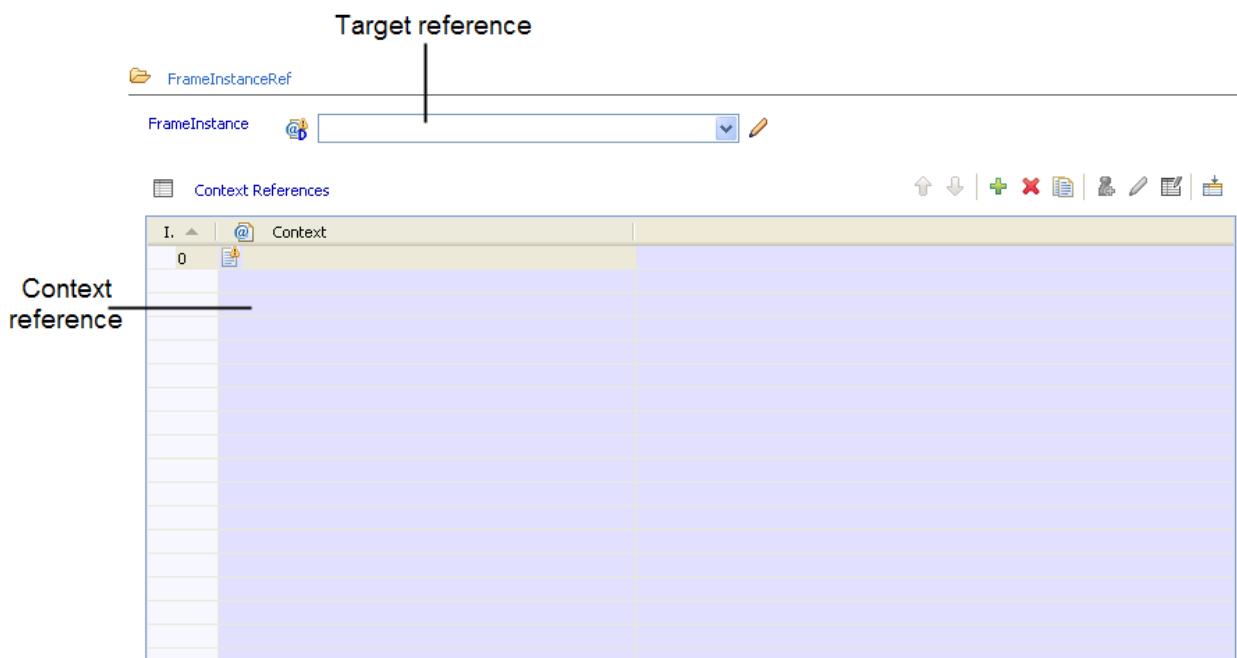


Figure 6.21. The GUI elements of the Editor view: instance references

In the editor, the target reference is represented by a drop-down list box and the context references are displayed as the elements of the table. In the drop-down list box you may select the target of the reference.

If the target cannot be uniquely referenced in your configuration, you need to specify the context of the target reference by adding one or more context references to the table. If valid selections are available in your configuration, you may select them in all drop-down list boxes of the instance reference.

6.4.1.3. Choices

AUTOSAR provides choices to select between different parameter sets. A choice in the AUTOSAR parameter definition model consists of a CHOICE-CONTAINER-DEF that includes PARAM-CONF-CONTAINER-DEFS.

In EB tresos Studio, a choice is displayed as a section of the Editor view. The section contains a drop-down list box, which represents the choice container CHOICE-CONTAINER-DEF, as well as the selectable subcontainers PARAM-CONF-CONTAINER-DEFS in separated tabs. The drop-down list box provides all available choices to select. EB tresos Studio enables the subcontainer of the currently selected choice and displays the title of its tab in bold font. The other subcontainers are disabled.

NOTE
Changing the selected choice


If you change the selected choice in the drop-down list box, the subcontainer previously enabled is now disabled. The newly selected one is then enabled instead and its tab is brought to the front.

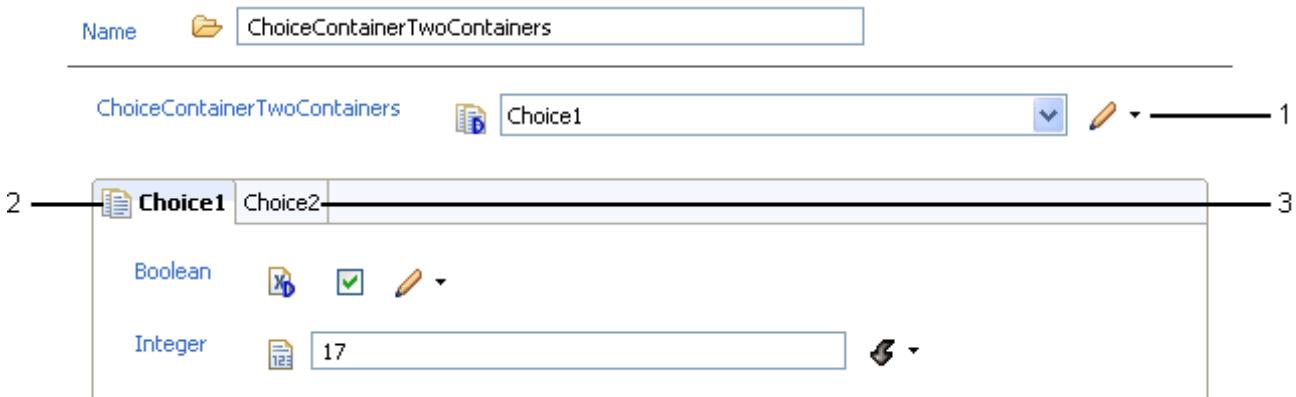


Figure 6.22. The GUI elements of the Editor view: choices

Name	Explanation
1	The drop-down list box, in which you may select the choice you want to activate. You can select only one choice at a time.
2	The subcontainer of the selected choice is enabled and its tab is initially brought to front. The tab's title is displayed in bold font.
3	EB tresos Studio provides the content of the other subcontainers in separated tabs.

TIP**Displaying subcontainers of not selected choices**

To have a look at the subcontainer of a not selected choice without changing the selected choice in the drop-down list box, click on the title of a tab. The content of the disabled subcontainer is then displayed.

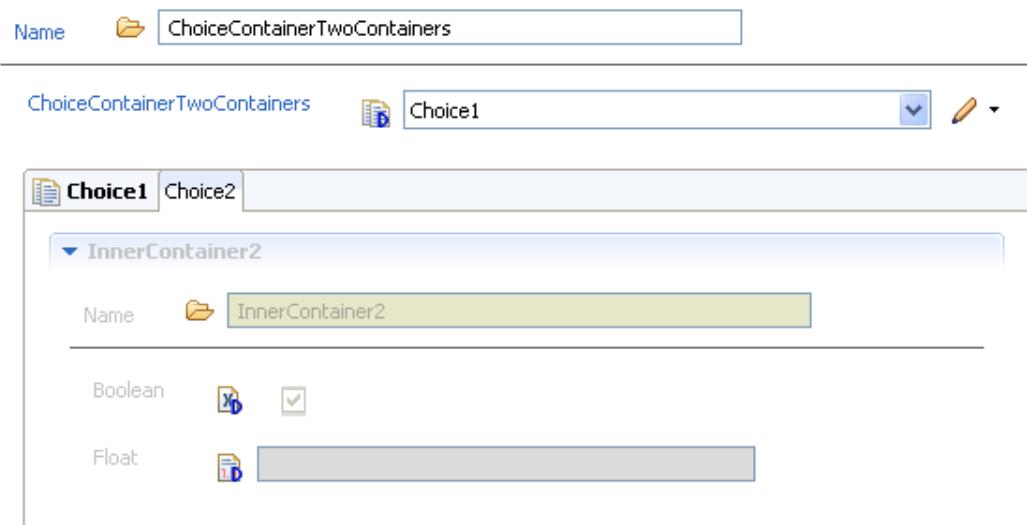


Figure 6.23. Viewing the content of a disabled choice

6.4.1.4. Tables

Tables represent all elements for the configuration of a user-selectable number. In AUTOSAR terms this means:
All elements that have a

- ▶ LOWER-MULTIPLICITY != UPPER-MULTIPLICITY , or
- ▶ LOWER-MULTIPLICITY == UPPER-MULTIPLICITY > 1.

How many instances of an element you need depends on the parameter and on the context you are using it in.

Parameters that must be displayed in a table are provided with their own tab.

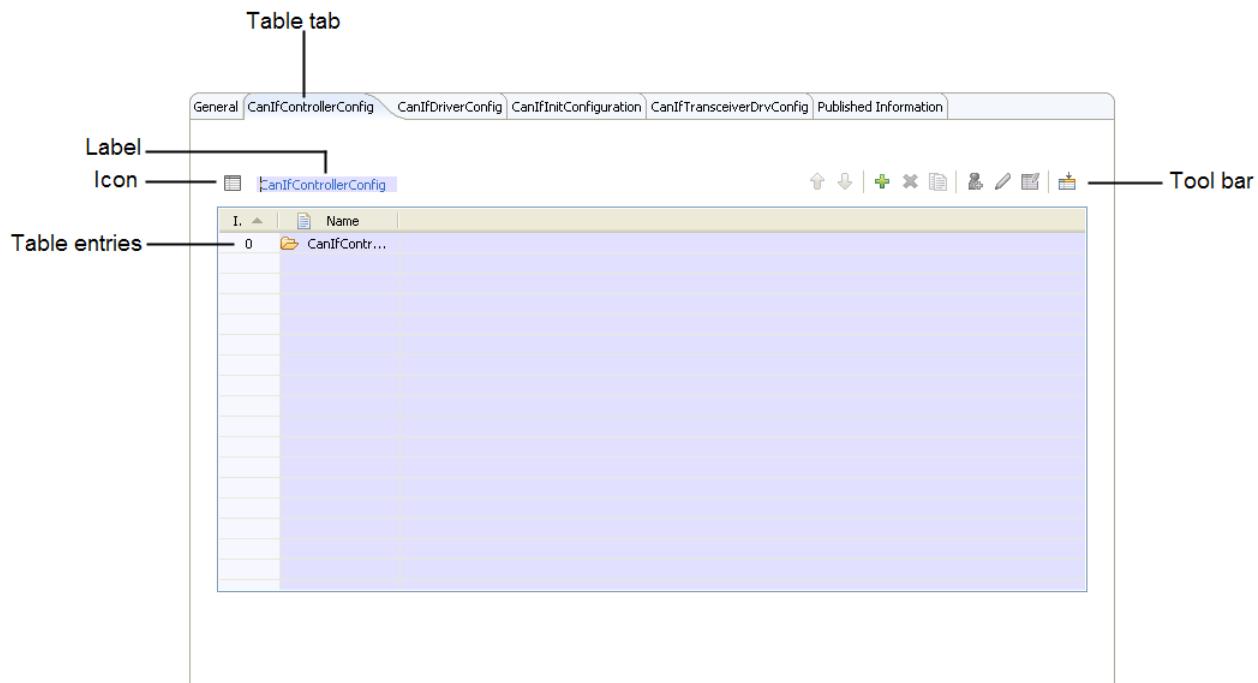


Figure 6.24. The GUI elements of the Editor view: tables

A table consists of:

- ▶ a label with an icon
- ▶ a list with table entries
- ▶ a tool bar

Name	Explanation
Table entries	<p>The layout of the table entries depends on the type of elements displayed. A table may display elements that group other elements, such as</p> <ul style="list-style-type: none"> ▶ AUTOSAR containers are displayed with an index column and a column in which the container name is editable. Additional columns may be available for parameters beyond the container. ▶ To navigate to the editor page in which the container content can be edited, double-click a cell in the index column. The tab of the container content opens up in the <i>Editor</i> view. ▶ To display information about the selected entry, click once on a cell in the index column. The information is displayed in the <i>Element Outline</i> view.

Name	Explanation
	<ul style="list-style-type: none"> ▶ Choices are displayed at least with an index column, a name column, and a choice column. ▶ To edit the content of the choice, double-click a cell in the index column. The tab of the choice content opens up in the <i>Editor</i> view. ▶ To display information about the selected entry, click once on a cell in the index column. The information is displayed in the <i>Element Outline</i> view. ▶ Parameters are displayed in two columns: The first is the index column, the second displays the current parameter value. The icon in the column header of the second column represents the entry type (e.g. reference). ▶ References are displayed in two columns: The first is the index column, the second displays the current reference value. The icon in the column header of the second column represents the entry type (e.g. reference).
Icon	<p>Shows an error or warning marker if there are too few or too many entries in the table.</p> <hr/> <p>TIP Displaying information about the table  Click on the icon to display information about the table itself in the <i>Element Outline</i> view.</p>
Label	<p>Displays the name of the list and indicates the saved status of entries.</p> <hr/> <p>NOTE Blue labels indicate unsaved values  If the list contains unsaved changes, the label turns blue.</p>
Table tab	Displays the configuration parameters in a table.
Tool bar	<p>Depending on the table, a tool bar with icons is available. Unavailable icons are grayed out. The following list explains the usage of the icons:</p> <ul style="list-style-type: none"> ▶  Select to move selected elements upwards in the list. ▶  Select to move selected elements downwards in the list. ▶  Select to add new elements with default values. ▶  Select to remove selected elements.

Name	Explanation
	<ul style="list-style-type: none"> ▶ Select to duplicate selected elements. ▶ Select to add required elements. ▶ Select to open row in a separate editor or window. ▶ Select to open the <i>Bulk Change</i> dialog, which allows to edit various rows simultaneously. This button is enabled if at least one row is selected. ▶ Select to edit visibility of columns.

To help increase navigation and editing speed, tables provide the following keyboard shortcuts:

Shortcut	Explanation
Arrow keys	Use to change the selected row.
Control+A	Selects all rows in the table.
Delete	Deletes the rows selected.
Insert	Inserts a new row at the end of the table.
F2	Starts editing the first editable column in the row selected.
▶ Tab	Switches between edited cells.
▶ Shift+Tab	
Space	Opens the editor of the container in the row selected.

TIP

To add a new entry to the table, double-click the unoccupied part of the table. A new entry is automatically inserted at the end of the list.



6.4.1.5. Optional elements

An AUTOSAR configuration contains optional configuration elements. You may decide if you want these configuration elements to be part of the configuration or not.

- ▶ To add the element to the configuration, select the toggle button when it is red:



Figure 6.25. The GUI elements of the Editor view: red toggle button

If you select the red toggle button, it turns green. The element is added to the configuration.

- ▶ To delete the element from the configuration, select this toggle button when it is green.



Figure 6.26. The GUI elements of the Editor view: green toggle button

If you select the green toggle button, it turns red. The element is no longer part of the configuration. But the configured values of the element (and its possible child elements) are saved. You may add it again at any time, because it is not permanently deleted.

If an element is switched off (not part of the configuration), the GUI of the element is grayed out.

6.5. Keyboard shortcuts

EB tresos Studio provides the following keyboard shortcuts which might help you speed up your work:

6.5.1. Shortcuts for the Workbench

Keyboard shortcut	Task
F1	Opens the EB tresos Studio online help.
Ctrl + N	Opens the <i>New Project Wizard</i> .
Ctrl + E	Opens a context menu in which you may switch between open editors.

6.5.2. Shortcuts for views

Keyboard shortcut	Task
Ctrl + F10	Opens a context menu in which you may customize the currently active view.
Ctrl + M	Maximizes the currently active view.

6.5.3. Shortcuts for editors and projects

Keyboard shortcut	Task
Ctrl + S	Saves the changes made in the currently open editor(s).

Keyboard shortcut	Task
Ctrl + T	Opens the <i>Search</i> dialog in which you may search for configuration parameter names and their values within files or within the data model.
Ctrl + Z/Y	Undo/redo the last change in your editor.
Alt + left arrow/right arrow	The cursor jumps to the previous/next tab of the editor.

6.5.4. Shortcuts for tables

Keyboard shortcut	Task
Ctrl + A	Selects all rows in the currently active table.
F2	Moves the cursor to the first editable column in the row selected.
Space	Opens the editor of the container in the row selected.
Tab/ Shift + Tab	Moves the cursor to the next editable cell.
Insert	Inserts a new row below the last edited row of the table.
Delete	Deletes the selected row.

7. Use cases

7.1. Overview

The EB tresos Studio user's guide provides information on operating the tool via the graphical user interface and also from the command line.

If you have never worked with EB tresos Studio, we encourage you to start reading [Chapter 4, Background information](#) before proceeding to the other chapters.

If you have worked with EB tresos Studio and are familiar with the concepts or you just need to look up something, please proceed to the task-oriented instructions.

- ▶ The chapter [Section 7.2, “Changing the EB tresos Studio setup”](#) describes how to get information about the current set-up of the tool and how to change the settings.
- ▶ If you want to create your first configuration project, or want to learn more about what you can do with projects, you should take a look at [Section 7.3, “Working with projects”](#).
- ▶ The chapter [Section 7.6, “Editing module configurations”](#) describes how you can create an ECU configuration for an AUTOSAR module and what you can do with it.
- ▶ The chapter [Section 7.7, “Upgrading projects and configurations”](#) covers upgrading to a newer AUTOSAR version, or to a new version of a module.
- ▶ All tasks around editing an ECU configuration are described in [Section 7.8, “Editing parameters of a module configuration”](#).
- ▶ How source code can be generated out of an ECU configuration is explained in [Section 7.9, “Generating code for projects”](#).
- ▶ The chapter [Section 7.10, “Importing and exporting”](#) provides information around the topic *importing and exporting configuration data*:
 - ▶ The chapter [Section 7.10.2, “Importing and exporting configuration data”](#) provides step-by-step instructions on how to import data in general.
 - ▶ The chapter [Section 7.10.2.2, “Importing and exporting configuration data from and to the AUTOSAR format”](#) shows how to import or export project configuration data in/from the AUTOSAR format.
 - ▶ The chapter [Section 7.10.2.6, “Importing CAN configurations from the DBC format”](#) shows how to import data from DBC file format and explains the mapping of the DBC information to AUTOSAR tags.
 - ▶ Accordingly, the chapter [Section 7.10.2.7, “Importing from the LDF format”](#) shows you how to import LDF data.
 - ▶ [Section 7.10.2.5, “Importing from the OIL format”](#) shows you how to import OIL files.

- ▶ EB tresos Studio also provides a command line mode, which is described in the chapter [Section 7.11, "Working with the command line"](#). The command line mode includes two sub-modes:
 - ▶ project mode for already existing code and
 - ▶ legacy mode for projects that have not been started yet.

7.2. Changing the EB tresos Studio setup

This section describes how you can change several aspects of the EB tresos Studio setup, including installing new licenses, finding information about installed modules and plugins and changing the program-wide preferences.

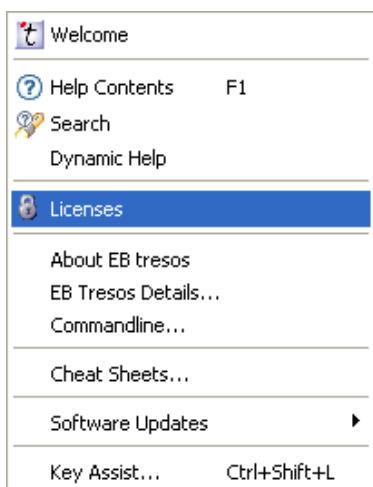
7.2.1. Managing licenses

EB tresos Studio provides a dialog window which lists all licenses currently installed and lets you remove or add licenses. You can also use this dialog to view information about each license like the licensed features or the expiry date and you can verify if the license works properly.

To open the *Licenses* dialog:

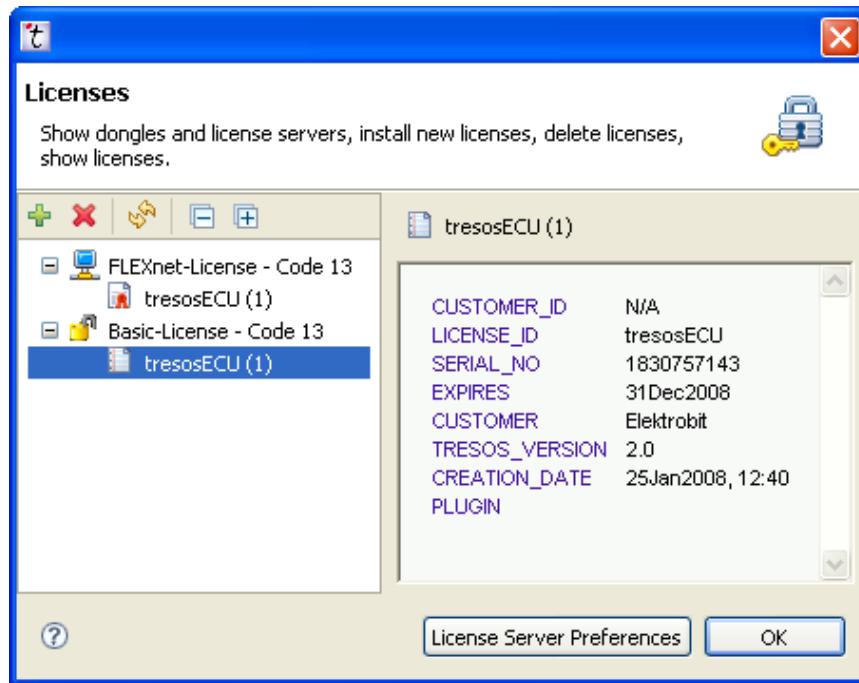
- ▶ Select the **Help** menu.

A submenu opens up.



- ▶ Select *Licenses* from the submenu.

The *Licenses* dialog opens up.



The dialog is split into two parts: a list of licenses on the left and an area where detailed information of a license is displayed. To view the detailed information you need to select one of the license on the left.

The list on the left is grouped by license type and customer ID. Possible license types are *FLEXnet License* (in case of a floating license), *Wibu-License* (in case of a dongled license) and *Basic-License* (in case of an undongled license). A *floating* license means that the license is verified by contacting a server in your computer network. A *dongled* license means that the license is verified by communicating with a hardware device plugged into your computer's USB port (see the picture below). It is possible to have more than one license of one type. In this case, there are multiple items displayed in one group.

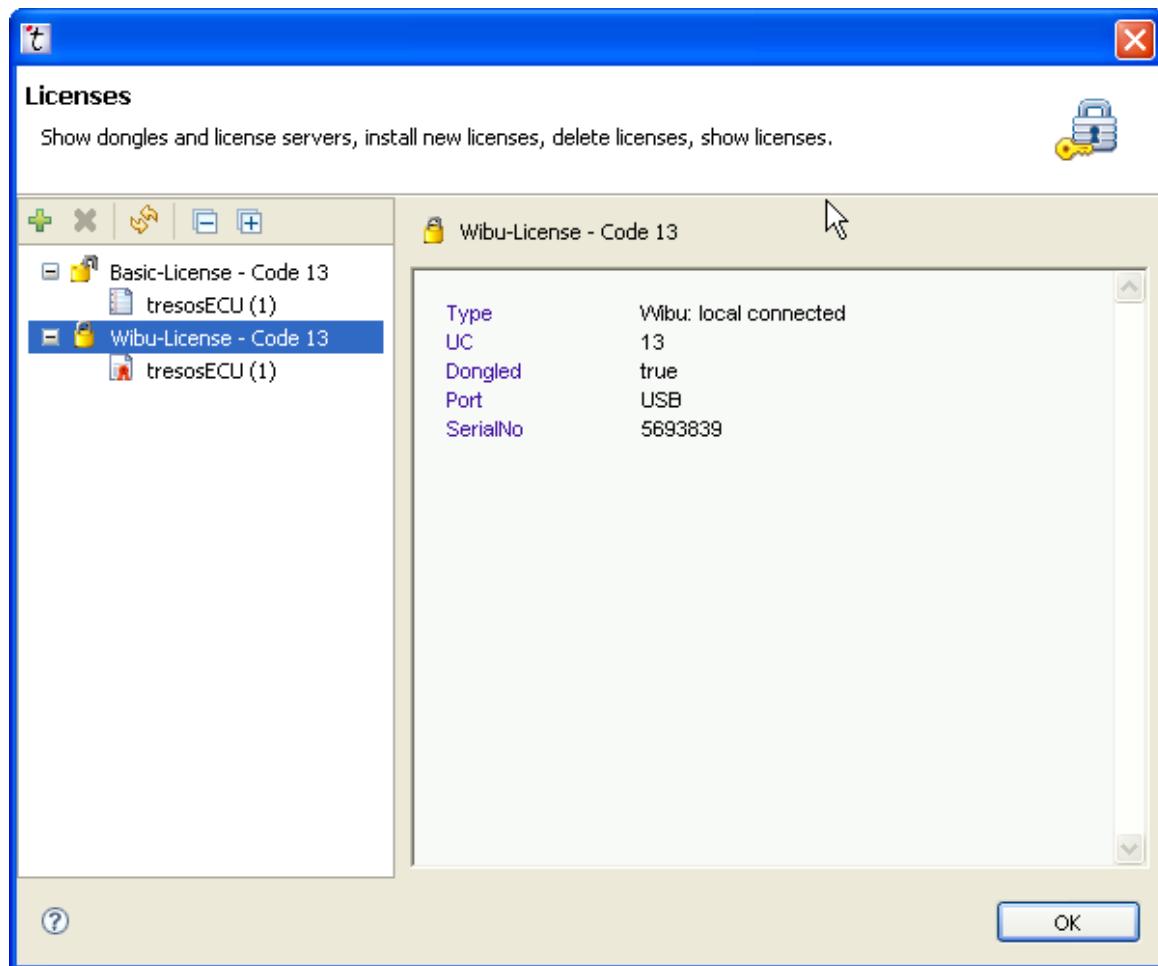


- ▶ Select a license from the list.

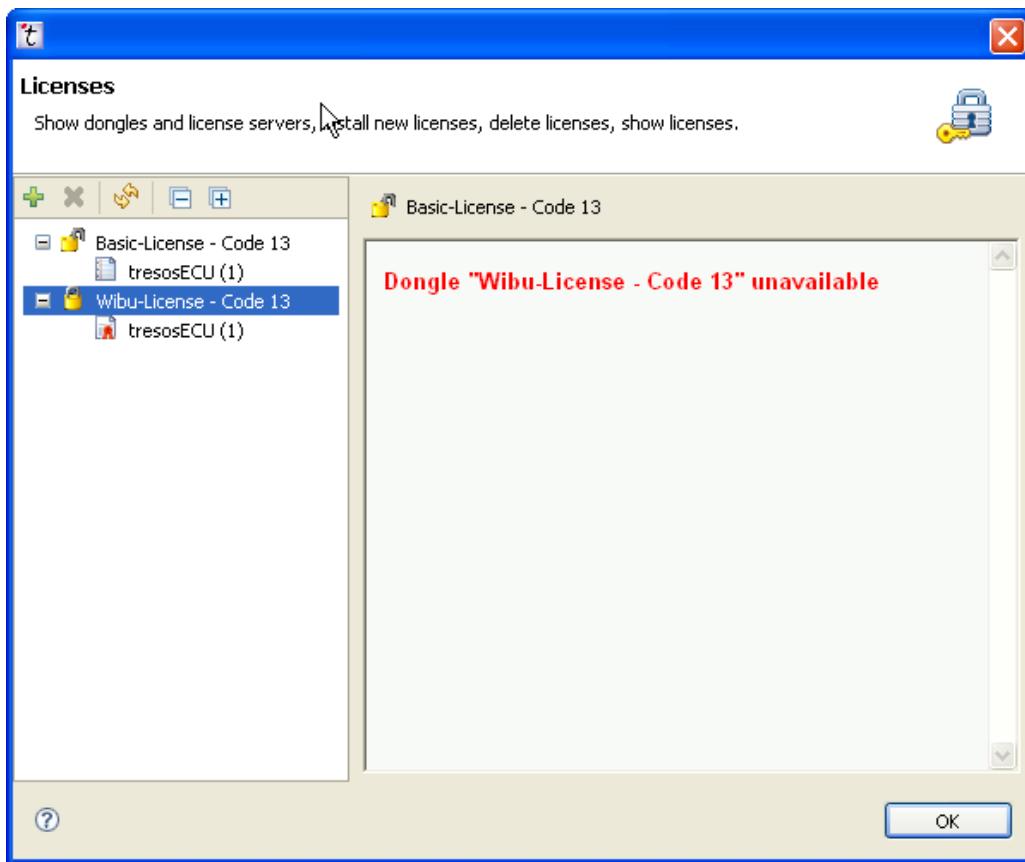
The license details are displayed on the right side of the *Licenses* window.

- ▶ The the icon expands the license list completely. Use this if you have a lot of licenses installed and want to see them all.
- ▶ The icon collapses the license list, so that only the groups are visible. Use this if you have a lot of groups expanded and want to revert the display to its initial state.

- ▶ If you use Floating Network licenses, press the button **License Server Preferences** to configure the License Server. See [Section 7.2.3.7, “Setting License Server preferences”](#) for details.



Example 7.1. The example shows information about a WibuKey dongle.



Example 7.2. The example shows a dongle that cannot be accessed by EB tresos Studio.

7.2.1.1. Adding a license

If you have purchased a new license, you have to add it to EB tresos Studio before you can use it. To add a license:

- ▶ Select the green cross  icon.
- A file dialog window opens up.
- ▶ Browse your computer for licenses in the `.lic` format.
- ▶ Select the license.
- ▶ Select **Open**.

The license is added to your list in the *Licenses* window.

7.2.1.2. Deleting a license

To delete a license:

- ▶ Select the license from the list of the *Licenses* window.
- ▶ Select the  icon.

This deletes the selected license from your list.

7.2.1.3. Refreshing license information

To refresh license information:

- ▶ Select the icon with two arrows  to update system information.

Your computer installs all available licenses from the configuration directory and updates the information whether and which licenses and dongles are available.

7.2.1.4. Auto-installing licenses

EB tresos Studio searches for license files in both the `bin` and the `configuration` directories under the `$TRESOS_BASE` once at startup and any time the license information is refreshed manually in the license dialog as described in [Section 7.2.1.3, “Refreshing license information”](#). So you may install a license file by just copying it to one of these directories.

- ▶ If EB tresos Studio is not running while you copy the license file, EB tresos Studio detects the new license during the next startup and automatically installs it.
- ▶ Otherwise - if EB tresos Studio is already running - open the license dialog and refresh the list as described in [Section 7.2.1.3, “Refreshing license information”](#). The newly installed license then appears in the list.

NOTE**License filename is changed in the `bin` directory; license is deleted in the configuration directory**

After installing the license(s) EB tresos Studio...

- ▶ ...changes the file ending(s) of the license file(s) under the `bin` directory from `.lic` to `.installed.lic`. The thus named file is then ignored when EB tresos Studio automatically searches for new licenses upon startup.
- ▶ ...deletes the license file(s) in the `configuration` directory. Therefore copy the licenses into the `bin` directory rather than in the `configuration` directory.

TIP**Specify license server for floating licenses**

The auto-install functionality is intended to be used in automated environments, where the effort of manually installing licenses is too expensive. If you install a floating license, you additionally have to specify the license server preferences. This can also be done file-based with the `flexlm.cfg` as described in [Section 7.2.3.7, “Setting License Server preferences”](#).

7.2.2. Viewing setup information

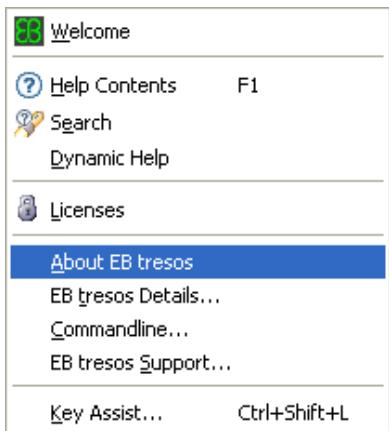
EB tresos Studio provides two dialogs which display information about the program and its setup: the *About EB tresos* dialog and the *EB tresos Details* dialog. This section will show you how these dialogs can be opened and will describe the information they contain.

7.2.2.1. Opening the About EB tresos dialog

To open the *About EB tresos* dialog:

- ▶ Select the **Help** menu.

A context menu opens up.



- ▶ Select *About EB tresos* from the context menu.

The *About EB tresos* dialog opens up.

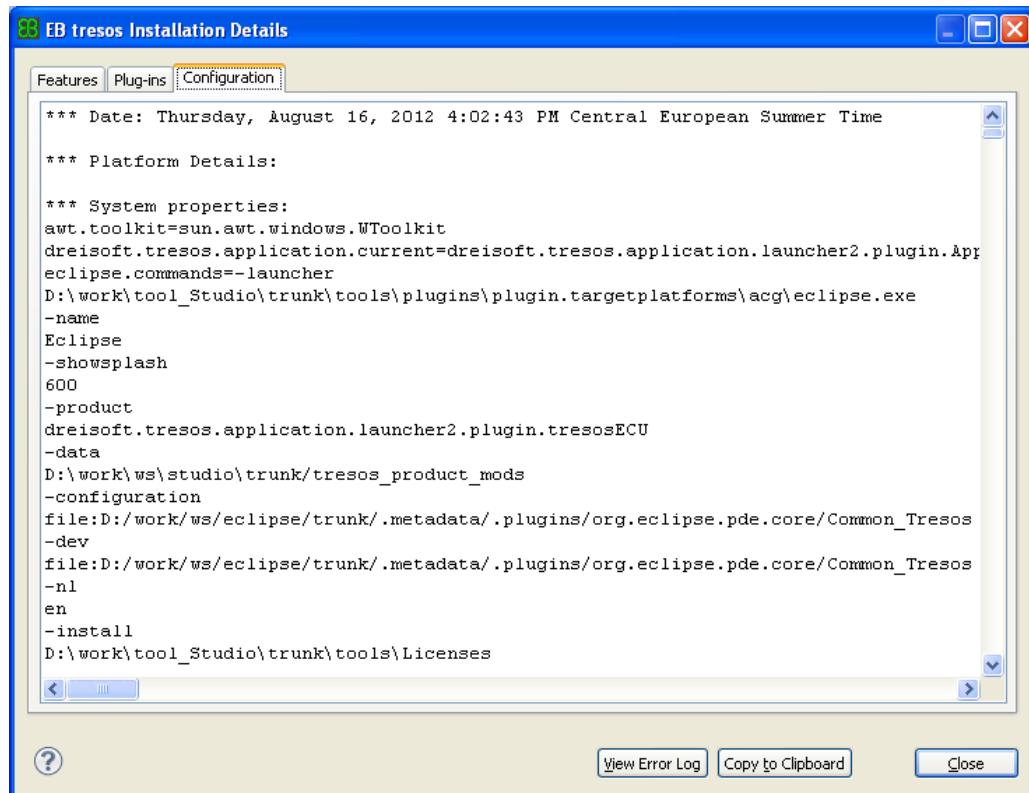


7.2.2.2. Finding program startup parameters and system property settings

To view internal information about how EB tresos Studio was launched and what internal system properties are set:

- ▶ In the *About EB tresos* window, select the **Installation Details** button.

The *EB tresos Installation Details* window opens up.



On the tab *Configuration*, this window provides detailed configuration information about program startup parameters and system property settings. In case of problems this information can be useful for EB support.

7.2.2.3. Opening the *EB tresos Details* dialog

To open the *EB tresos Details* dialog:

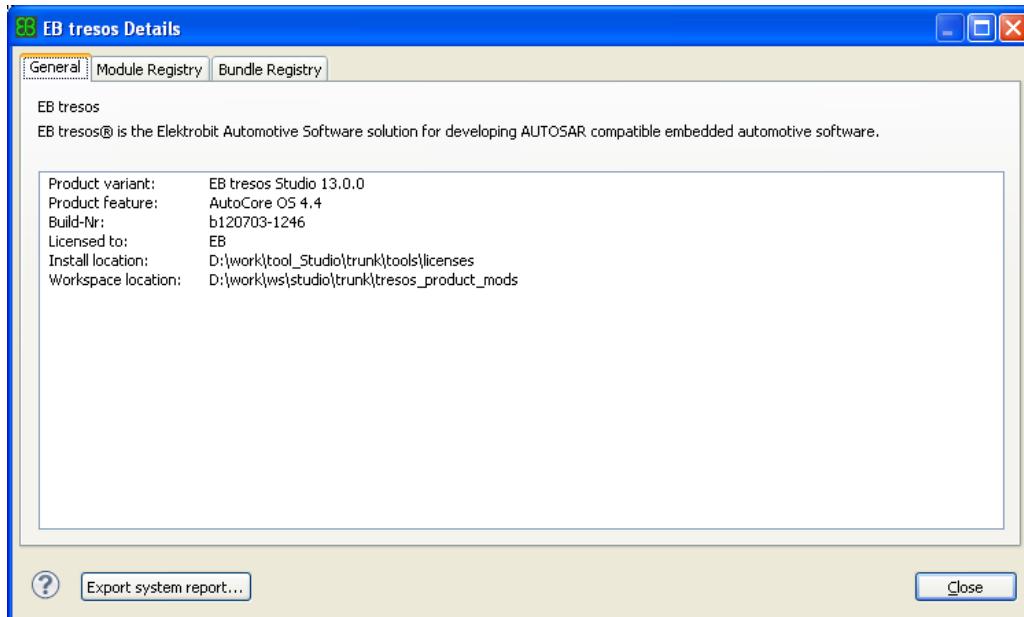
- ▶ Select the **Help** menu.

A context menu opens up.



- ▶ Select *EB tresos Details...* from the context menu.

The *EB tresos Details* window opens up.



7.2.2.4. Finding the location of the installation and the workspace

Select the *General* tab to display:

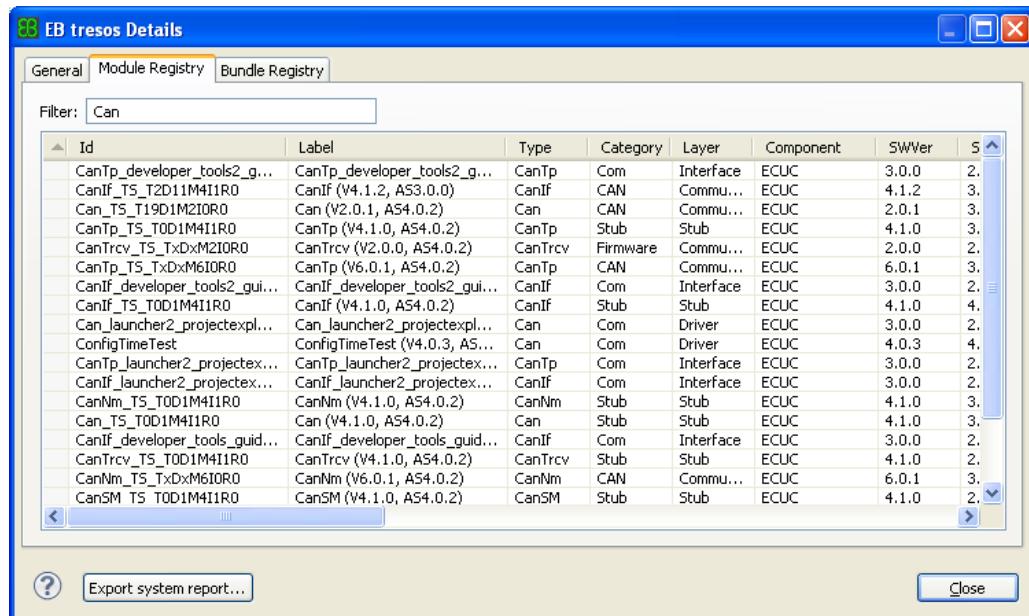
- ▶ the build number,
- ▶ the name of the licensee,
- ▶ the installation location on your computer, and
- ▶ the workspace location on your computer.

7.2.2.5. Finding all installed modules

To find out if a specific module is part of your installation and which Autosar specification version it implements:

- ▶ Select the Module Registry tab.
- The list below the tab displays all installed modules.
- ▶ Type the text you are looking for into the *Filter* textbox.

The table in the *Module Registry* tab now shows just entries that contain text that matches your entered text. To view all entries again, just clear the text control by deleting its content.

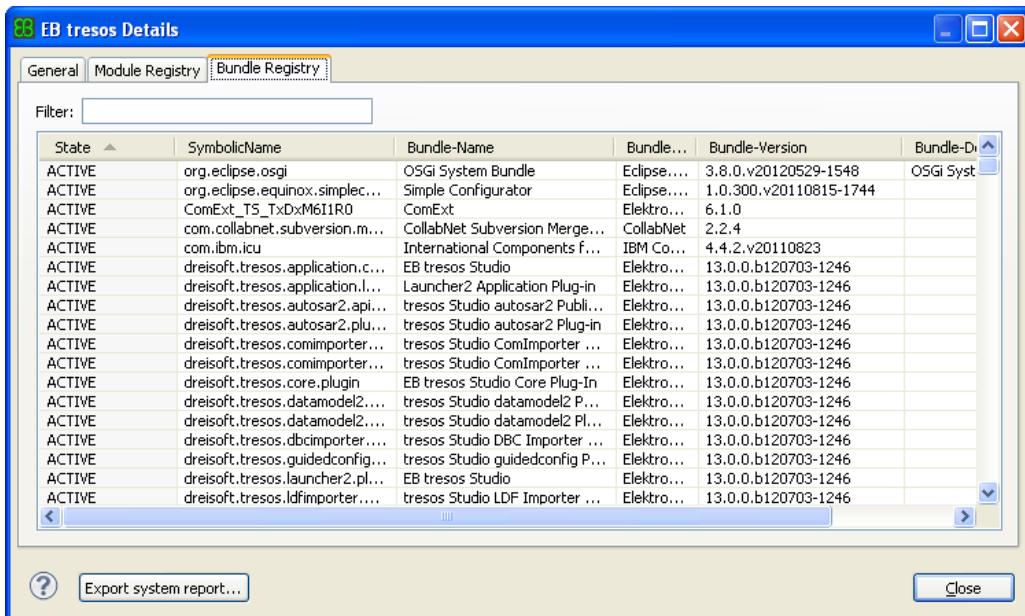


7.2.2.6. Finding all installed plug-ins

To find out if a specific plug-in is part of your EB tresos Studio installation and if it is working properly:

- ▶ Select the *Bundle Registry* tab.
- The list below the tab displays all installed plug-ins.
- ▶ Type the text you are looking for into the *Filter* text control.

The table in the *Bundle Registry* tab now shows just entries that contain text that matches your entered text. To view all entries again, just clear the text control by deleting its content.

**NOTE**

This list shows all plugins from the Eclipse framework upon which EB tresos Studio is built as well as EB tresos Studio core libraries. If a plugin could not be loaded properly, the line is colored red.

7.2.2.7. Exporting a system report of your EB tresos Studio installation

If you are having problems with your EB tresos Studio installation, it is useful for EB support to have as much information about your setup as possible. You can help EB support by attaching a system report of your EB tresos Studio installation to any support ticket. To generate this report as a text file:

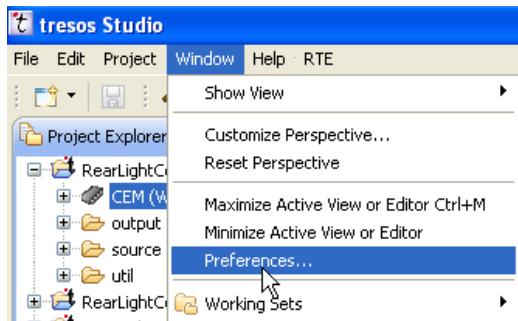
- ▶ In the *EB tresos Details* window, select the **Export system report...** button.

This opens up a dialog window, with which you may save the data as file to your computer.

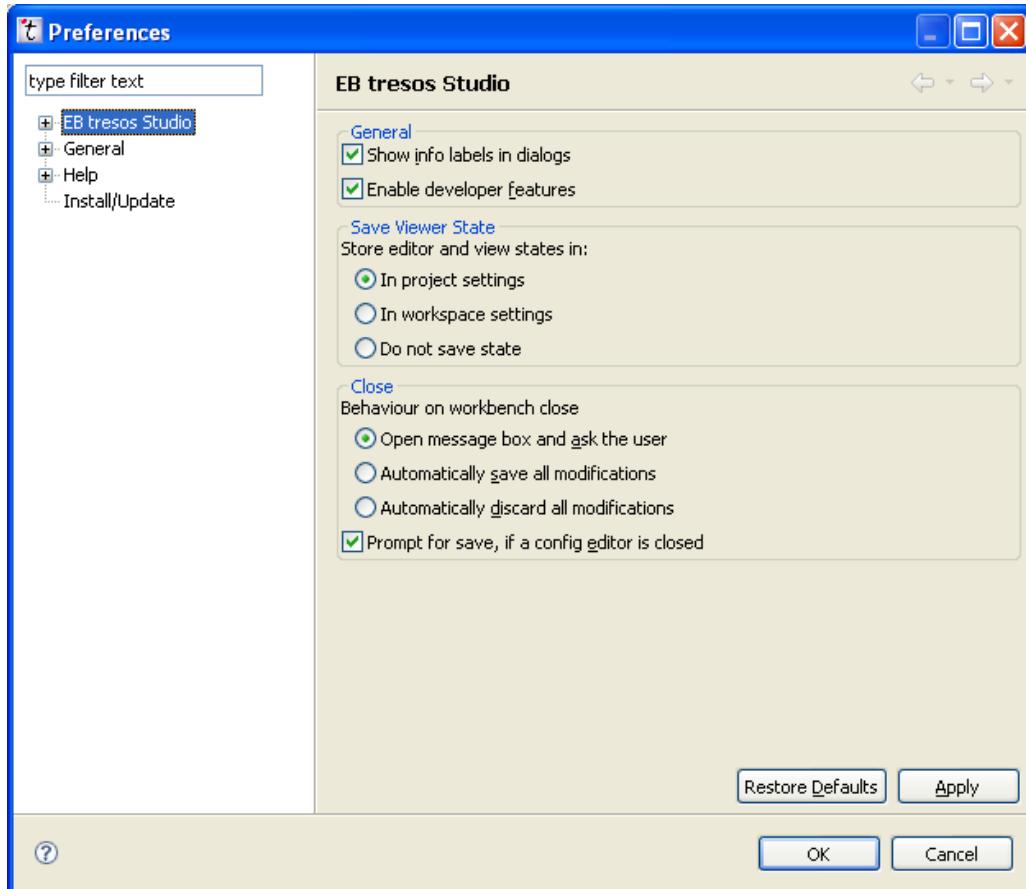
7.2.3. Setting preferences

To globally store settings for a workspace set your preferences in the Window menu.

- ▶ Select **Window**.
- ▶ Select *Preferences* from the drop down menu.



The *Preferences* window opens up.



- ▶ Select *EB tresos Studio* from the tree view on the left of the window.
- ▶ Double-click *EB tresos Studio*.

A subtree opens up underneath the tree item *EB tresos Studio*.

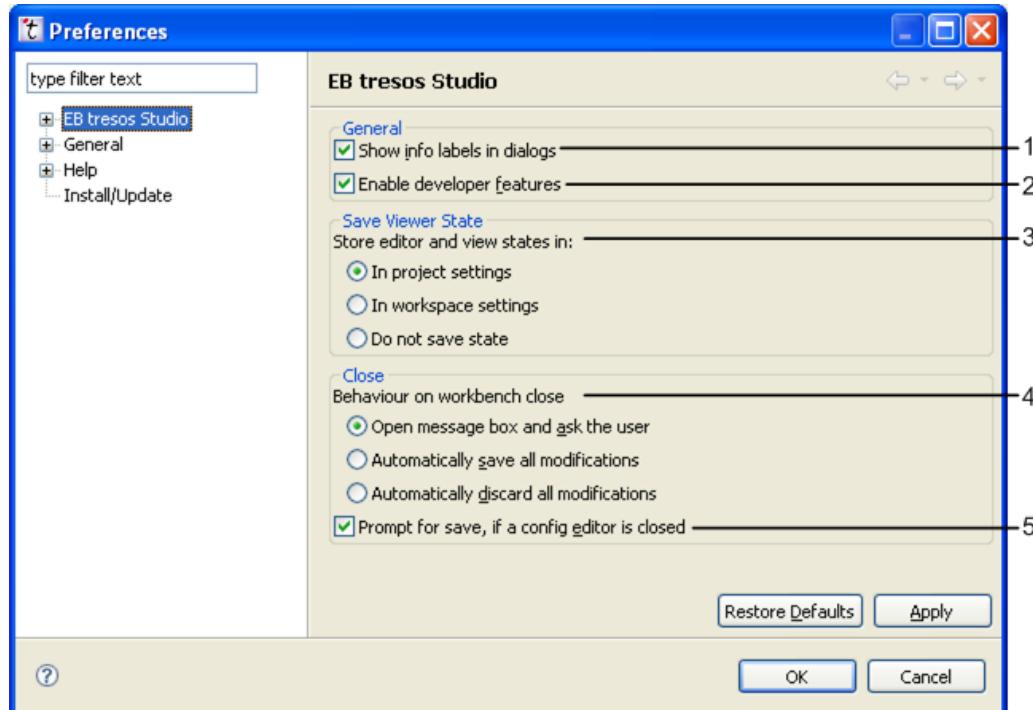
- EB tresos Studio
 - Assistance dialogs
 - AUTOSAR
 - Configuration
 - Error Log
 - Licenses
 - Parameter comments

The following subsection will explain the preference pages related to items of the subtree. On any page, you can press the **Restore Defaults** button to restore the default values of the page. You can press the **Apply** button to store the changes you made so far. To save the changes and close the complete dialog, press the **Ok** button. To cancel any changes you made and to close the dialog, press the **Cancel** button.

For more information on the other preference pages, refer to the [Eclipse Workbench User Guide](#). This Eclipse document is included in the EB tresos Studio onscreen help, which opens up when you press **F1**.

7.2.3.1. Setting general preferences

To set preferences about the general behavior of EB tresos Studio, use the following settings on the *Preferences EB tresos Studio* window:



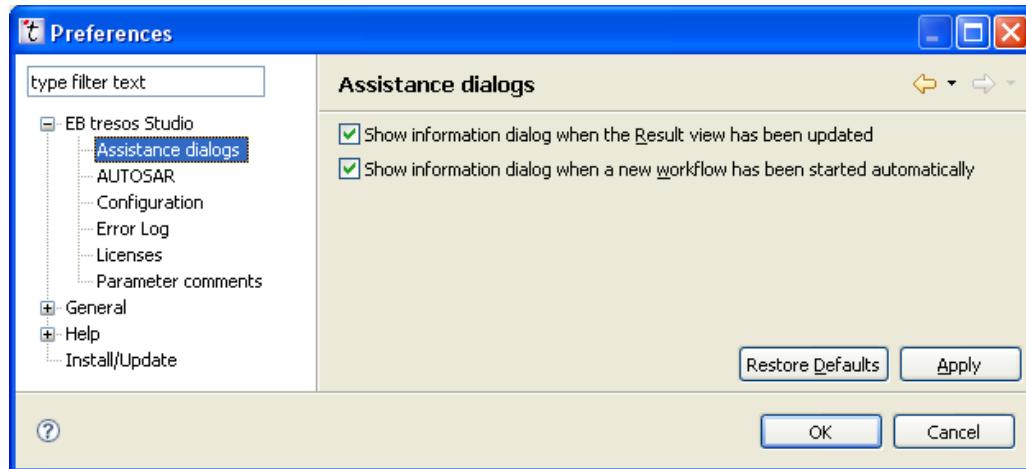
Number	Checkbox/Radiobutton	Instruction
1	Show info labels in dialogs	Select to enable help text for dialogs and wizards.
2	Enable developer features	Enables some features that are relevant for developers, who develop their own Autosar Basic Software modules. This is

Number	Checkbox/Radiobutton	Instruction
		described in a more detailed way in the EB tresos Studio developer's guide, which is located in your <code>\$TRESOS_BASE\doc\2.0_EB_tresos_Studio</code> directory.
3	Store editor and view states in:	<p>The configuration editors of EB tresos Studio are able to preserve their current state, e.g. columns shown in tables, column width, number format selected between editing sessions, by storing the state to disk when the editor is closed.</p> <p>The Store editor and view states in allows to configure where this information is saved to.</p> <p>In project settings The editor state is saved inside the EB tresos Studio project the editor belongs to. The state is written to files in a directory <code>.settings</code> beneath the project directory of the project.</p> <p>In workspace settings The state is saved into the meta-data area of the workspace currently opened. The meta-data area is stored in a hidden directory called <code>.metadata</code> beneath the workspace root directory.</p> <p>Do not save state The editor does not write or load any state.</p>
4	Behaviour on workbench close	<p>Open message box and ask the user Select to be asked every time you close the workbench whether you want to save changes. This will prevent you from closing the program with unsaved projects.</p> <p>Automatically save all modification Select if you want to save changes to your projects automatically upon closing. No dialog window will appear.</p> <p>Automatically discard all modifications Select if you do not want to save changes to your projects automatically and do not want to be reminded to save by a dialog window.</p>
5	Prompt for save...	Select if you want the configuration editor to remind you to save your settings before closing.

7.2.3.2. Setting Assistance dialog preferences

To set Assistance dialog preferences:

- ▶ select the *Preferences EB tresos Studio* subpage *Assistance dialogs* and use the following settings:



- ▶ Select **Show information dialog when the Result view has been updated** to get informed about new results in the *Results* view.

If checked, an information dialog is shown each time after running unattended wizards or assistance dialogs to inform you about new results in the *Results* view.

- ▶ Select **Show information dialog when a new workflow has been started automatically** to get informed if the current workflow automatically started a new workflow.

If checked, an information dialog is shown if a new workflow has been started automatically in the *Workflow* view.

7.2.3.3. Setting preferences for parameter comments

To set preferences for parameter comments:

- ▶ Select *Parameter comments* from the *EB tresos Studio* preferences.

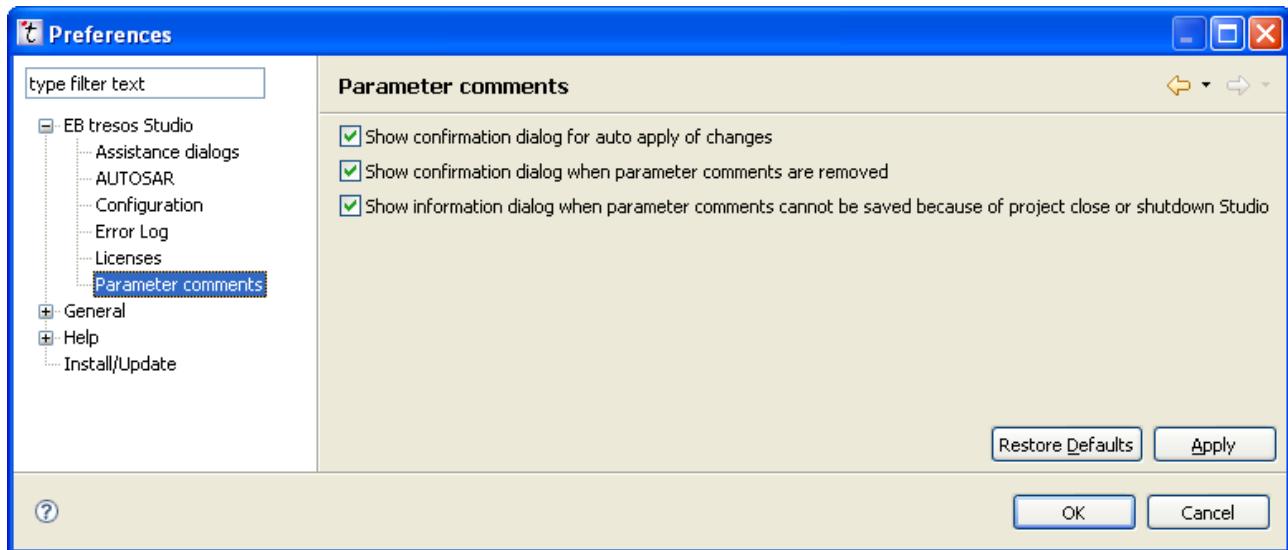


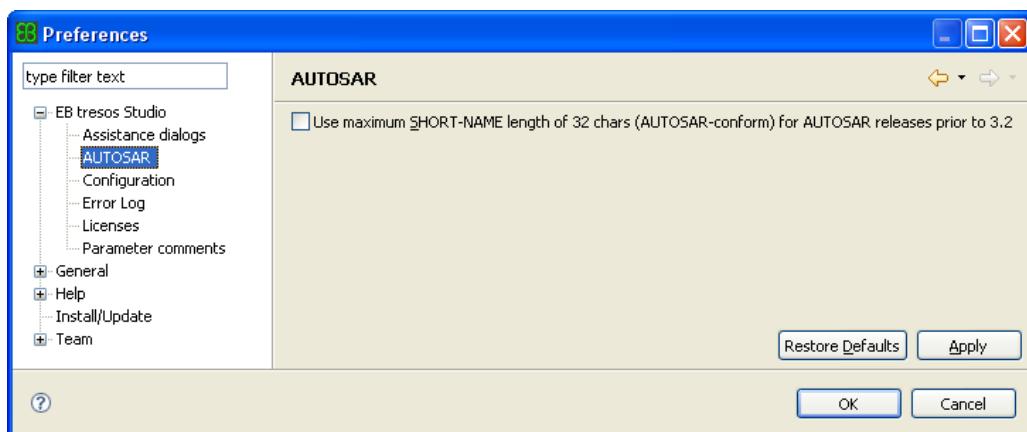
Figure 7.1. Setting up preferences for parameter comments

- ▶ If you want a confirmation dialog to open up each time your changes to a parameter comment have not been saved in the *Comments for <parameter name>* dialog, select **Show confirmation dialog for auto apply changes**.
- ▶ If you want a confirmation dialog to open up each time a parameter comment is removed, select **Show confirmation dialog when parameter comments are removed**.
- ▶ If you want an information dialog to open up each time an unapplied parameter comment cannot be saved (e.g. during shutdown of the application or project close), select **Show information dialog when parameter comments cannot be saved because of project close or shutdown EB tresos Studio**.

7.2.3.4. Setting AUTOSAR preferences

To set AUTOSAR preferences:

- ▶ select the *Preferences EB tresos Studio* subpage **AUTOSAR** and use the following settings:



- ▶ Select **Use maximum SHORT-NAME length of 32 chars (AUTOSAR-conform)** for AUTOSAR releases **prior to 3.2** to restrict the length of SHORT-NAMES to 32 chars according to the AUTOSAR specification for releases prior to 3.2.

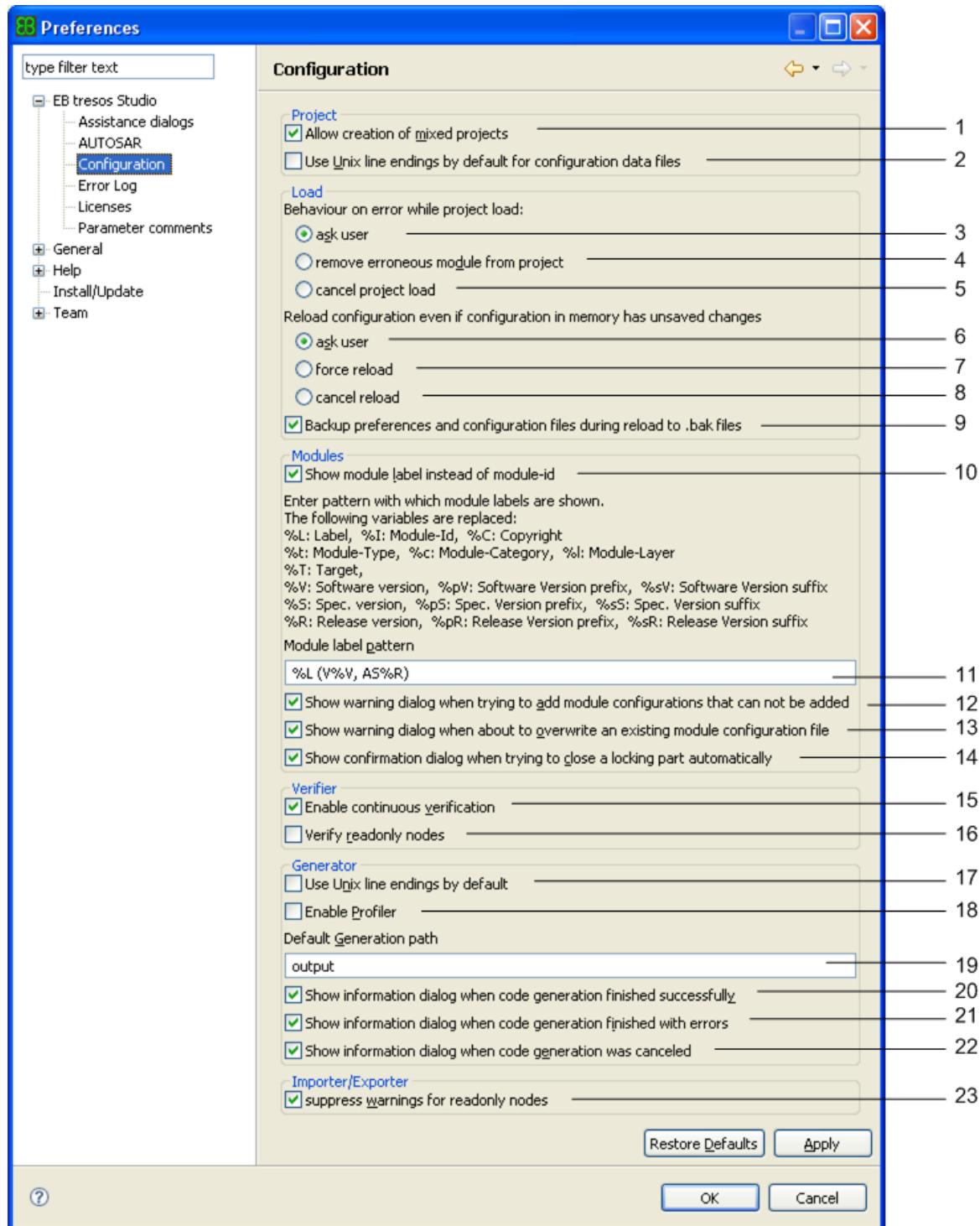
If unchecked, longer names are allowed within EB tresos Studio. To export such a configuration to an AUTOSAR file format, the validation must be deactivated. Problems may occur if other tools try to import this configuration.

Independent from this option, 128 characters are allowed for SHORT-NAMES for AUTOSAR release 3.-2 or higher (conforms to the AUTOSAR specification).

7.2.3.5. Setting configuration preferences

To set configuration preferences:

- ▶ select the *Preferences EB tresos Studio* subpage *Configuration* and use the following settings:

Figure 7.2. Preferences subpage *Configuration*

Number	Label	Instruction
1	Allow creation of mixed projects	Select to configure modules with different release versions. A mixed project may contain module configurations

Number	Label	Instruction
		with different release versions. If you enable this feature, the <i>Release Version</i> combobox in the project wizard (see Section 7.3.1, “Creating a new project”) contains the entry <i>mixed</i> .
2	Use Unix line endings by default for configuration files	Select to use Unix line endings instead of platform line endings in configuration data files. This is a project-specific configuration. You may change it for each project in the preferences. The configuration data files are located inside the config directory of the project directory. The Exporters also use this setting.
3	ask user	<p>If you try to load a project and an error occurs while loading one module, you are asked if you want to either</p> <ul style="list-style-type: none"> ▶ remove this module from the project and continue loading ▶ or to cancel the load process.
4	remove erroneous module from project	If you try to load a project and an error occurs while loading one module, the erroneous module is persistently removed from the project. The project load continues without that module.
5	cancel project load	If you try to load a project and an error occurs while loading one module, the load process is canceled. After that, you can resolve the problem manually and try loading the project again.
6	ask user	Enable this preference, if you want to be asked whether you want to discard your changes or cancel the reload process, when you are about to reload a project with unsaved changes.
7	force reload	Enable this preference, if you want to always discard changes without further notice, when you are about to reload a project with unsaved changes.
8	cancel reload	Enable this preference, if you always want the reload process to be canceled, when you are about to reload a project with unsaved changes. For further information on reloading a project, refer to Section 7.3.6.4, “Reloading a project” .
9	Back up preferences and configuration files during reload as .bak files	Select this preference, if you want EB tresos Studio to backup the preferences- and configuration files, when

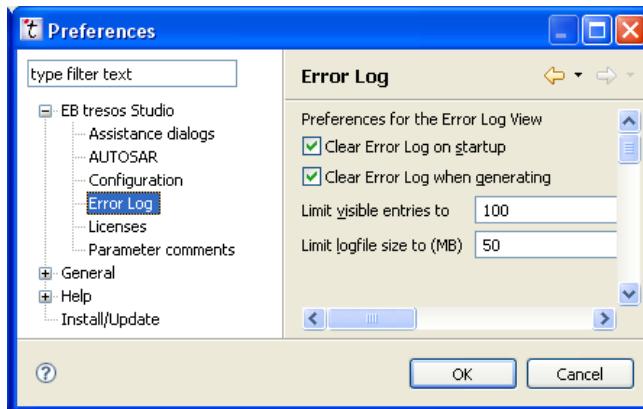
Number	Label	Instruction
		you reload an already loaded project. This backup then contains the current in-memory representation, including any unsaved changes. The backup files are stored in the same directories as the original files with a .bak file extension.
10	Show module label instead ...	If this checkbox is checked, the GUI uses the pattern defined in (11) to display a module. This setting is visible e.g. in the <i>Project Browser</i> or the <i>Add Module</i> dialog.
11	Module label pattern	Enter the pattern that defines what the module label should look like in the GUI.
12	Show warning dialog when trying to add module configurations that can not be added	If checked a warning dialog is shown any time you try to add a module configuration that can not be added, e.g. if there is already a configuration for that module and it does not allow multiple configurations.
13	Show warning dialog when about to overwrite an existing module configuration file	If checked a warning dialog is shown any time you uncheck the "Load existing file" option when adding a new module configuration.
14	Show confirmation dialog when trying to close a locking part automatically	If selected, a confirmation dialog is shown any time a function is chosen while a guided configuration dialog is blocking the project.
15	Enable continuous verification	<p>Select to enable/disable continuous verification.</p> <hr/> <p>NOTE  If enabled, the verification of the configuration will run continuously in the background for all loaded projects. Detected problems will be displayed in the <i>Problems View</i>.</p> <hr/> <p>WARNING  Even if you disable continuous verification, EB tresos Studio will still verify the configuration, when you start the code generation. No code will be generated, if the verification detects errors in the configuration.</p>
16	Verify read only nodes	Select to verify the read-only nodes (nodes that cannot be changed).
17	Use Unix line endings by default	Select to use Unix line endings in your generated code as default. You may override this workspace-wide default set-

Number	Label	Instruction
		ting in the project preferences for every project individually. See chapter Viewing and changing project properties Section 7.3.7, "Viewing and changing project properties" for details.
18	Enable Profiler	The template based code generator will provide profiling informations during the generator run. For each generated template file a profiler file is generated into the same output directory. Please note that enabling the profiler will slow down the generator process.
19	Default Generation path	Contains the default value for the code generator's output path. If you change this value, the <i>New Project</i> wizard will use this new value as default for the project's Generation path text box. Additionally, the legacy command line mode will also use this setting.
20	Show information dialog when code generation finished successfully	If checked an information dialog is shown after successful code generation.
21	Show information dialog when code generation finished with errors	If checked an information dialog is shown when code generation finished with errors.
22	Show information dialog when code generation was canceled	If checked an information dialog is shown after code generation was canceled.
23	Suppress warnings for readonly nodes	Select if you want to supress warnings for readonly nodes while running importer or exporter.

7.2.3.6. Setting Error Log preferences

To set Error Log preferences:

- ▶ Select the *Preferences EB tresos Studio* subpage *Error Log*

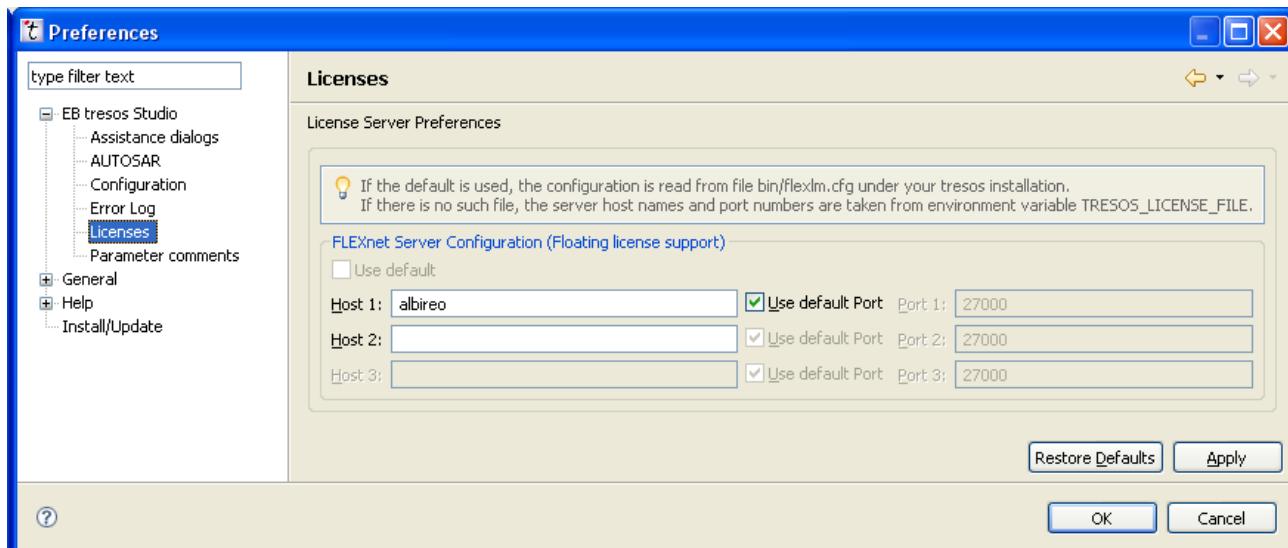


- ▶ Select **Clear Error Log** on startup, if you want the *Error Log* view to contain no messages when you start EB tresos Studio.
- ▶ Select **Clear Error Log** button, if you want to remove all messages from the error log. You can do this e.g. before generating code, to clearly see which messages are logged during code generation.
- ▶ To limit the number of shown entries in the *Error Log* enter the respective number of entries you wish to be displayed in the **Limit visible entries to** combobox. Entering big numbers may slow down the GUI.
- ▶ All errors are written to a logfile called `.tresoslog`. With increasing size of the logfile, the GUI may become slower and the memory usage might increase. The parameter **Limit logfile size to (MB)** provides an upper bound after which the content of the log is discarded and written to a backup file.

7.2.3.7. Setting License Server preferences

To configure the host name and port for the FlexLM license server:

- ▶ Select the *Preferences EB tresos Studio* subpage *Licenses*



Here you can define up to three flexlm license servers hostnames and their ports. This is a three-server redundancy configuration, but usually it is sufficient to only configure one server. It is recommended to use the default-port (automatically checks serveral ports).

- ▶ To load the configuration from the `flexlm.cfg`, click the **Use default** checkbox.
The configuration is being loaded right away.
- ▶ If the file does not exist or does not contain valid data, the configuration is loaded from the system environment variable `TRESOS_LICENSE_FILE`.

In both cases, the loaded values are displayed within the dialog, but are not editable there and thus greyed out.

The `flexlm.cfg` file must be located within the `bin`-directory of your EB tresos Studio installation containing the following information:

```
# Contains the three-server redundancy configuration flexlm
# floating licensing server(s) and port(s). Each line must
# contain the hostname and an optional port. It is recom-
# mended to suppress the port (what means, that the default
# port shall be used
localhost
SomeHost:0815
127.0.0.1
```

The value of the optional system environment variable `TRESOS_LICENSE_FILE` must be of the following syntax:

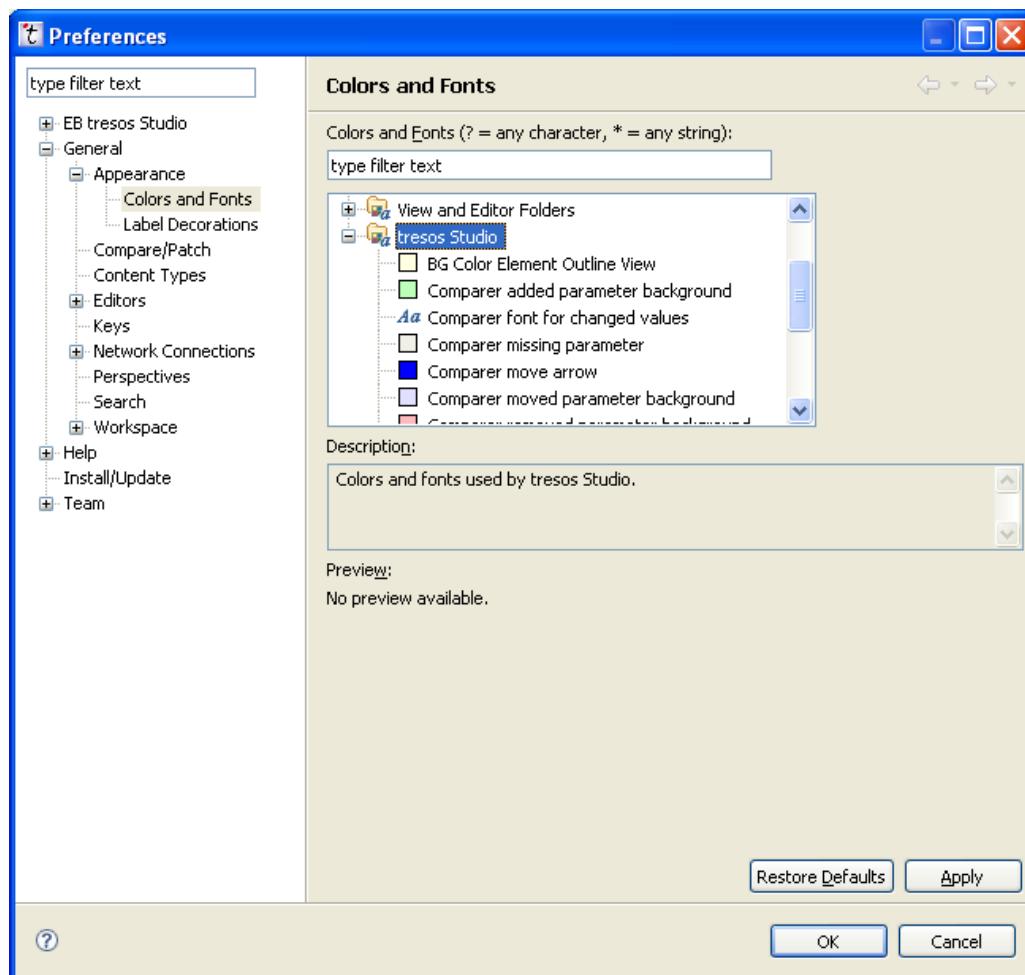
```
[[port]@-]host[;...[;...]]  
e.g.:  
@myhost1;27003@myhost2;myhost123
```

7.2.3.8. Setting colors and fonts

To set colors and fonts:

- ▶ Open the *Preferences* window and select *General* tree view on the left side.
- ▶ Open the tree view, select *Appearance*.
- ▶ Open the tree view under *Appearance* and select *Colors and Fonts*.

You see a categorized list of colors and fonts used in EB tresos Studio.

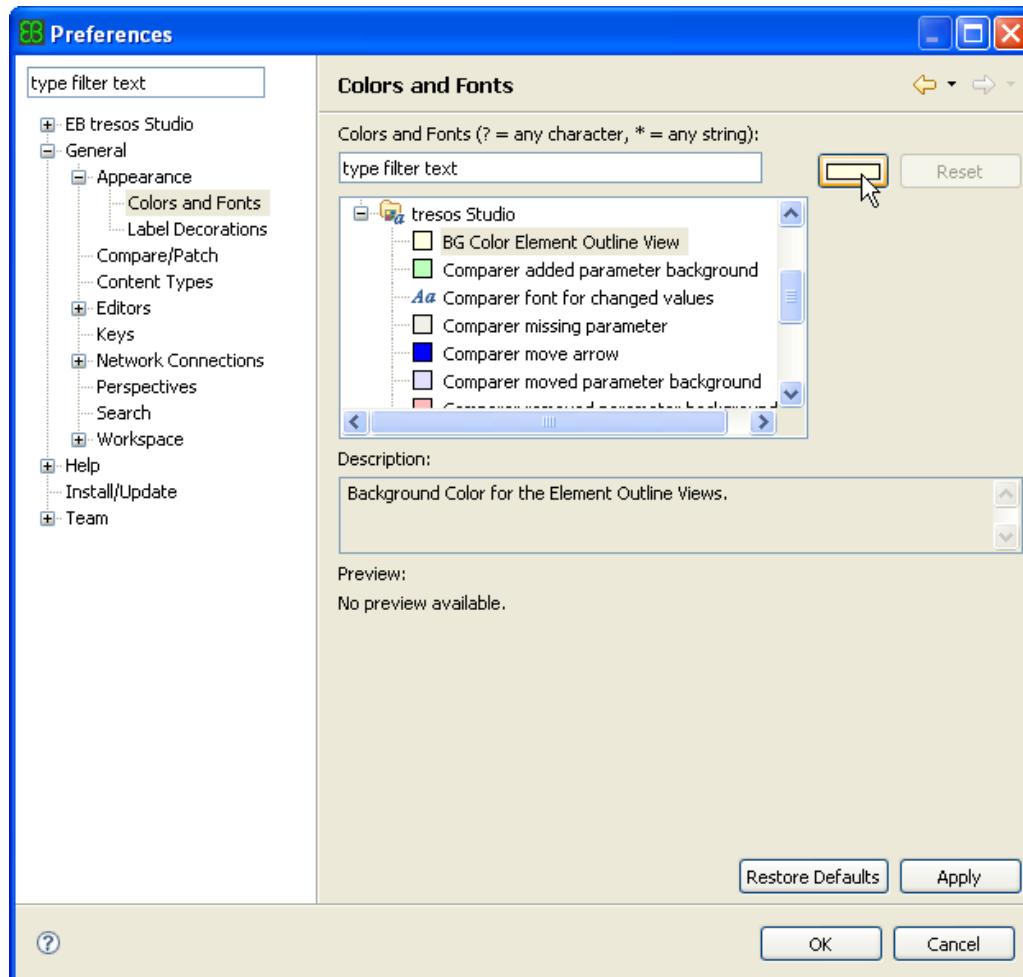


- ▶ Select any of the items to retrieve more information about it in the *Description:* field.

To change any of the colors:

- ▶ Select a color in the *Colors and Fonts* view on the right side.

The color will appear on a button left to a **Reset** button.

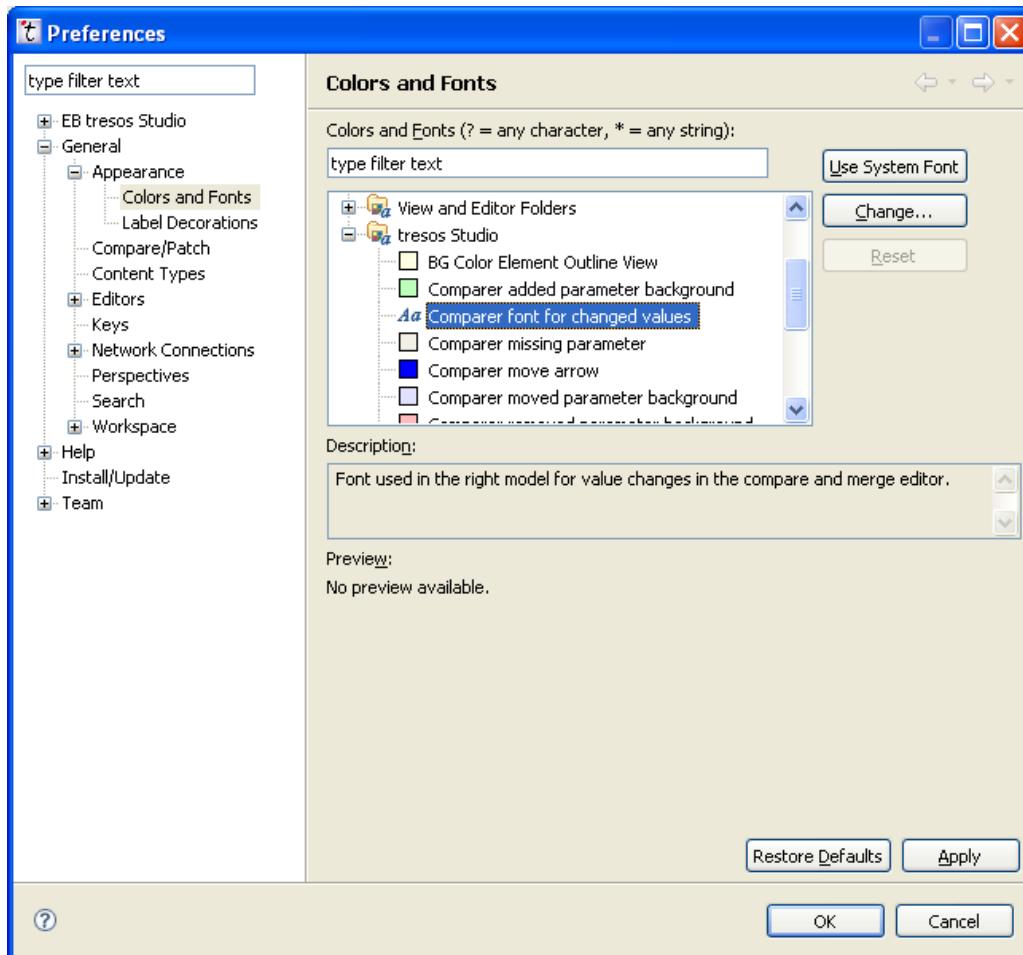


- ▶ Press the colored button.
- A window with color selections opens up.
- ▶ Select a color of your liking and click **OK**.
- EB tresos Studio now displays the new color in the setting you chose.

To change any of the fonts:

- ▶ Select a font in the *Colors and Fonts* view on the right side.

Buttons for adjusting the font will appear on the right.



- ▶ Press the **Change...** button.

A window to adjust the font opens up.

 - ▶ Adjust the font to your likings and click **OK**.
 - ▶ To reset the font to its default setting, select **Reset**.
 - ▶ You can also select **Use System Font** to set the font to a reasonable value chosen by the operating system.
 - ▶ Click **OK**.
- EB tresos Studio now uses the font in the setting you chose.

7.2.4. Working with multiple workspaces to store different projects and settings

A *workspace* in EB tresos Studio contains all your projects and associated information like your preferences (see [Section 7.2.3, “Setting preferences”](#)). If you want to keep some projects completely separated from others,

or if you want to work with different preferences, you can use multiple workspace and switch to the workspace you are planning to use.

NOTE

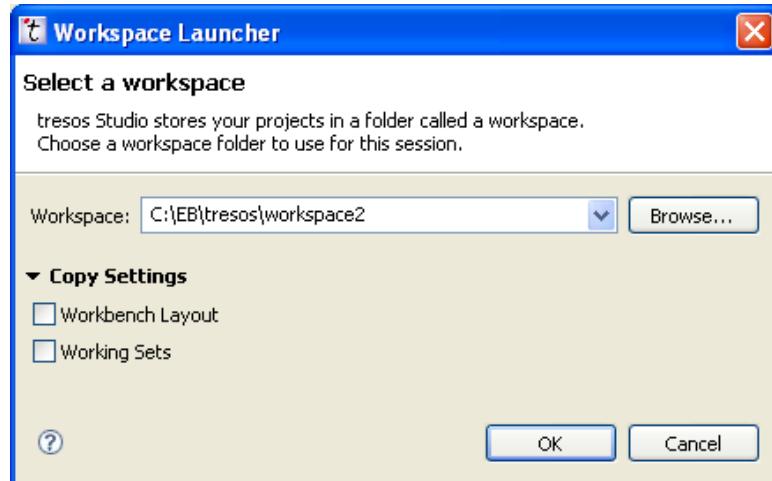
The default workspace is located directly inside your EB tresos Studio installation directory

and is called `workspace`. If you did not change anything you are using this workspace.



7.2.4.1. Switching the workspace

To switch the workspace, open the **File** menu and select *Switch workspace* and then *Other*. The *Workspace launcher* dialog opens:



Use the textbox labeled *Workspace* to type in your new workspace location. You can also use the **Browse** button to open up a Windows explorer to select the new location. If you have already used the switch workspace feature before, the directories used previously are available by clicking the drop-down arrow of the textbox.

If you click on *Copy settings* the dialog expands and you get more options (as depicted in the screenshot). Select **Workbench layout** if you have modified the view layout of EB tresos Studio and want to use the same layout in your new workspace. If you do not choose this option, the default layout will be used and EB tresos Studio looks like when you started it for the first time.

7.2.4.2. Finding out which workspace directory is used

If you want to check which workspace directory EB tresos Studio is currently using, open the *EB tresos Details* dialog (see [Section 7.2.2.4, “Finding the location of the installation and the workspace”](#)). In the list on the *General* tab under *Workspace location* you can see the directory of the current workspace.

7.2.5. Placing an EB tresos Studio installation under version control

It is possible to place a complete EB tresos Studio installation under version control. You can submit the complete EB tresos Studio installation directory to your version control system, except the following files and directories:

- ▶ `workspace/`,
- ▶ `configuration/.settings/`, and
- ▶ `configuration/org.eclipse.*.`

Some version control systems mark files as read-only when you check them out. In this case you need to make all files from the EB tresos Studio installation directory writable by the current user.

7.2.6. Adding new features and extensions to EB tresos Studio via external folders

If you want to extend EB tresos Studio with new features or with an application that is compatible with the Eclipse version used, you can add the plug-ins of these features or extensions to external folders. After you set a link to these folders from the EB tresos Studio installation, the plug-ins of the features and extensions are loaded into EB tresos Studio when you start EB tresos Studio. To remove the feature or application and have a clean EB tresos Studio installation again, you can simply remove this link.

Eclipse requires a specific directory structure for additional plug-in folders. The directory structure is similar to a normal Eclipse installation: plug-ins are stored in an `eclipse\plugins` folder, which may be anywhere on your PC. Links are stored in a `links` folder in your EB tresos Studio installation, which is referred to as `$TRESOS_BASE`.

To add an extension or new feature to your EB tresos Studio installation:

- ▶ Name the folder in which you have stored your new feature plug-ins `eclipse\plugins`, e.g. `c:\development\extensions\eclipse\plugins`.
- ▶ To link to this folder with your new feature plug-ins from EB tresos Studio, create a folder called `links` in your `$TRESOS_BASE` directory: `$TRESOS_BASE\links`.
- ▶ In the `$TRESOS_BASE\links` folder, create a text file with the suffix `.link`, e.g. `additional.link`. The name of the file extension is important, but you can freely choose the name of the file itself.
- ▶ Open your `.link` file in a text editor and add the path to the folder with the new feature plug-ins without the `eclipse/plugins` folders in this path. E.g. write `path=c:/development/extensions` into your `.link` file.

NOTE**Write the path to your external folder with normal slashes**

You must write the path to your external folder in the `.link` file with normal slashes, i.e. with /, not with backslashes (\).

- ▶ Save your `.link` file.
- ▶ To apply your changes, restart EB tresos Studio.
- ▶ To check whether your plug-ins are loaded correctly, select *EB tresos Details* from the **Help** menu. If your new feature appears in the *EB tresos Details* dialog, the link and all paths are correct.
If your new feature does not appear in the *EB tresos Details* dialog, check and correct all paths of the files and folders.

7.3. Working with projects

This section describes how to work with projects in EB tresos Studio. A project contains all the configurations for one ECU. Before being able to edit module configurations, or to generate code, you need to create a project first and add modules to the project. The project also stores meta-information about the ECU like e.g. the ECU ID or the target and derivate setting.

7.3.1. Creating a new project

To create a new configuration project, EB tresos Studio provides the *Project Wizard*. The Project Wizard guides you through the creation of a new project via several dialog pages. You proceed from one page to the next by pressing the **Next** button. You can always return to a previous page and change your settings by pressing the **Back** button. Only when you press the **Finish** button your settings are executed and a new project is created.

NOTE

Most of the wizard's buttons stay disabled until you have entered all required data.



7.3.1.1. Step 1: Starting the Project Wizard

To start the Project Wizard, there are two alternatives:

- ▶ Alternative 1: Right-click within the Project Explorer (see [Section 6.3.1, “Project Explorer view”](#)).
A context menu opens up.

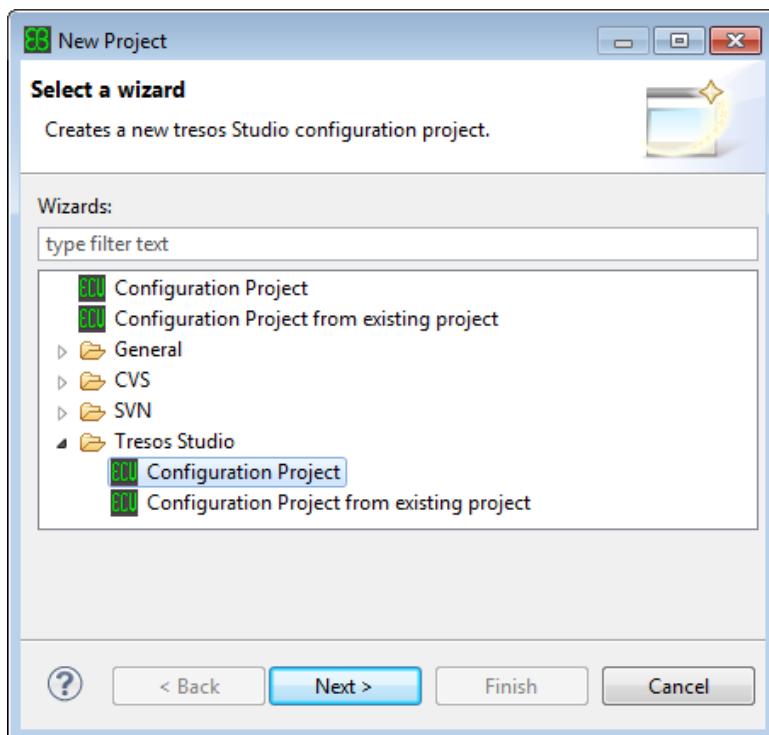
- ▶ Select **New**.
A context menu opens up.
- ▶ Select **Project**.
The Project Wizard *Select a wizard* window opens up.
- ▶ Select *Configuration Project* from the list.

NOTE

The *Configuration Project from existing project* option is used to upgrade an existing project to a newer AUTOSAR version. You can find details about this in [Section 7.7.1, “Converting a project to a new AUTOSAR version”](#).

- ▶ Select **Next**.

The *New Configuration Project* window opens up.



Alternative 2: to start the Project Wizard,

- ▶ Select the **File** menu.
A context menu opens up.
- ▶ Select **New**.
A context menu opens up.

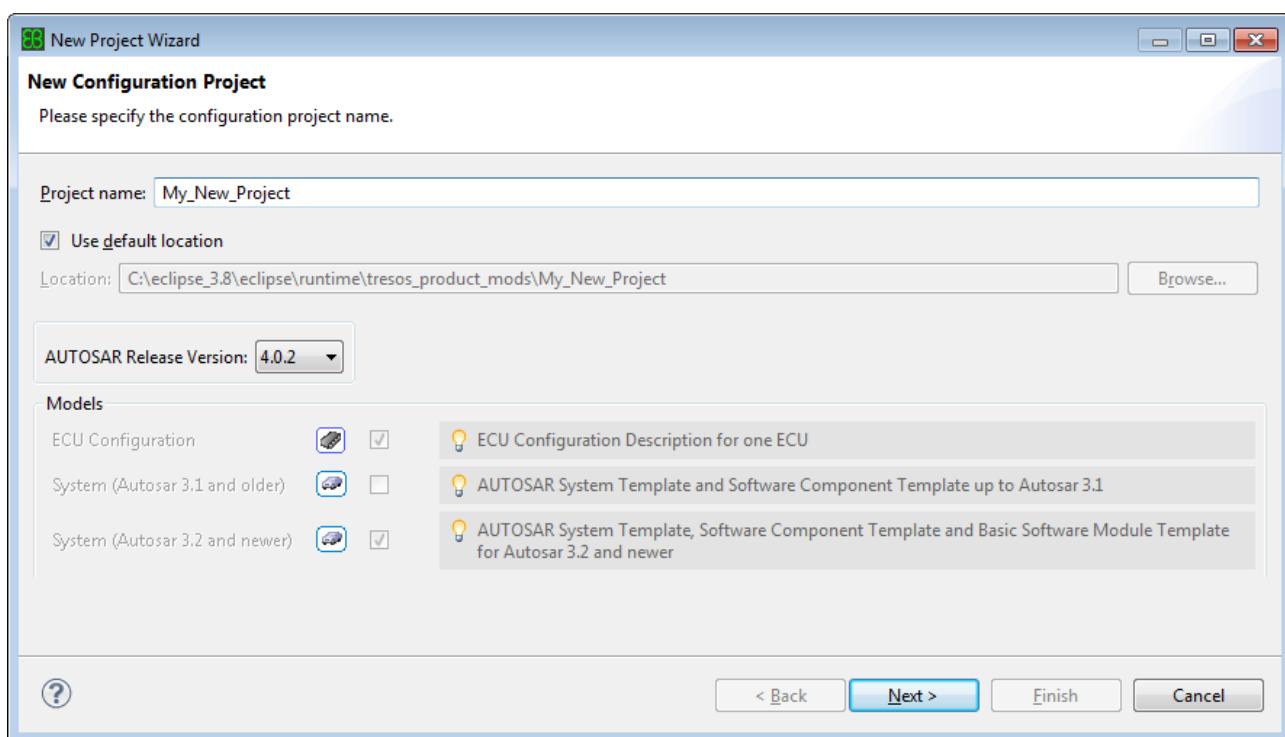
- ▶ Select *Configuration Project* from the list.

NOTE

The *Configuration Project from existing project* option is used to upgrade an existing project to a newer AUTOSAR version. Details about this can be found in [Section 7.7.1, "Converting a project to a new AUTOSAR version"](#).

- ▶ The *New Configuration Project* window opens up.

7.3.1.2. Step 2: Creating a new project



- ▶ Enter a project name in the *Project name* text control.

NOTE

The project name must be a valid AUTOSAR shortname.



- ▶ To store the project in the workspace folder, check the box **Use default location**.
- ▶ To store the project in another location, uncheck the checkbox and select a path.
- ▶ Fill in the *AUTOSAR Release Version*. Use the drop-down list box to select the version of your project.

For example, if you select *4.0.2*, you can add only modules with AUTOSAR release version 4.0.2 to your project. If you select *Mixed*, you may add all available modules for the selected target/derivative and configure module configurations with different release versions.

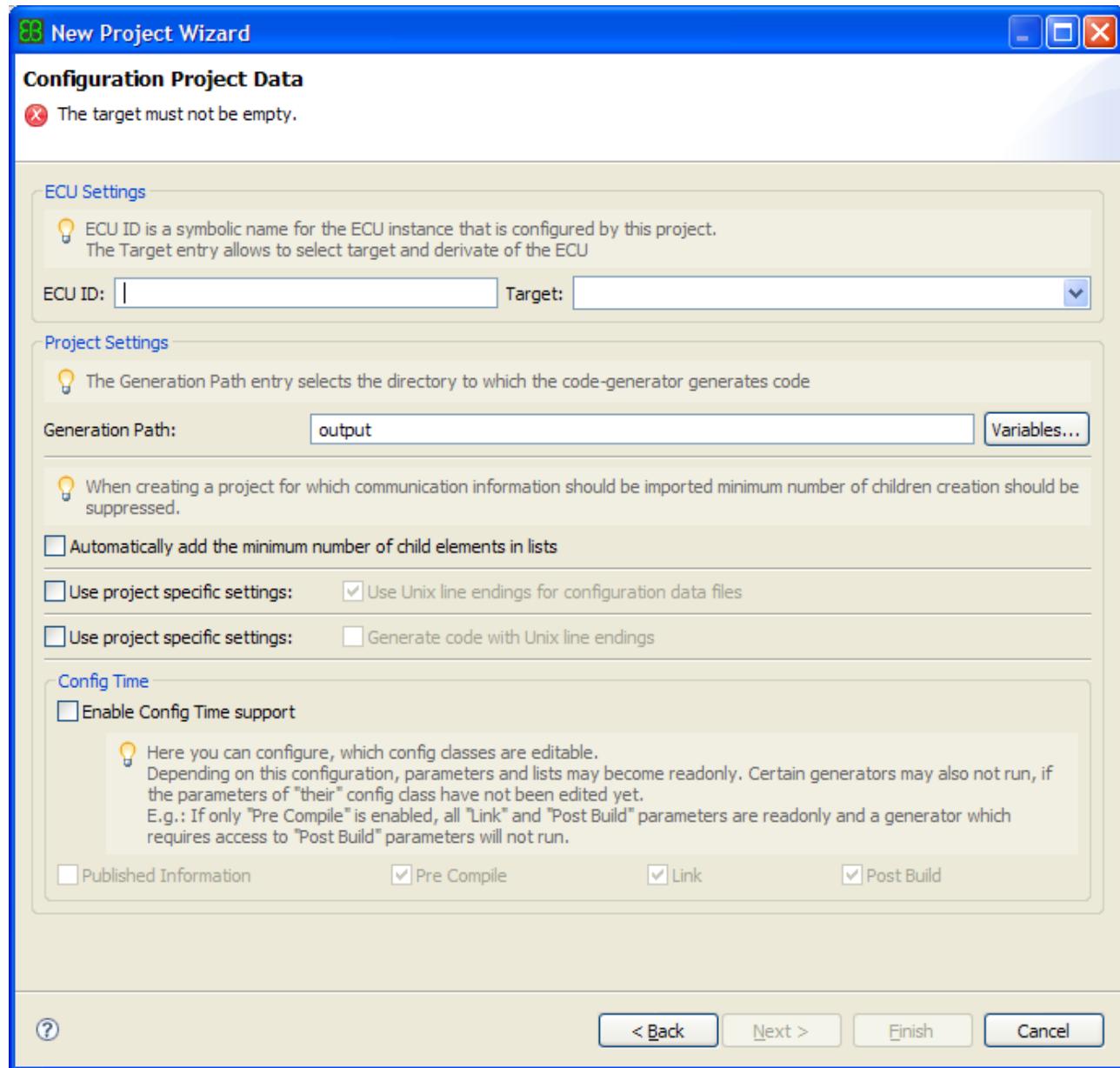
NOTE**This choice affects the Exporter**

This choice affects the Exporter and restricts the possible target content types (see [Section 7.10.2.2, “Importing and exporting configuration data from and to the AUTOSAR format”](#)).

-
- Select **Next**.

The *Configuration Project Data* window opens up.

7.3.1.3. Step 3: Configuring project properties



In the Configuration Project Data window:

- ▶ Fill in the *ECU ID*.

Most likely this ID is provided by the OEM as part of a system description that describes a vehicle as a whole.

- ▶ Fill in the *Target*. Use the drop down listbox to select the target/derivate of your ECU.
- ▶ Select the **Project Settings** according to your preferences.

- ▶ To automatically add the minimum number of child elements in lists select the checkbox **Automatically add....**
- ▶ To use project-specific settings, check the checkbox **Use project-specific settings**.
- ▶ *Line endings of configuration data files:* By default, the type of line endings which is used for the files which store the configuration data is controlled by a global workspace setting (see [Section 7.2.3.5, “Setting configuration preferences”](#)). If you check the checkbox **Use project-specific settings**, the local setting takes effect for the project which you are just creating. You can then check or uncheck the checkbox **User Unix line endings** The value you choose will only be active for this project.

NOTE

You can change this setting later after the project is created via the project preferences (see [Section 7.3.7.2, “Editing the Code Generator settings”](#)).

- ▶ *Line endings of generated code:* By default, the type of line endings of the files generated by the code generator (see [Section 7.9, “Generating code for projects”](#)) is controlled by a global workspace setting (see [Section 7.2.3.5, “Setting configuration preferences”](#)). If you check the checkbox **Use project-specific settings**, the local setting takes effect for the project which you are just creating. You can then check or uncheck the checkbox **Generate code with Unix line endings** and this setting will only be active for this project.

NOTE

You can change this setting later after the project is created via the project properties (see [Section 7.3.7.2, “Editing the Code Generator settings”](#)).

- ▶ Select **Enable Config Time Support** to allow only the configuration of parameters and references of specific configuration classes. Parameters and references that refer to one of the selected configuration class can then be edited by the user. By default this setting is deactivated and all parameters are editable (see [Section 4.5.2.1, “Configuration time of projects”](#))

- ▶ *Published Information:* This check box is never enabled by default, because parameters of the configuration class *Published Information* are fixed even before the pre-compile stage. These parameters can be seen as constants.
- ▶ *PreCompile:* If you select this check box, parameters of the configuration class *PreCompile* are editable.
- ▶ *Link:* If you select this check box, parameters of the configuration class *Link* are editable.
- ▶ *Post Build:* Makes parameters of the configuration class *Post Build* editable.

NOTE**Dependency of configuration classes to code generators**

If you limit the configuration to certain configuration classes, this might affect the code generators. If a code generator requires access to parameters of a not selected or not editable configuration class, this code generator will not run. For detailed information on code generators, refer to [Section 7.9.2, “Generating a project”](#).

TIP**You may change these settings later on as well**

You may also change these settings later on in the project properties, after the project has been created. For instructions on setting the project properties, refer to [Section 7.3.7.3, “Editing the ECU Configuration settings”](#).

- Select **Next**.

The *Module Configurations* page opens up.

7.3.1.4. Step 4: Defining module configurations

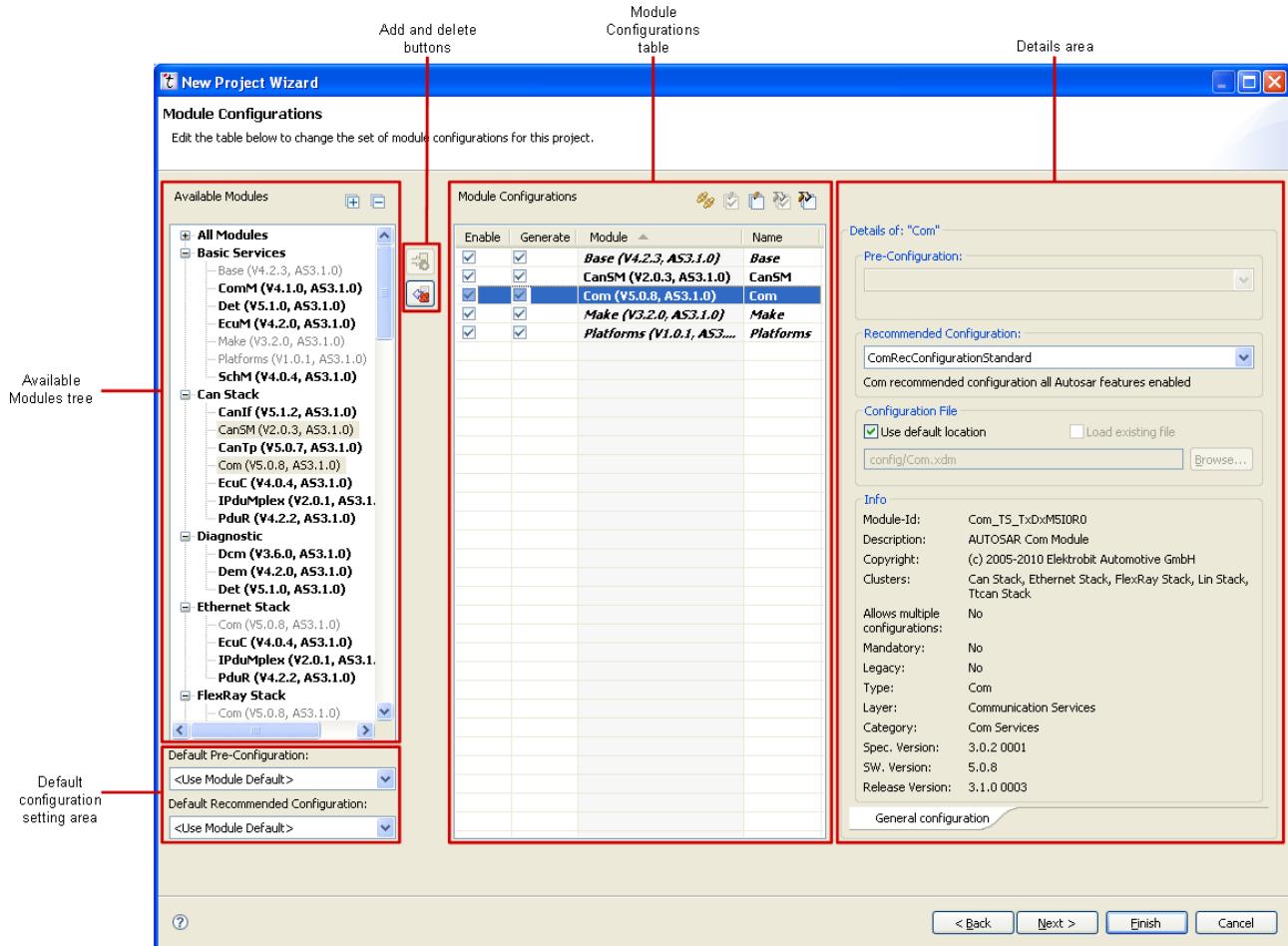


Figure 7.3. The *Module Configurations* page

On the *Module Configurations* page you may add or remove the module configurations to/from the project you want to create and you may edit the configuration settings of these module configurations. Some modules are mandatory for your architecture or AUTOSAR release version. These modules are added automatically and are already displayed in the *Module Configurations* table when you open the *Module Configurations* page. It is possible to remove them, but it is not recommended.

TIP



You may add module configurations later to your project

You may also create a project which initially contains no module configurations and add them later. For instructions for adding or editing module configurations to/in your project, refer to [Section 7.6, “Editing module configurations”](#).

When you are done setting up your initial module configurations, click the **Next** button and continue reading at [Section 7.3.1.5, “Step 5: Importing data into the project \(optional\)”](#).

7.3.1.4.1. Adding module configurations to your project

You may add single module configurations, several module configurations at once or clusters of module configurations to your project.

NOTE**You can recognize whether a module has been added by its font**

Modules that are still available and modules that already have been added to your project are displayed in different font styles in the *Available Modules* tree:

- ▶ Modules that have not been added to your project and thus are still available are displayed in bold font.
 - ▶ Modules that already have been added to your project are displayed in standard font. The modules that allow multiple configurations are displayed in black. All others are grayed-out and cannot be added to your project anymore.
-

To add new module configurations to your project:

- ▶ In the *Available Modules* tree, select all modules/clusters of modules for which you want to add configurations for.
 - ▶ Click the **Add** button .
-

TIP**Add single modules with a double-click**

Module configurations for single modules can also be added by double-clicking the required module.

-
- ▶ If the required module configurations cannot be added to your project, the following dialog opens up:

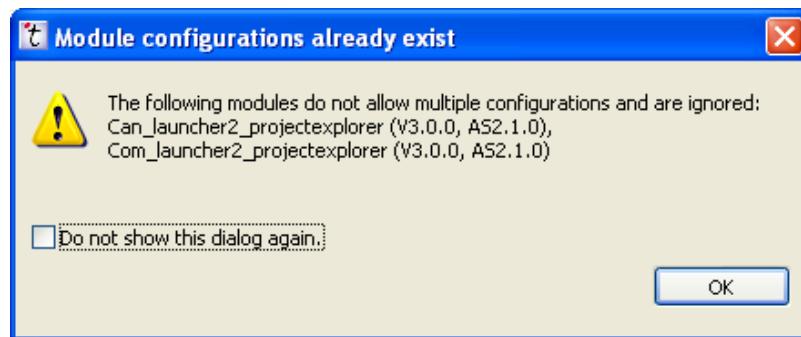


Figure 7.4. Some modules may be added only once to your project

The selected modules are added to your project as module configurations with default settings.

NOTE**Only one module configuration is added per selected module**

Since modules can be part of more than one cluster, it is possible that a selection in the *Available Modules* tree refers several times to the same module. Note that only one configuration is added in this case, even if the module allows multiple configurations.

As long as the recently added module configurations have not been stored into your project yet, you may edit the (default) settings of the module configuration. Mind that depending on the module, there might be more tab pages next to the *General configuration* tab on the bottom of the *Details* area.

7.3.1.4.2. Removing module configurations from your project

You may remove single module configurations or several module configurations at once from your project.

NOTE**You can recognize the state of a module by its font**

Module configurations that previously had been added to your project, mandatory module configurations and recently added module configurations are displayed in different fonts in the *Modules Configuration* table:

- ▶ Mandatory module configurations are displayed in italic font. It is not recommended to remove these module configurations from your project.
 - ▶ Recently added module configurations are displayed in bold font. You may edit them.
 - ▶ Module configurations that had been added to your project before are displayed in standard font.
-

To remove module configurations from your project:

- ▶ In the *Module Configurations* table, select the module configuration(s) you want to remove.

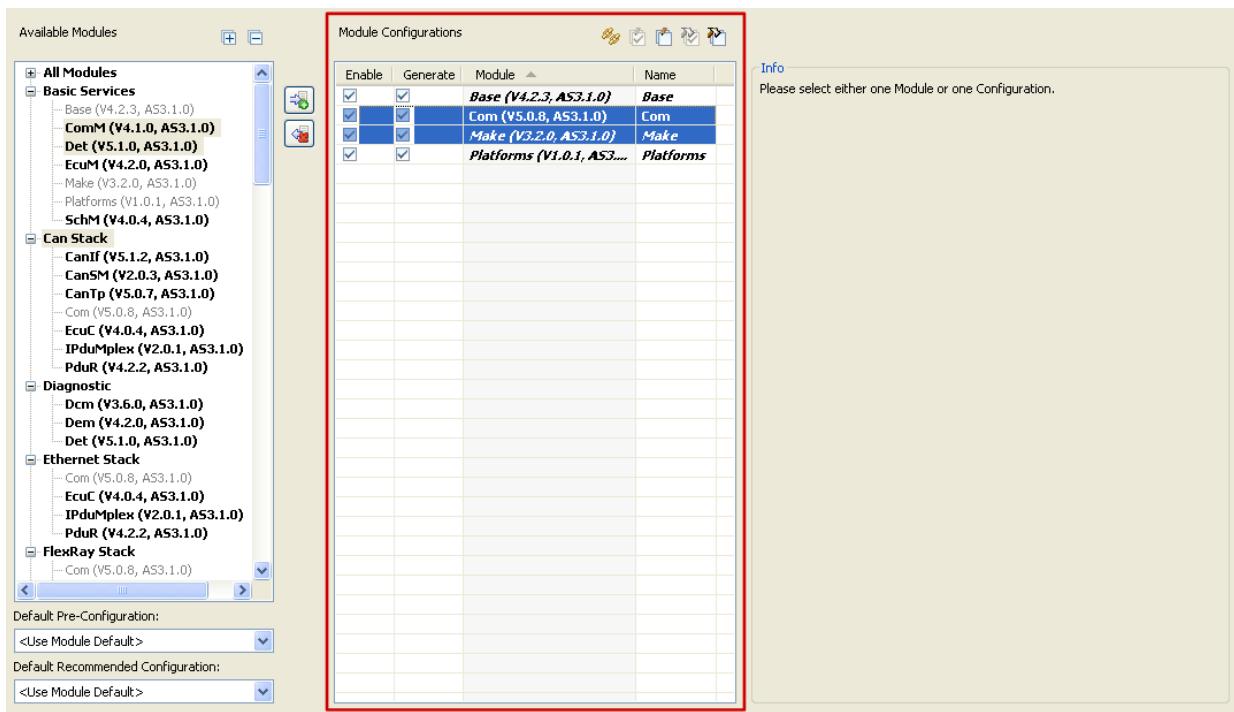


Figure 7.5. Removing module configurations

- ▶ Click the **Delete** button .
- ▶ If none of the selected module configurations refers to an already existing configuration file, the items selected are removed from the *Module Configurations* table without further notice. Otherwise the *Delete Module Configurations* dialog opens up

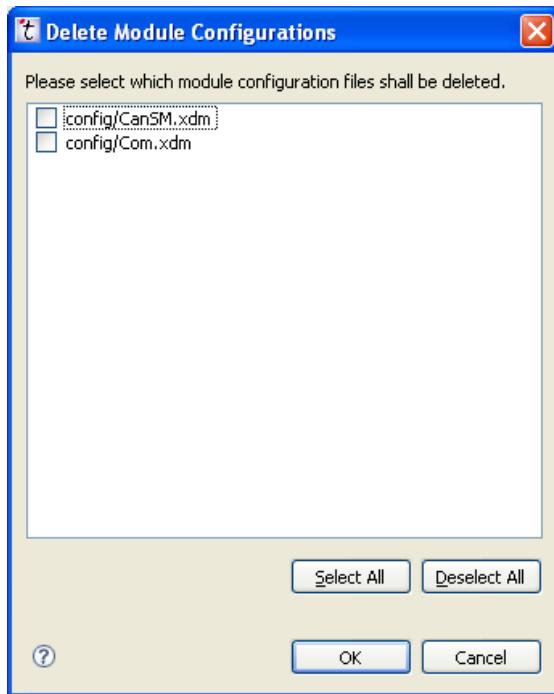


Figure 7.6. Confirming which module configuration files you want to remove

- ▶ Select the check box of the configuration files you want to delete.

NOTE**Data loss**

The selected files will be deleted from the file system when the changed module configurations are applied. After that, you cannot restore these files anymore.

- ▶ To confirm the deletion of the configuration files, click **OK**.

The items selected are removed from the *Module Configurations* table.

- ▶ To cancel the deletion of the *Module Configurations* and their referred configuration files, click **Cancel**.

7.3.1.4.3. Enabling and disabling module configurations

In the *Module Configurations* table you may (temporarily) enable or disable the module configurations you have added to your project. Disabled configurations are treated as if they do not exist, which means you cannot edit or verify them, and code generators cannot access them. To enable module configurations:

- ▶ To enable only one module configuration, select the **Enable** check box of the module configuration you want to enable in the *Module Configurations* table.

Module Configurations				
Enable	Generate	Module	Name	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Base (V4.2.3, AS3.1.0)	Base	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CanSM (V2.0.2, AS3.1.0)	CanSM	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Com (V5.0.8, AS3.1.0)	Com	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Make (V3.2.0, AS3.1.0)	Make	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Platforms (V1.0.1, AS3....)	Platforms	

Figure 7.7. Enabling one module configuration

- ▶ To enable several module configuration at once:
 - ▶ In the *Module Configurations* table, select the module configurations you want to enable.
 - ▶ In the tool bar of the *Module Configurations* table, click the **Enable** button

To disable module configurations:

- ▶ To disable only one module configuration in the *Module Configurations* dialog/page, clear the **Enable** check box of the module configuration you want to disable in the *Module Configurations* table.

Module Configurations				
Enable	Generate	Module	Name	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Base (V4.2.3, AS3.1.0)	Base	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CanSM (V2.0.2, AS3.1.0)	CanSM	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Com (V5.0.8, AS3.1.0)	Com	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Make (V3.2.0, AS3.1.0)	Make	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Platforms (V1.0.1, AS3....)	Platforms	

Figure 7.8. Disabling one module configuration

To disable several module configuration at once:

- ▶ In the *Module Configurations* table, select the module configurations you want to disable.
- ▶ In the tool bar of the *Module Configurations* page, click the **Disable** button

7.3.1.4.4. Enabling and disabling code generation for module configurations

In the *Module Configurations* table you may enable or disable code generation for the module configurations of your project.

WARNING Only partly generated code leads to incorrectly working ECUs


If you only generate the code of a part of the module configurations you later want to implement on an ECU, depending on which modules you choose, the code might not work correctly. This might lead to incorrectly working ECUs.

To avoid incorrectly working ECUs, generate the code of all module configurations in your project before deploying them on an ECU.

To enable code generation for module configurations:

- ▶ To enable code generation for only one module configuration, select the **Generate** check box of the module configuration for which you want to enable code generation in the *Module Configurations* table.

Module Configurations				
Enable	Generate	Module	Name	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Base (V4.2.3, AS3.1.0)	Base	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	CanSM (V2.0.2, AS3.1.0)	CanSM	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Com (V5.0.8, AS3.1.0)	Com	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Make (V3.2.0, AS3.1.0)	Make	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Platforms (V1.0.1, AS3....	Platforms	

Figure 7.9. Enabling code generation for one module configuration

- ▶ To enable code generation for several module configurations at once:
 - ▶ In the *Module Configurations* table, select the module configurations, for which you want to enable code generation.
 - ▶ In the tool bar of the *Module Configurations* page, click the **Enable code generation** button

To disable code generation for module configurations:

- ▶ To disable code generation for only one module, clear the **Generate** check box of the module configuration for which you want to disable code generation in the *Module Configurations* table.

Module Configurations				
Enable	Generate	Module	Name	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Base (V4.2.3, AS3.1.0)	Base	
<input type="checkbox"/>	<input type="checkbox"/>	CanSM (V2.0.2, AS3.1.0)	CanSM	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Com (V5.0.8, AS3.1.0)	Com	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Make (V3.2.0, AS3.1.0)	Make	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Platforms (V1.0.1, AS3....	Platforms	

Figure 7.10. Disabling code generation for one module configuration

- ▶ To disable code generation for several module configurations at once:

- ▶ In the *Module Configurations* table, select the module configurations, for which you want to disable code generation.
- ▶ In the tool bar of the *Module Configurations* page, click the **Disable code generation** button

7.3.1.4.5. Naming *Module Configurations*

When you add new module configurations to your project, the name of the new module configuration is set automatically to a name that is unique within the project using either the default configuration name provided by the module or the module name itself.

Instead of using the default name you may want to use another name for the module configuration.

NOTE
Only in recently added module configurations the name can be changed


You can only change the name of a module configuration if you added this module configuration during the same session of the dialog.

To edit the name of a module configuration:

- ▶ Add the required module configuration to your project.
- ▶ In the *Module Configurations* table, click into the *Name* cell of the module configuration whose name you want to change.
- ▶ Type the new name.



Figure 7.11. Editing the name of a new module configuration

NOTE
Name constraints


Mind that the module configuration name must meet the following constraints:

- ▶ The name must be unique within the project.
- ▶ The name must start with a letter.
- ▶ The name must contain only letters and digits plus the underscore character ('_').
- ▶ The name must not be empty.
- ▶ The name must not exceed 255 characters.

- ▶ Press **ENTER**.

The name of the module configuration is changed.

7.3.1.4.6. Choosing *Pre-Configurations*

Module suppliers optionally can equip modules with one or more preconfigurations that contain values for configuration parameters. If a module configuration of your project provides one or more preconfiguration(s), you must assign this module configuration to such a preconfiguration when adding it to your project. The parameters covered by the chosen preconfiguration are then fixed in the module configuration and cannot be edited by the user.

In the *Default configuration setting* area, you can set the *Default Pre-Configuration* which is used as a default value for all new module configurations. To support a more consistent selection, this preconfiguration setting is persisted. You can also edit the *Pre-Configuration* of an individual module configuration.

Setting the *Default Pre-Configuration*

In the *Default configuration setting* area of the *Module Configurations* page, you may edit the *Default Pre-Configuration* for new module configurations.

NOTE**The value set is a union set of all preconfigurations provided by all modules**

The preconfiguration names to choose from 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 preconfiguration is taken instead when you add it to your project.

To change the default preconfiguration, select the required preconfiguration in the **Default Pre-Configuration** drop-down list box.

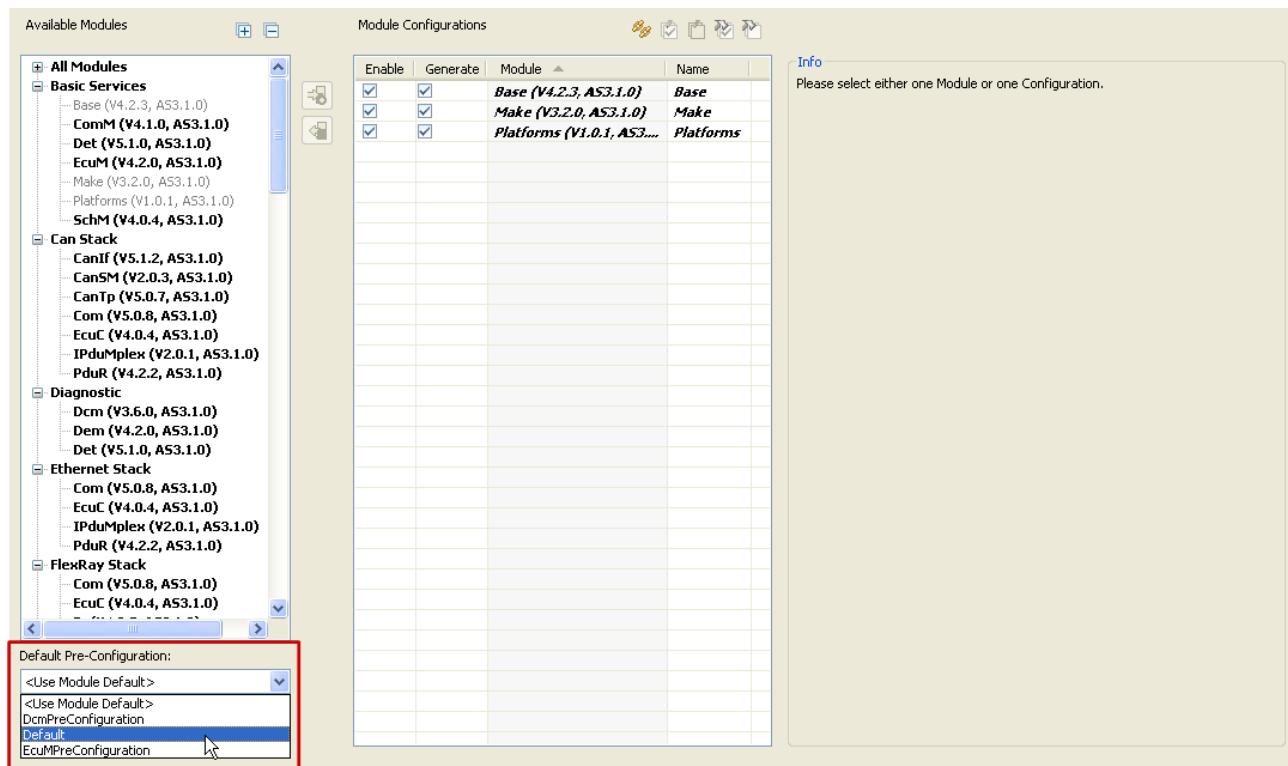


Figure 7.12. Selecting the default preconfiguration for new module configurations

- ▶ To use the respective default preconfiguration for new module configurations, select <Use Module Default>.

NOTE**Non-compilable code**

If you select <Use Module Default>, every module will be added to your project with a different preconfiguration. This might lead to incompatibilities of the module configurations and thus to non-compilable code.

- ▶ The selected default preconfiguration is saved, whenever you actually add new module configurations to your project.

Changing the preconfiguration of one specific module configuration

To change the preconfiguration of one specific module configuration:

- ▶ Add the required module configuration to your project.

NOTE**Only in recently added module configurations the preconfiguration can be changed**

You can only change the preconfiguration of a specific module configuration if you added this module configuration during the same session of the dialog.

- ▶ In the *Module Configurations* table, mark the module configuration whose preconfiguration you want to change.

In the *Details* area of the *Module Configurations* page, detailed information about the specific module is displayed.

The screenshot shows the EB tresos Studio interface with the 'Module Configurations' page open. On the left, there is a tree view of 'Available Modules' under 'All Modules'. The 'Com' module is selected in the main table, which has columns for 'Enable', 'Generate', 'Module', and 'Name'. The 'Name' column shows entries like 'Base (V4.2.3, AS3.1.0)', 'Com (V5.0.8, AS3.1.0)' (which is selected), 'Make (V5.2.0, AS3.1.0)', and 'Platforms (V1.0.1, AS3.1.0)'. To the right of the table is a large 'Details' dialog box for the selected 'Com' module. This dialog contains sections for 'Pre-Configuration' (a dropdown menu), 'Recommended Configuration' (set to 'ComRecConfigurationStandard'), 'Configuration File' (with a 'config/Com.xdm' file selected), and 'Info' (displaying various module details like Module-Id, Description, Copyright, etc.). A red box highlights the 'Details' dialog.

Figure 7.13. The *Details* area in the *Module Configurations* page

- ▶ In the **Pre-Configuration** drop-down list box in the *Details* area, select the required preconfiguration for the selected module configuration.

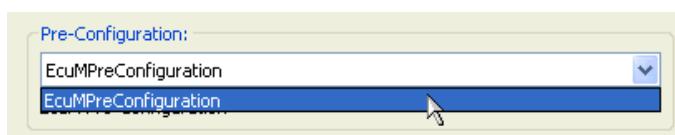


Figure 7.14. Changing the preconfiguration of one specific module

NOTE**Grayed-out preconfigurations are not available**

If the **Pre-Configuration** drop-down list is grayed-out, the selected module does not provide a preconfiguration of its own.

7.3.1.4.7. Applying recommended configurations

Module suppliers optionally can equip modules with one or more recommended configurations that contain values for configuration parameters. If a module configuration of your project provides one or more recommended configurations, you can choose one when you add a configuration for this module to your project. The parameter values covered by the chosen recommended configuration can be considered as a guideline. They may be edited by the user later.

In the *Default configuration setting area*, you can set the *Default Recommended Configuration* which is used as a default value for all new module configurations. To support a more consistent selection, this setting is persisted. You can also edit the *Recommended Configuration* of an individual module configuration.

Setting the *Default Recommended Configuration*

In the *Default configuration setting area* of the *Module Configurations* page, you may edit the *Default Recommended Configuration* for new module configurations.

NOTE**The value set is a union set of all recommended configurations provided by all modules**

The names of the recommended configurations to choose from 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 configuration is taken instead when you add it to your project. If the module does not provide a default configuration, no recommended configuration is applied.

To change the default recommended configuration, select the required recommended configuration in the **Default Recommended Configuration** drop-down list box.

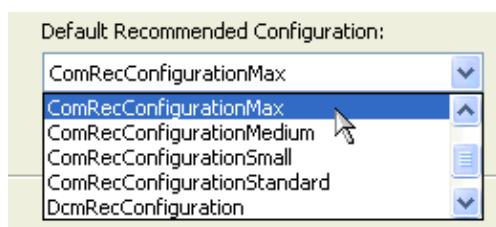


Figure 7.15. Selecting the default recommended configuration for new module configurations

- ▶ If you don't want to apply any of the recommended configurations, select <None>.
- ▶ To use the respective default recommended configuration for new module configurations, select <Use Module Default>.

NOTE**Non-compilable code**

If you select <Use Module Default>, every module will be added to your project with a different recommended configuration. This might lead to incompatibilities of the module configurations and thus to non-compilable code.

- ▶ The selected default recommended configuration is saved, when you actually add new module configurations to your project.

Changing the recommended configuration of one specific module configuration

To change the recommended configuration of one specific module configuration:

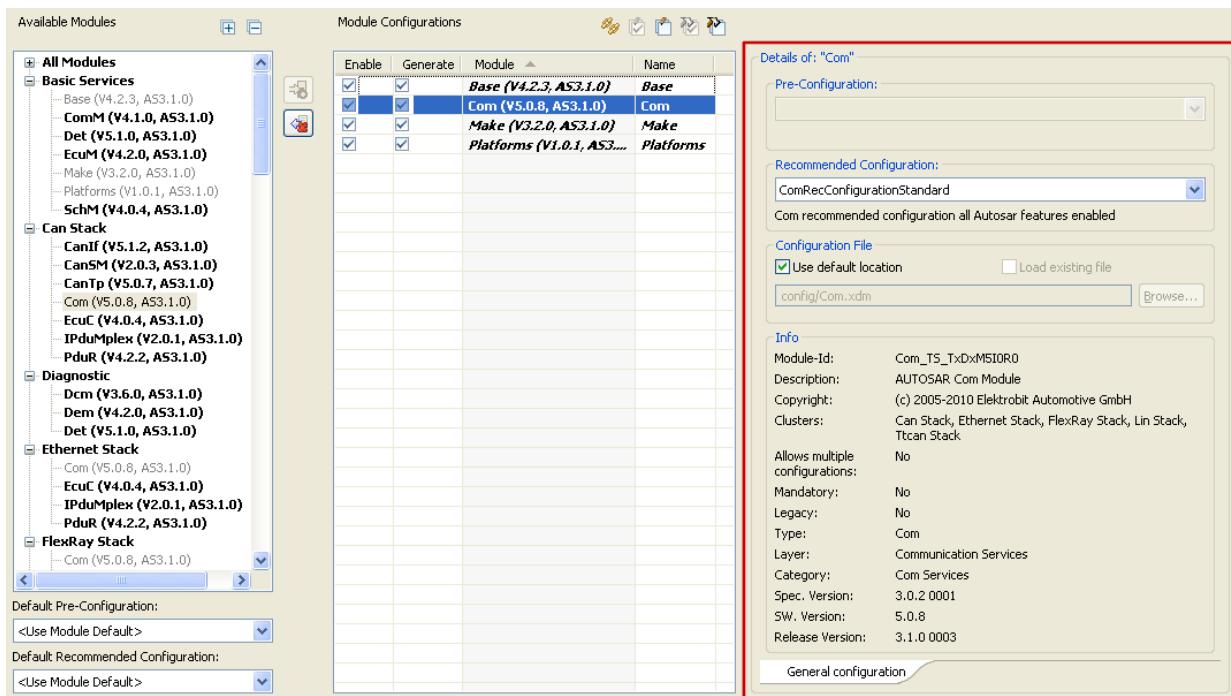
- ▶ Add the required module configuration to your project.

NOTE**Only in recently added module configurations the recommended configuration can be changed**

You can only change the recommended configuration of a specific module configuration if you have added this module configuration during the same session of the dialog.

- ▶ Mark the module configuration whose recommended configuration you want to change.

In the *Details* area of the *Module Configurations* page, detailed information about the specific module is displayed.

Figure 7.16. The *Details* area in the *Module Configurations* page

- ▶ In the **Recommended Configuration** drop-down list box, select the required recommended configuration for the selected module configuration.

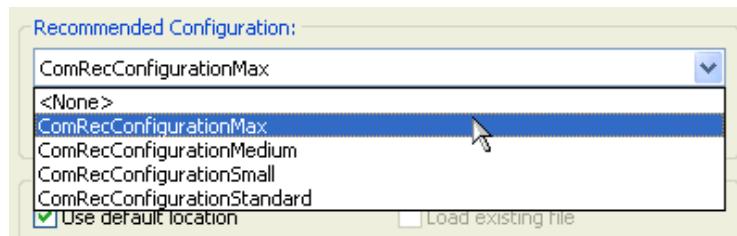


Figure 7.17. Changing the recommended configuration of one specific module

NOTE**Grayed-out recommended configurations are not available**

If the **Recommended Configuration** drop-down list is grayed-out, the selected module does not provide a recommended configuration of its own.

7.3.1.4.8. Editing the configuration file settings

The configuration file stores all parameters for a specific module configuration. For detailed information on how to edit those parameters, e.g. by using the Generic Configuration Editor, refer to [Section 7.8, “Editing parameters of a module configuration”](#).

When you add new module configurations to your project, per default the configuration file name is set to config/<moduleconfigurationname>.xdm, relative to your project folder. This is also referred to as the default location.

Instead of using the default location you can use another location or name for the configuration file. To use another location or file name:

- ▶ Add the required module configuration to your project.

NOTE**Only in recently added module configurations the configuration file can be changed**

You can only change the configuration file of a module configuration, if you have added this module configuration during the same session of the dialog.

- ▶ In the *Module Configurations* table, mark the module configuration whose configuration file you want to change.
- ▶ In the *Configuration File* panel of the *Details* area, clear the check box **Use default location**.

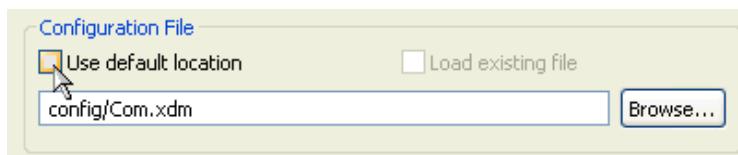


Figure 7.18. Editing the configuration file settings

- ▶ Enter or choose the new file name. Either use absolute path names or path names relative to your project.

NOTE**The file extension .xdm is mandatory**

You can only use file names ending with the extension .xdm. When you choose an existing file, you must make sure that it matches the targeted module configuration schema. Otherwise the new module configuration will fail to load when applied.

Using existing configuration files

Instead of configuring the module configuration from scratch you may also use an already existing configuration file.

NOTE**You cannot set preconfigurations and recommended configurations when you load an existing configuration file**

If you use an existing configuration file, you cannot apply any preconfiguration or recommended configuration.

To load an already existing configuration file into your project:

- ▶ In the *Configuration File* panel, browse to the file you want to load.

NOTE**The selected configuration file will be modified**

The selected configuration file will be modified while you edit the parameters of the module configuration. If you want to preserve the file as it is, copy the file to your project before loading it to your project.



Figure 7.19. Using an existing configuration file

Whenever you point the *Configuration File* panel to an existing file in the file system, the *Load existing file* check box is marked.

NOTE**Data loss**

If you clear the *Load existing file* check box, the existing file will be overwritten and cannot be restored again.

7.3.1.4.9. Upgrading module configurations

To upgrade module configurations from older software versions of the modules to newer ones:

- ▶ Click the **Upgrade** button  in the tool bar of the *Module Configurations* table.

The *Upgrade* dialog opens up.

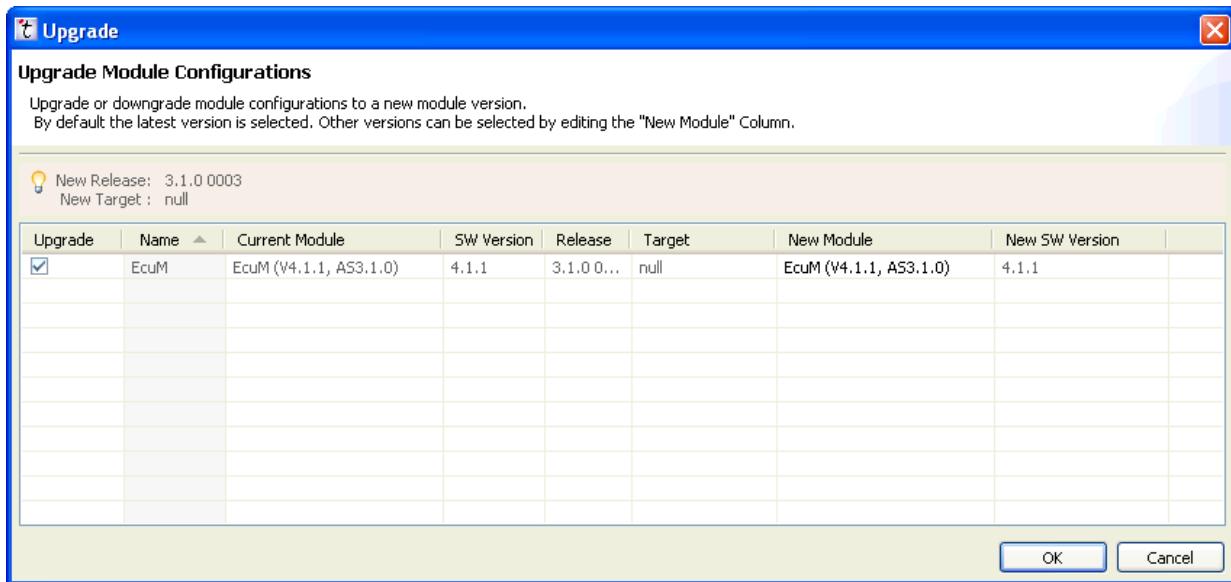
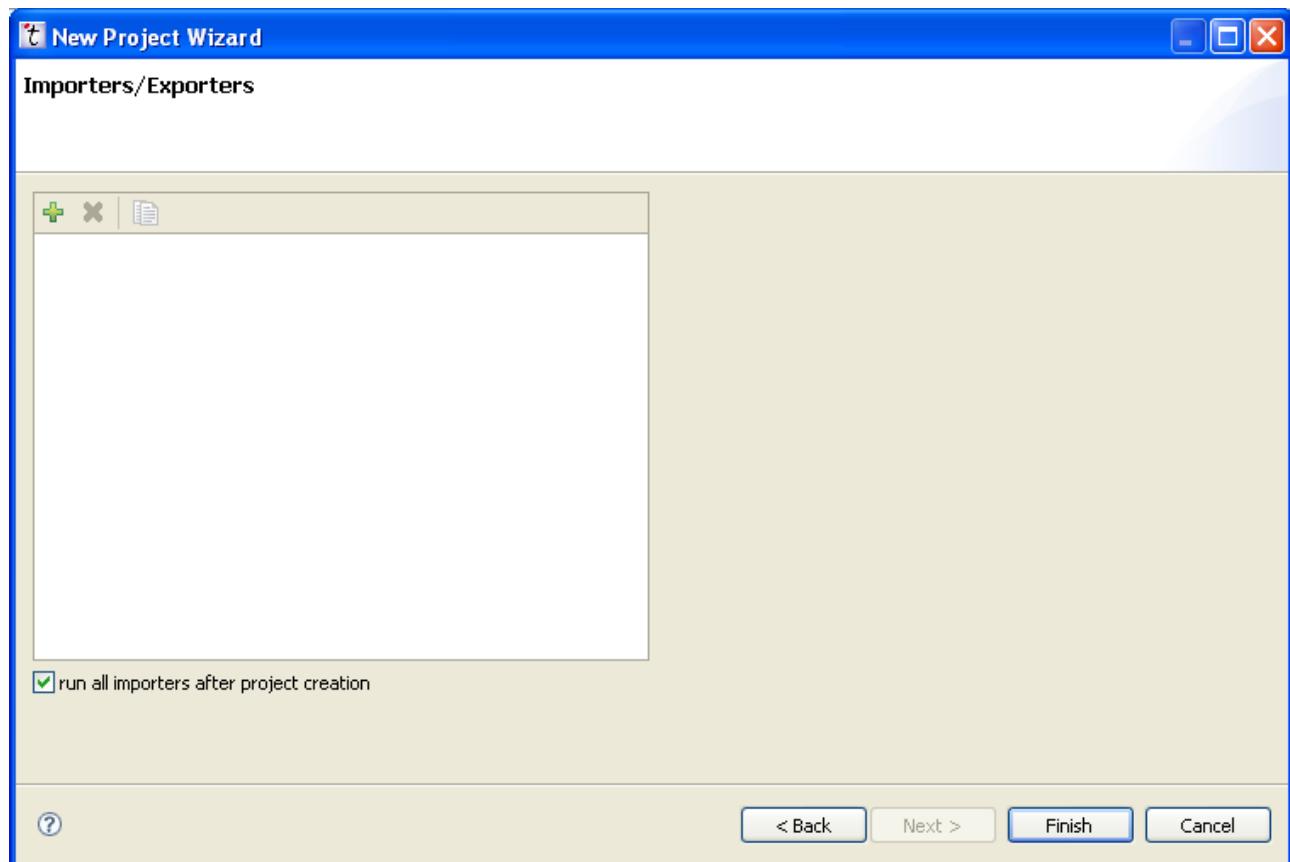


Figure 7.20. Upgrading module configurations in the *Upgrade* dialog

For detailed information on the *Upgrade* dialog, refer to [Section 7.7, “Upgrading projects and configurations”](#).

7.3.1.5. Step 5: Importing data into the project (optional)

It is possible to import configuration data into the project directly when it is created. This step is optional, you can also import the data later, or not at all. See [Section 7.10.2, “Importing and exporting configuration data”](#) for details on how to do this. If you want to skip this part, continue directly to the next section.



7.3.1.6. Step 6: Finishing set-up of the new project

To finish setting up the new project:

- ▶ Click the **Finish** button. The wizard closes and you see the main window again. If you are creating a project with a lot of modules or if you are importing big files the creation may take a while and a dialog with a progress bar may appear. If the work is completed, the progress bar dialog closes.

The new project is now visible in the *Project Explorer*. It is automatically loaded and you can start editing the module configurations.

7.3.2. Saving a project

If you have changed a project, e.g. by editing one of the module configurations, you need to save the changes to disk. Each module configuration is saved to a separate file. The location of the file has been set when the module was added to the project.

To save a project

- ▶ Select the project node in the *Project Explorer*.
- ▶ Right-click on the node.
A context menu opens up.
- ▶ Select **Save**

7.3.3. Renaming a project

If you want to change the name of a project after it has been created, you can rename it.

To rename a project

- ▶ Select the project node in the *Project Explorer*.
- ▶ Right-click on the node.
A context menu opens up.
- ▶ Select *Rename* and type in a new name.
The project now appears in the project explorer under its new name.

7.3.4. Copying a project

To copy a project:

- ▶ Right-click on the project node you want to copy.
A context menu opens up.
- ▶ Select the *Copy* entry
The project is copied to the clipboard.

To paste the project under a new name:

- ▶ Right-click in the *Project Explorer*.
A context menu opens up.
- ▶ Select the *Paste* entry.

A dialog opens up and prompts you for a name of the new project. Since the project name must be unique you have to decide on a new name. After entering the name press the **OK** button.

The dialog closes and the new project is visible in the *Project Explorer*.

WARNING

If you manually changed any paths of the project to absolute paths, both projects now work on the same directories. This may be the case if you changed e.g. the output directory, or the location of the config files during project creation or in the project properties. If you did not change anything, only directories relative to the project directory are used and both project are completely separated from another.

7.3.5. Deleting a project

It is possible to delete a project. Deleted projects cannot be opened again.

To delete a project

- ▶ Select the project node in the *Project Explorer*.
- ▶ Right-click on the node.
A context menu opens up.
- ▶ Select *Delete*. A dialog opens up asking you if you really want to delete the project. Additionally, there are two radio buttons in the dialog which let you make a choice:
 - ▶ if you decide to *also delete contents* ..., the project will completely be removed from disk. The disk space will be freed, but you will have no way of restoring the project.
 - ▶ if you decide to *do not delete contents*, all associations to the project inside EB tresos Studio are removed, but the project remains on disk (inside the `workspace` directory). If you want to restore the project you can do so by importing it again into EB tresos Studio (see [Section 7.10.1.1, “Importing a project”](#)).

7.3.6. Closing and opening projects

By default the projects are always open in EB tresos Studio. If you are working with several very big projects and if you want to reduce the memory footprint and increase the performance of EB tresos Studio you can close the projects that you currently do not need.

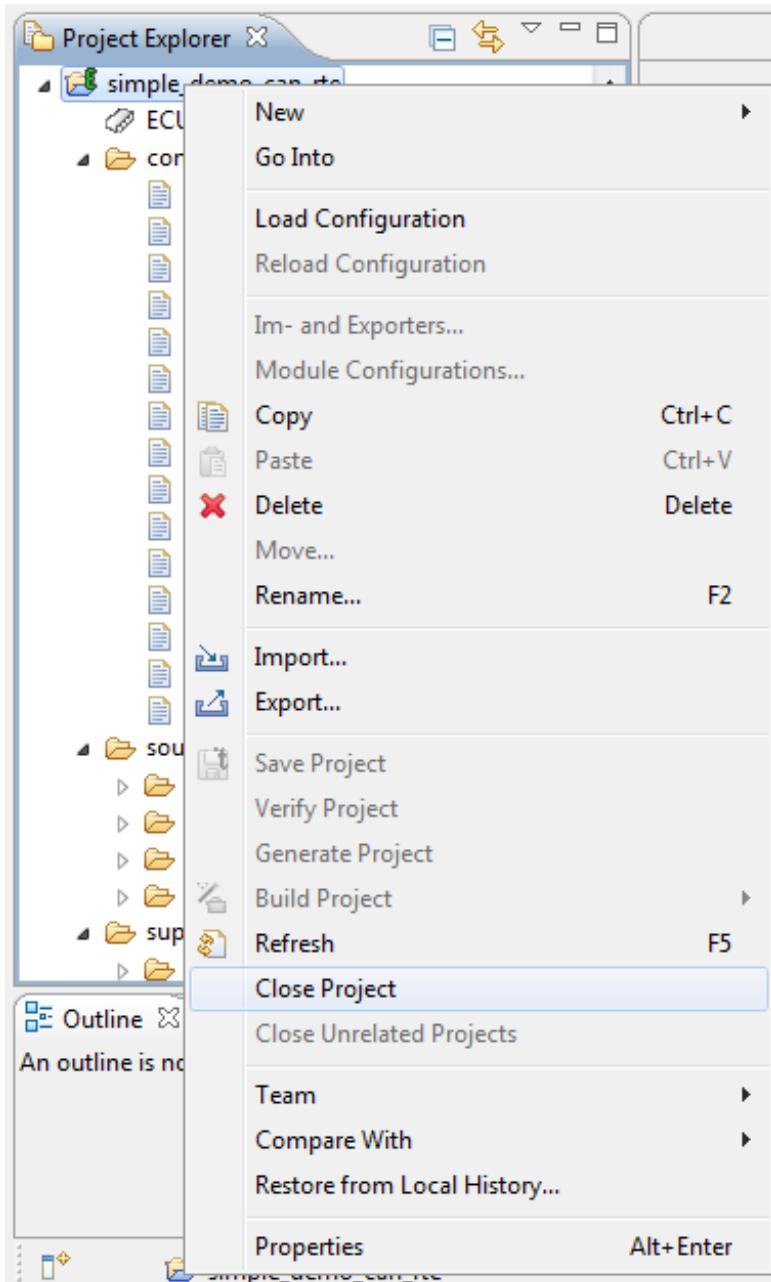
Closed projects only use very few resources, its content is not loaded and you cannot edit their configurations or generate code for them. A closed project needs to be opened and then loaded to be fully functional. More information about the state of a project can be found in [Section 6.3.1.1, “Project and ECU configuration states”](#).

7.3.6.1. Closing projects

To close a project:

- ▶ Select the project node in the *Project Explorer*.
- ▶ Right-click on the project node.

A context menu opens up.



- ▶ Select *Close Project*

The project selected is being closed.

To close all projects that are not referenced by your current project:

- ▶ Select the project node you want to keep open in the Project Explorer.
- ▶ Right-click on the node.

A context menu opens up.

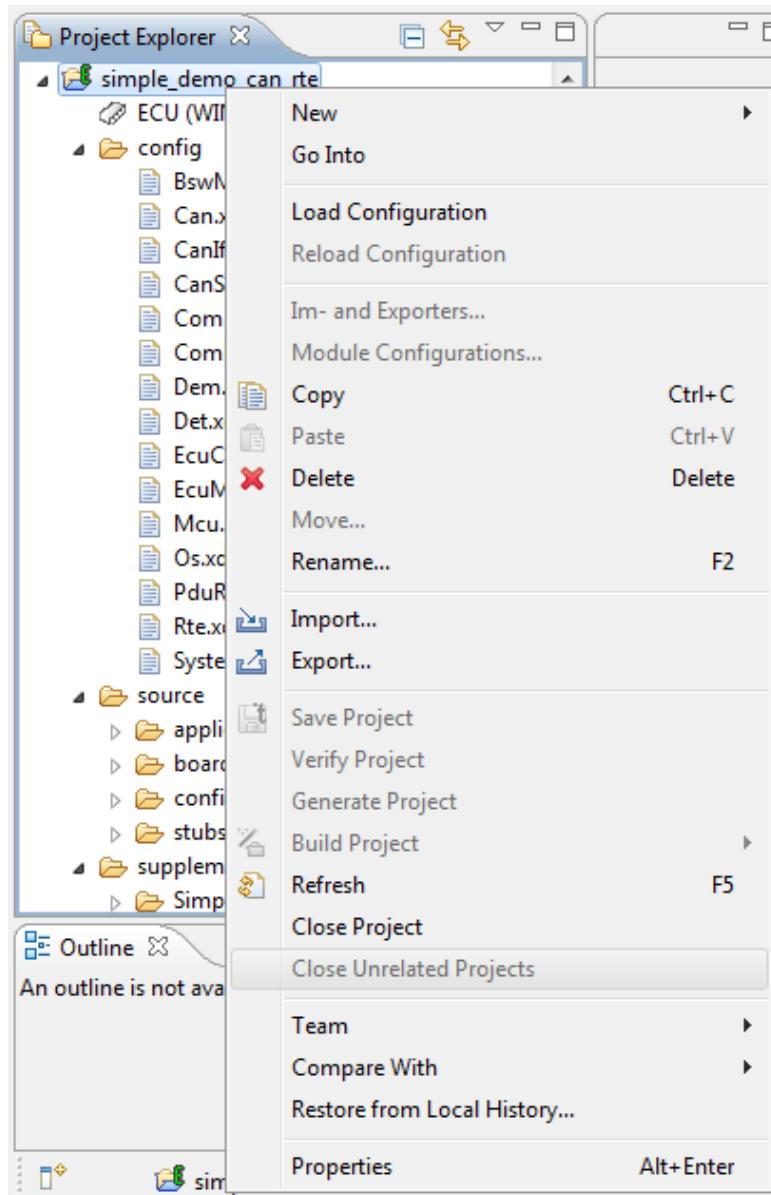


Figure 7.21. Closing all projects that are not referenced by the current project

- ▶ Select *Close Unrelated Projects*.

All projects that are not referenced by your current project are closed.

7.3.6.2. Opening a project

To open a closed project:

- ▶ Right-click on the closed project to open the *Project Explorer*'s context menu.
- ▶ Select **Open Project**.

The project is now open but yet unloaded.

7.3.6.3. Loading a project

To load an open project take one of the following two steps:

- ▶ Alternative 1: Double-click on the ECU configuration node

The project starts loading its configuration.

- ▶ Alternative 2: Right-click on the ECU configuration node to open the Project Explorer's context menu, then select *Load Configuration*

The project starts loading its configuration.

7.3.6.4. Reloading a project

If you modify any XDM file of a loaded project, e.g. by using the team collaboration support feature (see [Section 7.4, "Using team collaboration support"](#)), you must reload the project.

WARNING**Data loss**

It is strongly recommended to modify XDM files only, if the project is *not* loaded yet.

If you do edit XDM files in loaded projects, you must reload the project.

Otherwise all intermediate changes on the XDM files will be lost, when you save the configuration project the next time.

7.3.6.4.1. Reloading projects automatically

EB tresos Studio automatically reloads a loaded project in the following cases:

- ▶ If you modify any XDM file of a loaded project, e.g. by using the team collaboration support (see [Section 7.4, "Using team collaboration support"](#)) or by using the integrated text based file editor of EB tresos Studio.

- ▶ If you modify any XDM file of a loaded project from outside of EB tresos Studio, e.g. by using any text editor, and if you have set the EB tresos Studio preference *General -> Workspace -> Refresh automatically*. For further information on preferences settings, refer to [Section 7.2.3, “Setting preferences”](#).

7.3.6.4.2. Reloading projects manually

In some cases it is useful to manually reload an already loaded project, for instance after you have canceled an automatic reload, because there are unsaved changes in your ECU configuration project.

To manually reload a loaded project:

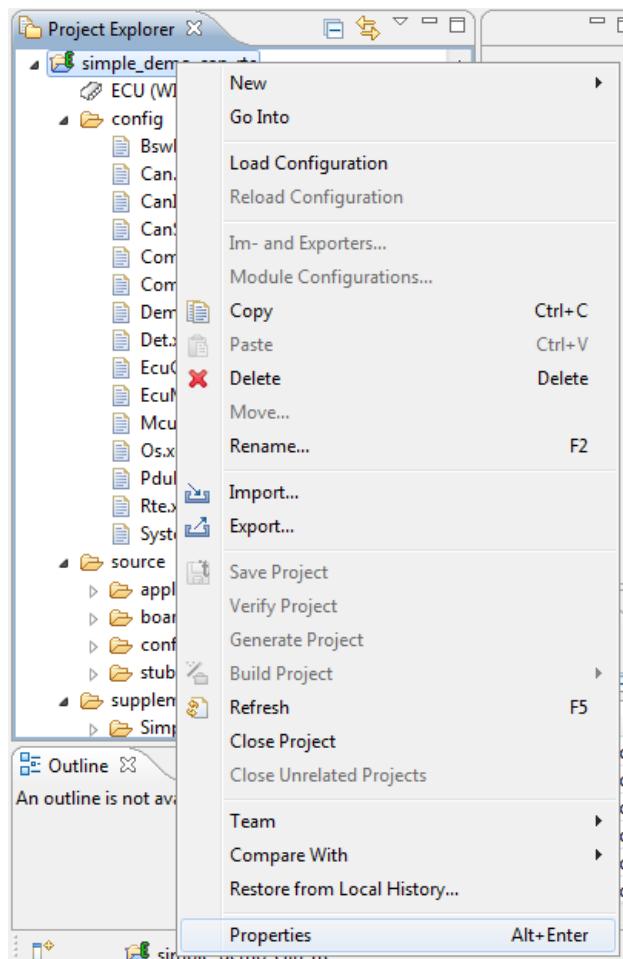
- ▶ Right-click the ECU configuration in the *Project Explorer* view. A context menu opens up.
- ▶ In the context menu, select *Reload Configuration*.

The project starts reloading its configuration.

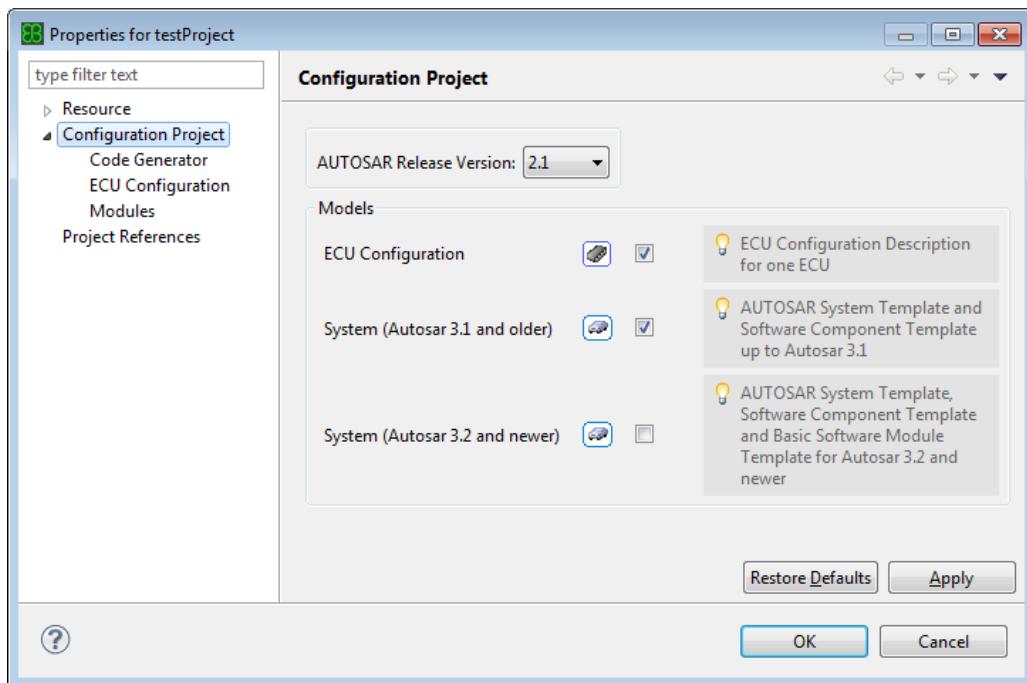
7.3.7. Viewing and changing project properties

The data you have entered in the *New Project* wizard can be viewed and changed. This is done in the *Properties for project name* dialog. To open this dialog:

- ▶ Select the project in the Project Explorer.
- ▶ Right-click on the project.
A context menu opens up.
- ▶ Select *Properties* from the menu.



The *Properties for project name* dialog opens up.

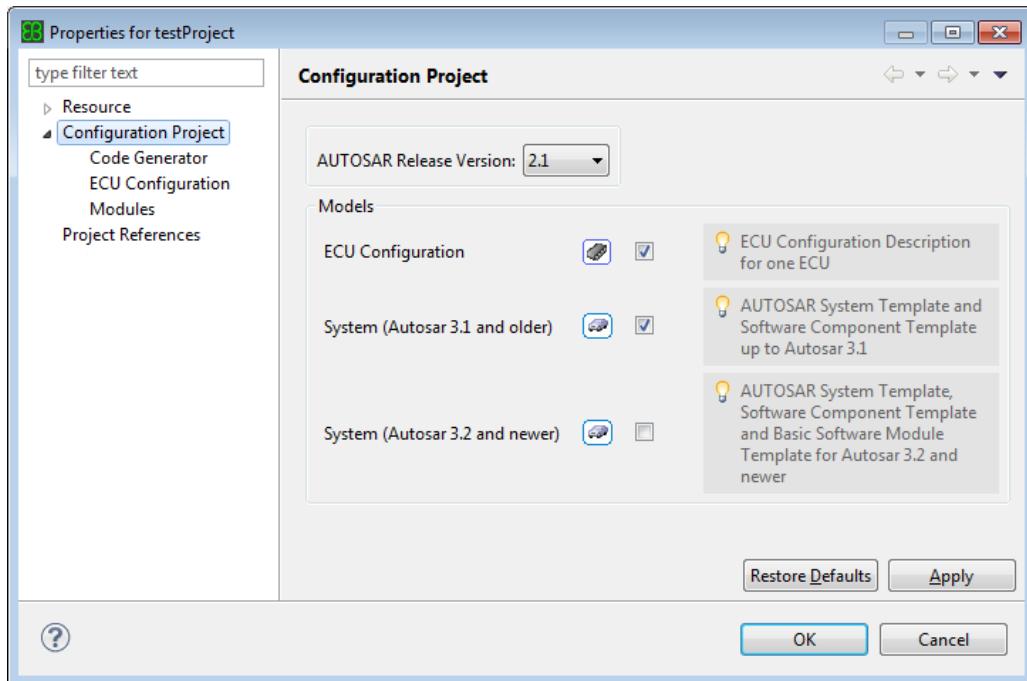


The *Properties for project name* dialog consists of a tree on the left and a main area. The available properties are divided into different categories which make up the items of the tree on the left. Some categories contain subcategories. If you choose a category by selecting an item of the tree, the properties of that category are displayed in the main area of the dialog. The categories which are relevant for the work with EB tresos Studio are grouped below *Configuration Project*. You can expand this tree item to see more categories.

7.3.7.1. Listing the supported models of the project

On the *Configuration Project* page of the project properties you can see which models are enabled in this project. The *ECU Configuration* model is always enabled. The *System* is automatically enabled when an *Rte* module is present in your project. You can also enable the *System* manually without an *Rte* module present, but currently this does not have any effect. In the *Models* section you can choose the *System* matching to the **AUTOSAR Release Version**.

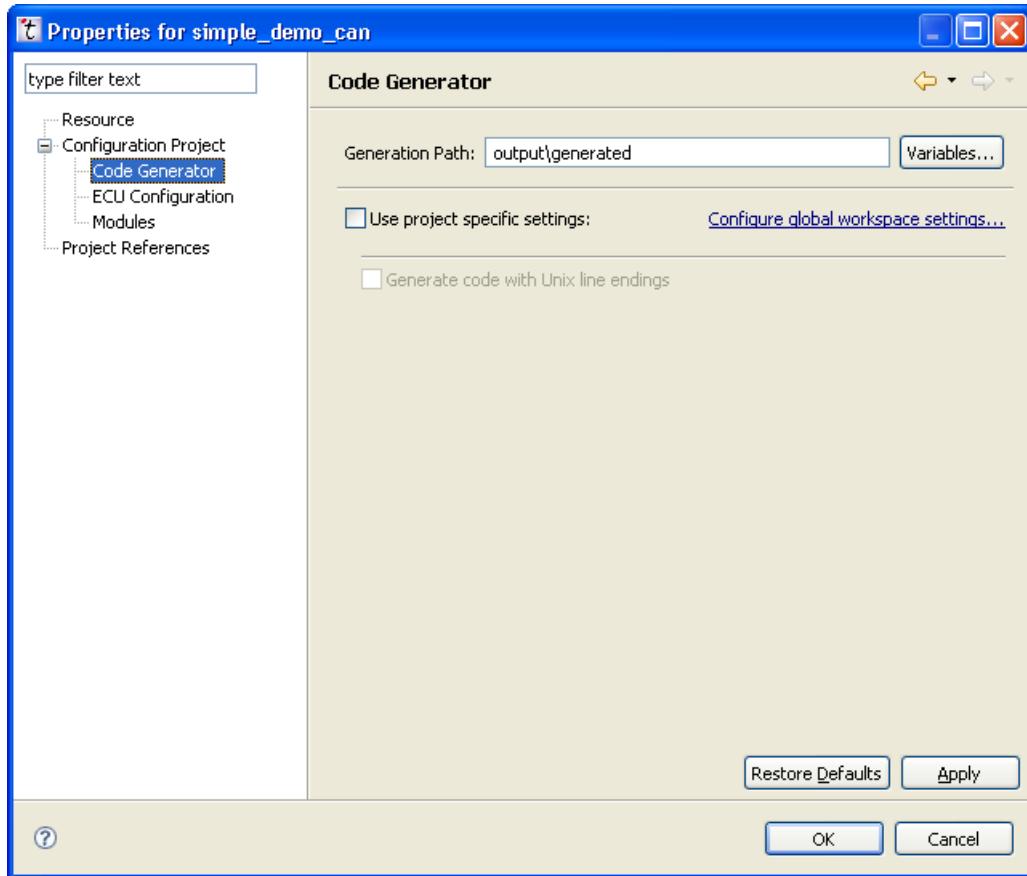
- ▶ In the **AUTOSAR Release Version** drop-down list box you may change the AUTOSAR version of your project.



7.3.7.2. Editing the Code Generator settings

The properties on the *Code Generator* page affect the behavior of code generators. To open this page:

- ▶ Select *Code Generator* from the tree view on the left.

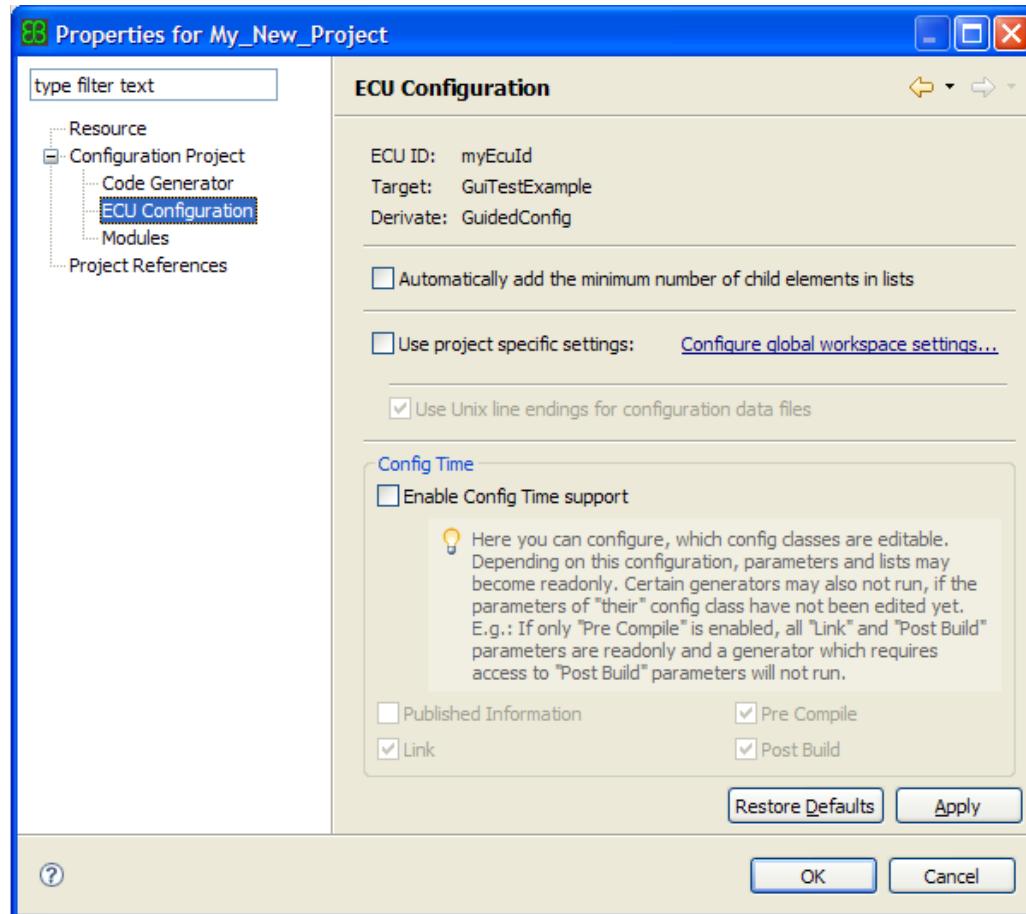


- ▶ The **Generation Path** textbox controls into which directory the generated code files are written to. If the value is a relative path, the path is considered to be relative to the project's directory. If you did not change the generation path during the creation of the project, the value is `output`. If you want to change the value, you may enter an absolute or a relative path. You can also use the predefined variables that are available by clicking the **Variables** button. There are variables defining the workspace directory, the tresos installation directory and so on.
- ▶ If the **Use project specific settings** checkbox is not set, the global workspace-wide value for this property is used. You can view and change this setting by clicking on the link *Configure global workspace settings* (see [Section 7.2.3.5, “Setting configuration preferences”](#)).
 - ▶ If you want to change the unix line endings for generated code just for this project:
 - ▶ Select *Code Generator* from the tree view.
 - ▶ Check the **Use project specific settings** checkbox.
 - ▶ Check the **Generate code with Unix line endings** checkbox if you want to have Unix line endings in your generated files. Uncheck the checkbox if you want to have Windows line endings in your files.

7.3.7.3. Editing the ECU Configuration settings

The *ECU Configuration* page allows you to change settings that influence the way how ECU configurations are handled by the tool. To open this page:

- ▶ Select *ECU Configuration* from the tree view on the left.



The main part of the dialog provides you with several options:

- ▶ You can view the *ECU ID*, *Target* and *Derivate* settings. These values have been set when the project was generated (see [Section 7.3.1.3, “Step 3: Configuring project properties”](#) and cannot be changed).
- ▶ If you want lists to be filled with default entries, select the **Automatically add...** check box. If the check box is cleared, the list elements will not be added automatically.
- ▶ You can change which line endings the configuration data files use. If the **Use project specific settings** check box is not set, the global workspace-wide value for this property is used. You can view and change this setting by clicking on the link *Configure global workspace settings* (see [Section 7.2.3.5, “Setting configuration preferences”](#)).

To change the setting for Unix line endings for configuration data files in this project independently of the global settings in EB tresos Studio, check **Use project specific settings**. Then change *Use Unix line endings for configuration data files* to the value you would like to have in this project.

- ▶ You can change the settings of the configuration time support. These values have been initially set when the project was created and can also be changed here (see [Section 7.3.1.3, “Step 3: Configuring project properties”](#)).

7.3.7.4. Viewing information about the project's modules

To view and edit the module configurations of your project, select *Modules* from the tree view at the left. The *Modules* page opens up.

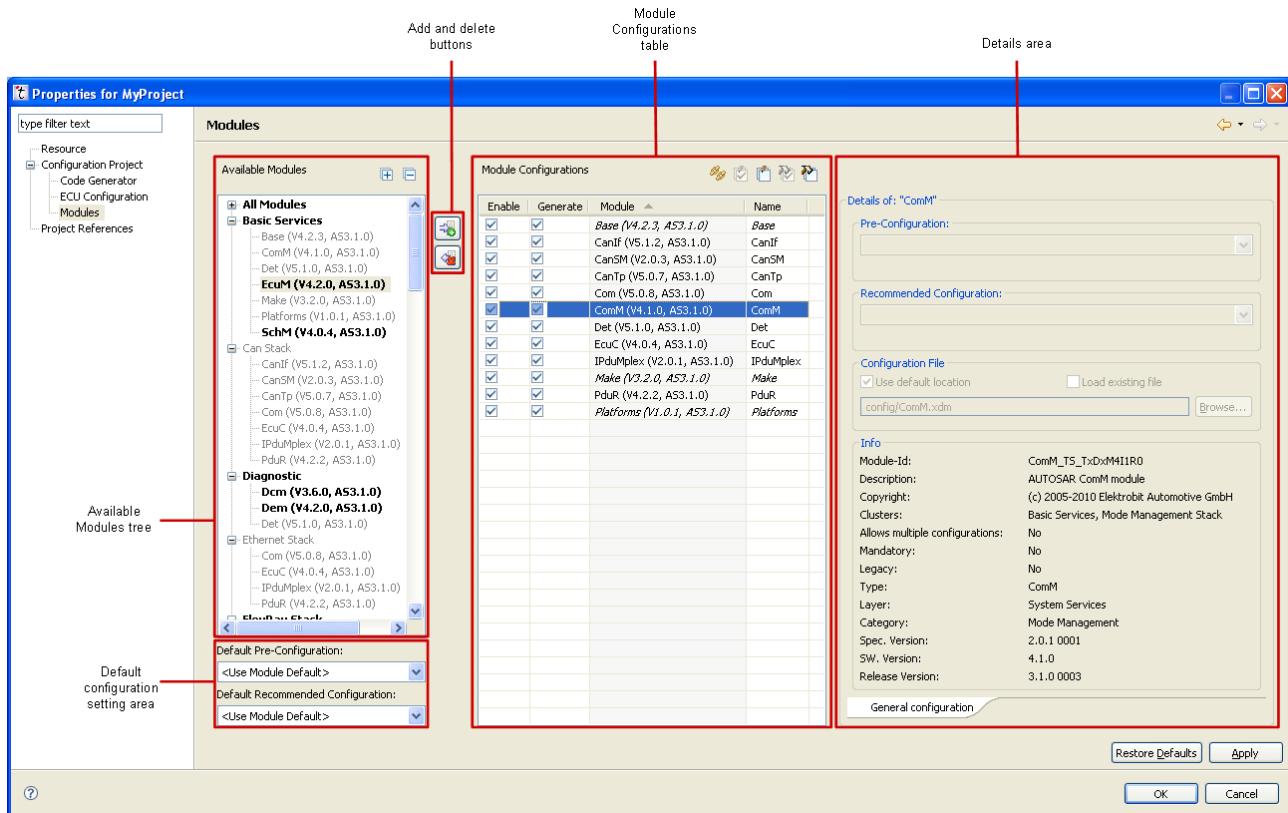


Figure 7.22. Editing module configurations on the *Modules* page of the project properties

7.3.7.4.1. Adding module configurations to your project

You may add single module configurations, several module configurations at once or clusters of module configurations to your project.

NOTE**You can recognize whether a module has been added by its font**

Modules that are still available and modules that already have been added to your project are displayed in different font styles in the *Available Modules* tree:

- ▶ Modules that have not been added to your project and thus are still available are displayed in bold font.
- ▶ Modules that already have been added to your project are displayed in standard font. The modules that allow multiple configurations are displayed in black. All others are grayed-out and cannot be added to your project anymore.

To add new module configurations to your project:

- ▶ In the *Available Modules* tree, select all modules/clusters of modules for which you want to add configurations for.
- ▶ Click the **Add** button .

TIP**Add single modules with a double-click**

Module configurations for single modules can also be added by double-clicking the required module.

- ▶ If the required module configurations cannot be added to your project, the following dialog opens up:

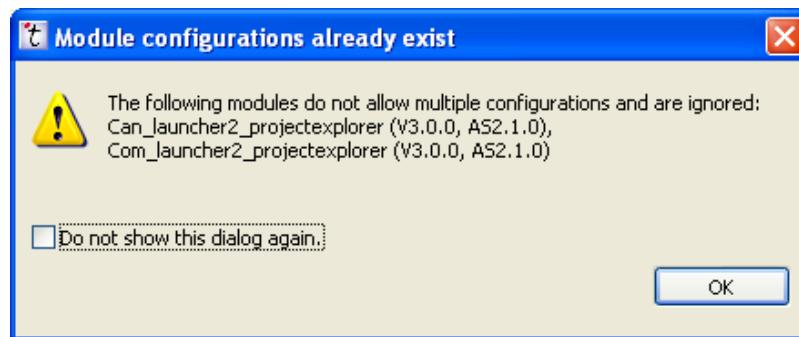


Figure 7.23. Some modules may be added only once to your project

The selected modules are added to your project as module configurations with default settings.

NOTE**Only one module configuration is added per selected module**

Since modules can be part of more than one cluster, it is possible that a selection in the *Available Modules* tree refers several times to the same module. Note that only one configuration is added in this case, even if the module allows multiple configurations.

As long as the recently added module configurations have not been stored into your project yet, you may edit the (default) settings of the module configuration. Mind that depending on the module, there might be more tab pages next to the *General configuration* tab on the bottom of the *Details* area.

7.3.7.4.2. Removing module configurations from your project

You may remove single module configurations or several module configurations at once from your project.

NOTE**You can recognize the state of a module by its font**

Module configurations that previously had been added to your project, mandatory module configurations and recently added module configurations are displayed in different fonts in the *Modules Configuration* table:

- ▶ Mandatory module configurations are displayed in italic font. It is not recommended to remove these module configurations from your project.
 - ▶ Recently added module configurations are displayed in bold font. You may edit them.
 - ▶ Module configurations that had been added to your project before are displayed in standard font.
-

To remove module configurations from your project:

- ▶ In the *Module Configurations* table, select the module configuration(s) you want to remove.

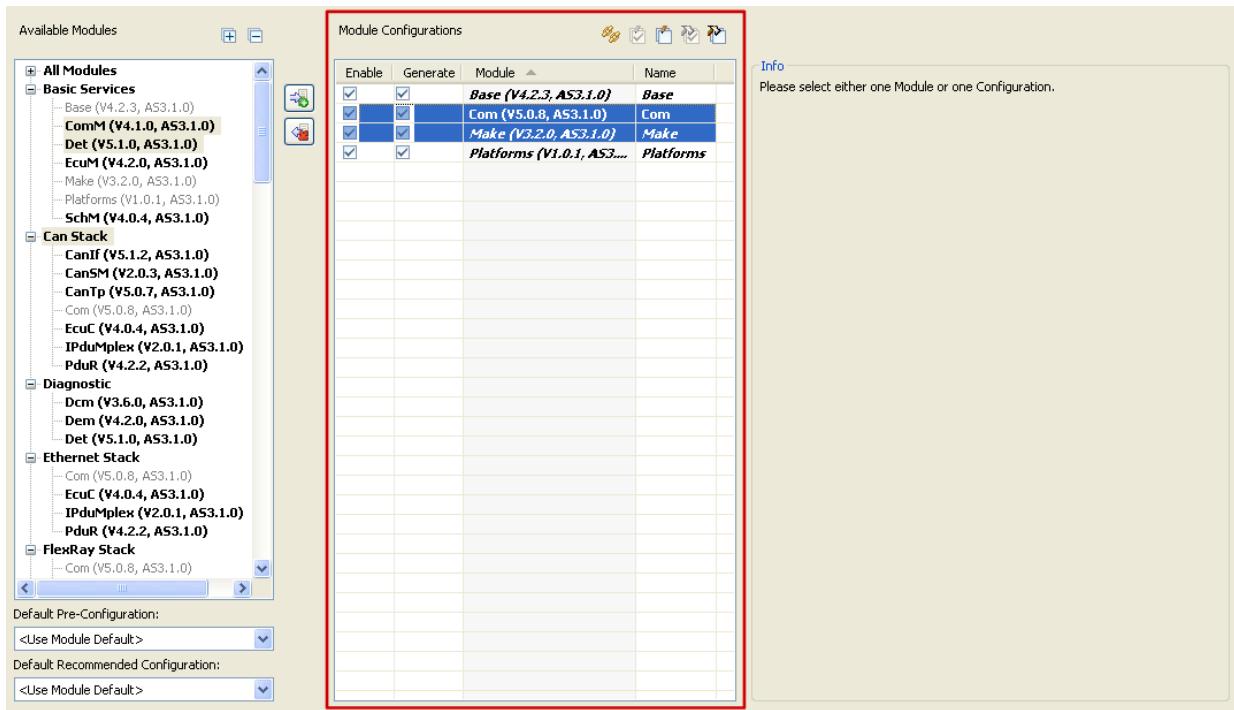


Figure 7.24. Removing module configurations

- ▶ Click the **Delete** button .
- ▶ If none of the selected module configurations refers to an already existing configuration file, the items selected are removed from the *Module Configurations* table without further notice. Otherwise the *Delete Module Configurations* dialog opens up

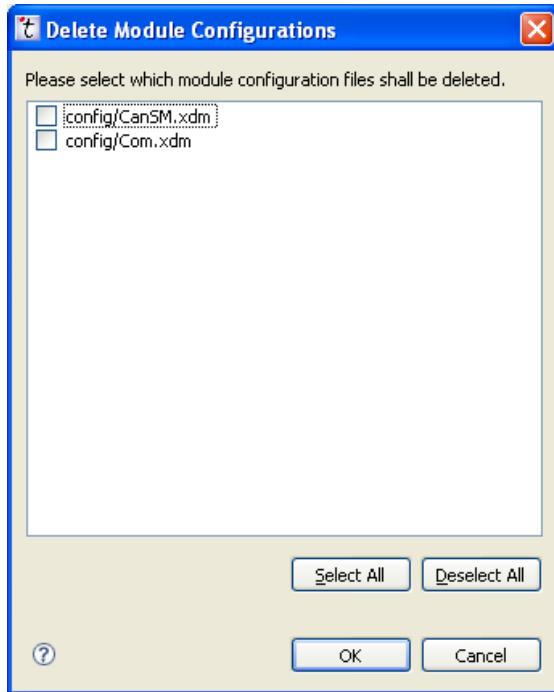


Figure 7.25. Confirming which module configuration files you want to remove

- ▶ Select the check box of the configuration files you want to delete.

NOTE**Data loss**

The selected files will be deleted from the file system when the changed module configurations are applied. After that, you cannot restore these files anymore.

-
- ▶ To confirm the deletion of the configuration files, click **OK**.
The items selected are removed from the *Module Configurations* table.
 - ▶ To cancel the deletion of the *Module Configurations* and their referred configuration files, click **Cancel**.

7.3.7.4.3. Enabling and disabling module configurations

In the *Module Configurations* table you may (temporarily) enable or disable the module configurations you have added to your project. Disabled configurations are treated as if they do not exist, which means you cannot edit or verify them, and code generators cannot access them. To enable module configurations:

- ▶ To enable only one module configuration, select the **Enable** check box of the module configuration you want to enable in the *Module Configurations* table.

Module Configurations				
Enable	Generate	Module	Name	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Base (V4.2.3, AS3.1.0)	Base	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CanSM (V2.0.2, AS3.1.0)	CanSM	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Com (V5.0.8, AS3.1.0)	Com	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Make (V3.2.0, AS3.1.0)	Make	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Platforms (V1.0.1, AS3....)	Platforms	

Figure 7.26. Enabling one module configuration

- ▶ To enable several module configuration at once:
 - ▶ In the *Module Configurations* table, select the module configurations you want to enable.
 - ▶ In the tool bar of the *Module Configurations* table, click the **Enable** button

To disable module configurations:

- ▶ To disable only one module configuration in the *Module Configurations* dialog/page, clear the **Enable** check box of the module configuration you want to disable in the *Module Configurations* table.

Module Configurations				
Enable	Generate	Module	Name	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Base (V4.2.3, AS3.1.0)	Base	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CanSM (V2.0.2, AS3.1.0)	CanSM	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Com (V5.0.8, AS3.1.0)	Com	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Make (V3.2.0, AS3.1.0)	Make	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Platforms (V1.0.1, AS3....)	Platforms	

Figure 7.27. Disabling one module configuration

To disable several module configuration at once:

- ▶ In the *Module Configurations* table, select the module configurations you want to disable.
- ▶ In the tool bar of the *Module Configurations* page, click the **Disable** button

7.3.7.4.4. Enabling and disabling code generation for module configurations

In the *Module Configurations* table you may enable or disable code generation for the module configurations of your project.

WARNING Only partly generated code leads to incorrectly working ECUs


If you only generate the code of a part of the module configurations you later want to implement on an ECU, depending on which modules you choose, the code might not work correctly. This might lead to incorrectly working ECUs.

To avoid incorrectly working ECUs, generate the code of all module configurations in your project before deploying them on an ECU.

To enable code generation for module configurations:

- ▶ To enable code generation for only one module configuration, select the **Generate** check box of the module configuration for which you want to enable code generation in the *Module Configurations* table.

Module Configurations				
Enable	Generate	Module	Name	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Base (V4.2.3, AS3.1.0)	Base	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	CanSM (V2.0.2, AS3.1.0)	CanSM	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Com (V5.0.8, AS3.1.0)	Com	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Make (V3.2.0, AS3.1.0)	Make	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Platforms (V1.0.1, AS3....	Platforms	

Figure 7.28. Enabling code generation for one module configuration

- ▶ To enable code generation for several module configurations at once:
 - ▶ In the *Module Configurations* table, select the module configurations, for which you want to enable code generation.
 - ▶ In the tool bar of the *Module Configurations* page, click the **Enable code generation** button

To disable code generation for module configurations:

- ▶ To disable code generation for only one module, clear the **Generate** check box of the module configuration for which you want to disable code generation in the *Module Configurations* table.

Module Configurations				
Enable	Generate	Module	Name	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Base (V4.2.3, AS3.1.0)	Base	
<input type="checkbox"/>	<input type="checkbox"/>	CanSM (V2.0.2, AS3.1.0)	CanSM	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Com (V5.0.8, AS3.1.0)	Com	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Make (V3.2.0, AS3.1.0)	Make	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Platforms (V1.0.1, AS3....	Platforms	

Figure 7.29. Disabling code generation for one module configuration

- ▶ To disable code generation for several module configurations at once:

- ▶ In the *Module Configurations* table, select the module configurations, for which you want to disable code generation.
- ▶ In the tool bar of the *Module Configurations* page, click the **Disable code generation** button

7.3.7.4.5. Naming *Module Configurations*

When you add new module configurations to your project, the name of the new module configuration is set automatically to a name that is unique within the project using either the default configuration name provided by the module or the module name itself.

Instead of using the default name you may want to use another name for the module configuration.

NOTE
Only in recently added module configurations the name can be changed


You can only change the name of a module configuration if you added this module configuration during the same session of the dialog.

To edit the name of a module configuration:

- ▶ Add the required module configuration to your project.
- ▶ In the *Module Configurations* table, click into the *Name* cell of the module configuration whose name you want to change.
- ▶ Type the new name.



Figure 7.30. Editing the name of a new module configuration

NOTE

Name constraints

Mind that the module configuration name must meet the following constraints:

- ▶ The name must be unique within the project.
- ▶ The name must start with a letter.
- ▶ The name must contain only letters and digits plus the underscore character ('_').
- ▶ The name must not be empty.
- ▶ The name must not exceed 255 characters.

- ▶ Press **ENTER**.

The name of the module configuration is changed.

7.3.7.4.6. Choosing *Pre-Configurations*

Module suppliers optionally can equip modules with one or more preconfigurations that contain values for configuration parameters. If a module configuration of your project provides one or more preconfiguration(s), you must assign this module configuration to such a preconfiguration when adding it to your project. The parameters covered by the chosen preconfiguration are then fixed in the module configuration and cannot be edited by the user.

In the *Default configuration setting* area, you can set the *Default Pre-Configuration* which is used as a default value for all new module configurations. To support a more consistent selection, this preconfiguration setting is persisted. You can also edit the *Pre-Configuration* of an individual module configuration.

Setting the *Default Pre-Configuration*

In the *Default configuration setting* area of the *Module Configurations* page, you may edit the *Default Pre-Configuration* for new module configurations.

NOTE**The value set is a union set of all preconfigurations provided by all modules**

The preconfiguration names to choose from 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 preconfiguration is taken instead when you add it to your project.

To change the default preconfiguration, select the required preconfiguration in the **Default Pre-Configuration** drop-down list box.

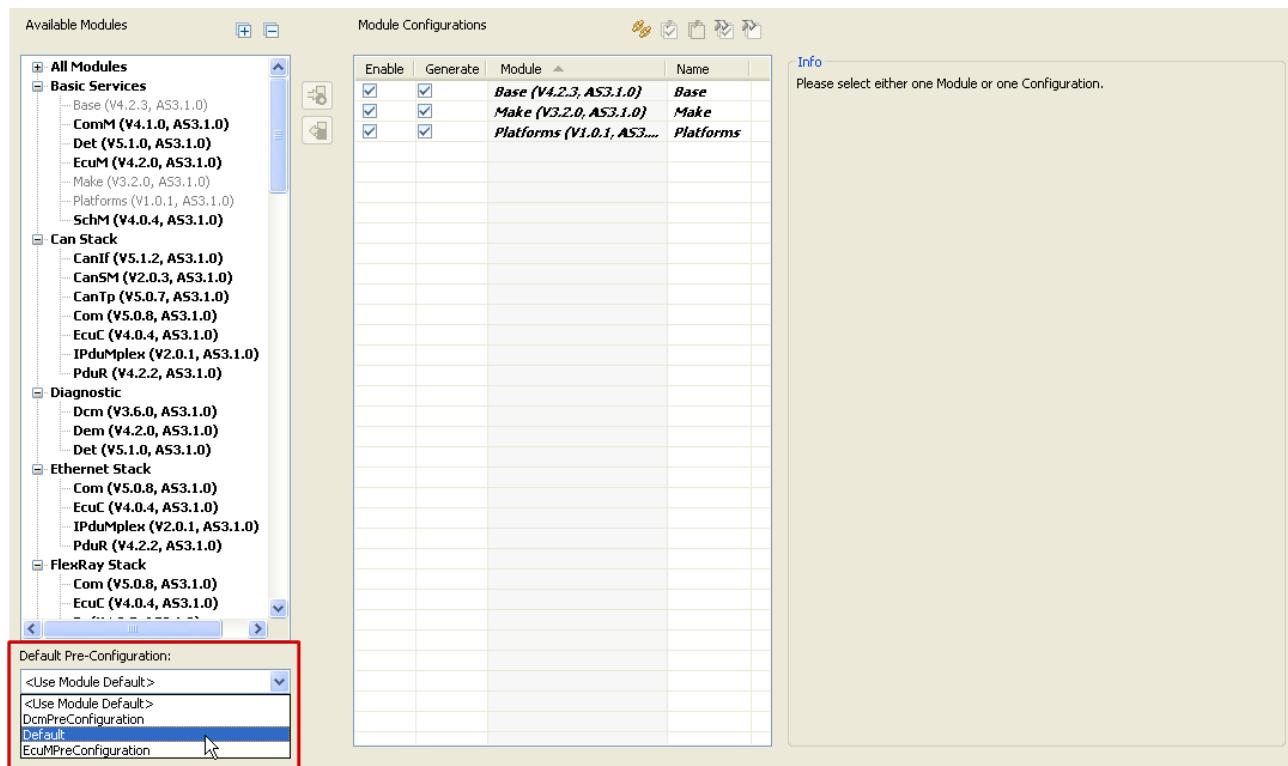


Figure 7.31. Selecting the default preconfiguration for new module configurations

- ▶ To use the respective default preconfiguration for new module configurations, select <Use Module Default>.

NOTE**Non-compilable code**

If you select <Use Module Default>, every module will be added to your project with a different preconfiguration. This might lead to incompatibilities of the module configurations and thus to non-compilable code.

- ▶ The selected default preconfiguration is saved, whenever you actually add new module configurations to your project.

Changing the preconfiguration of one specific module configuration

To change the preconfiguration of one specific module configuration:

- ▶ Add the required module configuration to your project.

NOTE**Only in recently added module configurations the preconfiguration can be changed**

You can only change the preconfiguration of a specific module configuration if you added this module configuration during the same session of the dialog.

- ▶ In the *Module Configurations* table, mark the module configuration whose preconfiguration you want to change.

In the *Details* area of the *Module Configurations* page, detailed information about the specific module is displayed.

The screenshot shows the EB tresos Studio interface with the 'Module Configurations' page open. On the left, there is a tree view of 'Available Modules' under 'All Modules'. The 'Com' module is selected in the main table, which has columns for 'Enable', 'Generate', 'Module', and 'Name'. The 'Name' column shows entries like 'Base (V4.2.3, AS3.1.0)', 'Com (V5.0.8, AS3.1.0)' (which is selected), 'Make (V5.2.0, AS3.1.0)', and 'Platforms (V1.0.1, AS3.1.0)'. To the right of the table is a large 'Details' dialog box for the selected 'Com' module. This dialog contains several sections: 'Pre-Configuration:' (a dropdown menu), 'Recommended Configuration:' (set to 'ComRecConfigurationStandard'), 'Configuration File' (with a 'config/Com.xdm' field and a 'Browse...' button), and 'Info' (a detailed list of module properties). The 'Info' section includes fields like 'Module-Id', 'Description', 'Copyright', 'Clusters', 'Allows multiple configurations', 'Mandatory', 'Legacy', 'Type', 'Layer', 'Category', 'Spec. Version', 'SW. Version', and 'Release Version'. A red box highlights the 'Info' section of the 'Details' dialog.

Figure 7.32. The *Details* area in the *Module Configurations* page

- ▶ In the **Pre-Configuration** drop-down list box in the *Details* area, select the required preconfiguration for the selected module configuration.

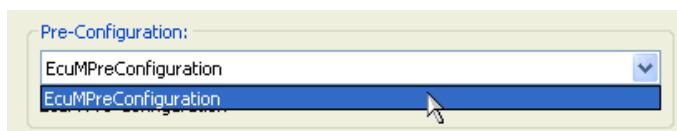


Figure 7.33. Changing the preconfiguration of one specific module

NOTE**Grayed-out preconfigurations are not available**

If the **Pre-Configuration** drop-down list is grayed-out, the selected module does not provide a preconfiguration of its own.

7.3.7.4.7. Applying recommended configurations

Module suppliers optionally can equip modules with one or more recommended configurations that contain values for configuration parameters. If a module configuration of your project provides one or more recommended configurations, you can choose one when you add a configuration for this module to your project. The parameter values covered by the chosen recommended configuration can be considered as a guideline. They may be edited by the user later.

In the *Default configuration setting area*, you can set the *Default Recommended Configuration* which is used as a default value for all new module configurations. To support a more consistent selection, this setting is persisted. You can also edit the *Recommended Configuration* of an individual module configuration.

Setting the *Default Recommended Configuration*

In the *Default configuration setting area* of the *Module Configurations* page, you may edit the *Default Recommended Configuration* for new module configurations.

NOTE**The value set is a union set of all recommended configurations provided by all modules**

The names of the recommended configurations to choose from 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 configuration is taken instead when you add it to your project. If the module does not provide a default configuration, no recommended configuration is applied.

To change the default recommended configuration, select the required recommended configuration in the **Default Recommended Configuration** drop-down list box.

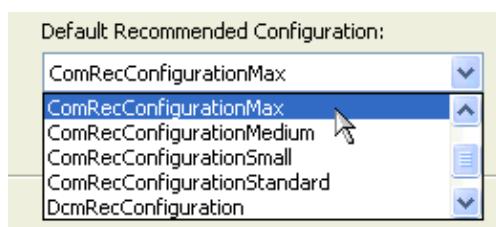


Figure 7.34. Selecting the default recommended configuration for new module configurations

- ▶ If you don't want to apply any of the recommended configurations, select <None>.
- ▶ To use the respective default recommended configuration for new module configurations, select <Use Module Default>.

NOTE**Non-compilable code**

If you select <Use Module Default>, every module will be added to your project with a different recommended configuration. This might lead to incompatibilities of the module configurations and thus to non-compilable code.

- ▶ The selected default recommended configuration is saved, when you actually add new module configurations to your project.

Changing the recommended configuration of one specific module configuration

To change the recommended configuration of one specific module configuration:

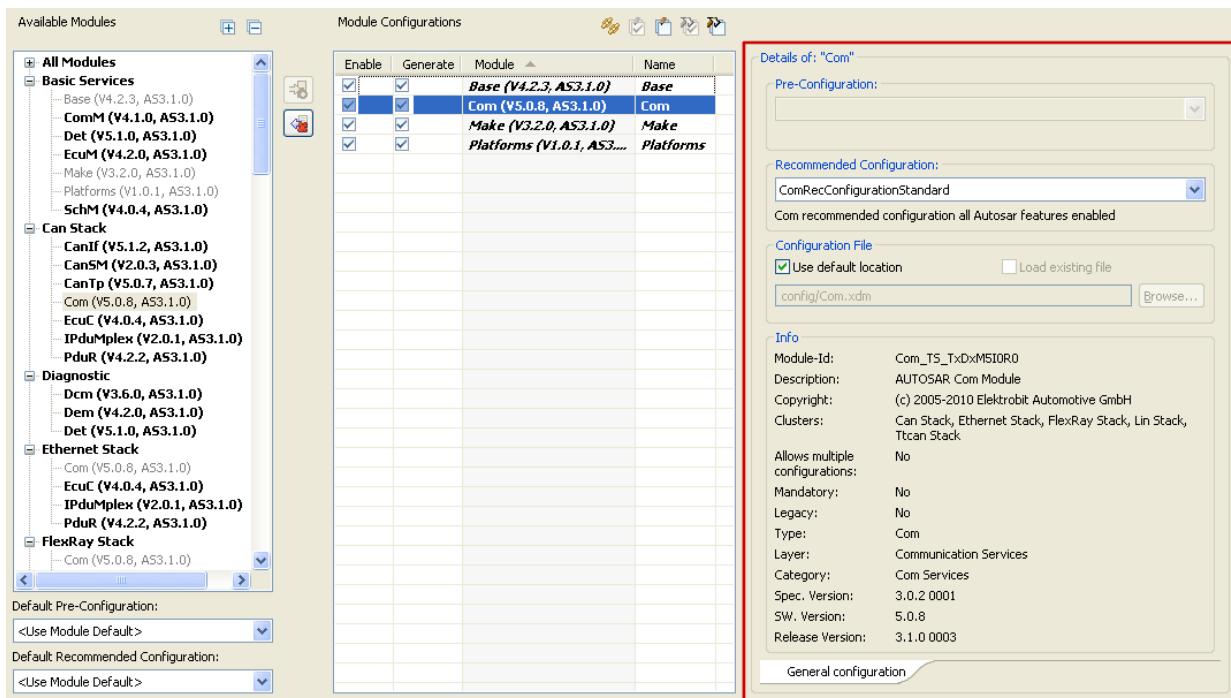
- ▶ Add the required module configuration to your project.

NOTE**Only in recently added module configurations the recommended configuration can be changed**

You can only change the recommended configuration of a specific module configuration if you have added this module configuration during the same session of the dialog.

- ▶ Mark the module configuration whose recommended configuration you want to change.

In the *Details* area of the *Module Configurations* page, detailed information about the specific module is displayed.

Figure 7.35. The *Details* area in the *Module Configurations* page

- ▶ In the **Recommended Configuration** drop-down list box, select the required recommended configuration for the selected module configuration.

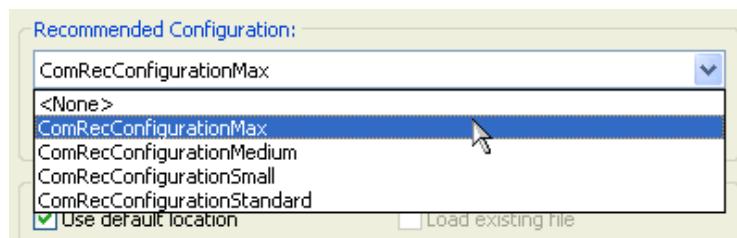


Figure 7.36. Changing the recommended configuration of one specific module

NOTE**Grayed-out recommended configurations are not available**

If the **Recommended Configuration** drop-down list is grayed-out, the selected module does not provide a recommended configuration of its own.

7.3.7.4.8. Editing the configuration file settings

The configuration file stores all parameters for a specific module configuration. For detailed information on how to edit those parameters, e.g. by using the Generic Configuration Editor, refer to [Section 7.8, “Editing parameters of a module configuration”](#).

When you add new module configurations to your project, per default the configuration file name is set to config/<moduleconfigurationname>.xdm, relative to your project folder. This is also referred to as the default location.

Instead of using the default location you can use another location or name for the configuration file. To use another location or file name:

- ▶ Add the required module configuration to your project.

NOTE**Only in recently added module configurations the configuration file can be changed**

You can only change the configuration file of a module configuration, if you have added this module configuration during the same session of the dialog.

- ▶ In the *Module Configurations* table, mark the module configuration whose configuration file you want to change.
- ▶ In the *Configuration File* panel of the *Details* area, clear the check box **Use default location**.

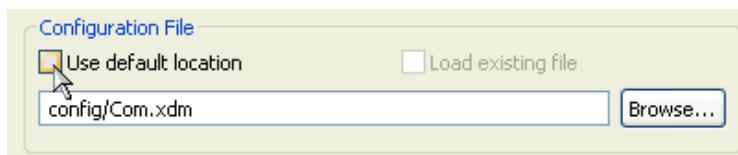


Figure 7.37. Editing the configuration file settings

- ▶ Enter or choose the new file name. Either use absolute path names or path names relative to your project.

NOTE**The file extension .xdm is mandatory**

You can only use file names ending with the extension .xdm. When you choose an existing file, you must make sure that it matches the targeted module configuration schema. Otherwise the new module configuration will fail to load when applied.

Using existing configuration files

Instead of configuring the module configuration from scratch you may also use an already existing configuration file.

NOTE**You cannot set preconfigurations and recommended configurations when you load an existing configuration file**

If you use an existing configuration file, you cannot apply any preconfiguration or recommended configuration.

To load an already existing configuration file into your project:

- ▶ In the *Configuration File* panel, browse to the file you want to load.

NOTE**The selected configuration file will be modified**

The selected configuration file will be modified while you edit the parameters of the module configuration. If you want to preserve the file as it is, copy the file to your project before loading it to your project.



Figure 7.38. Using an existing configuration file

Whenever you point the *Configuration File* panel to an existing file in the file system, the *Load existing file* check box is marked.

NOTE**Data loss**

If you clear the *Load existing file* check box, the existing file will be overwritten and cannot be restored again.

7.3.7.4.9. Upgrading module configurations

To upgrade module configurations from older software versions of the modules to newer ones:

- ▶ Click the **Upgrade** button  in the tool bar of the *Module Configurations* table.

The *Upgrade* dialog opens up.

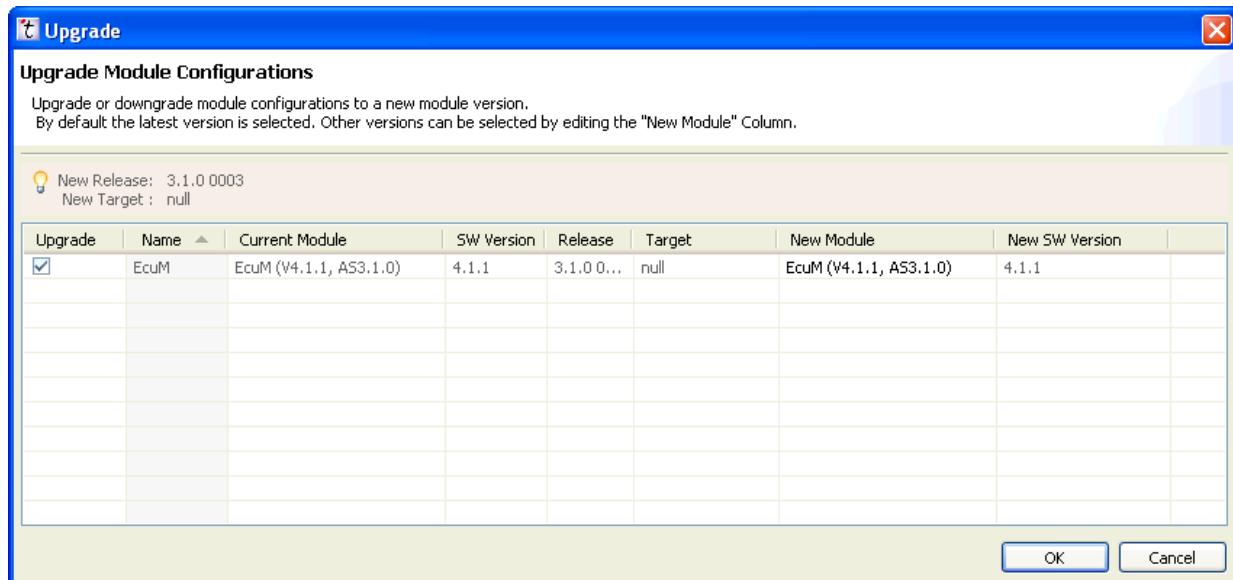


Figure 7.39. Upgrading module configurations in the *Upgrade* dialog

For detailed information on the *Upgrade* dialog, refer to [Section 7.7, “Upgrading projects and configurations”](#).

7.4. Using team collaboration support

7.4.1. Overview

This chapter provides instructions for working as a team with EB tresos Studio. This includes instructions on:

- ▶ working with version control systems such as CVS and SVN,
- ▶ synchronizing files and file versions,

This chapter contains instructions for tasks you perform when you work with a team with EB tresos Studio. For detailed background information on the Eclipse CVS support, refer to the [Eclipse Workbench User Guide](#). For detailed background information on the Eclipse SVN support, refer to the [Subclipse - Subversion Eclipse Plugin documentation](#). Both of these Eclipse documents are included in the EB tresos Studio onscreen help, which opens up when you press **F1**.

7.4.2. Working with CVS repositories

This chapter is interesting for you, if you want to connect your EB tresos Studio with your CVS repository and commit to shared EB tresos Studio projects directly from the EB tresos Studio GUI.

The state (e.g. locally modified, added) of your local files compared to the files in the CVS repository is typically indicated by overlay icons. The following overlay icons are displayed in the *Project Explorer* view:

Project Explorer overlay icons for CVS repository support



Indicates that the file or folder is under version control, with or without local modifications. Versioned files that are locally modified are additionally marked with a > character in front of the name in the *Project Explorer* view, e.g.

..  >Can.xdm



Indicates a new file or folder within a version controlled folder that has not yet been added to version control.

7.4.2.1. Registering your CVS repository location in EB tresos Studio

To access your CVS repositories from within EB tresos Studio you first need to register the repository location in EB tresos Studio. To register the repository location in EB tresos Studio:

- ▶ In the **Window** menu, select *Show view*. A context menu opens up.
- ▶ In the context menu that opens up, select *Other...*. The *Show View* dialog opens up.
- ▶ Expand *CVS* and select *CVS Repositories*. The *CVS Repositories* view opens up as a tab in the *Element Outline* view in the lower right corner of the EB tresos Studio workbench.

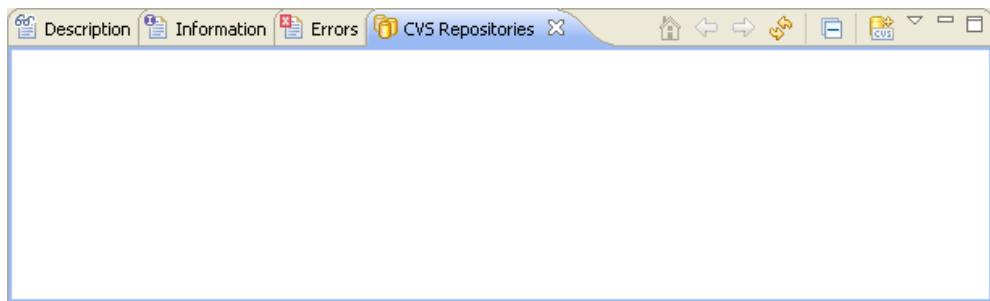


Figure 7.40. Registering a new repository location in the *CVS Repositories* view

- ▶ To register a new repository to be accessed via EB tresos Studio, click the button . The *Add CVS Repository* dialog opens up.

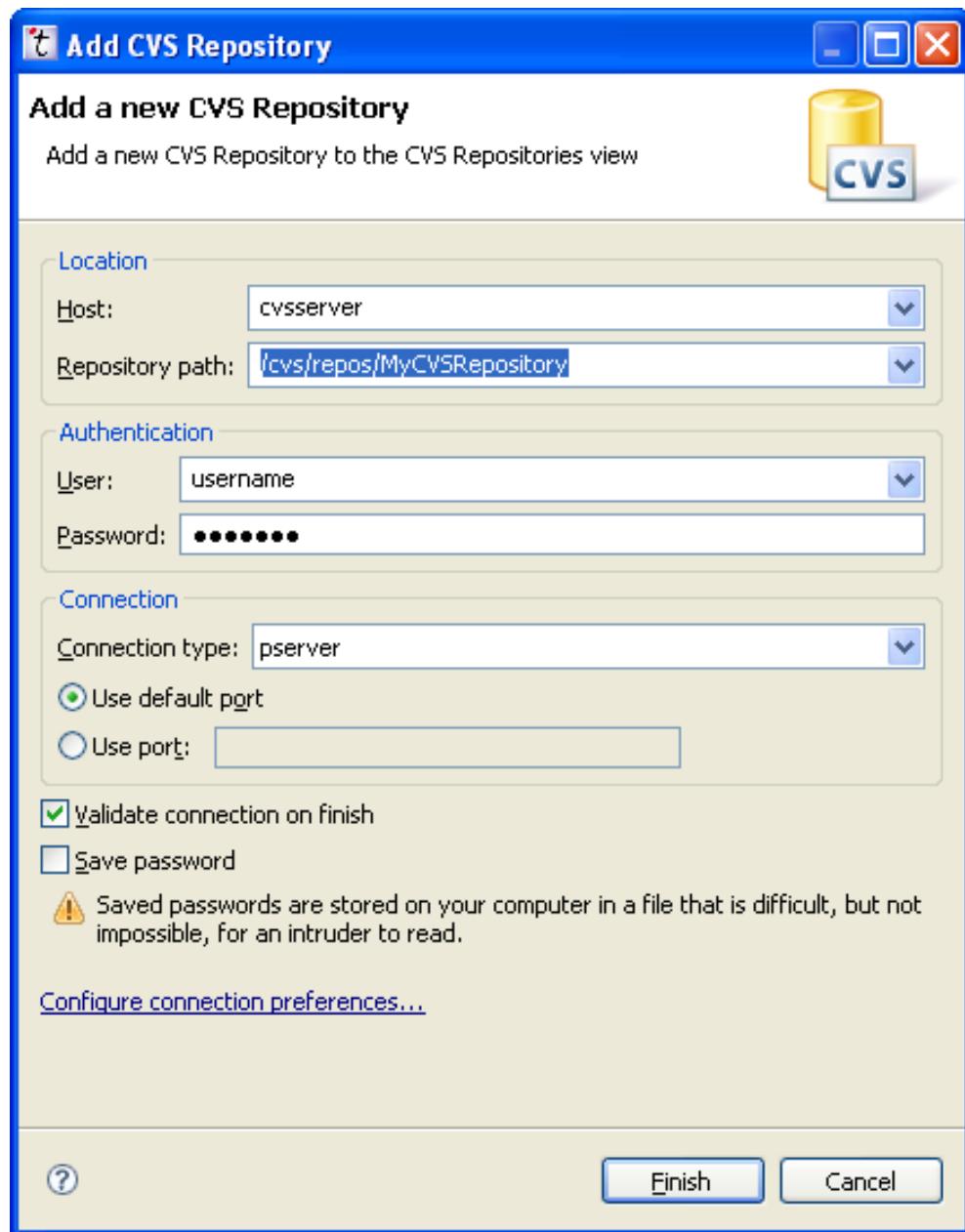


Figure 7.41. Adding a new repository location in EB tresos Studio

- ▶ Enter the connection data. If you do not know your connection data, consult your local CVS repository administrator.
- ▶ Click **Finish**. The repository is displayed in your *CVS Repositories* view.

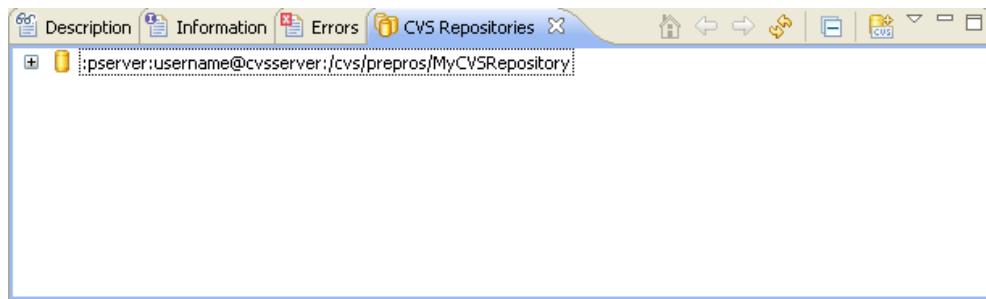


Figure 7.42. Registered repositories in the *CVS Repositories* view

7.4.2.2. Sharing an EB tresos Studio project with other team members

In this section you will learn how to share a project with other team members. This is equivalent to importing files to a CVS repository and thus only applies to projects that have not yet been added to the repository.

NOTE**You can only share whole projects, not single files**

Adding single files to a repository is not possible from within EB tresos Studio. Thus, if you want to share new files, you must share the project they reside in first.

To share a new project:

- ▶ Right-click this project in the *Project Explorer* view, select *Team* and then *Share Project...* from the context menus that open up:

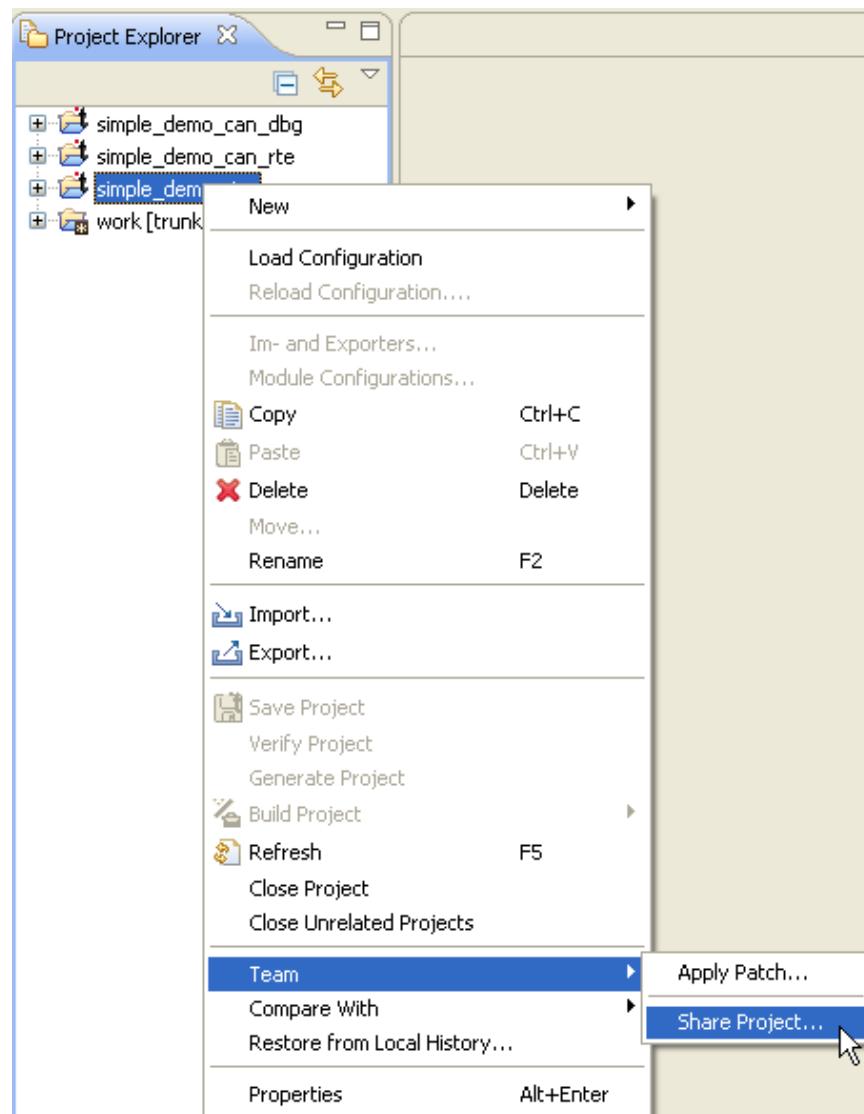


Figure 7.43. Selecting a project to share

- The *Share Project* dialog opens up.

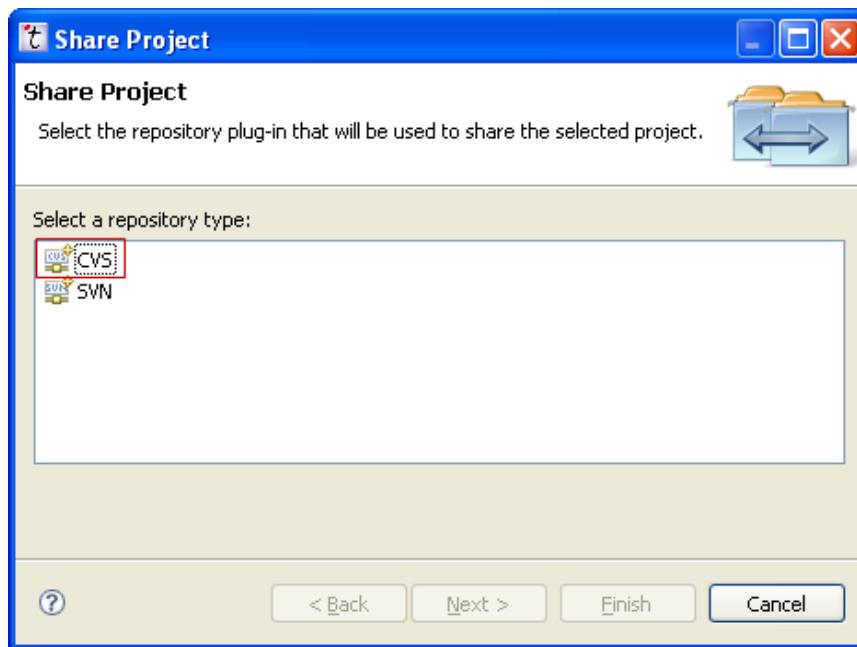


Figure 7.44. Selecting the repository type in which you want to share your project

- Select CVS and click **Next**.

The *Share Project with CVS Repository* page opens up

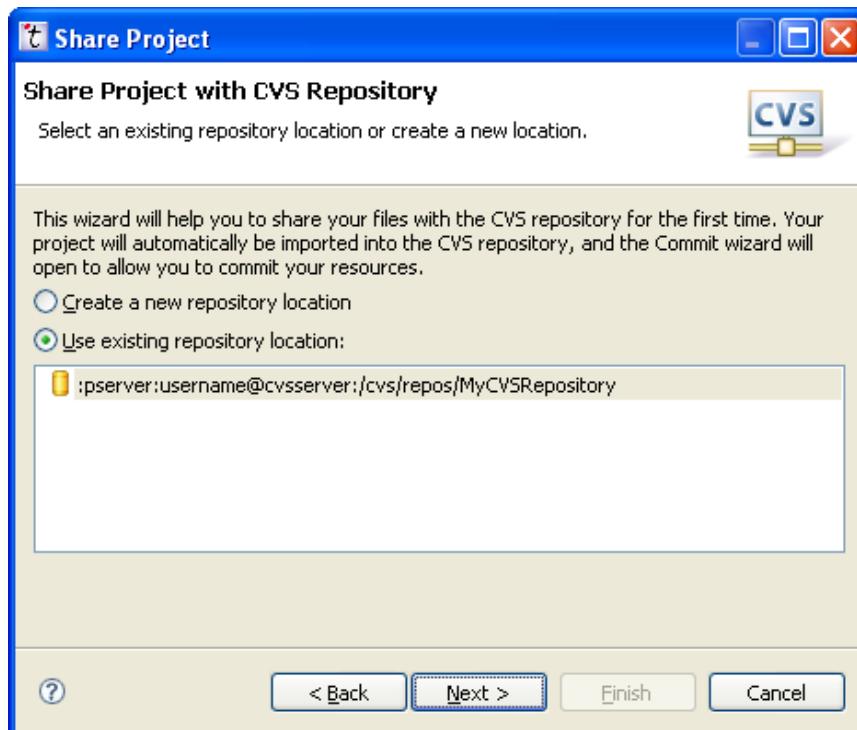


Figure 7.45. Selecting the repository location in which you want to share your project

- ▶ Select the repository you want to add your project to. You can either select an already registered repository location or register a new repository location. For information on how to register a new repository, refer to [Section 7.4.2.1, “Registering your CVS repository location in EB tresos Studio”](#).
- ▶ Click **Next**. The *Enter Folder Name* page opens up.

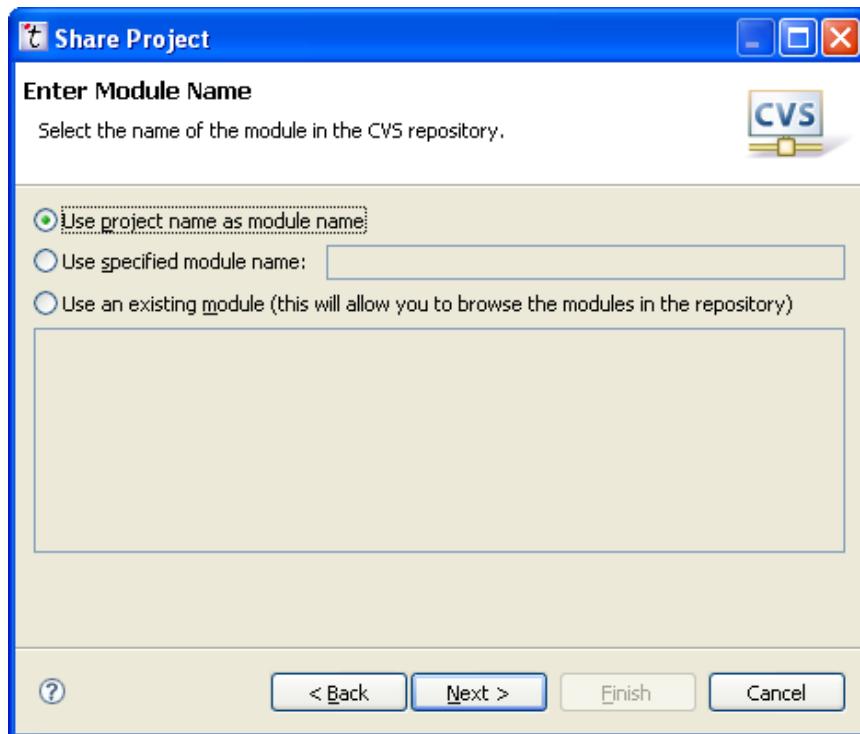


Figure 7.46. Entering a CVS module name for your project in the repository

- ▶ Enter the folder name in which you want to add your project to your CVS repository and click **Next**. The *Share Project Resources* page opens up.

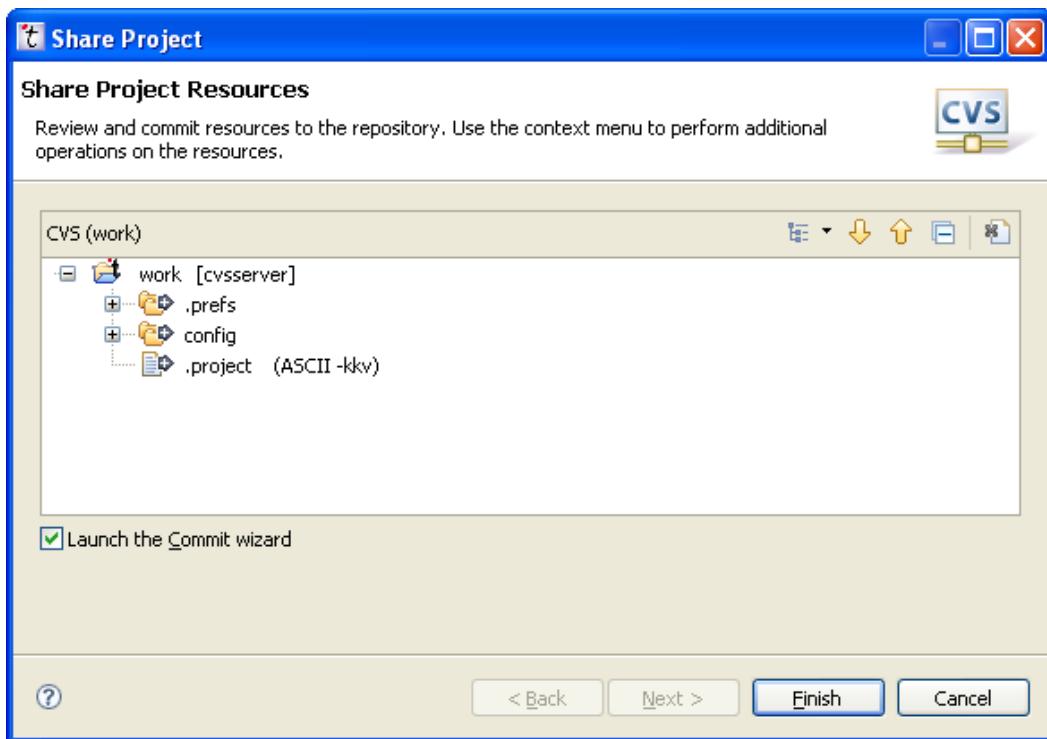


Figure 7.47. Reviewing and committing resources

- ▶ You can review the project content that will be shared and perform additional operations, such as adding files to the CVS ignore list, or excluding selected files that must not be added at this moment.
- ▶ Click **Finish**.

Your project is now under CVS control.

7.4.2.3. Checking out a shared project from your CVS repository into your EB tresos Studio workspace

In this section you will learn how to check out a CVS-managed project into your EB tresos Studio workspace. You can check out a project from a CVS repository into your EB tresos Studio workspace directly from the EB tresos Studio GUI.

To check out a project from a CVS repository into your EB tresos Studio workspace, you need to open the *Checkout Out As* dialog first. To open the *Checkout Out As* dialog:

- ▶ Right-click the shared project's folder in the *CVS Repositories* view, and select *Check Out As...* from the context menu. The *Check Out As* dialog opens up.

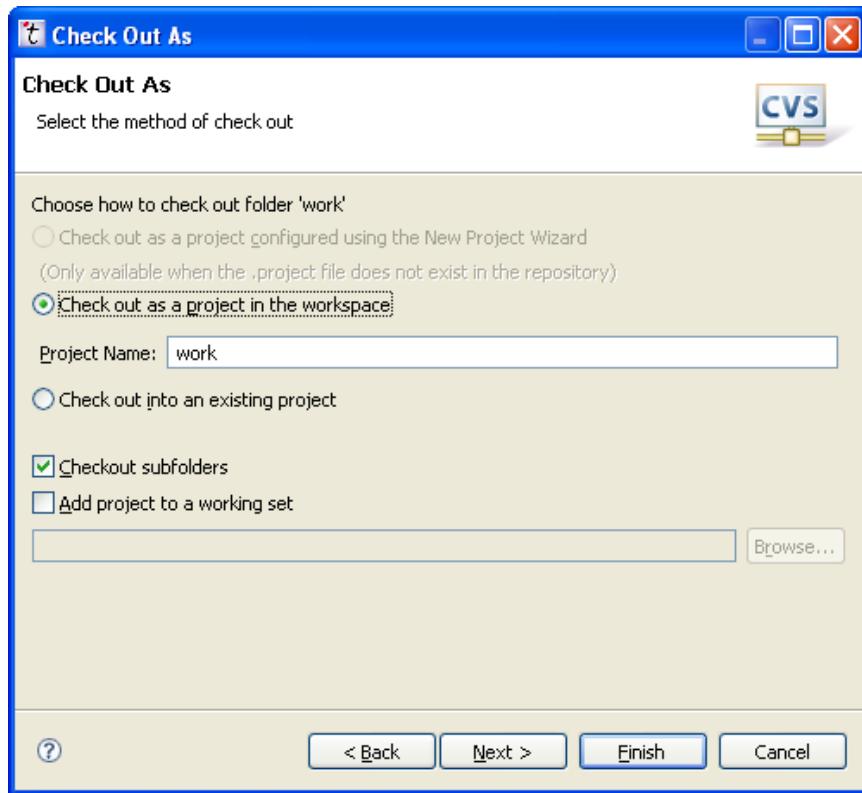


Figure 7.48. Checking out from CVS repository

- ▶ In the *Check Out As* dialog, you can specify how to check out the chosen CVS module:
 - ▶ If the selected CVS module already contains the project infrastructure file `.project`, select **Check out as project in the workspace**.
 - ▶ If the chosen module does not yet contain the project infrastructure file `.project`, select **Check out as project configured using the New Project Wizard** and click **Next**. The *New Project Wizard* opens up.

Follow the *New Project Wizard*. For instructions on creating a new project with the *New Project Wizard*, refer to [Section 7.3.1.2, “Step 2: Creating a new project”](#).

- ▶ To check out the selected CVS module into an already existing project, select **Check out into an existing project**.
- ▶ Click **Next** to proceed to further wizard pages, where you can provide further options, such as where to checkout the files to, or which CVS tag or branch to use for the checkout.

- ▶ Click **Finish**.

The selected files are being checked out. As a result you see the checked out project in your *Project Explorer* view, displayed with an overlay icon:

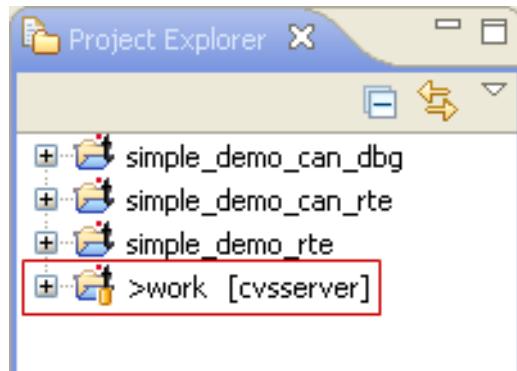


Figure 7.49. Checked out project in your *Project Explorer* view

For information on the meaning of the overlay icons, refer to [Project Explorer overlay icons for CVS repository support](#).

7.4.2.4. Updating, branching, and committing EB tresos Studio projects into your CVS repository

After you have registered and checked out your repository into EB tresos Studio, you may want to edit the files and commit your work to the repository.

Your files and folders are displayed with different overlay icons in the *Project Explorer* view. For information on the meaning of these overlay icons, refer to [Project Explorer overlay icons for CVS repository support](#).

To commit, update, branch your work to a repository or to choose another typical task for working with CVS repositories:

- ▶ Right-click the project in the *Project Explorer* view. A context menu opens up.
- ▶ In the context menu, select *Team*.
- ▶ The CVS context menu opens up.



Figure 7.50. The CVS context menu

In the CVS context menu you can select the tasks typical for working with CVS repositories, such as committing, updating, branching, tagging the repository etc.

WARNING Reload projects after updating


It is strongly recommended to only update files from CVS, when the configuration project is not loaded yet. After you have updated files in loaded configuration projects, the project needs to be reloaded. For further information on reloading the project, refer to [Section 7.3.6.4, “Reloading a project”](#).

If you want to diff files and file versions, refer to [Section 7.5, “Diffing files and file versions”](#).

7.4.3. Working with SVN repositories

This chapter is interesting for you, if you want to connect your EB tresos Studio with your SVN repository and commit to shared EB tresos Studio projects directly from the EB tresos Studio GUI.

The state (e.g. locally modified, added) of your local files compared to the files in the SVN repository is typically indicated by overlay icons. The following overlay icons are displayed in the *Project Explorer* view:

***Project Explorer* overlay icons for SVN repository support**



Indicates a local modification. Either a file is locally modified, or a folder contains a locally modified file.



Indicates a new file or folder within a version controlled folder that has not yet been added to version control.



Indicates that the file or folder was recently added to version control, but is not yet committed.



Indicates that the file or folder is under version control, without local modifications.

7.4.3.1. Registering your SVN repository location in EB tresos Studio

To access your SVN repositories from within EB tresos Studio, you need to register the repository location in EB tresos Studio first. To register the repository location in EB tresos Studio:

- ▶ In the **Window** menu, select *Show view*. A context menu opens up.
- ▶ In the context menu that opens up, select *Other....* The *Show View* dialog opens up.
- ▶ Expand *SVN* and select *SVN Repositories*. The *SVN Repositories* view opens up as a tab in the *Element Outline* view in the lower right corner of the EB tresos Studio workbench.

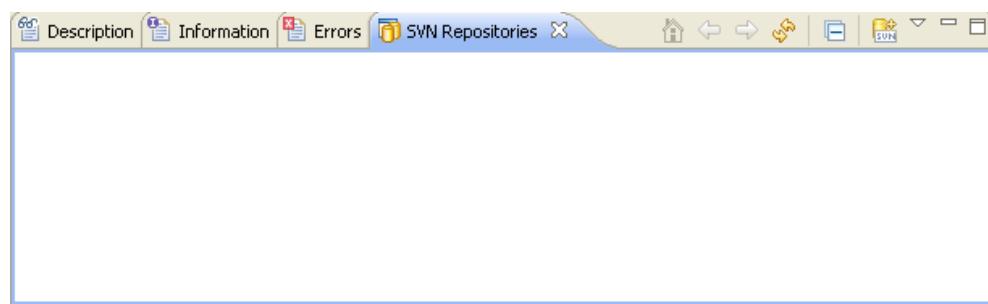


Figure 7.51. Registering a new repository location in the *SVN Repositories* view

- ▶ To register a new repository to be accessed via EB tresos Studio, click the button . The *Add SVN Repository* dialog opens up.

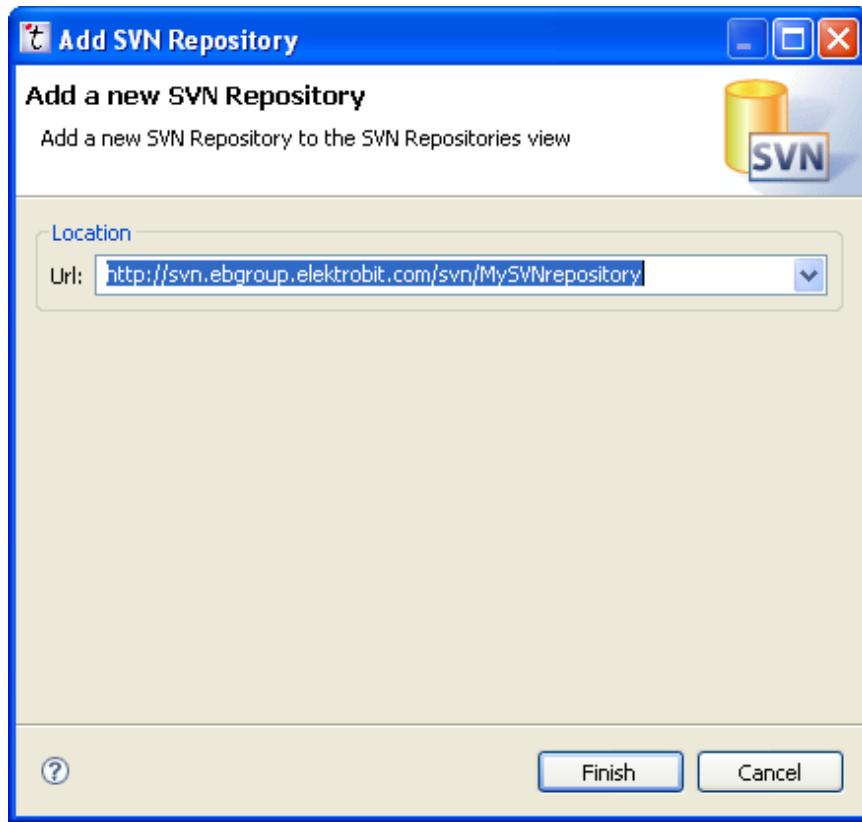


Figure 7.52. Adding a new repository location in EB tresos Studio

- ▶ In the **Location Url** text box, enter the link to your repository.
- ▶ Click **Finish**. The repository is displayed in your *SVN Repositories* view.

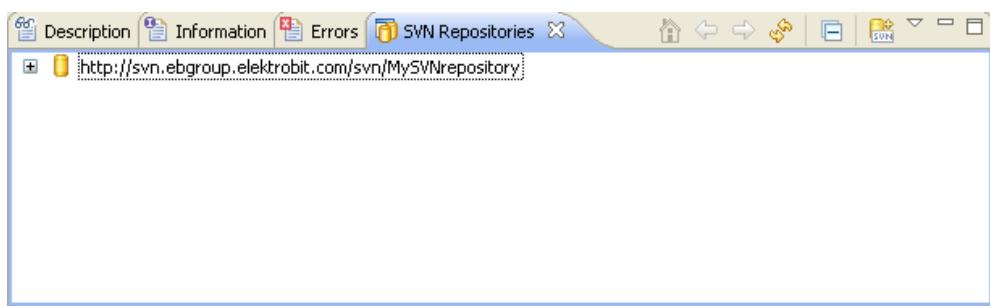


Figure 7.53. Registered repositories in the *SVN Repositories* view

7.4.3.2. Sharing an EB tresos Studio project with other team members

In this section you will learn how to share a project with other team members. This is equivalent to adding new files to an SVN repository and thus only applies to projects that have not yet been committed to the repository.

NOTE**You can only share whole projects, not single files**

Adding single files to a repository is not possible from within EB tresos Studio. Thus, if you want to share new files, you must share the project they reside in first.

To share a new project:

- ▶ Right-click this project in the *Project Explorer* view, select *Team* and then *Share Project...* from the context menus that open up:

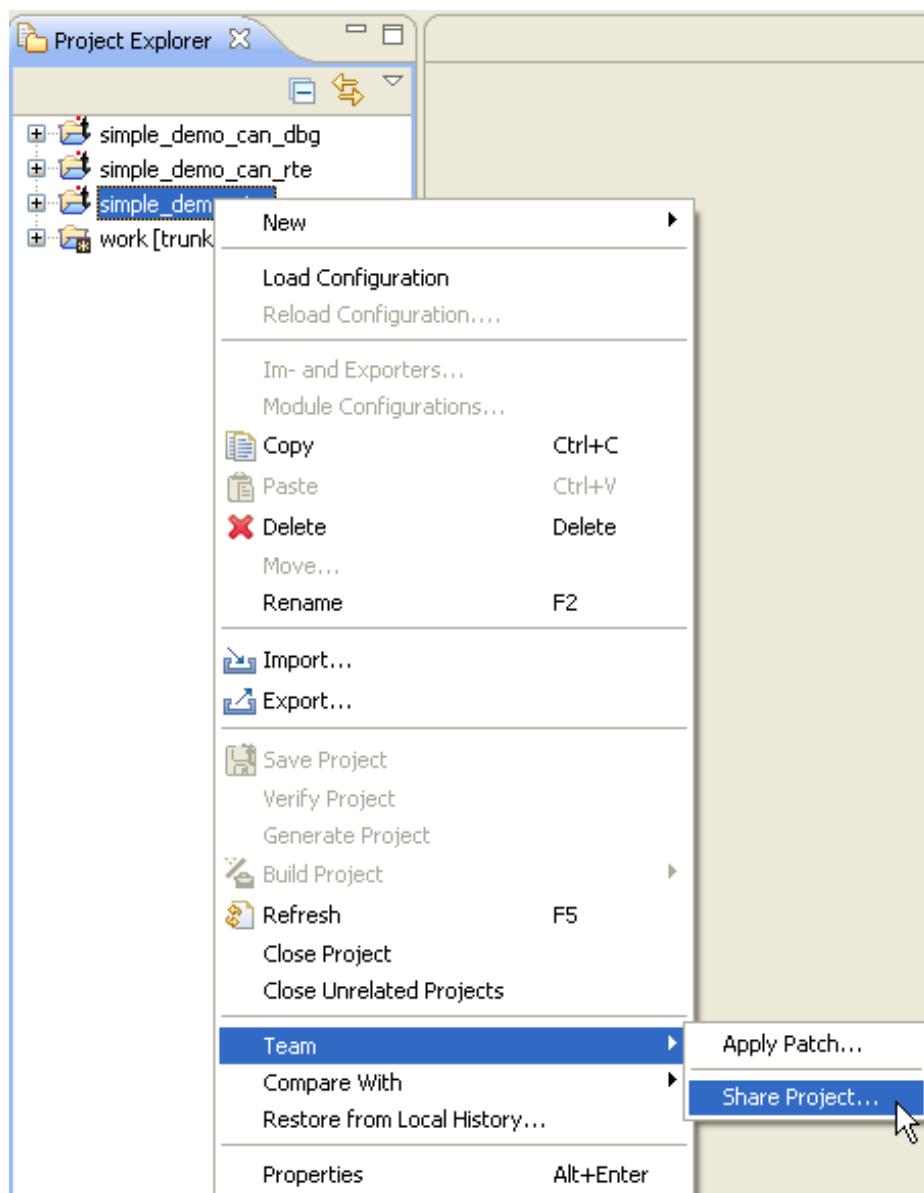


Figure 7.54. Selecting a project to share

- ▶ The *Share Project* dialog opens up.

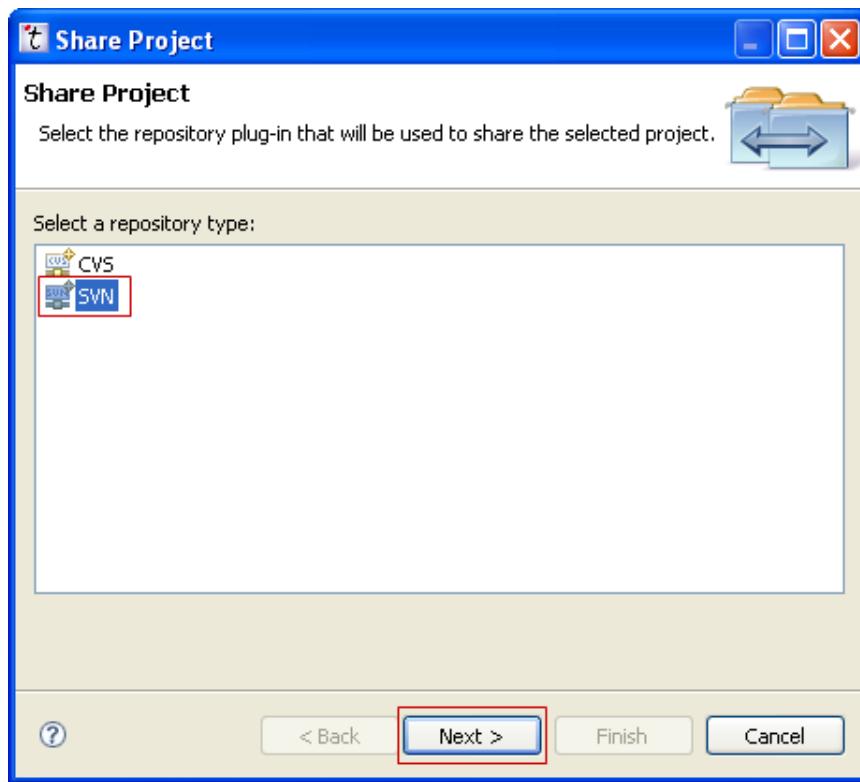


Figure 7.55. Selecting the repository type in which you want to share your project

- ▶ Select SVN and click **Next**.

The *Share Project with SVN Repository* page opens up

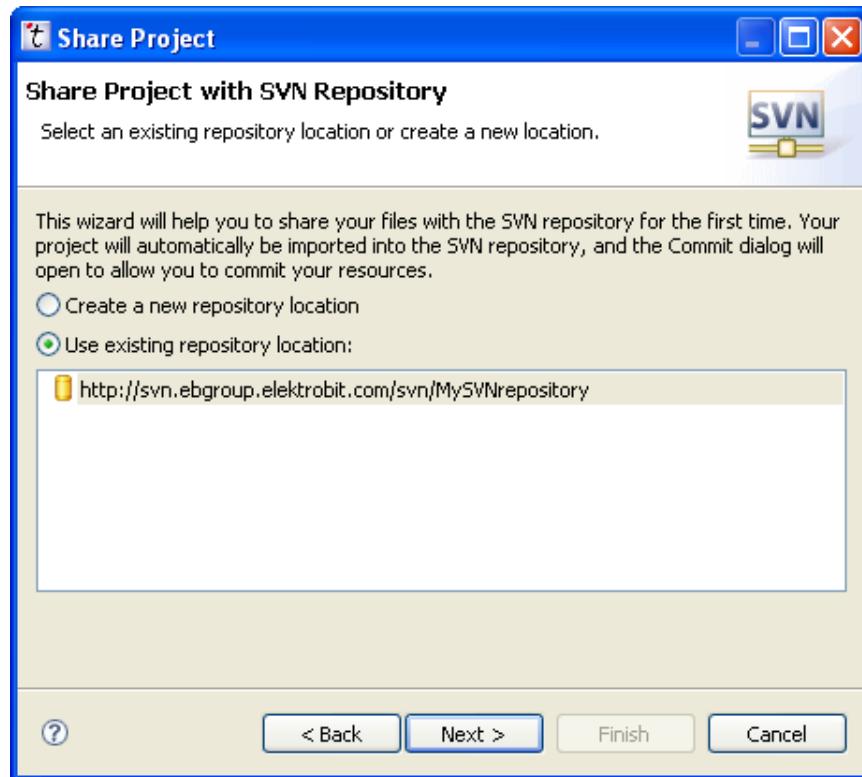


Figure 7.56. Selecting the repository location in which you want to share your project

- ▶ Select the repository you want to add your project to. You can either select an already registered repository location or register a new repository location.
- ▶ Click **Next**. The *Enter Folder Name* page opens up.

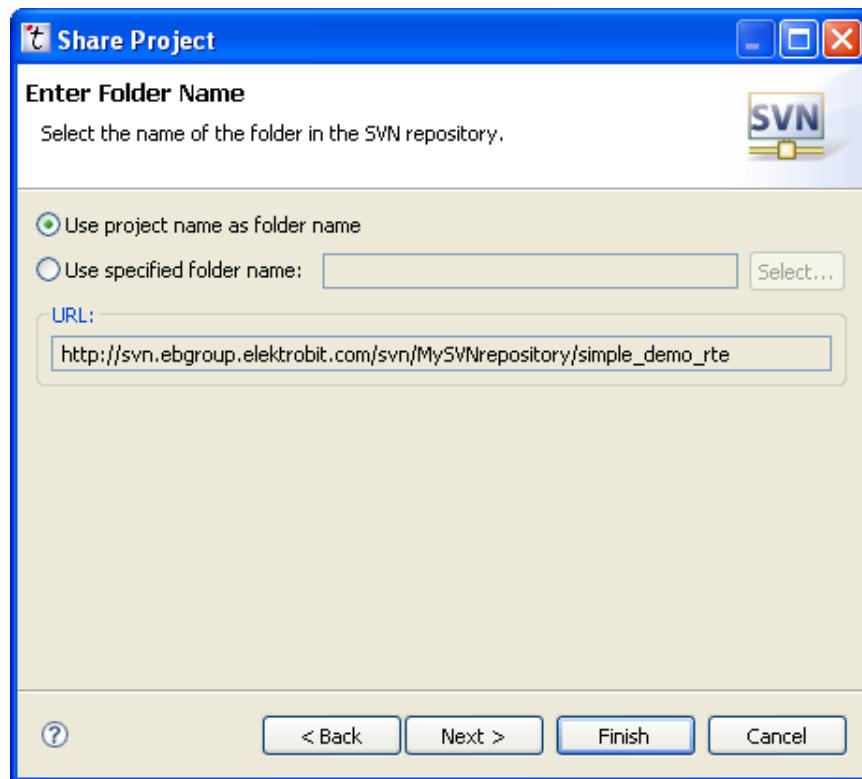


Figure 7.57. Entering a folder name for your project in the repository

- ▶ Enter the folder name in which you want to add your project to your SVN repository and click **Next**. The *Ready to Share Project* page opens up.

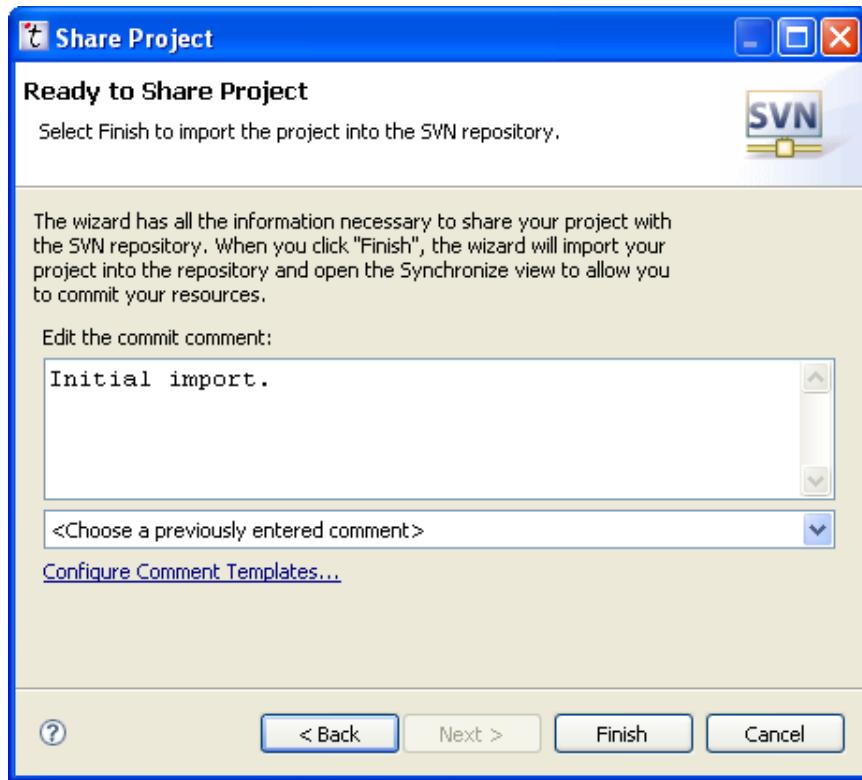


Figure 7.58. Entering a commit comment to your newly added project

- ▶ Enter a commit comment.
- ▶ To commit the added project to your repository, click **Finish**.

Your project is added and committed to your selected repository. After you have confirmed a confirmation dialog, the *Team Synchronizing* perspective opens up and displays the shared project. For information on the *Team Synchronizing* perspective, refer to [Section 7.4.4.2.1, “The Team Synchronizing perspective”](#).

7.4.3.3. Checking out a shared project from your SVN repository into your EB tresos Studio workspace

In this section you will learn how to check out an SVN-managed project into your EB tresos Studio workspace. You can check out a project from an SVN repository into your EB tresos Studio workspace directly from the EB tresos Studio GUI.

To check out an SVN-managed project into your EB tresos Studio workspace, you need to open the *Checkout from SVN* dialog first. To open the *Checkout from SVN* dialog:

- ▶ Right-click the registered repository in the *SVN Repositories* view, and select *Checkout...*. The *Checkout from SVN* dialog opens up.

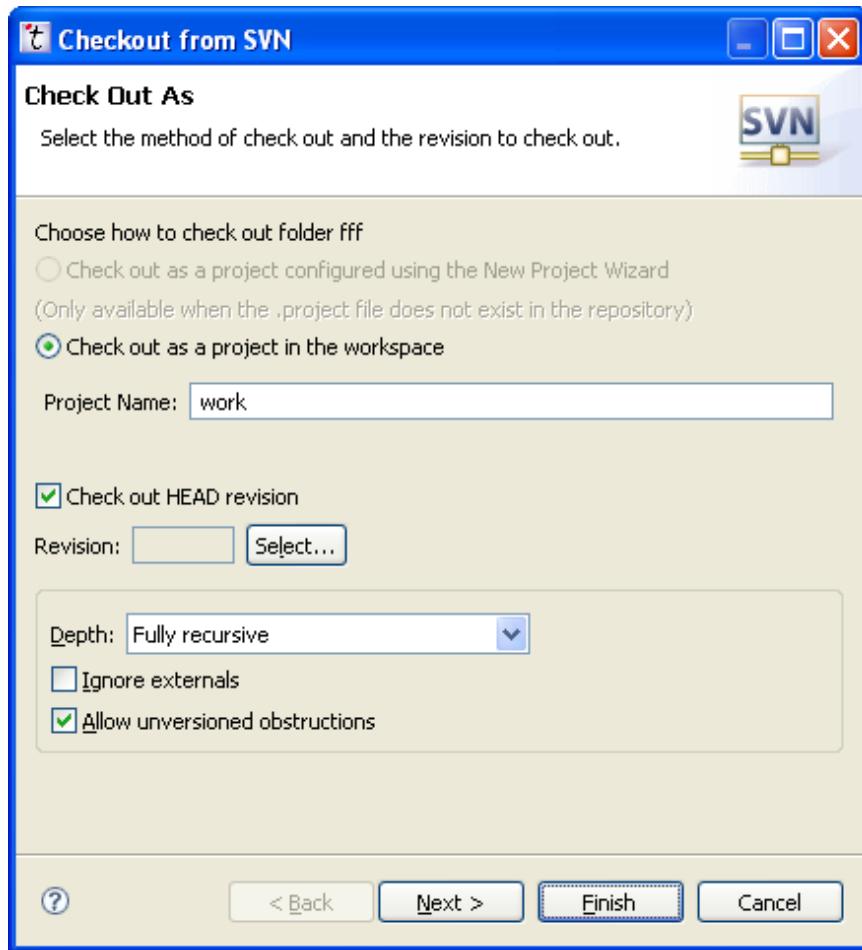


Figure 7.59. Checking out the SVN repository

- ▶ In the *Check Out As* page, you can specify how to check out the chosen SVN module:
 - ▶ If the selected SVN folder already contains the project infrastructure file `.project`, select **Check out as project in the workspace**.
 - ▶ If the chosen module does not yet contain the project infrastructure file `.project`, select **Check out as project configured using the New Project Wizard** and click **Next**. The *New Project Wizard* opens up.

Follow the *New Project Wizard*. For instructions on creating a new project with the *New Project Wizard*, refer to [Section 7.3.1.2, “Step 2: Creating a new project”](#).
 - ▶ If you do not want to check out the HEAD revision, clear the *Check out HEAD revision* option and enter the specific revision you want to check out.
 - ▶ If you do not want to use the default workspace location, click **Next** to proceed to another wizard page, where you can specify where to checkout the files to.
- ▶ Click **Finish**.

The selected files are being checked out to the indicated location. As a result you see the checked out project in your *Project Explorer* view, displayed with an overlay icon:

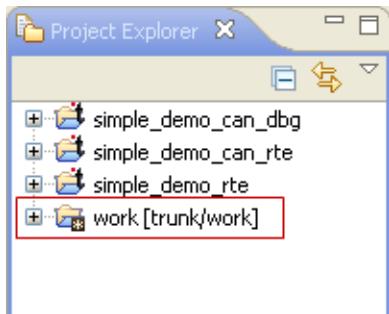


Figure 7.60. Checked out project in your *Project Explorer* view

For information on the meaning of the overlay icons, refer to [Project Explorer overlay icons for SVN repository support](#).

7.4.3.4. Updating, branching, and committing EB tresos Studio projects into your SVN repository

After you have registered and checked out your repository into EB tresos Studio, you may want to edit the files and commit your work to the repository.

Your files and folders are displayed with different overlay icons in the *Project Explorer* view. For information on the meaning of these overlay icons, refer to [Project Explorer overlay icons for SVN repository support](#).

To commit, update, branch your work to a repository or to choose another typical task for working with SVN repositories:

- ▶ Right-click the project in the *Project Explorer* view. A context menu opens up.
- ▶ In the context menu, select *Team*.

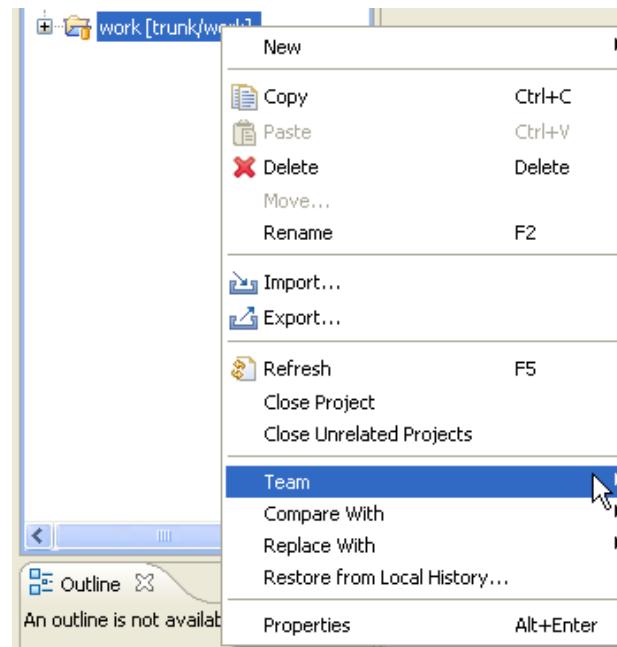


Figure 7.61. Opening the SVN context menu

- The SVN context menu opens up.

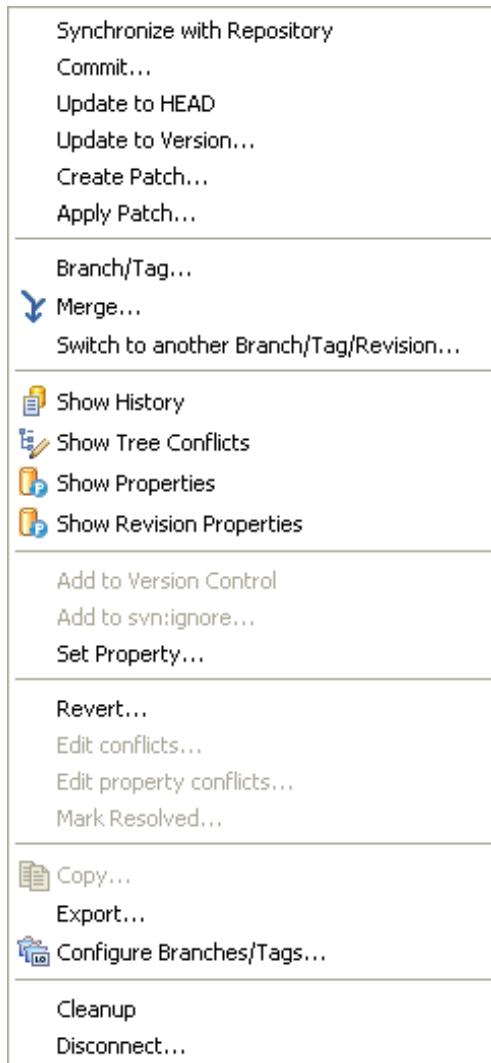


Figure 7.62. The SVN context menu

In the SVN context menu you can select the tasks typical for working with SVN repositories, such as committing, updating, branching, tagging the repository etc.

WARNING**Reload projects after updating**

It is strongly recommended to only update files from SVN, when the configuration project is not loaded yet. After you have updated files in loaded configuration projects, the project needs to be reloaded. For further information on reloading the project, refer to [Section 7.3.6.4, “Reloading a project”](#).

If you want to diff files and file versions, refer to [Section 7.5, “Differing files and file versions”](#).

7.4.4. Synchronizing your project with your repository

7.4.4.1. Overview

If you share your project with a team in a repository such as SVN or CVS, you might want to synchronize your work with the repository. Synchronizing means:

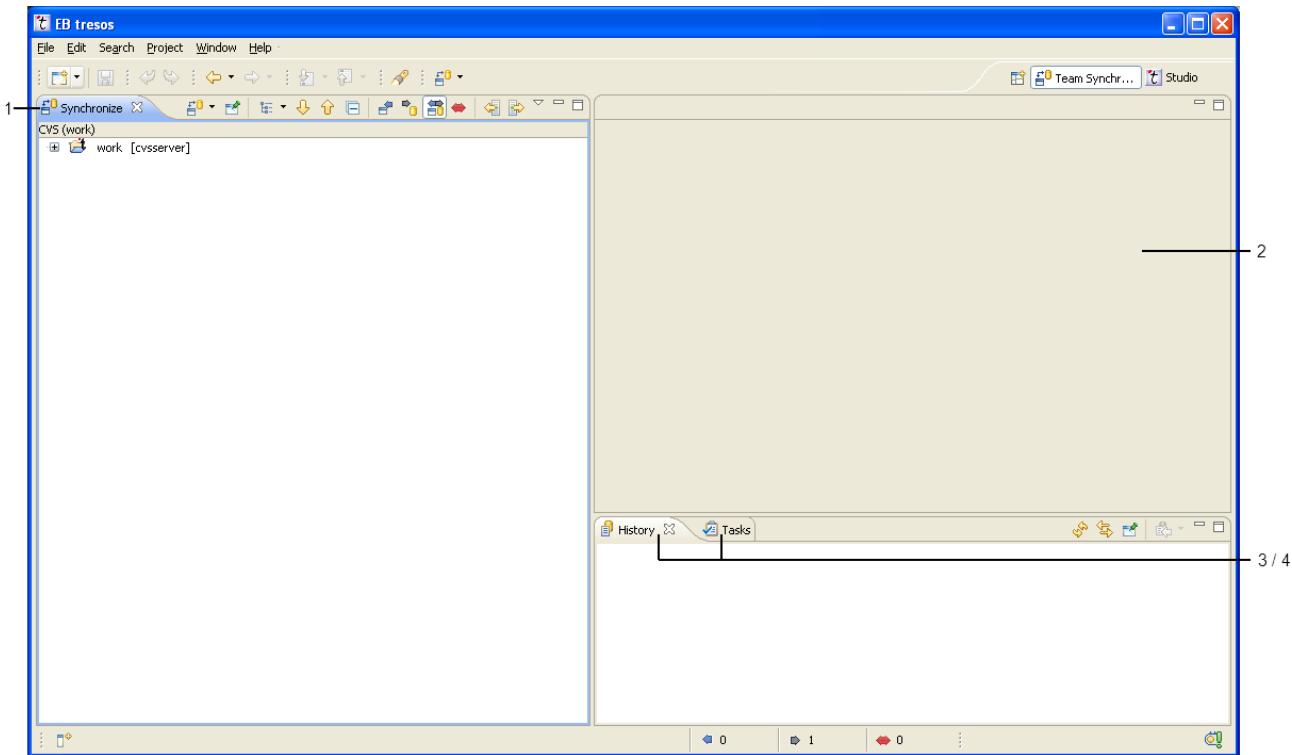
- ▶ viewing differences between your working copy and the repository,
- ▶ updating your project in EB tresos Studio to the latest repository version,
- ▶ committing your changes to the repository,
- ▶ committing and simultaneously updating your project to/with the repository.

7.4.4.2. Background information

To synchronize your project with repositories, EB tresos Studio provides a whole perspective in which differences between your local files and the repository are displayed. This makes synchronizing easier for you. Information on how to open the *Team Synchronizing* perspective is available in [Section 7.4.4.3, “Opening the Team Synchronizing perspective”](#). In this section you find basic information on the *Team Synchronizing* perspective and on its central view, the *Synchronize* view.

7.4.4.2.1. The *Team Synchronizing* perspective

The *Team Synchronizing* perspective consists of the following views:

Figure 7.63. The *Team Synchronizing* perspective

Number	Element	Description
1	Synchronize view	The <i>Synchronize</i> view displays differences between your working copy and the repository of a project. For a detailed description of the <i>Synchronize</i> view, refer to Section 7.4.4.2.2, “The Synchronize view” .
2	editor area	In the editor area, editors such as the <i>Compare</i> editor for text files or for XDM files open up. With these editors you can compare different versions or files with each other.
3	History view	In the <i>History</i> view, you can open the check-in history of a file, folder or project that is connected with a repository.
4	Tasks view	The <i>Tasks</i> view helps you to organize your work. You can add short task descriptions together with additional attributes such as a priority.

7.4.4.2.2. The *Synchronize* view

The *Synchronize* view is the central GUI element with which you may synchronize your files with a repository. It displays differences between selected workspace resources and the repository. The *Synchronize* view is

available on the left side of the *Team Synchronizing* perspective. For instructions on opening the *Team Synchronizing* perspective, refer to [Section 7.4.4.3, “Opening the Team Synchronizing perspective”](#).

The *Synchronize* view consists of the following two components:

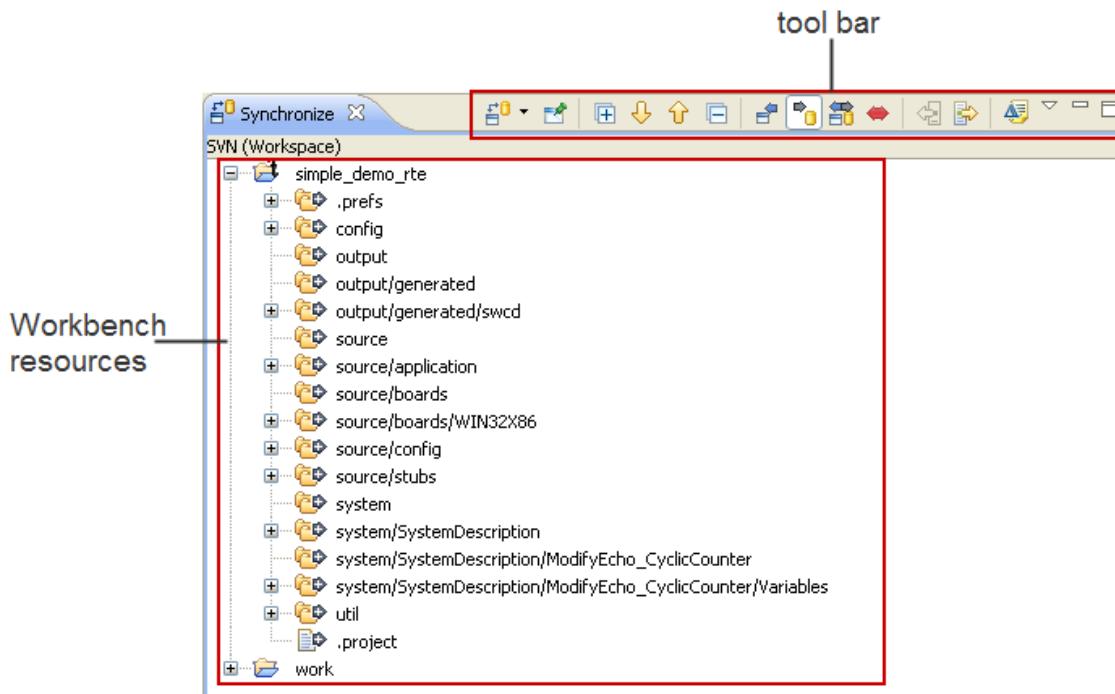


Figure 7.64. The *Synchronize* view

In the *Workbench resources* section of the *Synchronize* view, selected resources from your workbench are displayed. These workbench resources are displayed with overlay icons that indicate differences between the repository from which you have checked out these resources and your working copy. You find a list of these overlay icons with a description of their meanings at [Section 7.4.4.5, “Synchronizing your working copy with changes in the repository”](#).

The tool bar contains the following buttons:



Figure 7.65. The *Synchronize* view tool bar

Button	Name of the button	Description
	Synchronize	With this button you open a context menu in which you may choose which former set of resources you want to synchronize. From this menu, you can also open a wizard for creating a new set of resources to synchronize.

Button	Name of the button	Description
	Pin Current Synchronization	With this button you add a project to the context menu of the Synchronize button. A project that is pinned like this is automatically synchronized when the working copy changes.
	Expand All	With this button you expand all tree items displayed in the <i>Synchronize</i> view.
	Next Difference	With this button you jump to the next difference between the repository and your working copy in the resources, that are currently displayed in the <i>Synchronize</i> view.
	Previous Difference	With this button you jump to the previous difference between the repository and your working copy in the resources, that are currently displayed in the <i>Synchronize</i> view.
	Collapse All	With this button you collapse all tree items displayed in the <i>Synchronize</i> view.
	Incoming Mode	With this button you filter your synchronization for incoming changes from the repository. Click this button, if you want to update your local version to the head revision of the repository.
	Outgoing Mode	With this button you filter your synchronization for outgoing changes from your local version to the repository. Click this button, if you want to commit your local version to the repository.
	Incoming/Outgoing Mode	With this button you filter your synchronization for outgoing and incoming changes from your local version to the repository. Click this button, if you want to commit your local version to the repository and simultaneously update your local version to the head revision of the repository.
	Conflicts Mode	With this button you filter your synchronization for conflicts between your local copy and the repository. Click this button, if you want to find out which files you will have to merge before committing them to the repository.
	Update All Incoming Changes	With this button you update all files visible in the <i>Synchronize</i> view with non-conflicting incoming changes with the newer versions in your repository.
	Commit All Outgoing Changes	With this button you commit all non-conflicting local changes visible in the view to your repository.
	Show Change Sets (SVN repositories only)	With this button you can enable the support for change sets. When enabled, you can organize the content of the <i>Synchronize</i> view into change sets for clarity.

7.4.4.3. Opening the *Team Synchronizing* perspective

To synchronize workbench resources with a repository, open the *Synchronize* perspective:

- ▶ Either open the *Team Synchronizing* perspective via the *Project Explorer* view:
 - ▶ Select the resources you want to synchronize with the repository in the *Project Explorer* view.
 - ▶ Right-click the selection. A context menu opens up.
 - ▶ Select **Team**. The **Team Collaboration** context menu opens up.
 - ▶ In the **Team Collaboration** context menu, select **Synchronize with Repository**.

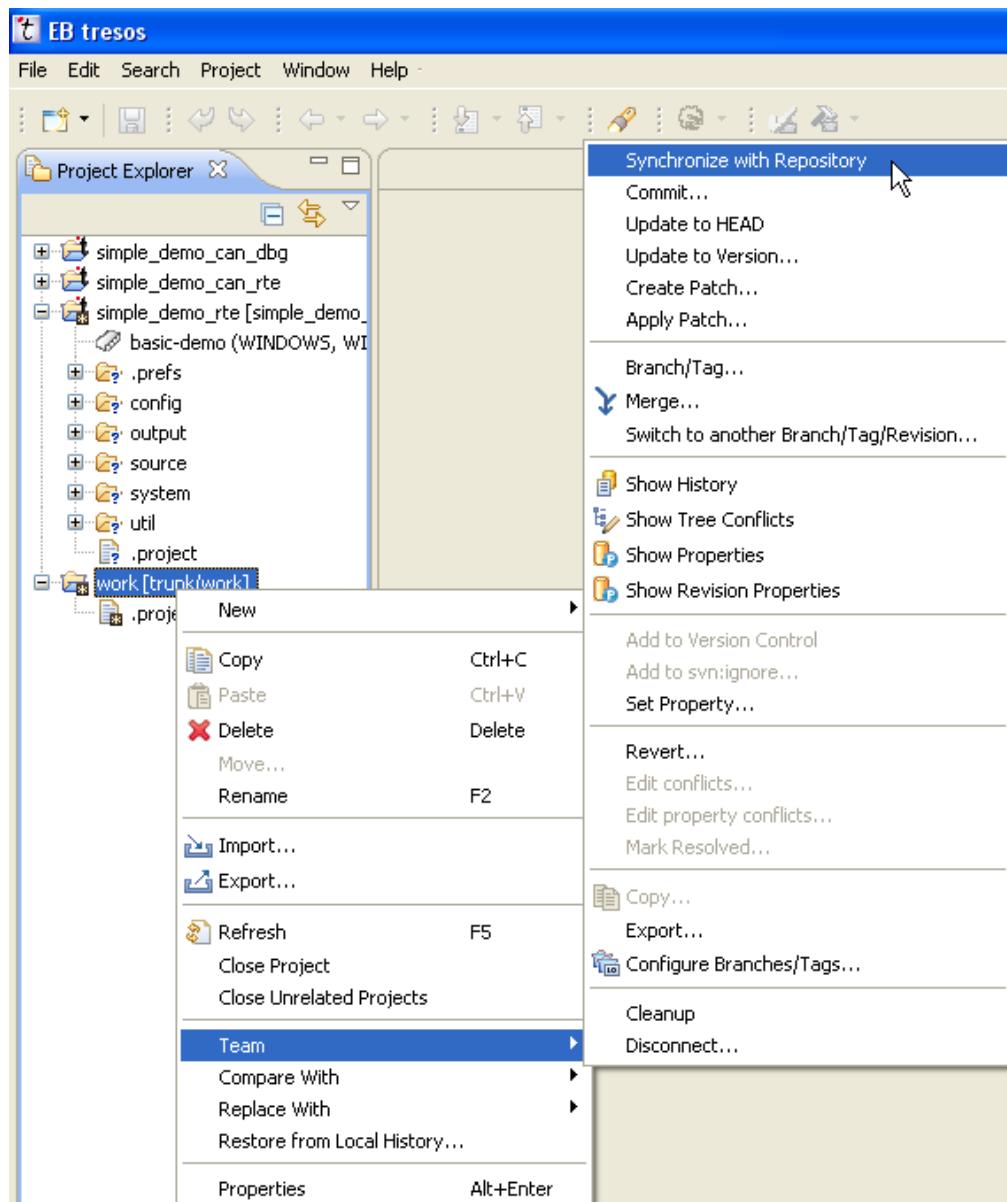


Figure 7.66. Opening the *Team Synchronizing* perspective

- ▶ Or select the *Team Synchronizing* perspective from the perspective bar:
 - ▶ In the perspective bar, click the **Open Perspective** button . A context menu opens up.

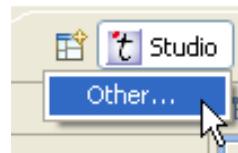


Figure 7.67. Perspective bar: Opening a new perspective

- ▶ Select *Other*. The *Open Perspective* dialog opens up.

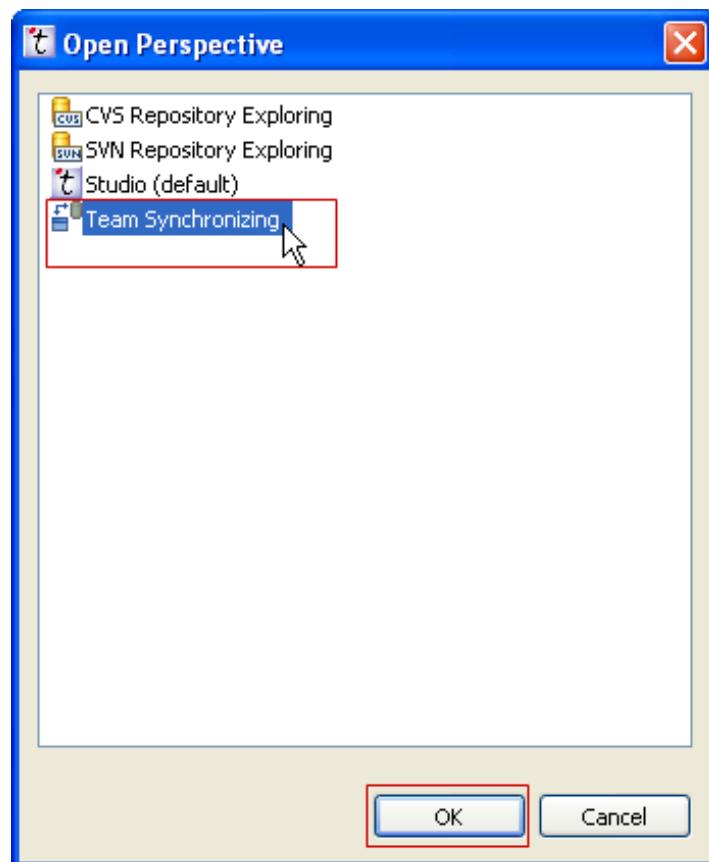


Figure 7.68. Opening the *Team Synchronizing* perspective via the perspective bar

- ▶ Select *Team Synchronizing* and click **OK**.

The *Team Synchronizing* perspective opens up. For a detailed description of the *Team Synchronizing* perspective, refer to [Section 7.4.4.3, “Opening the Team Synchronizing perspective”](#).

7.4.4.4. Viewing differences between your working copy and the base file in the repository

In the *Team Synchronizing* perspective you can view the differences between your working copy and the base file. The base file is the file you have checked out to edit as working copy. To find out how your working copy differs from the base file:

- ▶ In the *Synchronize* view, expand your project.
- ▶ Double-click the file you want to check for differences. A compare editor opens up in the editor area.

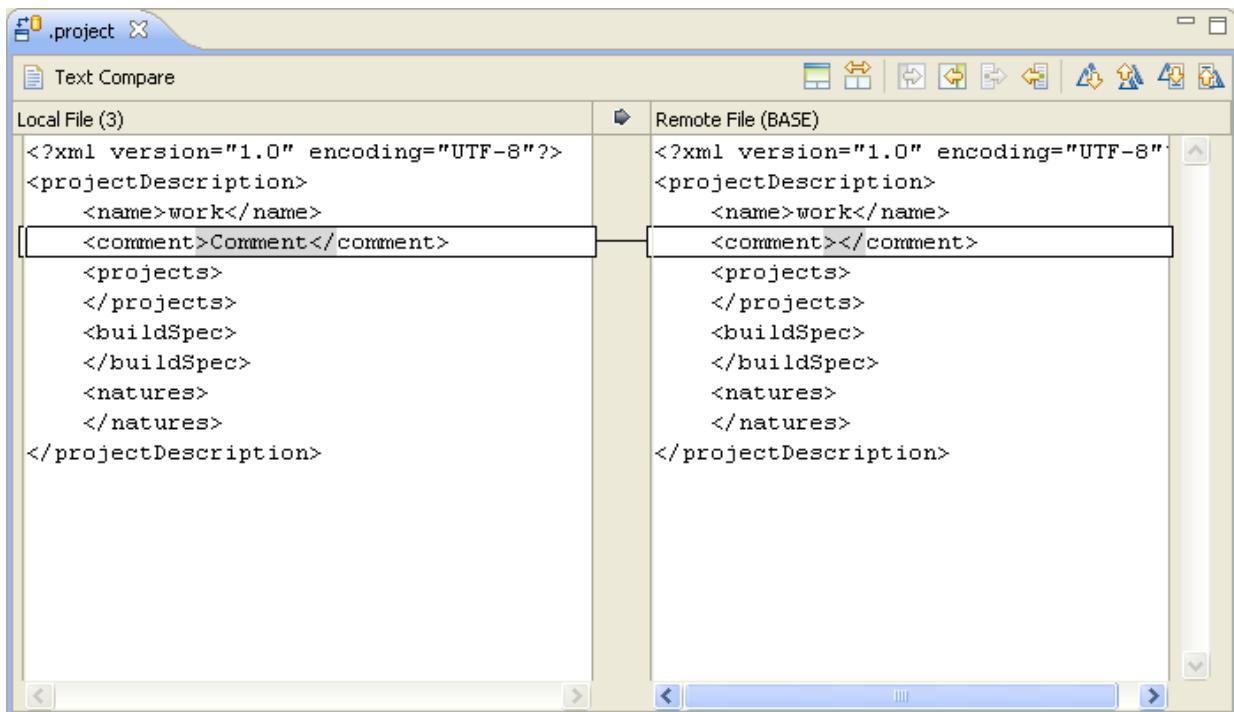


Figure 7.69. Differing with the *Text Compare* editor

- ▶ Which compare editor opens up depends on the file format you want to compare. For detailed information on the compare editors provided by EB tresos Studio, refer to [Section 7.5, “Differing files and file versions”](#).

7.4.4.5. Synchronizing your working copy with changes in the repository

The *Synchronize* view indicates differences between the repository and your working copy. These differences are displayed with the following overlay icons:



Outgoing change which indicates that the file is locally modified.



Outgoing change which indicates that the file has been added locally.



Outgoing change which indicates that the file has been removed locally.



Incoming change which indicates that there is a newer version available in the repository.



Incoming change which indicates that the file is new in the repository.



Incoming change which indicates that the file has been removed from the repository.



Conflicting change which indicates that the file has local modifications that conflict with changes that were checked in to the repository by another team member while you were editing the file.



Conflicting change which indicates that the file has been added locally, and conflicts with an equally named file, that has been added and committed to the repository by another team member.

There are three different ways of synchronizing your project with a repository:

- ▶ Synchronizing in *incoming mode*, i.e. updating your working copy to the head revision of the repository.
- ▶ Synchronizing in *outgoing mode*, i.e. committing your local changes to the repository.
- ▶ Synchronizing in *incoming/outgoing mode*, i.e. committing and updating your project simultaneously to the repository.

To synchronize your project with the repository:

1. In the tool bar of the *Synchronize* view, select the mode in which you want to synchronize your project with the repository:
 - ▶ To synchronize incoming changes (i.e. to update your project), click the **Incoming Mode** button . In the *Synchronize* view, new file versions are displayed that you have not yet updated from the repository.
 - ▶ To synchronize outgoing changes (i.e. to commit your project to the repository), click the **Outgoing Mode** button . In the *Synchronize* view, local changes are displayed that you have not yet committed to the repository.
 - ▶ To synchronize incoming and outgoing changes simultaneously (i.e. to commit your project to the repository and simultaneously update your project), click the **Incoming/Outgoing Mode** button . In the *Synchronize* view, local changes as well as new file versions in the repository are displayed.
2. In the *Synchronize* view, right-click the file, folder or project you want to synchronize. A context menu opens up.

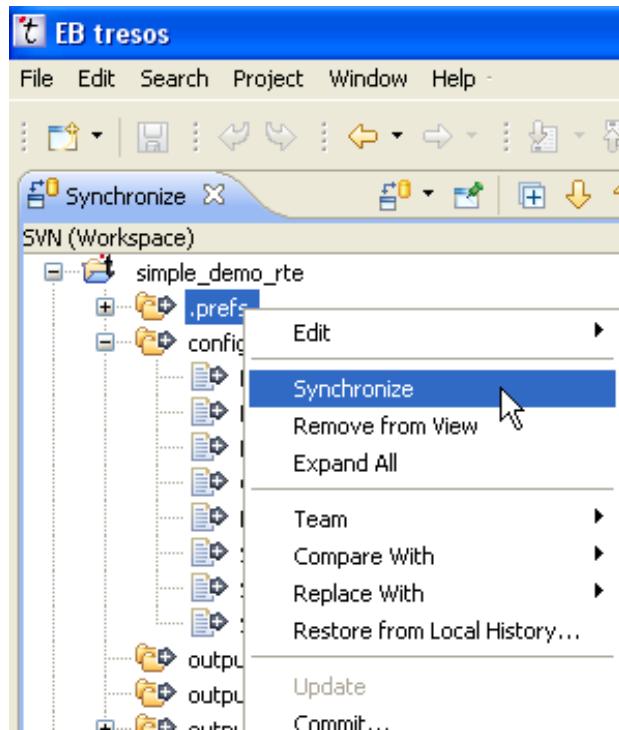


Figure 7.70. Synchronizing incoming changes

3. Select *Synchronize*.

Your project is being synchronized with the repository.

7.5. Differing files and file versions

7.5.1. Overview

EB tresos Studio provides the possibility to compare files or file versions with each other. This is helpful if you work in a team, but also if you want to compare local files with each other. EB tresos Studio provides two different compare editors (differs) for comparing files, depending on which kinds of files you want to compare:

- ▶ The Eclipse standard *Compare* editor for text files. For detailed information on this compare editor, refer to the [Eclipse Workbench User Guide](#), which is also included in the EB tresos Studio onscreen help.
- ▶ The *Compare* editor for XDM files, described in [Section 7.5.2, “Differing XDM files”](#).

7.5.2. Differing XDM files

7.5.2.1. Overview

In this chapter you will learn how to compare XDM files using the *Compare* editor for XDM files. While comparing, you can also merge differences into one of the compared XDM files.

7.5.2.2. Background information

In EB tresos Studio many parameters are stored in XDM files within the project, e.g. the parameters of module configurations, or project-specific settings such as the AUTOSAR release version or the target and derivative settings.

Especially (but not limited to) when working in teams on module configurations, the need arises to view and probably merge differences between XDM files, either from different configuration projects or when comparing one file version to another using some version control system, e.g. CVS or SVN.

To compare XDM files, EB tresos Studio provides the *Compare* editor for XDM files.

7.5.2.3. Opening the compare editor

To compare two XDM files with each other:

- ▶ Select both XDM files in the *Project Explorer* view.
- ▶ Right-click one of the two files. A context menu opens up.
- ▶ Select *Compare With*. Another context menu opens up.
- ▶ In this context menu, select *Each other*.

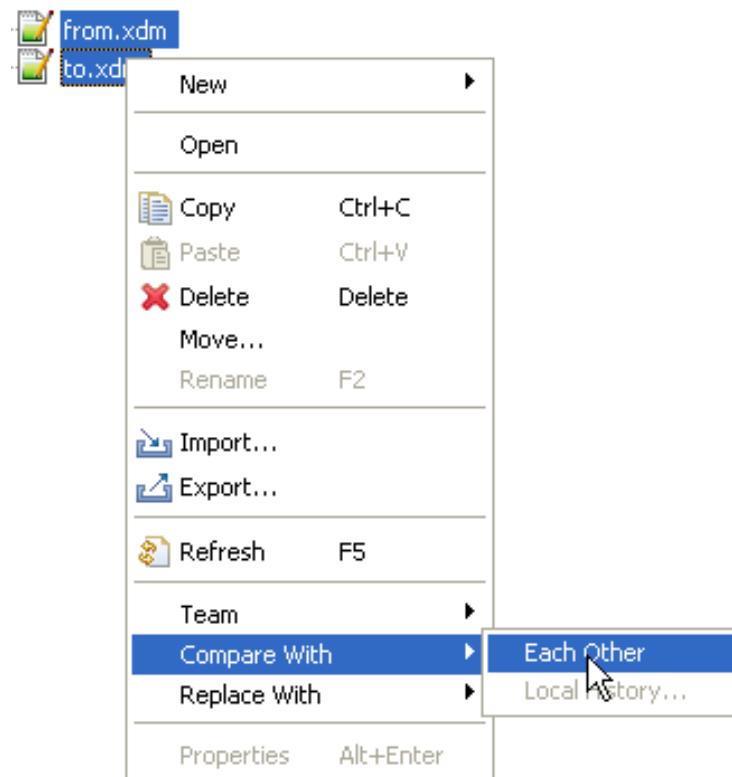


Figure 7.71. Directly comparing two XDM files

NOTE**Further ways to open the Compare editor for XDM files**

Apart from opening the compare editor from the *Project Explorer* view, there are further ways of opening this compare editor:

- ▶ Comparing different revisions from within the *History* view. For further information on the *History* view, refer to the [Eclipse Workbench User Guide](#). The Eclipse Workbench User Guide is also included in the EB tresos Studio onscreen help, which opens up when you press **F1**.
- ▶ Using the team *Synchronize* view in the *Team Synchronizing* perspective. For further information on the *Team Synchronizing* perspective, refer to [Section 7.4.4.2.1, “The Team Synchronizing perspective”](#).

-
- ▶ A progress indicator dialog opens up.
 - ▶ If both files are editable, another dialog opens up in which you may select into which file you want to merge the differences.

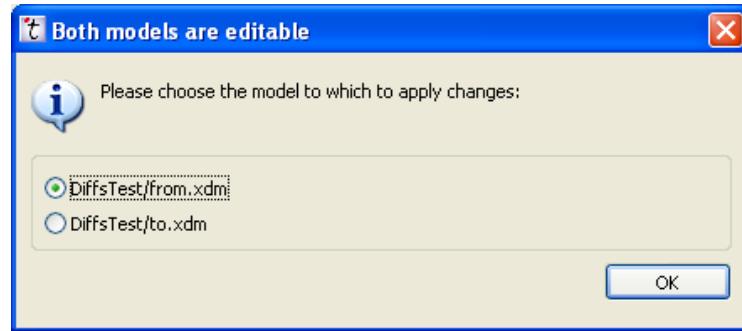


Figure 7.72. Selecting the XDM file to merge into

NOTE**Read-only mode**

 If none of the files are editable, the compare editor opens up in a read-only mode. You cannot merge differences then.

- ▶ The *Compare* editor for XDM files opens up in the editor area.

7.5.2.3.1. The Compare editor for XDM files

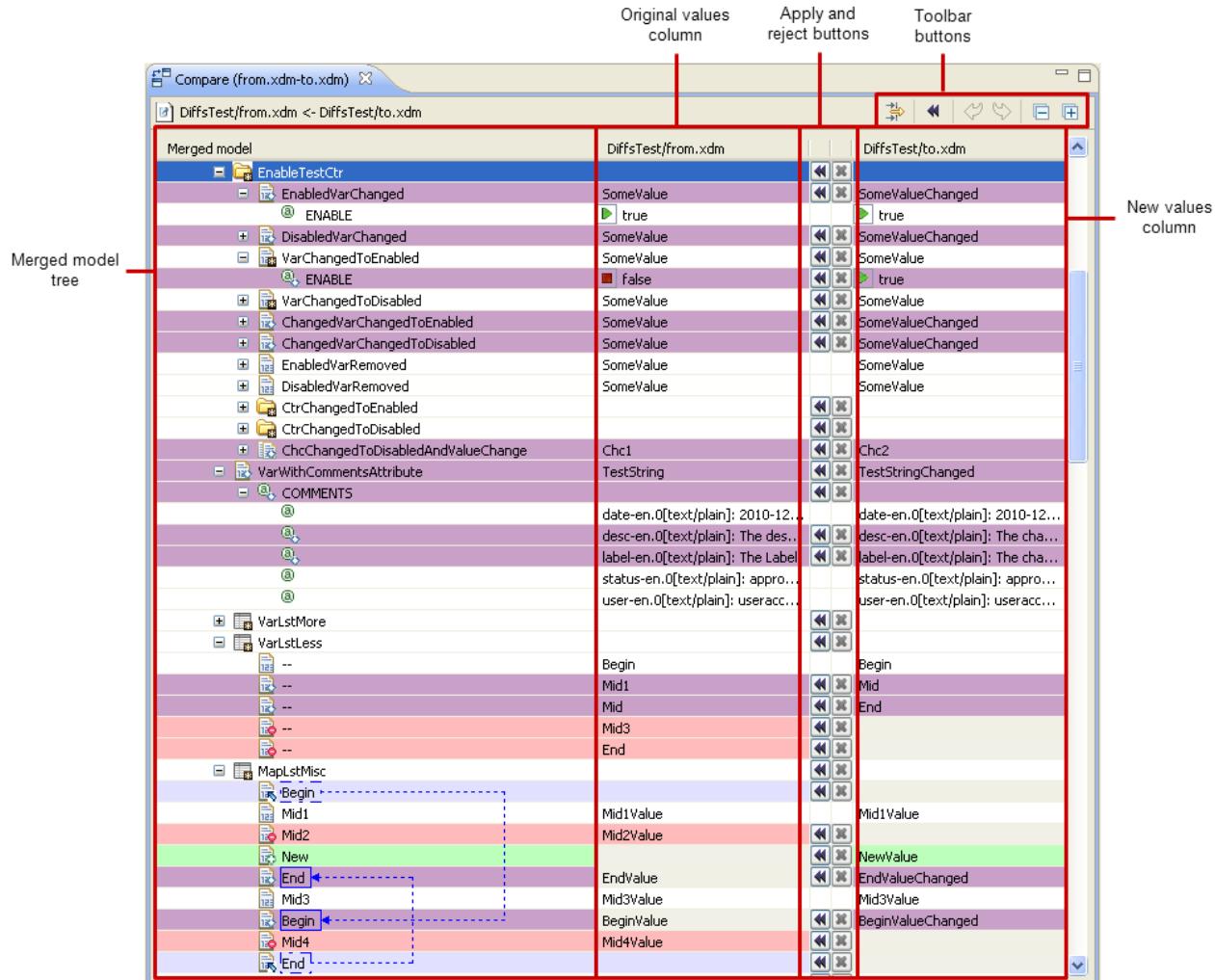


Figure 7.73. Compare editor for XDM files

Name	Explanation
Merged model tree	The left column called <i>Merged Model</i> shows the structure of the original XDM file. The original XDM file is taken as a reference schema. Differences are only made visible, if they fit into the schema of the original file. When a named parameter has been moved into another location within a list, the move is displayed by two entries with a dotted-line arrow in between.
Original values column	The column next to the merged model tree shows the values of the original model, to which the new model is compared to. Optional XDM nodes show their activation status with additional icons in front of the value:

Name	Explanation
	<ul style="list-style-type: none"> ▶  meaning enabled ▶  meaning disabled <p>Additionally, the value of disabled nodes are grayed out.</p>
Apply and reject buttons	The next two columns of the table contain the buttons for recursively applying  or rejecting  differences. In this context applying means to take over the changes from right to left. Rejecting means that the changes are not shown as a difference in this session any more. Mind that those two columns are only displayed, if at least one of the two compared files is editable.
New values column	<p>The column on the right shows the values of the new XDM file, which is being compared to the original XDM file.</p> <p>Optional XDM nodes are displayed in the same manner as in the original values column.</p> <p>You can select the font used in this column in the preferences settings as described in Section 7.2.3.8, “Setting colors and fonts”.</p>
Tool bar buttons	The tool bar of the <i>Compare</i> editor for XDM files provides further action buttons for filtering, applying all remaining differences, undo and redo support, and collapsing or expanding the tree.

Different background colors in the table represent the different types of differences. You can adjust those colors in the preferences settings as described in [Section 7.2.3.8, “Setting colors and fonts”](#).

The tool bar of the workbench contains two items for navigating to the next () and previous () difference. Those are bound to the keys **Ctrl + .** and **Ctrl + ,** by default.

A context menu is available for the items of the table, providing possible actions, like applying or rejecting all changes recursively or specific changes of the selected item, e.g.:



Figure 7.74. Context menu

When hovering over a changed item in the table, a tooltip appears, e.g.:

		4thStringValue	4thStringValue
Choice4	String		
EnableTestCtr	EnabledVarChange	Changed value: SomeValue --> SomeValueChanged	
	ENABLE	true	true
DisabledVarChanged		SomeValue	SomeValueChanged
VarChangedToEnabled	ENABLE	SomeValue	SomeValue
		false	true

Figure 7.75. Change tooltip

The status bar of the workbench displays two counters, one for the total number of differences, and another one for the number of unprocessed differences (i.e. applied or rejected).



Figure 7.76. Counters in the workbench status bar

7.5.2.3.2. Tree table icons and overlay icons

The icons used in the *Merged Model* tree column indicate the type of parameter. They are the same as the type icons used in the Generic Configuration Editor for parameters (see [Section 6.4.1.1, “Sections and parameters”](#)).

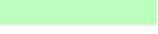
Differences in attributes are displayed with the icon in front of the changed tree item's child elements.

Overlay icons that are displayed over the type icon indicate the type of difference. The following list explains the meaning of all overlay icons, in ascending precedence, as only one overlay is displayed even if more changes exist for the item:

- ▶  Indicates a difference within the child elements of a tree item.
- ▶  Indicates a moved parameter value (the moved *to* position).
- ▶  Indicates a moved parameter value (the moved *from* position).
- ▶  Indicates an added parameter or attribute.
- ▶  Indicates a removed parameter or attribute.
- ▶  Indicates a changed value of a parameter or attribute.

7.5.2.4. Understanding the diffing results

Differences that are not yet processed (i.e. applied or rejected) between the compared datamodels are visualized by using overlay icons, different background colors, different values, etc. The following table describes the types of differences in detail:

Diff type	Overlay icon	Default background color	Explanation
Changed node value			The <i>Original values</i> column and the <i>New values</i> column display the different values.
Removed node			The <i>Original values</i> column displays the value of the removed node.
Added node			The <i>New values</i> column displays the value of the added node.
Moved node (from position)			The tree item displays the original position from where the tree item has been moved. This tree item neither displays any value nor any children nor any other difference. The tree item is surrounded by a frame (dotted line).
Moved node (to position)			The tree item displays the new position, to where the node has been moved. If the node also has a different value, it takes precedence for the overlay icon and background color. The tree item is surrounded by a frame (solid line). A blue arrow points from the original position of the node.

7.5.2.5. Navigating from difference to difference

In the *Compare* editor you can navigate from one difference to the next. This is useful if you want to compare large files that have few differences.

- ▶ To select the next item in the tree with an unprocessed difference, click the *Next Difference* button  in the workbench tool bar.
- ▶ To select the previous item in the tree with an unprocessed difference, click the *Previous Difference* button  in the workbench tool bar.

7.5.2.5.1. Filtering

In the *Compare* editor for XDM files you can filter the diffing results for processed and unprocessed differences. A difference is *processed*, if you have applied or rejected the difference. A difference is *unprocessed* if you have not applied or rejected this difference.

To filter all items that are already processed or do not have any differences:

- ▶ Click the *Only show unprocessed changes* toggle button once  in the tool bar of the compare editor.
- ▶ The button shows the following toggled state:  . All tree nodes that are already processed or do not have any differences become invisible.
- ▶ To disable the filtering, click the button once again.

7.5.2.5.2. Expanding and collapsing the tree

You can expand or collapse arbitrary parts of the *Merged model* tree:

- ▶ To expand the whole tree, click the *Expand all* button  in the tool bar of the compare editor.
- ▶ To collapse the whole tree, click the *Collapse all* button  in the tool bar of the compare editor.
- ▶ To expand one or more subtrees:
 - ▶ Select the tree items which are the parent elements of the subtrees you want to expand.
 - ▶ Right-click one of the selected tree items and select *Expand* from the context menu.
 - ▶ The selected tree items are expanded deeply, i.e. all child elements beneath the selected tree items become visible and expanded themselves.
- ▶ To collapse one or more subtrees:

- ▶ Select the tree items which are the parent elements of the subtrees you want to collapse.
- ▶ Right-click one of the selected tree items and select *Collapse* from the context menu.
- ▶ The selected tree items are collapsed deeply, i.e. all child elements beneath the selected tree items become invisible and collapsed themselves.
- ▶ To expand one level of a single tree item:
 - ▶ Click the expand button  next to the collapsed tree item you want to expand.
 - ▶ The tree item is expanded, i.e. its child elements become visible, which themselves may currently be collapsed or expanded.
- ▶ To collapse one level of a single tree item:
 - ▶ Click the collapse button  next to the expanded tree item you want to collapse.
 - ▶ The tree item is collapsed, i.e. its child elements become invisible. The child element's expansion state remains unchanged, though.

NOTE

Applying and rejecting actions do not consider the expansion state of an item



The **Apply** and **Reject** actions described in [the section called “Applying subtrees”](#) and [the section called “Rejecting subtrees”](#) do not consider the expansion state of any selected items, but always work on the complete subtrees.

7.5.2.5.3. Keyboard shortcuts

EB tresos Studio provides the following keyboard shortcuts when working with the XDM differ:

Keyboard shortcut	Task
Ctrl + .	Select the next difference.
Ctrl + ,	Select the previous difference.

7.5.2.6. Merging differences

In the *Compare* editor for XDM files you can directly merge the differences into one of the two compared files. Into which one of these two files these differences are merged you have to define when opening the compare editor. For instructions on how to do that refer to [Section 7.5.2.3, “Opening the compare editor”](#).

To merge a difference you have to either apply or reject this difference. After a difference has been applied or rejected, it is considered to be *processed*. Thus, an *unprocessed* difference is a difference that hasn't been applied or rejected yet.

NOTE**Save your file after merging to persist the change**

After you have applied any changes, the compare editor displays the dirty indicator (*) in the title tab *Comp. To make the changes persistent, you need to save the file.

WARNING**Reload projects after changing any XDM files**

It is strongly recommended to only merge XDM files, when the configuration project is not loaded yet. After you have merged and saved XDM files in loaded configuration projects, the project needs to be reloaded. For further information on reloading the project, refer to [Section 7.3.6.4, “Reloading a project”](#).

Otherwise all changes done by merging the XDM files will be lost, when you save the configuration project the next time.

7.5.2.6.1. Applying differences

You can merge differences from the right file to the left file by applying changes. You can either apply one difference at a time or you can apply all differences for one tree item with their child elements simultaneously.

Applying single differences

To apply one selected difference:

- ▶ Right-click the tree item with the difference you want to apply.

A context menu opens up:

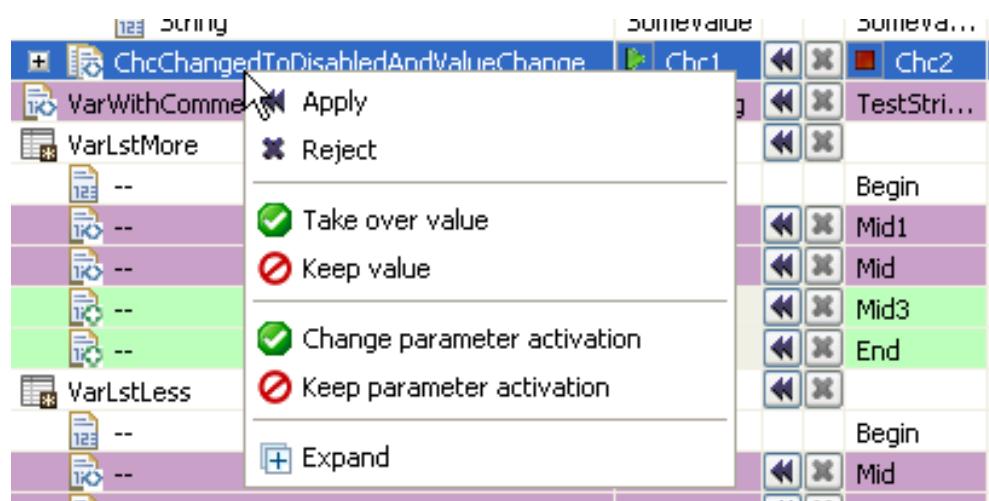


Figure 7.77. Applying one single difference

- ▶ Depending on the type of difference you want to apply, select the appropriate entry marked by the icon (i.e. select either *Take over value*, *Remove parameter*, *Add parameter*, *Change parameter activation* or *Move parameter*).
- ▶ The change is applied.

Applying subtrees

You can apply all changes of a tree item and all of its child elements in the tree simultaneously.

There are two alternatives to apply all changes of selected tree items recursively:

- ▶ Alternative 1:
 - ▶ Click the *Apply* button in the table for the tree items you want to apply.
- ▶ Alternative 2:
 - ▶ Select the tree items in the table for which you want to apply the changes.
 - ▶ Right-click one of these selected items and select *Apply* from the context menu.

The differences are taken over from the *Original values* column to the *New values* column and the processed tree items are collapsed.

Applying all remaining differences

You can also apply all unprocessed changes at once.

To apply all remaining differences:

- ▶ In the tool bar of the editor, click the button *Apply all unprocessed changes* .
- ▶ All remaining changes are applied to the left file.
- ▶ The counter for the unprocessed changes is set to zero.

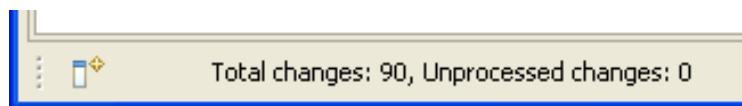


Figure 7.78. No more unprocessed changes

7.5.2.6.2. Rejecting changes

You can also reject changes. Rejected changes are not highlighted any longer in the compare editor and count as processed, both within the current compare session.

Rejecting single differences

To reject a selected difference:

- ▶ Right-click the tree item with the difference you want to reject.
- ▶ A context menu opens up:



Figure 7.79. Rejecting differences

- ▶ Depending on the type of difference you want to reject, select the appropriate entry marked by the icon (i.e. select either *Keep value*, *Keep parameter*, *Ignore parameter*, *Keep parameter activation* or *Do not move parameter*).
- ▶ The change is rejected.

Rejecting subtrees

You can reject all differences of a tree item and all of its child elements in the tree simultaneously.

There are two alternatives to reject all changes of selected tree items recursively:

- ▶ Alternative 1:
 - ▶ In the *Apply and reject buttons* column, click the button **Reject** next to the tree items you want to reject.
- ▶ Alternative 2:
 - ▶ Select the tree items you want to reject.
 - ▶ Right-click one of the selected items and select *Reject* from the context menu.

The processed nodes are rejected and become collapsed.

7.5.2.6.3. Undoing/Redoing operations

You can successively undo or redo the latest apply or reject operations.

- ▶ To undo the latest apply or reject operation, click the *Undo* button  in the tool bar of the compare editor.
- ▶ To redo the latest undo operation, click the *Redo* button  in the tool bar of the compare editor.

NOTE**Only operations on unsaved files can be undone**

Operations can only be undone as long as you do not save the changed file.



7.5.2.7. Limitations

The *Compare* editor for XDM files has the following limitations you need to be aware of.

7.5.2.7.1. Ignored differences

Not all possible datamodel differences are being supported by the compare editor.

7.5.2.7.2. Schema support

The algorithm used for calculating the differences has no explicit schema support, but takes the structure of the original file (the left side) as a reference. Any nodes within the compared file, that do not fit into that structure, are silently ignored.

This may even go as far as not showing any differences at all, when comparing files that have a completely incompatible structure. E.g. when you compare a module config XDM file with the `.prefs/preferences.xdm` file of a configuration project, no differences are displayed.

7.6. Editing module configurations

This chapter deals with the management of module configurations of an ECU configuration project. In this chapter you learn how to add or remove module configurations, where to view detailed information about module configurations, how to adjust certain module configuration details, how to change the default settings for the pre- and recommended configurations, how to upgrade the module configurations.

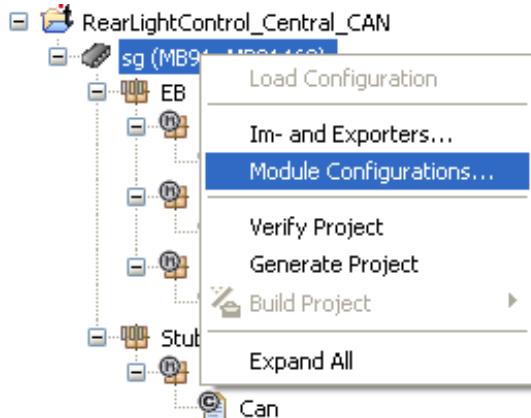
TIP**When to edit module configurations**

You can edit the set of the module configurations of an ECU configuration project in different situations:

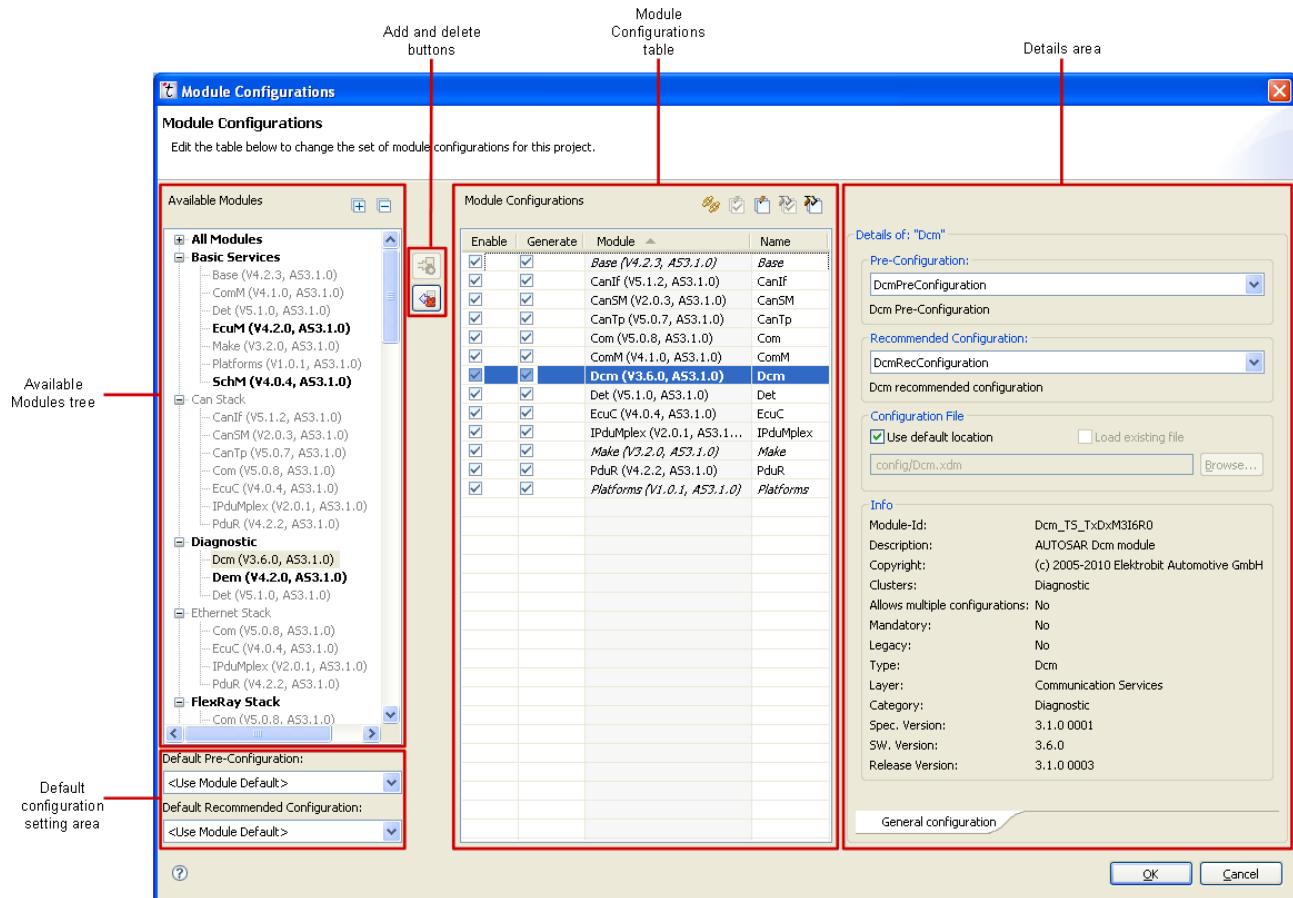
- ▶ During the creation of a new ECU configuration project using the *New Project Wizard*. For instructions refer to [Section 7.3.1.4, “Step 4: Defining module configurations”](#).
- ▶ For an existing project in the *Properties* dialog. For instructions refer to [Section 7.3.7.4, “Viewing information about the project's modules”](#).
- ▶ For an existing project in the *Module Configurations* dialog, as described below.

To edit the set of module configurations for a particular project, you need to open the *Module Configurations* dialog first:

- ▶ In the *Project Explorer*, right-click on the project or its ECU configuration node.
- ▶ Select **Module Configurations....**



The *Module Configurations* dialog opens up.

Figure 7.80. The *Module Configurations* dialog

The *Module Configurations* dialog consists of four areas:

- ▶ The *Available Modules* tree provides all modules which are available for the Target and Release Version of the project.
- ▶ In the *Default configuration settings* area, you may define the default pre- and recommended configuration for adding new module configurations.
- ▶ In the *Module Configurations* table, you see the module configurations that have been added to your project.
- ▶ The *Details* area provides detailed information either about a module selected within the *Available Modules* tree or a module configuration selected within the *Module Configurations* table. In the latter case it also provides edit functionality for newly added module configurations.

By pressing the **Ok**, the *Module Configurations* dialog closes and any changes are applied to the project.

By pressing **Cancel** the *Module Configurations* dialog closes without applying any changes.

WARNING **Changes are lost**

When canceling the dialog any changes you made are discarded without further notice.

7.6.1. Adding module configurations to your project

You may add single module configurations, several module configurations at once or clusters of module configurations to your project.

NOTE**You can recognize whether a module has been added by its font**

Modules that are still available and modules that already have been added to your project are displayed in different font styles in the *Available Modules* tree:

- ▶ Modules that have not been added to your project and thus are still available are displayed in bold font.
 - ▶ Modules that already have been added to your project are displayed in standard font. The modules that allow multiple configurations are displayed in black. All others are grayed-out and cannot be added to your project anymore.
-

To add new module configurations to your project:

- ▶ In the *Available Modules* tree, select all modules/clusters of modules for which you want to add configurations for.
 - ▶ Click the **Add** button .
-

TIP**Add single modules with a double-click**

Module configurations for single modules can also be added by double-clicking the required module.

- ▶ If the required module configurations cannot be added to your project, the following dialog opens up:

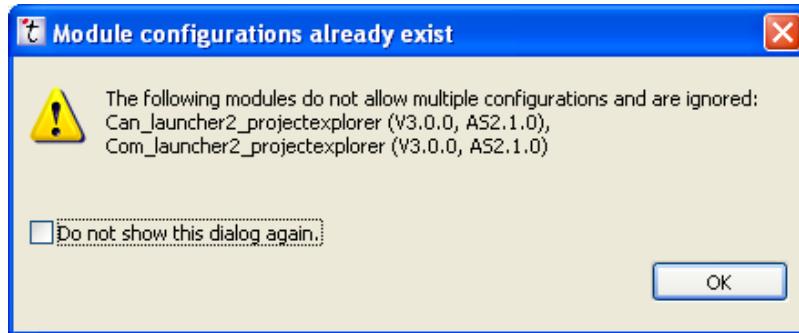


Figure 7.81. Some modules may be added only once to your project

The selected modules are added to your project as module configurations with default settings.

NOTE
Only one module configuration is added per selected module


Since modules can be part of more than one cluster, it is possible that a selection in the *Available Modules* tree refers several times to the same module. Note that only one configuration is added in this case, even if the module allows multiple configurations.

As long as the recently added module configurations have not been stored into your project yet, you may edit the (default) settings of the module configuration. Mind that depending on the module, there might be more tab pages next to the *General configuration* tab on the bottom of the *Details* area.

7.6.2. Removing module configurations from your project

You may remove single module configurations or several module configurations at once from your project.

NOTE

You can recognize the state of a module by its font

Module configurations that previously had been added to your project, mandatory module configurations and recently added module configurations are displayed in different fonts in the *Modules Configuration* table:

- ▶ Mandatory module configurations are displayed in italic font. It is not recommended to remove these module configurations from your project.
 - ▶ Recently added module configurations are displayed in bold font. You may edit them.
 - ▶ Module configurations that had been added to your project before are displayed in standard font.
-

To remove module configurations from your project:

- ▶ In the *Module Configurations* table, select the module configuration(s) you want to remove.

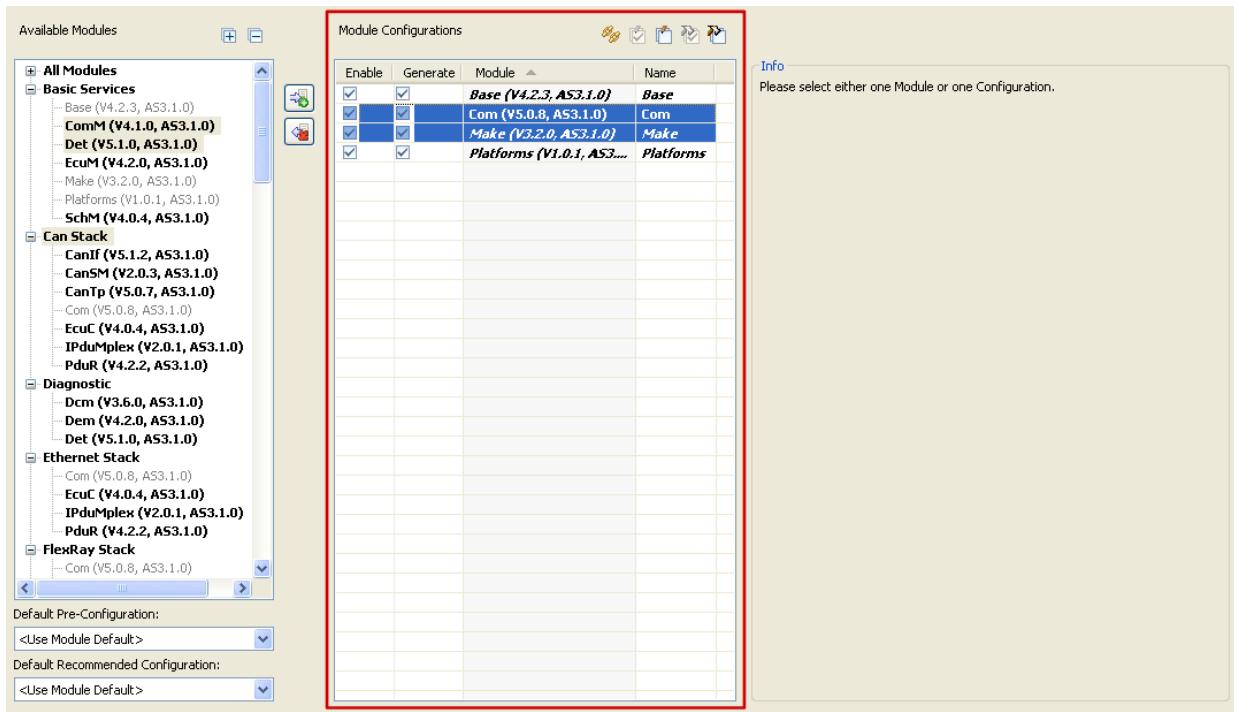


Figure 7.82. Removing module configurations

- ▶ Click the **Delete** button .
- ▶ If none of the selected module configurations refers to an already existing configuration file, the items selected are removed from the *Module Configurations* table without further notice. Otherwise the *Delete Module Configurations* dialog opens up

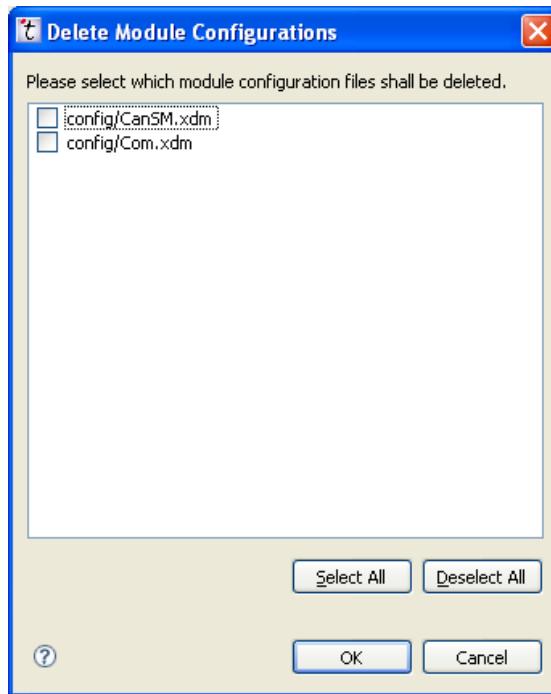


Figure 7.83. Confirming which module configuration files you want to remove

- ▶ Select the check box of the configuration files you want to delete.

NOTE**Data loss**

The selected files will be deleted from the file system when the changed module configurations are applied. After that, you cannot restore these files anymore.

-
- ▶ To confirm the deletion of the configuration files, click **OK**.
The items selected are removed from the *Module Configurations* table.
 - ▶ To cancel the deletion of the *Module Configurations* and their referred configuration files, click **Cancel**.

7.6.3. Enabling and disabling module configurations

In the *Module Configurations* table you may (temporarily) enable or disable the module configurations you have added to your project. Disabled configurations are treated as if they do not exist, which means you cannot edit or verify them, and code generators cannot access them. To enable module configurations:

- ▶ To enable only one module configuration, select the **Enable** check box of the module configuration you want to enable in the *Module Configurations* table.

Module Configurations			
Enable	Generate	Module	Name
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Base (V4.2.3, AS3.1.0)	Base
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CanSM (V2.0.2, AS3.1.0)	CanSM
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Com (V5.0.8, AS3.1.0)	Com
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Make (V3.2.0, AS3.1.0)	Make
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Platforms (V1.0.1, AS3....	Platforms

Figure 7.84. Enabling one module configuration

- ▶ To enable several module configuration at once:
 - ▶ In the *Module Configurations* table, select the module configurations you want to enable.
 - ▶ In the tool bar of the *Module Configurations* table, click the **Enable** button

Alternatively, you may enable module configurations in the *Project Explorer* view:

- ▶ To enable only one module configuration:
 - ▶ In the *Project Explorer* view, right-click on the module configuration you want to enable.
- A context menu opens up.

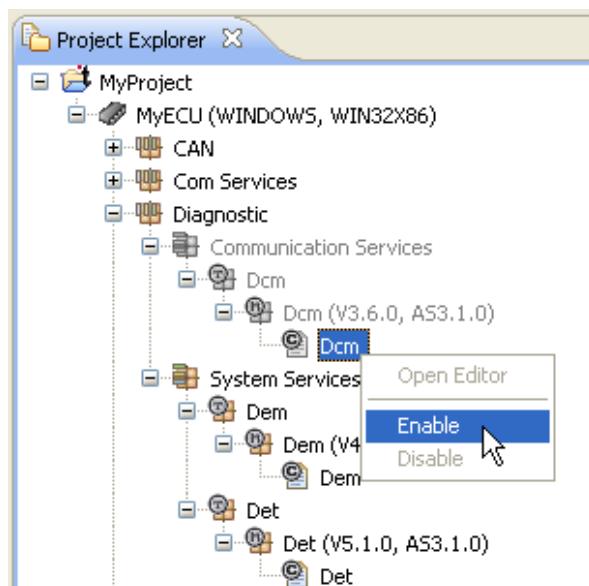


Figure 7.85. Enabling one module configuration

- ▶ In the context menu, select *Enable*.

The module configuration of the selected module is enabled and thus displayed in black standard font in the *Project Explorer* view.

- ▶ To enable all module configurations of one node:

- ▶ In the *Project Explorer* view, right-click on the category/layer/type/module node whose module configurations you want to enable.

A context menu opens up.

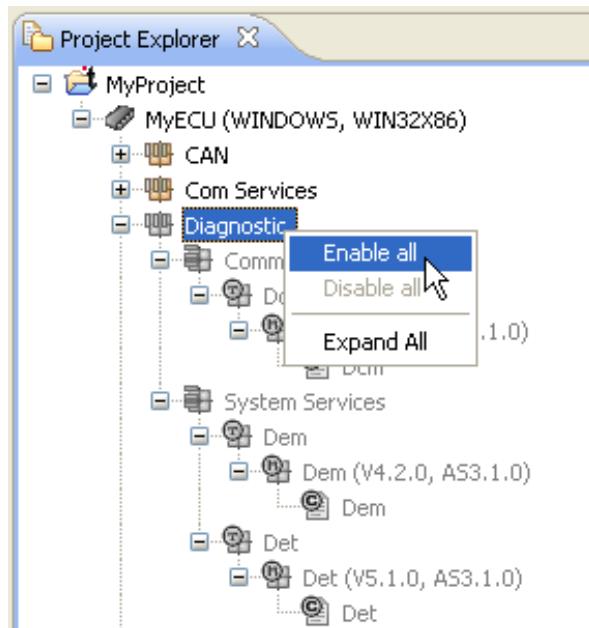


Figure 7.86. Enabling all module configurations of one node

- ▶ In the context menu, select *Enable all*.

NOTE**The *Enable all* command might not be available**

The *Enable all* command is only available if the selected node contains more than one module configuration and if one of these module configurations is disabled. If the node contains only one module configuration, the *Enable* command is displayed instead.

The module configurations of the selected node are enabled and are displayed in black standard font in the *Project Explorer* view.

To disable module configurations:

- ▶ To disable only one module configuration in the *Module Configurations* dialog/page, clear the **Enable** check box of the module configuration you want to disable in the *Module Configurations* table.

Module Configurations			
Enable	Generate	Module	Name
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Base (V4.2.3, AS3.1.0)	Base
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CanSM (V2.0.2, AS3.1.0)	CanSM
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Com (V5.0.8, AS3.1.0)	Com
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Make (V3.2.0, AS3.1.0)	Make
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Platforms (V1.0.1, AS3....	Platforms

Figure 7.87. Disabling one module configuration

To disable several module configurations at once:

- ▶ In the *Module Configurations* table, select the module configurations you want to disable.
- ▶ In the tool bar of the *Module Configurations* page, click the **Disable** button

Alternatively, you may disable module configurations in the *Project Explorer* view:

- ▶ To disable only one module configuration:
 - ▶ In the *Project Explorer* view, right-click on the module configuration you want to disable.

A context menu opens up.

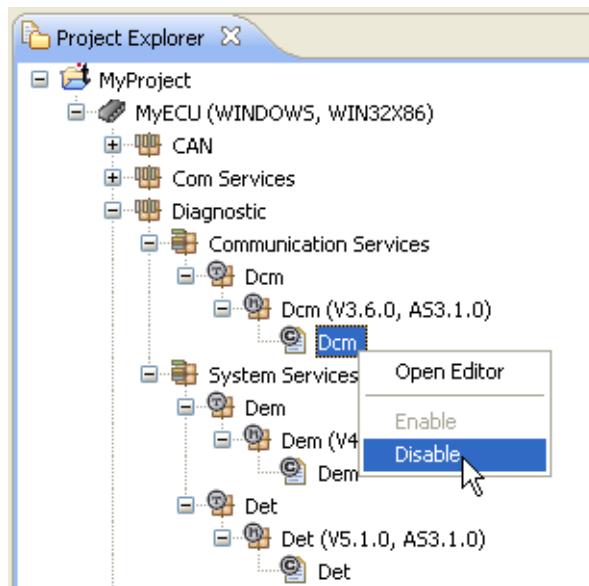


Figure 7.88. Disabling one module configuration

- ▶ In the context menu, select *Disable*.

The selected module configuration is now disabled and grayed out.

- ▶ To disable all module configurations of one node:

- ▶ In the *Project Explorer* view, right-click on the category/layer/type/module node whose module configurations you want to disable.

A context menu opens up.

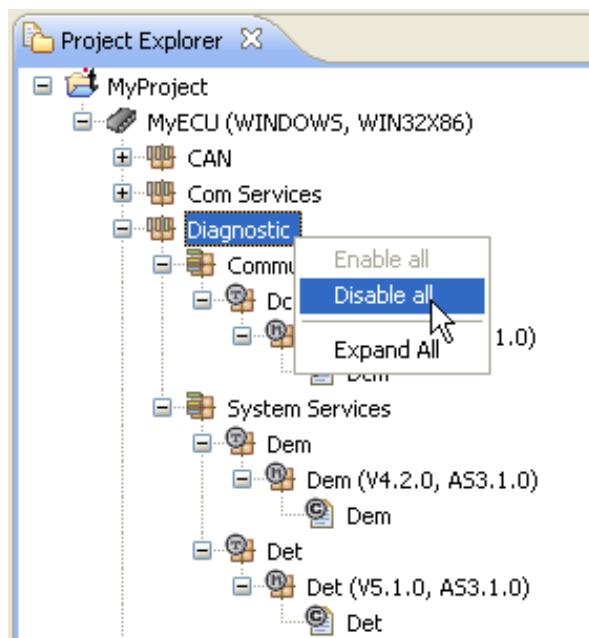


Figure 7.89. Disabling all module configurations of one node

- ▶ In the context menu, select *Disable all*.

NOTE**The *Disable all* command might not be available**

The *Disable all* command is only available if the selected node contains more than one module configuration and if one of these module configurations is enabled. If the node contains only one module configuration, the *Disable* command is displayed instead.

The selected module configurations are now disabled and grayed out.

7.6.4. Enabling and disabling code generation for module configurations

In the *Module Configurations* table you may enable or disable code generation for the module configurations of your project.

WARNING**Only partly generated code leads to incorrectly working ECUs**

If you only generate the code of a part of the module configurations you later want to implement on an ECU, depending on which modules you choose, the code might not work correctly. This might lead to incorrectly working ECUs.

To avoid incorrectly working ECUs, generate the code of all module configurations in your project before deploying them on an ECU.

To enable code generation for module configurations:

- ▶ To enable code generation for only one module configuration, select the **Generate** check box of the module configuration for which you want to enable code generation in the *Module Configurations* table.

Module Configurations				
Enable	Generate	Module	Name	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Base (V4.2.3, AS3.1.0)	Base	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	CanSM (V2.0.2, AS3.1.0)	CanSM	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Com (V5.0.8, AS3.1.0)	Com	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Make (V3.2.0, AS3.1.0)	Make	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Platforms (V1.0.1, AS3....	Platforms	

Figure 7.90. Enabling code generation for one module configuration

- ▶ To enable code generation for several module configurations at once:
 - ▶ In the *Module Configurations* table, select the module configurations, for which you want to enable code generation.
 - ▶ In the tool bar of the *Module Configurations* page, click the **Enable code generation** button

To disable code generation for module configurations:

- ▶ To disable code generation for only one module, clear the **Generate** check box of the module configuration for which you want to disable code generation in the *Module Configurations* table.

Module Configurations				
Enable	Generate	Module	Name	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Base (V4.2.3, AS3.1.0)	Base	
<input type="checkbox"/>	<input type="checkbox"/>	CanSM (V2.0.2, AS3.1.0)	CanSM	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Com (V5.0.8, AS3.1.0)	Com	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Make (V3.2.0, AS3.1.0)	Make	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Platforms (V1.0.1, AS3....	Platforms	

Figure 7.91. Disabling code generation for one module configuration

- ▶ To disable code generation for several module configurations at once:

- ▶ In the *Module Configurations* table, select the module configurations, for which you want to disable code generation.
- ▶ In the tool bar of the *Module Configurations* page, click the **Disable code generation** button

7.6.5. Naming *Module Configurations*

When you add new module configurations to your project, the name of the new module configuration is set automatically to a name that is unique within the project using either the default configuration name provided by the module or the module name itself.

Instead of using the default name you may want to use another name for the module configuration.

NOTE
Only in recently added module configurations the name can be changed


You can only change the name of a module configuration if you added this module configuration during the same session of the dialog.

To edit the name of a module configuration:

- ▶ Add the required module configuration to your project.
- ▶ In the *Module Configurations* table, click into the *Name* cell of the module configuration whose name you want to change.
- ▶ Type the new name.



Figure 7.92. Editing the name of a new module configuration

NOTE
Name constraints


Mind that the module configuration name must meet the following constraints:

- ▶ The name must be unique within the project.
- ▶ The name must start with a letter.
- ▶ The name must contain only letters and digits plus the underscore character ('_').
- ▶ The name must not be empty.
- ▶ The name must not exceed 255 characters.

-
- ▶ Press **ENTER**.

The name of the module configuration is changed.

7.6.6. Choosing *Pre-Configurations*

Module suppliers optionally can equip modules with one or more preconfigurations that contain values for configuration parameters. If a module configuration of your project provides one or more preconfiguration(s), you must assign this module configuration to such a preconfiguration when adding it to your project. The parameters covered by the chosen preconfiguration are then fixed in the module configuration and cannot be edited by the user.

In the *Default configuration setting* area, you can set the *Default Pre-Configuration* which is used as a default value for all new module configurations. To support a more consistent selection, this preconfiguration setting is persisted. You can also edit the *Pre-Configuration* of an individual module configuration.

7.6.6.1. Setting the *Default Pre-Configuration*

In the *Default configuration setting* area of the *Module Configurations* page, you may edit the *Default Pre-Configuration* for new module configurations.

NOTE**The value set is a union set of all preconfigurations provided by all modules**

The preconfiguration names to choose from 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 preconfiguration is taken instead when you add it to your project.

To change the default preconfiguration, select the required preconfiguration in the **Default Pre-Configuration** drop-down list box.

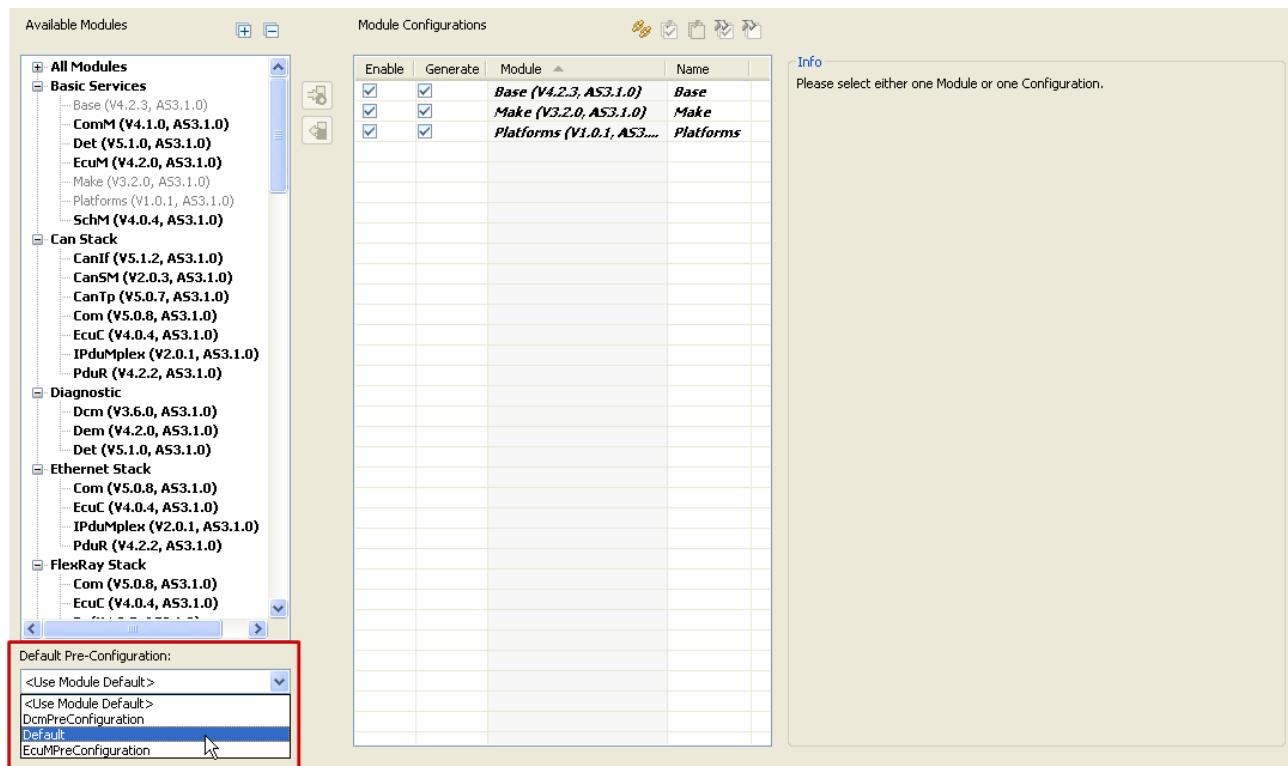


Figure 7.93. Selecting the default preconfiguration for new module configurations

- ▶ To use the respective default preconfiguration for new module configurations, select <Use Module Default>.

NOTE

Non-compilable code

If you select <Use Module Default>, every module will be added to your project with a different preconfiguration. This might lead to incompatibilities of the module configurations and thus to non-compilable code.

- ▶ The selected default preconfiguration is saved, whenever you actually add new module configurations to your project.

7.6.6.2. Changing the preconfiguration of one specific module configuration

To change the preconfiguration of one specific module configuration:

- ▶ Add the required module configuration to your project.

NOTE**Only in recently added module configurations the preconfiguration can be changed**

You can only change the preconfiguration of a specific module configuration if you added this module configuration during the same session of the dialog.

- ▶ In the *Module Configurations* table, mark the module configuration whose preconfiguration you want to change.

In the *Details* area of the *Module Configurations* page, detailed information about the specific module is displayed.

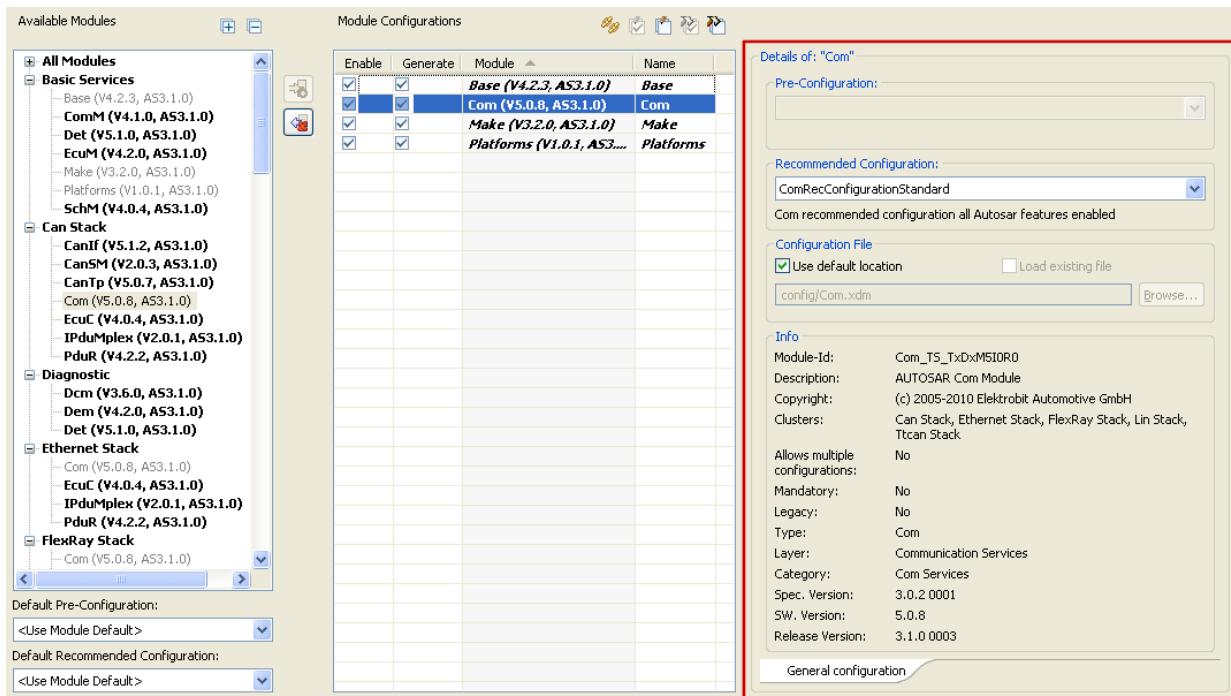


Figure 7.94. The *Details* area in the *Module Configurations* page

- ▶ In the **Pre-Configuration** drop-down list box in the *Details* area, select the required preconfiguration for the selected module configuration.

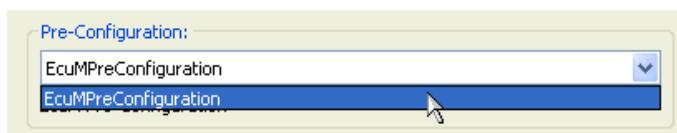


Figure 7.95. Changing the preconfiguration of one specific module

NOTE**Grayed-out preconfigurations are not available**

If the **Pre-Configuration** drop-down list is grayed-out, the selected module does not provide a preconfiguration of its own.

7.6.7. Applying recommended configurations

Module suppliers optionally can equip modules with one or more recommended configurations that contain values for configuration parameters. If a module configuration of your project provides one or more recommended configurations, you can choose one when you add a configuration for this module to your project. The parameter values covered by the chosen recommended configuration can be considered as a guideline. They may be edited by the user later.

In the *Default configuration setting* area, you can set the *Default Recommended Configuration* which is used as a default value for all new module configurations. To support a more consistent selection, this setting is persisted. You can also edit the *Recommended Configuration* of an individual module configuration.

7.6.7.1. Setting the *Default Recommended Configuration*

In the *Default configuration setting* area of the *Module Configurations* page, you may edit the *Default Recommended Configuration* for new module configurations.

NOTE**The value set is a union set of all recommended configurations provided by all modules**

The names of the recommended configurations to choose from 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 configuration is taken instead when you add it to your project. If the module does not provide a default configuration, no recommended configuration is applied.

To change the default recommended configuration, select the required recommended configuration in the **Default Recommended Configuration** drop-down list box.

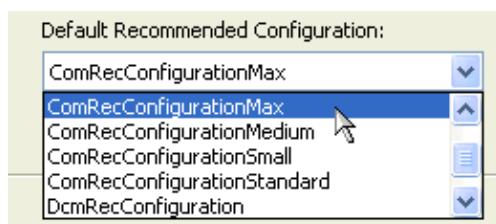


Figure 7.96. Selecting the default recommended configuration for new module configurations

- ▶ If you don't want to apply any of the recommended configurations, select <None>.
- ▶ To use the respective default recommended configuration for new module configurations, select <Use Module Default>.

NOTE**Non-compilable code**

If you select <Use Module Default>, every module will be added to your project with a different recommended configuration. This might lead to incompatibilities of the module configurations and thus to non-compilable code.

- ▶ The selected default recommended configuration is saved, when you actually add new module configurations to your project.

7.6.7.2. Changing the recommended configuration of one specific module configuration

To change the recommended configuration of one specific module configuration:

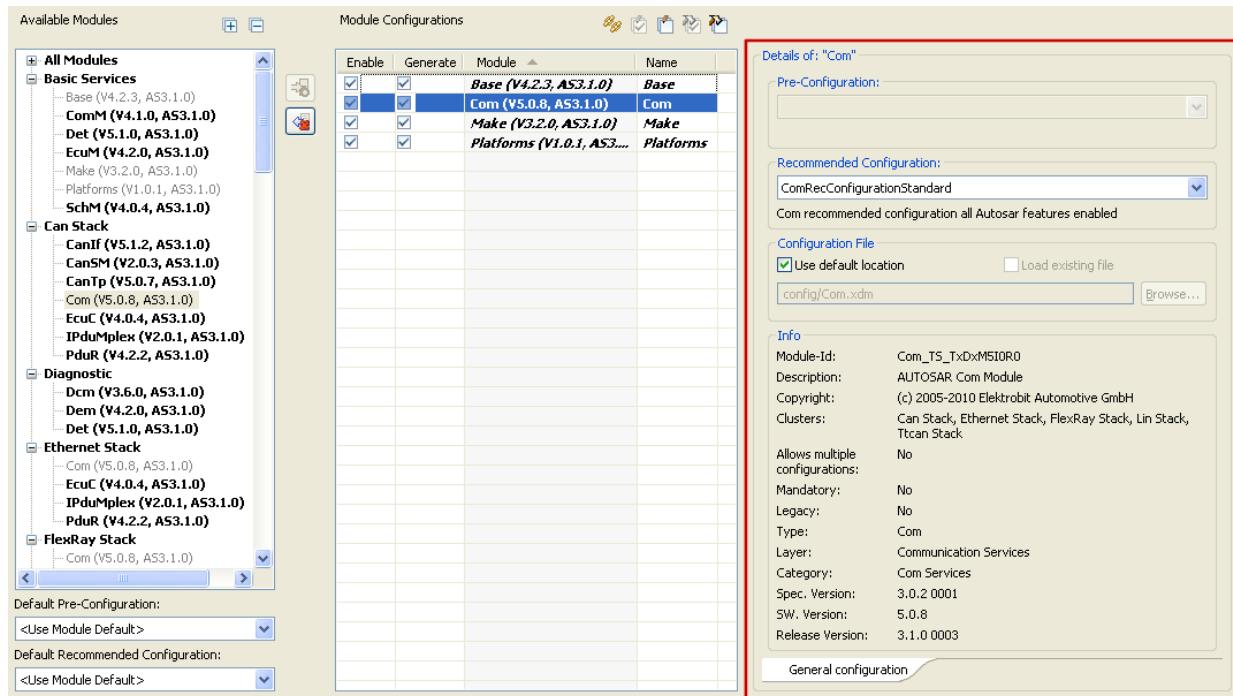
- ▶ Add the required module configuration to your project.

NOTE**Only in recently added module configurations the recommended configuration can be changed**

You can only change the recommended configuration of a specific module configuration if you have added this module configuration during the same session of the dialog.

- ▶ Mark the module configuration whose recommended configuration you want to change.

In the *Details* area of the *Module Configurations* page, detailed information about the specific module is displayed.

Figure 7.97. The *Details* area in the *Module Configurations* page

- ▶ In the **Recommended Configuration** drop-down list box, select the required recommended configuration for the selected module configuration.

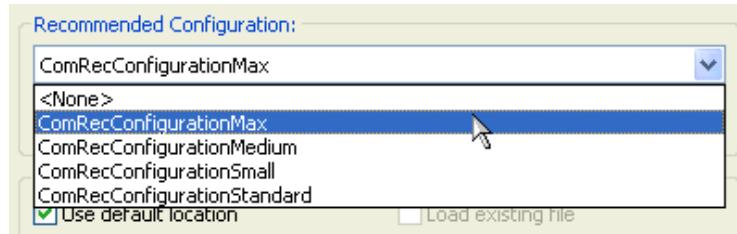


Figure 7.98. Changing the recommended configuration of one specific module

NOTE**Grayed-out recommended configurations are not available**

If the **Recommended Configuration** drop-down list is grayed-out, the selected module does not provide a recommended configuration of its own.

7.6.8. Editing the configuration file settings

The configuration file stores all parameters for a specific module configuration. For detailed information on how to edit those parameters, e.g. by using the Generic Configuration Editor, refer to [Section 7.8, “Editing parameters of a module configuration”](#).

When you add new module configurations to your project, per default the configuration file name is set to config/<moduleconfigurationname>.xdm, relative to your project folder. This is also referred to as the default location.

Instead of using the default location you can use another location or name for the configuration file. To use another location or file name:

- ▶ Add the required module configuration to your project.

NOTE**Only in recently added module configurations the configuration file can be changed**

You can only change the configuration file of a module configuration, if you have added this module configuration during the same session of the dialog.

- ▶ In the *Module Configurations* table, mark the module configuration whose configuration file you want to change.
- ▶ In the *Configuration File* panel of the *Details* area, clear the check box **Use default location**.

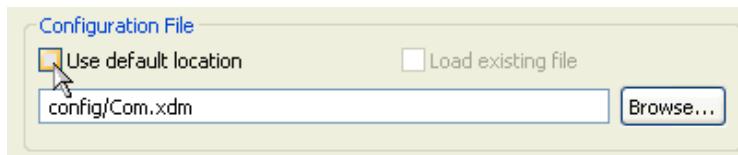


Figure 7.99. Editing the configuration file settings

- ▶ Enter or choose the new file name. Either use absolute path names or path names relative to your project.

NOTE**The file extension .xdm is mandatory**

You can only use file names ending with the extension `.xdm`. When you choose an existing file, you must make sure that it matches the targeted module configuration schema. Otherwise the new module configuration will fail to load when applied.

7.6.8.1. Using existing configuration files

Instead of configuring the module configuration from scratch you may also use an already existing configuration file.

NOTE**You cannot set preconfigurations and recommended configurations when you load an existing configuration file**

If you use an existing configuration file, you cannot apply any preconfiguration or recommended configuration.

To load an already existing configuration file into your project:

- ▶ In the *Configuration File* panel, browse to the file you want to load.

NOTE**The selected configuration file will be modified**

The selected configuration file will be modified while you edit the parameters of the module configuration. If you want to preserve the file as it is, copy the file to your project before loading it to your project.



Figure 7.100. Using an existing configuration file

Whenever you point the *Configuration File* panel to an existing file in the file system, the *Load existing file* check box is marked.

NOTE**Data loss**

If you clear the *Load existing file* check box, the existing file will be overwritten and cannot be restored again.

7.6.9. Upgrading module configurations

To upgrade module configurations from older software versions of the modules to newer ones:

- ▶ Click the **Upgrade** button in the tool bar of the *Module Configurations* table.

The *Upgrade* dialog opens up.

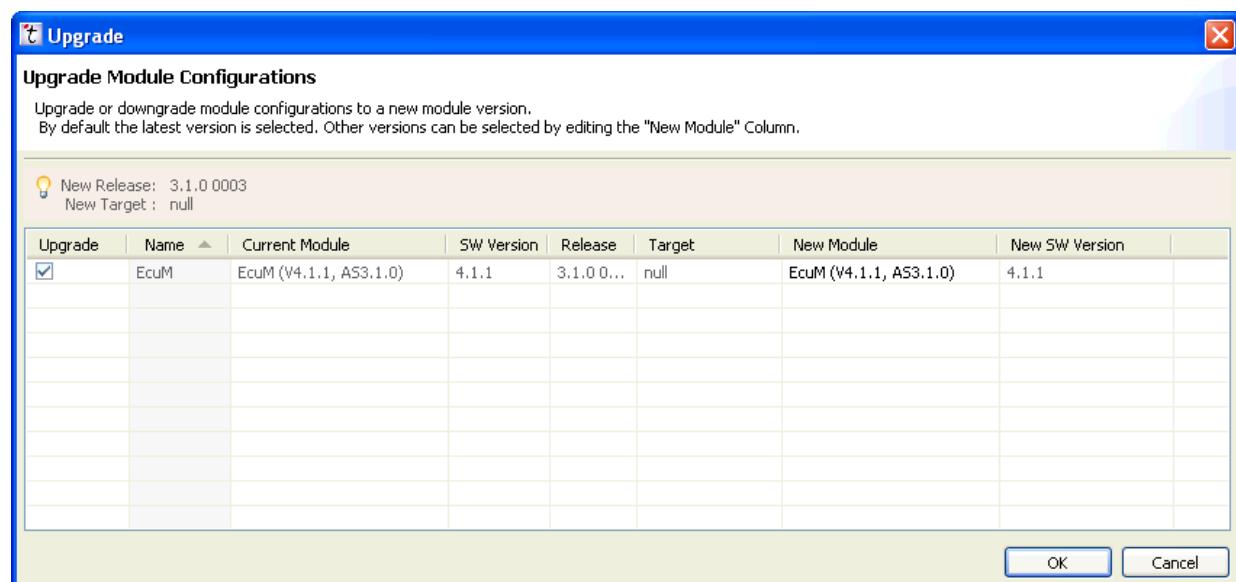


Figure 7.101. Upgrading module configurations in the *Upgrade* dialog

For detailed information on the *Upgrade* dialog, refer to [Section 7.7, “Upgrading projects and configurations”](#).

7.7. Upgrading projects and configurations

This chapter covers several aspects of working with projects and modules of different versions.

When a project is created, you have to decide on an AUTOSAR release version for the project. It may happen at a later point in your development cycle that you want to migrate your project to a newer AUTOSAR version.

[Section 7.7.1, “Converting a project to a new AUTOSAR version”](#) describes how EB tresos Studio can help you with this.

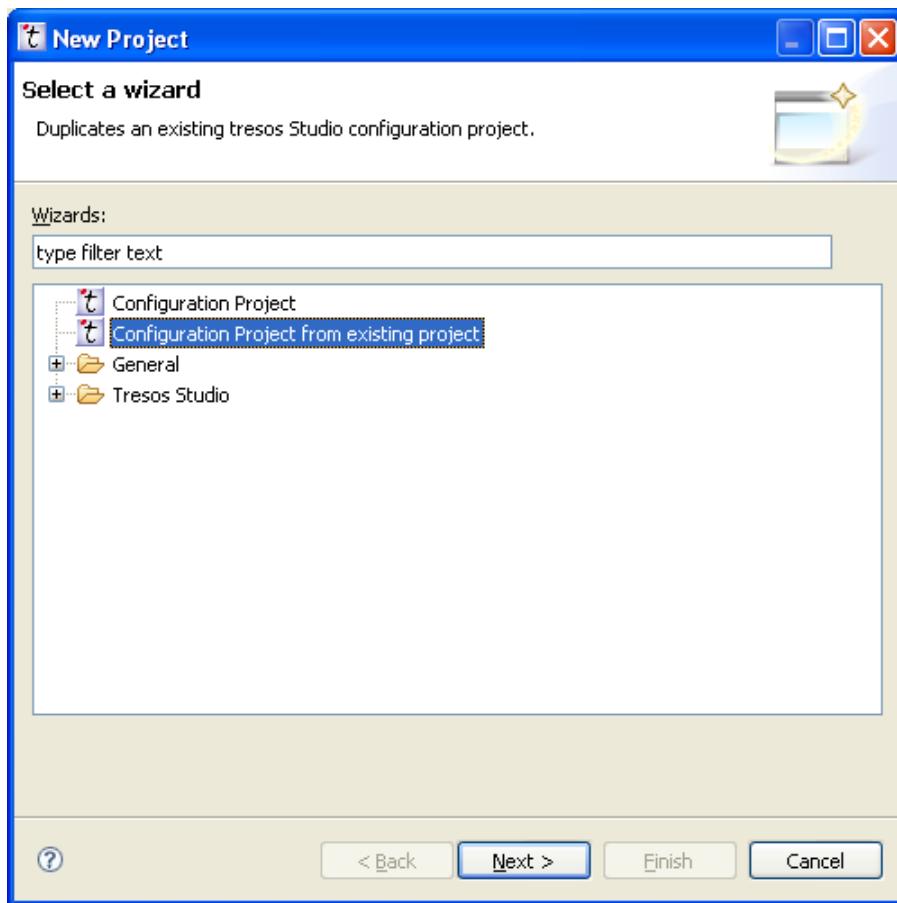
Regardless of the AUTOSAR release version number, the modules themselves contain a version number, called the *software version*. It is possible that you have installed modules of the same AUTOSAR version, but with different software versions. If you have started a project with modules of an older software version, you may want to upgrade these modules to newer versions. For this case the *Module Configurations* dialog provides an upgrade function that is accessible via a button. The usage of this feature is explained in [Section 7.7.2, “Upgrading module configurations to new module versions”](#).

To upgrade projects and configurations you need to install plug-ins. To find out how to install different plug-ins, refer to the EB tresos® installation guide. As PDF the EB tresos® installation guide is located at `$TRESOS_BASE/doc/1.0_Installation` with `$TRESOS_BASE` being the location on your computer where you installed EB tresos Studio. To access the installation in the Eclipse help, open EB tresos Studio and press **F1**, then navigate to the folder *Installation*.

7.7.1. Converting a project to a new AUTOSAR version

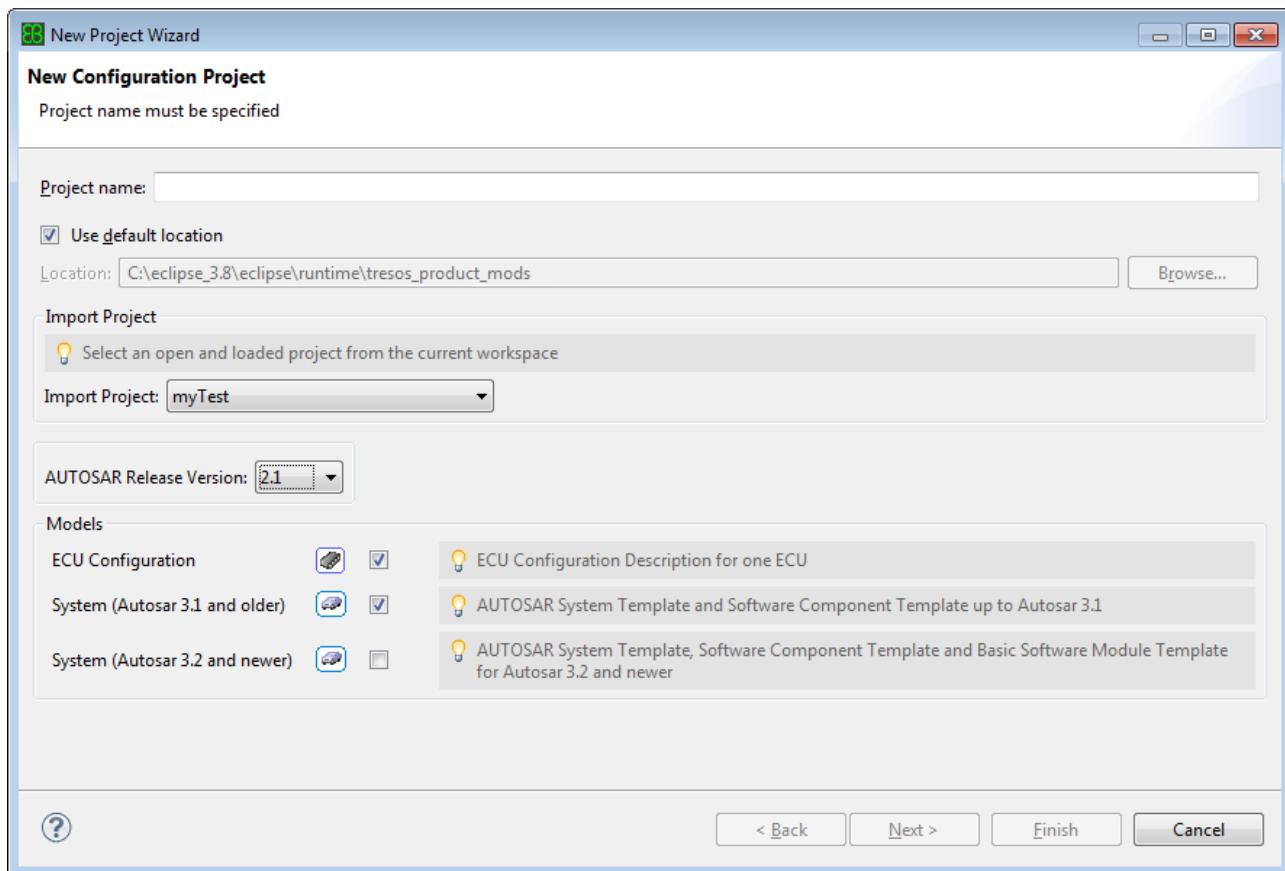
To convert a project (e.g. AUTOSAR 2.0 configuration) to a newer version (e.g. AUTOSAR 2.1 configuration):

- ▶ Follow the step 1 of [Section 7.3.1.1, “Step 1: Starting the Project Wizard”](#).
- ▶ In step 2, select *Configuration Project from existing Project*.



- Select *Next*.

The *New Configuration Project* window opens up.

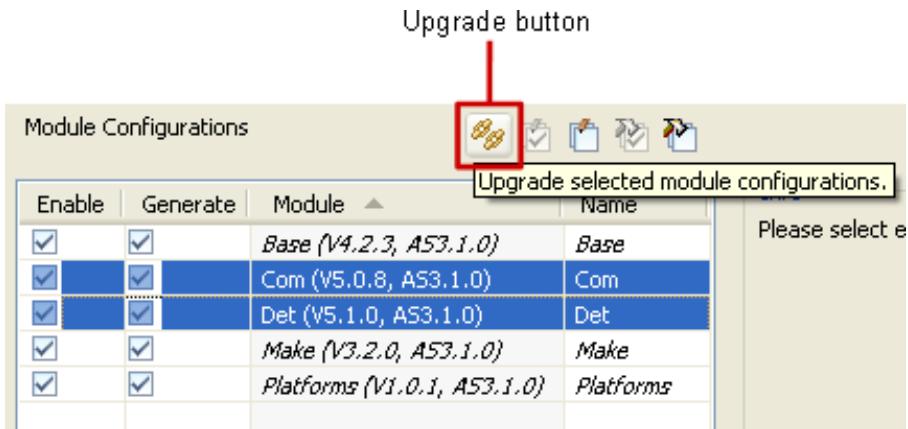


- ▶ Type in a name for the project in the *Project name* textbox.
- ▶ Provide the source project to copy from in the dropdown listbox *Import Project*.
- ▶ Proceed with the wizard analog to [Section 7.3.1.2, “Step 2: Creating a new project”](#).

The dropdown listbox *Import Project* allows you to select only projects of your current workspace, which are open and loaded. For opening and loading closed projects, please refer to chapter [Section 7.3.6, “Closing and opening projects”](#). If the project you would like to convert is not yet part of your workspace, please refer to chapter [Section 7.10.1.1, “Importing a project”](#) to import the desired project first. After you have imported the project into your workspace, the *Import Project* dropdown listbox will automatically list the imported project as available.

7.7.2. Upgrading module configurations to new module versions

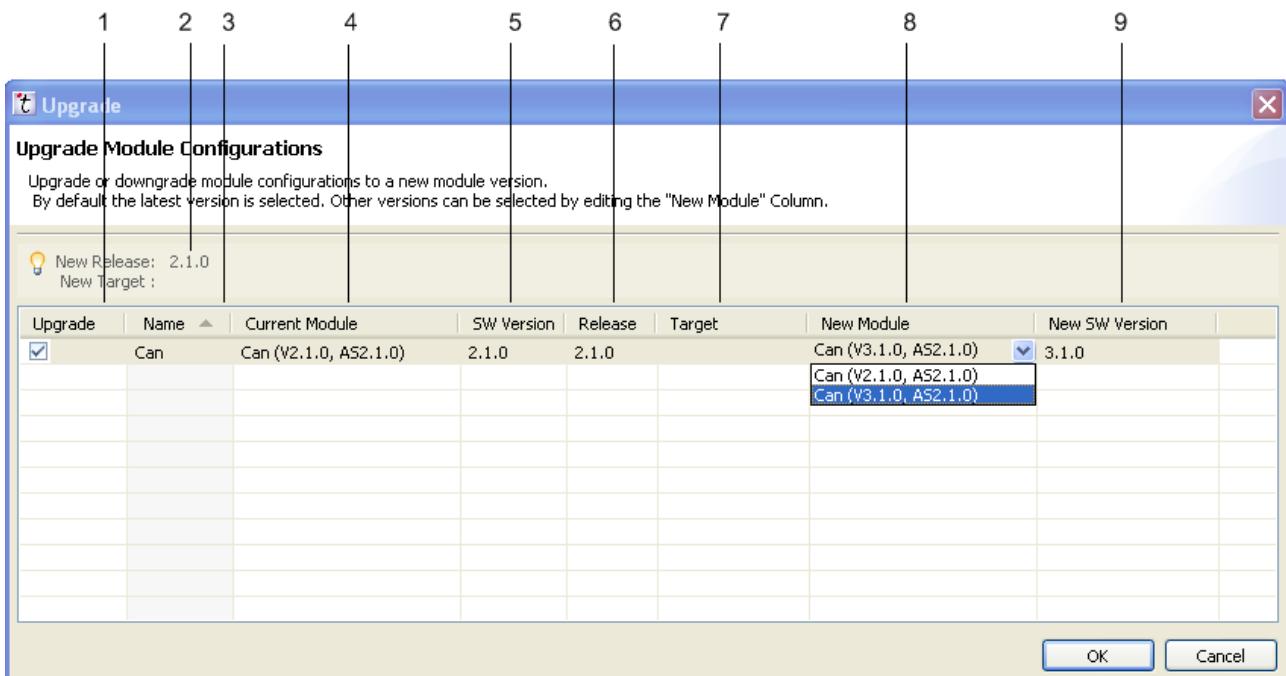
When new versions of modules are delivered and the configuration is changed, the new version is installed to EB tresos Studio as a new module with a new software version number. To switch a configuration between different versions of a module the *Module Configurations* dialog provides an upgrade function that is accessible via the **Upgrade** button.



The upgrade function is available from:

- ▶ The *New Project* wizard for new configuration projects (see: [Section 7.3.1, “Creating a new project”](#)).
- ▶ The *New Configuration Project from existing project* wizard for derived configuration projects (see: [Section 7.7.1, “Converting a project to a new AUTOSAR version”](#)).
- ▶ The *Module Configuration* dialog accessible via the context menu of the *Project Explorer* (see: [Section 7.6, “Editing module configurations”](#)).
- ▶ The *Properties for project name* dialog accessible via the context menu of the *Project Explorer* (see: [Section 7.3.7, “Viewing and changing project properties”](#)).

If you click the **Upgrade** button, the *Upgrade* dialog is shown. When module configurations are selected in the *Module Configurations* dialog, only these are selected for upgrade, if no configurations are selected all module configurations are presented for upgrade.



Number	Description
1	Column <i>Upgrade</i> : The dialog allows to select the module configurations that are actually upgraded via the checkboxes in the column labeled <i>Upgrade</i> .
2	The info box shows release version and the target of the project which matches all presented new modules.
3	Column <i>Name</i> : This column shows the name of the module configuration.
4	Column <i>Current Module</i> : Shows the name of the module that is currently used for the configuration.
5	Column <i>SW Version</i> : Shows the software version of the module that is currently used for the configuration.
6	Column <i>Release</i> : Shows the release version of the module that is currently used for the configuration.
7	Column <i>Target</i> : Shows the target and derivate of the module that is currently used for the configuration.
8	Column <i>New Module</i> : Allows to select the new module to use for the configuration. By default the latest available is selected. Clicking into a cell of this column presents a drop down box that allows to switch the module.
9	Column <i>New SW Version</i> : Shows the software version of the selected module.

Pressing the **Ok** button closes the dialog and updates the Module Configuration dialog. Closing the Module Configuration dialog actually switches the module configuration of the project to their new module definitions adapting configuration parameters if necessary.

7.8. Editing parameters of a module configuration

7.8.1. Overview

In this chapter you will learn how to edit parameters of a module configuration,

- ▶ either manually with the ECU Configuration Editor in [Section 7.8.3, “Opening the ECU Configuration Editor”](#),
- ▶ or automatically with unattended wizards [Section 7.8.7, “Autoconfiguring routine jobs”](#).

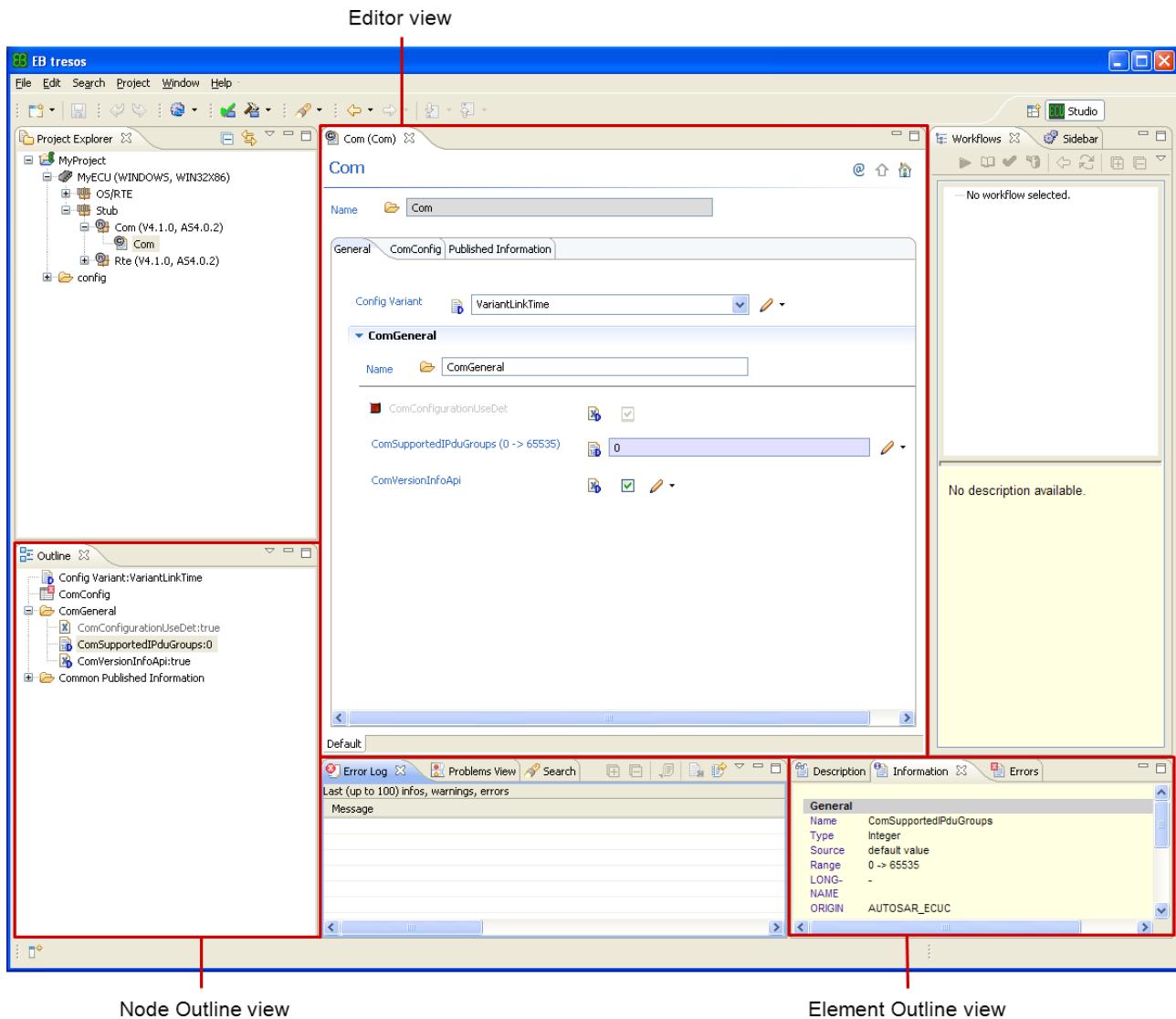
This chapter also shows you how to search for specific configuration parameters in [Section 7.8.8, “Searching for configuration parameter names and their values”](#).

7.8.2. Background information

EB tresos Studio provides a standard implementation of a Generic Configuration Editor that takes the parameter definition of a module as input and creates a dialog-based GUI.

An editor consists of:

- ▶ the *Editor view* in the center of the EB tresos Studio window, which enables editing the configuration data of a project
- ▶ the *Node Outline* view on the left bottom of the workbench, which shows a tree view of the editable configuration parameters that you may edit in the *Editor view*
- ▶ the *Element Outline* views on the right bottom of the workbench, which consists of multiple views that show information about the currently selected parameter in the *Editor view*



Detailed information on the Generic Configuration Editor is available in [Section 6.4.1, “The Generic Configuration Editor”](#).

For a detailed description of the *Node Outline* view and the *Element Outline* view, refer to [Section 6.3.3, “Node Outline view”](#), and [Section 6.3.4, “Element Outline views”](#).

7.8.3. Opening the ECU Configuration Editor

To configure a module, you need to open the ECU Configuration Editor:

- ▶ Expand the tree underneath the project in the *Project Explorer*.

The *Project Explorer* tree opens up to display several layers. It depends on your *Project Explorer* display settings which layers are displayed.

- ▶ Select the module configuration you want to edit. Either
 - ▶ double-click on the module configuration
 - ▶ or right-click on the module configuration and select *Open Editor*.

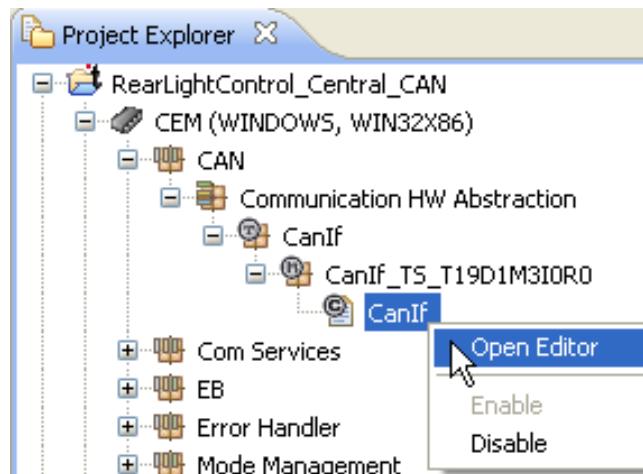


Figure 7.102. Opening the ECU Configuration Editor

The corresponding *Editor* view opens up to the right of the Project Explorer.

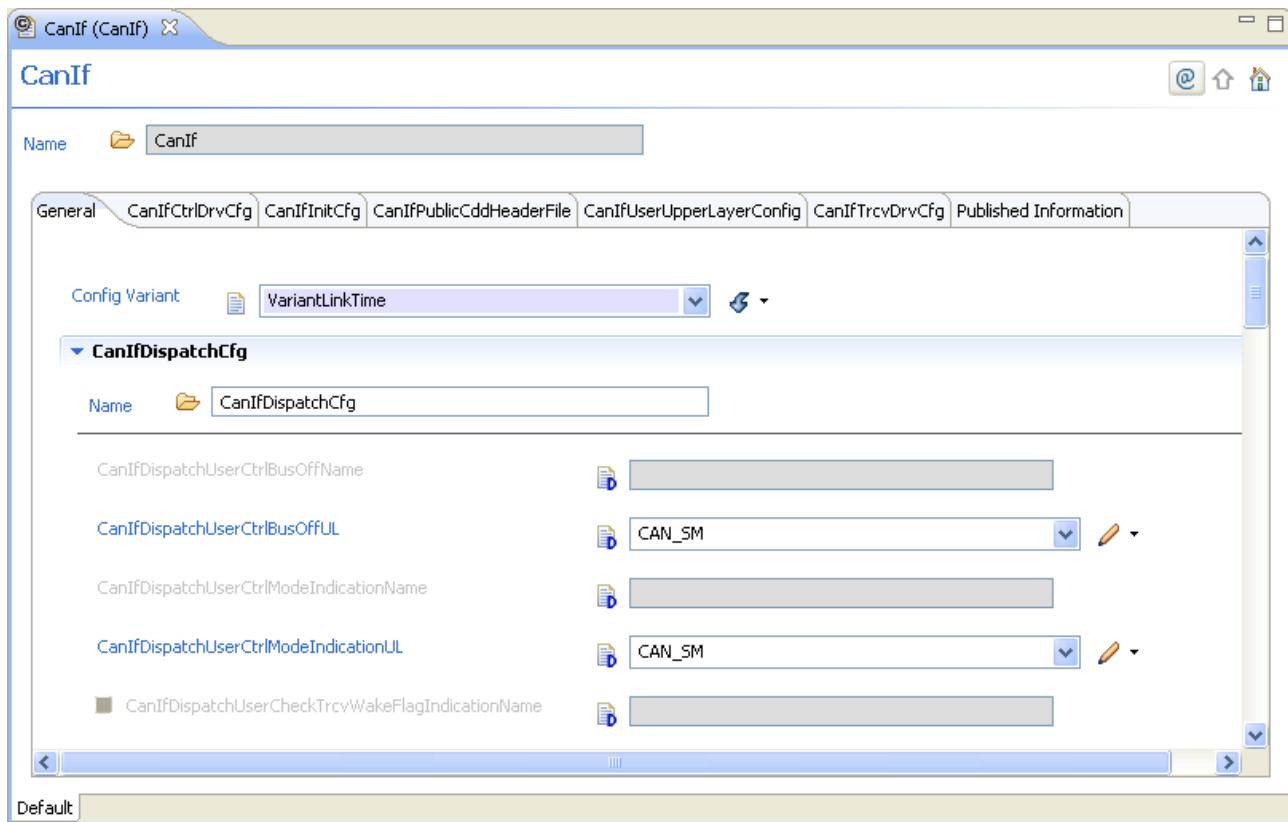


Figure 7.103. The Editor view

An *Editor* view of a module configuration consists of several pages, which are displayed as tabs in the *Editor* view. Please refer to [Section 6.3.2, “The Editor view”](#) to get to know the basic set-up of this view and its basic functionalities.

7.8.4. Setting the configuration variant of the module

The first parameter on the *General* tab is the configuration variant, displayed as **Config Variant** drop-down list box in the *Editor* view. The configuration variant defines which configuration class is used for each parameter of the module and influences the editable state of the parameters. For background information on the configuration variant refer to [Section 4.5.2.1, “Configuration time of projects”](#).

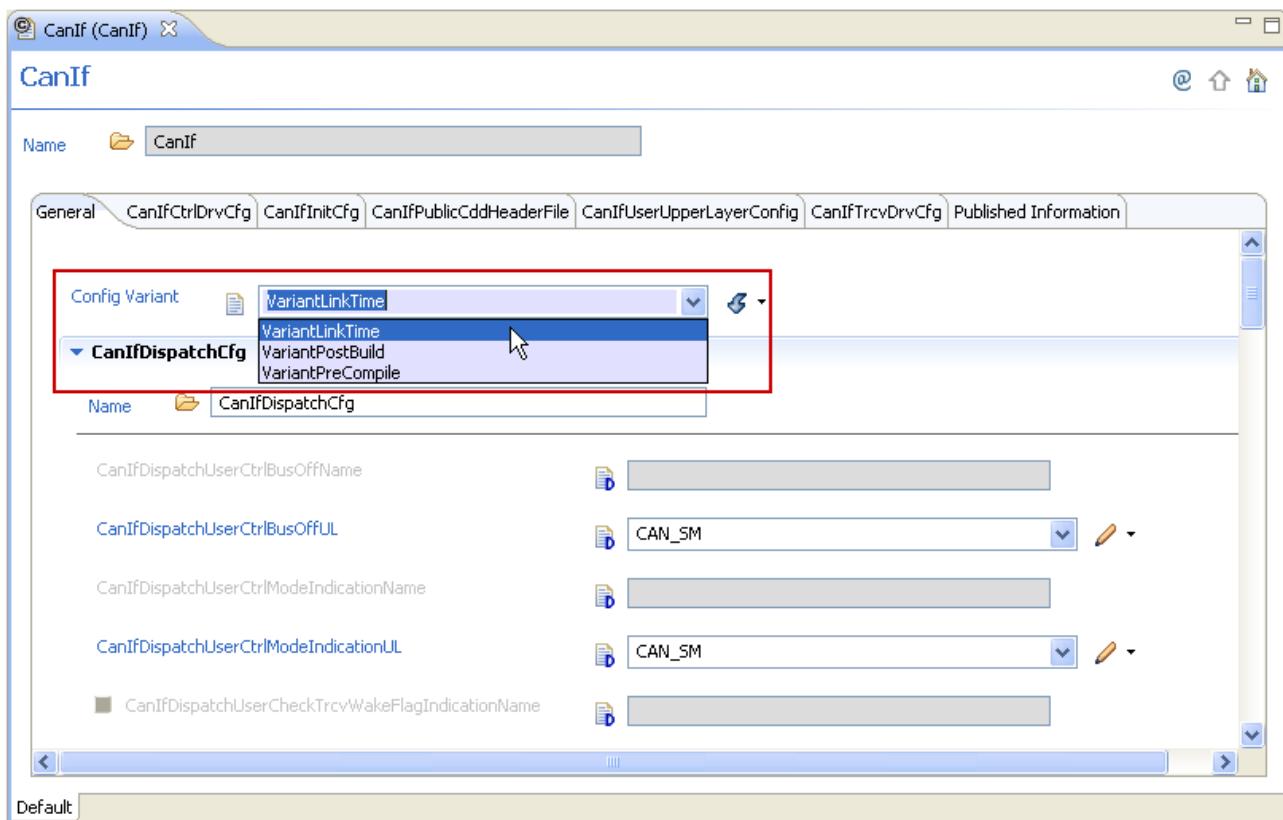


Figure 7.104. The configuration variant parameter

- ▶ To specify that the module is configured before compiling the source code, select *VarianPreCompile*.
- ▶ To specify that the object code of the module receives parts of its configuration from another object code file, select *VariantLinkTime*.
- ▶ To specify that the module gets the parameters of its configuration either by downloading a separate file to the ECU memory (`POST-BUILD-LOADABLE`) or by defining multiple configuration sets and choosing one during boot-time (`POST-BUILD-SELECTABLE`), select *VariantPostBuild*.

7.8.5. Adding parameter comments

In EB tresos Studio, you may add comments to a parameter. In this comment you may add a summary, a description and set the current status of this comment. It is possible to add several comments in different languages to a parameter.

TIP**Undoing/Redoing changes of parameter comments**

All applied changes to parameter comments as well as remove comment actions are applied to the undo stack. Thus, you may undo or redo the changes.

7.8.5.1. Adding and editing a parameter comment

To add or edit a parameter comment:

1. In the Generic Configuration Editor, click or right-click the type icon of the required parameter.

A context menu opens up.



Figure 7.105. Adding or editing a parameter comment

TIP**Finding context menus**

To find out whether a context menu is available, move your mouse pointer over the type icon. If the pointer turns into a hand, a context menu is available. To open it, click or right-click on the icon.

2. In this context menu, select *Add/Edit Comments*.

The *Comments for <parameter name>* dialog opens up.

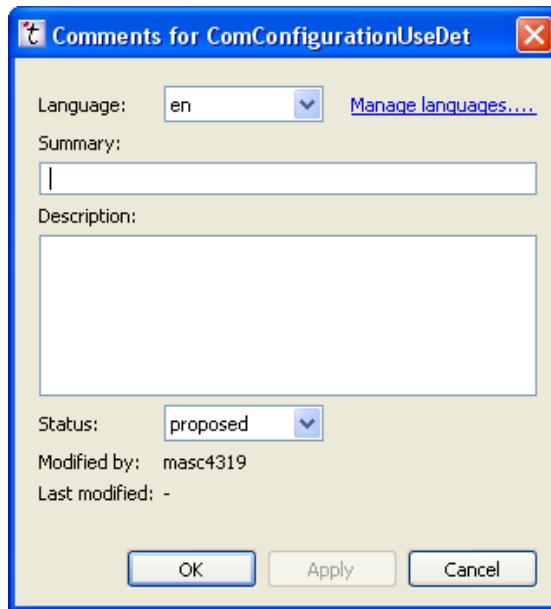


Figure 7.106. Adding a new comment to a parameter

- ▶ In the **Language** drop-down list box, select the language in which you want to write the comment.

To change the set of languages that appear in the **Language** drop-down list box, click the **Manage languages...** link next to the **Language** drop-down list box. For further information on changing the languages for the comments, refer to [Section 7.8.5.2, “Managing languages of parameter comments”](#).

- ▶ In the **Summary** text box, you may add a summary to your comment. The summary is then displayed in the Generic Configuration Editor when hovering with the mouse pointer over the parameter icon.
- ▶ In the **Description** text field, you may add your comment.
- ▶ In the **Status** drop-down list box, select a status for your comment.

The fields *Modified by* and *Last modified* are filled in automatically by EB tresos Studio

You may now choose between three alternative steps to continue:

- ▶ To apply the changes and close the dialog, click **OK**.
- ▶ To apply the current changes without closing the dialog, click **Apply**. The **Apply** button is only enabled if you have changed the values.
- ▶ To discard the last changes and close the dialog, click **Cancel**.

TIP**You may edit comments of several parameters without closing the dialog**

The *Comments for <parameter name>* dialog is a non modal dialog. This means, that you may add or edit comments for several parameters without closing and reopening the dialog.

To do that, select another configuration parameter in the Generic Configuration Editor or in the *Outline* view while the *Comments for <parameter name>* dialog is open.

The *Comments for <parameter name>* dialog always displays the name of the currently selected configuration parameter in the title bar of the dialog.

A confirmation dialog appears, if your comment contains unapplied changes and you

- ▶ have changed the language in the **Language** drop-down list box, or
- ▶ have selected another configuration parameter in the Generic Configuration Editor or via the *Outline* view.

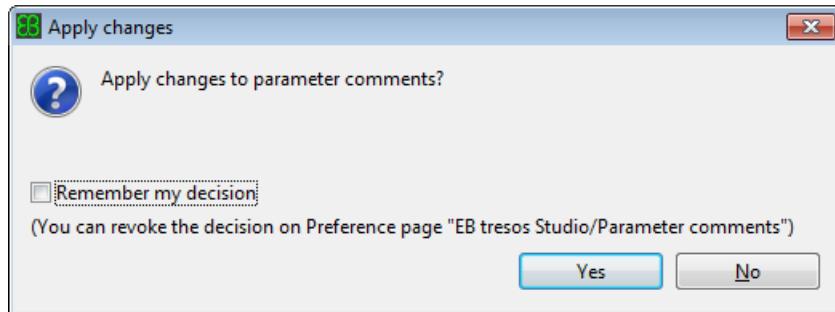


Figure 7.107. The Apply Changes dialog

If you check the *Remember my decision* check box, this dialog will not appear the next time when unapplied changes may be unintentionally discarded. Then the remembered user decision is executed without asking the user again.

NOTE**Turning confirmation dialogs on and off**

In the EB tresos Studio preferences you may change whether this confirmation dialogs shall appear or not. for further information on the EB tresos Studio preferences, refer to [Section 7.2.3.3, “Setting preferences for parameter comments”](#).

NOTE**If you close your project, your changes cannot be saved**

If your comment contains unapplied changes and you close the current project, the unapplied changes cannot be saved. In this case, an information dialog appears and your last comment values are written into an entry in the *Error Log* view.

7.8.5.2. Managing languages of parameter comments

To manage the languages that are available in the **Language** drop-down list box of the *Comments for <parameter name>* dialog, select the **Manage Languages ...** link in the *Comments for <parameter name>* dialog.

The *Manage Languages* dialog opens up:



Figure 7.108. Managing languages for parameter comments

- ▶ To add a new language, select the required language. Click **OK** to apply your changes and close the dialog. The new language is now available in the **Language** drop-down list box of the *parameter comment dialog*.
- ▶ To remove a language select the language you want to remove. Click **OK** to apply changes and close the dialog. The removed language is also removed from the language box of the *Comments for <parameter name>* dialog.

NOTE
You cannot remove the default language


The default language **en** as well as all languages for which comments already exist, are highlighted in gray background color and cannot be removed.

7.8.5.3. Removing parameter comments

To remove all comments for one parameter, click or right-click the type icon of the required parameter and select *Remove Comments* from the context menu that opens up.

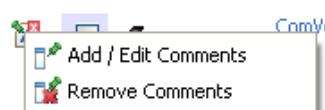


Figure 7.109. Removing parameter comments

A confirmation dialog appears:

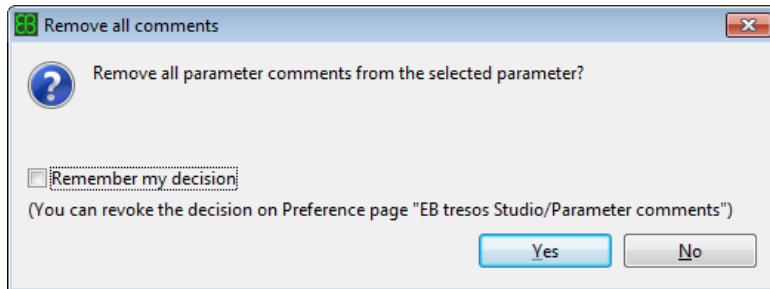


Figure 7.110. Confirming the removal of parameter comments

If you check the *Remember my decision* check box, this dialog will not appear the next time but the remembered user decision is executed without asking the user again.

7.8.5.4. Generating a parameter comment report

With the application example *DemoReportGenerator*, you may generate a report for the parameter comments. To generate a parameter comment report:

1. Install the application example *DemoReportGenerator*.
 - ▶ You find the *DemoReportGenerator* application example in the <tresos-install-dir>/demos/*Studio* folder.
 - ▶ Copy the *DemoReportGenerator* to the plugins directory <tresos-install-dir>/plugins.
 - ▶ Restart EB tresos Studio.
2. Generate a report:
 - ▶ Load the configuration of the project you want to create a report for into EB tresos Studio.
 - ▶ Select the configuration project node in the *Project Explorer*.
 - ▶ In the tool bar, select the arrow down button next to the **Generate** button . A context menu opens up.
 - ▶ In the context menu, select *config_report*.

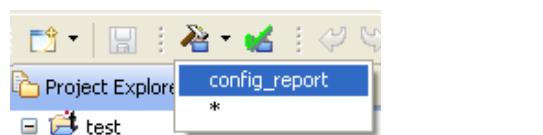


Figure 7.111. Generating a parameter comment report

The parameter comment report is generated. You find this report at <project-root>/output/reports/ecu_configuration_report.csv.

7.8.5.4.1. Reading the parameter comment report

After you have generated the parameter comment report, you find it at <project-root>/output/reports/ecu_configuration_report.csv.

Open the *ecu_configuration_report.csv* file.

You see a table with several rows, one for each configuration parameter. The available columns are described below.

	A	B	C	D	E	F	G	H	I	J	K
1	Project: test										
2	Module	Summary	Comment	Status	User	Date	Language	Value	Path		
1	PublicApiTest	This is a summary	You can add a multi-line description.	todo	masc4319	2011-01-12T11:56:19 CET	en		ExamplesCommon	Bool	
25	PublicApiTest								ExamplesCommon	Int	
26	PublicApiTest								ExamplesCommon	String	
27	PublicApiTest								ExamplesCommon	StringList	
28	PublicApiTest								ExamplesResourceFile	SubDerivate	
29	PublicApiTest								ExamplesResourceFile	InputFrequency	
30	PublicApiTest							4000000.0	ExamplesResourceFile	OutputFrequency	
31	PublicApiTest							PRE_1.0	ExamplesResourceFile	OutputFrequencyPrescaler	
32	PublicApiTest								ExamplesResourceFile	FeatureCan	
33	PublicApiTest								ExamplesResourceFile	FeatureCan	CanParam1
34	PublicApiTest								ExamplesResourceFile	FeatureCan	CanParam2

Figure 7.112. A parameter comment report

1	Module	Displays the module name of the node currently selected.
2	Summary	Displays the summary of the comment of the current node. If no comment has been added to this parameter, this cell is empty.
3	Comment	Displays the comment description of the current node. If no comment has been added to this parameter, this cell is empty.
4	Status	Displays the status of the comment of the current node. If no comment has been added to this parameter, this cell is empty.
5	User	Displays the name of the user who has added the comment for the current node. If no comment has been added to this parameter, this cell is empty.
6	Date	Displays the date and time of the last edit of the comment. If no comment has been added to this parameter, this cell is empty.
7	Language	Displays in which language the comment is written. If no comment has been added to this parameter, this cell is empty.
8	Value	If the node is a parameter that contains a value, this value is displayed in this column.
9	Path	Displays the path of the root node in the model. To enhance filter and search support, the nodes of the path are displayed in the columns to the right of the <i>Path</i> column.

7.8.5.4.2. Customizing the DemoReportGenerator

The DemoReportGenerator is based on a template-based code generator and therefore you may adapt it to your needs. This code generator is defined in the `plugin.xml` file of the DemoReportGenerator plugin. In the `plugin.xml` file, the template based code generator is registered via the extension point `dreisoft.-tresos.launcher2.plugin.generator`.

In the folder `generate\reports`, a `.csv` file is stored, which is the template for the output file and uses code generator macros. For information on how to change these code generator templates, refer to the EB tresos Studio developer's guide, chapter Generating code/Template-based Code Generator API. Information on the XPath functions is available in the EB tresos Studio developer's guide, chapter Configuration models/XPath API.

7.8.6. Editing several table rows simultaneously

To change several rows of a table at the same time, EB tresos Studio provides the *Bulk Change* dialog. This *Bulk Change* dialog keeps you from having to change the same value in different rows manually by automating the value change over several rows.

- ▶ To bulk change values in several table rows, select all table rows in which you would like to change values. For details about tables and its entries, refer to [Section 6.4.1.4, “Tables”](#).
- ▶ Open the *Bulk Change* dialog.

To open the *Bulk Change* dialog, click on the  button in the table tool bar.

The *Bulk Change* dialog opens up.

- ▶ Alternatively, you may open the *Bulk Change* dialog from the context menu of a table.

To open the *Bulk Change* dialog via the context menu of the table, right-click on the table and select *Bulk Change...* from the context menu.

The *Bulk Change* dialog opens up.

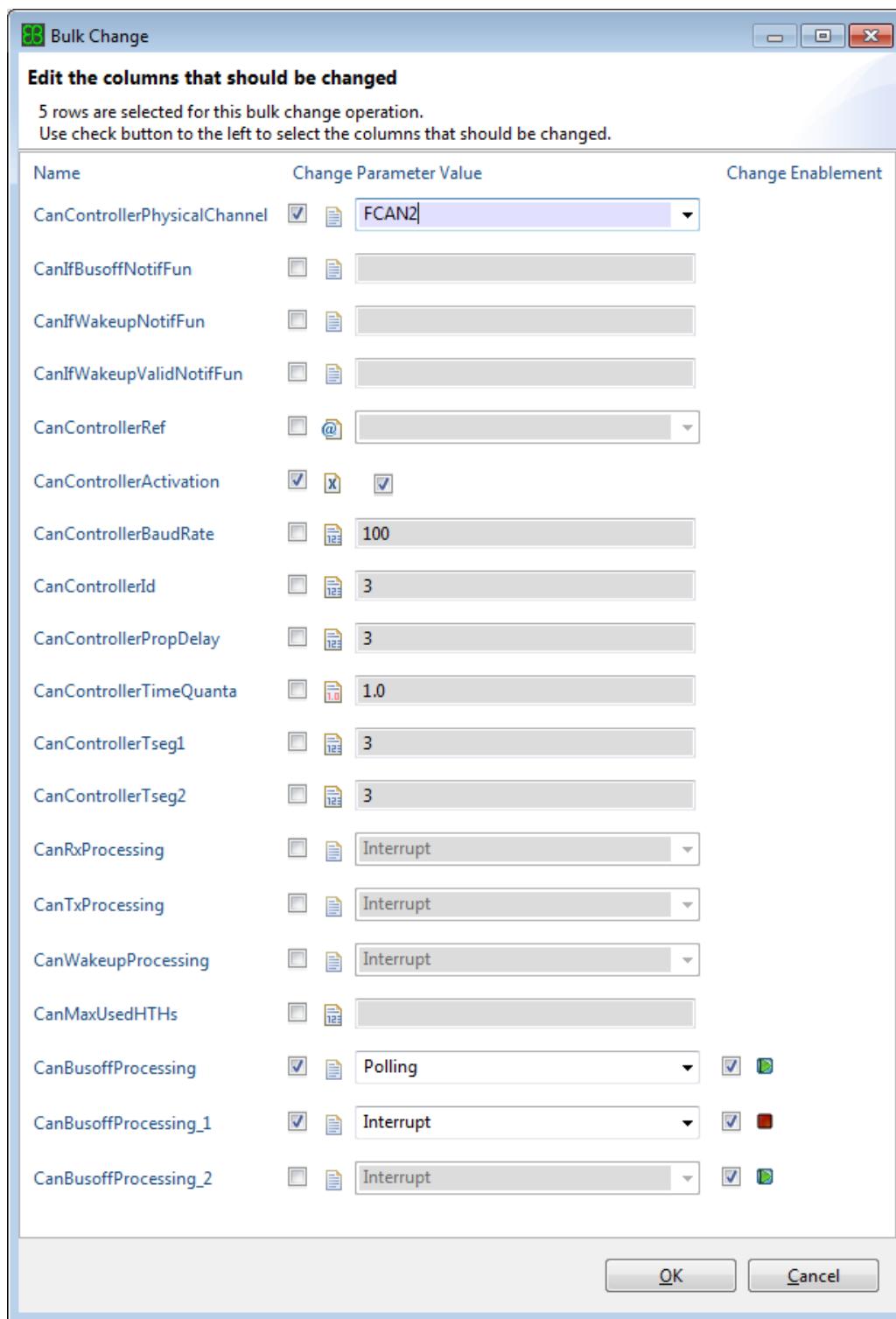


Figure 7.113. The Bulk Change dialog

The *Bulk Change* dialog provides an entry field for each parameter, including optional parameters, and parameters located inside optional containers, regardless if they are visible or not in the table. You can change the value of a parameter or the enablement, or both.

- ▶ To change which columns are displayed, use the tool bar of the table. For information on the tool bar of the table, refer to [Section 6.4.1.4, “Tables”](#).
- ▶ To edit the value of a field, click the checkbox listed under the section *Change Parameter Value*, which follows the name of the parameter.
- ▶ If the parameter is optional, the *Change Enablement* section is displayed. To change the enablement, click the checkbox below the section *Change Enablement* and the **enablement** button becomes active. This button has two states: the green state means enabled and the red state means disabled. The same representation is used in the Generic Editor.

The parameters are changed to the value *entered/selected* when you close the *Bulk Change* dialog.

- ▶ To execute the change, click on **Ok**.
- ▶ To cancel the operation, click on **Cancel**.
- ▶ To undo the operation, click on the undo button in the EB tresos Studio tool bar.

7.8.7. Autoconfiguring routine jobs

If you use the ECU Configuration Editor, you have to edit all values manually. This may be a tedious task. EB tresos Studio supports you performing those routine jobs by supplying *unattended* wizards. These wizards guide you through difficult configuration jobs and handle routine tasks that do not require user-interaction.

Each unattended wizard has its own graphical user interface which is suited for the wizard's task. You need to configure each unattended wizard before you can run it within your project.

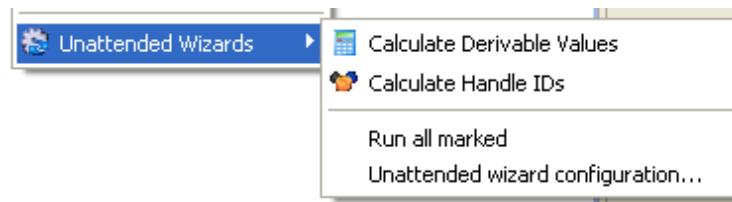
The list of available wizards can be extended by wizards that are designed for the modules installed. By default EB tresos Studio ships with two wizards: the *Calculate Derivable Values* and the *Calculate Handle IDs*. These two wizards are described in the following section. The module-specific unattended wizards are explained in the module's documentation.

7.8.7.1. Configuring unattended wizards

Before you can use unattended wizards, you need to configure them. To configure an unattended wizard:

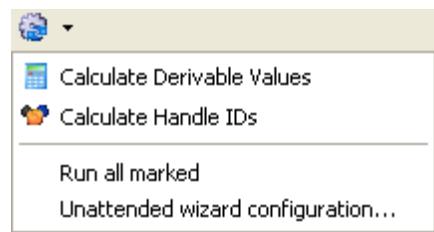
- ▶ In the *Project Explorer*, select the project on which you would like the wizard to operate.
- ▶ Alternative 1: In the menu bar, *Project* menu, select the *Unattended Wizards* item.

A submenu opens up.

Figure 7.114. The *Unattended Wizards* submenu

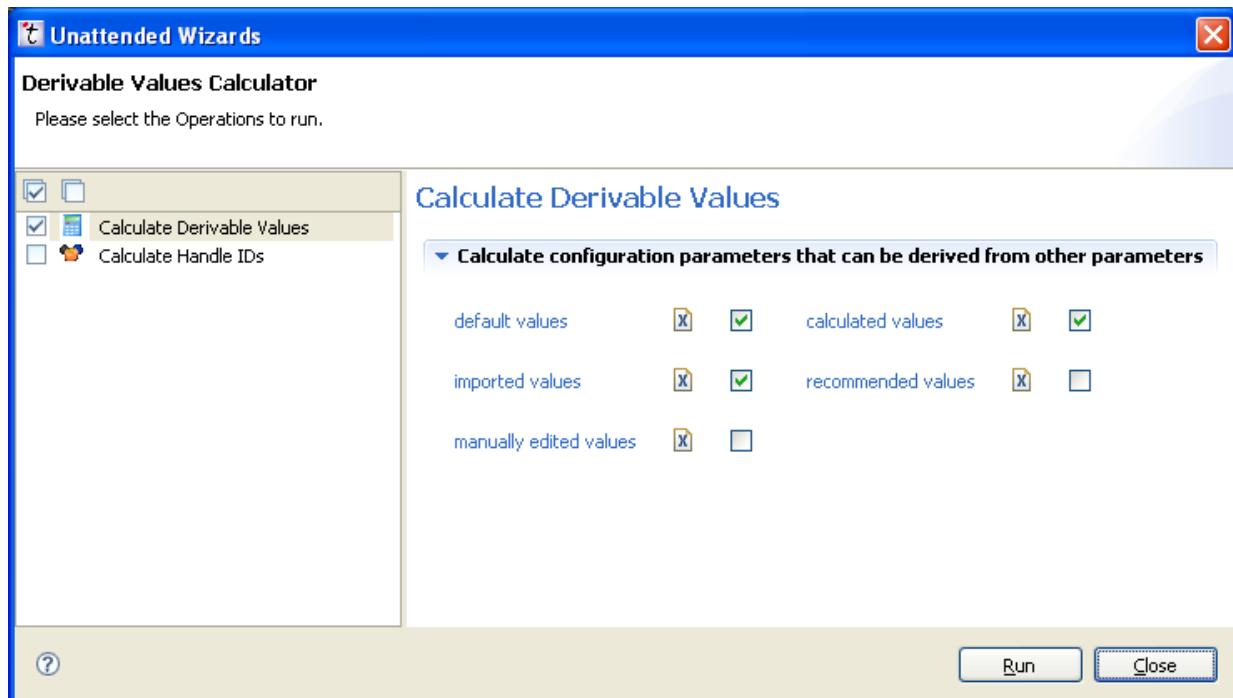
- ▶ Alternative 2: In the tool bar, select the arrow icon next to the gear button.

A menu opens up.

Figure 7.115. The *Unattended Wizards* menu

- ▶ Select *Unattended wizard configuration....*

The *Unattended Wizards* configuration dialog opens up.

Figure 7.116. Configuring unattended wizards with the *Unattended Wizards* configuration dialog

- ▶ On the left side, the dialog lists all available unattended wizards. Select the wizard you want to configure from this list. The right part of the dialog window now displays the user interface of the wizard. The appearance of the wizard depends on the wizard you have selected. If the wizard is not configured correctly, warnings or error messages are displayed at the top of the window.
- ▶ To close the *Unattended Wizards* configuration dialog, click on **Close**. All changes you made in the configuration will be saved when closing the *Unattended Wizards* configuration dialog.
- ▶ To run a wizard, refer to [Section 7.8.7.2, “Running unattended wizards”](#).

7.8.7.2. Running unattended wizards

You may either run one single wizard or several wizards in a row. In both cases, the output is displayed in the *Results* view after the wizard has finished the work.

- ▶ If you want to know how to run one single unattended wizard, go to [Section 7.8.7.2.1, “Running a single unattended wizard”](#).
- ▶ If you want to know how to run several unattended wizards in a row, go to [Section 7.8.7.2.2, “Running multiple unattended wizards”](#).
- ▶ If you want to find information on the *Results* view that opens up after an unattended wizard has finished running, go to [Section 7.8.7.2.3, “Result feedback”](#).
- ▶ Some unattended wizards can be run from the command line. You find instructions for launching unattended wizards from the command line in [Section 7.11.2.9, “Executing unattended wizards on projects”](#).

To read additional information such as command line options or prerequisites, consult the documentation of the unattended wizard you want to run.

7.8.7.2.1. Running a single unattended wizard

There are two ways to run a single unattended wizard:

Alternative 1:

- ▶ In the *Project Explorer*, select the project on which you would like the wizard to operate.
- ▶ In the tool bar, select the arrow button next to the gear button .

A menu opens up.



Figure 7.117. *Unattended Wizards* menu

- ▶ In the menu, select the wizard you want to run.

The wizard runs automatically.

- ▶ When the wizard has finished, the *Results* view opens up. To read about the *Results* view, proceed to [Section 7.8.7.2.3, “Result feedback”](#).

Alternative 2 provides an opportunity for you to configure the unattended wizard before running it:

- ▶ In the *Project Explorer*, select the project on which you would like the wizard to operate.

- ▶ In the tool bar, select the arrow button next to the gear button

A submenu opens up.

- ▶ In the submenu, select *Unattended wizard configuration....*

The *Unattended Wizards* configuration dialog opens up.

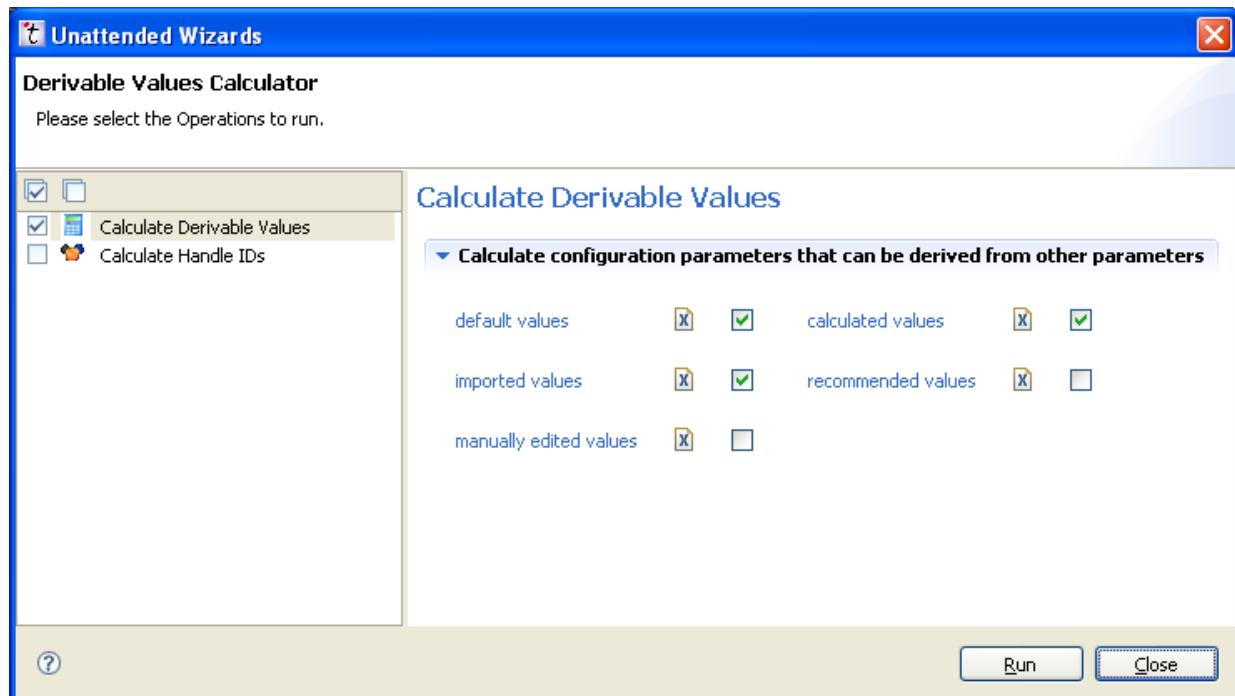


Figure 7.118. The *Unattended Wizards* configuration dialog

- ▶ In the dialog, highlight the wizard you want to run by selecting it from the list on the left side.
- ▶ The configuration options for the wizard are displayed on the right side of the dialog. You may change the configuration before running the unattended wizard.
- ▶ Click on the **Run** button to run the wizard.

After the wizard has finished, the *Results* view opens up. To read more about the *Results* view, proceed to [Section 7.8.7.2.3, “Result feedback”](#).

7.8.7.2.2. Running multiple unattended wizards

You may define a set of unattended wizards which then can be run one after another with only one command. This is also called *batched execution*.

NOTE**Configure the unattended wizards before running them**

You can only select the check box of unattended wizards you have configured before.

To define a set of wizards:

1. Open the *Unattended Wizards* configuration dialog.
 - ▶ In the *Project Explorer*, select the project on which you would like the wizard to operate.
 - ▶ In the tool bar, select the arrow button next to the gear button .
- A submenu opens up.
- ▶ In the submenu, select *Unattended wizard configuration....*

The *Unattended Wizards* configuration dialog opens up.

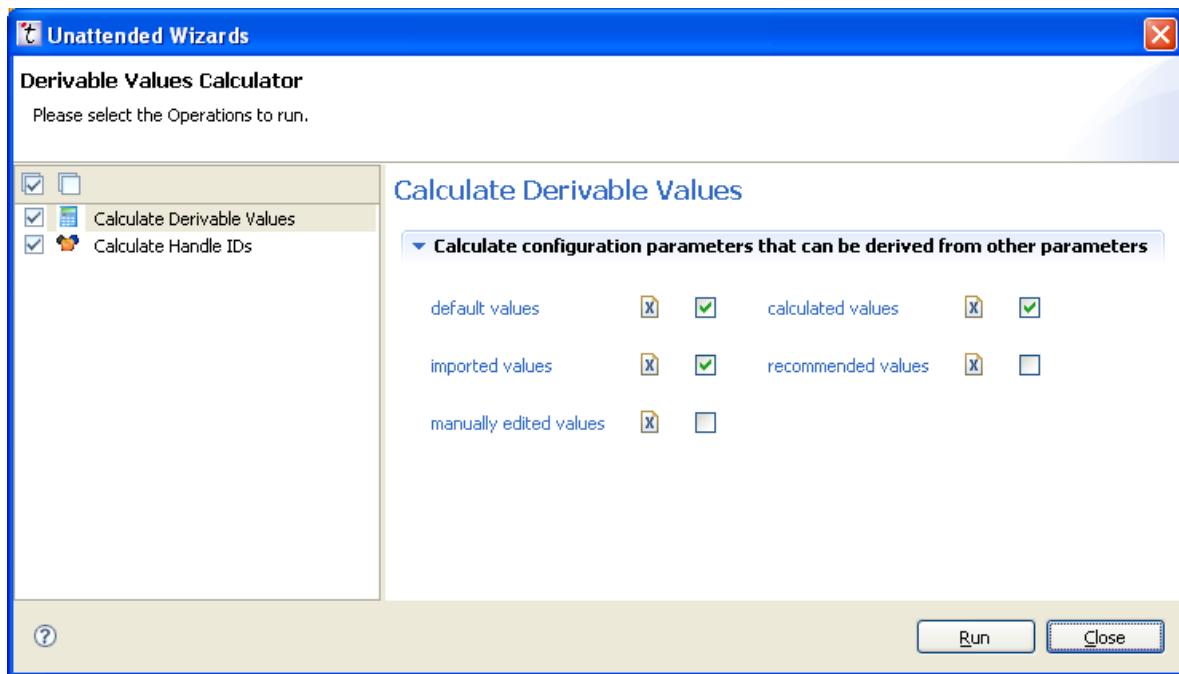


Figure 7.119. Defining the set of unattended wizards to run in a row

2. On the left side of the dialog you see a list of unattended wizards. For each wizard you want to include in the batched execution, select the corresponding check box.
3. Close the *Unattended Wizards* configuration dialog by clicking on the **Close** button. The state of the check boxes is saved automatically when you close the *Unattended Wizards* configuration dialog.

There are two ways to run the wizards you have just checked in the dialog.

Alternative 1:

- ▶ Make sure the configuration project on which you want the wizards to run is still selected in the *Project Explorer*.
- ▶ In the tool bar, select the arrow button next to the gear button .
- A menu opens up.
- ▶ In the menu, select *Run all marked*
- All selected wizards are run.

Alternative 2:

- ▶ Make sure the configuration project on which you want the wizards to run is still selected in the *Project Explorer*.
- ▶ In the tool bar, click on the gear button .

All wizards that you have checked earlier are run.

After the wizards have finished, the *Results* view opens up. For information on the *Results* view, proceed to [Section 7.8.7.2.3, “Result feedback”](#).

7.8.7.2.3. Result feedback

After one or more unattended wizards have completed execution, the *Results* view opens up.

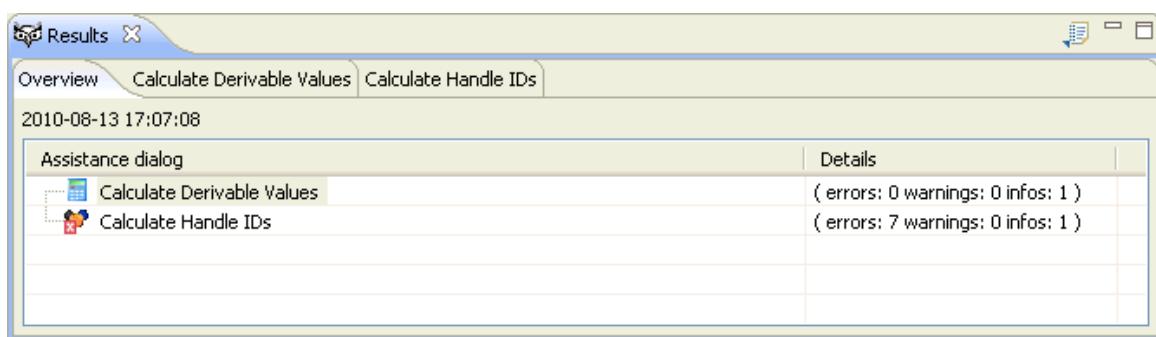


Figure 7.120. The *Results* view

The *Results* view displays the result of each wizard run. The view consists of several tabs. The exact amount of tabs and their content depends on the wizards that have been executed.

The leftmost tab always is the *Overview*. This tab displays a summary of the results of each wizard. In the image above, you see the results of two wizards, the *Calculate Derivable Values* and the *Calculate Handle IDs*. The *Calculate Derivable Values* has one *info* message as output. To view this message, click on the *Calculate Derivable Values* tab.

Every wizard that was executed has a result page in the view. To open the result page, click on the tab with the name of the wizard.

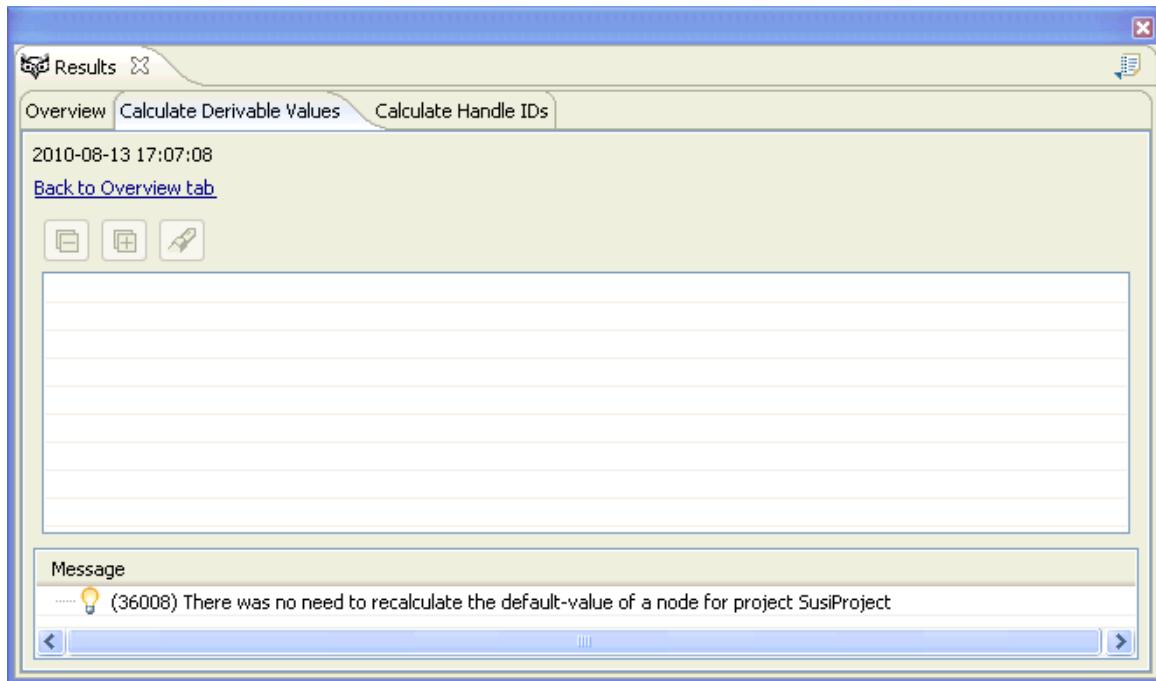


Figure 7.121. The *Results* view displays the results of an unattended wizard

The output of the unattended wizards is displayed as a list at the bottom of the window, which is labeled *Message*. This list contains all errors, warnings and info messages that occurred while the unattended wizards were running. Some unattended wizards display more detailed information as GUI elements in the middle of the *Results* view.

In the image above, the wizard does not display additional GUI elements, but you can see an info message at the bottom of the window.

To export the results of an unattended wizard:

- ▶ Click on the export button .

A dialog opens up, in which you may enter a filename and directory to save the results.

- ▶ Browse to the directory of your choice and enter a name for the exported file.
- ▶ Click on **Open**

The results of the unattended wizard are being exported.

7.8.7.3. Configuring the Calculate Derivable Values wizard

The *Calculate Derivable Values* recalculates the value of nodes which must have a default value. If you choose this wizard, the *Error Log* view displays a list of all values that have been recalculated or changed.

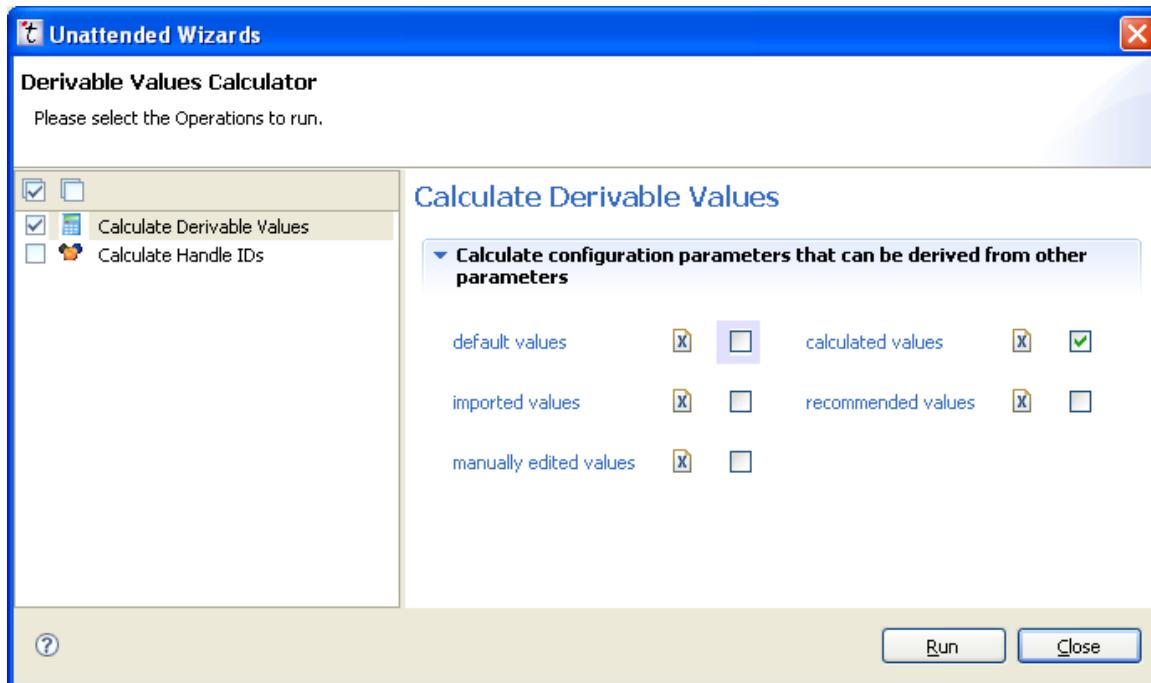


Figure 7.122. The Calculate Derivable Values wizard

In the *Calculate Derivable Values*, you may select which values you would like this wizard to recalculate:

Entry	Instructions
default values	To recalculate all values for which the module provider defined a default value calculation formula, select default values .
calculated values	To recalculate all values which have been calculated before, select calculated values . Calculated values are e.g. the values calculated by another unattended wizard, such as the <i>Calculate Handle IDs</i> .
imported values	To recalculate all values that have been overwritten by an import, select imported values .
recommended values	To recalculate all recommended values, select recommended values .
manually edited values	To recalculate all values you have been changed before, select manually edited values .

Table 7.1. Calculate values that contain

7.8.7.4. Configuring the Calculate Handle IDs wizard

The *Calculate Handle IDs* calculates the handle IDs for the modules of the communication stack.

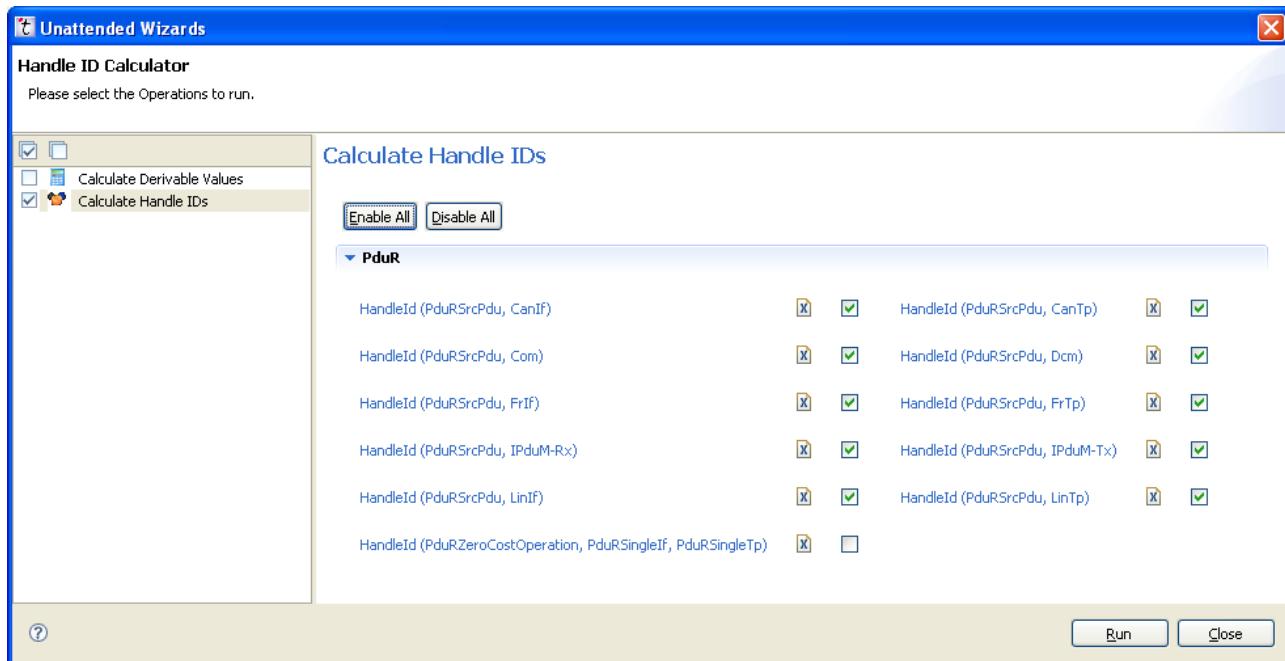


Figure 7.123. The Calculate Handle IDs wizard

The appearance of this wizard depends on the modules available in your project. There is a check box available for every handle ID parameter. If you check a check box, the handle IDs will be calculated for the corresponding parameter when the wizard is run.

7.8.8. Searching for configuration parameter names and their values

EB tresos Studio provides the option of searching for configuration parameter names and their values within files or within the data model.

There are two alternative ways to start the search.

Alternative 1:

- ▶ Press **Ctrl+T**
- ▶ The *Search* dialog opens up and the *ECU Configuration Search* tab is selected.

Alternative 2:

- ▶ Select the *Search* menu from the main menu.

A submenu opens up.

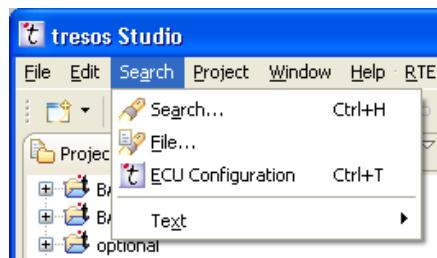
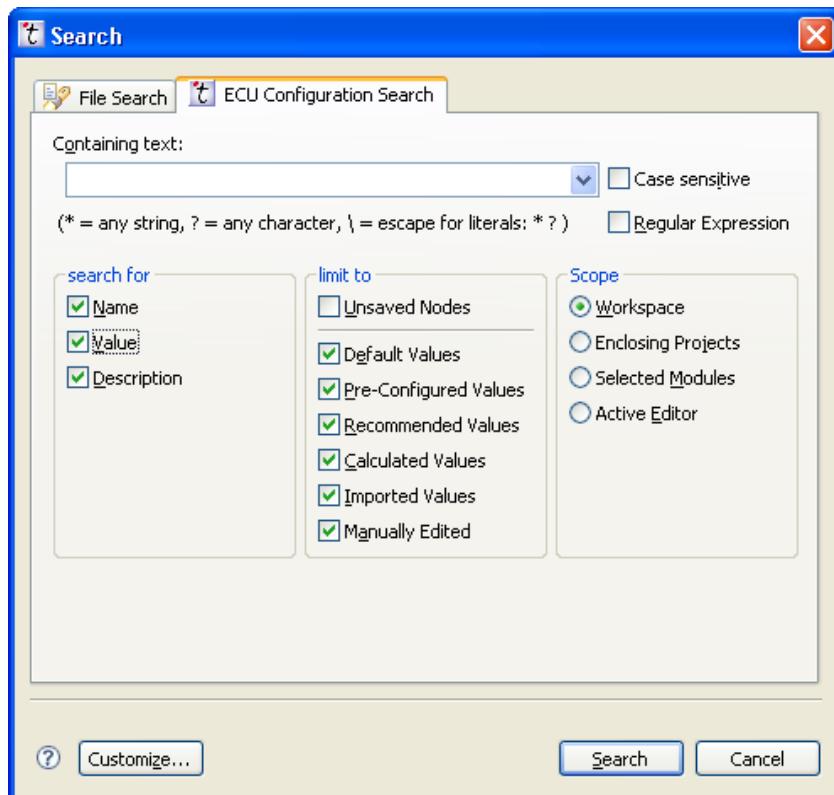


Figure 7.124. The Search submenu

- ▶ Select the *ECU Configuration* submenu.
- ▶ The *Search* dialog opens up and the *ECU Configuration Search* tab is selected.

Figure 7.125. The *Search* dialog

NOTE**Other tabs within the Search dialog are not used within EB tresos Studio**

The *Search* dialog also serves other purposes than to search within the data model of a project. Thus, it provides several other tabs. These tabs are contributed by the Eclipse framework and have no use in EB tresos Studio. Please consult the Eclipse documentation for more information.

7.8.8.1. Setting search options

To find configuration parameter names and their values within the data model, select the tab *ECU Configuration Search*. As you open the tab, the last search executed is automatically restored.

The *Search* dialog consists of the following elements:

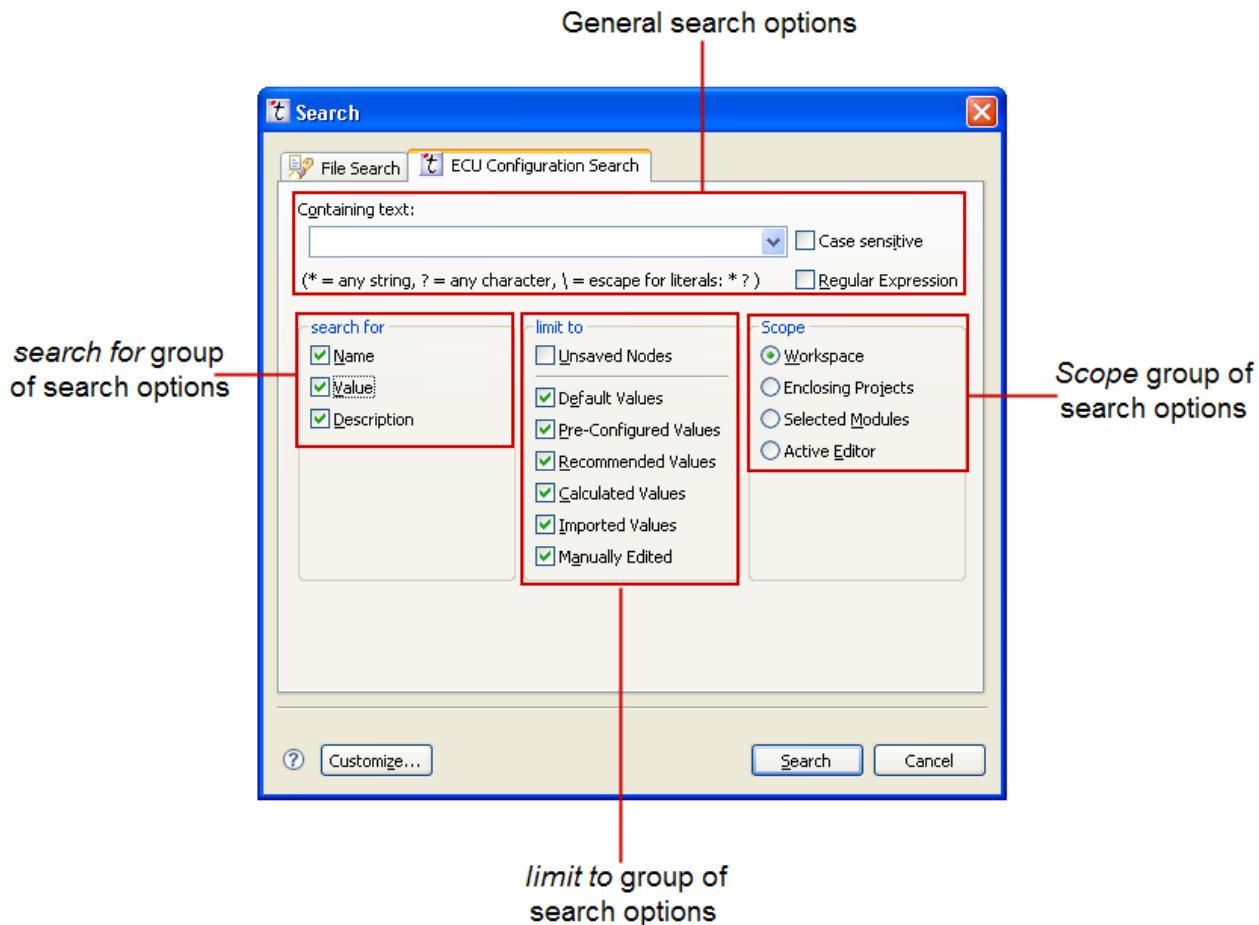


Figure 7.126. The *Search* dialog

7.8.8.1.1. Defining general search options

In the top part of the *Search* dialog, you may define general search options:

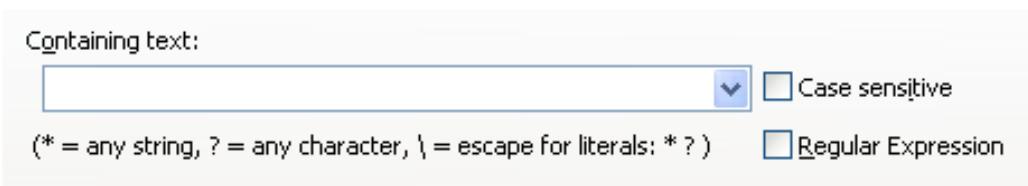


Figure 7.127. General search options

Field/button name	Instructions
Containing text	<ul style="list-style-type: none"> ▶ To restore an earlier search, select one of your last searches from the Containing text drop-down combo box. All options for the respective search are restored. ▶ To search for a new text string, select Containing text and type in your text. You may use the wild cards ? for characters and * for strings. ? and * can be escaped by using the \, e.g. *.
Case sensitive	To search for case sensitive text strings, select the Case sensitive check box. This option is not available for regular expressions.
Regular Expression	<p>To have the searched text interpreted as regular expression, select the Regular Expression check box.</p> <p>NOTE The Regular Expression button starts a content assist function  If you check the Regular Expression check box, a content assist function will start.</p>

7.8.8.1.2. Defining which kind of text is searched

In the bottom left part of the *Search* dialog, you may define the *search for* group of search options:

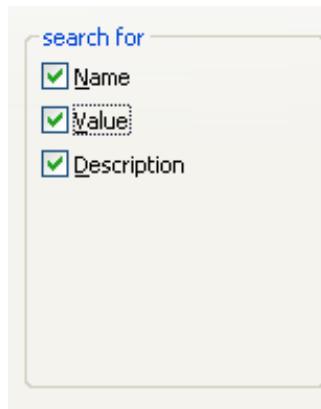


Figure 7.128. The *search for* group of search options

Field/button name	Instructions: Select the check box
Name	To search for the name of nodes.
Value	To search for the value of nodes.
Description	To search for node descriptions, i.e. the text you see in the <i>Element Description</i> view on the bottom right corner of the main window.

7.8.8.1.3. Limiting the kind of parameters that are searched

In the bottom middle part of the *Search* dialog, you may define the *limit to* group of search options:

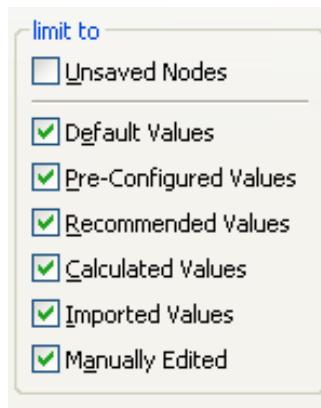


Figure 7.129. The *limit to* group of search options

Field/button name	Instructions: Select the check box
Unsaved Nodes	To restrict your search to unsaved nodes, i.e. those which have not yet been saved after an import.

Field/button name	Instructions: Select the check box
Default Values	To display all elements that are set to their default value.
Pre-Configured Values	To display all elements that are set to preconfigured values.
Recommended Values	To display all elements that are set to recommended values.
Calculated Values	To display all elements that are calculated.
Imported Values	To display all elements that are imported.
Manually Edited	To display all elements that are manually edited.

7.8.8.1.4. Defining the scope of the search

In the bottom right part of the *Search* dialog, you may define the *Scope* group of search options:

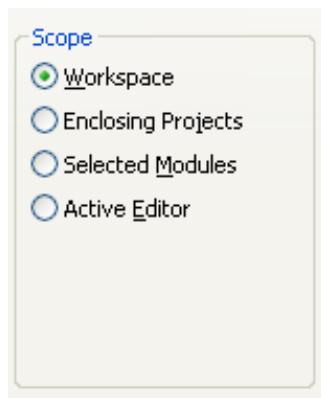


Figure 7.130. The Scope group of search options

Field/button name	Instructions: Select the option button
Workspace	To search in all open projects.
Enclosing Projects	To search in all projects whose tree contains a selection within the <i>Project Explorer</i> view. NOTE Projects are not included automatically  The project in the currently opened editor is not automatically included.
Selected Modules	To search for all modules whose parent- or sub-tree contains a selection within the <i>Project Explorer</i> .
Active Editor	To search within the module of the currently active editor.

Proceed to [Section 7.8.8.2, “Displaying search results”](#) to read information about starting the search.

7.8.8.2. Displaying search results

To display the search results, open the *ECU Configuration Search* in the *Search* dialog and select the search options of your choice.

- ▶ To start the search, select the **Search** button in the *Search* dialog.

The *Search* view opens up on the bottom of the GUI next to the *Error Log* and the *Problems* view. The *Search* view contains all the results of the search, i.e. the configuration nodes that contain the searched text.

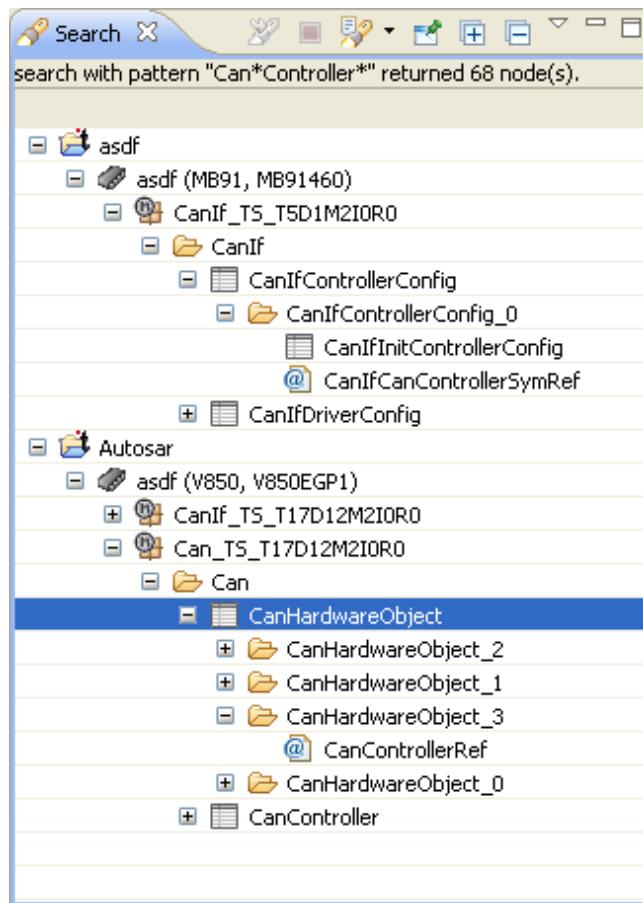


Figure 7.131. The Search view

The view displays a node tree for each project for which a configuration node was found. The upper part of the tree is the same as shown within the *Project Explorer* view. The lower part of the tree is the same as shown within the *Outline* view. Using the + and - icons at the top of the view expands or collapses the trees.

To open the *Editor* view of a node:

- ▶ Right-click on the node.

A context menu opens up.

- ▶ Select *Show in Editor*.

The configuration node is displayed in the *Editor* view.

To open the *Editor* view of a leaf node:

- ▶ Double-click on the node.

A context menu opens up.

- ▶ Select *Show in Editor*.

The configuration node is selected and displayed in the *Editor* view.

To read about how to configure the parameters with the ECU Configuration Editor, go to [Section 6.4.1, “The Generic Configuration Editor”](#).

7.9. Generating code for projects

After you have finished configuring a project, you can generate the source code of it. The resulting code depends on the values you have entered in the editor. A module only generates if the creator of the module registered a code generator for it.

NOTE**Your configuration project must not contain errors**

You can only generate code if the configuration project contains no errors.

7.9.1. Verifying a project

Some problems with the configured values of a project are only discovered when you generate the code of your project. Thus, before you actually generate the code, you may choose to verify the code first. During verification the same processing steps are executed as during code generation. But the difference is that no files are written to the disk.

The result of the verification is written to the *Error Log* view.

The verification is limited to parameters of the selected configuration time and to parameters you have configured before generating the project. This means:

- ▶ If your configuration time is `LINK` only, all `LINK` and `PRE-COMPIL` parameters are verified.

- ▶ If you have selected POST-BUILD in your configuration time settings, all parameters are verified.

For further information on configuration time and configuration classes, refer to [Section 4.5.2.1, “Configuration time of projects”](#).

NOTE**Your configuration project must not contain errors**

You can only verify your project if no error messages are shown in the *Problems View* view.

TIP**Do not block info messages during verification of code**

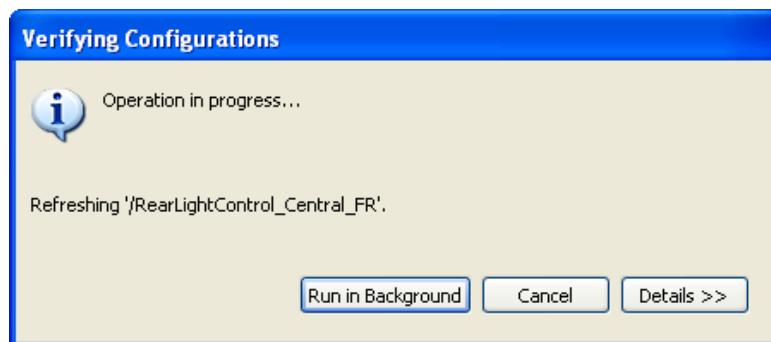
When verifying code, make sure that info messages are not blocked in the *Error Log*. Otherwise you might miss feedback.

There are two alternatives to verify a project:

Alternative 1: To start the generators to verify a project:

- ▶ Select an ECU configuration node within the project from the *Project Explorer*.
- ▶ Right-click on the ECU configuration node.
- ▶ Select *Verify Project*.

A dialog window opens up and starts the verification process. The window closes automatically when verifying the project is finished.



Alternative 2: To verify a project:

- ▶ Select an ECU configuration node within the project from the *Project Explorer*.
- ▶ Select the **Verify** button  from the tool bar.

A dialog window opens up and the tool starts verifying the project. The window closes automatically when verifying the project is finished.

7.9.2. Generating a project

If you generate code of your project, code is generated for all modules for which you have enabled code generation in the *Module Configurations* dialog (see [Section 7.6, “Editing module configurations”](#)).

The result of the generation is written to the *Error Log* view.

NOTE
Your configuration project must not contain errors


You can only generate code if the configuration project contains no errors.

NOTE
The code generator depends on the configuration time


The code generator checks whether the configuration time of your project fits to its own configuration time. This means that the configuration time of the project must be earlier or equal to the configuration time of the generator.

If a code generator does not define its own configuration time, disable the configuration time support in your project. Otherwise the code generator cannot start.

TIP

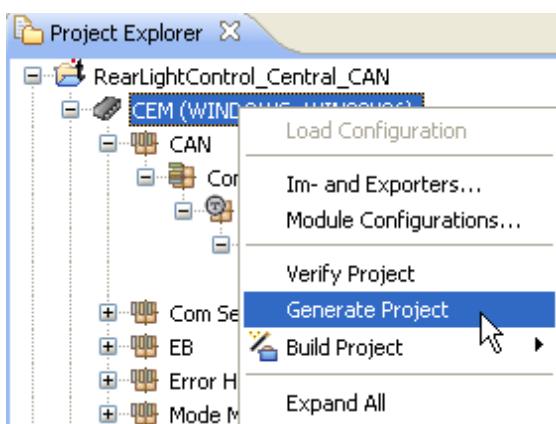
Do not block info messages during generation of code

When you generate code, make sure that info messages are not blocked in the *Error Log*. Otherwise you might miss feedback. Refer to [Section 6.3.5.2, “Filtering Error Log messages by their severity level”](#) for information on messages in the *Error Log*.

There are two alternatives to generate code:

Alternative 1: To generate code:

- ▶ Select an ECU configuration node within the project.
- ▶ Right-click on the ECU configuration node.
- ▶ Select *Generate Project*.



A dialog window opens up and starts generating code.

The generated code is written to the output folder set in the project preferences.

Alternative 2: Alternatively, to generate code:

- ▶ Select the **Generate** button  from the tool bar.

A dialog window opens up and starts generating code.

The generated code is written to the output folder set in the project preferences.

7.9.3. Building a project

To build a project means running the registered code generator engines in a specific mode which is different than the mode in which you generate code or verify the project. The available modes depend on the project you have currently selected and on the modules available in this project. The exact names of the generator modes and their outcome is controlled by the modules themselves. Please refer to the documentation of the modules present in your project.

The result of building the project is written to the *Error Log* view.

NOTE**The *Problems View* view must not show errors**

You can only build your project if the *Problems View* view shows no errors.

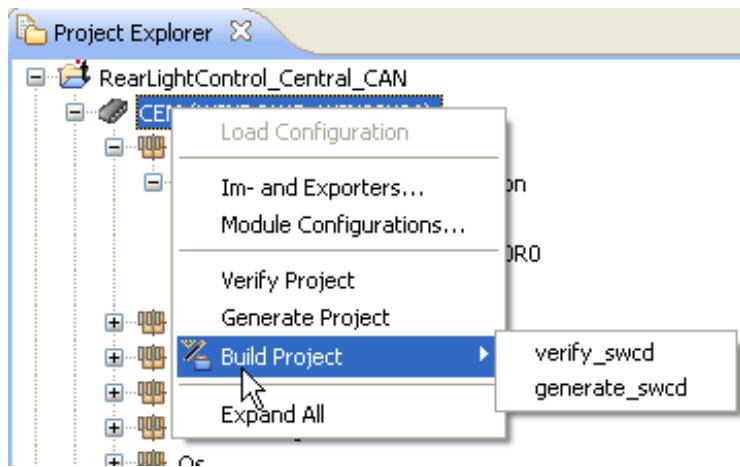
TIP**Do not block info messages when you build a project**

When you build a project, make sure that info messages are not blocked in the *Error Log*. Otherwise you might miss feedback. Refer to [Section 6.3.5.2, “Filtering Error Log messages by their severity level”](#) for information on messages in the *Error Log*.

There are two alternatives to build a project:

Alternative 1: To build code:

- ▶ Select an ECU configuration node within the project.
- ▶ Right-click on the ECU configuration node.
- ▶ Select *Build Project*.



A dialog window opens up and starts building code.

The generated code is written to the output folder set in the project preferences.

Alternative 2: Alternatively, to build code:

- ▶ Click the arrow next to the **Generate** button in the tool bar.

A context menu opens up. The menu items of this context menu depend on your project and on the modules available in this project.



- ▶ Select the mode you wish to run.

A dialog window opens up and starts building code.

The generated code is written to the output folder set in the project preferences.

7.10. Importing and exporting

This section presents the different ways of importing and exporting projects and configuration data.

7.10.1. Exchanging a complete EB tresos Studio project with other installations

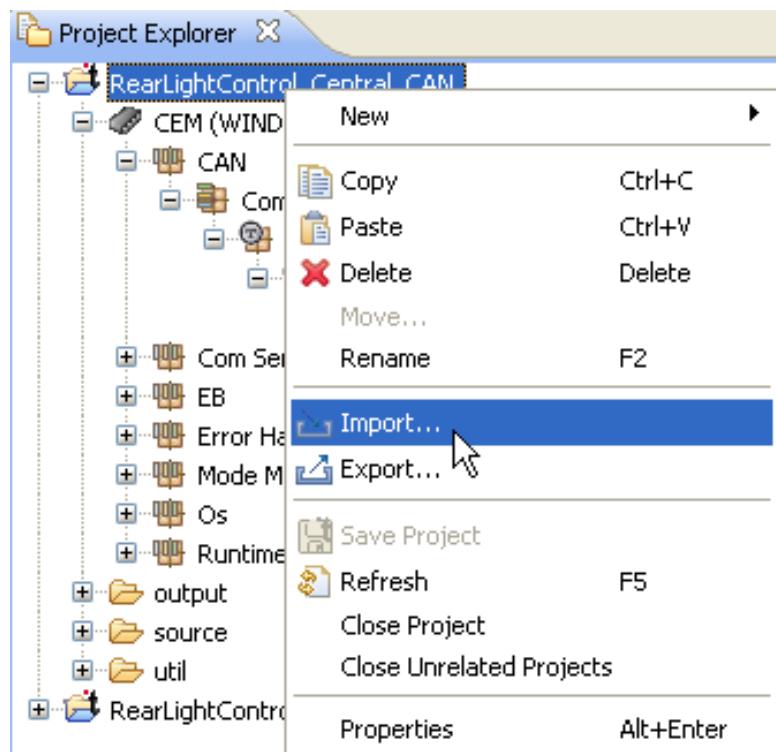
With EB tresos Studio you may import and export projects, e.g. to exchange a project with another EB tresos Studio installation, or to backup a workspace on external memory space. This feature is not the same as importing and exporting a configuration! To import or export a project configuration, please proceed to [Section 7.10.2, “Importing and exporting configuration data”](#). The project import and export is done through the Eclipse framework of EB tresos Studio

7.10.1.1. Importing a project

To import a project from other locations to the currently active workspace:

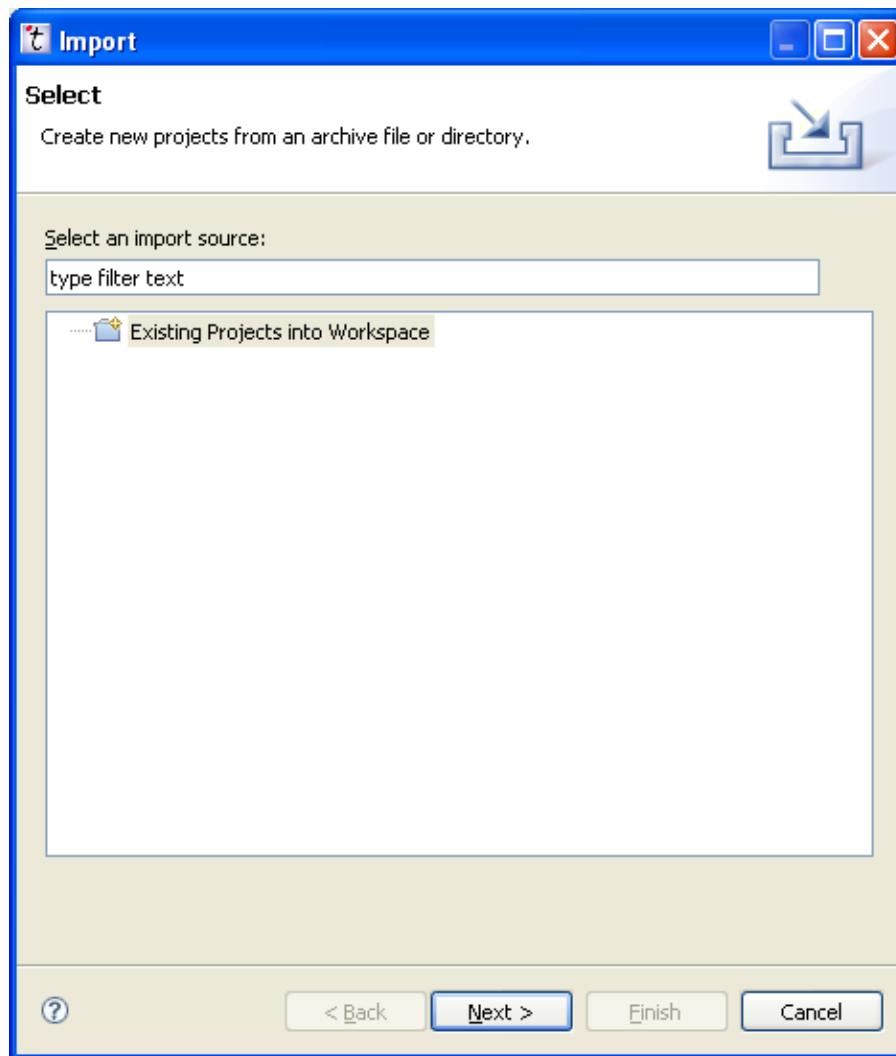
- ▶ Select the project node in the Project Explorer.
- ▶ Right-click on the node.

A context menu opens up.



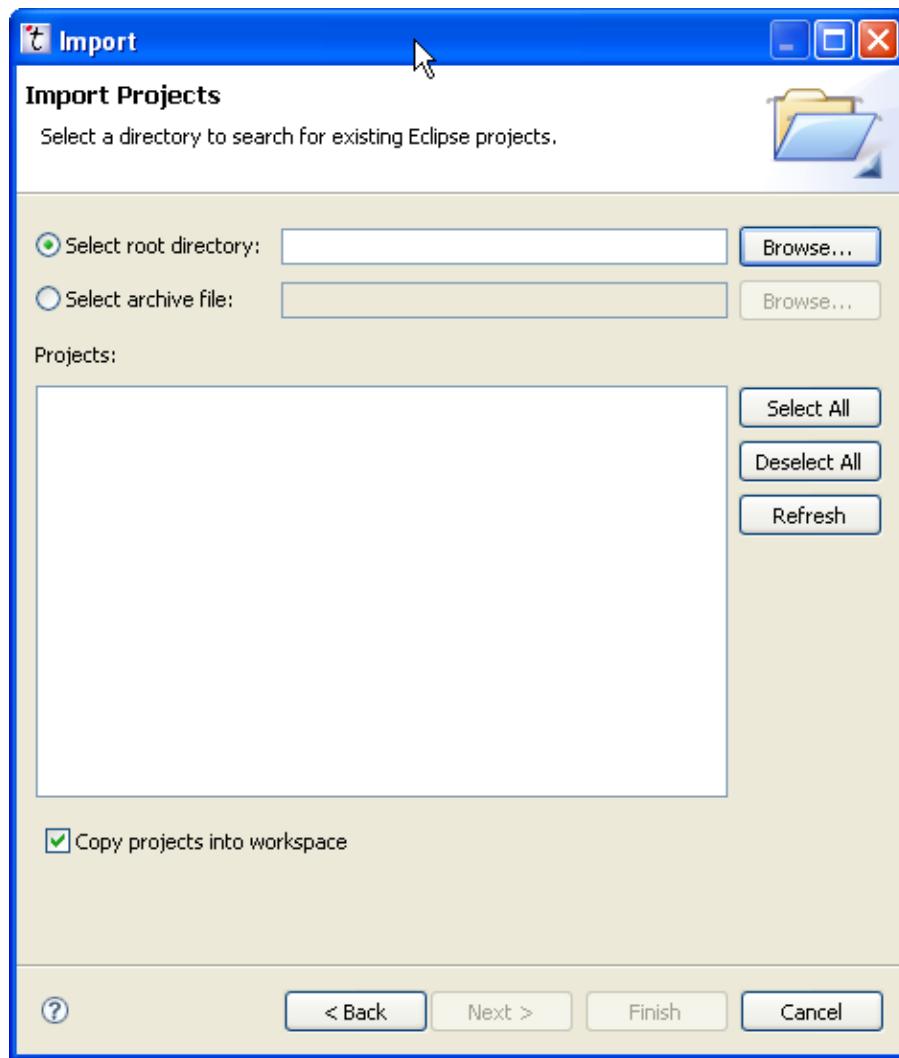
- ▶ Select **Import**.

The *Import* window opens up.



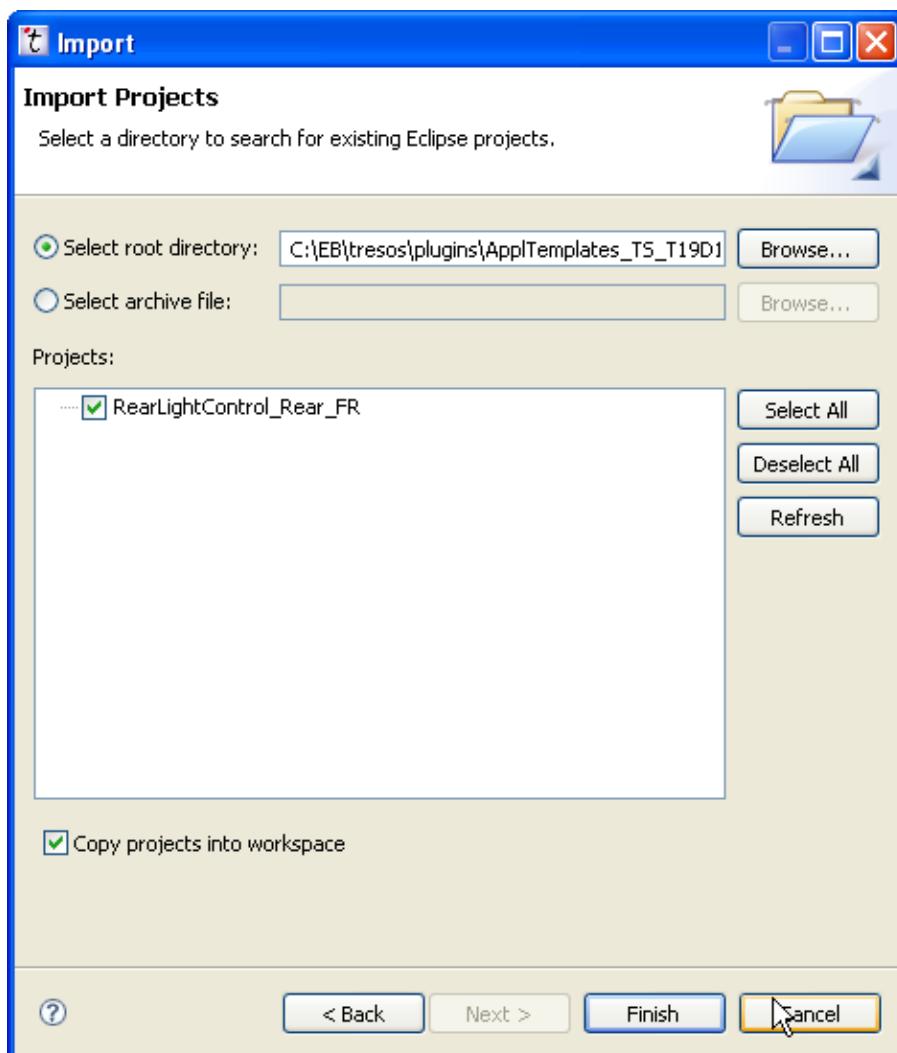
- ▶ Select *Existing Projects into Workspace*.
- ▶ Double-click *Existing Projects into Workspace*.

The *Import* window opens up.



- ▶ If you want to import a project that is located in the file system, click onto the radio button **Select root directory** and enter the path to the project's directory in the textbox. If you click the **Browse** button, you can choose an existing directory.
- ▶ If you want to import a project that is contained in an archive, click onto the radio button **Select archive file** and enter the path and name of the archive file in which the project is contained into the textbox. If you click the **Browse** button, you can choose an existing archive file.

All projects that can be imported from your source directory or archive are now listed in the *Projects* listbox. Note that a project is not listed if a project with the same name already exists in the currently active workspace.



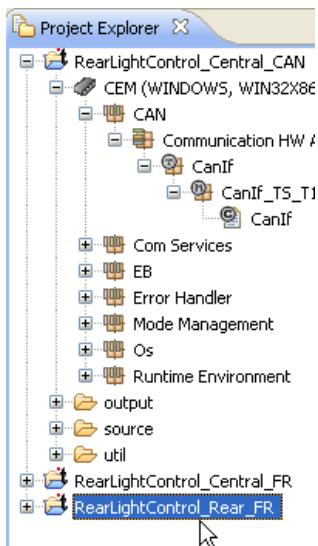
- ▶ Uncheck the projects you do not want to import from the list.
- ▶ To copy the project content into the currently active workspace, check the **Copy projects into workspace** checkbox. This option is only available if the import source is a directory and not an archive.

WARNING

If you do not check this box, EB tresos Studio generates a linked project. This means that you will later work on the files directly inside the folder from which you imported the project and not on a local copy inside your workspace.

- ▶ After you have chosen all projects that you want to import, click the **Finish** button to start the actual import.

The projects you imported are now listed as projects in the *Project Explorer*.

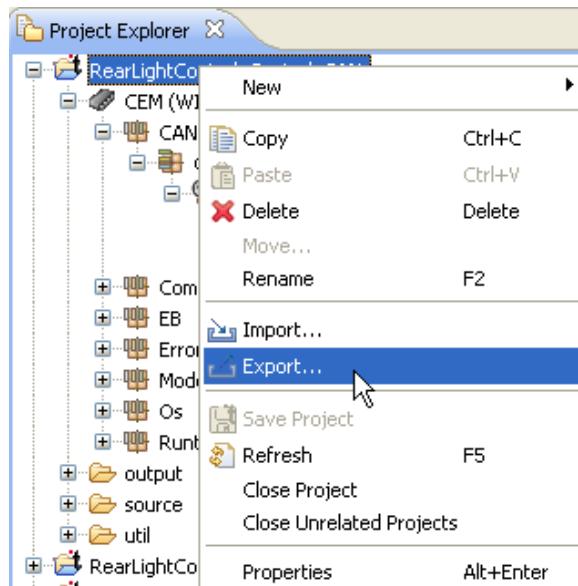


7.10.1.2. Exporting a project

If you want to share your project with other EB tresos Studio installations, or if you want to make a backup copy of a project, you can export the project. To export projects from the currently active workspace to other locations:

- ▶ Select the project node in the Project Explorer.
- ▶ Right-click on the node.

A context menu opens up.



- ▶ Select *Export*.

The *Export* window opens up.

Proceed with either of the following steps:

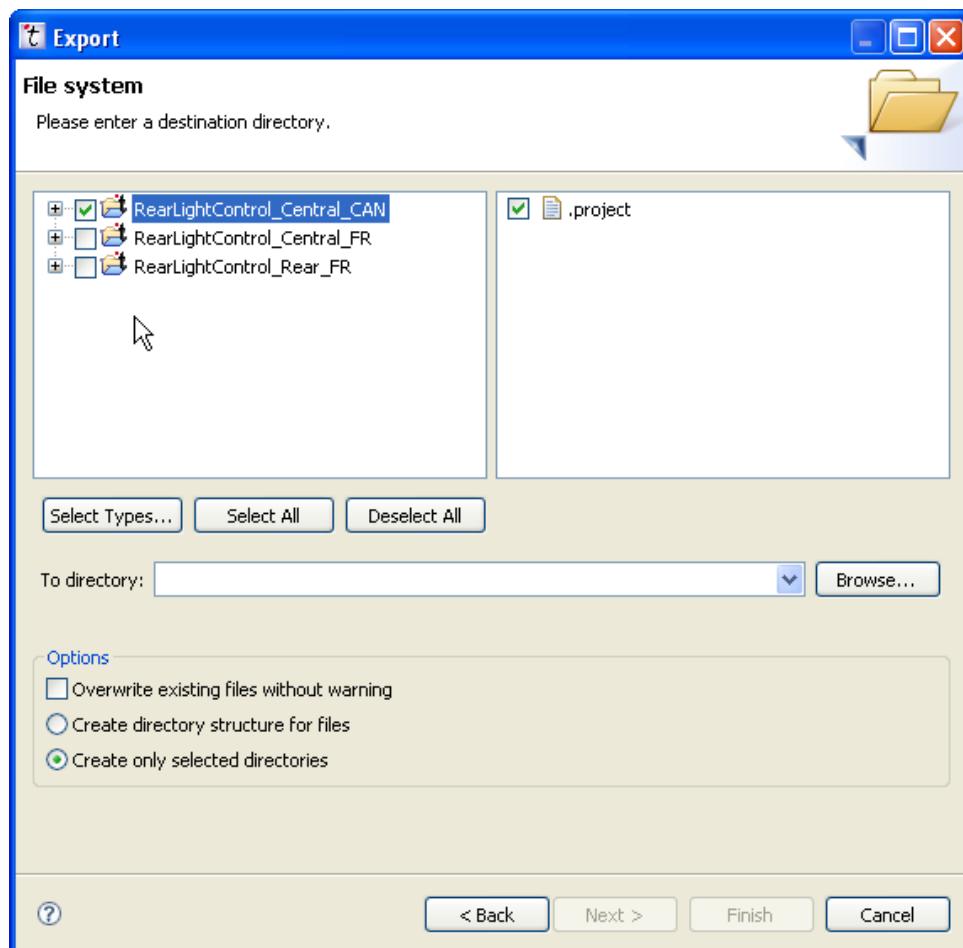
- ▶ [Section 7.10.1.2.1, “Exporting a project as copy”](#) exporting a project as copy,
- ▶ [Section 7.10.1.2.2, “Exporting a project as archive”](#) or exporting a project as archive.

7.10.1.2.1. Exporting a project as copy

To copy the EB tresos Studio workspace data to another location, export the project as directory structure.

- ▶ In the Select window, select *File System* as export destination.
- ▶ Click the **Next** button.

The *File system* page is shown.



- ▶ In the list on the left of the dialog you can see all projects of your current workspace. Check all the projects you would like to export. In case you want to exclude some parts of the project, you can also expand the

project in the tree and check or uncheck the contained resources. This is not recommended, as partially exported projects are usually not fully functional.

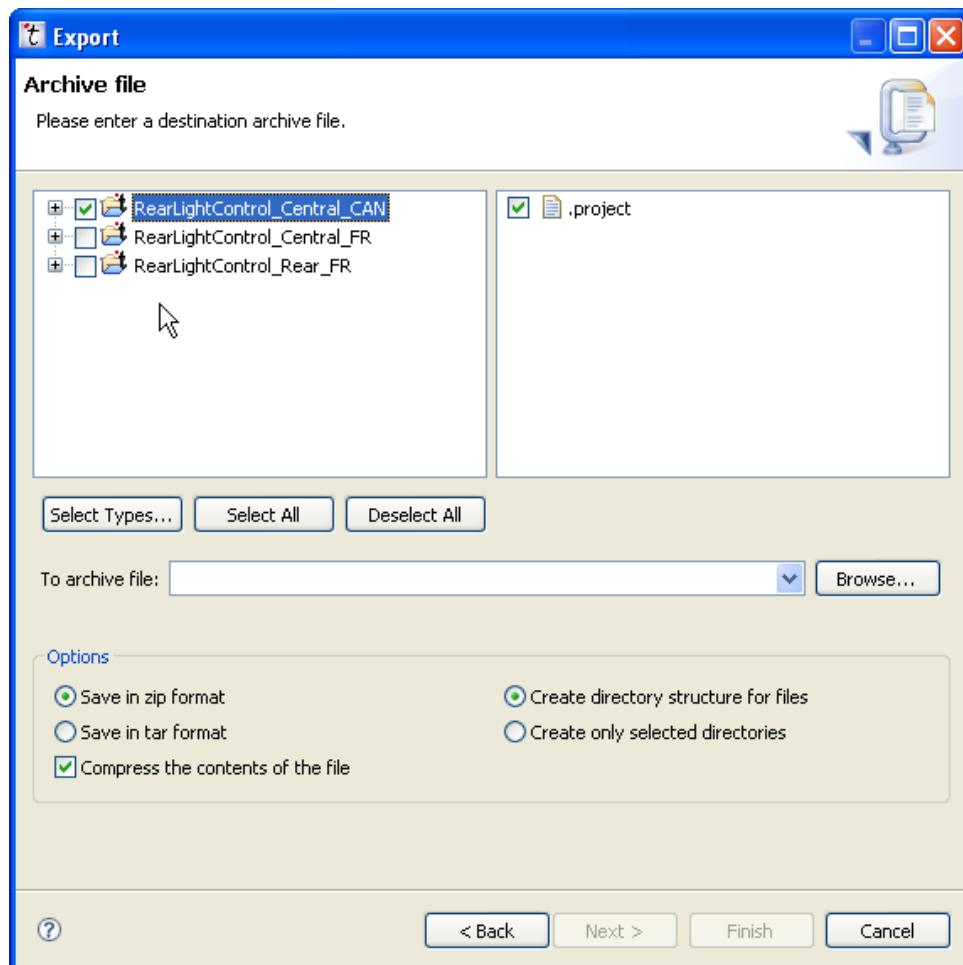
- ▶ Enter the directory into which you want to export the projects into the dropdown textbox *To directory*. You can also use the **Browse** button to choose a directory.
- ▶ In *Options*, select how you would like the project export to be performed.
- ▶ To export the projects, select **Finish**.

7.10.1.2.2. Exporting a project as archive

To store one ore more complete EB tresos Studio projects as compressed archive, export them as archive. This may be useful, e.g. if you want to exchange a project with another EB tresos Studio installation.

- ▶ In the *Select* window, select *Archive File* as export destination.
- ▶ Click the **Next** button.

The *Archive file* page is shown.



- ▶ In the list on the left of the dialog you can see all projects of your current workspace. Check all the projects you would like to export. In case you want to exclude some parts of the project, you can also expand the project in the tree and check or uncheck the contained resources. This is not recommended, as partially exported projects are usually not fully functional.
- ▶ Enter the filename of the archive into the dropdown textbox *To archive file*. You can also use the **Browse** to choose a file.
- ▶ In *Options*, select how you would like the file system export to be performed.
- ▶ To export the projects, select **Finish**.

7.10.2. Importing and exporting configuration data

Importers and exporters are used to import and export from and to the configuration data of a project from file formats such as the AUTOSAR ECU Configuration, OIL-format, DBC-format, LDF-format, and Fibex-format. Importing and exporting of complex project configurations includes selecting a file to store data to or load from, selecting a subset of the configuration parameters, selecting name prefixes, etc.

7.10.2.1. Step-by-step guide

Before you can export or import configuration data you need to create an export or import configuration for it. This configuration enables you to run the importer with the same settings later again. It stores all the settings you have entered for that particular import or export, and you can run that importer or exporter later again just by selecting it from a list.

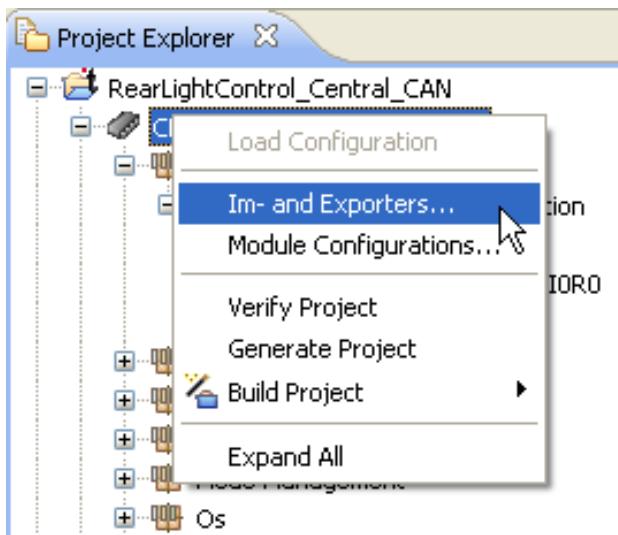
NOTE

Do not confuse import or export configurations with configuration projects. Importing and exporting of whole projects is covered in [Section 7.10.1, “Exchanging a complete EB tresos Studio project with other installations”](#)

7.10.2.1.1. Step 1: Using the Create, manage and run im- and exporters window

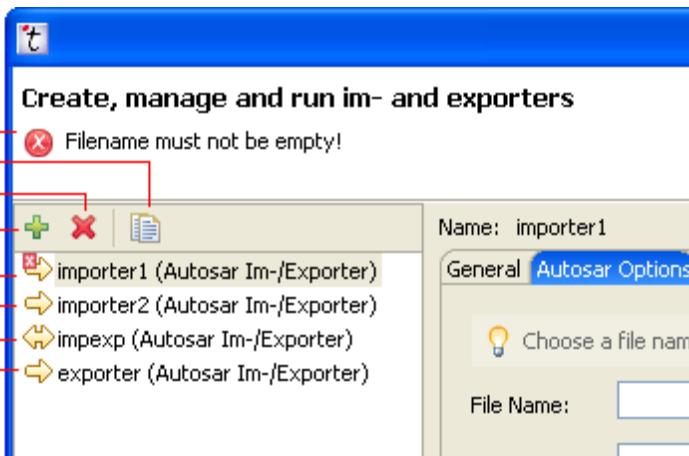
To create an import or export configuration for a configuration project:

- ▶ Select the ECU configuration node of the project you want to import to.
- ▶ Make sure your project is loaded. if not, load your project first (see [Section 7.3.6.3, “Loading a project”](#)).
- ▶ Right-click on the ECU configuration node.



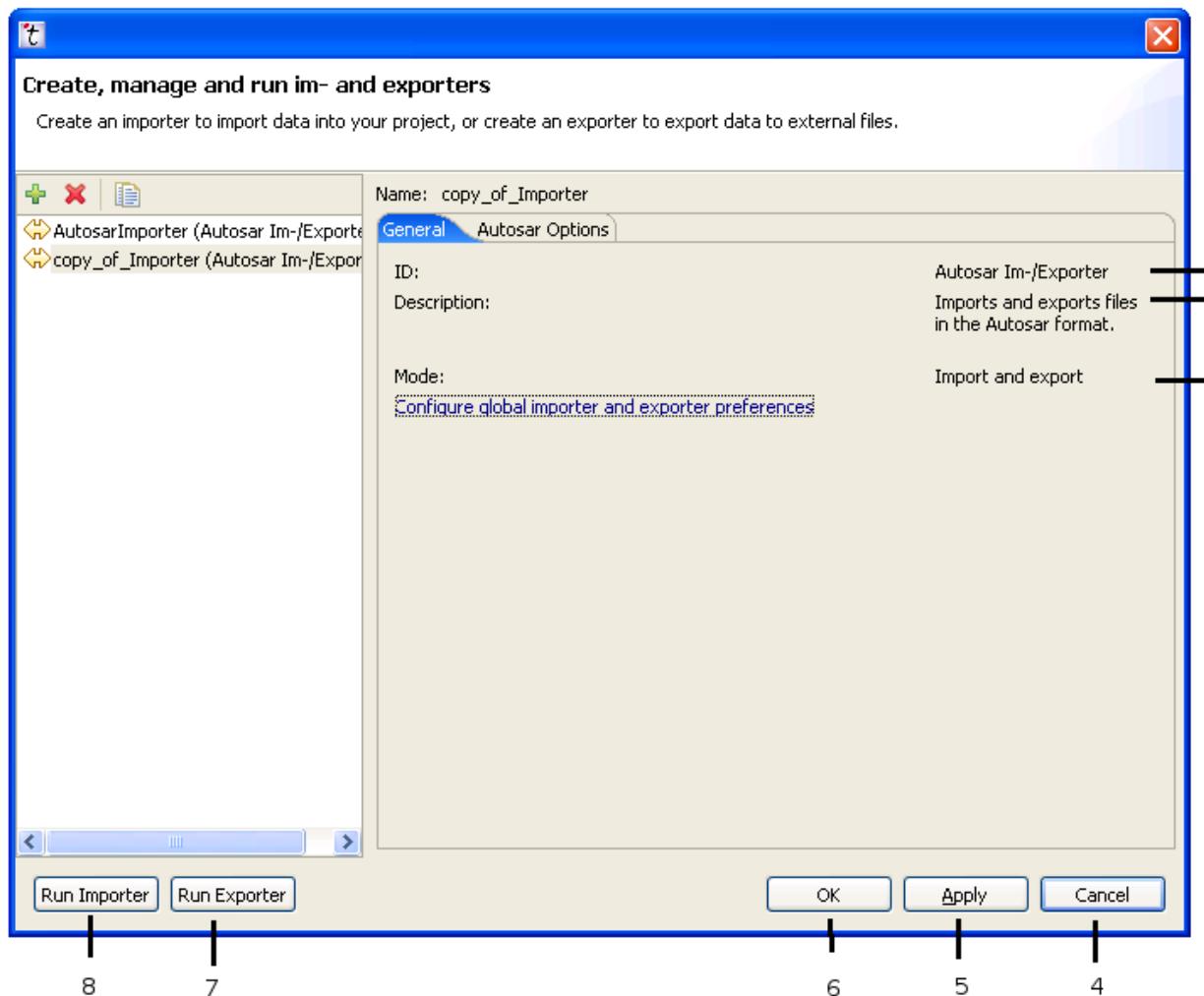
- ▶ Select *Im- and Exporters*.

The window *Create, manage and run im- and exporters* opens up.



Number	Icon	Explanation
1		If there is a problem with the importer or exporter configuration, it is shown here. If there are no problems, this section just contains a describing text.
2		Duplicate the selected import/export configuration.
3		Deletes the selected import/export configuration.
4		Adds a new import/export configuration.
5		The red cross in the top left corner indicates that this configuration contains an error.

Number	Icon	Explanation
6	→	This configuration will import data into EB tresos Studio.
7	↔	This configuration will be able to do both: import and export data.
8	←	This configuration will export data from EB tresos Studio into other formats.



Number	Menu option/button	Explanation
1	<i>ID</i>	Type of the importer/exporter. Determines the format that is being imported or exported as well as the overall functionality of the importer/exporter.
2	<i>Description</i>	Describes the importer/exporter type that is currently selected.
3	<i>Mode</i>	The configuration mode. Modes can be: ▶ import

Number	Menu option/button	Explanation
		<ul style="list-style-type: none"> ▶ export ▶ import and export
4	Cancel	Discards all changes that have been made since you opened the window or since you clicked the Apply button. Closes the window without asking if you want it closed.
5	Apply	Saves changes to the settings. The window stays open.
6	OK	Saves changes to the settings and closes the window.
7	Run Exporter	Exports data into another format by using the selected export configuration.
8	Run Importer	Imports data into EB tresos Studio by using the selected import configuration.

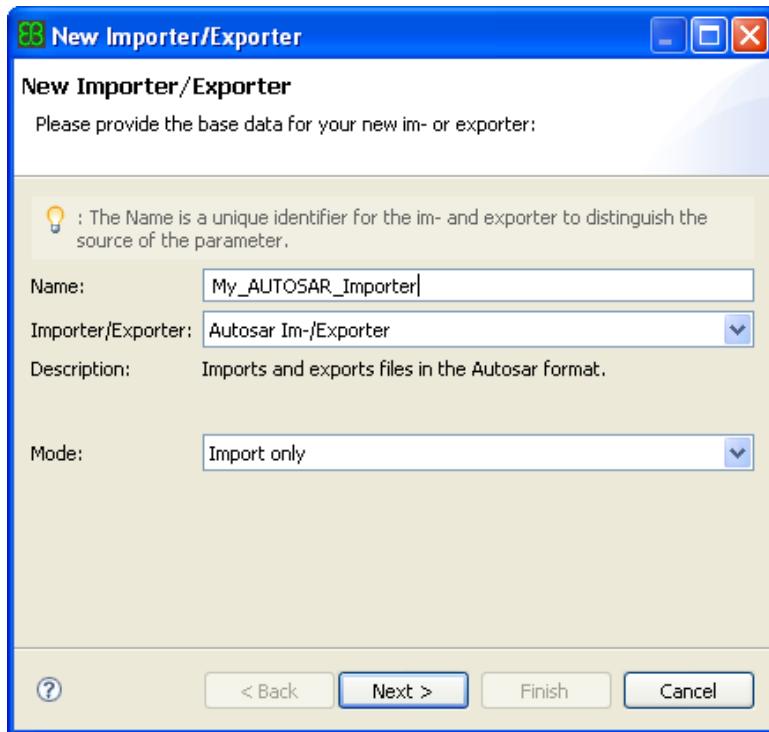
If you select one of the importer or exporter configuration on the left, the main part of the window displays the data of that configuration. The layout and the available GUI depends on the importer or exporter which you selected. All importers and exporters have an *Overview* tab which displays general information. Depending on the type, there may also other tabs be present.

7.10.2.1.2. Step 2: Creating an export or import configuration

In the window Create, manage and run im- and exporters:

- ▶ Select the **Create a new im- or exporter** button.

The *New Importer/Exporter* window opens up.



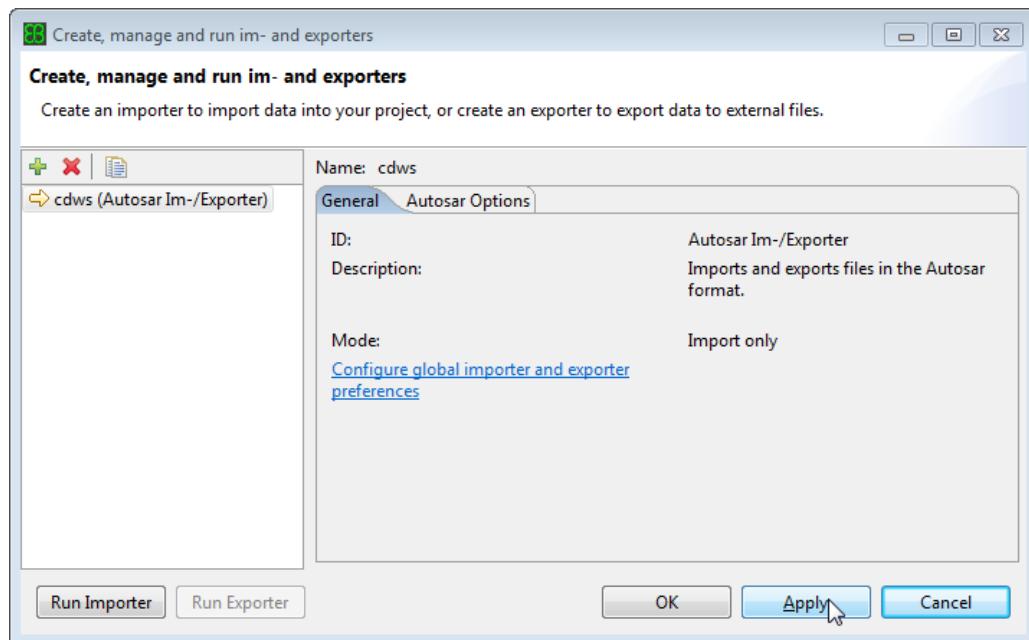
- ▶ Enter a name for your new configuration into the *Name* textbox.
- ▶ Select the type of importer or exporter from the *Importer/Exporter* dropdown listbox. The *Description* text displays a short text explaining what the selected importer or exporter does.
- ▶ Select the mode (import, export, or import and export) from the *Mode* dropdown listbox.
- ▶ Select **Next**.
- ▶ Depending on the importer or exporter, further pages may open up into which you may enter importer- or exporter-specific data.

For further information please refer to the chapter of the respective importer or exporter.

- ▶ To switch between pages, press the **Back** button or the **Next** button.
- ▶ To finish creating the importer or exporter configuration, select the **Finish** button.
- ▶ To dismiss the entered configuration, select the **Cancel** button .

7.10.2.1.3. Step 3: Saving changes to an import or export configuration

After creating the importer or exporter configuration, you can still make changes to it in the *Create, manage and run im- and exporters* window. To save any changes you made:



- Select **Apply**.

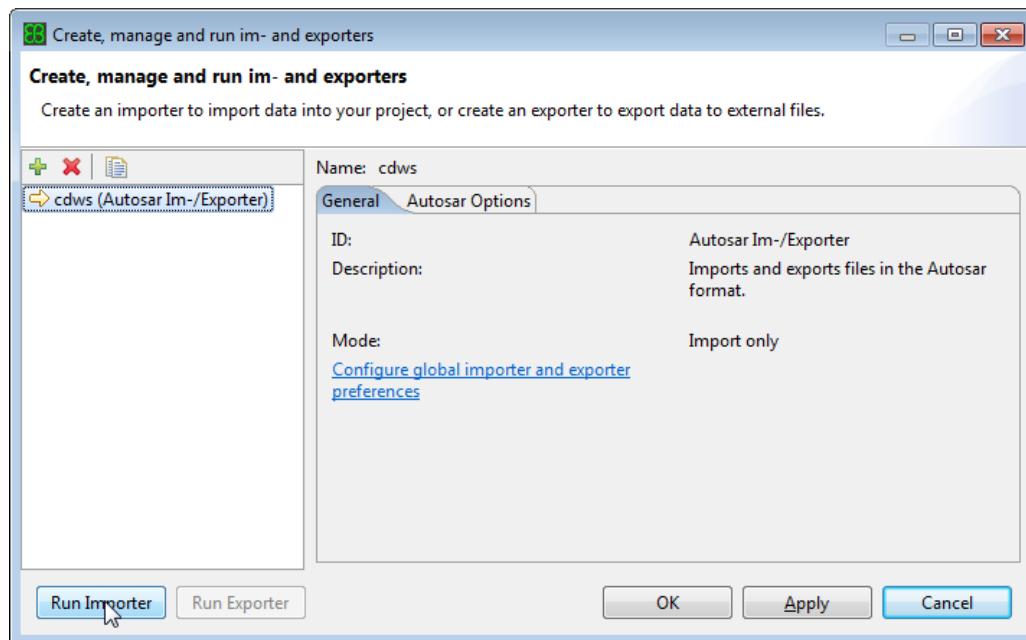
To discard changes, select **Cancel** (instead of **Apply**).

7.10.2.1.4. Step 4: Importing or exporting project configuration data

To import or export configuration data:

- Select the import or export configuration which you want to run from the list on the left side, i.e. a configuration you just created.

Depending on whether the selected configuration is set to run in import, export, or in both modes, the **Run Importers** and **Run Exporters** buttons are enabled or disabled.



- ▶ Select **Run Importer** or **Run Exporter**, depending on the task you wish to perform.

A status window opens up to inform you about the progress of your task. Then a message box provides feedback whether the importer or exporter has run successfully or not.



Possible feedback types are:

- ▶ Success
- ▶ Warning
- ▶ Error

In case of error or warning feedback, EB tresos Studio logs the feedback to the error log (see [Section 6.3.5, "Error Log view"](#)).

7.10.2.2. Importing and exporting configuration data from and to the AUTOSAR format

The EB tresos Studio GUI allows you to read and write files from and to the AUTOSAR format. For details about file content types and AUTOSAR content types, please refer to chapter [Section 4.5.3.2, “Content types for AUTOSAR module configuration files”](#)

7.10.2.2.1. How the import works

The configuration data imported from an AUTOSAR file is added to the existing data of the project.

- ▶ Not yet existing elements are added.
- ▶ Already existing elements with the same AUTOSAR SHORT-NAME path are overwritten.
- ▶ When overwriting a list of elements without SHORT-NAME (e.g. a list of integer values), each new element overwrites the existing element at the same position within the list. If the existing list contains more elements, the last elements which are not overwritten stay unchanged.
- ▶ If an importer is used a second time (or an importer with the same name), elements that have been removed from the import file are removed from the project, too.

The AUTOSAR importer is not able to do any of the following things:

- ▶ remove elements (which have not been imported before) from the project
- ▶ set an element to its default value
- ▶ enable an optional disabled element without changing its value
- ▶ disable an optional element

7.10.2.2.2. Importing and exporting configuration data

To import or export project configuration data in the AUTOSAR format:

- ▶ Follow the steps in chapter [Section 7.10.2, “Importing and exporting configuration data”](#) to Step 2.
- ▶ In the *New Importer/Exporter* window, select the Importer/Exporter type *Autosar Im-/Exporter*.
- ▶ Select whether you want to import, export or do both:

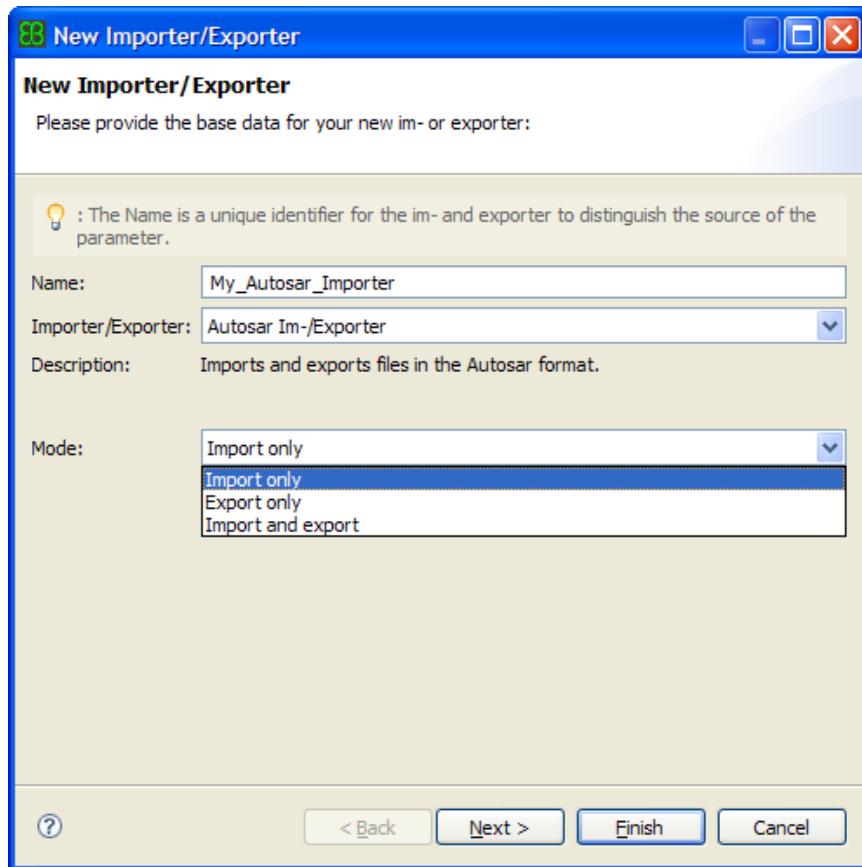


Figure 7.132. Selecting the import/export mode

NOTE**Configuration classes are always disabled in the *Import and export* mode**

In the mode *Import and export* of the *Autosar Im/Exporter* the configuration class handling option is always disabled.

- ▶ Click **Next**.

The *Autosar options* page opens up.

TIP**You may change these values later on as well**

You may change the values of the *Autosar Im-/Exporter* later on in the *Create, manage and run im- and exporters* dialog as well.

Depending on the selected import/export mode, the *Autosar options* page differs in some options. However, the first three options are always the same:

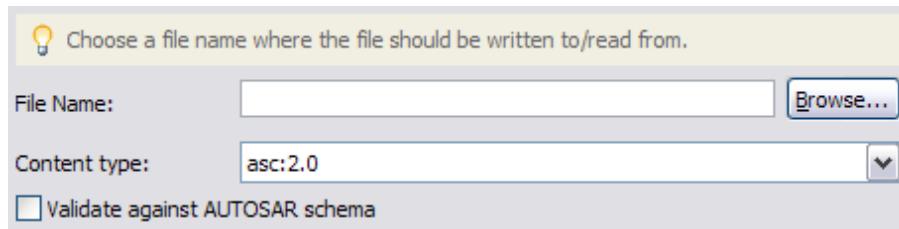


Figure 7.133. Common Autosar *Im/Exporter* options

- ▶ Enter the file name of the file on which the importer or exporter operates into the *File Name* text box. When you click the **Browse** button, a file appears that lets you choose a file.

If you have selected import mode, the filename specifies the file from which the data will be imported. If you have selected export mode, the file name specifies the file into which the exported data will be written.

- ▶ Select the content type of the selected file from the *Content type* drop-down list box. Available content types depend on the release version of your project and the importer/exporter mode:
 - ▶ *Mixed release versions*: If your project's release version is mixed, all content types are available.
 - ▶ *Import only mode*: Choose a content type lower or equal to your project's release version number.
 - ▶ *Export only mode*: Choose a content type equal or higher than your project's release version number.
 - ▶ *Import and export mode*: Use only the content type of your project's release version. This means, if the release version of your project is e.g. 2.0, you cannot import files with content type asc:2.1, but you can export to this format.

For more information on content types, see [Section 4.5.3, “File content types”](#)

- ▶ To enable validation, select the check box **Validate**:
 - ▶ When importing a file, the file will be validated against the XSD schema of the selected content type. If the file is not successfully validated, the file is not imported or exported and an error message opens up.
 - ▶ When exporting a file, the exported file will be written according to the selected content type. Additionally, the file will be validated against the XSD schema of the selected content type.

The configuration class support acts differently in import mode and export mode. In *Import and export mode*, the configuration class handling is always disabled.

Configuration class handling in import mode

How configuration classes are handled, depends on whether you have selected import mode, export mode or import and export mode. In this section you find instructions for configuring configuration classes in import mode. On the *Autosar Options* page of the *Autosar Importer and Exporter*.

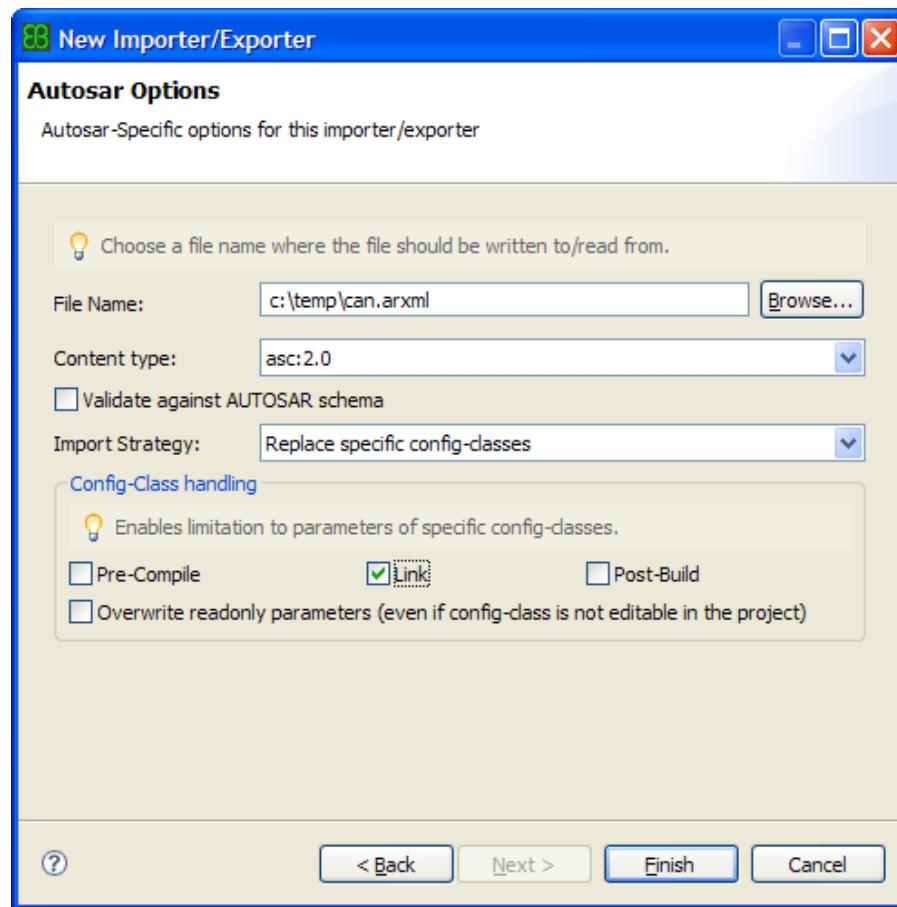


Figure 7.134. Configuring import options for configuration classes

- ▶ In the *Import Strategy* drop-down list box, select one of the following import strategies:
 - ▶ To import the complete module configuration and replace the existing one completely, select *Replace*.
 - ▶ To import only parameters and references of specific configuration classes, select *Replace specific Config Classes*. Then select the configuration classes you want to import in the *Config Class handling* panel of the *Autosar Options* page. Parameters and references with the selected configuration classes are imported and replace the existing parameters and references. Note that you can only import configuration classes that are editable in your project.
 - ▶ To combine the existing module configuration with the one to be imported, select *Merge*.
 - ▶ To combine the existing module configuration with specific configuration classes of the module configuration you want to import, select *Merge specific Config Classes*. Then select the configuration classes you want to import in the *Config Class handling* panel of the *Autosar Options* page. Parameters and references with the selected configuration classes are imported and merged with the existing parameters and references. Note that you can only import configuration classes that are editable in your project.

- ▶ If enabled, select the *Overwrite read-only parameters* check box. If this option is selected, the importer also imports and overwrites not editable parameters of the ECU configuration. For instructions for changing the configuration time of your project, refer to [Section 7.3.7.3, “Editing the ECU Configuration settings”](#).
- ▶ Click **Finish**. The selected configuration data is being imported into your project.

Configuration class handling in export mode

How configuration classes are handled, depends on whether you have selected import mode, export mode or import and export mode. In this section you find instructions for configuring configuration classes in export mode. On the *Autosar Options* page of the *Autosar Importer and Exporter*.

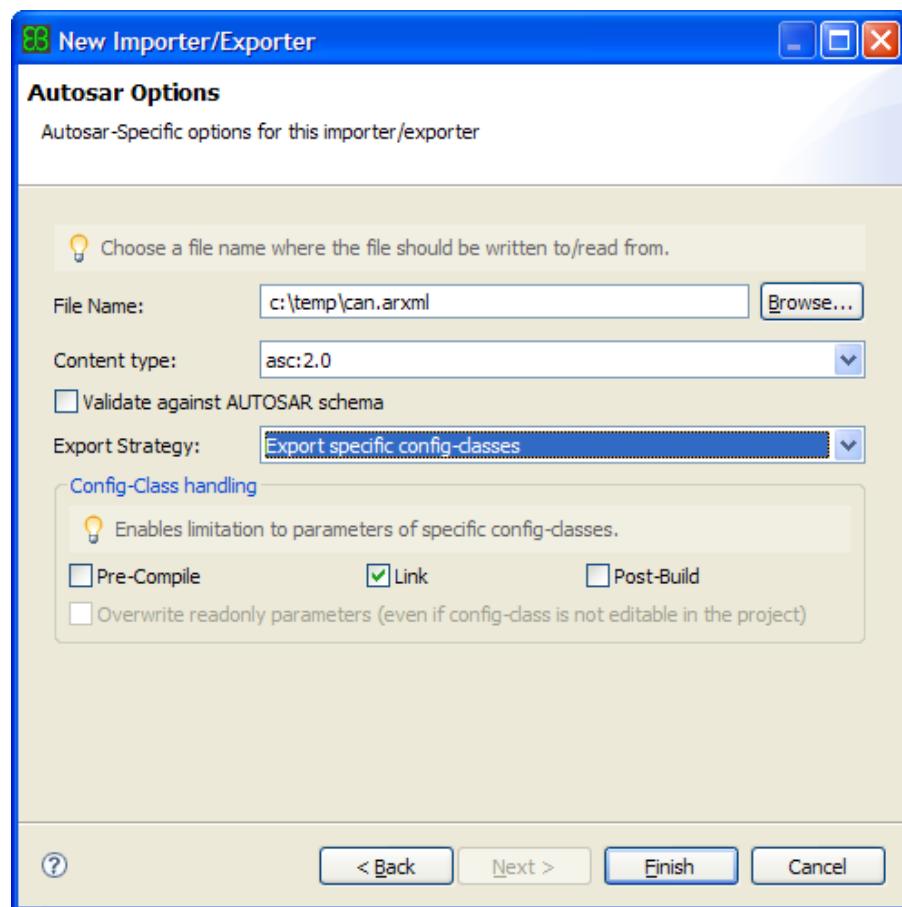


Figure 7.135. Configuring export options for configuration classes

- ▶ In the *Export Strategy* drop-down list box, select one of the following export strategies:
 - ▶ To export the complete module configuration into the specified file, select *Export*.
 - ▶ To export only parameters and references of specific configuration classes, select *Export specific Config Classes*. Then select the configuration classes you want to export in the *Config Class handling*

panel of the *Autosar Options* page. The selected configuration classes are exported into the specified file.

- ▶ Click **Finish**. The selected configuration data is being exported into the specified file.

7.10.2.3. Importing AUTOSAR system descriptions

7.10.2.3.1. Overview

EB tresos Studio provides an importer that allows to import AUTOSAR system description (Sys-D) and software component description files (SWC-D). In this chapter you learn:

- ▶ which ECU configuration modules can be configured (from section [Section 7.10.2.3.2, “Background information”](#))
- ▶ how to import system information (from section [Section 7.10.2.3.3, “Importing system information”](#))

7.10.2.3.2. Background information

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)
- ▶ IpduM (IPDU Multiplexer)
- ▶ CanIf (CAN Interface)
- ▶ Can (CAN Driver)
- ▶ CanTp (CAN Transport Protocol)
- ▶ CanNm (CAN Network Management)
- ▶ FrIf (FlexRay Interface)
- ▶ Fr (FlexRay Driver)
- ▶ FrTp (FlexRay Transport Layer)
- ▶ FrNm (FlexRay Network Management)
- ▶ LinIf (LIN Interface)
- ▶ Lin (LIN Driver)
- ▶ LinTp (LIN Transport Protocol)
- ▶ EcuC (EcuC Module)

- ▶ Nm (Network Management Interface)

Supported AUTOSAR meta model formats are:

- ▶ AUTOSAR 2.1 (XSD Document Version 2.1.3 Revision 0017, xmlns="http://autosar.org/2.1.2")
- ▶ AUTOSAR 2.1 (XSD Document Version 2.1.4 Revision 0018, xmlns="http://autosar.org/2.1.4")
- ▶ 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")
- ▶ AUTOSAR 3.2.1 (XSD Document Version 3.5.0 Revision 0001, xmlns="http://autosar.org/3.2.1")
- ▶ AUTOSAR 3.2.2 (XSD Document Version 3.6.0 Revision 0002, xmlns="http://autosar.org/3.2.2")
- ▶ AUTOSAR 4.0.2 (XSD Document Version 4.1.0 Revision 0002, xmlns="http://autosar.org/r4.0")
- ▶ AUTOSAR 4.0.3 (XSD Document Version 4.2.1 Revision 0003, xmlns="http://autosar.org/r4.0")
- ▶ AUTOSAR 4.1.0 (snapshot 2012-08-09, XSD Document Version 4.3.0 Revision 0000, xmlns="http://autosar.org/r4.0")

NOTE

It is possible to import files with different meta model versions within one single importer. 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)
-

7.10.2.3.3. Importing system information

To import configurations from the AUTOSAR Sys-D and SWC-D format:

- ▶ Follow the instructions of Step 1 until you arrive at the window *Create, manage and run im- and exporters*: [Section 7.10.2, “Importing and exporting configuration data”](#).

Then return to this set of instructions.

In the window *Create, manage and run im- and exporters*:

- ▶ Select the **Create a new im- or exporter** button.

The *New Importer/Exporter* window opens up.

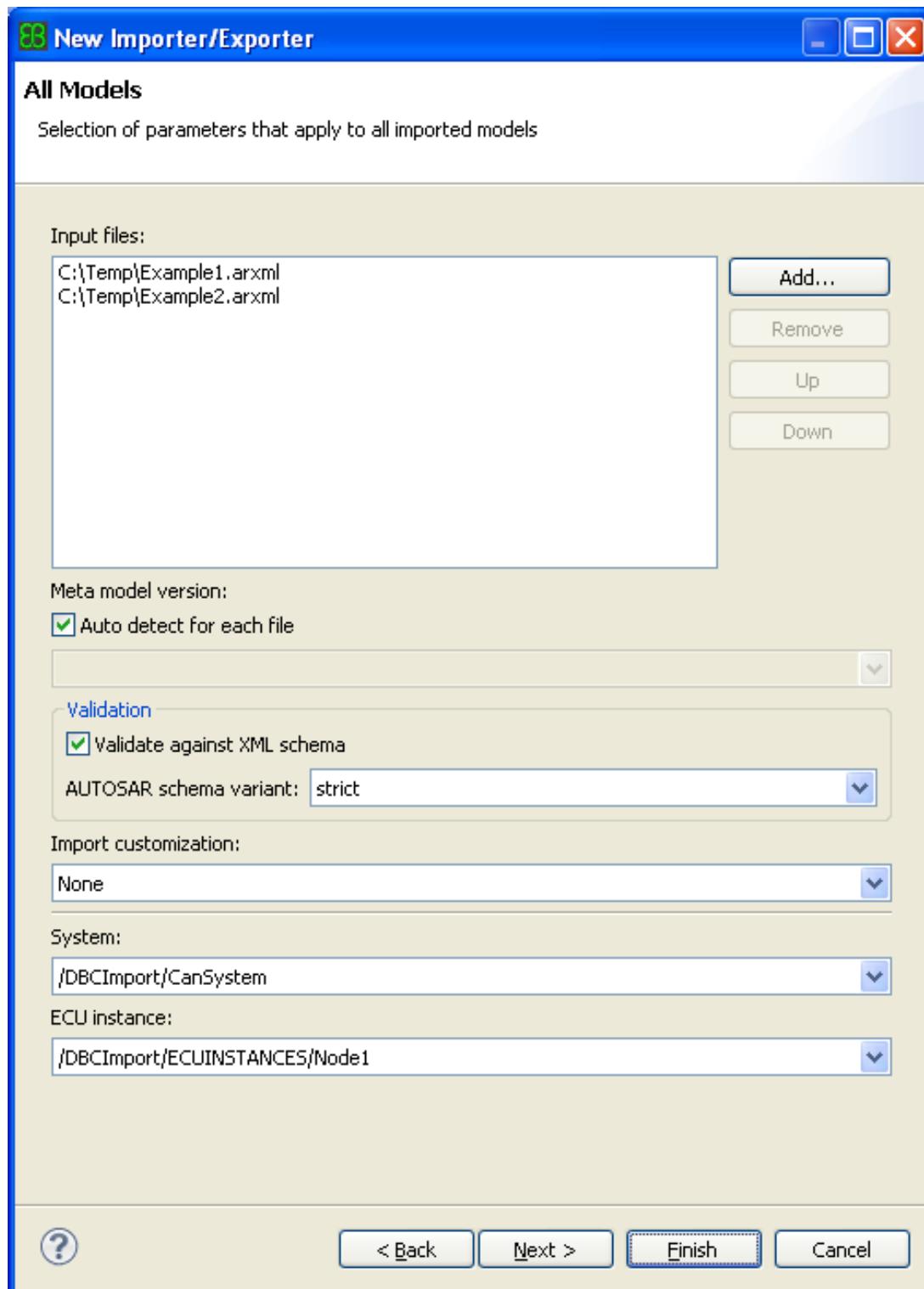
The dialog guides you through four pages as described in the following:



- ▶ Enter the name of the importer configuration into the *Name* text box.
- ▶ Select *System Description Importer (Autosar 2.1/3.0/3.1/3.2/4.0/4.1)* from the *Importer/Exporter* drop-down list box.
- ▶ In the *Mode* drop-down list box the value **Import only** is preselected. No other value is available.

The *System Description Importer* can only be used to import files. How to export to the AUTOSAR format see chapter [Section 7.10.2.4, “Exporting AUTOSAR system description”](#).

- ▶ To proceed to the next page, press the **Next** button. On the next page you may enter generic information.



- With the **Add** and **Remove** buttons on the right of the *Input Files* textbox, you may add files to load or remove files from the list of files to be loaded. Every time you add or remove a file, EB tresos Studio

attempts to automatically detect the AUTOSAR version of the files in the list and updates the values of the *System* and *ECU instance* drop-down list box.

- ▶ With the **Up** and **Down** buttons on the right of the *Input Files* textbox, you may change the order of the list of files to be loaded. The buttons are only enabled for single selection.

NOTE **Specify import order of files**



If one system model element is available in several files, it is necessary to specify which file has the highest priority to be imported. The uppermost file in the list is the one with the highest priority.

To define a file with the highest priority during the import, move this file up to the very top of the list.

- ▶ The *Meta model version* is set to *Auto detect for each file* by default when creating a new importer. Then every file is parsed with the meta model version specified in the file.

If the meta model version of a file cannot be auto detected (e.g. for Autosar version 3.1.5 which is currently not supported by EB tresos Studio), then an error is shown on top of the dialog and you need to manually choose the Meta model version.

To choose the AUTOSAR version of the input files manually, you need to deselect the *Auto detect for each file* check box and select one item in the drop-down list box *Meta model version*.

Then every file is read using the selected Meta model version and the drop-down list boxes *System* and *ECU instance* will only show values of files which are readable with the selected Meta model version.

NOTE **If the importer has errors, you cannot run the importer anymore**



When the list of input files or the Meta model version has changed, then the selected files are parsed and some consistency checks are performed, e.g. the files are validated against the XML schema.

If an error has occurred, the error message is shown on top of the dialog. Then it is not possible to run the importer.

- ▶ To validate the imported files against the XML schema, select the **Validate against XML Schema** check box. If this option is used in combination with *Auto detect for each file* option, then each file is validated against its own schema.

If more than one XML schema variant is available for the AUTOSAR meta model versions of the imported files, please choose one from the **AUTOSAR schema variant** combo box.

NOTE**Validation errors**

If the input files have different meta model versions and auto detection is not selected, the validation will most probably produce errors. In this case, it is recommended to disable the **Validate against XML Schema** check box.

- ▶ If the *Import customization* drop-down list box is enabled, select an import customization.

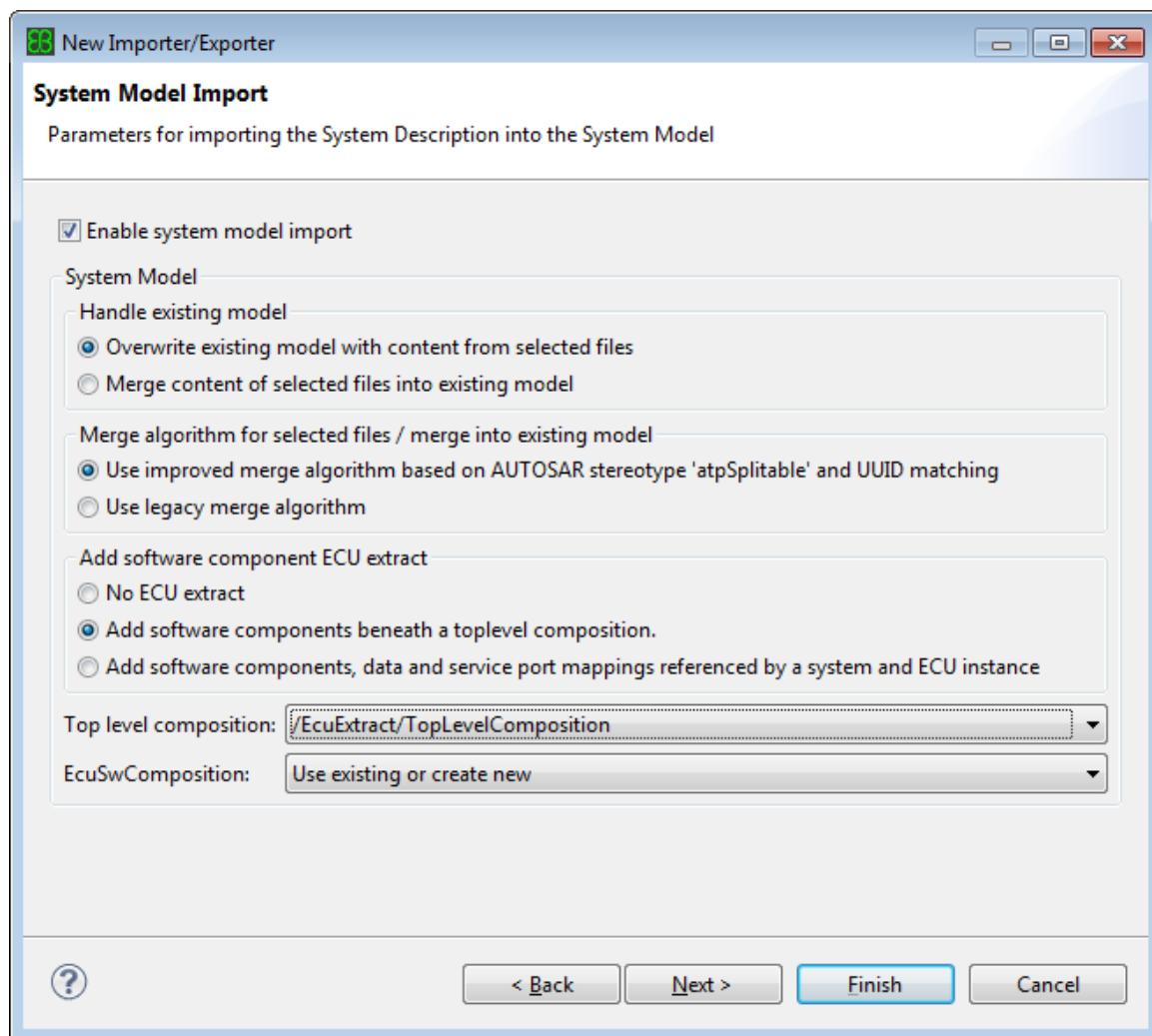
The *Import customization* drop-down list box provides available import customizations registered for this kind of importer in combination with the selected meta model version. An import customization affects the import process and modifies the imported data. In most cases this list will be empty and thus grayed out.

- ▶ You may choose a system element from the *System* drop-down list box. This is only necessary, if you want to import software components and mappings from a system element (options available on the *System Model Import* window) or if you want to import an ECU configuration (setting on the *ECU Configuration Import* window).
- ▶ You may choose an ECU instance element from the *ECU instance* drop-down list box. This is only necessary, if you want to import software components and mappings from a system element (options available on the *System Model Import* window) or if you want to import an ECU configuration (setting on the *ECU Configuration Import* window).

If you have chosen *None* from the *System* drop-down list box, the *ECU instance* drop-down list box is grayed out.

After having finished the *All Models* page, press the **Next** button to configure the system model import.

The *System Model Import* window opens up.



- ▶ To enable system model import into the project, check the *Enable system model import* checkbox. If the project has no system model, the complete page including the checkbox is grayed out.
- ▶ Choose how to handle the existing model:
 - ▶ To completely replace the system model in the project, select *Overwrite existing model with content from selected files*. If you selected several files, these files are merged and the merged model replaces the existing system model in the project.
 - ▶ To add the system model information of the selected files to the existing model, select *Merge content of selected files into existing model*.
- ▶ Choose the merge algorithm:
 - ▶ *Use improved merge algorithm based on AUTOSAR stereotype 'atpSplittable' and UUID matching*

The AUTOSAR meta model stereotypes <splittable> (3.x) / <atpSplittable> (4.x) define which elements can be split up over several physical files. The new improved merge algorithm is based on this concept.

In addition to this, the new merge algorithm uses UUID matching to find e.g. renamed elements. For example: you imported a file into the existing project model and rename an element in the file afterwards. If the element still has the same UUID, then the merge algorithm will find the correct entity and rename it in the project model when importing the file again.

For newly created System Description importer, this option is set as default.

► *Use legacy merge algorithm*

If you want to get the same result as in previous EB tresos Studio releases, then select the legacy merge algorithm.

NOTE**Usage merge algorithm**

The selected merge algorithm is used to merge multiple selected input files as well as to merge the content of the selected files into the existing model.

Although specifying to overwrite the existing model in the project, it is necessary to choose the merge algorithm if more than one file is imported. In this case these files are merged and the merged model replaces the existing system model in the project.

- The *Software component import* radio buttons offer you three possible choices, which software components, data mappings and service port mappings shall be imported. The selection here also influences, which other bits of information must be given:

► *Do not import software components and mappings:*

All software components, EcuSwCompositions, data mappings and service port mappings are removed from the system model to import.

No additional information is needed to perform this. The *Top level composition* drop-down list box and the *EcuSwComposition* drop-down list box are grayed out.

► *Import software components from top level composition, do not import mappings:*

All data mappings and service port mappings are removed, software components and EcuSwCompositions are retained in the system model to import.

You must also select a top level composition software component type element from the *Top level composition* drop-down list box to perform this. You may optionally choose an EcuSwComposition element from the *EcuSwComposition* drop-down list box.

► *Import software components, data and service port mappings from system:*

All software components, EcuSwCompositions, data mappings and service port mappings are retained in the system model to import.

You must also select a system element from the *System* drop-down list box and an ECU instance element from the *ECU instance* drop-down list box, both on the *All Models* window, to perform this.

You may optionally choose an `EcuSwComposition` element from the *EcuSwComposition* drop-down list box.

- ▶ If you have selected to *Import software components from top level composition, do not import mappings*, choose one of the offered composition software component type elements from the *Top level composition* drop-down list box to be used as top level composition.

The *Top level composition* drop-down list box is greyed out, if the parameter is not needed.

- ▶ If you have selected to either *Import software components from top level composition, do not import mappings* or *Import software components, data and service port mappings from system*, choose one of the offered `EcuSwComposition` elements from the *EcuSwComposition* drop-down list box. To use any `EcuSwComposition` that is already referenced in the project system model or to create a new one, choose *Use existing or create new*.

The *EcuSwComposition* drop-down list box is grayed out, if the parameter is not needed.

- ▶ Select **Next** to proceed to the next configuration window.

The *ECU Configuration Import* window opens up.

NOTE

Since the *System Description Importer* does not create module configurations, you need to ensure that a module configuration is present for every module you would like to import (EcuC, Com, PduR, IpduM, CanIf, Can, FrIf, Fr, FrTp, FrNm, LinIf, Lin, CanTp, CanNm, LinTp, Nm).

If no module configuration for a special module is available, the drop-down list box for this module is disabled and shows *No config available in project*.



- ▶ Enable or disable the import of the ECU configuration by checking or unchecking the *Enable ECU Configuration import* checkbox.

- ▶ To import into a module configuration (e.g. EcuC,Com,...), select it in the corresponding *Configuration for ...* drop-down list box.
- ▶ For details on the *Prefix* text box refer to the EB tresos AutoCore documentation.
- ▶ For details on the *Instance suffix for Frames, PDUs and Signals* check box refer to the EB tresos AutoCore documentation.
- ▶ In the *Buffer Assignment in Can/CanIf* drop-down list box, select the buffer assignment policy for the imported Can and CanIf configuration. For detailed information, please refer to [Section 7.10.2.6.3, “Importing DBC configurations”](#).

To end the importer configuration press the **Finish** button. This closes the *SysD Importer Options* window.

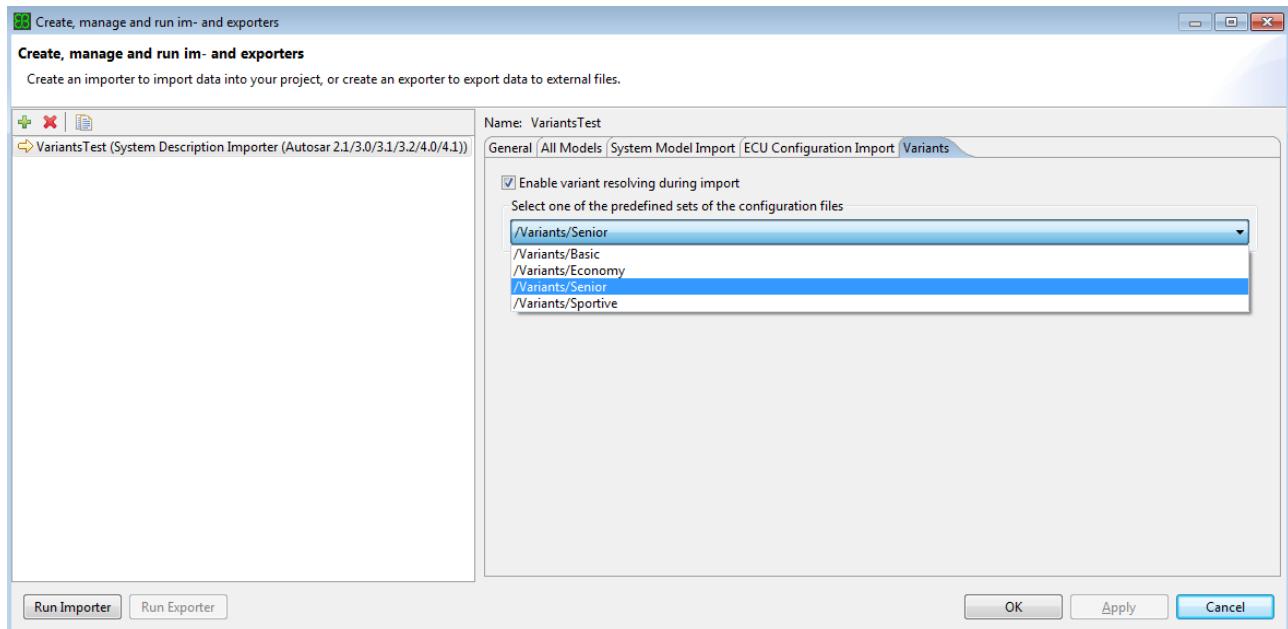
To run the Sys-D Importer, select the importer name you just created on the left side of the *Create, manage, and run im- and exporters* window and press the **Run Importer** button.

7.10.2.3.4. References

Details on the ECU extract to AUTOSAR parameter mapping are available in the EB tresos AutoCore documentation.

7.10.2.3.5. Variants tab

Variant resolving can be enabled using the Variants tab from the System Description Importer Wizard. The tab contains a checkbox that will allow the user to enable variant resolving, and a combobox that will allow selecting a predefined Variant from a list. The predefined Variants are defined in the input files.



More details about the Variant Handling are available in [Section 7.12, “Variant Handling”](#).

7.10.2.4. Exporting AUTOSAR system description

7.10.2.4.1. Overview

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 into a file format of your choice. In this chapter you learn:

- ▶ which AUTOSAR meta model formats are supported (at [Section 7.10.2.4.2, “Background information”](#)), and
- ▶ how to export system information (at [Section 7.10.2.4.3, “Exporting system information”](#)).

7.10.2.4.2. Background information

You can only export system information from a system model of the configuration project you are currently working on.

Depending on the version of the system model (1 or 2), the following shows a list of supported AUTOSAR meta model formats.

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 4.0.2 (XSD Document Version 4.1.0 Revision 0002, xmlns="http://autosar.org/r4.0")
- ▶ AUTOSAR 4.0.3 (XSD Document Version 4.2.1 Revision 0003, xmlns="http://autosar.org/r4.0")
- ▶ AUTOSAR 4.1.0 (snapshot 2012-08-09, XSD Document Version 4.3.0 Revision 0000, xmlns="http://autosar.org/r4.0")

7.10.2.4.3. Exporting system information

To export configurations into an AUTOSAR Sys-D format:

- ▶ Open the *Create, manage and run im- and exporters* dialog by right-clicking the ECU node of your project in the *Project Explorer* view and selecting *Im- and Exporters....*

The *Create, manage and run im- and exporters* dialog opens up.

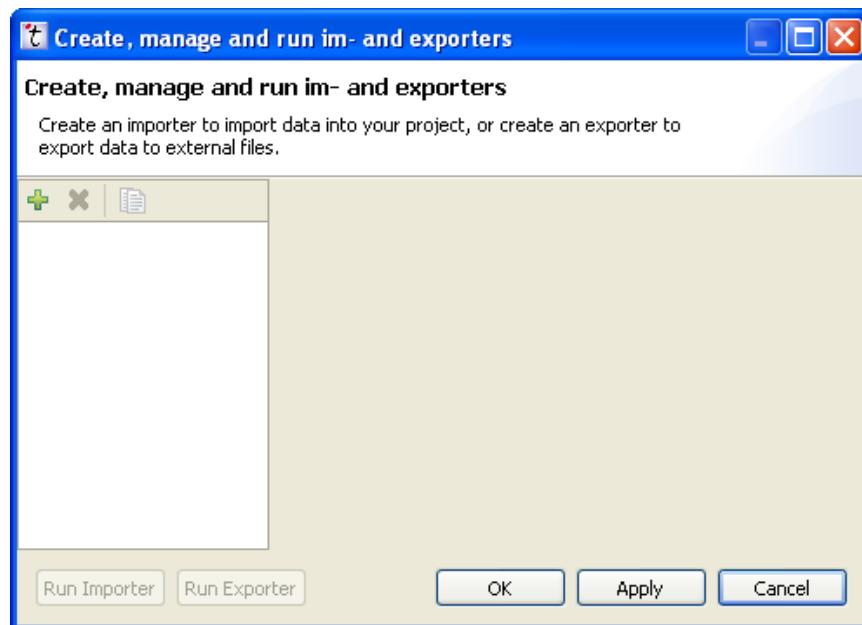


Figure 7.136. The *Create, manage and run im- and exporters* dialog

- ▶ To add a new exporter, select the **Create a new im- or exporter** button .

The *New Importer/Exporter* dialog opens up.

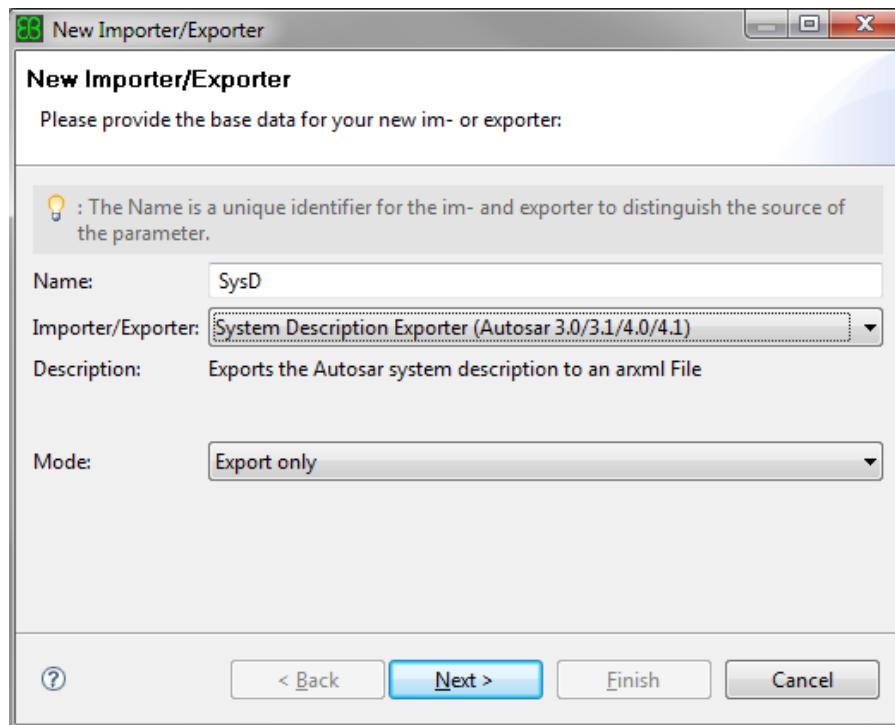


Figure 7.137. Configuring your new exporter

- ▶ In the *Name* text box, enter the name of your exporter configuration.
- ▶ In the *Importer/Exporter* drop-down list box, select *System Description Exporter (Autosar 3.0/3.1/4.0/4.1)*.
- ▶ In the *Mode* drop-down list box the value **Export only** is preselected. No other value is available.

With the *System Description Exporter* you can only export into one single file.

- ▶ To proceed to the next page, click **Next**. The *System Model Export* page opens up.

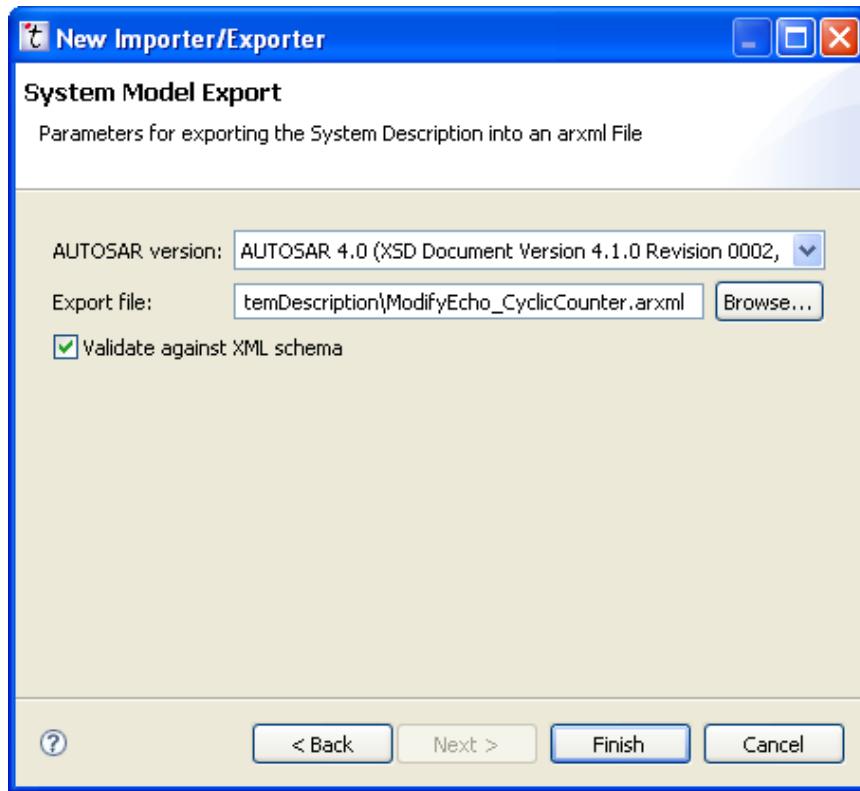


Figure 7.138. Specifying your exporter

- ▶ In the *AUTOSAR version* drop-down list box, select the AUTOSAR version you want to export to.
- ▶ In the *Export file* text box, browse to the file you want to export to or browse to a directory of your choice and enter a new file name.

NOTE**Selected files are overwritten**

If you select a file that already exists, this file will be overwritten by the exported file.

-
- ▶ If you want to validate the exported file against the Sys-D XML schema of the selected AUTOSAR version, select the **Validate against XML Schema** check box.
 - ▶ Click **Finish**. The configuration of the exporter is saved and the *New Importer/Exporter* dialog closes.
 - ▶ To run your exporter, select the exporter name you just created on the left side of the *Create, manage, and run im- and exporters* dialog.
 - ▶ Select **Run Exporter**.
 - ▶ When the export is finished, you find the exported Sys-D file in the directory you have specified.

7.10.2.5. Importing from the OIL format

7.10.2.5.1. Overview

EB tresos Studio provides an integrated OIL Importer to import Os configurations from OIL files. If an OIL file is available for import, the OIL Importer simplifies and speeds up the configuration of the Os module. It is also possible to import OIL files via the command line. This chapter describes the tool's support for the OIL format from the GUI and the command line.

7.10.2.5.2. OIL Importer and file format

The OIL Importer can read an OIL file and is able to configure the Os module.

The OIL Importer creates default configurations for hardware-related options. It creates one Os configuration.

During the import, the OIL Importer creates several containers for frames and signals. The names chosen for these containers are based on the names defined in the OIL file.

The mapping of the OIL format to the AUTOSAR parameters is described in the chapter `The Oil attribute naming translation` of the Os module documentation.

7.10.2.5.3. Importing OIL configurations

Since the OIL Importer does not create module configurations, you need to assure that an Os module configuration is present.

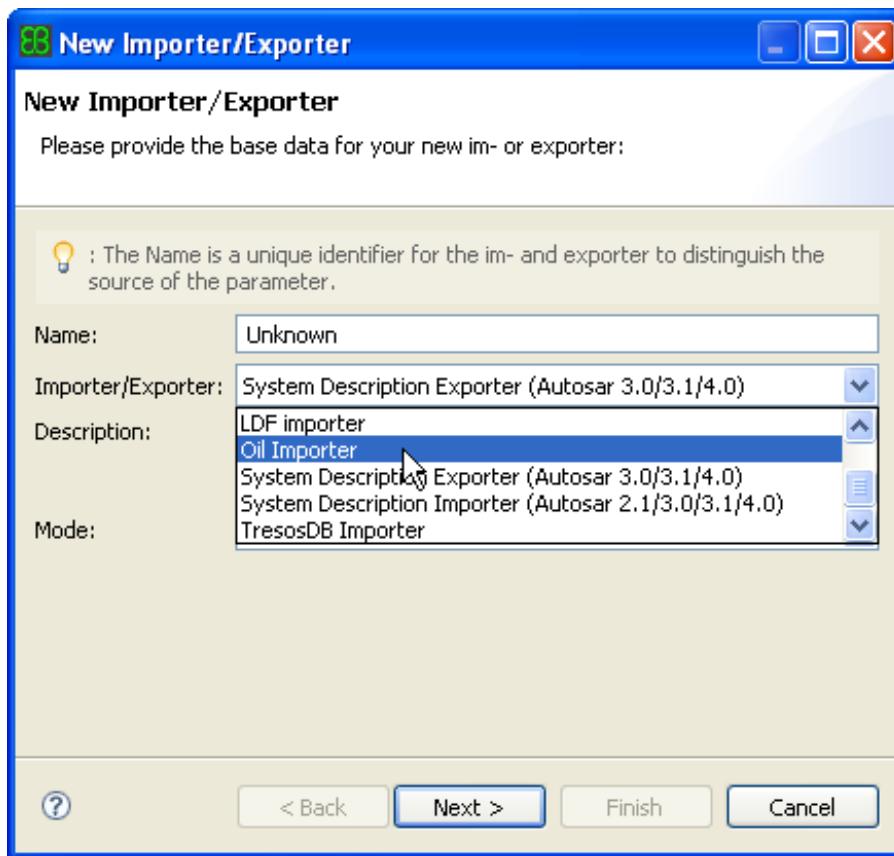
To import a configuration from the OIL format:

- ▶ Refer to [Section 7.10.2, “Importing and exporting configuration data”](#).
- ▶ Follow the instructions of Step 1 until you arrive at the window *Create, manage and run im- and exporters*.

In the window *Create, manage and run im- and exporters*:

- ▶ Select the **Create a new im- or exporter** button.

The *New Importer/Exporter* window opens up.



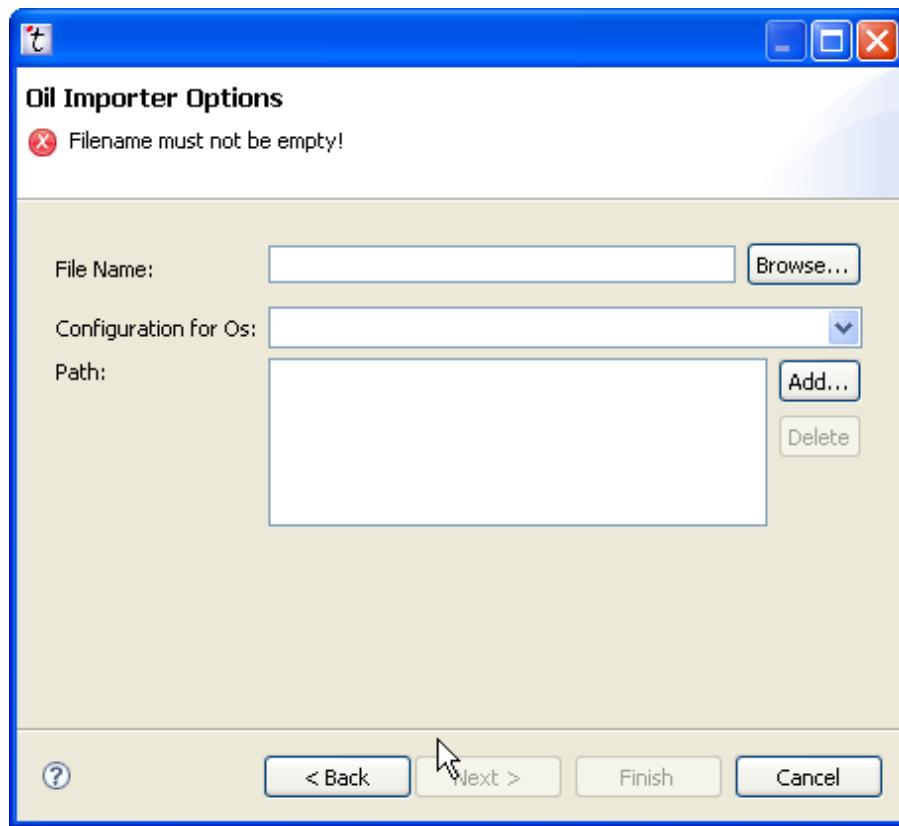
To enter the basic information for the new importer or exporter configuration:

- ▶ Select the *Name* textbox and enter a name for your new configuration.
- ▶ Select **OIL Importer** from the **Importer/Exporter** dropdown listbox.
- ▶ Select the *Mode* line dropdown listbox.
- ▶ Select **Import only** from the dropdown listbox.

The OIL Importer can only be used to import files. Exporting to OIL format is not possible.

- ▶ Click the **Next** button.

The *OIL Importer Options* page is shown.



- ▶ In the *File Name* textbox browse for the OIL file you want to import into EB tresos Studio.
- ▶ In the *Configuration for Os* menu line, select the target Os configuration for the import.
- ▶ In the *List of Path names* menu line, select a list of path names as include paths to additional OIL files.

To save the importer configuration:

- ▶ Select **Finish**.

This saves your importer configuration and closes the *OIL Importer Options* window.

To run the OIL Importer, select the file name you just created on the left side of the *Create, manage, and run im- and exporters* window.

- ▶ Select **Run Importer**.

The OIL configuration is being imported.

7.10.2.5.4. Importing OIL files via the command line

The OIL Importer of EB tresos Studio can also be used on the command line, e.g. to import OIL files in batch mode. This import function is encapsulated into the **legacy convert** command.

You may need to edit the configuration that has been created during the import before you use it to configure an ECU.

OIL Importer parameters for the command line

The commandline OIL Importer requires passing some system properties to the command line via the `-D` option. These are:

- ▶ `target` (mandatory): Specifies the target for which to import.
- ▶ `oil.include` (optional): List of pathnames, where possible oil include files are located.
- ▶ `oil.package` (mandatory) ARPackage name of the selected Os configuration.
- ▶ `oil.config` (mandatory) ConfigName of the Os configuration selected .

OIL import example

The following example shows a typical use of the OIL Importer (all on one line):

```
tresos_cmd.bat -Dttarget=v850
-Doil.include=C:\;D:\parser
-Doil.package=Os
-Doil.config=Os
-Dmsg1648=false legacy convert
network.oil output.epc
```

This line will import all frames and signals from the file `network.oil` into the Os module for the V850 architecture.

The `-Dmsg1648=false` parameter disables a warning message that would otherwise appear. The warning - if not disabled - informs users that all nodes that are required are generated automatically. Since this applies only to nodes that are not extracted from the OIL file, you may safely ignore this warning.

7.10.2.6. Importing CAN configurations from the DBC format

7.10.2.6.1. Overview

EB tresos Studio provides an integrated DBC Importer to ease and speed up the configuration of the CAN stack, if an appropriate DBC file is provided. To import LIN configurations, please continue reading at [Section 7.10.2.7, “Importing from the LDF format”](#).

7.10.2.6.2. Background information

The DBC Importer can read a DBC file, fill in the system model of a project, and is able to configure one or more of the following modules:

- ▶ Com (Communication)
- ▶ PduR (PDU Router)
- ▶ IpduM (IPDU Multiplexer)
- ▶ CanIf (CAN Interface)
- ▶ Can (CAN Driver)
- ▶ EcuC (EcuC Module)
- ▶ CanNm (CAN Network Management)
- ▶ CanTp (CAN Transport Layer)
- ▶ Nm (Network Management Interface)

For a selected controller of a DBC file, the DBC Importer imports messages and signals into the respective module configurations. The PDUs created are routed from CanIf to Com (receive messages) and from Com to CanIf (transmit messages) respectively.

NOTE The DBC Importer can only import CAN networks from a DBC file.



DBC files provide a feature that allows (and is commonly used by) the creator of the DBC file to attach any number of additional so-called *attributes* to messages and signals. The meaning of those attributes is defined by the creator of the DBC file, so the DBC Importer can naturally not import the information defined there. There is, however, a possibility for the user to provide a meaning for those attributes by creating a *DBC Attribute Definition File* (DADF) that tells the importer what to do with certain attributes found in the DBC file. How DAD files are created is described in [the section called “DBC Attribute Definition Files \(DADF\)”](#).

NOTE **Modules are not configured automatically by the DBC Importer**



The DBC Importer configures CanTp, CanNm, and Nm modules only if you had selected **Import customization**. The DBC Importer then detects CanTp N-PDUs, or CanNm NM-PDUs by using customer-specific properties in the imported .dbc file, for instance certain CAN-ID ranges, or *attributes* assigned to messages. If you do not select **Import customization**, the modules CanTp, CanNm, and Nm are not configured when you run the importer.

During the import, the DBC Importer creates a lot of containers for messages and signals. The names chosen for these containers are based on the names defined in the DBC file (for details refer to the EB tresos AutoCore documentation). Optionally all names can be extended by a user defined prefix and/or a calculated

instance suffix. If you want to know which names are created in which way, refer to the EB tresos AutoCore documentation.

Limitations

This section lists the restrictions of the EB tresos Studio DBC Importer.

- ▶ No LIN configurations can be imported with the DBC Importer. (There is an LDF Importer to do this, see [Section 7.10.2.7, “Importing from the LDF format”](#))
- ▶ All messages that are either sent or received by the selected controller are expected to be signal-carrying messages. This means that all signals and PDUs that are either sent or received by the selected controller will be entered into the Com configuration. Gateways are not supported. Transport Layer and Network Management configurations can only be configured if an import customization is employed, see also [Section 7.10.2.6.2, “Background information”](#). Alternatively, you can use the **Do not import Can Ids** field to prevent N-PDUs and NM-PDUs from being added to your ECU configuration.
- ▶ Not every DBC file may be understood by the DBC Importer, because DBC is a proprietary format and the DBC Importer can only interpret a subset of it (which was up to now sufficient to extract the information required). See [the section called “DBC grammar”](#) for more information on this subject.

7.10.2.6.3. Importing DBC configurations

The DBC Importer enables the import into the system model and into the ECU configuration of the project. The GUI of the importer enables entering parameters for system model as well as ECU configuration.

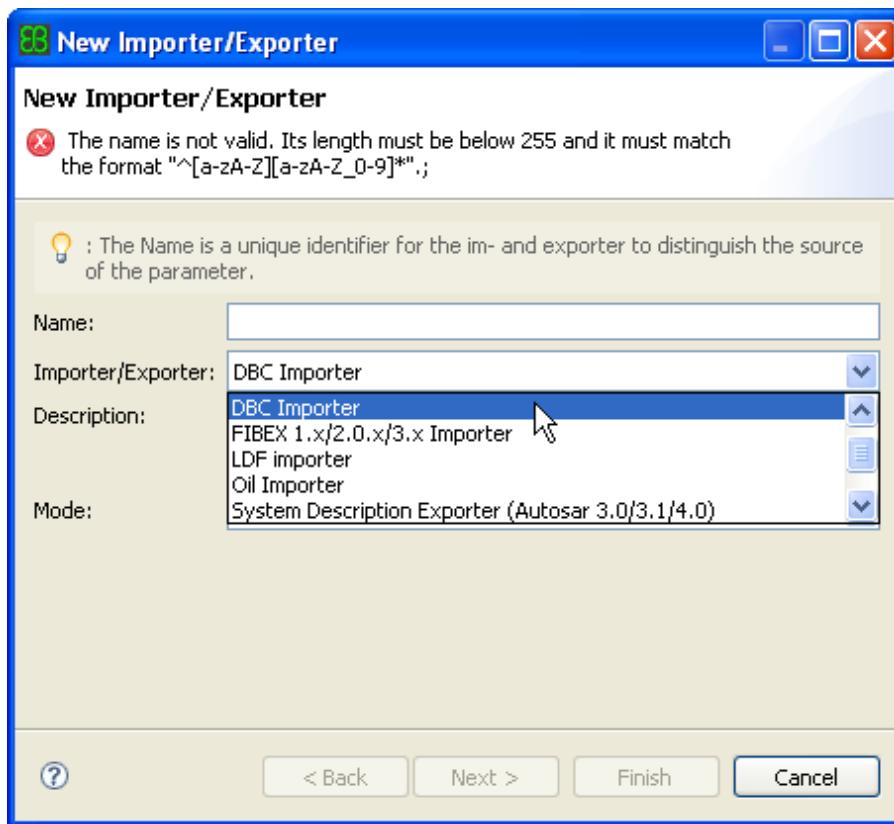
To import configurations from the DBC format:

- ▶ Refer to [Section 7.10.2, “Importing and exporting configuration data”](#).
- ▶ Follow the instructions of Step 1 until you arrive at the window *Create, manage and run im- and exporters*.

In the window *Create, manage and run im- and exporters*:

- ▶ Select the **Create a new im- or exporter** button.

The *New Importer/Exporter* window opens up.



To enter the basic information for the new importer or exporter configuration:

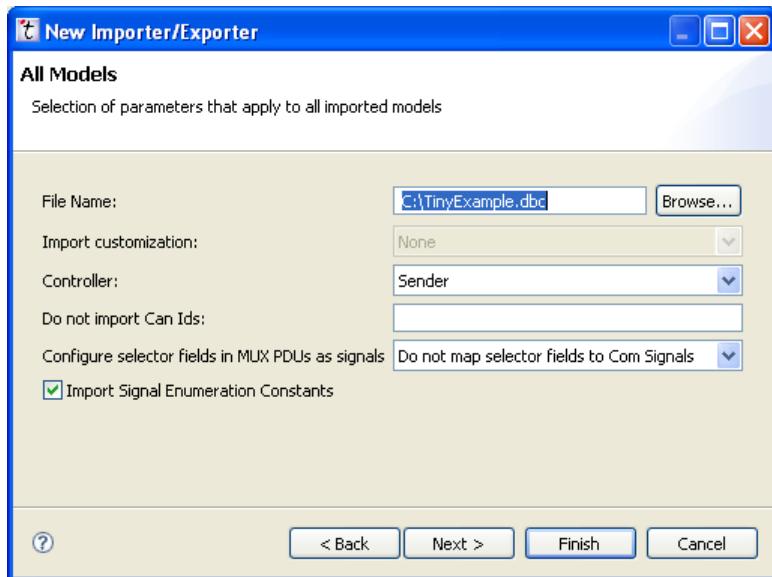
- ▶ Select the *Name* text box.
- ▶ Enter a name for your new configuration.
- ▶ Select *DBC Importer* from the **Importer/Exporter** drop-down list box.
- ▶ Select the *Mode* drop-down list box.
- ▶ Select **Import only** from the drop-down list box.

The DBC Importer can only be used to import files. Exporting to DBC format is not possible.

- ▶ Select **Next**.

The *All Models* window opens up.

The *All Models* window is provided for entering general information.



- ▶ Select the *File Name* file selection entry.
- ▶ Select the DBC file that you would like to import.
- ▶ If the *Import customization* drop-down list box is enabled, select an import customization.

The *Import customization* drop-down list box provides available import customizations registered for this kind of importer. An import customization affects the import process and modifies the imported data. In most cases this list will be empty and thus grayed out.

- ▶ In the *Controller* drop-down list box, select the controller contained in the DBC file you would like to import. This field is populated with the controllers available after a DBC file is selected
- ▶ In the **Do not import Can Ids** text box, you may specify ranges of CAN-IDs that you do not want to import. You may enter multiple IDs or ID-ranges, both in decimal or hexadecimal notation separated by semicolons. For example: 0–10;0x12;0x14–0x16; filters IDs from 0 to 10, 0x12 and 0x14 to 0x16. Syntax errors are displayed while you type.
- ▶ With the **Configure selector fields in MUX PDUs as signals** drop down list box, you may specify how multiplexed PDUs and their contained selector fields shall be imported. Three modes are available: **Do not map selector fields to Com Signals**, **Map selector fields to Com Signals**, and **AUTOSAR compliant mapping**. For detailed information about this option, refer to [the section called “I-Pdu Multiplexer Configuration”](#).
- ▶ To import the content of `VAL_` tokens from the DBC file, select the **Import Signal Enumeration Constants** check box. Further information on the `VAL_` tokens is available at (see [the section called “DBC grammar”](#)).

For each imported signal enumeration, the generated Rte code contains a C language macro that maps a symbolic name to a specific value. The symbolic name can be used by the application code. The generated code, however, is only valid if the following conditions meet:

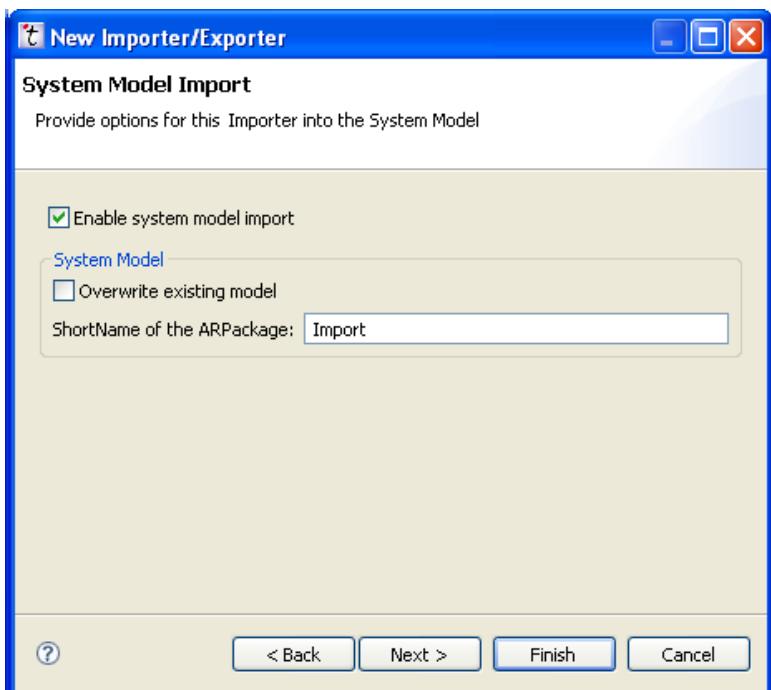
- ▶ All symbolic names must be valid C language identifiers.

- ▶ Every symbolic name must represent exactly one numeric value.

If you require the generated Rte code and the imported file violates any of these conditions, clear the **Import Signal Enumeration Constants** check box.

- ▶ Select **Next**.

The *System Model Import* window opens up.

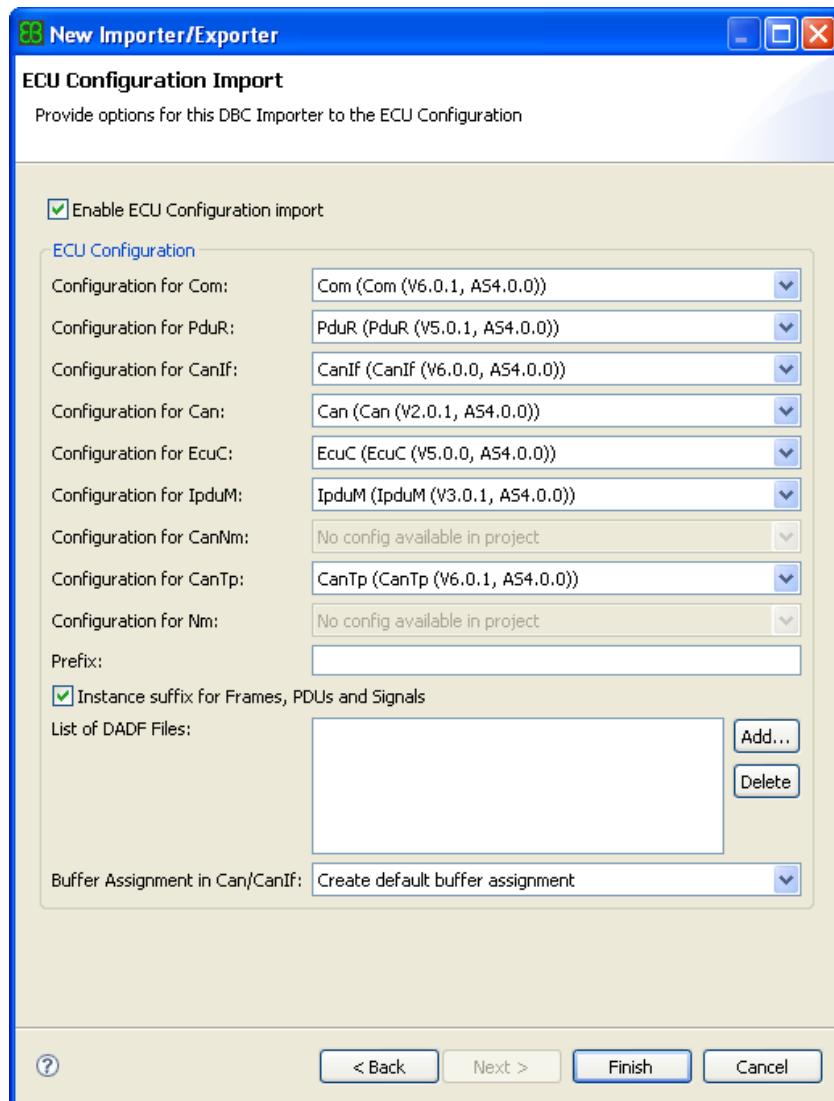


- ▶ To enable system model import into the project, check the *Enable system model import* checkbox. If the project has no system model, the complete page including the checkbox is greyed out.
- ▶ To completely replace the system model in the project, check the checkbox *Overwrite existing model*.
- ▶ To select the AUTOSAR SHORT-NAME of the AUTOSAR AR-PACKAGE into which the system model of the source file is to be imported, enter the AUTOSAR SHORT-NAME into the **ShortName of the ARPackage** textbox.
- ▶ Select **Next** to proceed to the next configuration window.

The *ECU Configuration Import* window is provided for entering ECU configuration-specific parameters.

NOTE

If an ECU configuration is to be imported, you need to ensure that a module configuration is present for every module you would like to import into. In the current example, these are: Com, PduR, CanIf, Can, EcuC, IpduM, CanNm, CanTp, and Nm.



- ▶ To enable ECU configuration import, select the *Enable ECU Configuration Import* check box.
- ▶ To import into a **Com** module configuration, select it in the *Configuration for Com* drop-down list box.
- ▶ To import into a **PduR** module configuration, select it in the *Configuration for PduR* drop-down list box.
- ▶ To import into a **CanIf** module configuration, select it in the *Configuration for CanIf* drop-down list box.
- ▶ To import into a **Can** module configuration, select it in the *Configuration for Can* drop-down list box.
- ▶ To import into a **EcuC** module configuration, select it in the *Configuration for EcuC* drop-down list box.

- ▶ To import into a `IPduM` module configuration, select it in the *Configuration for IPduMpLex* drop-down list box.
- ▶ To import into a `CanNm` module configuration, select it in the *Configuration for CanNm* drop-down list box.
- ▶ To import into a `CanTp` module configuration, select it in the *Configuration for CanTp* drop-down list box.
- ▶ To import into an `Nm` module configuration, select it in the *Configuration for Nm* drop-down list box.
- ▶ For details on the *Prefix* text box refer to the EB tresos AutoCore documentation.
- ▶ For details on the *Instance Handling* check box refer to the EB tresos AutoCore documentation.
- ▶ In the *List of DADF Files* list box, you may optionally add one or more DADF files to be applied during the import.

To configure the Buffer Assignment that shall be made during DBC import:

- ▶ In the *Buffer Assignment in Can/CanIf* drop-down list box you can select the type of buffer assignment that shall be made during import of Can and CanIf. Currently two options are available.
 - ▶ *Create default buffer assignment*: Up to four hardware objects (send and/or receive hardware objects for Basic and/or Extended CanIds) are configured.
 - ▶ *Do not create buffer assignment*: No hardware object is created; they must be configured manually.

To save the importer configuration:

- ▶ Select **Finish**.

This saves your importer configuration and closes the window.

- ▶ To run the DBC Importer, select the importer name you just created on the left side of the *Create, manage, and run im- and exporters* window.
- ▶ Select **Run Importer**.

The DBC configuration is being imported.

7.10.2.6.4. References

Details on the DBC to AUTOSAR parameter mapping are available in the EB tresos AutoCore documentation.

DBC Attribute Definition Files (DADF)

DBC files provide a feature that allows (and is commonly used by) the creator of the DBC file to attach any number of additional so-called *attributes* to messages and signals. The meaning of those attributes is defined by the creator of the DBC file, so the DBC Importer can naturally not import the information defined there.

DAD files are XML files that can be used to define how those attributes shall be evaluated by the importer. To introduce the format, we proceed as follows. First, the possible XML-tags of a DAD file are defined, together

with their allowed attributes. Then the syntax, i.e. the allowed inclusion structure of the XML-tags, is declared. Given this, we provide the semantical significance of the defined objects. Finally, we supply a short, commented example of a DAD file.

The following text:

- ▶ lists the limitations of the DADF format,
- ▶ defines the possible XML tags of a DADF and the allowed attributes of these XML tags,
- ▶ declares the syntax (the allowed inclusion) of the XML tags, and
- ▶ provides the semantic significance of the defined objects.

At the end of this information, you will find a DADF example with comments.

Limitations

This section lists the limitations of the DADF mapping.

- ▶ Only Attributes defined for signals and messages can be accessed
- ▶ There are no conditionals, i.e. if an attribute is mapped to an AUTOSAR-element identified by its DEFINITION-REF, this mapping will unconditionally be applied to all elements identified, for which the attribute is set.

DADF XML tags and attributes

The following table lists the XML tags and the attributes allowed:

XML tags	Attributes
assignments	--
path	name
valueMap	attrName
default	value
attribute	attrName
mapping	from, to

Explanation of the syntax definition

The XML syntax for DAD files is defined by using production rules.

- ▶ The rule `left --> right` means that `left` includes/produces `right`.

- ▶ The asterisk * is the Kleene star, thus object* means that object may appear any number of times, including never.
- ▶ The plus symbol + in object+ means that object may appear any number of times but at least once.
- ▶ A vertical bar separates parameters in a list from which one parameters must be chosen or used, thus left | right means left or right.
- ▶ The question mark ? in object? means that object may occur once or never.

DADF XML syntax

The following inclusion structure is allowed in DADF files:

- ▶ assignments --> path*
- ▶ path --> (valueMap | default | attribute) path*
- ▶ valueMap --> mapping+
- ▶ default -->
- ▶ attribute -->
- ▶ mapping -->

DADF XML semantics

XML tag	Semantics
assignments	The tag is the outermost tag and serves as parenthesis for the remaining tags.
path	<p>This tag serves to specify the AUTOSAR parameters to be inserted/modified by the DAD file by identifying its schema path. Paths start at module level, i.e. they may not contain an AR-Package name.</p> <p>Define paths in the <code>name</code> attribute of the <code>path</code> tag.</p> <p>At every point in the DAD file, there is a current path. This path is empty as long as no path has been specified. For every new open <code>path</code> tag, the content of the <code>name</code> attribute of this tag is added to the current path.</p> <p>Example: To set the path <code>Com/ComSignal/ComSignalTransferProperty</code> as the current path, use the <code>path</code> tag:</p> <pre><path name="Com/ComSignal/ComSignalTransferProperty"></path></pre> <p>You can also nest several <code>path</code> tags:</p>

XML tag	Semantics
	<pre data-bbox="409 406 1183 507"><path name="Com/ComSignal"> <path name="ComSignalTransferProperty"></path> </path></pre>
valueMap	<p>Defines how the value for an AUTOSAR parameter is retrieved from a DBC file.</p> <p>Use the tag with one or more mapping tags to map enumeration values from DBC to values in EB tresos Studio.</p> <p>This tag is always wrapped into one or more path tags. Therefore, whenever it is used, the DAD file's current path at the valueMap tag position is not empty. That current path needs to reference an AUTOSAR parameter.</p>
default	<p>Defines how the value for an AUTOSAR parameter is retrieved from a DBC file.</p> <p>To override the default value of a parameter, specify your own default value with the default tag. To do so, set the attribute value of the tag to the desired value.</p> <p>This tag is always wrapped into one or more path tags. Therefore, whenever it is used, the DAD file's current path at the default tag position is not empty. That current path needs to reference an AUTOSAR parameter.</p>
attribute	<p>Defines how the value for an AUTOSAR parameter is retrieved from a DBC file.</p> <p>If attribute A has an ordinal value, such as an integer or a string, and you want the AUTOSAR parameter (selected by the current path) to receive that value, use the attribute tag and specify the name of A in attrName.</p> <p>This tag is always wrapped into one or more path tags. Therefore, whenever it is used, the DAD file's current path at the attribute tag position is not empty. That current path needs to reference an AUTOSAR parameter.</p>
mapping	<p>Use mapping tags to map enumeration values from DBC to values in EB tresos Studio.</p> <p>Use the from attribute in the mapping tag to specify the enumeration value from the DBC file to the AUTOSAR parameter. The AUTOSAR parameter value is specified in the to attribute.</p>

Example DAD file

The following is example code for a DBC Attribute Definition File. It shows how the EB tresos importer can be configured so that the values of the attributes are mapped correctly into your module configurations.

If you remove the comments, the file is a valid DAD file. The file is associated with a DBC file that defines a CAN network with just two controllers with two messages, each of which contains just one signal. This network uses three attributes.

The definitions have to be always wrapped by *assignment* tags.

```
<assignments>
```

This example shows how to map enumeration attributes of DBC to AUTOSAR parameter values. The path given below states the path to the parameter definition targeted. In this case it states that the destination are all ComSignal containers of the Com plugin.

```
<path name="Com/ComSignal">
```

The path given below defines the name of a parameter of the container that is specified in the above path.

```
<path name="ComSignalTransferProperty">
```

At this point, a concrete parameter is identified. What follows is the definition of where in the DBC file the information for the value of this parameter can be found. Since the parameter specifies a property of a signal, it is assumed, that this value can be found in an attribute that is attached to a signal in the DBC file. In this case, this is the attribute with the identifier "Attribute2". This DBC attribute is defined to be an enumeration (in the DBC file), therefore the importer needs to know which enumeration values from the DBC file are mapped to which AUTOSAR values. This is done inside the mapping tags, where the DBC value is stored in *from*, whereas the AUTOSAR value is stored in *to*.

```
<valueMap attrName="Attribute2">
  <mapping from="one" to="Triggered"/>
  <mapping from="two" to="Pending"/>
  <mapping from="three" to="Triggered"/>
  <mapping from="four" to="Triggered"/>
</valueMap>
</path>
```

The path below defines the name of another parameter. The complete path is Com/ComSignal/ComSignalInitValue.

```
<path name="ComSignalInitValue">
```

The attribute with the identifier `Attribute3` in the DBC file is of ordinal type, so here it suffices to provide the identifier of that attribute. The value of the attribute will be copied to the parameter specified by the path above.

```
<attribute attrName="Attribute3"/>
</path>
</path>
```

The two closing *path* tags above remove first the trailing `ComSignalInitValue` and then the trailing `Com/ComSignal` from the current path. Note that these were the strings added to the current path by the corresponding opening *path* tags. So, at this point the current path is empty again. Therefore, in the next *path* tag, we have to start at module level again in order to specify a correct path to an AUTOSAR parameter.

```
<path name="Com/ComIPdu/ComIPduMinimumDelayFactor">
  <attribute attrName="Attribute1"/>
</path>
```

All parameters that are not explicitly created by the DBC Importer will be set to their default value. A list of all parameters is available in the EB tresos AutoCore documentation. DAD files, however, allow you to override that default value as shown below.

```
<path name="PduR/PduRRoutingTable/PduRRoutingPath">
  <path name="SduLength">
    <default value="8"/>
  </path>
</path>
</assignments>
```

The DBC parser

DBC is not an open file format, but a proprietary format defined by Vector Informatik. For this reason, not every DBC file may be importable by the DBC Importer. The following chapter describes which DBC files the DBC Importer can read.

For reading DBC files a DBC parser is applied. Every parser is based on a *grammar*. Since the grammar of the DBC format is not freely available, EB deduced a subset of their own, which allows parsing the information needed by the DBC Importer.

DBC grammar

The DBC Importer uses the following (*EBNF*) grammar to parse DBC files. The DBC Importer ignores lines that start with tags, which are not specified here.

```
dbc = version, ns, bs, controllers, {message}, {definition} ;  
  
definition = comment | attrdef | attrinit | attribute | envVarDef | valueTable |  
            senderList;  
  
version = 'VERSION', quotedString ;  
  
ns = 'NS_-', ':', { UC | '_' } ;  
  
bs = 'BS_-', ':' | 'BS_-, ':' integer : integer, integer; (* values ignored *)  
  
controllers = 'BU_-', ':', identifier, {identifier} (*controller names*) ;  
  
message = 'BO_-', integer (*CAN-id*), identifier (*message name*), ':',  
         integer (*size of data in bytes*), identifier (*sender*),  
         signal, {signal} ;  
  
senderList = 'BO_TX_BU_-', integer (*CAN-id*),  
            identifier {',', identifier} (*sender*) ;  
  
signal = 'SG_-', identifier (*signal name*), ['M'|('m', integer)], ':',  
        integer (*startbit*), '|', integer (*bit length*), '@',  
        ('0' | '1') (*endianess*), ('+' | '-') (*type*),  
        '(', integer (*factor*), ',', integer (*offset*), ')',  
        '[', integer (*minimum*), '|', integer (*maximum*), ']',  
        quotedString (*unit [km, h, kg, ...]*),  
        identifier {',', identifier} (*receiver*) ;  
  
signalGroup = 'SIG_GROUP_-', integer (*CAN-id*), identifier  
             (*signal group name*), integer (*not used*),  
             ':' identifier {',', identifier} (*signal*) ;  
  
sigValType = 'SIG_VALTYPE_ ', integer (*CAN-id*), identifier (*signal name*),  
             ':', integer (*signal type*);  
  
valueTable = 'VAL_-', integer (*CAN-id*), identifier (*signal*),  
            valueList {',', valueList};"
```

```
valueList = integer (*value*), quotedString (*symbol*);  
  
valueTableEnvVar = 'VAL_-', integer (*CAN-id*),  
                   valueListEnvVar {',', valueListEnvVar};"  
  
valueListEnvVar ::= /* empty */  
                  |  
                  integer, quotedString; (* values ignored *)  
  
comment = 'CM_-' , comment-body ';' ;  
  
comment-body = message-comment | signal-comment | envVar-comment |  
             controller-comment | net-Comment | otherComment ;  
  
message-comment = 'BO_-' , integer (*CAN-id*), quotedString (*comment*) ;  
  
signal-comment = 'SG_-' , integer (*CAN-id*), identifier (*signal name*),  
                quotedString (*comment*) ;  
  
envVar-comment = 'EV_-' , identifier, quotedString (*comment*) ;  
  
controller-comment = 'BU_-' , identifier, quotedString (*comment*) ;  
  
net-comment = quotedString (*comment*) ;  
  
otherComment = ? This rule should eat all comment for  
objects which are not supported ? ;  
  
attrdef = 'BA_DEF_-', {'BO_-' | 'BU_-' | 'SG_-' | 'EV_-'},  
         quotedString (*attribute name*), dataType, ';' ;  
  
dataType = 'STRING' |  
          'ENUM', quotedString, {',', quotedString} |  
          'INT', integer (*lower bound*), integer (*upper bound*) |  
          'HEX', integer (*lower bound*), integer (*upper bound*) |  
          'FLOAT', float (*lower bound*), float (*upper bound*) ;  
  
attrinit = 'BA_DEF_DEF_-', quotedString (*attribute name*), attrVal ';' ;  
  
attrVal = float | integer | quotedString ;
```

```

attribute = 'BA_-' , quotedString (*attribute name*), attrDest, attrVal, ';' ;

attrDest = 'BU_-' , identifier (*controller name*) |
           'BO_-' , integer (*CAN-id*) |
           'SG_-' , integer (*CAN-id*), identifier (*signal name*) |
           'EV_-' , identifier {*env. var. name*} ;

envVarDef = 'EV_-' , identifier (*variable name*), ':',
            integer (*value type*),
            '[' , integer (*minimum*), '|', integer (*maximum*), ']',
            quotedString (*unit*), integer (*default value*),
            integer (*number of envVarDef*),
            identifier (*contains access information*),
            identifier {',' , identifier} (*controllers*), ';' ;

exponent = [Ee], ['+' | '-'], digit, {digit} ;
integer = ['-'] digit, {digit} ;
float = ['-'], digit, {digit}, '.', {digit}, [exponent] | [ '-' ], digit,
        {digit}, exponent ;
identifier = (letter | '_-'), {letter | digit | '_-' } ;
digit = '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9' ;
letter = LC | UC ;
nl = ? new line ? ;
UC = 'A' | 'B' | 'C' | 'D' | 'E' | 'F' | 'G' | 'H' | 'I' | 'J' | 'K' |
     'L' | 'M' | 'N' | 'O' | 'P' | 'Q' | 'R' | 'S' | 'T' | 'U' | 'V' |
     'W' | 'X' | 'Y' | 'Z' ;
LC = 'a' | 'b' | 'c' | 'd' | 'e' | 'f' | 'g' | 'h' | 'i' | 'j' | 'k' |
     'l' | 'm' | 'n' | 'o' | 'p' | 'q' | 'r' | 's' | 't' | 'u' | 'v' |
     'w' | 'x' | 'y' | 'z' ;
quotedString = '''', any - ''', ''' ;
any = ? any character ? ;
ws = ? any white space character, i.e. space or tab ? ;

```

The code uses the following conventions:

```

left --> right
left is composed of right

A , B
A is followed by B

```

{ A }

A may be present one or more times

[A]

A may be present zero or one times

A | B

either A or B must be present

single quotes ''

denote literals i.e. strings that need to appear in this form in the DBC file

? A ?

A is a non-formal description

(A)

are used to group other expressions (as parentheses do in maths)

I-Pdu Multiplexer Configuration

The DBC file format defines Multiplexer configuration information differently than the AUTOSAR system template does. Multiplexer selector fields in the DBC file format are defined as a special kind of signals, so-called *mode signals*.

If a PDU contains such a mode signal, it usually also contains signals depending on this mode signal. Such signals are called *mode dependent signals* and are only transmitted if the mode signal contains a certain value.

The AUTOSAR system template defines so called *DynamicPart* bit ranges which depend on the value of the selector field and *StaticPart* bit ranges which are independent from the selector field value.

There are many possible mappings between the DBC and the AUTOSAR I-Pdu Multiplexer configuration representations. For instance, a straight forward mapping between the DBC signals and AUTOSAR bit ranges would be to map a mode signal onto the selector field, to map any mode dependent signal onto a single *DynamicPart*, and to map all mode independent signals to one single *StaticPart*. This mapping very likely leads to suboptimal I-pduM configurations with many *DynamicParts* involving unnecessary memory and runtime overhead.

The DBC Importer provides an option with which you can specify how multiplexer selector fields, *StaticParts*, and *DynamicParts* shall be configured. Three modes are available:

- ▶ **Do not map selector fields to Com Signals:** In this mode, the multiplexer selector field is *not* included as signal in any static or dynamic SignalIPdu.
- ▶ **Map selector fields to Com Signals:** In this mode, the multiplexer selector field is included as *mode dependent signal* in every dynamic SignalIPdu. Since the value of this signal must not be changed, the use is limited.
- ▶ **AUTOSAR compliant mapping:** As in the previous mode, this mode includes the multiplexer selector field as *mode dependent signal*. Moreover, the bit ranges of the static and dynamic parts are not *stuffed*,

i.e. the bit offsets of the bit ranges in the Multiplexed PDUs and in the contained static and dynamic SignalIPdus are identical. Additionally all dynamic SignalIPdus of belonging to the same Multiplexed PDU own identical bit ranges.

- ▶ If you need low RAM and ROM consumption, and your IPduM does not need to comply to the AUTOSAR standard, select the first mode.
- ▶ If you require that the values of the selector fields of the dynamic SignalIPdus can be retrieved at runtime and stuffing of the bit ranges for low RAM and ROM consumption still shall be performed, select the second mode.
- ▶ If you require that the that the IPduM shall be configured in an AUTOSAR compliant way, and/or better runtime behavior at the expense of higher RAM/ROM consumption select the third mode.

The following example outlines how the DBC Importer performs the IPDU Multiplexer configuration mapping if the first mode (**Do not map selector fields to Com Signals**) was selected:

The CAN message defined in the example contains a number of *big endian* and *little endian* signals which either depend on the mode signal or not. The mode signal itself is assumed to lie on bits [0..1] of the message and is not explicitly shown. Dynamic signals depending on different values usually overlap each other; however the outlined example does not show such overlaps to keep the example simple.

	7	6	5	4	3	2	1	0
0	SL	SL	SL					
1	SL		SL	SL		SL	SL	SL
2	DL2		DL1	DL1	DL1		SL	SL
3						DB1	DB1	DB1
4	DB1	DB1	DB1				SB	SB
5	SB		DL3	DL3	DL3			
6								
7	DB2	DB2	DB2					

The first character on the label of each bit indicates whether the signal occupying the bit is either mode independent/static (S), or mode-dependent/dynamic (D). The second character describes the endianess of the occupying signal, which is either big endian (B), or little endian (L). Bits, which belong to mode-dependent signals, additionally display the mode signals value they depend on: For instance DL3 indicates that the signal of the bit is only transmitted if the mode signals has a value of 3.

In the first step, the algorithm traverses all bits of the PDU beginning with the least significant bit of the PDU (bit 0) and ends with its most significant bit, bit 63. The algorithm assigns the signal bits either to static or to dynamic bit ranges depending on whether they are occupied by static or dynamic signals.

- ▶ A bit range starts and ends with a bit which is actually occupied by a signal.

- ▶ If a static bit follows a dynamic bit, the current dynamic bit range ends, and a new static bit range begins.
- ▶ If a dynamic bit follows a static bit, the current static bit range ends, and a new dynamic bit range begins.
- ▶ A sequence of unused bits is added to a static or dynamic part if it is not longer than seven bits and surrounded by two static or dynamic bits. In all other cases, the first empty bit succeeding a bit range confines it, as well as the last empty bit before a bit range.

7	6	5	4	3	2	1	0
SL	SL	SL					
SL		SL	SL		SL	SL	SL
DL2		DL1	DL1	DL1		SL	SL
					DL1	DL1	DL1
DB1	DB1	DB1				SB	SB
SB		DL3	DL3	DL3			
DB2	DB2	DB2					

In the second step, the algorithm groups the bit ranges according to the values the mode signal may occupy. If a bit range does not contain any bit depending on a certain mode signal value, the bit range is removed from the value group. The following figure displays the static bit ranges of the CAN message and the dynamic bit ranges assigned to value 1 of the mode signal.

7	6	5	4	3	2	1	0
0	SL	SL	SL				
1	SL		SL	SL		SL	SL
2						SL	SL
3							
4						SB	SB
5	SB						
6							
7							

7	6	5	4	3	2	1	0
					DL1	DL1	DL1
						DL1	DL1

The last step performs the byte stuffing for the static bit range set and all dynamic bit range sets. If there is an interval $[n*8 .. (n + 1)*8 - 1]$ which does not occupy any bit range, the interval is removed completely,

i.e. all following bit ranges are shifted up by one byte. If remaining full bytes of a CAN message do not occupy any bit range, they are also truncated. The next figure displays the result of the byte stuffing step.

The static part on the left hand side makes up the static `SignalIPdu`, which is five bytes long. The dynamic part on the right hand side makes up the dynamic `SignalIPdu` depending on the switch value 1.

	7	6	5	4	3	2	1	0
0	SL	SL	SL					
1	SL		SL	SL		SL	SL	
2						SL	SL	
3						SB	SB	
4	SB							

	7	6	5	4	3	2	1	0
			DL1	DL1	DL1			
						DL1	DL1	DL1
	DB1	DB1	DB1					

The tables below show the properties of the `StaticPart` and entities which are the result of the transformation.

startPosition	bitLength	pduMultiplexerDataPartByteOrder
5	13	MOSTSIGNIFICANTBYTELAST
24	2	MOSTSIGNIFICANTBYTELAST
39	1	MOSTSIGNIFICANTBYTELAST

The tables below show the properties of the `DynamicPart` entities depending on the switch value 1 which are the result of the transformation.

startPosition	bitLength	pduMultiplexerDataPartByteOrder	selector-FieldCode
3	8	MOSTSIGNIFICANTBYTELAST	1
21	3	MOSTSIGNIFICANTBYTELAST	1

Moreover, the algorithm creates one `SignalIPdu` entity for every selector field code value to which the corresponding `DynamicPart` entities refer. If one or more `StaticPart` entities have been created, a corresponding `SignalIPdu` is created to which all `StaticPart` entities refer. If any of the `StaticPart` and/or `DynamicPart` entities were shifted during the byte stuffing step, the contained signals are shifted by the same amount of bytes to ensure consistency.

In the second mode **Map selector fields to Com Signals**, the selector field code of a Multiplexer PDU is handled as an ordinary *mode dependent signal*, thus the configured example differs in two aspects:

- ▶ Every dynamic `SignalIPdu` owns a bit range occupying the bits in which the selector field resides.
- ▶ Every dynamic `SignalIPdu` additionally contains the selector field as dedicated signal.

The following paragraphs outline the layout of the static and dynamic PDUs if the third mode (**AUTOSAR compliant mapping**) is selected.

The example below displays the original Multiplexer configuration. In addition, the selector field code "SFC" is displayed occupying bits 0 and 1.

	7	6	5	4	3	2	1	0
0	SL	SL	SL				SFC	SFC
1	SL		SL	SL		SL	SL	SL
2	DL2		DL1	DL1	DL1		SL	SL
3						DB1	DB1	DB1
4	DB1	DB1	DB1				SB	SB
5	SB		DL3	DL3	DL3			
6								
7	DB2	DB2	DB2					

This mode leads to the following layout of the contained static and dynamic SignalIPdus. The static SignalIPdu is shown on the left hand side, the layout of all dynamic SignalIPdus is displayed on the right hand side:

	7	6	5	4	3	2	1	0
0	SL	SL	SL					
1	SL		SL	SL		SL	SL	SL
2							SL	SL
3								
4						SB	SB	
5	SB							

	7	6	5	4	3	2	1	0	
0							SFC	SFC	
1									
2	DL2		DL1	DL1	DL1				
3							DB1	DB1	DB1
4	DB1	DB1	DB1						
5			DL3	DL3	DL3				
6									
7	DB2	DB2	DB2						

The bit range stuffing step as described above is omitted which causes empty bytes in the SignalIPdus. Since the bit ranges of all dynamic parts must be identical, this import mode does not take into account whether a bit range is actually required for a dynamic SignalIPdu or not. This leads to identical layouts of all dynamic SignalIPdus. In comparison to the other import modes, this mode may lead to a significant higher amount of required RAM, especially if there are many dynamic SignalIPdus (i.e. many different selector field codes) per Multiplexed PDU.

7.10.2.7. Importing from the LDF format

EB tresos Studio provides an integrated LDF Importer to import LIN configurations from LDF files. If an LDF file is available for import, the LDF Importer simplifies and speeds up the configuration of the LIN stack.

7.10.2.7.1. Overview

The LDF Importer can read an LDF file and is able to import into the system model of the project and to configure one or more of the following modules:

- ▶ Com (Communication)
- ▶ PduR (PDU Router)
- ▶ LinIf (LIN Interface)
- ▶ Lin (LIN Driver)
- ▶ EcuC (Pseudo module that contains the global PDU list)

If an ECU configuration is imported, the LDF Importer imports frames and signals into the respective module configurations. The PDUs created are routed from LinIf to Com (receive frames) and from Com to LinIf (transmit frames) respectively.

During the import, the LDF Importer creates many containers for messages and signals. The names chosen for these containers are based on the names defined in the LDF file. Optionally all names can be extended by a user defined prefix and/or a calculated instance suffix. If you want to know which names are created in which way, refer to the EB tresos AutoCore documentation.

7.10.2.7.2. LDF Importer and file format

LDF (Lin Description File) is a standardized and open file format for LIN networks. LDF files describe the nodes connected to the LIN network, frames and signals transmitted on the network, and Schedule Tables describing the transmission pattern. A LIN network contains one master node and one or more Slave nodes. The master node is the only node for which configuration data is imported since Slave nodes are not considered to be AUTOSAR nodes.

A specification on the file format and on the LIN protocol itself can be obtained from <http://www.lin-subbus.org>.

7.10.2.7.3. Importing LDF configurations

The LDF Importer allows to import into system model and into the ECU configuration of the project. The GUI of the importer allows to enter parameters for both import scenarios.

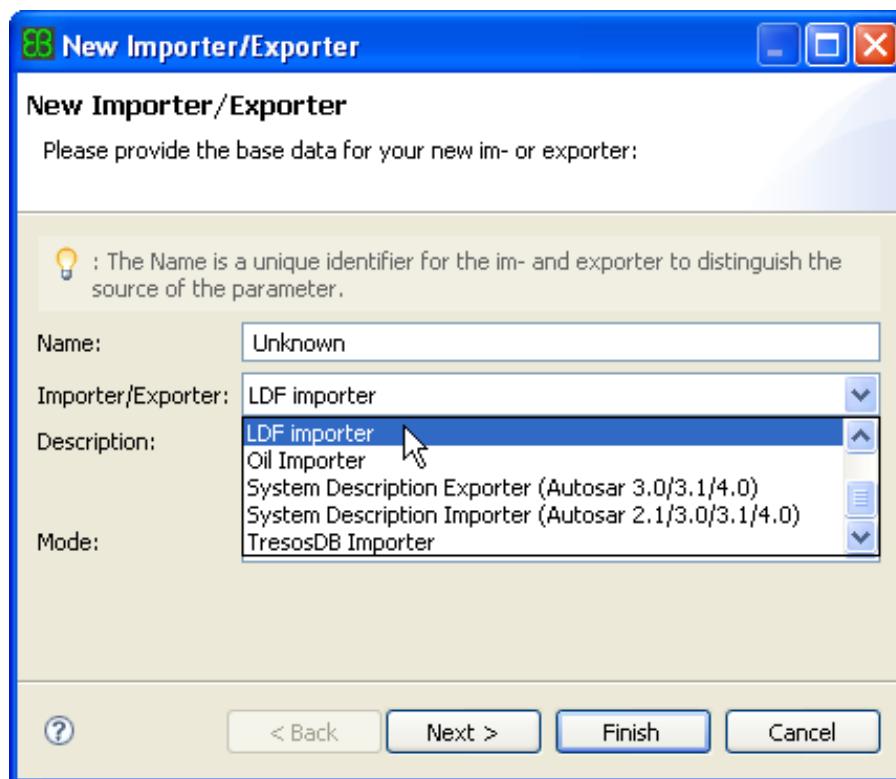
To import configurations from the LDF format:

- ▶ Refer to chapter [Section 7.10.2, “Importing and exporting configuration data”](#).
- ▶ Follow the instructions of Step 1 until you arrive at the window *Create, manage and run im- and exporters*.

In the window *Create, manage and run im- and exporters*:

- ▶ Select the **Create a new im- or exporter** button.

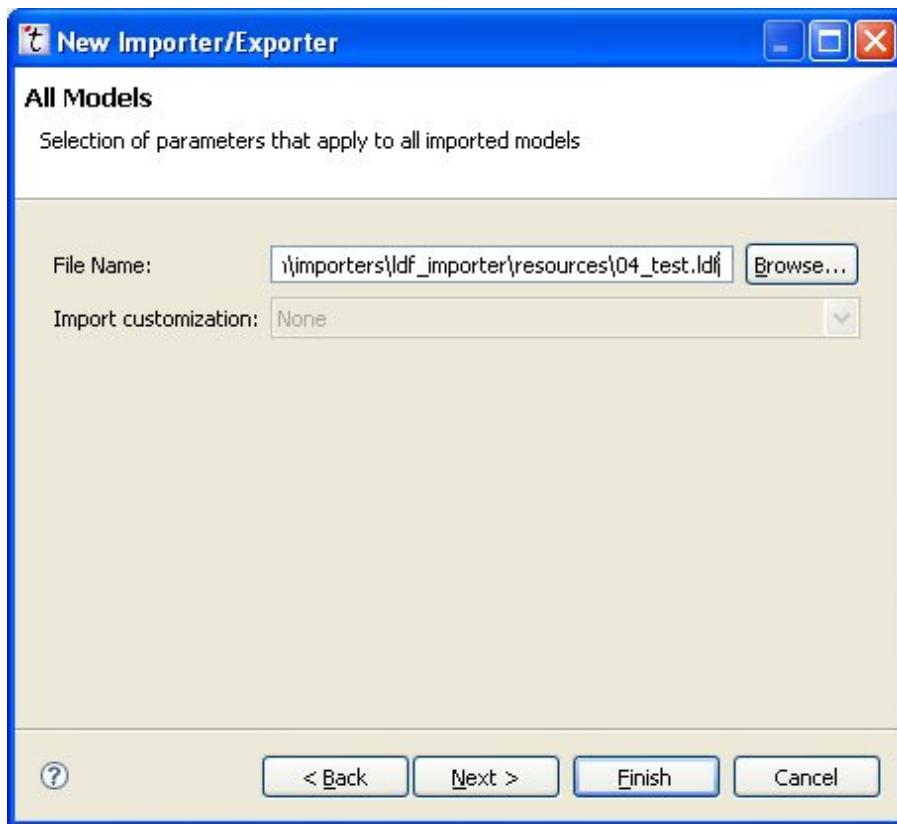
The *New Importer/Exporter* window opens up.



To enter the basic information for the new importer or exporter configuration:

- ▶ Select the *Name* textbox.
- ▶ Enter a name for your new configuration.
- ▶ Select *LDF Importer* from the **Importer/Exporter** dropdown listbox.
- ▶ Select the *Mode* dropdown listbox.
- ▶ Select **Import only** from the dropdown listbox.
- ▶ Select **Next**.

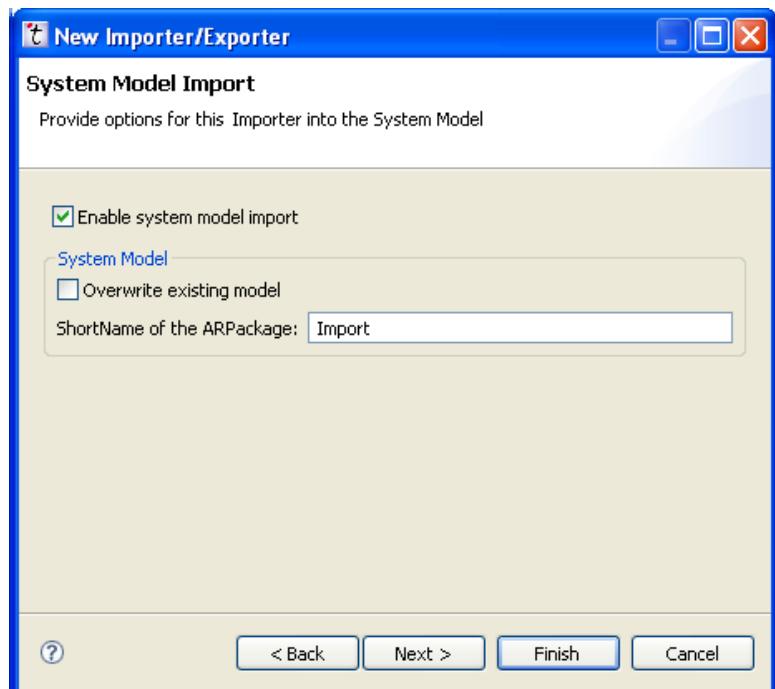
After pressing the **Next** button you enter a page, in which you may enter general parameters.



- ▶ In the *File Name* file selection entry enter the name of an LDF file.
- ▶ If the *Import customization* drop-down list box is enabled, select an import customization.

The *Import customization* drop-down list box provides available import customizations registered for this kind of importer. An import customization affects the import process and modifies the imported data. In most cases this list will be empty and thus grayed out.

Press the **Next** button to switch to a page, in which you may enter parameters for the system model import.

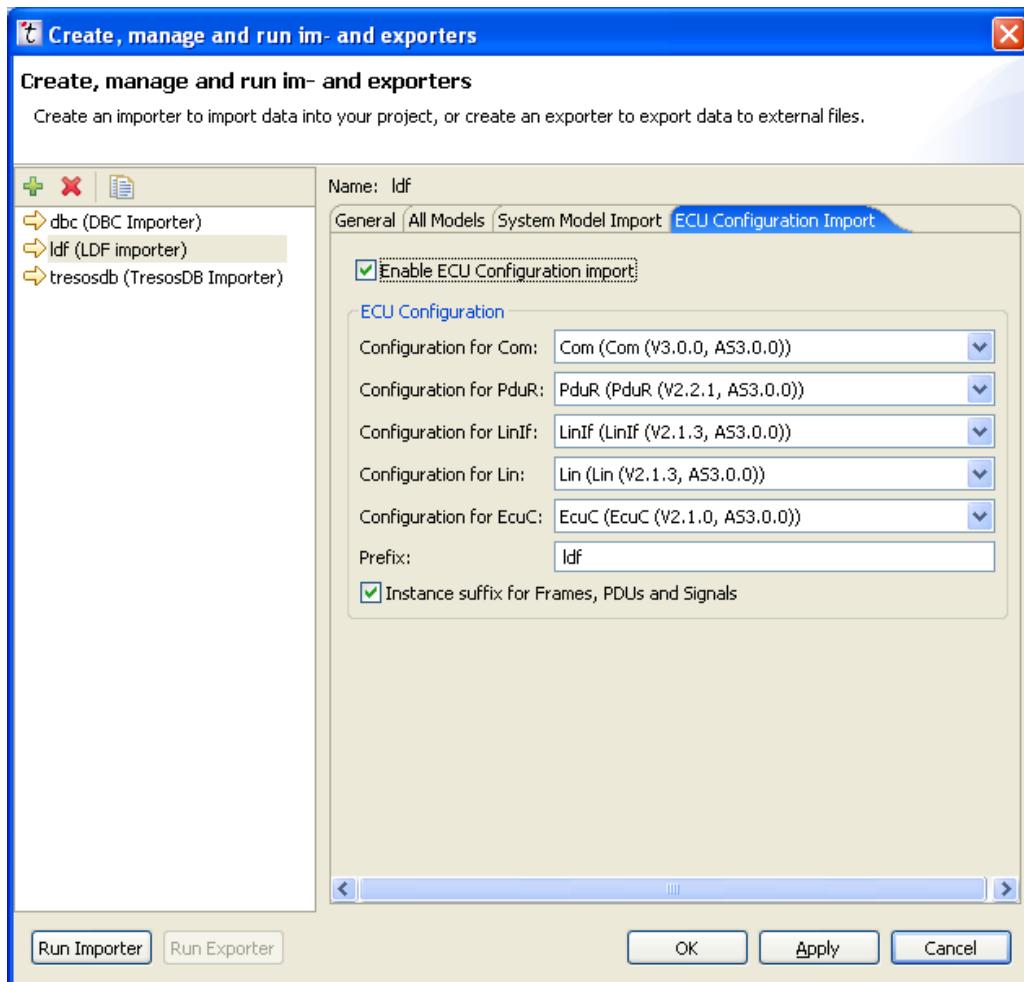


- ▶ To enable system model import into the project, check the *Enable system model import* checkbox. If the project has no system model, the complete page including the checkbox is greyed out.
- ▶ To completely replace the system model in the project, check the checkbox *Overwrite existing model*.
- ▶ To select the AUTOSAR SHORT-NAME of the AUTOSAR AR-PACKAGE into which the system model of the source file is to be imported, enter the AUTOSAR SHORT-NAME into the **ShortName of the ARPackage** textbox.
- ▶ Select **Next** to proceed to the next configuration window.

The *ECU Configuration Import* window is provided for entering ECU configuration-specific parameters.

NOTE

Since the LDF Importer does not create module configurations, you need to assure that a module configuration is present for every module you would like to import into (Com, PduR, LinIf, Lin, EcuC).



- ▶ Check the *Enable ECU Configuration import* checkbox to enable importing ECU configuration parameters.
- Uncheck the *Enable ECU Configuration import* checkbox to disable importing ECU configuration parameters.
- ▶ In the *Configuration for Com* dropdown listbox, select the target Com configuration for the import.
- ▶ In the *Configuration for PduR* dropdown listbox, select the target PduR configuration for the import.
- ▶ In the *Configuration for LinIf* dropdown listbox, select the target LinIf configuration for the import.
- ▶ In the *Configuration for Lin* dropdown listbox, select the target Lin configuration for the import.
- ▶ In the *Configuration for EcuC* dropdown listbox, select the target EcuC configuration for the import.
- ▶ For details on the *Prefix* text box refer to the EB tresos AutoCore documentation.

- ▶ For details on the *Instance Handling* check box refer to the EB tresos AutoCore documentation.

To save the importer configuration:

- ▶ Select **Finish**.

This saves your importer configuration and closes the *LDF Importer Options* window.

To run the LDF Importer, select the importer name you just created on the left side of the *Create, manage, and run im- and exporters* window.

- ▶ Select **Run Importer**.

The LDF configuration is being imported.

7.10.2.7.4. References

Details on the LDF to AUTOSAR parameter mapping are available in the EB tresos AutoCore documentation.

7.10.2.8. Importing from the Fibex format

7.10.2.8.1. Overview

EB tresos Studio provides an integrated Fibex Importer to import configurations from Fibex files.

The EB tresos Studio Fibex Importer is able to read a Fibex file, fill in the system model of a project and to configure one or more of the following modules:

- ▶ Com (Communication)
- ▶ PduR (PDU Router)
- ▶ IpduM (IPDU Multiplexer)
- ▶ FrIf (FlexRay Interface)
- ▶ Fr (FlexRay Driver)
- ▶ FrTp (FlexRay Transport Layer)
- ▶ FrNm (FlexRay Network Management)
- ▶ EcuC (EcuC Module)
- ▶ Can (Can Driver)

- ▶ CanNm (Can Network Management)
- ▶ CanTp (Can Transport Layer)
- ▶ CanIf (CAN Interface)
- ▶ Nm (Network Management Interface)

Supported Fibex versions are: 3.0.0, 3.1.0, 3.1.1. For detailed background information please refer to the Fibex specification available at <http://www.asam.net>.

7.10.2.8.2. Background information

Limitations

Known restrictions/limitations:

- ▶ Only FlexRay and CAN clusters are imported.
- ▶ When importing Fibex 3.x files, the Fibex Importer processes only the first `TIMING` parameter of each type (`CYCLIC`, `EVENT-CONTROLLED`, `REQUEST-CONTROLLED`) of a given PDU.
- ▶ For Fibex 3.x the schema has been extended to allow multiple `SEGMENT-POSITIONS` within a `STATIC-PART/DYNAMIC-PART`. Also, the optional tag `BIT-POSITION-DEMUX` has been added as child of a `SEGMENT-POSITION`.

These modifications do not affect the validity of existing Fibex 3.x instances.

7.10.2.8.3. Importing Fibex configurations

The Fibex Importer enables importing configurations into the system model and into the ECU configuration of the project. The GUI of the importer allows to enter parameters for both these import scenarios.

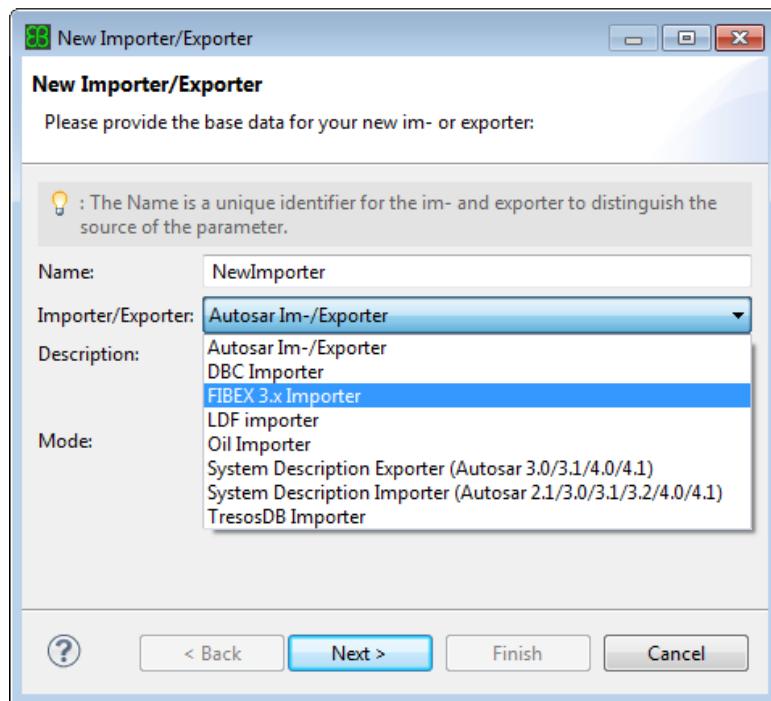
To import configurations from the Fibex format:

- ▶ Refer to [Section 7.10.2, “Importing and exporting configuration data”](#).
- ▶ Follow the instructions of Step 1 until you arrive at the window *Create, manage and run im- and exporters*.

In the window *Create, manage and run im- and exporters*:

- ▶ Select the **Create a new im- or exporter** button.

The *New Importer/Exporter* window opens up.



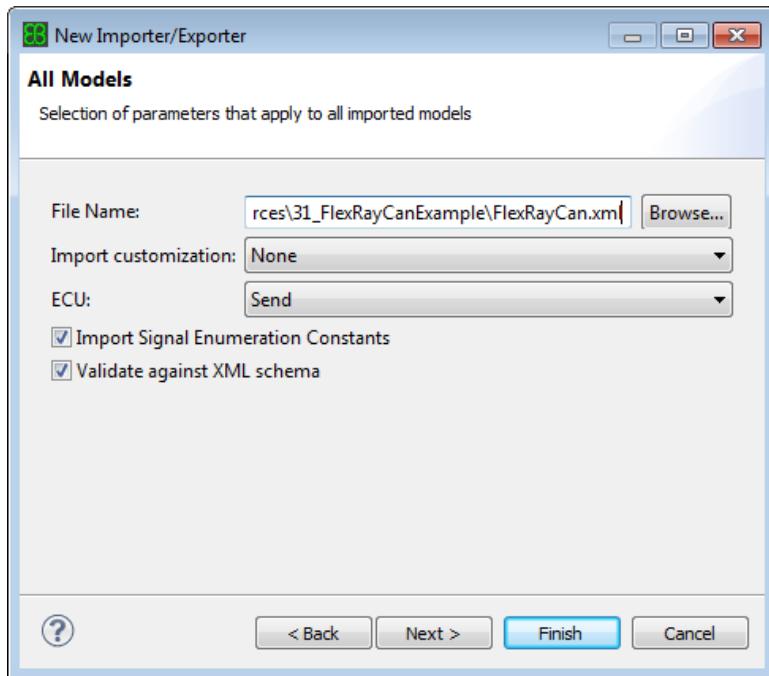
To enter the basic information for the new importer or exporter configuration:

- ▶ Select the *Name* textbox.
- ▶ Enter a name for your new configuration.
- ▶ Select *Fibex 3.x Importer* from the **Importer/Exporter** dropdown listbox.
- ▶ Select **Import only** from the *Mode* dropdown listbox.

The Fibex Importer can only be used to import files. Exporting to Fibex format is not possible.

- ▶ Select **Next**.

On the following page generic information can be entered.



- ▶ Use the **File Name** file selection entry to select the Fibex file that should be imported.
- ▶ If the *Import customization* drop-down list box is enabled, select an import customization.

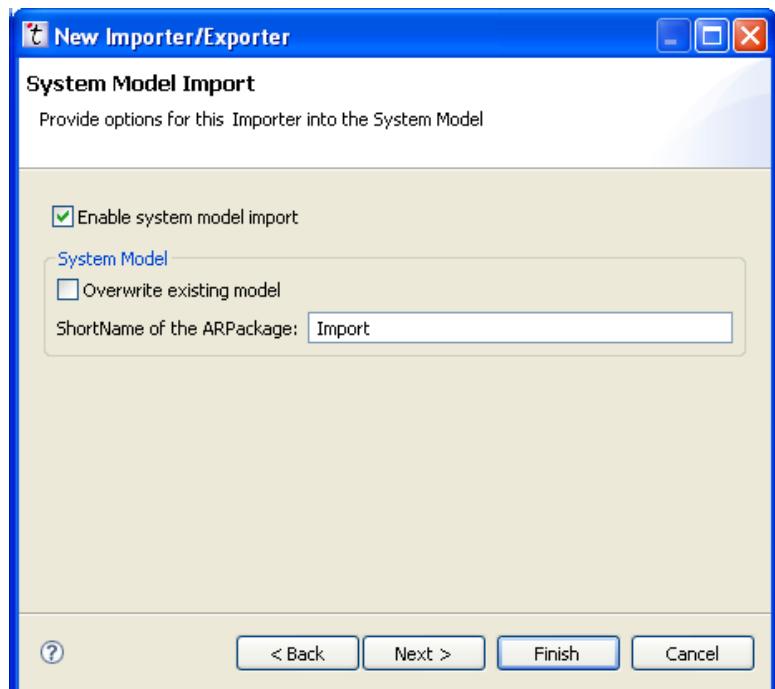
The *Import customization* drop-down list box provides available import customizations registered for this kind of importer. An import customization affects the import process and modifies the imported data. In most cases this list will be empty and thus grayed out.

- ▶ Via the **ECU** dropdown listbox the ECU that should be imported from the Fibex file must be selected.
- ▶ To import COMPU-CONST elements from the Fibex file, select the check box **Import Signal Enumeration Constants**.

For background information on this option refer to [Section 7.10.2.6.3, “Importing DBC configurations”](#).

- ▶ If parser errors occur during an import, you can turn off the XML validation by deselecting the check box **Validate against XML schema**.

Press the **Next** button to switch to a page where parameters for the system model import can be entered.

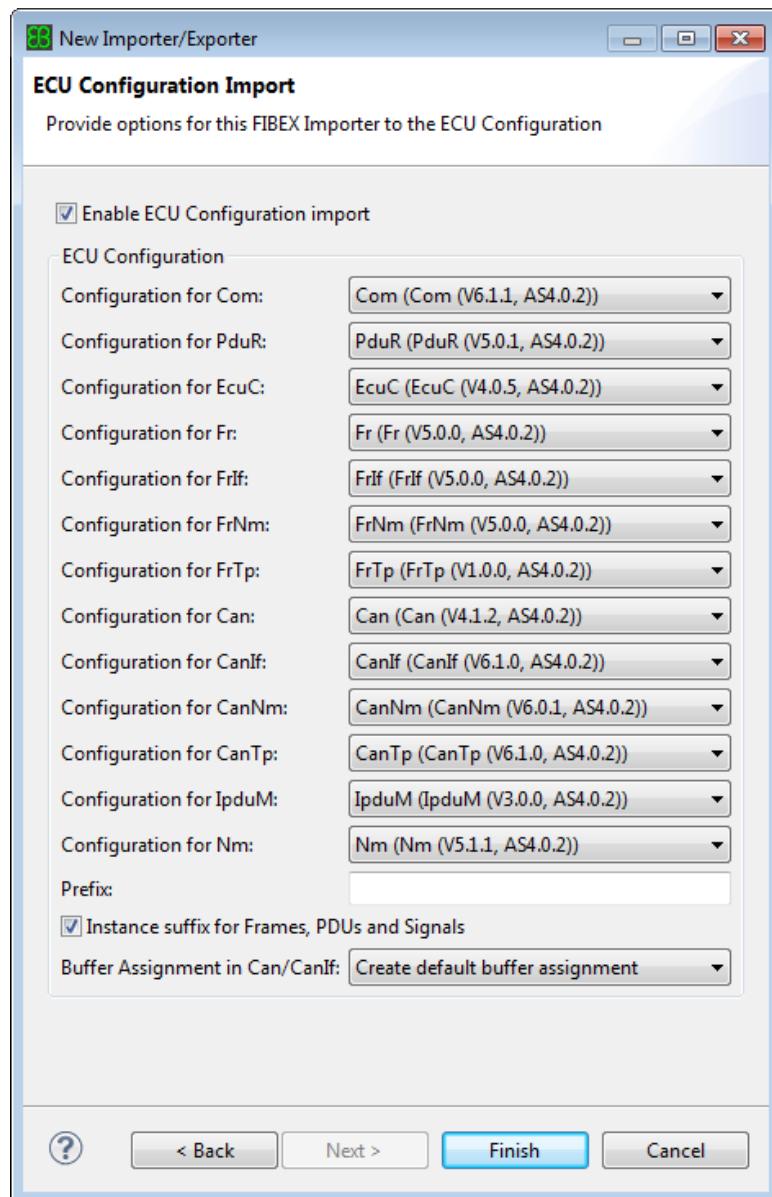


- ▶ To enable system model import into the project, check the *Enable system model import* checkbox. If the project has no system model, the complete page including the checkbox is greyed out.
- ▶ To completely replace the system model in the project, check the checkbox *Overwrite existing model*.
- ▶ To select the AUTOSAR SHORT-NAME of the AUTOSAR AR-PACKAGE into which the system model of the source file is to be imported, enter the AUTOSAR SHORT-NAME into the **ShortName of the ARPackage** textbox.
- ▶ Select **Next** to proceed to the next configuration window.

The *ECU Configuration Import* window is provided for entering ECU configuration-specific parameters.

NOTE

Since the Fibex Importer does not create module configurations, you need to ensure that a module configuration is present for every module you would like to import. Modules available are: Com, PduR, IpduM, FrIf, Fr, FrTp, FrNm, EcuC, Can, CanNm, CanTp, CanIf, Nm.



- ▶ The *Enable ECU Configuration Import* check box allows to disable ECU configuration import.
- ▶ To import into a `Com` module configuration, select it in the *Configuration for Com* drop-down list box.
- ▶ To import into a `PduR` module configuration, select it in the *Configuration for PduR* drop-down list box.
- ▶ To import into a `EcuC` module configuration, select it in the *Configuration for EcuC* drop-down list box.
- ▶ To import into a `Fr` module configuration, select it in the *Configuration for Fr* drop-down list box.

- ▶ To import into a `FrTp` module configuration, select it in the *Configuration for FrTp* drop-down list box.
- ▶ To import into a `FrNm` module configuration, select it in the *Configuration for FrNm* drop-down list box.
- ▶ To import into a `FrIf` module configuration, select it in the *Configuration for FrIf* drop-down list box.
- ▶ To import into a `Can` module configuration, select it in the *Configuration for Can* drop-down list box.
- ▶ To import into a `CanNm` module configuration, select it in the *Configuration for CanNm* drop-down list box.
- ▶ To import into a `CanTp` module configuration, select it in the *Configuration for CanTp* drop-down list box.
- ▶ To import into a `CanIf` module configuration, select it in the *Configuration for CanIf* drop-down list box.
- ▶ To import into a `IpduM` module configuration, select it in the *Configuration for IpduMplex* drop-down list box.
- ▶ To import into an `Nm` module configuration, select it in the *Configuration for Nm* drop-down list box.
- ▶ For details on the *Prefix* text box refer to the EB tresos AutoCore documentation.
- ▶ For details on the *Instance Handling* drop-down list box refer to the EB tresos AutoCore documentation.

To save the importer configuration:

- ▶ Select **Finish**.

This saves your importer configuration and closes the *FIBEX Importer Options* window.

- ▶ To run the Fibex Importer, select the importer name you just created on the left side of the *Create, manage, and run im- and exporters* window.
- ▶ Select **Run Importer**.

The Fibex configuration is being imported.

7.10.2.9. Importing configurations from the TDB format

7.10.2.9.1. Overview

EB tresos Studio provides an integrated TDB Importer to import a system model as well as ECU configurations from TDB files into the project.

7.10.2.9.2. Background information

When importing into the ECU configuration the EB tresos Studio TDB Importer is able to read a TDB file and to configure one or more of the following modules:

- ▶ Com (Communication)
- ▶ PduR (PDU Router)
- ▶ EcuC (ECU Configuration)
- ▶ Can (Can Driver)
- ▶ CanIf (Can Interface)
- ▶ CanTp (CAN Transport Layer)
- ▶ CanNm (CAN Network Management)
- ▶ Fr (FlexRay Driver)
- ▶ FrIf (FlexRay Interface)
- ▶ FrTp (FlexRay Transport Layer)
- ▶ FrNm (FlexRay Network Management)
- ▶ Lin (LIN Driver)
- ▶ LinIf (LIN Interface)
- ▶ LinTp (LIN Transport Protocol)
- ▶ IpduM (IPDU Multiplexer)
- ▶ Nm (Network Management Interface)

Limitations

Lin and LinIf module configurations can only be imported if they are in AUTOSAR version 2.1.

7.10.2.9.3. Importing TDB configurations

The TDB Importer enables importing into system model and into the ECU configuration of the project. The GUI of the importer allows to enter parameters for both import scenarios.

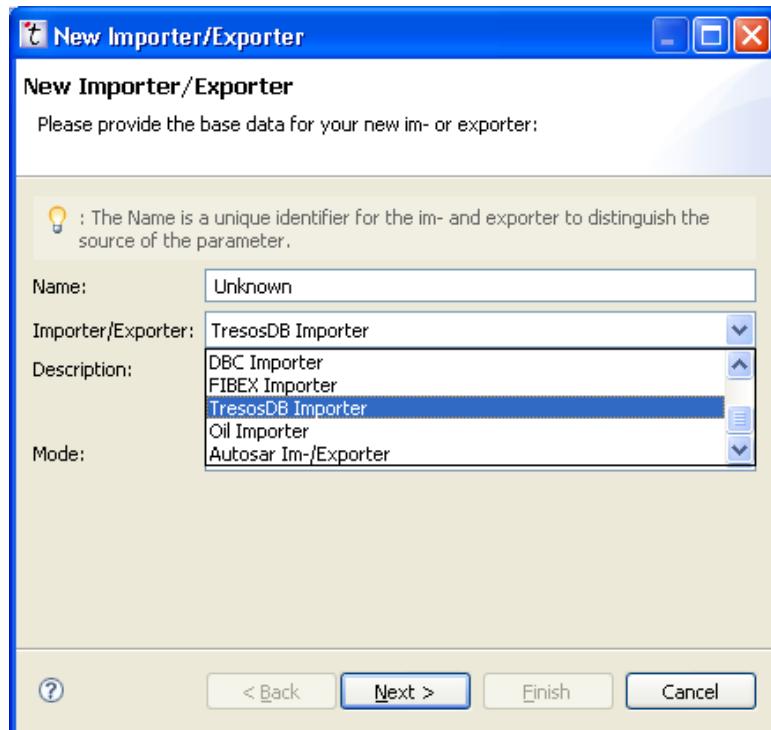
To import configurations from the TDB format:

- ▶ Refer to [Section 7.10.2, “Importing and exporting configuration data”](#).
- ▶ Follow the instructions of Step 1 until you arrive at the window *Create, manage and run im- and exporters*.

In the window *Create, manage and run im- and exporters*:

- ▶ Select the **Create a new im- or exporter** button.

The *New Importer/Exporter* window opens up.



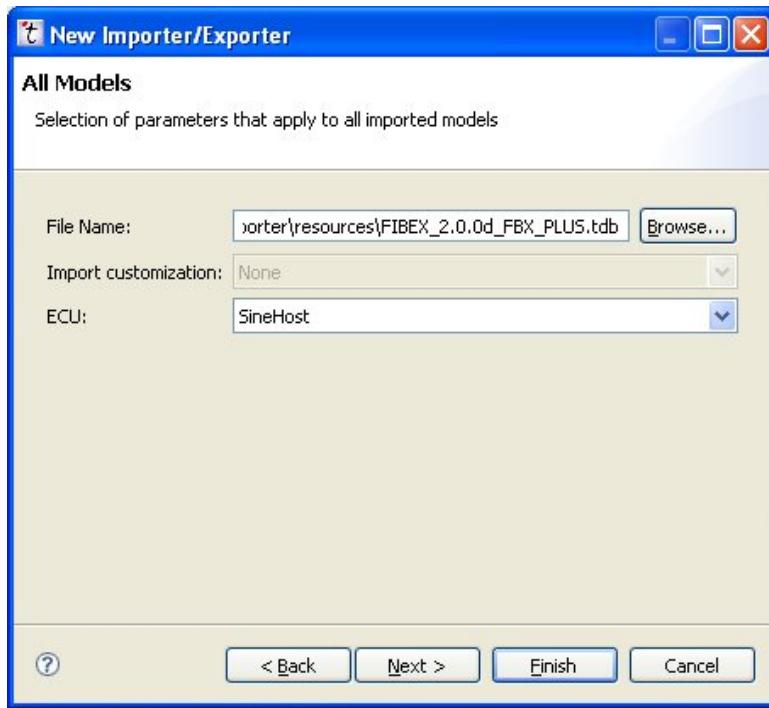
To enter the basic information for the new importer or exporter configuration:

- ▶ Select the *Name* textbox.
- ▶ Enter a name for your new configuration.
- ▶ Select *TDB Importer* from the **Importer/Exporter** dropdown listbox.
- ▶ Select **Import only** from the *Mode* dropdown listbox.

The TDB Importer can only be used to import files. Exporting to TDB format is not possible.

- ▶ Select **Next**.

After pressing the **Next** button, you may enter general information.

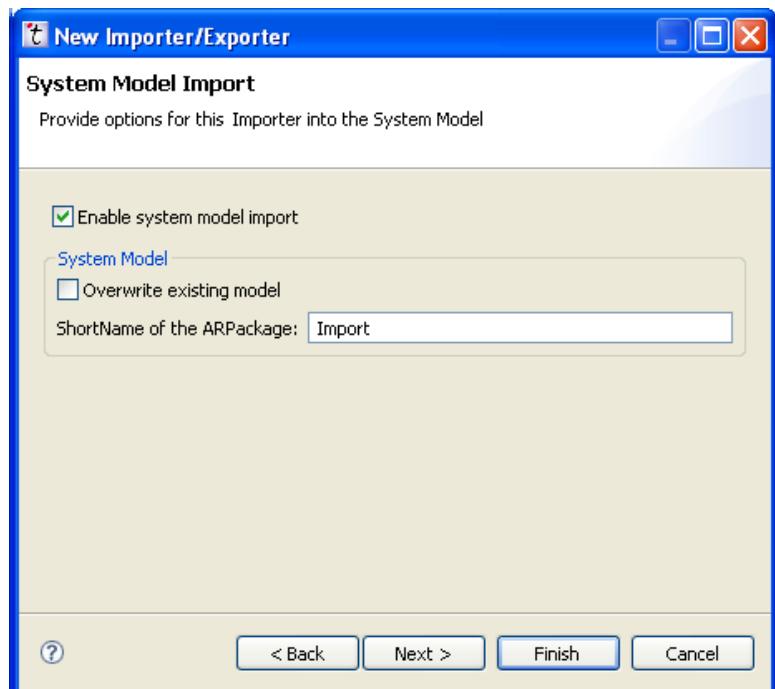


- ▶ Use the **File Name** file selection entry to select the TDB file that you would like to import.
- ▶ If the *Import customization* drop-down list box is enabled, select an import customization.

The *Import customization* drop-down list box provides available import customizations registered for this kind of importer. An import customization affects the import process and modifies the imported data. In most cases this list will be empty and thus grayed out.

- ▶ To import the ECU from the TDB file, select the ECU from the **ECU** dropdown listbox.

Press the **Next** button to switch to a page where parameters for the system model import can be entered.

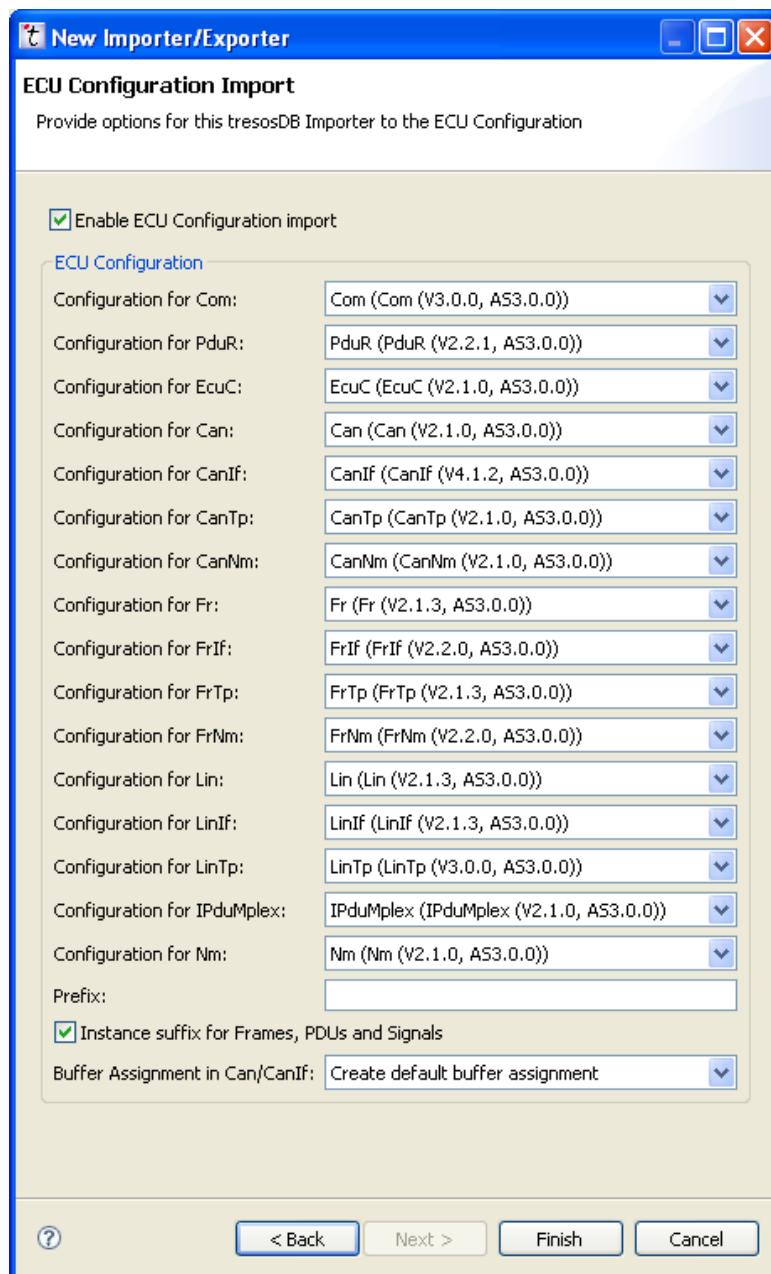


- ▶ To enable system model import into the project, check the *Enable system model import* checkbox. If the project has no system model, the complete page including the checkbox is greyed out.
- ▶ To completely replace the system model in the project, check the checkbox *Overwrite existing model*.
- ▶ To select the AUTOSAR SHORT-NAME of the AUTOSAR AR-PACKAGE into which the system model of the source file is to be imported, enter the AUTOSAR SHORT-NAME into the **ShortName of the ARPackage** textbox.
- ▶ Select **Next** to proceed to the next configuration window.

The *ECU Configuration Import* window is provided for entering ECU configuration-specific parameters.

NOTE

Since the TDB Importer does not create module configurations, you need to assure that a module configuration is present for every module you would like to import into (EcuC, Com, PduR, IpduM, CanIf, Can, FrIf, Fr, FrTp, FrNm, LinIf, Lin, CanTp, CanNm, LinTp, Nm).



- ▶ To disable ECU configuration import, select the *Enable ECU Configuration Import* check box .
- ▶ To import into a Com module configuration, select it in the *Configuration for Com* drop-down list box.
- ▶ To import into a PduR module configuration, select it in the *Configuration for PduR* drop-down list box.

- ▶ To import into a `EcuC` module configuration, select it in the *Configuration for EcuC* drop-down list box.
- ▶ To import into a `Can` module configuration, select it in the *Configuration for Can* drop-down list box.
- ▶ To import into a `CanIf` module configuration, select it in the *Configuration for CanIf* drop-down list box.
- ▶ To import into a `CanTp` module configuration, select it in the *Configuration for CanTp* drop-down list box.
- ▶ To import into a `CanNm` module configuration, select it in the *Configuration for CanNm* drop-down list box.
- ▶ To import into an `Fr` module configuration, select it in the *Configuration for Fr* drop-down list box.
- ▶ To import into an `FrIf` module configuration, select it in the *Configuration for FrIf* drop-down list box.
- ▶ To import into an `FrTp` module configuration, select it in the *Configuration for FrTp* drop-down list box.
- ▶ To import into an `FrNm` module configuration, select it in the *Configuration for FrNm* drop-down list box.
- ▶ To import into a `Lin` module configuration, select it in the *Configuration for Lin* drop-down list box.
- ▶ To import into a `LinIf` module configuration, select it in the *Configuration for LinIf* drop-down list box.
- ▶ To import into a `LinTp` module configuration, select it in the *Configuration for LinTp* drop-down list box.
- ▶ To import into a `IpduM` module configuration, select it in the *Configuration for IPduMplex* drop-down list box.
- ▶ To import into an `Nm` module configuration, select it in the *Configuration for Nm* drop-down list box.
- ▶ For details on the *Prefix* text box refer to the EB tresos AutoCore documentation.
- ▶ For details on the *Instance Handling* check box refer to the EB tresos AutoCore documentation.

To save the importer configuration:

- ▶ Select **Finish**.

This saves your importer configuration and closes the *TDB Importer Options* window.

- ▶ To run the TDB Importer, select the importer name you just created on the left side of the *Create, manage, and run im- and exporters* window.
- ▶ Select **Run Importer**.

The TDB configuration is being imported.

7.10.2.9.4. Mapping TDB parameters to AUTOSAR parameters

Details on the TDB to AUTOSAR parameter mapping are available in the EB tresos AutoCore documentation.

7.11. Working with the command line

EB tresos Studio provides a commandline interface to trigger

- ▶ code generation and verification
 - ▶ for configuration projects (for details, refer to [Section 7.11.2, “Commands for project-based code generation and configuration”](#)),
 - ▶ independently from projects (for details, refer to [Section 7.11.3, “Commands for legacy code generation”](#)),
- ▶ module development functions (for details, refer to [Section 7.11.4, “Commands for module development”](#)), and
- ▶ file conversion functions (for details, refer to [Section 7.11.5, “Commands for converting and merging files”](#)).

7.11.1. General

7.11.1.1. Starting the commandline

To start the commandline:

- ▶ Open the Windows command prompt.
- ▶ Change into the directory, into which you have installed EB tresos Studio; e.g. C:\EB\tresos.
- ▶ Change into the `bin` subdirectory.
- ▶ You can execute a command by using `tresos_cmd.bat`; e.g. `type tresos_cmd.bat --help` to see a short help for the commandline options.

7.11.1.2. The commandline syntax

An example of a commandline is:

```
tresos_cmd.bat -Dinfo=false -data c:\workspace generate myproject
```

The EB tresos Studio commandline has a general syntax, which is used throughout all available commands. Each command is composed of specific blocks. Some of the blocks may be left out, but the overall order of the blocks is important for the command to be recognized properly. A commandline consists of the following blocks:

```
tresos_cmd.bat [<system_property>...] [-data <workspace_dir>]  
<command>
```

The position of these blocks is important.

Block	Description
<system_property>	<p>System properties control the behavior of the command. Some system properties are understood by every command and some are only valid for a specific command. The list of system properties understood by all commands is available in Section 7.11.1.3, “System properties common to all commands”.</p> <p>System properties are always the first block on the commandline. You may specify as many system properties as you want, or none at all.</p>
<workspace_dir>	<p>This block lets you control on which workspace the commandline operates. You may omit this block. Details of the contents of the block are available in Section 7.11.1.4, “Specifying the workspace on the commandline”.</p>
<command>	<p>This block tells EB tresos Studio what to do, e.g. to generate code. This block consists of the command itself and additional parameters that control what the command does. The other sections of this chapter describe all the commands that are available and their parameters.</p>

7.11.1.3. System properties common to all commands

The following system properties can be used on all the commands listed below:

System property	Description
-Dinfo=false	Disables printing of any info messages.
-Dwarning=false	Disables printing of any warning messages.
-Derror=false	Disables printing of any error messages.
-Dmsg<code>=false	Disables printing of any messages with the given numerical error code <code>.
-Ddebug.packages=<Package>:<Package>:...	Enables Log4J debugging for specific Java packages.
-DValidate=<boolean>	<p>Enables or disables the validation of read and written files (e.g. against an AUTOSAR XSD schema-file).</p> <p>If this property is not set, the behavior is context dependent: by default read files are not validated while written files are validated.</p>

System property	Description
	NOTE  AUTOSAR configuration files written by a code generator are never validated, even if this property is set to <code>true</code> .
<code>-Dbase.path=<directory></code>	The base directory for the command.
<code>-DrestrictShortName=true</code>	Enables the preference Use maximum SHORT-NAME length of 32 chars (AUTOSAR-conform) for AUTOSAR releases prior to 4.0

An example for the use of system properties is:

```
tresos_cmd.bat -Dinfo=false -Dmsg1234=false generate myproject
```

This code example generates code for a project called *myproject*.

When the commandline is executed, the output does not contain any informational messages (`-Dinfo=false`) and additionally the output of the message with the code `1234` is also suppressed (`-Dmsg1234=false`).

7.11.1.4. Specifying the workspace on the commandline

By default EB tresos Studio loads the workspace used in the last program run (for an explanation of the concept of workspaces, refer to [Section 4.5, “EB tresos Studio projects and workspaces”](#)).

When you execute commands on the commandline, you may want EB tresos Studio to use a specific workspace directory and not the one you used last.

- ▶ To specify another workspace directory, use the `-data` parameter.

This parameter changes the workspace directory for the current program run only. If you execute another command, or start the GUI, the workspace used previously is reused.

The `-data` option must be placed behind `tresos_cmd.bat`, after any system properties, and takes a directory as argument.

The following is an example for how you may change the workspace:

```
tresos_cmd.bat -Dinfo=false -data c:\workspace generate myproject
```

This code generates code for a project called *myproject*. As workspace, the directory `c:\workspace` is used. This implies that the project *myproject* is part of that workspace. When the commandline is executed, the output does not contain any informational messages.

TIP

You may also omit the system property `-Dinfo=false`. It is only present in the code example above to demonstrate the correct order of the commandline parameters.

The following example does the same as the example above, but uses the workspace which was used last:

```
tresos_cmd.bat -Dinfo=false generate myproject
```

7.11.2. Commands for project-based code generation and configuration

Use the project-based mode of the command line to generate code, verify configurations, or to change the configuration for an existing EB tresos Studio project stored in an EB tresos Studio workspace (see [Section 4.5, “EB tresos Studio projects and workspaces”](#)).

7.11.2.1. Importing projects

To access projects for generating 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 EB tresos Studio command line allows to import projects found in the file system into the current workspace.

7.11.2.1.1. Syntax

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

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

7.11.2.1.2. System properties

This command only recognizes the system properties common to all commands (see [Section 7.11.1.3, “System properties common to all commands”](#)).

7.11.2.1.3. Parameters

In addition to the common parameters `<system_property>` and `<workspace>`, this command also recognizes the following parameters:

Parameter	Description
<code>-c</code>	If you provide the <code>-c</code> commandline switch, a copy of the project directory is added to the workspace directly beneath the workspace root directory. The directory name is the same as the project name.
<code><project path></code>	A path to an Eclipse project (a directory with a <code>.project</code> file).

7.11.2.1.4. Example

The following code is an example of how to use the command `importProject`:

```
tresos_cmd.bat importProject -c C:\temp\myproject
```

This command imports the project located in `c:\temp\myproject` into the current workspace. The `-c` command line option instructs EB tresos Studio to copy the project directory into the directory `myproject` beneath the workspace directory.

7.11.2.2. Deleting projects

To support batch processing, the EB tresos Studio commandline allows to remove projects from the current workspace.

7.11.2.2.1. Syntax

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

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

7.11.2.2.2. System properties

This command only recognizes the system properties common to all commands (see [Section 7.11.1.3, “System properties common to all commands”](#)).

7.11.2.2.3. Parameters

In addition to the common parameters <system_property> and <workspace>, this command also recognizes the following parameters:

Parameter	Description
-d	If you provide the -d commandline switch, the content of the project directory is removed from the file system.
<project>	The name of an Eclipse project located in the current workspace.

7.11.2.2.4. Example

The following code is an example of how to use the command deleteProject:

```
tresos_cmd.bat deleteProject -d myproject
```

This command removes the project named `myproject` from the current workspace. The `-d` commandline option instructs EB tresos Studio to delete the project directory from the file system.

7.11.2.3. Executing importers configured for a project

For each project you can configure importers via the GUI. These importers allow to import or export configuration data from or to ECU configuration formats or system formats such as DBC, LDF, etc. The commandline of EB tresos Studio allows to execute these configured importers.

7.11.2.3.1. Syntax

Importers can execute in import or export mode on the commandline.

To execute an importer in import mode use:

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

To execute an importer in export mode use:

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

7.11.2.3.2. System properties

The importer commands only recognize the system properties common to all commands. For a list of system properties common to all commands, refer to [Section 7.11.1.3, “System properties common to all commands”](#).

7.11.2.3.3. Parameters

In addition to the common parameters `<system_property>` and `<workspace>`, the importer command also recognizes the following parameters:

Parameter	Description
<code><project></code>	The name of the configuration project on which you want the execute importers.
<code><importer></code>	The name of the importer configuration to execute. You may provide this parameter multiple times to run multiple importers. An importer configuration with the given name must exist in the project.

For further information on how to configure importers for a project see [Section 7.10, “Importing and exporting”](#).

7.11.2.3.4. Example

The following commandline is an example of how to use the command `import`:

```
tresos_cmd.bat import myproject myimporter
```

The example executes the importer `myimporter` configured in the project `myproject`.

7.11.2.4. Listing importers configured for a project

For each project you can configure importers via the GUI. These importers allow to import or export configuration data from or to ECU configuration formats or system formats such as DBC, LDF, etc. The commandline of EB tresos Studio allows to list importers configured for a project.

7.11.2.4.1. Syntax

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

7.11.2.4.2. System properties

The **listimporter** command only recognizes the system properties common to all commands (see [Section 7.11.1.3, “System properties common to all commands”](#)).

7.11.2.4.3. System properties

The importer commands only recognize the system properties common to all commands (see [Section 7.11.1.3, “System properties common to all commands”](#)).

7.11.2.4.4. Parameters

In addition to the common parameters `<system_property>` and `<workspace>`, the **listimporter** command also recognizes the following parameters:

Parameter	Description
<code><project></code>	The name of the configuration project for which you want the list importers.

For further information on how to configure importers for a project see [Section 7.10, “Importing and exporting”](#).

7.11.2.4.5. Example

The following commandline is an example of how to use the command `listimporter`:

```
tresos_cmd.bat listimporter myproject
```

The example lists all importers for project *myproject*.

7.11.2.5. Verifying a project

To verify the configuration of all modules and to print out all warnings and errors, use the **verify** command.

7.11.2.5.1. Syntax

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

7.11.2.5.2. System properties

This command only recognizes the system properties common to all commands (see [Section 7.11.1.3, “System properties common to all commands”](#)).

7.11.2.5.3. Parameters

In addition to the common parameters `<system_property>` and `<workspace>`, this command also recognizes the following parameters:

Parameter	Description
<code><project></code>	The name of the configuration project on which you want the command to run.

7.11.2.5.4. Example

The following code is an example of how to use the command `verify`:

```
tresos_cmd.bat -Dinfo=false -Dwarning=false verify myproject
```

This command verifies the configuration of the project *myproject* and does not write warning and info messages as output.

7.11.2.6. Generating code for a project

To verify the configuration and to generate code for the modules of a configuration project and to print out all warnings and errors, use the `generate` command.

7.11.2.6.1. Syntax

```
tresos_cmd.bat [<system_property>...] [-data <workspace>]
    generate [<option>...] <project>
```

7.11.2.6.2. System properties

This command only recognizes the system properties common to all commands (see [Section 7.11.1.3, “System properties common to all commands”](#)).

7.11.2.6.3. Options

The following options are supported by this command:

Parameter	Description
-o <output_dir>	By default, the same output directory as in the GUI is used. If you want to generate your code into a different directory on the commandline, use this option and replace <output_dir> by the directory you want to generate the code to.
-g <module-id>	By default, the modules which are enabled for generation in the GUI will be generated (see Section 7.6, “Editing module configurations”). You can use the -g option to override that setting and to generate only the modules you specify on this commandline. You can use the -g option multiple times if you want to generate code for more than one module. The argument <module-id> must be replaced by the ID of the module you want to generate code for, e.g. Com_TS_T16D4M2IOR0. By default, only the module configurations for which code is being generated are verified. To verify all module configurations, even if no code is generated for them, use the -a option.
<p>NOTE You can only specify module-IDs of modules that are part of your project.</p> 	
-a	If you are using the -g option, only the configurations of the modules specified are verified before the code is generated. If you want all module configurations to be verified, even if no code is generated for them, use this option.

7.11.2.6.4. Parameters

In addition to the common parameters <system_property> and <workspace>, this command also recognizes the following parameters:

Parameter	Description
<project>	The name of the configuration project on which you want the command to run.

7.11.2.6.5. Example

The following code is an example of how to use the command generate:

```
tresos_cmd.bat -Dinfo=false -Dwarning=false generate myproject
```

This command generates code for the modules of the configuration project *myproject* and does not write warning and info messages as output.

7.11.2.7. Listing custom-built generation modes

Some modules may define additional generator modes, which you run with the `make` command (see [Section 7.11.2.8, “Executing custom-built generation modes for projects”](#)). The `listmodes` allows to list all modes supported by the available generators. Whether or not a generator is available, depends on whether the module that provides that generator is part of your project.

7.11.2.7.1. Syntax

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

TIP


Some of the EB tresos AutoCore modules define additional generator modes. If you purchased these modules, please refer to the EB tresos AutoCore documentation to learn more about supported modes.

7.11.2.7.2. System properties

This command only recognizes the system properties common to all commands (see [Section 7.11.1.3, “System properties common to all commands”](#)).

7.11.2.7.3. Parameters

In addition to the common parameters <system_property> and <workspace>, this command also recognizes the following parameters:

Parameter	Description
<project>	The name of the configuration project on which you want the command to run.

7.11.2.7.4. Example

The following code is an example of how to use the command `listmodes`:

```
tresos_cmd.bat listmodes myproject
```

This command lists all the available modes for the configuration project *myproject* and writes them to the commandline.

7.11.2.8. Executing custom-built generation modes for projects

Some modules provide a custom-built generators. You may run these generators by calling a generator mode, other than **generate** and **verify**. The name of the generator mode and what it does depends on the module. To find out if a module contributes a custom mode, please refer to the documentation of the modules you are running with EB tresos Studio.

7.11.2.8.1. Syntax

To run a module with a custom-built generator in your project, use the following commandline:

```
tresos_cmd.bat [<system_property>...] [-data <workspace>]  
make [<option>...] <mode> <project>
```

7.11.2.8.2. System properties

By default, this command only recognizes the system properties common to all commands (see [Section 7.11.1.3, “System properties common to all commands”](#)). But it is possible for custom-built generators to

recognize other system properties. For further information, please refer to the documentation of the generator you are planning to use.

7.11.2.8.3. Parameters

In addition to the common parameters `<system_property>` and `<workspace>`, this command also recognizes the following parameters:

Parameter	Description
<code>-o <output_dir></code>	By default, the same output directory as in the GUI is used. If you want to generate your code into a different directory on the commandline, use this option and replace <code><output_dir></code> with the directory you want to generate the code to.
<code>-g <module-id></code>	By default, the modules, which are enabled for generation in the GUI, are generated (see Section 7.6, “Editing module configurations”). You can use the <code>-g</code> option to override that setting and to generate only the modules you specify on this commandline. You can use the <code>-g</code> option multiple times, if you want to generate code for more than one module. The argument <code><module-id></code> must be replaced by the ID of the module you want to generate code for, e.g. <code>Com_TS-T16D4M2I0R0</code> . Only the module configurations for which code is being generated are verified.
<code><mode></code>	<p>The name of the custom-built generator. Please refer to the documentation of the generator you are planning to use to find out more about the usage.</p> <p>NOTE  To use a custom-built generator on a project, the module which provides that generator must be part of the project.</p> <p>TIP  You can use the <code><listmodes></code> command (see Section 7.11.2.7, “Listing custom-built generation modes”) to find out which generator modes are available.</p>
<code><project></code>	The name of the configuration project on which you want the command to run.

7.11.2.8.4. Example

The following code is an example of how to use the command `make`:

```
tresos_cmd.bat make generate_SWC-T myproject
```

This command runs a custom-built generator called *generate_SWC-T* on the project *myproject*.

7.11.2.9. Executing unattended wizards on projects

If you have created a configuration for an unattended wizard of a project in the GUI, you can execute this wizard on the commandline. For more information about unattended wizard, please refer to [Section 7.8.7, “Autoconfiguring routine jobs”](#).

7.11.2.9.1. Syntax

To execute an unattended wizard:

```
tresos_cmd.bat [<system_property>...] [-data <workspace>]
                                         autoconfigure <project> [<wizard>] ...
```

7.11.2.9.2. System properties

This command does only recognize the system properties common to all commands (see [Section 7.11.1.3, “System properties common to all commands”](#)).

7.11.2.9.3. Parameters

In addition to the common parameters *<system_property>* and *<workspace>*, this command does also recognize the following parameters:

Parameter	Description
<i><project></i>	The name of the configuration project on which you want the unattended wizard to run.
<i><wizard></i>	The name of the unattended wizard that shall be run. You can run multiple wizards in a row by providing several names separated by spaces. If you do not specify any wizards, all configured wizards are executed. To retrieve a list of all available wizards for the given project, please use the <code>listautoconfig</code> -

Parameter	Description
	ure command (see Section 7.11.2.10, “Listing all unattended wizards for a given project”).

7.11.2.9.4. Example

The following code is an example of how to use the command `autoconfigure`:

```
tresos_cmd.bat autoconfigure myproject
```

This command runs all unattended wizards that are configured and checked in the GUI on the project *myproject*. The behaviour of these wizards is controlled by the settings provided in the GUI.

7.11.2.9.5. Running the Calculate Default Values wizard on the command line

To run the Calculate Default Values wizard as configured in the GUI on a project called *myproject*, use:

```
tresos_cmd.bat autoconfigure myproject AutoCalc
```

7.11.2.9.6. Running the Calculate Handle-Ids wizard on the command line

To run the Calculate Handle-Ids wizard as configured in the GUI on a project called *myproject*, use:

```
tresos_cmd.bat autoconfigure myproject AutoHandleId
```

7.11.2.10. Listing all unattended wizards for a given project

You can list all available unattended wizards for a given project on the commandline. For more information about unattended wizards, please refer to [Section 7.8.7, “Autoconfiguring routine jobs”](#).

7.11.2.10.1. Syntax

To list all available unattended wizards for a special project:

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

7.11.2.10.2. System properties

This command does only recognize the system properties common to all commands (see [Section 7.11.1.3, “System properties common to all commands”](#)).

7.11.2.10.3. Parameters

In addition to the common parameters `<system_property>` and `<workspace>`, this command does also recognize the following parameter:

Parameter	Description
<code><project></code>	The name of the configuration project for which you want retrieve the list of available unattended wizards.

7.11.2.10.4. Example

The following code is an example of how to use the command `listautoconfigure`:

```
tresos_cmd.bat listautoconfigure myproject
```

This command lists all unattended wizards that are available for the project *myproject*.

7.11.3. Commands for legacy code generation

In addition to the *project-based* mode, EB tresos Studio also support the *legacy* mode. This mode works independently from projects.

The following section describes all legacy mode commands that handle the generation of code. There are more legacy mode commands which will be explained in later sections.

TIP**Specify a workspace location in legacy mode only if you need it to store e.g. the error log file**

Specifying a workspace location in legacy mode does not have much effect, because there are no projects in legacy mode. However, the workspace is still used internally, e.g. as location for the error log file.

7.11.3.1. Listing all modules available in legacy mode

7.11.3.1.1. Syntax

To print out the names of all installed modules that support legacy code generation, use the `legacy list` command:

```
tresos_cmd.bat [<system_property>...] [-data <workspace>] legacy list
```

7.11.3.1.2. System properties

This command only recognizes the system properties common to all commands. For a list of those commands, refer to [Section 7.11.1.3, “System properties common to all commands”](#).

7.11.3.1.3. Example

The following is an example of how to use the command `legacy list`:

```
tresos_cmd.bat legacy list
```

7.11.3.2. Generating code

7.11.3.2.1. Syntax

Use the `legacy generate` command to verify module configurations and, if no errors occur, generate code for them:

```
tresos_cmd.bat [<system_property>...] [-data <workspace>]
    legacy generate [<options>]
    <config-file>[@<type>]...
```

7.11.3.2.2. System properties

In addition to the common system properties (listed in [Section 7.11.1.3, “System properties common to all commands”](#)), legacy generate command also recognizes the following system properties:

System property	Description
-DmergeConfigs=(true false)	<p>Merges the parameters of the module configurations. This is useful, if you have split the parameters of a module configuration into several files (e.g. one file contains standard and another contains vendor-specific parameters).</p> <p>The default is <code>false</code> for this command.</p> <p>To merge the parameters of the module configurations, set to <code>true</code>.</p> <p>To not merge the parameters, set this property to <code>false</code>. Instead of merging parameters, the content of the file read last overwrites the previous content.</p>
-DverifyReadOnly=(true false)	<p>Enables/disables the verification of read-only configuration parameters. The default is <code>false</code> for this command.</p> <p>To enable the verification of read-only configuration parameters, set to <code>true</code>.</p> <p>To disable the verification of read-only configuration parameters, set to <code>false</code>.</p>
-DssuppressReadOnlyWarnings=(true false)	<p>Suppresses warnings about read-only configuration parameters. The default is <code>false</code> for this command.</p> <p>To suppress warnings, and not log them or print them out, set to <code>true</code>.</p> <p>To enable warnings and log them or print them out, set to <code>false</code>.</p>
-DpredefinedVariant=<Name>	<p>This property enables the variant resolving and specifies the predefined Variant name that will be taken from the input files.</p> <p>If the parameter is not set, by default the variants resolving is disabled.</p> <p>For detailed information about variant handling please refer to Section 7.12, “Variant Handling”.</p>
-Dttarget=<target>	Defines the target architecture, e.g. <code>-Dttarget=WINDOWS</code> . This is necessary when working with module configurations, which use ECU Re-

System property	Description
	sources. Further information on ECU Resources is available in the EB tresos Studio developer's guide, chapter Configuration models/ECU Resource Manager API .
-Dderivate=<derivate>	Defines the target's derivate, e.g. -Dderivate=Win32x86. This is necessary when working with module configurations, which use ECU Resources. Further information on ECU Resources is available in the EB tresos Studio developer's guide, chapter Configuration models/ECU Resource Manager API . If the ECU Resources require additional parameters, such as subderivate , you need to specify any additional parameter with -D<parameter-name>=<parameter-value>. Information on how to specify these parameters is available in the EB tresos Studio developer's guide, chapter ECU Resource Manager API/Characteristics when using the legacy commandline/Defining parameter values for the DefaultEcuResourceFinder .
-DEcuResourceModuleIds=<Module1;Module2;...>	Defines a comma- or semicolon-separated list of module-IDs, which are necessary for finding relevant ECU Resources, e.g. -DEcuResourceModuleIds=Com_TS_T99D01M01I05R01;Rte_TS_T99D01M01I04R02. Information on how to define configured modules for the ECU Resource Finder is available in the EB tresos Studio developer's guide, chapter ECU Resource Manager API/Characteristics when using the legacy commandline/Defining configured modules for the DefaultEcuResourceFinder .

7.11.3.2.3. Options

The following options are supported by the legacy generate command:

Parameter	Description
-a	<p>Sets that all configurations are verified of all modules before generating code, even if code is not generated for all modules.</p> <p>The default is that only the module configurations for which code is being generated are verified.</p> <p>To verify all module configurations, even if no code is generated for them, use this option.</p> <p>To verify only the configurations of the modules specified before generating code, use the -g option. This option is explained below.</p>

Parameter	Description
<code>-c <config-path></code>	<p>Sets the directory in which the module configurations are searched as <code><config-path></code>.</p> <p>By default, the directory <code><base.path>\bin</code> is used, with <code>base.path</code> being the value of the system property <code>-Dbase.path</code>.</p> <p>For details about the system property <code>-Dbase.path</code>, refer to Section 7.11.1.3, "System properties common to all commands". If the system property is not set, an error is displayed.</p>
<code>-d<var>=<value></code>	<p>Defines a variable called <code><var></code> with value <code><value></code>, which is then available to the code generators.</p> <p>If a code generator for a module supports this feature, please refer to this generator's documentation for a list of the available variables and allowed values.</p>
<code>-g <module-id></code>	<p>By default, the code for all specified module configurations is generated. To override that default setting and generate code only for a subset of these modules, use the <code>-g</code> option.</p> <p>You may use this option multiple times, if you want to generate code for more than one module.</p> <p>Replace the argument <code><module-id></code> with the ID of the module you want to generate code for, e.g. <code>Com_TS_T16D4M2I0R0</code>.</p> <p>By default, only the module configurations for which code is being generated are verified.</p> <p>To verify all module configurations, even if no code is generated for them, use the <code>-a</code> option.</p> <div style="display: flex; align-items: center;"> NOTE  You can only specify module-IDs of modules that are installed in the copy of EB tresos Studio you are running. </div>
<code>-n <config-name></code>	<p>Specifies the name of the module configurations of the configuration files to load.</p> <p>If you use this option, other module configurations contained in the specified files, are ignored. You may use this option multiple times.</p>
<code>-o <output-path></code>	<p>Sets the directory to which the generated files are written. The files are written to the directory specified as <code><output-path></code></p>

Parameter	Description
	<p>By default, the directory <base.path>\output is used, with base.path being the value of the system property -Dbase.path.</p> <p>For details about the -Dbase.path system property, refer to Section 7.11.1.3, “System properties common to all commands”. If the system property is not set, an error is displayed.</p>
-u	Specifies that the generated files end with Unix line endings.

7.11.3.2.4. Parameters

In addition to the common parameters <system_property> and <workspace>, the legacy generate command also recognizes the following parameters:

Parameter	Description
<config-file>[@<type>]	<p>Use <config-file> to specify the name of the file that contains the module configuration(s) you want to load.</p> <p>The parameter can be passed multiple times to load more than one file. You may use the optional argument @<type> to explicitly specify the content type of the file.</p> <p>For information which content types are available, refer to Section 4.5.3, “File content types”.</p>

7.11.3.2.5. Example

The following code is an example of how to use the command legacy generate :

```
tresos_cmd.bat -DValidate=true legacy generate -n Com -g
    Com_TS_T16D4M2I0R0 -g Custom -u All.epc@asc:4.0.3
```

This example loads a module configuration called *Com* from the file *All.epc*. If *All.epc* contains other module configurations as well, they are ignored.

The *.epc* file is treated as AUTOSAR 4.0.3 file.

The module definition for the configuration is either contained in the module *Com_TS_T16D4M2I0R0*.

The configuration is validated against its definition. The code is generated for the two modules *Com_TS_T16D4M2I0R0* and *Custom* (provided that the module *Custom* does not need a configuration). The generated files contain Unix line feeds.

7.11.3.3. Verifying configurations

Use the `legacy verify` command to verify module configurations and to print out all infos, warnings, and errors. This command executes the same generation steps as the `legacy generate` command with the only difference that no files are written to disc.

7.11.3.3.1. Syntax

```
tresos_cmd.bat [<system_property>...] [-data <workspace>]
    legacy verify [<options>]
    <config-file>[@<type>]...
```

7.11.3.3.2. System properties

In addition to the common system properties listed in [Section 7.11.1.3, “System properties common to all commands”](#), the `legacy verify` command also recognizes the following system properties:

System property	Description
<code>-DmergeConfigs=(true false)</code>	Merges the parameters of the module configurations. This is useful, if you have split the parameters of a module configuration into several files (e.g. one file containing standard- and one containing vendor-specific parameters). The default is <code>false</code> for this command. To merge the parameters of the module configurations, set to <code>true</code> . To not merge the parameters but instead to overwrite the previous content with the content of the file read last, set to <code>false</code> .
<code>-DverifyReadOnly=(true false)</code>	Enables/disables the verification of read-only configuration parameters. The default is <code>false</code> for this command. To enable the verification of read-only configuration parameters, set to <code>true</code> . To disable the verification of read-only configuration parameters, set to <code>false</code> .
<code>-DsuppressReadOnlyWarnings= (true false)</code>	Suppresses warnings about read-only configuration parameters. Warnings are not logged or printed out. The default is <code>false</code> for this command.

System property	Description
	<p>To suppress warnings, set to <code>true</code>.</p> <p>To keep warnings logged or print them out, set to <code>false</code>.</p>
<code>-DpredefinedVariant=<Name></code>	<p>This property enables the variant resolving and specifies the predefined Variant name that will be taken from the input files.</p> <p>If the parameter is not set, by default the variants resolving is disabled.</p> <p>For detailed information about variant handling please refer to Section 7.12, “Variant Handling”.</p>
<code>-Dttarget=<target></code>	<p>Defines the target architecture, e.g. <code>-Dttarget=WINDOWS</code>. This is necessary when working with module configurations, which use ECU Resources. Information on ECU Resources is available in the EB tresos Studio developer's guide, chapter Configuration models/ECU Resource Manager API.</p>
<code>-Dderivate=<derivate></code>	<p>Defines the target's derivate, e.g. <code>-Dderivate=Win32x86</code>. This is necessary when working with module configurations, which use ECU Resources. Information on the ECU Resources Manager API is available in the EB tresos Studio developer's guide, chapter Configuration models/ECU Resource Manager API. If the ECU Resources require additional parameters, such as <code>subderivate</code>, you need to specify any additional parameter with <code>-D<parameter-name>=<parameter-value></code>. For information on how to specify these parameters, refer to the EB tresos Studio developer's guide, chapter Configuration models/ECU Resource Manager API/Characteristics when using the legacy commandline/Defining parameter values for the DefaultEcuResourceFinder.</p>
<code>-DEcuResourceModuleIds=<Module1;Module2;...></code>	<p>Defines a comma- or semicolon-separated list of module-IDs, which are necessary for finding relevant ECU Resources, e.g. <code>-DEcuResourceModuleIds=Com_TS_T99D01M01I05R01;Rte_TS_T99D01M01I04R02</code>. For information on how to define the configured modules for the ECU Resource Finder, refer to the EB tresos Studio developer's guide, chapter Configuration models/ECU Resource Manager API/Characteristics when using the legacy commandline/Defining configured modules for the DefaultEcuResourceFinder.</p>

7.11.3.3.3. Options

The following options are supported by the `legacy verify` command:

Parameter	Description
<code>-a</code>	<p>Sets that all configurations of all modules are verified before code is generated, even if code is not generated for all modules.</p> <p>The default is that only the module configurations, for which code is being generated, are verified.</p> <p>To verify all module configurations, even if no code is generated for them, use this option.</p> <p>To verify only the configurations of the modules specified before generating code, use the <code>-g</code> option. This option is explained below.</p>
<code>-c <config-path></code>	<p>Searches for the module configuration files in the directory you specified as <code><config-path></code>.</p> <p>By default, the files are searched in the directory <code><base.path>\bin</code>, with <code>base.path</code> being the value of the system property <code>-Dbase.path</code>.</p> <p>For details about the system property <code>-Dbase.path</code>, refer to Section 7.11.1.3, "System properties common to all commands". If the system property is not set, an error is displayed.</p>
<code>-d<var>=<value></code>	<p>Defines a variable named <code><var></code> with value <code><value></code>, which is then available to the code generators.</p> <p>If a code generator for a module supports this feature, please refer to this generator's documentation for a list of the available variables and allowed values.</p>
<code>-g <module-id></code>	<p>By default, the code for all specified module configurations is generated. To override that default setting and generate code only for a subset of these modules, use the <code>-g</code> option.</p> <p>You may use this option multiple times, if you want to generate code for more than one module.</p> <p>Replace the argument <code><module-id></code> with the ID of the module you want to generate code for, e.g. <code>Com_TS_T16D4M2IOR0</code>.</p> <p>By default, only the module configurations for which code is being generated are verified.</p>

Parameter	Description
	<p>To verify all module configurations, even if no code is generated for them, use the <code>-a</code> option.</p> <p>NOTE  You can only specify module-IDs of modules that are installed in the copy of EB tresos Studio you are running.</p>
<code>-n <config-name></code>	<p>Specifies the name of the module configurations of the configuration files to load.</p> <p>If you use this option, other module configurations contained in the specified files, are ignored. You may use this option multiple times.</p>

7.11.3.3.4. Parameters

In addition to the common parameters `<system_property>` and `<workspace>`, this command does also recognize the following parameters:

Parameter	Description
<code><config-file>[@<type>]</code>	<p>Use <code><config-file></code> to specify the name of the file that contains the module configuration(s) you want to load.</p> <p>The parameter can be passed multiple times to load more than one file. You may use the optional argument <code>@<type></code> to explicitly specify the content type of the file.</p> <p>For information which content types are available, refer to Section 4.5.3, “File content types”.</p>

7.11.3.3.5. Example

The following code is an example of how to use the command `legacy verify`:

```
tresos_cmd.bat -DValidate=true legacy verify -n Com -g
Com_TS_T16D4M2I0R0 -g Custom -u All.epc@asc:4.0.3
```

This example loads a module configuration called `Com` from the file `All.epc`. If `All.epc` contains other module configurations as well, they are ignored.

The `.epc` file is treated as AUTOSAR 4.0.3 file.

The module definition for the configuration may be contained in the module `Com_T16D4M2I0R0`. The configuration is validated against its definition.

7.11.3.4. Listing all generation modes

It is possible that a module provides a custom-built generator mode other than **generate** and **verify**.

The name of these additional and module-specific generator modes and their functionality depend upon the individual module.

- ▶ To find out if a module provides any additional modes, please refer to the documentation of the modules you are running with EB tresos Studio.
- ▶ To find out which generation modes the modules you use have, use the command **legacy listmodes**.
- ▶ To run such modes, call the generator mode from the commandline. For details, refer to [Section 7.11.3.5, “Running generation modes”](#).

7.11.3.4.1. Syntax

To list all available legacy generation modes for generators of all modules in legacy mode, use the following syntax:

```
tresos_cmd.bat [<system_property>...] [-data <workspace>]  
    legacy listmodes
```

7.11.3.4.2. System properties

The **legacy listmodes** command only recognizes the system properties common to all commands. For a list of these system properties, refer to [Section 7.11.1.3, “System properties common to all commands”](#).

7.11.3.4.3. Example

The following is an example of how to use the **legacy listmodes** command:

```
tresos_cmd.bat legacy listmodes
```

If you run this command you will see all module-IDs and their available generation modes.

7.11.3.5. Running generation modes

To run generator modes in legacy mode other than **generate** and **verify**, use the **legacy make** command.

The name of these additional and module-specific generator modes and their functionality depend upon the individual module. To list all available generation modes, refer to [Section 7.11.3.4, “Listing all generation modes”](#).

7.11.3.5.1. Syntax

If a module supports the legacy mode, you can run its custom-built generator on a configuration with the **legacy make** command:

```
tresos_cmd.bat [<system_property>...] [-data <workspace>]
    legacy make <mode> [<options>]
    <config-file>[@<type>] ...
```

7.11.3.5.2. System properties

In addition to the common system properties ([Section 7.11.1.3, “System properties common to all commands”](#)), the **legacy make** command also recognizes the following system properties:

System property	Description
-Dttarget=<target>	Defines the target architecture, e.g. -Dttarget=WINDOWS. This is necessary when working with module configurations, which use ECU Resources. Information on the ECU Resource Manager API is available in the EB tresos Studio developer's guide, chapter Configuration models/ECU Resource Manager API .
-Dderivate=<derivate>	Defines the target's derivate, e.g. -Dderivate=Win32x86. This is necessary when working with module configurations, which use ECU Resources. Information on the ECU Resource Manager API is available in the EB tresos Studio developer's guide, chapter Configuration models/ECU Resource Manager API . If the ECU Resources require additional parameters, such as subderivate , you need to specify any additional parameter with -D<parameter-name>=<parameter-value>. Information on how to specify these parameters is available in the EB tresos Studio developer's guide, chapter Configuration models/ECU Resource Manager API/Characteristics when using the legacy command-

System property	Description
	line/Defining parameter values for the DefaultEcuResourceFinder... .
-DEcuResourceModuleIds=<Module1;Module2;...>	Defines a comma- or semicolon-separated list of module-IDs, which are necessary for finding relevant ECU Resources, e.g. -DEcuResourceModuleIds=Com_TS_T99D01M01I05R01;Rte_TS_T99D01M01I04R02. Information on how to define these configured modules for the ECU Resource Finder is available in the EB tresos Studio developer's guide, chapter Configuration models/ECU Resource Manager API/Characteristics when using the legacy commandline/Defining configured modules for the DefaultEcuResourceFinder.
-DpredefinedVariant=<Name>	<p>This property enables the variant resolving and specifies the predefined Variant name that will be taken from the input files.</p> <p>If the parameter is not set, by default the variants resolving is disabled.</p> <p>For detailed information about variant handling please refer to Section 7.12, “Variant Handling”).</p>

7.11.3.5.3. Options

The following options are supported by the **legacy make** command:

Parameter	Description
-a	<p>Sets that all configurations are verified of all modules before generating code, even if code is not generated for all modules.</p> <p>The default is that only the module configurations for which code is being generated are verified.</p> <p>To verify all module configurations, even if no code is generated for them, use this option.</p> <p>To verify only the configurations of the modules specified before generating code, use the -g option. This option is explained below.</p>
-c <config-path>	<p>Sets the directory in which the module configurations are searched as <config-path> .</p> <p>By default, the directory <base.path>\bin is used, with base.path being the value of the system property -Dbase.path .</p>

Parameter	Description
	<p>For details about the system property <code>-Dbase.path</code>, refer to Section 7.11.1.3, “System properties common to all commands”. If the system property is not set, an error is displayed.</p>
<code>-d<var>=<value></code>	<p>Defines a variable called <code><var></code> with value <code><value></code>, which is then available to the code generators.</p> <p>If a code generator for a module supports this feature, please refer to this generator's documentation for a list of the available variables and allowed values.</p>
<code>-g <module-id></code>	<p>By default, the code for all specified module configurations is generated. To override that default setting and generate code only for a subset of these modules, use the <code>-g</code> option.</p> <p>You may use this option multiple times, if you want to generate code for more than one module.</p> <p>Replace the argument <code><module-id></code> with the ID of the module you want to generate code for, e.g. <code>Com_TS_T16D4M2IOR0</code>.</p> <p>By default, only the module configurations for which code is being generated are verified.</p> <p>To verify all module configurations, even if no code is generated for them, use the <code>-a</code> option.</p>
	<p>NOTE You can only specify module-IDs of modules that are installed in the copy of EB tresos Studio you are running.</p>
<code>-n <config-name></code>	<p>Specifies the name of the module configurations of the configuration files to load.</p> <p>If you use this option, other module configurations contained in the specified files, are ignored. You may use this option multiple times.</p>
<code>-o <output-path></code>	<p>Sets the directory to which the generated files are written. The files are written to the directory specified as <code><output-path></code>.</p> <p>By default, the directory <code><base.path>\output</code> is used, with <code>base.path</code> being the value of the system property <code>-Dbase.path</code>.</p>

Parameter	Description
	For details about the <code>-Dbase.path</code> system property, refer to Section 7.11.1.3, "System properties common to all commands" . If the system property is not set, an error is displayed.
<code>-u</code>	Specifies that the generated files end with Unix line endings.

7.11.3.5.4. Parameters

In addition to the common parameters `<system_property>` and `<workspace>`, the **legacy make** command also recognizes the following parameters:

Parameter	Description
<code><config-file>[@-<type>]</code>	<p>Use <code><config-file></code> for the name of the file that contains the module configuration(s) that you want to load.</p> <p>You may pass the parameter multiple times to load more than one file.</p> <p>Use the optional argument <code>@<type></code> to explicitly specify the content type of the file. For information, which content types are available, refer to Section 4.5.3, "File content types".</p>

7.11.3.5.5. Example

The following code is an example of how to use the **legacy make** command:

```
tresos_cmd.bat -DmergeConfigs=true legacy make generate_SWC-T
    myEpc.epc myOther.epc
```

This example loads the content of the two files `myEpc.epc` and `myOther.epc`.

The module-configurations contained in `myEpc.epc` are merged with the ones in `myOther.epc`, but not overwritten.

A custom-built generator called `generate_SWC-T` is invoked. The above example assumes that the modules of the configurations contained in the loaded files support this generation mode.

7.11.4. Commands for module development

The commands in this chapter cover printing out data models or creating and checking vendor-specific module definitions, amongst others. The commands are useful for debugging purposes and module development.

NOTE**This chapter is for module developers only**

The commands in this chapter are useful for debugging and module developing only. If you are using the GUI, you may not want to read this chapter.

The following topics are available:

- ▶ [Section 7.11.4.1, “Listing a data model”](#)
- ▶ [Section 7.11.4.2, “Creating vendor-specific module definitions \(VSMD\)”](#)
- ▶ [Section 7.11.4.3, “Checking vendor-specific against standard module definitions”](#)
- ▶ [Section 7.11.4.4, “Listing all rules used for VSMD checking”](#)
- ▶ [Section 7.11.4.5, “Encrypting and signing modules”](#)

7.11.4.1. Listing a data model

Use the `legacy tree` command to print out the data model loaded from modules and module configurations.

7.11.4.1.1. Syntax

```
tresos_cmd.bat [<system_property>...] [-data <workspace>]
    legacy tree [<options>] <config-file>[@<type>]...
```

7.11.4.1.2. System properties

In addition to the common system properties (see [Section 7.11.1.3, “System properties common to all commands”](#)), this command also recognizes the following system properties:

System property	Description
<code>-DmergeConfigs=(true false)</code>	Merges the parameters of the module configurations. This is useful, if you have split the parameters of a module configuration into several files (e.g. one file containing standard- and one containing vendor-specific parameters). The default is <code>false</code> for this command.

System property	Description
	To merge the parameters of the module configurations, set the property to <code>true</code> . To not merge the parameters, set the property to <code>false</code> . The content of the file read last overwrites the previous content.

7.11.4.1.3. Options

The following options are supported by this command:

Parameter	Description
<code>-c <config-path></code>	Use this option to set the directory in which module configuration files are searched to <code><config-path></code> . The default directory for module configuration files is <code><base.path>\bin</code> , with <code>base.path</code> being the value of the system property <code>-Dbase.path</code> (for details on <code>-Dbase.path</code> , refer to Section 7.11.1.3, “System properties common to all commands”). If the system property is not set, an error is displayed.
<code>-n <config-name></code>	Use this option to specify the name of the module configurations of the configuration files you wish to load. If you use this option, other module configurations, which are contained in the files you specified, are ignored. You can use this option multiple times.

7.11.4.1.4. Parameters

In addition to the common parameters `<system_property>` and `<workspace>`, this command also recognizes the following parameters:

Parameter	Description
<code><config-file>[@<type>]</code>	Use <code><config-file></code> to specify the name of the file, which contains the module configuration(s) you wish to load. You may pass this parameter multiple times to load more than one file. Use the optional argument <code>@<type></code> to explicitly specify the content type of the file. For information, which content types are available, refer to Section 4.5.3, “File content types” .

7.11.4.1.5. Example

The following code is an example of how to use the command `legacy tree`:

```
tresos_cmd.bat legacy tree -n Com All.epc@asc:4.0.3
```

This example loads a module configuration named *Com* from the file *All.epc* into the data model and prints the model as a tree. If this file contains other module configurations as well, they are ignored. The *.epc* file is treated as AUTOSAR 4.0.3 file.

7.11.4.2. Creating vendor-specific module definitions (VSMD)

According to the AUTOSAR specification Specification of ECU Configuration, a vendor has to derive a vendor-specific module definition (VSMD) from the corresponding AUTOSAR standard module definition (StMD). You can use the `vsmd` command to let EB tresos Studio derive a basic vendor-specific module definition for you.

TIP

Since it is possible to specify different content and file types for StMD and VSMD, you may also use the `vsmd` command to convert files from one format to the other.

For further information on content types, refer to [Section 4.5.3, “File content types”](#).

To derive a VSMD from a StMD, several rules apply. These rules are part of the AUTOSAR specification Specification of ECU Configuration. The most important one requires you to choose your own package name in the module definition file under which your module definition is located.

If you use the `vsmd` command, it creates a reference to the StMD using the VSMD with the `REFINED-MODULE.DEF` tag. In addition, new UUIDs are created for each element of the VSMD.

Additionally, a default `ADMIN-DATA` section is created for each contained `MODULE-DEF`.

7.11.4.2.1. Syntax

Use the `vsmd` command to create a basic vendor-specific module definition from a standard module definition as shown in the following example code:

```
tresos_cmd.bat [<system_property>...] [-data <workspace>]
    legacy vsmd [<options>] <stmd-file>[@<type>]...
    <vsmd-file>[@<type>] <package> [<origin>]
```

7.11.4.2.2. System properties

In addition to the common system properties (see [Section 7.11.1.3, “System properties common to all commands”](#)), this command also recognizes the following system properties:

System property	Description
-DmergeConfigs= (true false)	<p>Merges the parameters of the module configurations. This is useful, if you have split the parameters of a module configuration into several files, e.g. one file that contains standard- and one that contains vendor-specific parameters.</p> <p>The default is <code>true</code> for this command.</p> <p>To merge the parameters of the module configurations, set the property to <code>true</code>.</p> <p>To not merge the parameters, set the property to <code>false</code>. The content of the file read last overwrites the content.</p>
-DwriteDefaults= (true false)	<p>Deprecated property.</p> <p>Forces the LOWER-MULTIPLICITY and UPPER-MULTIPLICITY tags to be printed to the output-file, even if their value is 1.</p> <p>The default value for this parameter is <code>false</code> and thus the tags are usually omitted.</p> <p>This property was used in previous versions to achieve conformance to XSD schema files. This property has been replaced by content types. For information about content types, refer to Section 4.5.3, “File content types”.</p>
-Duuids=(true false)	<p>Creates new random unique IDs for each node. The default is <code>true</code>.</p> <p>To create new random unique IDs for each node, set to <code>true</code>.</p>
-DecuParamDef= (true false)	<p>Generates a ECU-PARAMETER-DEFINITION tag into the output file. The default is <code>true</code>.</p> <p>To generate a ECU-PARAMETER-DEFINITION tag into the output file, set to <code>true</code>.</p> <p>To not generate a ECU-PARAMETER-DEFINITION tag into the output file, set to <code>false</code>.</p>

System property	Description
-DecuParamDefName= <name>	Changes the name of the ECU-PARAMETER-DEFINITION tag in the output file. The default value is myEcuParameterDefinition.
-DbswModuleDesc= (true false)	<p>Generates a BSW-MODULE-DESCRIPTION tag into the output file. The default is true.</p> <p>To generate a BSW-MODULE-DESCRIPTION tag into the output file, set to true.</p> <p>To not generate a BSW-MODULE-DESCRIPTION tag into the output file, set to false.</p>
-DbswModuleDescName= <name>	Changes the name of the BSW-MODULE-DESCRIPTION tag in the output file. The default value is myBswModuleDescription.
-DvsmdPath= <as-path>	<p>The <as-path> argument explicitly sets the value of the VEN-DOR-SPECIFIC-MODULE-DEF-REF parameter inside the BSW-MODULE-DESCRIPTION container of the output file. By default, this path is determined automatically.</p> <p>Precondition: To use -DvsmdPath=<as-path>, set the property -DbswModuleDesc=true.</p>
-DrecommendedConfigPath= <as-path>	<p>The <as-path> argument explicitly sets the value of the RECOMMENDED-CONFIGURATION-REF parameter inside the BSW-MODULE-DESCRIPTION container of the output file. If you do not set this property, no recommended configuration is loaded.</p> <p>Precondition: To use -DrecommendedConfigPath=<as-path>, set the property -DbswModuleDesc=true.</p>
-DpreConfigPath= <as-path>	<p>The <as-path> argument explicitly sets the value of the PRECONFIGURED-CONFIGURATION-REF parameter inside the BSW-MODULE-DESCRIPTION container of the output file. If you do not set this property, no preconfiguration is loaded.</p> <p>Precondition: To use -DpreConfigPath=<as-path>, set the property -DbswModuleDesc=true.</p>
-DssuppressReadOnlyWarnings= (true false)	<p>Suppresses warnings regarding read-only configuration parameters; does not log them or print them out. The default is false for this command.</p> <p>To suppress warnings, not log them or print them out, set to true.</p> <p>To keep warnings, to log them or print them out, set to false.</p>

System property	Description
-DMapOptionalAsList=(true false)	<p>Creates a surrounding list for elements with LOWER-MULTIPLICITY=0 and UPPER-MULTIPLICITY=1. For backward-compatibility reasons, the default value is true</p> <p>Precondition: To use -DMapOptionalAsList=(true false), the content type of the output file must be XDM.</p> <p>To create a surrounding list for elements with LOWER-MULTIPLICITY=0 and UPPER-MULTIPLICITY=1, set to true.</p> <p>To not create a surrounding list, set to false. Instead, the affected elements are marked as optional.</p>

System property	Description
	<p>NOTE</p>  <p>Leaving this property at the default, or explicitly setting it to <code>true</code> can lead to problems!</p> <p>If you leave this property at the default value or explicitly set it to <code>true</code>, the XPaths of the affected elements depend on their LOWER-MULTIPLICITY value.</p> <p>If the LOWER-MULTIPLICITY is 0, a list is created. If the LOWER-MULTIPLICITY is later changed to 1, no list is created (with the UPPER-MULTIPLICITY always being at 1). The XPath is different in both cases.</p> <p>Code templates, which then wish to address this parameter in the configuration need to be able to handle both cases.</p> <p>However, if you set this system property to <code>false</code>, the XPath stays the same in any case.</p>
	<p>TIP</p>  <p>If you are writing a code-template, you can use the following XPath function in the template to determine if the parameter is configured or not:</p> <pre>node:exists(OptionalVariable).</pre>

7.11.4.2.3. Options

The following options are supported by this command:

Parameter	Description
-c <config-path>	<p>Use this option to perform the search for module configuration files in the directory you specify as <config-path>.</p> <p>By default, the directory <base.path>\bin is used for the search, with <code>base.path</code> being the value of the system property <code>-Dbase.path</code> (see Section 7.11.1.3, “System properties common to all commands”). If the system property is not set, an error is displayed.</p>
-o <output-path>	<p>Use this option to write the generated files to the directory specified as <output-path>.</p>

Parameter	Description
	<p>By default, the directory <base.path>\output is used, with base.path being the value of the system property -Dbase.path. For details about this system property, refer to Section 7.11.1.3, “System properties common to all commands”.</p> <p>If the system property is not set, an error is displayed.</p>
-u	Use this option, to generate all files with Unix line endings.
-n <config-name>	<p>Specifies the name of the module configurations of the configuration files to load.</p> <p>If you use this option, other module configurations contained in the specified files, are ignored. You may use this option multiple times.</p>

7.11.4.2.4. Parameters

In addition to the common parameters <system_property> and <workspace>, this command also recognizes the following parameters:

Parameter	Description
<stmd-file>[@<type>]	<p>Defines the file name that contains the standardized module definition (StMD) from which to create the vendor-specific module definition (VSMD). If the file contains more than one module definition, all files are loaded.</p> <p>Use the optional argument @<type> to explicitly specify the content type of the file. To find out which content types are available, refer to Section 4.5.3, “File content types”.</p>
<vsmd-file>[@<type>]	<p>Defines the name of the file to which the created VSMD is written. Use the optional argument @<type> to explicitly specify the content type of the file. To find out which content types are available, refer to Section 4.5.3, “File content types”.</p> <p>If you do not specify any content type, the file extension is interpreted. Possible extensions are: *.xdm, *.arxml, *.asc, *.epd, *.bmd, *.epc.</p>
<package>	Defines the name of the AUTOSAR package under which your module definition is located.
<origin>	Defines the origin of the parameters. This parameter is only used if the content type of the output file is XDM. If that is so, each parameter in

Parameter	Description
	the XDM file gets an attribute named __ORIGIN with the text that you specified here as value.

7.11.4.2.5. Example

The following code is an example of how to use the command legacy vsmd:

```
tresos_cmd.bat -DMapOptionalAsList=false legacy vsmd
    Com.epd MyCom.xdm TS_T16D4M2I0R0
```

This example creates a VSMD file with the name `MyCom.xdm` from the StMD `Com.epd`. The created VSMD contains the AR-PACKAGE `TS_T16D4M2I0R0`. Optional elements are not surrounded by lists in the output file, which is the recommended way of doing this.

7.11.4.3. Checking vendor-specific against standard module definitions

There are some rules, you must follow when you derive a vendor-specific module definition (VSMD) from a standardized module definition (StMD). Please refer to the AUTOSAR specification Specification of ECU Configuration for more information about these rules.

7.11.4.3.1. Syntax

To check if a VSMD conforms to a StMD, use the `legacy vsmdcheck` command as shown in the following code example:

```
tresos_cmd.bat [<system_property>...] [-data <workspace>]
    legacy vsmdcheck
    <stmd-file>[@<type>] <vsmd-file>[<type>]...
    <rule|ruleset>[,<rule|ruleset>...]
    [-ignore <rule>]... [-xml <xmlfile>]
```

7.11.4.3.2. System properties

In addition to the common system properties listed in [Section 7.11.1.3, “System properties common to all commands”](#), this command also recognizes the following system properties:

System property	Description
-DssuppressReadOnlyWarnings= (true false)	<p>Suppresses warnings about read-only parameters; the warnings are not logged or printed out. The default is <code>false</code> for this command.</p> <p>To suppress warnings, set to <code>true</code>.</p> <p>To log or print out read-only parameters, set to <code>false</code>.</p>
-Dttarget= <target>	<p>Defines the target architecture, e.g. <code>-Dttarget=WINDOWS</code>. This is necessary when working with module configurations, which use ECU Resources. Information on the ECU Resources Manager API is available in the EB tresos Studio developer's guide, chapter Configuration models/ECU Resource Manager API.</p>
-Dderivate= <derivate>	<p>Defines the target's derivate, e.g. <code>-Dderivate=Win32x86</code>. This is necessary when working with module configurations which use ECU Resources. Information on ECU Resources is available in the EB tresos Studio developer's guide, chapter Configuration models/ECU Resource Manager API. If the ECU Resources require additional parameters such as <code>subderivate</code> then you need to specify any additional parameter with <code>-D<parameter-name>=<parameter-value></code>. For information on how to specify these parameters, refer to the EB tresos Studio developer's guide, chapter Configuration models/ECU Resource Manager API/Characteristics when using the legacy commandline/Defining parameter values for the <code>DefaultEcuResourceFinder</code>.</p>
-DEcuResourceModuleIds= <Module1;Module2;...>	<p>Defines a comma- or semicolon-separated list of module-IDs that are necessary for finding relevant ECU Resources, e.g. <code>-DEcuResourceModuleIds=Com_TS_T99D01M01I05R01;Rte_TS_T99D01M01I04R02</code>. For information on how to define configured modules for the ECU Resource Finder, refer to the EB tresos Studio developer's guide, chapter Configuration models/ECU Resource Manager API/Characteristics when using the legacy commandline/Defining configured modules for the <code>DefaultEcuResourceFinder</code>.</p>

7.11.4.3.3. Parameters

In addition to the common parameters `<system_property>` and `<workspace>`, this command also recognizes the following parameters:

Parameter	Description
<stmd-file>[@<type>]	Defines the file name that contains the standardized module definition (StMD). If the file contains more than one module definitions, all definitions are loaded. Use the optional argument @<type> to explicitly specify the content type of the file. For information, which content types are available, refer to Section 4.5.3, “File content types” .
<vsmd-file>[@<type>]	Defines the file name that contains the vendor-specific module definition (VSMD). If the file contains more than one module definitions, all definitions are loaded. Use the optional argument @<type> to explicitly specify the content type of the file. For information, which content types are available, refer to Section 4.5.3, “File content types” .
<rule ruleset>	Defines the rule or rule-set to be used. For a list of available rules, refer to the legacy listrules command in Section 7.11.4.4, “Listing all rules used for VSMD checking” . Available rule-sets are: asc:2.0, asc:2.1, asc:3.0, asc:3.1, asc:3.2, asc:4.0.2 and asc:4.-0.3.
-ignore <rule>	Defines the name of a rule which shall not be checked. For a list of available rules, refer to the legacy listrules command in Section 7.11.4.4, “Listing all rules used for VSMD checking” .
-xml <xmlfile>	Defines the name of a file into which the result shall be saved in XML format.

7.11.4.3.4. Example

The following code is an example of how to use the command legacy vsmdcheck:

```
tresos_cmd.bat -DValidate=true legacy vsmdcheck StMD.bmd@asc:4.0.3
                VSMD.epd@asc:4.0.3 asc:4.0.3 -ignore EcucSws_1035
```

This example validates the VSMD file with the name VSMD.epd against the StMD file with the name StMD.bmd. Both files are regarded as AUTOSAR 4.0.3 files. The rule-set used for the checks is asc:4.0.3, but the rule EcucSws_1035 is not checked.

7.11.4.4. Listing all rules used for VSMD checking

7.11.4.4.1. Syntax

To list all rules that are used for the legacy `vsmdcheck` command, use the `legacy listrules` command:

```
tresos_cmd.bat [<system_property>...] [-data <workspace>] legacy listrules
```

7.11.4.4.2. System properties

This command only recognizes the system properties common to all commands. For a list of these system properties, refer to [Section 7.11.1.3, “System properties common to all commands”](#).

7.11.4.4.3. Example

The following example lists all rules used for VSMD checking with the `legacy listrules` command:

```
tresos_cmd.bat legacy listrules
```

7.11.4.5. Encrypting and signing modules

If you are developing your own module for use in EB tresos Studio as an EB partner, you may use the `crypto` command to encrypt and sign modules. For more information about this command, refer to the EB tresos Studio developer's guide, chapter [Developing modules/How to sign modules with the crypto command](#).

7.11.5. Commands for converting and merging files

This section shows you how to convert files to different formats and how to merge the content of files.

7.11.5.1. Merging files and converting between different file formats

To merge and convert files, use the `convert` command. This command reads one or more files with any content type known to EB tresos Studio. The command merges the content of the read files and writes all read data to an output file with one of the following file formats:

- ▶ XDM
- ▶ AUTOSAR module configuration
- ▶ AUTOSAR system description

There are two different types of data that can be written to the output file: module configuration data and system description data. The input files must contain the same type of data that is supposed to be written to the output

file. If it does not contain the same type of data, EB tresos Studio does not necessarily report an error, when you try to convert between the different types of data. However, the output file will probably be almost empty. For information, which file formats contain which kind of data, refer to [Section 4.5.3, “File content types”](#).

7.11.5.1.1. Syntax

To use the command `convert`, use the following syntax:

```
tresos_cmd.bat [<system_property>...] [-data <workspace>]
  legacy convert [<options>]
  <input-file>[@<type>]... <output-file>[@<type>]
```

7.11.5.1.2. System properties

The command `convert` recognizes the common system properties listed in [Section 7.11.1.3, “System properties common to all commands”](#). In addition to the common system properties, the `convert` command also recognizes additional system properties, depending on the kind of data to read and write.

Additional system properties when converting module configuration data:

System property	Description
<code>-DmergeConfigs=(true false)</code>	Merges parameters of module configurations. Merging parameters is useful, if you have split the parameters of a module configuration into several files, e. g. if one file contains standard and another contains vendor-specific parameters. The default is <code>true</code> for this command. To merge the parameters of the module configurations, set the property to <code>true</code> . To not merge parameters, set this property to <code>false</code> . The content of the file read last overwrites the previous content.
<code>-DwriteDefaults=(true false)</code>	Deprecated property. Use it to force the <code>LOWER-MULTIPLICITY</code> and <code>UPPER-MULTIPLICITY</code> tags to be printed to the output file, even if their value is <code>1</code> . The default value for this parameter is <code>false</code> and thus the tags are usually omitted. This property was used in previous versions of EB tresos Studio to achieve conformance to XSD schema files. These schema files are now replaced by content types. For details about content files, refer to Section 4.5.3, “File content types” .
<code>-Duuids=(true false)</code>	To create new random unique IDs for each node, set this property to <code>true</code> . The default value is <code>true</code> .

System property	Description
-DssuppressReadOnlyWarnings=(true false)	<p>To not log or print out warnings regarding read-only configuration parameters, set to <code>true</code>. The default value for this command is <code>false</code>, which logs and prints out warnings about read-only configuration parameters.</p>
-DMapOptionalAsList=(true false)	<p>This parameter can only be used, if the content type of the output file is XDM.</p> <ul style="list-style-type: none"> ▶ Set to <code>true</code> to create a surrounding list for elements with LOWER-MULTIPLICITY= 0 and UPPER-MULTIPLICITY=1. ▶ Set to <code>false</code> to not create surrounding lists. Instead, the affected elements are marked as optional. <p>For backward-compatibility reasons, the default value is <code>true</code>.</p>
	<p>NOTE</p>  <p>Leaving this property at the default, or explicitly setting it to true can lead to problems!</p> <p>If you leave this property at the default or explicitly set it to <code>true</code>, the XPaths of the affected elements depend on their LOWER-MULTIPLICITY value. If the LOWER-MULTIPLICITY is 0, a list is created. If the LOWER-MULTIPLICITY is later changed to 1, no list is created (with the UPPER-MULTIPLICITY always being at 1). In both cases, the XPath is different. Code templates, which then try to address this parameter in the configuration, need to be able to handle both cases. However, if you set this system property to <code>false</code>, the XPath always stays the same.</p> <hr/> <p>TIP</p>  <p>If you are writing a code template, you may use the following XPath function in the template to determine if the parameter is configured or not: <code>node:exists(OptionalVariable)</code> .</p>
-DmergeAttributes=(true false)	<p>This property can only be used, if you have set -DmergeConfigs=true. The property controls how to deal with the attributes of nodes that are present in more than one input file. The default value is <code>true</code>.</p>

System property	Description
	<ul style="list-style-type: none"> ▶ If this property is set to <code>true</code>, attributes with multiple values are merged so that the resulting attribute contains all values. How single-value attributes are dealt with, is controlled by the value of the property <code>-DoverwriteAttributes</code>. ▶ If this property is set to <code>false</code>, the result only contains those attributes and their values that were present when the node was read the first time.
<code>-DoverwriteAttributes=(true false)</code>	<p>This property can only be used, if you have set – <code>DmergeAttributes=true</code>. The property controls how to deal with the attributes of nodes that are present in more than one input file. The default value is <code>true</code>.</p> <p>If more than one input file is used, this property controls how to handle the attributes of node present in several files.</p> <ul style="list-style-type: none"> ▶ Set the property to <code>true</code>, to overwrite the values of the attributes with their last occurrence. ▶ Set the property to <code>false</code>, to use the value of the first occurrence of that attribute.
<code>-DwriteXPathAttributes=(true false)</code>	<p>When you write an AUTOSAR schema file, not all XPath expressions used for attributes can be converted to a fixed value as needed by the AUTOSAR format. The commandline option <code>writeXPathAttributes</code> controls how to handle such non-computable attributes.</p> <p>The default value for <code>writeXPathAttributes</code> is <code>true</code>.</p> <p>If you set <code>writeXPathAttributes</code> to the value <code>true</code>, and:</p> <ul style="list-style-type: none"> ▶ If a value can be determined, the converter tries to evaluate the expressions of computable attributes and writes the resulting value. ▶ If an attribute cannot be evaluated, e.g. if it accesses a non-existing configuration value, the commandline operation fails with an error. <p>If you set <code>writeXPathAttributes</code> to the value <code>false</code> computable attributes are never evaluated. Instead</p> <ul style="list-style-type: none"> ▶ if the corresponding AUTOSAR XML tag is optional, it is omitted ▶ or a reasonable default is written. <p>A warning is issued on the commandline for each computable attribute.</p>

System property	Description
	<p>When any of the above conversions has taken place: If the conversion succeeds with no error and the input to the conversion is an AUTOSAR-compatible model, an AUTOSAR-compatible output is produced.</p> <p>Depending on the value of <code>writeXPathAttributes</code>, the results of the conversion vary:</p> <ol style="list-style-type: none"> 1. If <code>writeXPathAttributes</code> is set to <code>true</code> and if a XDM parameter definition is converted to AUTOSAR format, AUTOSAR tags that result from a computable attribute, which could not be evaluated, are handled the following way: <ul style="list-style-type: none"> ▶ LOWER-MULTIPLICITY : Conversion fails with an error. ▶ UPPER-MULTIPLICITY : Conversion fails with an error. ▶ ENUM-LITERAL-DEF : Conversion fails with an error. ▶ DEFAULT-VALUE : Conversion fails with an error. ▶ MIN : <ul style="list-style-type: none"> ▶ If no <code>INVALID</code> attribute is used in the input model, no <code>MIN</code> tag is written. ▶ If an <code>INVALID</code> attribute is used that is not of type <code>Range</code>, a warning is issued on the commandline and a reasonable default is written to the resulting AUTOSAR file. ▶ If an <code>INVALID</code> attribute of type <code>Range</code> is used but no minimum value is set, no <code>MIN</code> tag is written. ▶ MAX : <ul style="list-style-type: none"> ▶ If no <code>INVALID</code> attribute is used in the input model, no <code>MAX</code> tag is written. ▶ If an <code>INVALID</code> attribute is used that is not of type <code>Range</code>, a warning is issued on the commandline and a reasonable default is written to the resulting AUTOSAR file. ▶ If an <code>INVALID</code> attribute of type <code>Range</code> is used but no maximum value is set, no <code>MAX</code> tag is written. 2. If <code>writeXPathAttributes</code> is set to <code>false</code> and if an XDM parameter definition is converted to AUTOSAR format, AUTOSAR tags that result from a computable attribute are handled the following way, no matter if the tag can be evaluated or not:

System property	Description
	<ul style="list-style-type: none"> ▶ LOWER-MULTIPLICITY : A warning is printed in the commandline. A LOWER-MULTIPLICITY of 0 is written to the AUTOSAR file. ▶ UPPER-MULTIPLICITY : A warning is printed in the commandline. An UPPER-MULTIPLICITY of * is written to the AUTOSAR file. ▶ ENUM-LITERAL-DEF : A warning is printed and a single ENUM-LITERAL-DEF with the value undefined is written to the AUTOSAR file. ▶ DEFAULT-VALUE : A warning is printed. No DEFAULT-VALUE is written to the AUTOSAR file. ▶ MIN : <ul style="list-style-type: none"> ▶ If no INVALID attribute is used in the input model, no MIN tag is written. ▶ If an INVALID attribute is used that is not of type Range, a warning is issued on the commandline and a reasonable default is written to the resulting AUTOSAR file. ▶ If an INVALID attribute of type Range is used but no minimum value is set, no MIN tag is written. ▶ MAX : <ul style="list-style-type: none"> ▶ If no INVALID attribute is used in the input model, no MAX tag is written. ▶ If an INVALID attribute is used that is not of type Range, a warning is issued on the commandline and a reasonable default is written to the resulting AUTOSAR file. ▶ If an INVALID attribute of type Range is used but no maximum value is set, no MAX tag is written.
-DpreserveConfiguration= (true false)	<p>With this property you may convert a pre-configuration file without changing the values configured.</p> <p>The default value is <code>false</code>.</p> <ul style="list-style-type: none"> ▶ To add missing nodes and overwrite default values, keep the default value at <code>false</code>. ▶ To not add missing nodes and not overwrite values, set <code>-DpreserveConfiguration</code> to <code>true</code>.

System property	Description
	The imported configuration is not changed in any way.
-DpredefinedVariant=<Name>	<p>This property enables the variant resolving and specifies the predefined Variant name that will be taken from the input files.</p> <p>If the parameter is not set, by default the variants resolving is disabled.</p> <p>For detailed information about variant handling please refer to Section 7.12, “Variant Handling”).</p>
-Dtarger=<target>	Defines the target architecture, e.g. -Dtarger=WINDOWS. This is necessary when working with module configurations, which use ECU Resources. Further information on ECU Resources is available in the EB tresos Studio developer's guide, chapter Configuration models/ECU Resource Manager API .
-Dderivate=<derivate>	Defines the target's derivate, e.g. -Dderivate=Win32x86. This is necessary when working with module configurations, which use ECU Resources. Further information on ECU Resources is available in the EB tresos Studio developer's guide, chapter Configuration models/ECU Resource Manager API . If the ECU Resources require additional parameters, such as subderivate , you need to specify any additional parameter with -D<parameter-name>=<parameter-value>. Information on how to specify these parameters is available in the EB tresos Studio developer's guide, chapter ECU Resource Manager API/Characteristics when using the legacy commandline/Defining parameter values for the DefaultEcuResourceFinder .
-DEcuResourceModuleIds=<Module1;Module2;...>	<p>Defines a comma- or semicolon-separated list of moduleIds that are necessary for finding relevant ECU Resources, e.g.</p> <pre>-DEcuResourceModuleIds= Com_TS_T99D01M01I05R01;Rte_TS_T99D01M01I04R02</pre> <p>. Information on how to define configured modules for the Resource Finder is available in the EB tresos Studio developer's guide, chapter ECU Resource Manager API/Characteristics when using the legacy commandline/Defining configured modules for the DefaultEcuResourceFinder .</p>

Additional system properties when converting system description data:

System property	Description
-DcanIdFilterValueDBC=<mask>	<p>Pattern that matches CAN-IDs, which are not to be imported by the DBC Importer. The pattern can consist of multiple CAN-IDs or ranges of CAN-IDs separated by semicolon, e.g.</p> <p>-DcanIdFilterValueDBC="10;20;30-40"</p> <p>This prevents the following CAN-IDs from being loaded: 10, 20 and 30 up to and including 40.</p>
-DecuInstance=<path>	<p>(AUTOSAR) path to an ECU-INSTANCE entity inside an imported system model.</p> <p>This option selects the ECU that is going to be imported. If you want to import software components, data and service port mappings from a system, use this option. For further information, refer to the table below.</p> <p>This option applies to the AUTOSAR System Description Importer.</p> <p>If this option is set, you need to also set the option <code>-Dsystem</code>.</p> <p>If this option is set, you must not set the option <code>-DtoplevelComposition</code>.</p>
-DecuSwComposition=<path>	<p>(AUTOSAR) path to an ECU-SW-COMPOSITION entity inside an imported system model.</p> <p>This option selects the ECU SW composition for the imported ECU.</p> <p>This option applies to the AUTOSAR System Description Importer.</p> <p>You may only use this option, if one of the following conditions is met:</p> <ul style="list-style-type: none"> ▶ The options <code>-DecuInstance</code> and <code>-Dsystem</code> are set. ▶ The option <code>-DtoplevelComposition</code> is set.
-DpackageDBC=<name>	<p>Names the AR-PACKAGE in the project system model, into which to import system parameters from a DBC file.</p> <p>The default value for this option is <code>DBCImport</code></p>
-DimportSignalEnumsDBC=(true false)	<p>If set to <code>false</code>, Signal Enumeration Constants will not be imported.</p> <p>The default value for this option is <code>true</code></p>
-DselectorFieldHandlingDBC=<mode>	<p>If <mode> is set to <code>NoComSignalForSelector</code>, multiplexer signals are not mapped to Com signals.</p>

System property	Description
	<p>If <mode> is set to <code>ComSignalForSelector</code>, multiplexer signals are mapped to Com signals.</p> <p>If <mode> is set to <code>AutosarCompliant</code>, multiplexer signals are mapped to Com signals, identical segments for all dynamic parts are created, no stuffing of segments is performed.</p> <p>The default value for this option is <code>NoComSignalForSelector</code>. For detailed information on this option, refer to the section called “I-Pdu Multiplexer Configuration”.</p>
<code>-DpackageFIBEX=<name></code>	<p>Names the AR-PACKAGE in the project system model, into which to import system parameters from a Fibex file.</p> <p>The default value for this option is <code>FIBEXImport</code></p>
<code>-DimportSignalEnumsFbx=(true false)</code>	<p>If set to <code>false</code>, Signal Enumeration Constants will not be imported.</p> <p>The default value for this option is <code>true</code></p>
<code>-DpackageLDF=<name></code>	<p>Names the AR-PACKAGE in the project system model, into which to import system parameters from a LDF file.</p> <p>The default value for this option is <code>LDFImport</code></p>
<code>-Dsystem=<path></code>	<p>(AUTOSAR) path to the SYSTEM entity inside an imported system model.</p> <p>This option selects the system that is going to be imported. If you want to import software components, data and service port mappings from a system, use this option. For further information, refer to the table below.</p> <p>This option applies to the AUTOSAR System Description Importer.</p> <p>If this option is set, you need to also set the option <code>-DecuInstance</code>.</p> <p>If this option is set, you must not set the option <code>-DtoplevelComposition</code>.</p>
<code>-DtoplevelComposition=<path></code>	<p>(AUTOSAR) path to a COMPOSITION-TYPE entity inside an imported system model.</p> <p>This option selects the top level composition that is going to be imported. If you want to import software components from a top level composition, and do not want to import mappings, use this option. For further information, refer to the table below.</p> <p>This option applies to the AUTOSAR System Description Importer.</p>

System property	Description
	<p>If this option is set, you must not set the options <code>-DecuInstance</code> and <code>-Dsystem</code>.</p>
<code>-DatpSplitableMerge=(true false)</code>	<p>This property specifies if the stereotype <code>splitable</code> (3.x) / <code>atpSplitable</code> (4.-x) is used to merge multiple system models.</p> <p>If the property is set to <code>true</code>, the <code>atpSplitable</code> stereotype is used to merge, else the old merge algorithm is used.</p> <p>The default is <code>false</code>.</p>
<code>-DignoreOpenReferences=(true false)</code>	<p>This property can only be used during conversion of a AUTOSAR system description. The property controls how to deal with dangling references of the system description during export.</p> <p>If the property is set to <code>true</code> all dangling references will be ignored during conversion and the target file will be created.</p> <p>If the property is set to <code>false</code> the conversion will be canceled if the input file contains dangling references.</p> <p>The default is <code>true</code>.</p>
<code>-DvalidationVariant=(strict standard)</code>	<p>This property defines the AUTOSAR XML schema variant against which the selected files are validated, if there is more than one variant available (e.g. AUTOSAR 4 XML files).</p> <p>Only applies, if the system property <code>Validate</code> is set to <code>true</code> (see Section 7.11.1.3, “System properties common to all commands”).</p> <p>The default is <code>strict</code>.</p>

NOTE**Selecting the software components, data mappings and service port mappings to be imported by the AUTOSAR System Description Importer**

EB tresos Studio does not provide an additional parameter to select, which software components, data mappings and service port mappings are to be imported by the AUTOSAR System Description Importer. Instead of a separate parameter, the items to be imported are determined by evaluating, which other parameters are provided. The following rules apply:

Options set	Imported software components, data mappings and service port mappings
None of the following is set: Neither the options <code>-DecuInstance</code> and <code>-Dsystem</code> nor the option <code>-DtoplevelComposition</code> .	All software components, <code>EcuSwComposition s</code> , data mappings and service port mappings are removed from the system model to be imported.
The option <code>-DtoplevelComposition</code> is set.	All data mappings and service port mappings are removed. Software components and <code>EcuSwComposition s</code> are retained in the system model to be imported.
The options <code>-DecuInstance</code> and <code>-Dsystem</code> are set.	All software components, <code>EcuSwComposition s</code> , data mappings and service port mappings are retained in the system model to be imported.

7.11.5.1.3. Options

The `convert` command supports following options:

Parameter	Description
<code>-c <config-path></code>	By default, the directory <code><base.path>\bin</code> is used to search the module configuration files. To search the module configuration files in the directory specified as <code><config-path></code> , use the option <code>-c <config-path></code> . <code>base.path</code> is the value of the system property <code>-Dbase.path</code> . If the system property is not set, an error is displayed. For details about system properties, refer to Section 7.11.1.3, “System properties common to all commands” .
<code>-n <config-name></code>	Name of the module configurations of the configuration files to load. Use this option to ignore other module configurations contained in the files specified. You may use this option multiple times.
<code>-u</code>	Use this option to generate files with Unix line endings.

7.11.5.1.4. Parameters

In addition to the common parameters `<system_property>` and `<workspace>`, the `convert` command also recognizes the following parameters:

Parameter	Description
<code><input-file>[@<type>]</code>	Use <code><input-file></code> for the name of the file to read in. The parameter can be passed multiple times to load more than one file. To explicitly specify the content type of the file, use the optional argument <code>@<type></code> . For more information on which content types are available, refer to Section 4.5.3, “File content types” .
<code><output-file>[@<type>]</code>	Use <code><output-file></code> for the name of the file to write. To explicitly specify the content type of the file, use the optional argument <code>@<type></code> . For more information on which content types are available, refer to Section 4.5.3, “File content types” .

7.11.5.1.5. Example

The following are several examples of how to use the command `legacy convert`:

```
tresos_cmd.bat -DValidate=false -DMapOptionalAsList=false
    legacy convert in.epd out.xdm
```

The example above simply converts an EPD file to an XDM file and marks optional parameters with the optional attribute instead of creating lists around them.

```
tresos_cmd.bat -DValidate=false -DMapOptionalAsList=false
    -Duuids=true legacy convert
    in1.epd in2.epd in3.xdm out.epc@asc:4.0.3
```

This example loads the three files `in1.epd`, `in2.epd` and `in3.xdm`. The content of the files is merged and saved to the output file called `out.epc`. The content of the output file conforms to AUTOSAR 4.0.3. No validation is done during the conversion and new UUIDs are created for each node.

```
tresos_cmd.bat -Dsysten="/SYSTEM/system"
    -DecuInstance="/AUTOSAR/ECUInstance/ecuInstance_1"
    legacy convert
    in1.xml@fibex in2.arxml@sysd out.tdb
```

This example loads the files *in1.xml* (importing the file using the Fibex Importer) and *in2.arxml* (importing the file as AUTOSAR system description) and merges the content of the two files. The system /SYSTEM/system and the ECU /AUTOSAR/ECUInstance/ecuInstance_1 must both be contained in the file *in2.arxml*). Both are selected for import. All software components, EcuSwComposition s, data mappings and service port mappings are retained in the system model. The resulting system model is saved to the output file called *out.-tdb* using the tresosDB file format.

```
tresos_cmd.bat -DValidate=false -DignoreOpenReferences=false  
    legacy convert System20.arxml@sysd System314.arxml@sysd:3.1.4
```

The example converts an AUTOSAR system description in version 2.0 into an AUTOSAR system description in version 3.1.4. If the source file contains dangling references the file will not be converted (-DignoreOpenReferences=false).

7.11.5.2. Using XDM files in different EB tresos Studio versions

If you are working with several releases of EB tresos Studio on the same configuration data, you may want to use an XDM created with a new version of EB tresos Studio in your old installation. Newer versions of EB tresos Studio are always able to read old XDM files, but they may add new features to the XDM format that cannot be understood by older versions.

It is possible to use such a file with an older EB tresos Studio version. But you have to convert that file first into the older version. During that conversion, information that cannot be mapped into the old format may get lost!

NOTE

The XDM versions numbers are not related to the AUTOSAR version numbers.



To convert a newer version of the XDM format to the older version, use the `legacy convert` command together with the EB tresos Studio installation with which you created the file.

The `legacy convert` command is explained in [Section 7.11.5.1, “Merging files and converting between different file formats”](#). All you need to do to convert between XDM file versions is to specify the correct content type for the files. The available content types are listed in [Section 4.5.3, “File content types”](#). You can use any `xdm:*` content type.

7.11.5.2.1. Example

```
tresos_cmd.bat legacy convert new.xdm old.xdm@xdm:2.0
```

This example converts an XDM from a new EB tresos Studio version to the older XDM 2.0 format.

7.11.5.3. Upgrading the AUTOSAR compliance of XDM files

The XDM files contain a representation of the AUTOSAR data structures. Sometimes it is necessary to convert that representation to a newer AUTOSAR revision because the newer revision contains incompatible changes. E.g. within AUTOSAR 2.x choice containers do not have a SHORT-NAME, but in AUTOSAR 3.x they do. Instead of editing these changes manually inside the XDM file you may use the `legacy xdmconvert` command.

7.11.5.3.1. Syntax

The syntax of the command is shown in the following example:

```
tresos_cmd.bat [<system_property>...] [-data <workspace>]
    legacy xdmconvert <mode>
    <input-file>[@<type>]... <output-file>[@<type>]
    [-variant <variant>]
```

7.11.5.3.2. System properties

This command only recognizes the system properties common to all commands. For details about common system properties refer to [Section 7.11.1.3, “System properties common to all commands”](#).

7.11.5.3.3. Parameters

In addition to the common parameters `<system_property>` and `<workspace>`, this command also recognizes the following parameters:

Parameter	Description
<code><input-file>[@-<type>]</code>	Use <code><input-file></code> for the name of the file to read in. The parameter can be passed multiple times to load more than one file. The optional argument <code>@-</code>

Parameter	Description
	<type> can be used to explicitly specify the content type of the file. See Section 4.5.3, “File content types” for more information on what content types are available.
<output-file>[@-<type>]	Use <output-file> for the name of the file to write. To explicitly specify the content type of the file, use the optional argument @<type>. For more information on which content types are available, refer to Section 4.5.3, “File content types” .
<mode>	Converts an XDM file from one AUTOSAR version to another. Currently, two modes are supported: <code>asc2xToAsc30</code> and <code>asc2xToAsc31</code> . They convert an XDM file (representing an AUTOSAR 2.x file) to an XDM file (representing an AUTOSAR 3.0 or 3.1 file).
-variant <variant>	<p>This parameter defines which SUPPORTED-CONFIG-VARIANT needs to be set at the MODULE-DEF. This parameter is optional. Possible values are:</p> <ul style="list-style-type: none"> ▶ <code>pre</code> for a pre-compile time configuration, ▶ <code>link</code> for a link-time configuration, and ▶ <code>post</code> for a post-build configuration. <p>For further information, refer to the EB tresos Studio developer's guide, chapter Developing modules/How to upgrade an AUTOSAR 2.x module to 3.x .</p>

7.12. Variant Handling

7.12.1. General

7.12.1.1. Variant Handling general information

EB tresos Studio does not currently store variants (as defined by AUTOSAR).

Variants are resolved on import.

7.12.1.2. Resolving variants example: commandline

TIP

You may enable variant resolving by using the system property: -DpredefinedVariant=<Name>.



This property can be used with the following commands:

- ▶ legacy convert
- ▶ legacy generate
- ▶ legacy make
- ▶ legacy verify

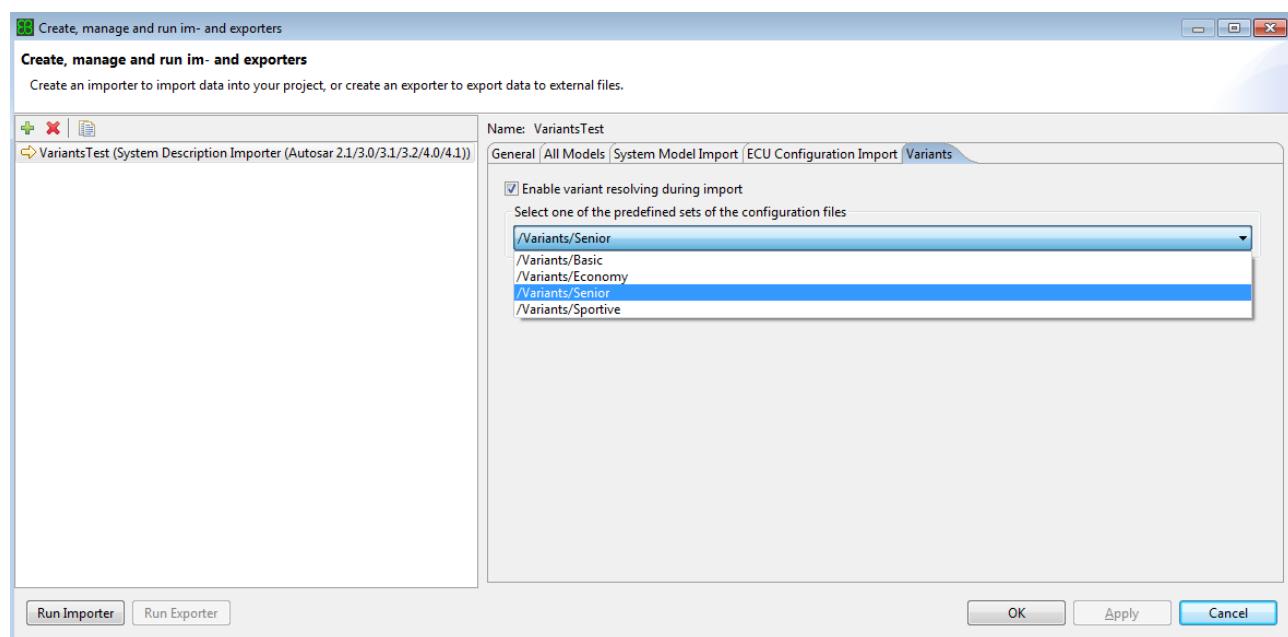
The following example can be used in the commandline:

```
tresos_cmd.bat -DpredefinedVariant=/Variants/Basic legacy convert inputFile@sysd
outputFile@sysd:4.0.3
```

7.12.1.3. Resolving variants example: Im- and Export wizard

TIP

Variants resolving can be enabled using the Im- and Export wizard. The configuration can be done in Variants tab.



More details about the variants tab are available in [Section 7.10.2.3.5, “Variants tab”](#).

7.13. Post-build loadable configuration using Multiple Configuration Containers

7.13.1. General post-build loadable concept

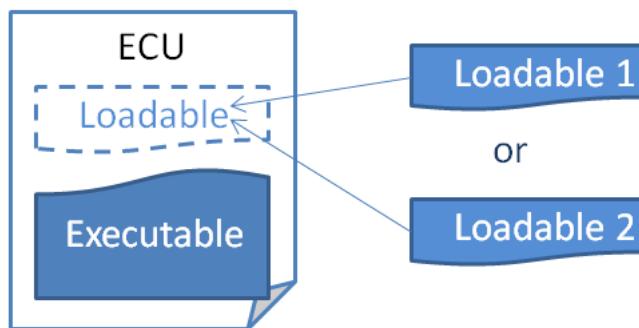


Figure 7.139. Loadables for an ECU

The picture above visualizes the use-case of the post-build loadable (PBL) concept. There, an ECU executable is flashed to an ECU which will later on be completed by any loadable configuration. For this use-case, it is not necessary to re-compile or re-build the BSW module since the loadable only contains configuration data which will be stored at a known memory location. Therefore, this can e.g. be used by the OEM to use the same ECU within different cars with just a slightly changed (loadable) configuration.

7.13.2. Using Multiple Configuration Containers for post-build loadables

To configure multiple post-build loadables within one configuration project, so-called "Multiple Configuration Containers" (MCCs) may be used where each MCC represents one loadable configuration. The developer of the module (and its code-generator) selectively decided which parameters of the MCC may be changed at post-build time and are part of the loadable. Therefore, after changing the configuration time of the project to "post-build only" (see chapter [Section 4.5.2.1, “Configuration time of projects”](#)), only those parameters can still be changed.

7.13.3. Defining valid sets of "Multiple Configuration Containers"

Since each module of the project may potentially configure several MCCs, it is necessary to define which MCCs throughout all modules belong to one post-build loadable configuration. This can be configured using the generic configuration editor of EB tresos Studio of one "Post-build configuration Management" module (PbcfgM).

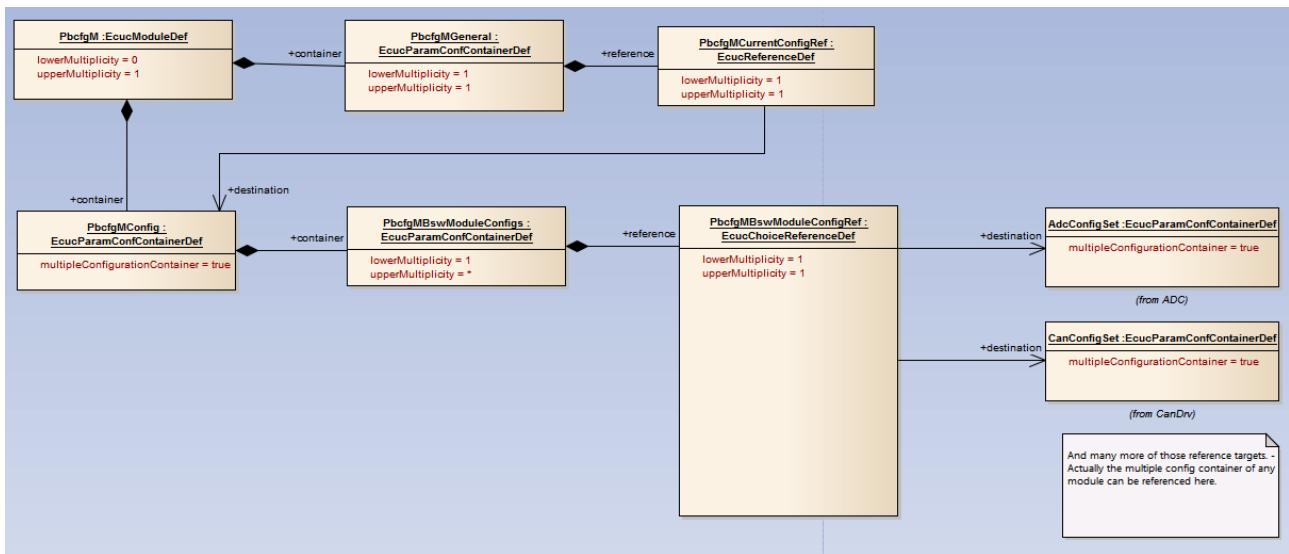


Figure 7.140. PbcfgM module

As shown in the picture above, the PbcfgM module also contains a MCC "PbcfgMConfig" with a container "PbcfgMBswModuleConfigRef" which contains references to the right MCC of each module. Via this mechanism, each MCC of the PbcfgM defines one set of MCCs which belong to one post-build loadable configuration. If there are e.g. two loadable configurations defined within one project there must be two MCCs defined within the PbcfgM module.

As you can see in the picture, there is also a reference called "PbcfgMCurrentConfigRef" which selects one MCC of the PbcfgM module - and with this a complete post-build loadable configuration. This reference is required to be able to decide which is the "current set of MCCs". The following sub-chapters describe what this is used for.

TIP

For technical reasons, the "current MCC" of a module is automatically moved to the first position within its list of MCCs if the configuration of the PbcfgM changes.

7.13.3.1. Generating code

When generating code at post-build time, the code-generators will only generate code for the configured "current set of MCCs". To generate code for another post-build loadable configuration, the reference "PbcfgMCurrentConfigRef" must be configured properly.

7.13.3.2. Importing a system description

When importing a system description, the ECU configuration is automatically updated (if "ECU configuration import" is enabled within the importer preferences). Since an import logically belongs to exactly one ECU loadable only, this is always be done for the "current set of MCCs" only. To import a system description into another post-build loadable configuration, the reference "PbcfgMCurrentConfigRef" must therefore be configured properly.

7.13.3.3. Guided Configuration wizards

Guided configuration wizards may only run on one specific MCC per module (e.g. the Handheld wizard). Therefore it may be required to configure the reference "PbcfgMCurrentConfigRef" properly. Please consider the description of the specific wizard for details.

7.13.4. Value Constraints

7.13.4.1. Readonly

When changing the config time of a project to "post-build only", most of the precompile and linktime parameters will become readonly. The exception of this rule is, if such a parameter is contained within a post-build changeable container whose content is not the target of a precompile or linktime reference, the parameter is editable (even if it is precompile or linktime). This also applies for new post-build changeable container instances. Or in other words: if a precompile parameter is editable at post-build time, it is contained within a post-build changeable container and has not been used by any precompile or linktime generator.

7.13.4.2. Dependencies between parameters

Two MCCs within one module represent two different loadable configurations but they have a huge portion of duplicated parameters which actually represent the same value. For this reason, there are some restrictions between the parameters within different MCCs:

- ▶ The value of precompile and linktime parameters and references that exist in several MCCs must have the same value in all MCCs. See also Autosar constraint [constr_5501] for details.
- ▶ A container as part of a list may only be removed at post-build time, if it - or one of its subcontainers - is not target of a precompile or linktime reference. See also Autosar constraint [constr_5504].
- ▶ A precompile or linktime parameter which exists in several MCCs shall have the same value in all MCCs. See also Autosar constraint [constr_5501] for details.
- ▶ EcuFunctionNames parameters may not introduce new symbolic names at post-build time. See also Autosar constraint [constr_5503] for details.

The reason for all those constraints is, that the ECU executable (which has already been flashed to the ECU) relies on the parameter values and changing them is simply not possible anymore since the associated code has already been generated.

8. References

8.1. Upgrading system models

In this chapter you will learn when the system model of a project is upgraded to a higher AUTOSAR version and how this is done.

The System Description Importer and the TDB Importer directly import into the system model specified in the imported files; the importers for Fibex, LDF and DBC files convert the data contained in the imported files into a system model of AUTOSAR version 3.x.

When you create your EB tresos Studio project, you must specify the AUTOSAR release version for this project. If you then use an importer based on an older AUTOSAR major release version to import configuration data, or if you import a system description/TDB file based on an older AUTOSAR major release version, the imported data is automatically upgraded to the project's AUTOSAR release version.

The following sections describe how the data is upgraded between two specific AUTOSAR major release versions. The upgraded entities and their properties are referred to by their XML tags as specified in the respective AUTOSAR XML schema files. Note that only entities defined in the AUTOSAR system template are described, i.e. no description concerning the upgrade of entities defined in the AUTOSAR software component template is provided here.

The following sections will refer to the system model that is upgraded as *source model*, whereas the system model that is the result of the upgrade is called *target model*.

The entities and parameters of source and target model are described in the XML schema files for AUTOSAR 3.1.4 (source model) and AUTOSAR 4.0.2 (target model). In analogy to the AUTOSAR system model path definitions, the notation `<parenttag>/<childtag>` describes an XML child tag `<childtag>` within the context of its enclosing `<parenttag>`.

Wherever the importers for Fibex and LDF files provide useful system data for which no counterpart in the AUTOSAR version 3.x system model exists, the importers store the data in custom parameters (as opposed to canonical AUTOSAR parameters). Since AUTOSAR 4.x introduced parts of these parameters, the model upgrade uses the custom parameters in the source model to configure newly introduced AUTOSAR 4.x parameters in the target model. The following sections explicitly describe which data of the target model is configured during LDF or Fibex import.

8.1.1. Upgrading a system model from AUTOSAR 3.1.x to AUTOSAR 4.0.2

This section describes how an AUTOSAR 3.1.x system model is upgraded to AUTOSAR 4.0.2.

NOTE



This chapter does not describe the upgrade of AUTOSAR 3.2.x system models

This chapter does *not* apply to the upgrade of AUTOSAR 3.2.x system models. For information on upgrading your AUTOSAR 3.2.x system model, refer to [Section 8.1.2, “Upgrading a system model from AUTOSAR 3.2.x to AUTOSAR 4.0.3”](#).

If not stated explicitly, all entities in the 4.0.x target model are located in the same package as they were in the 3.1.x source model.

If not explicitly specified otherwise, the parameters `SHORT-NAME`, `CATEGORY`, and `UUID` of Identifiable entities are directly upgraded into their counterparts in the target model.

8.1.1.1. Topology

8.1.1.1.1. SYSTEM

Source model entity: `SYSTEM`

Target model entity: `SYSTEM`

Target model property	Source model property
<code>SYSTEM-VERSION</code>	<code>SYSTEM-VERSION</code>
<code>FIBEX-ELEMENT-REFS</code>	<code>FIBEX-ELEMENT-REFS</code>
<code>MAPPINGS/SYSTEM-MAPPING</code>	<code>MAPPING</code>
<code>SYSTEM/ROOT-SOFTWARE-COMPOSITIONS/ROOT-SW-COMPOSITION-PROTOTYPE</code>	<code>SYSTEM/SOFTWARE-COMPOSITION</code>

Changed or moved parameters

`FIBEX-ELEMENT-REFS`

In `FIBEX-ELEMENT-REFS`, references to the following types of source model entities are upgraded to their target model counterparts: `CAN-CLUSTER`, `COMMUNICATION-CLUSTER`, `DCM-I-PDU`, `ECU-INSTANCE`, `FLEXRAY-CLUSTER`, `FRAME`, `GATEWAY`, `I-PDU-GROUP`, `I-SIGNAL`, `LIN-CLUSTER`, `MULTIPLEXED-I-PDU`, `N-PDU`, `NM-PDU`, `SIGNAL-I-PDU`, `SUBSTITUTION-FRAME`.

8.1.1.1.2. ECU-INSTANCE

Source model entity: ECU-INSTANCE

Target model entity: ECU-INSTANCE

Target model property	Source model property
ASSOCIATED-COM-I-PDU-GROUP-REFS/ASSOCIATED-COM-I-PDU-GROUP-REF	ASSOCIATED-I-PDU-GROUP-REFS/ASSOCIATED-I-PDU-GROUP-REF
COM-CONFIGURATION-GW-TIME-BASE	COM-PROCESSING-PERIOD
COM-CONFIGURATION-RX-TIME-BASE	COM-PROCESSING-PERIOD
COM-CONFIGURATION-TX-TIME-BASE	COM-PROCESSING-PERIOD
COMM-CONTROLLERS	COMM-CONTROLLERS
CONNECTORS	CONNECTORS
DIAGNOSTIC-ADDRESS	DIAGNOSTIC-ADDRESS
SLEEP-MODE-SUPPORTED	SLEEP-MODE-SUPPORTED
WAKE-UP-OVER-BUS-SUPPORTED	WAKE-UP-OVER-BUS-SUPPORTED
No target model parameter is available.	COM-CONFIGURATION-ID
No target model parameter is available.	PDU-R-CONFIGURATION-ID
Not upgraded.	RESPONSE-ADDRESSSS/RESPONSE-ADDRESS

Reason for not upgraded source model parameter

RESPONSE-ADDRESSSS/RESPONSE-ADDRESS

The parameter RESPONSE-ADDRESSSS/RESPONSE-ADDRESS is not upgraded to the target model since the contained diagnostic addresses cannot be uniquely assigned to concrete interfaces.

8.1.1.1.3. CAN-CLUSTER

Source model entity: CAN-CLUSTER

Target model entity: CAN-CLUSTER

Target model property	Source model property
CAN-CLUSTER-CONDITIONAL/PHYSICAL CHANNELS	PHYSICAL-CHANNELS
CAN-CLUSTER-CONDITIONAL/PROTOCOL-NAME	PROTOCOL-NAME
CAN-CLUSTER-CONDITIONAL/PROTOCOL-VERSION	PROTOCOL-VERSION

Target model property	Source model property
CAN-CLUSTER-CONDITIONAL/SPEED	SPEED/1000
No target model parameter is available.	MAX-FRAME-LENGTH
No target model parameter is available.	NM-REPEAT-MESSAGE-SUPPORT

Changed or moved parameters

NM-NODE-DETECTION-ENABLED and NM-NODE-ID-ENABLED

The source model parameters NM-NODE-DETECTION-ENABLED and NM-NODE-ID-ENABLED are configured in the target model entity NM-ECU, see [Section 8.1.1.7.2, “NM-ECU”](#).

SPEED

In the target model, the parameter SPEED is given in kbit/s, whereas the source model provides it in bit/s. If the source model parameter is not divisible by 1000 without rest, it cannot be upgraded at all.

NM-BUS-LOAD-REDUCTION-ACTIVE and NM-WAIT-BUS-SLEEP-TIME

The parameters NM-BUS-LOAD-REDUCTION-ACTIVE and NM-WAIT-BUS-SLEEP-TIME are configured as NM-BUSLOAD-REDUCTION-ACTIVE and NM-WAIT-BUS-SLEEP-TIME in the target model entity CAN-NM-CLUSTER, see [Section 8.1.1.7.3, “CAN-NM-CLUSTER”](#).

NM-BUS-LOAD-REDUCTION-ENABLED

The parameter NM-BUS-LOAD-REDUCTION-ENABLED is configured as NM-BUSLOAD-REDUCTION-ENABLED in the target model entity CAN-NM-CLUSTER-COUPLING, see [Section 8.1.1.7.8, “FLEXRAY-NM-CLUSTER-COUPLING”](#).

NM-LOWER-CAN-ID and NM-UPPER-CAN-ID

The parameters NM-LOWER-CAN-ID and NM-UPPER-CAN-ID are configured in the target model entity CAN-NM-RANGE-CONFIG parameters LOWER-CAN-ID and UPPER-CAN-ID, which is aggregated in CAN-NM-NODE, see [Section 8.1.1.7.4, “CAN-NM-NODE”](#).

8.1.1.4. CAN-COMMUNICATION-CONTROLLER

Source model entity: CAN-COMMUNICATION-CONTROLLER

Target model entity: CAN-COMMUNICATION-CONTROLLER

Target model property	Source model property
CATEGORY	CATEGORY
CAN-COMMUNICATION-CONTROLLER-CONDITIONAL/CAN-CONTROLLER-ATTRIBUTES/CAN-CONTROLLER-CONFIGURATION/SYNC-JUMP-WIDTH	CAN-CONTROLLER-ATTRIBUTES/CAN-CONTROLLER-CONFIGURATION/SYNC-JUMP-WIDTH

Target model property	Source model property
CAN-COMMUNICATION-CONTROLLER-CONDITION-AL/CAN-CONTROLLER-ATTRIBUTES/CAN-CONTROLLER-CONFIGURATION/TIME-SEG-1	CAN-CONTROLLER-ATTRIBUTES/CAN-CONTROLLER-CONFIGURATION/TIME-SEG-1
CAN-COMMUNICATION-CONTROLLER-CONDITION-AL/CAN-CONTROLLER-ATTRIBUTES/CAN-CONTROLLER-CONFIGURATION/TIME-SEG-2	CAN-CONTROLLER-ATTRIBUTES/CAN-CONTROLLER-CONFIGURATION/TIME-SEG-2
No target model parameter is available.	CAN-CONTROLLER-ATTRIBUTES/CAN-CONTROLLER-CONFIGURATION/NUMBER-OF-SAMPLES
No target model parameter is available.	CAN-CONTROLLER-ATTRIBUTES/CAN-CONTROLLER-CONFIGURATION/PROPAGATION-DELAY
CAN-COMMUNICATION-CONTROLLER/CAN-CONTROLLER-CONFIGURATION-REQUIREMENTS	No source model parameter is available.

8.1.1.1.5. CAN-COMMUNICATION-CONNECTOR

Source model entity: COMMUNICATION-CONNECTOR

Target model entity: CAN-COMMUNICATION-CONNECTOR

Target model property	Source model property
CATEGORY	CATEGORY
COMM-CONTROLLER-REF	COMM-CONTROLLER-REF
ECU-COMM-PORT-INSTANCES	ECU-COMM-PORT-INSTANCES

Changed or moved parameters

CHANNEL-REF

The source model parameter CHANNEL-REF has been replaced by the target model parameter COMM-CONNECTORS/COMMUNICATION-CONNECTOR-REF-CONDITIONAL of the target model entity CAN-PHYSICAL-CHANNEL.

NM-ADDRESS and TP-ADDRESS

The source model parameters NM-ADDRESS and TP-ADDRESS are configured in the target model entities CAN-NM-NODE and TP-ADDRESS, see [Section 8.1.1.7.4, “CAN-NM-NODE”](#).

8.1.1.6. CAN-PHYSICAL-CHANNEL

Source model entities: PHYSICAL-CHANNEL, COMMUNICATION-CONNECTOR

Target model entity: CAN-PHYSICAL-CHANNEL

Target model property	Source model property
CATEGORY	CATEGORY
FRAME-TRIGGERINGS	FRAME-TRIGGERINGSS
PDU-TRIGGERINGS	I-PDU-TRIGGERINGS
I-SIGNAL-TRIGGERINGS	I-SIGNAL-TRIGGERINGS

Changed or moved parameters

COMM-CONNECTORS

The target model parameter COMM-CONNECTORS is configured by retrieving all target model CAN-COMMUNICATION-CONNECTOR entities that meet the following condition: The source model counterpart of the CAN-COMMUNICATION-CONNECTOR references the source model counterpart of the upgraded CAN-PHYSICAL-CHANNEL via CHANNEL-REF.

CAN-TP-CONNECTION-CHANNEL

The CAN-TP-CONNECTION-CHANNEL entities aggregated in TP-CHALNELS are upgraded to CAN-TP-CONNECTION and CAN-TP-CHANNEL entities in the target model, aggregated in CAN-TP-CONFIG.

8.1.1.1.7. FLEXRAY-CLUSTER

Source model entity: FLEXRAY-CLUSTER

Target model entity: FLEXRAY-CLUSTER/FLEXRAY-CLUSTER-CONDITIONAL

Target model property	Source model property
PHYSICAL-CHANNELS	PHYSICAL-CHANNELS
PROTOCOL-NAME	PROTOCOL-NAME
PROTOCOL-VERSION	PROTOCOL-VERSION
SPEED	SPEED/1000
ACTION-POINT-OFFSET	ACTION-POINT-OFFSET
BIT	BIT
CAS-RX-LOW-MAX	CAS-RX-LOW-MAX
COLD-START-ATTEMPTS	COLD-START-ATTEMPTS
CYCLE	CYCLE
CYCLE-COUNT-MAX	(NUMBER-OF-CYCLES - 1) or 63, if NUMBER-OF-CYCLES is not available

Target model property	Source model property
DYNAMIC-SLOT-IDLE-PHASE	DYNAMIC-SLOT-IDLE-PHASE
LISTEN-NOISE	LISTEN-NOISE
MACRO-PER-CYCLE	MACRO-PER-CYCLE
MACROTICK-DURATION	MACROTICK-DURATION
MAX-WITHOUT-CLOCK-CORRECTION-FATAL	MAX-WITHOUT-CLOCK-CORRECTION-FATAL
MAX-WITHOUT-CLOCK-CORRECTION-PASSIVE	MAX-WITHOUT-CLOCK-CORRECTION-PASSIVE
MINISLOT-ACTION-POINT-OFFSET	MINISLOT-ACTION-POINT-OFFSET
MINISLOT-DURATION	MINISLOT-DURATION
NETWORK-IDLE-TIME	NETWORK-IDLE-TIME
NETWORK-MANAGEMENT-VECTOR-LENGTH	NETWORK-MANAGEMENT-VECTOR-LENGTH
NUMBER-OF-MINISLOTS	NUMBER-OF-MINISLOTS
NUMBER-OF-STATIC-SLOTS	NUMBER-OF-STATIC-SLOTS
OFFSET-CORRECTION-START	OFFSET-CORRECTION-START
PAYLOAD-LENGTH-STATIC	PAYLOAD-LENGTH-STATIC
SAMPLE-CLOCK-PERIOD	SAMPLE-CLOCK-PERIOD
STATIC-SLOT-DURATION	STATIC-SLOT-DURATION
SYMBOL-WINDOW	SYMBOL-WINDOW
SYMBOL-WINDOW-ACTION-POINT-OFFSET	ACTION-POINT-OFFSET
SYNC-FRAME-ID-COUNT-MAX	SYNC-NODE-MAX
TRANSMISSION-START-SEQUENCE-DURATION	TRANSMISSION-START-SEQUENCE-DURATION
WAKEUP-RX-IDLE	WAKE-UP-SYMBOL-RX-IDLE
WAKEUP-RX-LOW	WAKE-UP-SYMBOL-RX-LOW
WAKEUP-RX-WINDOW	WAKE-UP-SYMBOL-RX-WINDOW
WAKEUP-TX-ACTIVE	WAKE-UP-SYMBOL-TX-LOW
WAKEUP-TX-IDLE	WAKE-UP-SYMBOL-TX-IDLE
No target model parameter is available.	MAX-FRAME-LENGTH
No target model parameter is available.	NM-REPEAT-MESSAGE-SUPPORT
No target model parameter is available.	BUS-GUARDIAN-ENABLE-PART
No target model parameter is available.	CAS-RX-LOW-MIN
No target model parameter is available.	MACRO-INITIAL-OFFSET
No target model parameter is available.	MAX-INITIALISATION-ERROR

Target model property	Source model property
No target model parameter is available.	MAX-PROPAGATION-DELAY
No target model parameter is available.	MIN-PROPAGATION-DELAY
No target model parameter is available.	OFFSET-CORRECTION-MAX
DETECT-NIT-ERROR	No source model parameter is available.
IGNORE-AFTER-TX	No source model parameter is available.
SAFETY-MARGIN	No source model parameter is available.

Changed or moved parameters

NM-NODE-DETECTION-ENABLED and NM-NODE-ID-ENABLED

The source model parameters NM-NODE-DETECTION-ENABLED and NM-NODE-ID-ENABLED are configured in the target model entity NM-ECU, see [Section 8.1.1.7.2, “NM-ECU”](#).

SPEED

In the target model, the parameter SPEED is given in kBit/s, whereas the source model provides it in Bit/s. If the source model parameter is not divisible by 1000 without rest, it cannot be upgraded at all.

8.1.1.8. FLEXRAY-COMMUNICATION-CONTROLLER

Source model entity: FLEXRAY-COMMUNICATION-CONTROLLER

Target model entity: FLEXRAY-COMMUNICATION-CONTROLLER/FLEXRAY-COMMUNICATION-CONTROLLER-CONDITIONAL

Target model property	Source model property
CATEGORY	CATEGORY
ACCEPTED-STARTUP-RANGE	ACCEPTED-STARTUP-RANGE
ALLOW-HALT-DUE-TO-CLOCK	ALLOW-HALT-DUE-TO-CLOCK
ALLOW-PASSIVE-TO-ACTIVE	ALLOW-PASSIVE-TO-ACTIVE
CLUSTER-DRIFT-DAMPING	CLUSTER-DRIFT-DAMPING
DECODING-CORRECTION	DECODING-CORRECTION
DELAY-COMPENSATION-A	DELAY-COMPENSATION-A
DELAY-COMPENSATION-B	DELAY-COMPENSATION-B
EXTERN-OFFSET-CORRECTION	EXTERN-OFFSET-CORRECTION
EXTERN-RATE-CORRECTION	EXTERN-RATE-CORRECTION
KEY-SLOT-ID	KEY-SLOT-ID

Target model property	Source model property
KEY-SLOT-ONLY-ENABLED	SINGLE-SLOT-ENABLED
KEY-SLOT-USED-FOR-START-UP	KEY-SLOT-USED-FOR-START-UP
KEY-SLOT-USED-FOR-SYNC	KEY-SLOT-USED-FOR-SYNC
LATEST-TX	LATEST-TX
LISTEN-TIMEOUT	LISTEN-TIMEOUT
MACRO-INITIAL-OFFSET-A	MACRO-INITIAL-OFFSET-A
MACRO-INITIAL-OFFSET-B	MACRO-INITIAL-OFFSET-B
MAXIMUM-DYNAMIC-PAYLOAD-LENGTH	MAXIMUM-DYNAMIC-PAYLOAD-LENGTH
MICRO-INITIAL-OFFSET-A	MICRO-INITIAL-OFFSET-A
MICRO-INITIAL-OFFSET-B	MICRO-INITIAL-OFFSET-B
MICRO-PER-CYCLE	MICRO-PER-CYCLE
MICROTICK-DURATION	MICROTICK-DURATION
OFFSET-CORRECTION-OUT	OFFSET-CORRECTION-OUT
RATE-CORRECTION-OUT	RATE-CORRECTION-OUT
SAMPLES-PER-MICROTICK	SAMPLES-PER-MICROTICK
WAKE-UP-PATTERN	WAKE-UP-PATTERN
No target model parameter is available.	DYNAMIC-SEGMENT-ENABLE
No target model parameter is available.	MAX-DRIFT
No target model parameter is available.	MICRO-PER-MACRO-NOM
No target model parameter is available.	START-UP-NODE
No target model parameter is available.	SYNC-SLOT
EXTERNAL-SYNC	No source model parameter is available.
FALL-BACK-INTERNAL	No source model parameter is available.
FLEXRAY-FIFOS	No source model parameter is available.
NM-VECTOR-EARLY-UPDATE	No source model parameter is available.
SECOND-KEY-SLOT-ID	No source model parameter is available.
TWO-KEY-SLOT-MODE	No source model parameter is available.

8.1.1.9. FLEXRAY-COMMUNICATION-CONNECTOR

Source model entity: FLEX-RAY-COMMUNICATION-CONNECTOR

Target model entity: FLEXRAY-COMMUNICATION-CONNECTOR

Target model property	Source model property
CATEGORY	CATEGORY
COMM-CONTROLLER-REF	COMM-CONTROLLER-REF
ECU-COMM-PORT-INSTANCES	ECU-COMM-PORT-INSTANCES
WAKE-UP-CHANNEL	WAKE-UP-CHANNEL

Changed or moved parameters

ECU-COMM-PORT-INSTANCES

In ECU-COMM-PORT-INSTANCES, the aggregated source model entities are upgraded to their target model counterparts: FRAME-PORT, I-PDU-PORT, SIGNAL-PORT.

CHANNEL-REF

The source model parameter CHANNEL-REF has been replaced by the target model parameter COMM-CONNECTORS/COMMUNICATION-CONNECTOR-REF-CONDITIONAL of the target model entity FLEXRAY-PHYSICAL-CHANNEL.

NM-ADDRESS and TP-ADDRESS

The source model parameters NM-ADDRESS and TP-ADDRESS are configured in the target model entities FLEXRAY-NM-NODE and TP-ADDRESS, see [Section 8.1.1.7.7, “FLEXRAY-NM-NODE”](#).

8.1.1.10. FLEXRAY-PHYSICAL-CHANNEL

Source model entities: PHYSICAL-CHANNEL, FLEX-RAY-COMMUNICATION-CONNECTOR

Target model entity: FLEXRAY-PHYSICAL-CHANNEL

Target model property	Source model property
CATEGORY	CATEGORY
FRAME-TRIGGERINGS	FRAME-TRIGGERINGSS
PDU-TRIGGERINGS	I-PDU-TRIGGERINGS
I-SIGNAL-TRIGGERINGS	I-SIGNAL-TRIGGERINGS
CHANNEL-NAME	CHANNEL-NAME

Changed or moved parameters

COMM-CONNECTORS

The target model parameter COMM-CONNECTORS is configured by retrieving all target model FLEX-RAY-COMMUNICATION-CONNECTOR entities that meet the following condition: The source model counterpart of

the FLEX-RAY-COMMUNICATION-CONNECTOR references the source model counterpart of the upgraded FLEXRAY-PHYSICAL-CHANNEL via CHANNEL-REF.

FLEX-RAY-TP-CHANNEL entities in TP-CHANNELS

The FLEX-RAY-TP-CHANNEL entities aggregated in TP-CHANNELS are upgraded to FLEXRAY-TP-CONNECTION-CONTROL entities in the target model, aggregated in FLEXRAY-TP-CONFIG, see [Section 8.1.1.5.1, "FLEXRAY-TP-CONFIG"](#).

8.1.1.11. LIN-CLUSTER

Source model entity: LIN-CLUSTER

Target model entity: LIN-CLUSTER/LIN-CLUSTER-CONDITIONAL

Target model property	Source model property
PHYSICAL-CHANNELS	PHYSICAL-CHANNELS
PROTOCOL-NAME	PROTOCOL-NAME
PROTOCOL-VERSION	PROTOCOL-VERSION
SPEED	SPEED/1000
No target model parameter is available.	MAX-FRAME-LENGTH
No target model parameter is available.	NM-REPEAT-MESSAGE-SUPPORT

Changed or moved parameters

SPEED

In the target model, the parameter SPEED is given in kbit/s, whereas the source model provides it in bit/s. If the source model parameter is not divisible by 1000 without rest, it cannot be upgraded at all.

LIN-SCHEDULE-TABLE entities of the source model parameter SCHEDULE-TABLES

The LIN-SCHEDULE-TABLE entities aggregated in the source model in SCHEDULE-TABLES are moved to the corresponding LIN-PHYSICAL-CHANNEL in the target model, see [Section 8.1.1.16, "LIN-PHYSICAL-CHANNEL"](#).

8.1.1.12. LIN-MASTER

Source model entity: LIN-MASTER

Target model entity: LIN-MASTER/LIN-MASTER-CONDITIONAL

Target model property	Source model property
CATEGORY	CATEGORY

Target model property	Source model property
TIME-BASE	TIME-BASE
TIME-BASE-JITTER	TIME-BASE-JITTER
PROTOCOL-VERSION	No source model parameter is available.

8.1.1.13. LIN-SLAVE

Source model entity: LIN-SLAVE

Target model entity: LIN-SLAVE/LIN-SLAVE-CONDITIONAL

Target model property	Source model property
CATEGORY	CATEGORY
CONFIGURED-NAD	CONFIGURED-NAD
LIN-ERROR-RESPONSE	LIN-ERROR-RESPONSE
PROTOCOL-VERSION	PROTOCOL-VERSION
SUPPLIER-ID	Source model parameters are available via LDF import only.
FUNCTION-ID	Source model parameters are available via LDF import only.
VARIANT-ID	Source model parameters are available via LDF import only.

Reasons for not configured target model parameters

SUPPLIER-ID, FUNCTION-ID, VARIANT-ID

The target model parameters SUPPLIER-ID, FUNCTION-ID, VARIANT-ID have no counterparts in the source model. Therefore they are only configured if the imported data originates from an LDF import, see [Section 7.10.2.7, “Importing from the LDF format”](#).

8.1.1.14. LIN-ERROR-RESPONSE

Source model entity: LIN-ERROR-RESPONSE

Target model entity: LIN-ERROR-RESPONSE

Target model property	Source model property
FRAME-TRIGGERING-REF	FRAME-TRIGGERING-REF

Target model property	Source model property
RESPONSE-ERROR-POSITION	RESPONSE-ERROR-POSITION

8.1.1.15. LIN-COMMUNICATION-CONNECTOR

Source model entity: COMMUNICATION-CONNECTOR

Target model entity: LIN-COMMUNICATION-CONNECTOR

Target model property	Source model property
CATEGORY	CATEGORY
COMM-CONTROLLER-REF	COMM-CONTROLLER-REF
ECU-COMM-PORT-INSTANCES	ECU-COMM-PORT-INSTANCES
LIN-CONFIGURABLE-FRAMES/LIN-CONFIGURABLE-FRAME	Source model parameters are available via LDF import only.

Reason for not configured target model parameters

LIN-CONFIGURABLE-FRAMES, and LIN-CONFIGURABLE-FRAME

The target model parameter LIN-CONFIGURABLE-FRAMES/LIN-CONFIGURABLE-FRAME are only configured, if the imported data originates from an LDF import, see [6] for details about the LDF file format.

Changed or moved parameters

ECU-COMM-PORT-INSTANCES

In ECU-COMM-PORT-INSTANCES, the aggregated source model entities are upgraded into their target model counterparts: FRAME-PORT, I-PDU-PORT, SIGNAL-PORT.

CHANNEL-REF

The source model parameter CHANNEL-REF has been replaced by the target model parameter COMM-CONNECTORS/COMMUNICATION-CONNECTOR-REF-CONDITIONAL of the target model entity LIN-PHYSICAL-CHANNEL.

TP-ADDRESS

The source model parameter TP-ADDRESS is configured in the target model entity TP-ADDRESS, see [Section 8.1.1.6.1, “LIN-TP-CONFIG”](#).

INITIAL-NAD

The target model parameter INITIAL-NAD is configured by taking the CONFIGURED-NAD parameter value of the LIN-SLAVE which is referenced via COMM-CONTROLLER-REF.

8.1.1.16. LIN-PHYSICAL-CHANNEL

Source model entities: PHYSICAL-CHANNEL, COMMUNICATION-CONNECTOR

Target model entity: LIN-PHYSICAL-CHANNEL

Target model property	Source model property
CATEGORY	CATEGORY
FRAME-TRIGGERINGS	FRAME-TRIGGERINGSS
PDU-TRIGGERINGS	I-PDU-TRIGGERINGS
I-SIGNAL-TRIGGERINGS	I-SIGNAL-TRIGGERINGS

Changed or moved parameters

COMM-CONNECTORS

The target model parameter COMM-CONNECTORS is configured by retrieving all target model COMMUNICATION-CONNECTOR entities that meet the following condition: The source model counterpart of the COMMUNICATION-CONNECTOR references the source model counterpart of the upgraded LIN-PHYSICAL-CHANNEL via CHANNEL-REF.

LIN-TP-CHANNEL entities aggregated in TP-CHANNELS

The LIN-TP-CHANNEL entities aggregated in TP-CHANNELS are upgraded to LIN-TP-CONFIGURATION entities in the target model, which are now aggregated in LIN-TP-CONFIG, see [Section 8.1.1.6.1, “LIN-TP-CONFIG”](#).

LIN-PHYSICAL-CHANNEL

The LIN-PHYSICAL-CHANNEL aggregates in its parameter SCHEDULE-TABLES all LIN-SCHEDULE-TABLE entities the source model entity LIN-CLUSTER aggregates in its corresponding parameter.

8.1.1.2. Communication

8.1.1.2.1. CAN-FRAME-TRIGGERING

Source model entity: CAN-FRAME-TRIGGERING

Target model entity: CAN-FRAME-TRIGGERING

Target model property	Source model property
CAN-ADDRESSING-MODE	CAN-ADDRESSING-MODE
CATEGORY	CATEGORY
IDENTIFIER	IDENTIFIER

Target model property	Source model property
FRAME-PORT-REFS/FRAME-PORT-REF	FRAME-PORT-REFS/FRAME-PORT-REF
FRAME-REF	FRAME-REF
PDU-TRIGGERINGS/PDU-TRIGGERING-REF-CONDITIONAL	I-PDU-TRIGGERING-REFS/I-PDU-TRIGGERING-REF

8.1.1.2.2. FLEXRAY-FRAME-TRIGGERING

Source model entity: FLEXRAY-FRAME-TRIGGERING

Target model entity: FLEXRAY-FRAME-TRIGGERING

Target model property	Source model property
ABSOLUTELY-SCHEDULED-TIMINGS/FLEXRAY-ABSOLUTELY-SCHEDULED-TIMING	ABSOLUTELY-SCHEDULED-TIMINGS/ABSOLUTELY-SCHEDULED-TIMING
CATEGORY	CATEGORY
PAYLOAD-PREAMBLE-INDICATOR	PAYLOAD-PREAMBLE-INDICATOR
FRAME-PORT-REFS/FRAME-PORT-REF	FRAME-PORT-REFS/FRAME-PORT-REF
FRAME-REF	FRAME-REF
PDU-TRIGGERINGS/PDU-TRIGGERING-REF-CONDITIONAL	I-PDU-TRIGGERING-REFS/I-PDU-TRIGGERING-REF
ALLOW-DYNAMIC-L-SDU-LENGTH	No source model parameter is available.
MESSAGE-ID	No source model parameter is available.

8.1.1.2.3. FLEXRAY-ABSOLUTELY-SCHEDULED-TIMING

Source model entity: ABSOLUTELY-SCHEDULED-TIMING

Target model entity: FLEXRAY-ABSOLUTELY-SCHEDULED-TIMING

Target model property	Source model property
COMMUNICATION-CYCLE/CYCLE-COUNTER	COMMUNICATION-CYCLE/CYCLE-COUNTER
COMMUNICATION-CYCLE/CYCLE-REPETITION/BASE-CYCLE	COMMUNICATION-CYCLE/CYCLE-REPETITION/BASE-CYCLE
COMMUNICATION-CYCLE/CYCLE-REPETITION/CYCLE-REPETITION	COMMUNICATION-CYCLE/CYCLE-REPETITION/CYCLE-REPETITION
SLOT-ID	SLOT-ID

8.1.1.2.4. LIN-FRAME-TRIGGERING

Source model entity: LIN-FRAME-TRIGGERING

Target model entity: LIN-FRAME-TRIGGERING

Target model property	Source model property
CATEGORY	CATEGORY
FRAME-PORT-REFS/FRAME-PORT-REF	FRAME-PORT-REFS/FRAME-PORT-REF
FRAME-REF	FRAME-REF
IDENTIFIER	IDENTIFIER
LIN-CHECKSUM	CHECKSUM-TYPE
PDU-TRIGGERINGS/PDU-TRIGGERING-REF-CONDITIONAL	I-PDU-TRIGGERING-REFS/I-PDU-TRIGGERING-REF

Changed or moved parameters

RELATIVELY-SCHEDULED-TIMING

The source model RELATIVELY-SCHEDULED-TIMING entities aggregated in LIN-FRAME-TRIGGERING/RELATIVELY-SCHEDULED-TIMINGS are now aggregated in the target model entity LIN-SCHEDULE-TABLE, see [Section 8.1.1.2.10, "LIN-SCHEDULE-TABLE"](#).

8.1.1.2.5. CAN-FRAME

Source model entity: FRAME

Target model entity: CAN-FRAME

Target model property	Source model property
CATEGORY	CATEGORY
FRAME-LENGTH	FRAME-LENGTH
PDU-TO-FRAME-MAPPINGS/PDU-TO-FRAME-MAPPING	PDU-TO-FRAME-MAPPINGS/PDU-TO-FRAME-MAPPING

Changed or moved parameters

FRAME-TRIGGERING

If the FRAME-TRIGGERING referencing FRAME is of type CAN-FRAME-TRIGGERING, the FRAME is upgraded to a CAN-FRAME in the target model.

If FRAME-TRIGGERINGS of different cluster types reference one and the same FRAME in the source model, a copy of the FRAME is created in the target model for each cluster type. If, for instance, one CAN-FRAME-TRIGGERING and one FLEXRAY-FRAME-TRIGGERING reference the same FRAME element in the source model, the FRAME element is upgraded to one CAN-FRAME and one FLEXRAY-FRAME in the target model.

8.1.1.2.6. FLEXRAY-FRAME

Source model entity: FRAME

Target model entity: FLEXRAY-FRAME

Target model property	Source model property
CATEGORY	CATEGORY
FRAME-LENGTH	FRAME-LENGTH
PDU-TO-FRAME-MAPPINGS / PDU-TO-FRAME-MAPPING	PDU-TO-FRAME-MAPPINGS / PDU-TO-FRAME-MAPPING

Changed or moved parameters

FRAME-TRIGGERING

If the FRAME-TRIGGERING referencing a FRAME is of type FLEXRAY-FRAME-TRIGGERING, the FRAME is upgraded to a FLEXRAY-FRAME in the target model.

If FRAME-TRIGGERINGS of different cluster types reference one and the same FRAME in the source model, a copy of the FRAME is created in the target model for each cluster type. If, for instance, one CAN-FRAME-TRIGGERING and one FLEXRAY-FRAME-TRIGGERING reference the same FRAME element in the source model, the FRAME element is upgraded to one CAN-FRAME and one FLEXRAY-FRAME in the target model.

8.1.1.2.7. LIN-UNCONDITIONAL-FRAME

Source model entity: FRAME

Target model entity: LIN-UNCONDITIONAL-FRAME

Target model property	Source model property
CATEGORY	CATEGORY
FRAME-LENGTH	FRAME-LENGTH
PDU-TO-FRAME-MAPPINGS / PDU-TO-FRAME-MAPPING	PDU-TO-FRAME-MAPPINGS / PDU-TO-FRAME-MAPPING

Changed or moved parameters

FRAME-TRIGGERING

If the FRAME-TRIGGERING referencing a FRAME is of type LIN-FRAME-TRIGGERING, the FRAME is upgraded to a LIN-UNCONDITIONAL-FRAME in the target model.

If FRAME-TRIGGERINGS of different cluster types reference one and the same FRAME in the source model, a copy of the FRAME is created in the target model for each cluster type. If, for instance, one CAN-FRAME-TRIGGERING and one FLEXRAY-FRAME-TRIGGERING reference the same FRAME element in the source model, the FRAME element is upgraded to one CAN-FRAME and one FLEXRAY-FRAME in the target model.

8.1.1.2.8. LIN-SPORADIC-FRAME

Source model entity: SUBSTITUTION-FRAME

Target model entity: LIN-SPORADIC-FRAME

Target model property	Source model property
CATEGORY	CATEGORY
FRAME-LENGTH	FRAME-LENGTH
PDU-TO-FRAME-MAPPINGS/PDU-TO-FRAME-MAPPING	PDU-TO-FRAME-MAPPINGS/PDU-TO-FRAME-MAPPING
SUBSTITUTED-FRAME-REFS/SUBSTITUTED-FRAME-REF	SUBSTITUTED-FRAME-REFS/SUBSTITUTED-FRAME-REF
LIN-EVENT-TRIGGERED-FRAME/COLLISION-RESOLVING-SCHEDULE-REF	No source model parameter is available.

NOTE



The parameters FRAME-LENGTH, and PDU-TO-FRAME-MAPPINGS/PDU-TO-FRAME-MAPPING might not bear useful information

The parameters FRAME-LENGTH, and PDU-TO-FRAME-MAPPINGS/PDU-TO-FRAME-MAPPING only exist due to inheritance peculiarities of the source and target models. In the context of a LIN-SPORADIC-FRAME or SUBSTITUTION-FRAME, they do not bear useful information.

Changed or moved parameters

SUBSTITUTION-FRAME

If the SUBSTITUTION-FRAME in the source model has its parameter SUBSTITUTION-TYPE set to SPORADIC, it is upgraded to a LIN-SPORADIC-FRAME in the target model.

8.1.1.2.9. LIN-EVENT-TRIGGERED-FRAME

Source model entity: SUBSTITUTION-FRAME

Target model entity: LIN-EVENT-TRIGGERED-FRAME

Target model property	Source model property
CATEGORY	CATEGORY
FRAME-LENGTH	FRAME-LENGTH
PDU-TO-FRAME-MAPPINGS/PDU-TO-FRAME-MAPPING	PDU-TO-FRAME-MAPPINGS/PDU-TO-FRAME-MAPPING
LIN-UNCONDITIONAL-FRAME-REFS/LIN-UNCONDITIONAL-FRAME-REF	SUBSTITUTED-FRAME-REFS/SUBSTITUTED-FRAME-REF

NOTE



The parameters FRAME-LENGTH, and PDU-TO-FRAME-MAPPINGS/PDU-TO-FRAME-MAPPING might not contain useful information

The parameters FRAME-LENGTH, and PDU-TO-FRAME-MAPPINGS/PDU-TO-FRAME-MAPPING only exist due to inheritance peculiarities of the source and target models. In the context of a LIN-EVENT-TRIGGERED-FRAME or SUBSTITUTION-FRAME, they do not bear useful information.

Changed or moved parameters

SUBSTITUTION-FRAME

If the SUBSTITUTION-FRAME in the source model has its parameter SUBSTITUTION-TYPE set to EVENT-TRIGGERED, it is upgraded to a LIN-EVENT-TRIGGERED-FRAME in the target model.

8.1.1.2.10. LIN-SCHEDULE-TABLE

Source model entity: LIN-SCHEDULE-TABLE

Target model entity: LIN-SCHEDULE-TABLE

Target model property	Source model property
CATEGORY	CATEGORY
RUN-MODE	RUN-MODE
RESUME-POSITION	No source model parameter is available.
No target model parameter is available.	PRIORITY

Changed or moved parameters

LIN-SCHEDULE-TABLE/TABLE-ENTRYS

The target model parameter LIN-SCHEDULE-TABLE/TABLE-ENTRYS aggregates all source model RELATIVELY-SCHEDULED-TIMING entities that reference the corresponding LIN-SCHEDULE-TABLE in the source model.

8.1.1.2.11. ASSIGN-FRAME-ID

Source model entity: ASSIGN-FRAME-ID-TIMING

Target model entity: ASSIGN-FRAME-ID

Target model property	Source model property
ASSIGNED-CONTROLLER-REF	ASSIGNED-CONTROLLER-REF
DELAY	DELAY
POSITION-IN-TABLE	POSITION-IN-TABLE
ASSIGNED-FRAME-TRIGGERING-REF	ASSIGNED-FRAME-TRIGGERING-REF

8.1.1.2.12. UNASSIGN-FRAME-ID

Source model entity: UNASSIGN-FRAME-ID-TIMING

Target model entity: UNASSIGN-FRAME-ID

Target model property	Source model property
ASSIGNED-CONTROLLER-REF	ASSIGNED-CONTROLLER-REF
DELAY	DELAY
POSITION-IN-TABLE	POSITION-IN-TABLE
UNASSIGNED-FRAME-TRIGGERING-REF	UNASSIGNED-FRAME-TRIGGERING-REF

8.1.1.2.13. ASSIGN-NAD

Source model entity: ASSIGN-NAD-TIMING

Target model entity: ASSIGN-NAD

Target model property	Source model property
ASSIGNED-CONTROLLER-REF	ASSIGNED-CONTROLLER-REF
DELAY	DELAY

Target model property	Source model property
POSITION-IN-TABLE	POSITION-IN-TABLE
NEW-NAD	NEW-NAD

8.1.1.2.14. FREE-FORMAT

Source model entity: DATA-TIMING

Target model entity: FREE-FORMAT

Target model property	Source model property
ASSIGNED-CONTROLLER-REF	ASSIGNED-CONTROLLER-REF
DELAY	DELAY
POSITION-IN-TABLE	POSITION-IN-TABLE
BYTE-VALUES/BYTE-VALUE	FREE-FORMAT/BYTE-VALUES/BYTE-VALUE

8.1.1.2.15. PDU-TRIGGERING

Source model entity: I-PDU-TRIGGERING

Target model entity: PDU-TRIGGERING

Target model property	Source model property
CATEGORY	CATEGORY
I-PDU-PORT-REFS/I-PDU-PORT-REF	I-PDU-PORT-REFS/I-PDU-PORT-REF
I-PDU-REF	I-PDU-REF
I-SIGNAL-TRIGGERINGS/I-SIGNAL-TRIGGING-REF-CONDITIONAL	I-SIGNAL-TRIGGERING-REFS/I-SIGNAL-TRIGGING-REF
No target model parameter is available.	TIMING-REQUIREMENT

8.1.1.2.16. I-SIGNAL-TRIGGERING

Source model entity: I-SIGNAL-TRIGGERING

Target model entity: I-SIGNAL-TRIGGERING

Target model property	Source model property
CATEGORY	CATEGORY
I-SIGNAL-GROUP-REF	SIGNAL-REF

Target model property	Source model property
I-SIGNAL-REF	SIGNAL-REF
I-SIGNAL-PORT-REFS/I-SIGNAL-PORT-REF	I-SIGNAL-PORT-REFS/I-SIGNAL-PORT-REF

Changed or moved parameters

SIGNAL-REF

The source model parameter SIGNAL-REF is upgraded to the target parameter I-SIGNAL-REF, if the referenced source model entity represents an I-SIGNAL that references in turn a SYSTEM-SIGNAL. If the referenced I-SIGNAL references a SYSTEM-SIGNAL-GROUP, the parameter is upgraded to I-SIGNAL-GROUP-REF.

8.1.1.2.17. DCM-I-PDU

Source model entity: DCM-I-PDU

Target model entity: DCM-I-PDU

Target model property	Source model property
CATEGORY	CATEGORY
LENGTH	LENGTH
DIAG-PDU-TYPE	No source model parameter is available.

Changed or moved parameters

LENGTH

During the conversion process, the parameter LENGTH given in units of bits in the source model is converted to bytes as required by the target model.

8.1.1.2.18. I-SIGNAL-I-PDU

Source model entity: SIGNAL-I-PDU

Target model entity: I-SIGNAL-I-PDU

Target model property	Source model property
CATEGORY	CATEGORY
I-PDU-TIMING-SPECIFICATIONS/I-PDU-TIMING	I-PDU-TIMING-SPECIFICATION

Target model property	Source model property
I-SIGNAL-TO-PDU-MAPPINGS/I-SIGNAL-TO-I-PDU-MAPPING	SIGNAL-TO-PDU-MAPPINGS/I-SIGNAL-TO-I-PDU-MAPPING
LENGTH	LENGTH
UNUSED-BIT-PATTERN	UNUSED-BIT-PATTERN
PDU-COUNTERS/SIGNAL-I-PDU-COUNTER	No source model parameter is available.
PDU-REPLICATIONS/SIGNAL-I-PDU-REPLICATION	No source model parameter is available.

Changed or moved parameters

LENGTH

During the conversion process, the parameter LENGTH given in units of bits in the source model is converted to bytes as required by the target model.

8.1.1.2.19. MULTIPLEXED-I-PDU

Source model entity: MULTIPLEXED-I-PDU

Target model entity: MULTIPLEXED-I-PDU

Target model property	Source model property
CATEGORY	CATEGORY
DYNAMIC-PARTS/DYNAMIC-PART	DYNAMIC-PART
LENGTH	LENGTH
SELECTOR-FIELD-BYTE-ORDER	SELECTOR-FIELD-BYTE-ORDER
SELECTOR-FIELD-LENGTH	SELECTOR-FIELD-LENGTH
SELECTOR-FIELD-START-POSITION	SELECTOR-FIELD-START-POSITION
STATIC-PARTS/STATIC-PART	STATIC-PART
TRIGGER-MODE	TRIGGER-MODE
UNUSED-BIT-PATTERN	UNUSED-BIT-PATTERN

DYNAMIC-PART and STATIC-PART: differences between source model and the EB tresos Studio system model

The system model provided by EB tresos Studio differs from the source model in such a way that it allows multiple DYNAMIC-PART elements per MULTIPLEXED-I-PDU. In the upgrade process, these multi-

ple DYNAMIC-PART elements are aggregated into one single DYNAMIC-PARTS/DYNAMIC-PART element, which in turn aggregates multiple SEGMENT-POSITION and multiple DYNAMIC-PART-ALTERNATIVES/DYNAMIC-PART-ALTERNATIVE parameters.

For each unique selector field code found in any DYNAMIC-PART, one DYNAMIC-PART-ALTERNATIVE element is created, which contains the code in its SELECTOR-FIELD-CODE parameter. In the target model, DYNAMIC-PART-ALTERNATIVE/I-PDU-REF references the I-PDU that this selector field code is addressing. DYNAMIC-PART-ALTERNATIVE/INITIAL-DYNAMIC-PART is set to TRUE for the selector field code configured to represent the initially selected DYNAMIC-PART.

Moreover, the system model provided by EB tresos Studio allows multiple STATIC-PART elements per MULTIPLEXED-I-PDU. In the upgrade process, these multiple STATIC-PART elements are aggregated to one single STATIC-PARTS/STATIC-PART element, which in turn aggregates multiple SEGMENT-POSITION parameters.

Changed or moved parameters

LENGTH

During the conversion process, the parameter LENGTH given in units of bits in the source model is converted to bytes as required by the target model.

8.1.1.2.20. DYNAMIC-PART

Source model entity: DYNAMIC-PART

Target model entity: DYNAMIC-PART

Target model property	Source model property
DYNAMIC-PART-ALTERNATIVES/DYNAMIC-PART-ALTERNATIVE	DYNAMIC-PART-ALTERNATIVES/DYNAMIC-PART-ALTERNATIVE
SEGMENT-POSITIONS/SEGMENT-POSITION	SEGMENT-POSITIONS/SEGMENT-POSITION

NOTE

The source model differs from the EB tresos Studio system model

The EB tresos Studio internal system model differs from the source model in how MULTIPLEXED-I-PDUS are stored, see [Section 8.1.1.2.19, “MULTIPLEXED-I-PDU”](#).



8.1.1.2.21. DYNAMIC-PART-ALTERNATIVE

Source model entity: DYNAMIC-PART-ALTERNATIVE

Target model entity: DYNAMIC-PART-ALTERNATIVE

Target model property	Source model property
I-PDU-REF	I-PDU-REF
SELECTOR-FIELD-CODE	SELECTOR-FIELD-CODE
INITIAL-DYNAMIC-PART	INITIAL-DYNAMIC-PART

NOTE**The source model differs from the EB tresos Studio system model**

The EB tresos Studio internal system model differs from the source model in how MULTIPLEXED-I-PDUS are stored, see [Section 8.1.1.2.19, “MULTIPLEXED-I-PDU”](#).

8.1.1.2.22. STATIC-PART

Source model entity: STATIC-PART

Target model entity: STATIC-PART

Target model property	Source model property
I-PDU-REF	I-PDU-REF
SEGMENT-POSITIONS/SEGMENT-POSITION	SEGMENT-POSITIONS/SEGMENT-POSITION

NOTE**The source model differs from the EB tresos Studio system model**

The EB tresos Studio internal system model differs from the source model in how MULTIPLEXED-I-PDUS are stored, see [Section 8.1.1.2.19, “MULTIPLEXED-I-PDU”](#).

8.1.1.2.23. SEGMENT-POSITION

Source model entity: SEGMENT-POSITION

Target model entity: SEGMENT-POSITION

Target model property	Source model property
SEGMENT-BYTE-ORDER	SEGMENT-BYTE-ORDER
SEGMENT-LENGTH	SEGMENT-LENGTH
SEGMENT-POSITION	SEGMENT-POSITION

NOTE**The source model differs from the EB tresos Studio system model**

The EB tresos Studio internal system model differs from the source model in how MULTIPLEXED-I-PDUS are stored, see [Section 8.1.1.2.19, “MULTIPLEXED-I-PDU”](#).

8.1.1.2.24. N-PDU

Source model entity: N-PDU

Target model entity: N-PDU

Target model property	Source model property
CATEGORY	CATEGORY
LENGTH	LENGTH

Changed or moved parameters

LENGTH

During the upgrade process, the parameter LENGTH given in units of bits in the source model is converted to bytes as required by the target model.

8.1.1.2.25. NM-PDU

Source model entity: NM-PDU

Target model entity: NM-PDU

Target model property	Source model property
CATEGORY	CATEGORY
LENGTH	LENGTH
UNUSED-BIT-PATTERN	UNUSED-BIT-PATTERN

If a Fibex file has been imported into the EB tresos Studio source model, the following target model parameters are set additionally:

Target model property	Fibex model property
NM-DATA-INFORMATION	NM-PDU-USAGE/CONTAINS-DATA

Target model property	Fibex model property
NM-VOTE-INFORMATION	NM-PDU-USAGE/CONTAINS-VOTE
I-SIGNAL-TO-I-PDU-MAPPINGS/I-SIGNAL-TO-I-PDU-MAPPING	PDU/SIGNAL-INSTANCES

Changed or moved parameters

LENGTH

During the conversion process, the parameter LENGTH given in units of bits in the source model is converted to bytes as required by the target model.

NM-CBV-POSITION, NM-NID-POSITION, and NM-USER-DATA-LENGTH

The source model parameters NM-CBV-POSITION, NM-NID-POSITION, and NM-USER-DATA-LENGTH are used to configure the target model entity CAN-NM-CLUSTER, see [Section 8.1.1.7.3, “CAN-NM-CLUSTER”](#).

8.1.1.2.26. PDU-TO-FRAME-MAPPING

Source model entity: PDU-TO-FRAME-MAPPING

Target model entity: PDU-TO-FRAME-MAPPING

Target model property	Source model property
CATEGORY	CATEGORY
PACKING-BYTE-ORDER	PACKING-BYTE-ORDER
PDU-REF	PDU-REF
START-POSITION	START-POSITION
UPDATE-INDICATION-BIT-POSITION	UPDATE-INDICATION-BIT-POSITION

8.1.1.2.27. I-PDU-TIMING

Source model entity: I-PDU-TIMING

Target model entity: I-PDU-TIMING

Target model property	Source model property
MINIMUM-DELAY	MINIMUM-DELAY
TRANSMISSION-MODE-DECLARATION	TRANSMISSION-MODE-DECLARATION

Target model property	Source model property
No target model parameter is available.	REQUEST-CONTROLLED-TIMING

Changed or moved parameters

I-PDU-TIMING

In the target model, one TRANSMISSION-MODE-DECLARATION is added to the I-PDU-TIMING, even if no counterpart exists in the source model. This newly created TRANSMISSION-MODE-DECLARATION entity aggregates via TRANSMISSION-MODE-TRUE-TIMING the parameters EVENT-CONTROLLED-TIMING and CYCLIC-TIMING, which are aggregated directly in the source model I-PDU-TIMING.

8.1.1.2.28. I-SIGNAL-I-PDU-GROUP

Source model entity: I-PDU-GROUP

Target model entity: I-SIGNAL-I-PDU-GROUP

Target model property	Source model property
COMMUNICATION-DIRECTION	COMMUNICATION-DIRECTION
COMMUNICATION-MODE	COMMUNICATION-MODE
CONTAINED-I-SIGNAL-I-PDU-GROUP-REFS/CONTAINED-I-SIGNAL-I-PDU-GROUP-REF	CONTAINED-I-PDU-GROUPS-REFS/CONTAINED-I-PDU-GROUPS-REF
I-SIGNAL-I-PDUS/I-SIGNAL-I-PDU-REF-CONDITIONAL	I-PDU-REFS/I-PDU-REF

8.1.1.2.29. I-SIGNAL-TO-I-PDU-MAPPING

Source model entity: I-SIGNAL-TO-I-PDU-MAPPING

Target model entity: I-SIGNAL-TO-I-PDU-MAPPING

Target model property	Source model property
CATEGORY	CATEGORY
I-SIGNAL-GROUP-REF	SIGNAL-REF
I-SIGNAL-REF	SIGNAL-REF
PACKING-BYTE-ORDER	PACKING-BYTE-ORDER
START-POSITION	START-POSITION
TRANSFER-PROPERTY	TRANSFER-PROPERTY

Target model property	Source model property
UPDATE-INDICATION-BIT-POSITION	UPDATE-INDICATION-BIT-POSITION

Changed or moved parameters

SIGNAL-REF

The source model parameter SIGNAL-REF is upgraded to the target model parameter I-SIGNAL-REF, if the referenced source model entity represents an I-SIGNAL that in turn references a SYSTEM-SIGNAL. If the referenced I-SIGNAL references a SYSTEM-SIGNAL-GROUP, the parameter is upgraded to I-SIGNAL-GROUP-REF.

TRANSFER-PROPERTY

If a Fibex 3.x file has been imported into the EB tresos Studio source model, TRANSFER-PROPERTY is set according to its representation in Fibex files. Fibex files currently only support TRIGGERED-ON-CHANGE as parameter value, or no value at all.

8.1.1.2.30. I-SIGNAL

Source model entity: I-SIGNAL

Target model entity: I-SIGNAL

Target model property	Source model property
CATEGORY	CATEGORY
SYSTEM-SIGNAL-REF	SYSTEM-SIGNAL-REF

Changed or moved parameters

I-SIGNAL

If the source model I-SIGNAL references a SYSTEM-SIGNAL via SYSTEM-SIGNAL-REF, an I-SIGNAL is created in the target model.

LENGTH

The target model parameter LENGTH is retrieved from the SYSTEM-SIGNAL referenced via SYSTEM-SIGNAL-REF in the source model.

DATA-TYPE-POLICY

The value of the target model parameter DATA-TYPE-POLICY is set to the value LEGACY.

INIT-VALUE

The target model parameter INIT-VALUE is configured by retrieving the SYSTEM-SIGNAL of the source model I-SIGNAL, and then obtaining the entity referenced via INIT-VALUE-REF, which can be one of the following types:

- ▶ ARRAY-SPECIFICATION
- ▶ BOOLEAN-LITERAL
- ▶ CHAR-LITERAL
- ▶ CONSTANT-REFERENCE
- ▶ INTEGER-LITERAL
- ▶ OPAQUE-LITERAL
- ▶ REAL-LITERAL
- ▶ RECORD-SPECIFICATION, or
- ▶ STRING-LITERAL

If the entity is of the type STRING-LITERAL, the content of its VALUE parameter is upgraded to a TEXT-VALUE-SPECIFICATION. In all other cases, the content of VALUE is upgraded to a NUMERICAL-VALUE-SPECIFICATION.

NETWORK-REPRESENTATION-PROPS

For the configuration of the target model parameter NETWORK-REPRESENTATION-PROPS, the data type of the source SYSTEM-SIGNAL, referenced via DATA-TYPE-REF is retrieved. If this data type contains a SW-DATA-DEF-PROPS/INVALID-VALUE parameter, a NETWORK-REPRESENTATION-PROPS parameter is created for the target model entity I-SIGNAL. It contains in SW-DATA-DEF-PROPS-CONDITIONAL/IMPLEMENTATION-DATA-TYPE-REF and SW-DATA-DEF-PROPS-CONDITIONAL/BASE-TYPE-REF references to the upgraded data type entities in the target model. Moreover, the parameter INVALID-VALUE is configured. If the source model SW-DATA-DEF-PROPS/INVALID-VALUE contains a STRING-LITERAL, this is upgraded to an TEXT-VALUE-SPECIFICATION. In all other cases it is upgraded to a NUMERICAL-VALUE-SPECIFICATION.

8.1.1.2.31. I-SIGNAL-GROUP

Source model entity: I-SIGNAL

Target model entity: I-SIGNAL-GROUP

Target model property	Source model property
CATEGORY	CATEGORY
SYSTEM-SIGNAL-REF	SYSTEM-SIGNAL-REF

Changed or moved parameters

I-SIGNAL

If the source model I-SIGNAL references a SYSTEM-SIGNAL-GROUP via SYSTEM-SIGNAL-REF, an I-SIGNAL-GROUP is created in the target model.

I-SIGNAL-GROUP

The parameters INIT-VALUE and NETWORK-REPRESENTATION-PROPS are not configured for target model I-SIGNAL-GROUPS.

I-SIGNAL-REFS/I-SIGNAL-REF

I-SIGNAL-REFS/I-SIGNAL-REF references all target model I-SIGNALS belonging to the I-SIGNAL-GROUP. An I-SIGNAL belongs to an I-SIGNAL-GROUP, if the SYSTEM-SIGNAL of the I-SIGNAL also belongs to the SYSTEM-SIGNAL-GROUP of the I-SIGNAL-GROUP. The SYSTEM-SIGNAL of the I-SIGNAL and the SYSTEM-SIGNAL-GROUP of the I-SIGNAL-GROUP are both referenced via SYSTEM-SIGNAL-REF. A SYSTEM-SIGNAL-GROUP references its SYSTEM-SIGNALS in SYSTEM-SIGNAL-REFS/SYSTEM-SIGNAL-REF.

8.1.1.2.32. SYSTEM-SIGNAL

Source model entity: SYSTEM-SIGNAL

Target model entity: SYSTEM-SIGNAL

Target model property	Source model property
CATEGORY	CATEGORY

Changed or moved parameters

DATA-TYPE-REF

The source model parameter DATA-TYPE-REF is configured in the target model I-SIGNAL entity parameter NETWORK-REPRESENTATION-PROPS/SW-DATA-DEF-PROPS-CONDITIONAL/IMPLEMENTATION-DATA-TYPE-REF, and NETWORK-REPRESENTATION-PROPS/SW-DATA-DEF-PROPS-CONDITIONAL/BASE-TYPE-REF, see [Section 8.1.1.2.30, "I-SIGNAL"](#).

LENGTH

The source model parameter LENGTH is configured in the target model I-SIGNAL entity parameter LENGTH, see [Section 8.1.1.2.30, "I-SIGNAL"](#).

DYNAMIC-LENGTH

The value of the target model parameter DYNAMIC-LENGTH is set to the value false.

8.1.1.2.33. I-SIGNAL-PORT

Source model entity: SIGNAL-PORT

Target model entity: I-SIGNAL-PORT

Target model property	Source model property
CATEGORY	CATEGORY
COMMUNICATION-DIRECTION	COMMUNICATION-DIRECTION
TIMEOUT	TIMEOUT

8.1.1.2.34. FRAME-PORT

Source model entity: FRAME-PORT

Target model entity: FRAME-PORT

Target model property	Source model property
CATEGORY	CATEGORY
COMMUNICATION-DIRECTION	COMMUNICATION-DIRECTION

8.1.1.2.35. I-PDU-PORT

Source model entity: I-PDU-PORT

Target model entity: I-PDU-PORT

Target model property	Source model property
CATEGORY	CATEGORY
COMMUNICATION-DIRECTION	COMMUNICATION-DIRECTION

8.1.1.2.36. TRANSMISSION-MODE-DECLARATION

Source model entity: TRANSMISSION-MODE-DECLARATION

Target model entity: TRANSMISSION-MODE-DECLARATION

Target model property	Source model property
TRANSMISSION-MODE-CONDITIONS / TRANSMISSION-MODE-CONDITION	TRANSMISSION-MODE-CONDITIONS / TRANSMISSION-MODE-CONDITION
TRANSMISSION-MODE-FALSE-TIMING	TRANSMISSION-MODE-FALSE-TIMING

Changed or moved parameters

TRANSMISSION-MODE-DECLARATION **and an** TRANSMISSION-MODE-TRUE-TIMING

A TRANSMISSION-MODE-DECLARATION **and a** TRANSMISSION-MODE-TRUE-TIMING are always added to the aggregating I-PDU-TIMING, see [Section 8.1.1.2.27, “I-PDU-TIMING”](#) for details.

If the TRANSMISSION-MODE-DECLARATION has been newly created, one TRANSMISSION-MODE-CONDITIONS/TRANSMISSION-MODE-CONDITION is added, containing a DATA-FILTER of type ALWAYS.

8.1.1.2.37. TRANSMISSION-MODE-CONDITION

Source model entity: TRANSMISSION-MODE-CONDITION

Target model entity: TRANSMISSION-MODE-CONDITION

Target model property	Source model property
DATA-FILTER	DATA-FILTER
I-SIGNAL-IN-I-PDU-REF	SIGNAL-IN-I-PDU-REF

8.1.1.2.38. TRANSMISSION-MODE-FALSE-TIMING

Source model entity: TRANSMISSION-MODE-FALSE-TIMING

Target model entity: TRANSMISSION-MODE-TIMING

Target model property	Source model property
CYCLIC-TIMING	CYCLIC-TIMING
EVENT-CONTROLLED-TIMING	EVENT-CONTROLLED-TIMING

8.1.1.2.39. EVENT-CONTROLLED-TIMING

Source model entity: EVENT-CONTROLLED-TIMING

Target model entity: EVENT-CONTROLLED-TIMING

Target model property	Source model property
NUMBER-OF-REPETITIONS	NUMBER-OF-REPEATS
REPETITION-PERIOD	REPETITION-PERIOD

8.1.1.2.40. CYCLIC-TIMING

Source model entity: CYCLIC-TIMING

Target model entity: CYCLIC-TIMING

Target model property	Source model property
TIME-OFFSET	STARTING-TIME
TIME-PERIOD	REPEATING-TIME
No target model parameter is available.	FINAL-REPETITIONS

8.1.1.2.41. TIME-RANGE-TYPE

Source model entity: TIME-RANGE-TYPE

Target model entity: TIME-RANGE-TYPE

Target model property	Source model property
TOLERANCE/ABSOLUTE-TOLERANCE	TOLERANCE/ABSOLUTE-TOLERANCE
TOLERANCE/RELATIVE-TOLERANCE	TOLERANCE/RELATIVE-TOLERANCE

8.1.1.2.42. ABSOLUTE-TOLERANCE

Source model entity: ABSOLUTE-TOLERANCE

Target model entity: ABSOLUTE-TOLERANCE

Target model property	Source model property
ABSOLUTE	ABSOLUTE

8.1.1.2.43. RELATIVE-TOLERANCE

Source model entity: RELATIVE-TOLERANCE

Target model entity: RELATIVE-TOLERANCE

Target model property	Source model property
RELATIVE	RELATIVE

8.1.1.3. Gateway

8.1.1.3.1. FRAME-MAPPING

Source model entity: FRAME-MAPPING

Target model entity: FRAME-MAPPING

Target model property	Source model property
SOURCE-FRAME-REF	SOURCE-FRAME-REF
TARGET-FRAME-REF	TARGET-FRAME-REF

8.1.1.3.2. I-PDU-MAPPING

Source model entity: I-PDU-MAPPING

Target model entity: I-PDU-MAPPING

Target model property	Source model property
SOURCE-I-PDU-REF	SOURCE-I-PDU/SOURCE-I-PDU-REF
TARGET-I-PDU/TARGET-I-PDU-REF	TARGET-I-PDU/TARGET-I-PDU-REF
No target model parameter is available.	SOURCE-I-PDU-REF/PDU-MAPPINGS-DE-FAULT-VALUE

8.1.1.3.3. TARGET-I-PDU-REF

Source model entity: TARGET-I-PDU-REF

Target model entity: TARGET-I-PDU-REF

Target model property	Source model property
DEFAULT-VALUE	DEFAULT-VALUE
TARGET-I-PDU-REF	TARGET-I-PDU-REF

8.1.1.3.4. PDU-MAPPING-DEFAULT-VALUE

Source model entity: PDU-MAPPING-DEFAULT-VALUE

Target model entity: PDU-MAPPING-DEFAULT-VALUE

Target model property	Source model property
DEFAULT-VALUE-ELEMENTS/DEFAULT-VALUE-ELEMENT	DEFAULT-VALUE-ELEMENTS/DEFAULT-VALUE-ELEMENT

8.1.1.3.5. DEFAULT-VALUE-ELEMENT

Source model entity: DEFAULT-VALUE-ELEMENT

Target model entity: DEFAULT-VALUE-ELEMENT

Target model property	Source model property
ELEMENT-BYTE-VALUE	ELEMENT-BYTE-VALUE
ELEMENT-POSITION	ELEMENT-POSITION

8.1.1.3.6. I-SIGNAL-MAPPING

Source model entity: SIGNAL-MAPPING

Target model entity: I-SIGNAL-MAPPING

Target model property	Source model property
SOURCE-SIGNAL-REF	SOURCE-SIGNAL-REF
TARGET-SIGNAL-REF	TARGET-SIGNAL-REF

8.1.1.4. CAN Transport Layer

8.1.1.4.1. CAN-TP-CONFIG

For each CAN-CLUSTER in the source model that has at least one CAN-TP-CONNECTION-CHANNEL aggregated in its PHYSICAL-CHANNEL, one CAN-TP-CONFIG element is created in the target model. It resides in the same AR-PACKAGE as the related CAN-CLUSTER.

New parameters in the target model

CAN-TP-ECU

For each ECU-INSTANCE in the source model that is attached to the PHYSICAL-CHANNEL, one CAN-TP-ECU element is created in the target model. It references the target model counterpart of the source model

ECU-INSTANCE in ECU-INSTANCE-REF. All created CAN-TP-ECU elements are aggregated in CAN-TP-CONFIG/TP-ECUS.

CAN-TP-ECU/CYCLE-TIME-MAIN-FUNCTION

If a Fibex file has been imported into the EB tresos Studio source model, CAN-TP-ECU/CYCLE-TIME-MAIN-FUNCTION is additionally set in the target model using the Fibex parameter TP-ECU/CYCLE-TIME-MAIN-FUNCTION.

CAN-TP-NODE

For each source model COMMUNICATION-CONNECTOR attached to a PHYSICAL-CHANNEL of a CAN-CLUSTER for which a CAN-TP-CONFIG has been created in the target model, one CAN-TP-NODE is added. In CONNECTOR-REF, CAN-TP-NODE references the COMMUNICATION-CONNECTOR counterpart in the target model. CAN-TP-NODE is aggregated in CAN-TP-CONFIG/TP-NODES. Its parameter ST-MIN is set to the maximum MINIMUM-SEPARATION-TIME value of all CAN-TP-CONNECTIONS the CAN-TP-NODE is receiving.

If a Fibex file has been imported into the EB tresos Studio source model, the following additional CAN-TP-NODE parameter values are set:

Target model property	Fibex model property
MAX-FC-WAIT	TP-NODE/MAX-FC-WAIT
ST-MIN	TP-NODE/STMIN
TIMEOUT-AR	TP-NODE/TIMEOUT-AR
TIMEOUT-AS	TP-NODE/TIMEOUT-AS

CAN-TP-ADDRESS

An additional entity CAN-TP-ADDRESS is created, which the CAN-TP-NODE references via TP-ADDRESS-REF. The CAN-TP-ADDRESS contains in its TP-ADDRESS parameter the value obtained from the COMMUNICATION-CONNECTOR parameter TP-ADDRESS. It is aggregated in CAN-TP-CONFIG/TP-ADDRESSS.

Changed or moved parameters

TP-NODE/STMIN

TP-NODE/STMIN is upgraded from its representation according to the ISO 15765-2 specification as given in Fibex to seconds as required by the target model.

8.1.1.4.2. CAN-TP-CONNECTION

Source model entity: CAN-TP-CONNECTION-CHANNEL

Target model entity: CAN-TP-CONNECTION

Target model property	Source model property
ADDRESSING-FORMAT	ADDRESSING-FORMAT
DATA-PDU-REF	DATA-PDU-REF
FLOW-CONTROL-PDU-REF	FLOW-CONTROL-PDU-REF
MAX-BLOCK-SIZE	BLOCK-SIZE
PADDING-ACTIVATION	PADDING-ACTIVATION
TP-SDU-REF	TP-SDU-REF
Not upgraded.	MULTICAST-ADDRESSING
MULTICAST-REF	No source model parameter is available.

The created CAN-TP-CONNECTION is aggregated in CAN-TP-CONFIG/TP-CONNECTIONS.

If a Fibex file has been imported into the EB tresos Studio source model, the following additional CAN-TP-CONNECTION parameter values are set:

Target model property	Fibex model property
TIMEOUT-BS	TP-CHANNEL/TIMEOUT-BS
TIMEOUT-CR	TP-CHANNEL/TIMEOUT-CR
TRANSMIT-CANCELLATION	TP-CHANNEL/TRANSMIT-CANCELLATION
MULTICAST-REF	TP-CHANNEL/TP-CONNECTIONS/TP-CONNECTION/TP-MULTICAST-CONNECTION/
No target model parameter is available.	MULTICAST-ADDRESSING

New parameters in the target model

TRANSMITTER-REF

The parameter TRANSMITTER-REF references the CAN-TP-NODE that has been created in the target model for the COMMUNICATION-CONNECTOR, which is referenced by SOURCE-REF in the source model.

RECEIVER-REFS/RECEIVER-REF

The parameter RECEIVER-REFS/RECEIVER-REF references the CAN-TP-NODE that has been created in the target model for the COMMUNICATION-CONNECTOR referenced by TARGET-REF in the source model.

CAN-TP-CHANNEL

For every created CAN-TP-CONNECTION, one additional CAN-TP-CHANNEL is created, which the CAN-TP-CONNECTION references in CAN-TP-CHANNEL-REF. The CHANNEL-MODE of the CAN-TP-CHANNEL is set to HALF-DUPLEX-MODE. It is aggregated in CAN-TP-CONFIG/TP-CHANNELS.

TP-CHANNEL/CHANNEL-ID

If a Fibex file has been imported into the EB tresos Studio source model, CAN-TP-CHANNEL/CHANNEL-ID is set additionally to its Fibex counterpart TP-CHANNEL/CHANNEL-ID.

Changed or moved parameters

MINIMUM-SEPARATION-TIME

The parameter MINIMUM-SEPARATION-TIME is upgraded to the receiver CAN-TP-NODE, see [Section 8.1.1.4.1, “CAN-TP-CONFIG”](#).

8.1.1.5. FlexRay Transport Layer

8.1.1.5.1. FLEXRAY-TP-CONFIG

For each FLEXRAY-CLUSTER in the source model that contains at least one FLEX-RAY-TP-CHANNEL in one of its PHYSICAL-CHANNELS, one FLEXRAY-TP-CONFIG element is created in the target model. It resides in the same AR-PACKAGE as the related FLEXRAY-CLUSTER.

New parameters in the target model

FLEXRAY-TP-ECU

For each ECU-INSTANCE in the source model which is attached to one of the PHYSICAL-CHANNELS, one FLEXRAY-TP-ECU element is created in the target model. In ECU-INSTANCE-REF it references the target model counterpart of the source model ECU-INSTANCE. All created FLEXRAY-TP-ECU elements are aggregated in FLEXRAY-TP-CONFIG/TP-ECUS.

FLEXRAY-TP-ECU/CYCLE-TIME-MAIN-FUNCTION

If a Fibex file has been imported into the EB tresos Studio source model, FLEXRAY-TP-ECU/CYCLE-TIME-MAIN-FUNCTION is additionally set using the Fibex parameter TP-ECU/CYCLE-TIME-MAIN-FUNCTION.

FLEXRAY-TP-NODE

For each source model COMMUNICATION-CONNECTOR attached to any PHYSICAL-CHANNEL of a FLEXRAY-CLUSTER for which a FLEXRAY-TP-CONFIG has been created in the target model, one FLEXRAY-TP-NODE is added. In CONNECTOR-REFS/CONNECTOR-REF, it references the COMMUNICATION-CONNECTOR counterpart in the target model and is aggregated in FLEXRAY-TP-CONFIG/TP-NODES.

TP-ADDRESS

An additional entity TP-ADDRESS is created which the FLEXRAY-TP-NODE references via TP-ADDRESS-REF. The TP-ADDRESS contains in its TP-ADDRESS parameter the value obtained from the COM-

MUNICATION-CONNECTOR parameter TP-ADDRESS. It is aggregated in FLEXRAY-TP-CONFIG/TP-AD-DRESSS.

8.1.1.5.2. FLEXRAY-TP-CONNECTION

Source model entity: FLEX-RAY-TP-CONNECTION

Target model entity: FLEXRAY-TP-CONNECTION

Target model property	Source model property
DIRECT-TP-SDU-REF	DIRECT-TP-SDU-REF
REVERSED-TP-SDU-REF	REVERSED-TP-SDU-REF
BANDWIDTH-LIMITATION	No source model parameter is available.

The created FLEXRAY-TP-CONNECTION is aggregated in FLEXRAY-TP-CONFIG/TP-CONNECTIONS.

New parameters in the target model

FLEXRAY-TP-PDU-POOL

For every created FLEXRAY-TP-CONNECTION, one additional FLEXRAY-TP-PDU-POOL is created and aggregated in FLEXRAY-TP-CONFIG/PDU-POOLS. The FLEXRAY-TP-CONNECTION references two FLEXRAY-TP-PDU-POOLS in TX-PDU-POOL-REF and RX-PDU-POOL-REF.

Via RX-PDU-POOL-REF, FLEXRAY-TP-PDU-POOL/N-PDU-REFS references all target model counter-parts of the source model N-PDUS that the ECU-INSTANCE receives and that are referenced in the source model either in FLEX-RAY-TP-CONNECTION/FLOW-CONTROL-PDU-REF or in FLEX-RAY-TP-CONNEC-TION/TRANSMIT-PDU-REFS/TRANSMIT-PDU-REF.

Via TX-PDU-POOL-REF, FLEXRAY-TP-PDU-POOL/N-PDU-REFS references all target model counter-parts of the source model N-PDUS that the ECU-INSTANCE sends and that are referenced in the source model either in FLEX-RAY-TP-CONNECTION/FLOW-CONTROL-PDU-REF or in FLEX-RAY-TP-CONNEC-TION/TRANSMIT-PDU-REFS/TRANSMIT-PDU-REF.

TRANSMITTER-REF

The parameter TRANSMITTER-REF references the FLEXRAY-TP-NODE that has been created in the target model for the COMMUNICATION-CONNECTOR referenced by SOURCE-REF in the source model.

RECEIVER-REFS/RECEIVER-REF

The parameter RECEIVER-REFS/RECEIVER-REF references the FLEXRAY-TP-NODE that has been created in the target model for the COMMUNICATION-CONNECTOR referenced by TARGET-REF in the source model.

FLEXRAY-TP-CONNECTION/MULTICAST-REF

If a Fibex file has been imported into the EB tresos Studio source model, FLEXRAY-TP-CONNECTION/MULTICAST-REF is set according to the content of the Fibex parameter TP-CHANNEL/TP-CONNECTIONS/TP-CONNECTION/TP-MULTICAST-CONNECTION.

8.1.1.5.3. FLEXRAY-TP-CONNECTION-CONTROL

Source model entity: FLEX-RAY-TP-CHANNEL

Target model entity: FLEXRAY-TP-CONNECTION-CONTROL

Target model property	Source model property
ACK-TYPE	ACK-TYPE
No target model parameter is available.	EXTENDED-ADDRESSING
No target model parameter is available.	MAXIMUM-MESSAGE-LENGTH
No target model parameter is available.	MINIMUM-SEPARATION-TIME
No target model parameter is available.	MULTICAST-SEGMENTATION
No target model parameter is available.	PDU-POOLS
No target model parameter is available.	TIME-CS
No target model parameter is available.	TRANSMIT-CANCELLATION

The created FLEXRAY-TP-CONNECTION-CONTROL is aggregated in FLEXRAY-TP-CONFIG/TP-CONNECTION-CONTROLS.

In the source model, the FLEX-RAY-TP-CHANNEL aggregates all its FLEX-RAY-TP-CONNECTION elements. In the target model, the FLEXRAY-TP-CONNECTION elements reference the FLEXRAY-TP-CONNECTION-CONTROL they belong to via TP-CONNECTION-CONTROL-REF.

If a Fibex file has been imported into the EB tresos Studio source model, the following additional FLEXRAY-TP-CONNECTION-CONTROL parameter values are set:

Target model property	Fibex model property
MAX-BUFFER-SIZE	TP-CHANNEL/MAX-BLOCK-SIZE
TIMEOUT-BS	TP-CHANNEL/TIMEOUT-BS
MAX-RETRIES	TP-CHANNEL/MAX-RETRIES
TIMEOUT-CR	TP-CHANNEL/TIMEOUT-CR
TIMEOUT-AR	max (TP-NODE/TIMEOUT-AR)
TIMEOUT-AS	max (TP-NODE/TIMEOUT-AS)
MAX-AR	max (TP-NODE/MAX-AR)

Target model property	Fibex model property
MAX-AS	max(TP-NODE/MAX-AS)
MAX-FR-IF	max(TP-NODE/MAX-FRIF)
TIME-BUFFER	max(TP-NODE/TIME-BUFFER)
TIME-FR-IF	max(TP-NODE/TIME-FRIF)
MAX-FC-WAIT	max(TP-NODE/BUFFER-REQUEST)

For the configuration of TIMEOUT-AR, TIMEOUT-AS, MAX-AR, MAX-AS, MAX-FR-IF, TIME-BUFFER, TIME-FR-IF, MAX-FC-WAIT the maximum of the respective Fibex parameter values of all TP-NODES connected to the TP-CHANNEL is taken.

New parameters in the target model

FLEX-RAY-TP-CONNECTION

Every target model counterpart of the source model FLEX-RAY-TP-CONNECTION which is aggregated in FLEX-RAY-TP-CHANNEL/TP-CONNECTIONS/FLEX-RAY-TP-CONNECTION, references the FLEXRAY-TP-CONNECTION-CONTROL element in its parameter TP-CONNECTION-CONTROL-REF.

8.1.1.6. LIN Transport Layer

8.1.1.6.1. LIN-TP-CONFIG

For each LIN-CLUSTER in the source model that contains at least one LIN-TP-CHANNEL in its PHYSICAL-CHANNEL, one LIN-TP-CONFIG element is created in the target model. It resides in the same AR-PAGE as the related LIN-CLUSTER.

New parameters in the target model

LIN-TP-NODE

For each source model COMMUNICATION-CONNECTOR attached to a PHYSICAL-CHANNEL of a LIN-CLUSTER for which a LIN-TP-CONFIG has been created in the target model, one LIN-TP-NODE is added. In CONNECTOR-REF, it references the COMMUNICATION-CONNECTOR counterpart in the target model. It is aggregated in LIN-TP-CONFIG/TP-NODES.

Target model property	Source model property
MAX-NUMBER-OF-RESP-PENDING-FRAMES	No source model parameter is available.
P-2-MAX	No source model parameter is available.

Target model property	Source model property
P-2-TIMING	No source model parameter is available.

TP-ADDRESS

An additional entity TP-ADDRESS is created which the LIN-TP-NODE references via TP-ADDRESS-REF. The LIN-TP-ADDRESS contains in its TP-ADDRESS parameter the value obtained from the COMMUNICATION-CONNECTOR parameter TP-ADDRESS. It is aggregated in LIN-TP-CONFIG/TP-ADDRESSSS.

8.1.1.6.2. LIN-TP-CONNECTION

Source model entity: LIN-TP-CHANNEL

Target model entity: LIN-TP-CONNECTION

Target model property	Source model property
DATA-PDU-REF	DATA-PDU-REF
FLOW-CONTROL-REF	FLOW-CONTROL-REF
LIN-TP-N-SDU-REF	LIN-TP-N-SDU-REF
No target model parameter is available..	MULTICAST-ADDRESSING
MULTICAST-REF	No source model parameter is available.

The created LIN-TP-CONNECTION is aggregated in LIN-TP-CONFIG/TP-CONNECTIONS.

New parameters in the target model

TRANSMITTER-REF

The parameter TRANSMITTER-REF references the LIN-TP-NODE that has been created in the target model for the COMMUNICATION-CONNECTOR referenced by SOURCE-REF in the source model.

RECEIVER-REFS/RECEIVER-REF

The parameter RECEIVER-REFS/RECEIVER-REF references the LIN-TP-NODE that has been created in the target model for the COMMUNICATION-CONNECTOR referenced by TARGET-REF in the source model.

8.1.1.7. Network Management

8.1.1.7.1. NM-CONFIG

For every SYSTEM element in the source model, one NM-CONFIG element is created, which resides in the same AR-PACKAGE as the related SYSTEM.

8.1.1.7.2. NM-ECU

For every ECU-INSTANCE in the source model, one NM-ECU instance is created in the target model. The related ECU-INSTANCE is referenced in ECU-INSTANCE-REF.

If a Fibex file has been imported into the EB tresos Studio source model, the following additional NM-ECU parameter values are set:

Target model property	Fibex model property
NM-REMOTE-SLEEP-IND-ENABLED	NM-ECU/REMOTE-SLEEP-INDICATION-ENABLED
NM-PASSIVE-MODE-ENABLED	NM-ECU/PASSIVE-MODE-ENABLED
NM-BUS-SYNCHRONIZATION-ENABLED	NM-ECU/BUS-SYNCHRONIZATION-ENABLED
NM-REPEAT-MSG-IND-ENABLED	NM-ECU/NM-ECU-EXTENSIONS/NM-ECU-EXTENSION/REPEAT-MSG-INDICATION-ENABLED
NM-MULTIPLE-CHANNELS-ENABLED	NM-ECU/MULTIPLE-CHANNELS-ENABLED
NM-COM-CONTROL-ENABLED	NM-ECU/COM-CONTROL-ENABLED
NM-STATE-CHANGE-IND-ENABLED	NM-ECU/STATE-CHANGE-INDICATION-ENABLED
NM-PDU-RX-INDICATION-ENABLED	NM-ECU/PDU-RX-INDICATION-ENABLED

NOTE


NM-REPEAT-MSG-IND-ENABLED is only configured for NM-ECUs connected to CAN-NM-CLUSTERS

The parameter NM-REPEAT-MSG-IND-ENABLED is only configured for NM-ECUs connected to CAN-NM-CLUSTERS.

New parameters in the target model

NM-NODE-DETECTION-ENABLED

NM-NODE-DETECTION-ENABLED is configured by taking the maximum of NM-NODE-DETECTION-ENABLED of any FLEXRAY-COMMUNICATION-CONTROLLER of the ECU-INSTANCE (supported by AUTOSAR 3.1 rev 0 system description file imports only). If none of these FLEXRAY-COMMUNICATION-CONTROLLER elements has the value set, the maximum NM-NODE-DETECTION-ENABLED value of all COMMUNICATION-CLUSTER elements to which the ECU-INSTANCE is connected, is taken.

NM-NODE-ID-ENABLED

NM-NODE-ID-ENABLED is determined by taking the maximum NM-NODE-ID-ENABLED value of all COMMUNICATION-CLUSTER elements to which the ECU-INSTANCE is connected (supported by AUTOSAR 3.1 rev 0 system description file imports only).

NM-USER-DATA-ENABLED

NM-USER-DATA-ENABLED is configured by taking the maximum NM-USER-DATA-ENABLED value of all CAN-COMMUNICATION-CONTROLLERS and FLEXRAY-COMMUNICATION-CONTROLLERS to which the ECU-INSTANCE is connected.

8.1.1.7.3. CAN-NM-CLUSTER

Source model entity: CAN-CLUSTER

Target model entity: CAN-NM-CLUSTER

Target model property	Source model property
NM-BUSLOAD-REDUCTION-ACTIVE	NM-BUS-LOAD-REDUCTION-ACTIVE
NM-NETWORK-TIMEOUT	NM-TIMEOUT-TIME
NM-REPEAT-MESSAGE-TIME	NM-REPEAT-MESSAGE-STATE-TIME
NM-REMOTE-SLEEP-INDICATION-TIME	NM-REMOTE-SLEEP-INDICATION-TIME
NM-WAIT-BUS-SLEEP-TIME	NM-WAIT-BUS-SLEEP-TIME
NM-CHANNEL-SLEEP-MASTER	No source model parameter is available.
NM-IMMEDIATE-NM-CYCLE-TIME	No source model parameter is available.
NM-IMMEDIATE-NM-TRANSMISSIONS	No source model parameter is available.

The created CAN-NM-CLUSTER is aggregated in NM-CONFIG/NM-CLUSTERS. It references the related CAN-CLUSTER in COMMUNICATION-CLUSTER-REF.

If a Fibex file has been imported into the EB tresos Studio source model, the following additional CAN-NM-CLUSTER parameter values are set:

Target model property	Fibex model property
NM-CHANNEL-ID	NM-CLUSTER/CHANNEL-ID
NM-CHANNEL-ACTIVE	NM-CLUSTER/CHANNEL-ACTIVE
NM-MESSAGE-TIMEOUT-TIME	NM-CLUSTER/MSG-TIMEOUT-TIME

New parameters in the target model

NM-MSG-CYCLE-TIME and NM-REMOTE-SLEEP-INDICATION-TIME

The EB tresos Studio internal system model additionally configures the parameters NM-MSG-CYCLE-TIME and NM-REMOTE-SLEEP-INDICATION-TIME, if the source model has been imported from a 3.1 rev 0 AUTOSAR system description file which provides the corresponding source parameters in CAN-CLUSTER/NM-REMOTE-SLEEP-INDICATION-TIME and CAN-CLUSTER/NM-MSG-CYCLE-TIME.

NM-CBV-POSITION, NM-NID-POSITION

NM-CBV-POSITION, NM-NID-POSITION are calculated by taking the maximum NM-CBV-POSITION, NM-NID-POSITION value of any NM-PDU transmitted over the CAN-CLUSTER in the source model, then dividing the result by eight.

NM-USER-DATA-LENGTH

NM-USER-DATA-LENGTH is calculated by taking the maximum NM-USER-DATA-LENGTH value of any NM-PDU transmitted over the CAN-CLUSTER.

8.1.1.7.4. CAN-NM-NODE

For every CAN-COMMUNICATION-CONTROLLER connected to a CAN-CLUSTER in the source model, one CAN-NM-NODE entity is created. It is aggregated in CAN-NM-CLUSTER/NM-NODES.

New parameters in the target model

NM-RANGE-CONFIG

If the COMMUNICATION-CLUSTER to which the CAN-COMMUNICATION-CONTROLLER is connected defines values for both NM-LOWER-CAN-ID and NM-UPPER-CAN-ID, one NM-RANGE-CONFIG element is added to the CAN-NM-NODE, having set LOWER-CAN-ID and UPPER-CAN-ID accordingly.

CONTROLLER-REF

CONTROLLER-REF is configured to reference the target model counterpart of the CAN-COMMUNICATION-CONTROLLER.

NM-NODE-ID

The parameter NM-NODE-ID is derived from the NM-ADDRESS parameter value of COMMUNICATION-CONNECTOR referencing the source model counterpart of the CAN-COMMUNICATION-CONTROLLER.

NM-IF-ECU-REF

NM-IF-ECU-REF is configured to reference the NM-ECU element, which has been created for the source model ECU-INSTANCE that aggregates the CAN-COMMUNICATION-CONTROLLER this CAN-NM-NODE has been created for.

RX-NM-PDU-REFS/RX-NM-PDU-REF

RX-NM-PDU-REFS/RX-NM-PDU-REF is configured to reference all NM-PDUS the CAN-COMMUNICATION-CONTROLLER is receiving.

TX-NM-PDU-REFS/TX-NM-PDU-REF

TX-NM-PDU-REFS/TX-NM-PDU-REF is configured to reference all NM-PDUS the CAN-COMMUNICATION-CONTROLLER is sending.

NM-MSG-REDUCED-TIME

NM-MSG-REDUCED-TIME is configured by CAN-COMMUNICATION-CONTROLLER/NM-MSG-REDUCED-TIME (supported by AUTOSAR 3.1 rev 0 system description file imports only).

NM-MSG-CYCLE-OFFSET

NM-MSG-CYCLE-OFFSET is configured by CAN-COMMUNICATION-CONTROLLER/NM-MSG-CYCLE-OFFSET (supported by AUTOSAR 3.1 rev 0 system description file imports only).

8.1.1.7.5. CAN-NM-CLUSTER-COUPING

CAN-NM-CLUSTER-COUPING defines parameters which have to be set consistently among two coupled CAN-NM-CLUSTER elements. *Coupled* means that least one NM-ECU exists, which is connected to both CAN-NM-CLUSTER elements. The coupling is transitive, which means that if CAN-NM-CLUSTER A is coupled to CAN-NM-CLUSTER B and CAN-NM-CLUSTER B is coupled to CAN-NM-CLUSTER C, CAN-NM-CLUSTER A is also coupled to CAN-NM-CLUSTER C.

For each of the coupled sets of CAN-NM-CLUSTER elements, and for each CAN-NM-CLUSTER not coupled to any other CAN-NM-CLUSTER, one element is created, aggregated in CAN-NM-CONFIG/NM-CLUSTER-COUPLINGS.

If a Fibex file has been imported into the EB tresos Studio source model, the following additional CAN-NM-CLUSTER-COUPING parameter values are set (provided that they are defined consistently in the imported Fibex file):

Target model property	Fibex model property
NM-IMMEDIATE-RESTART-ENABLED	NM-CLUSTER-COUPINGS/NM-CLUSTER-COUPLING/IMMEDIATE-RESTART-ENABLED

New parameters in the target model

COUPLED-CLUSTER-REFS/COUPLED-CLUSTER-REF

COUPLED-CLUSTER-REFS/COUPLED-CLUSTER-REF is configured to reference all CAN-NM-CLUSTERS that are coupled.

NM-BUSLOAD-REDUCTION-ENABLED

If the NM-BUS-LOAD-REDUCTION-ENABLED parameters of the CAN-CLUSTER counterparts of the coupled CAN-NM-CLUSTERS yield a consistent value, the NM-BUSLOAD-REDUCTION-ENABLED of the CAN-NM-CLUSTER-COUPING is configured accordingly.

8.1.1.7.6. FLEXRAY-NM-CLUSTER

One FLEXRAY-NM-CLUSTER is created per FLEXRAY-CLUSTER in the source model. It is aggregated in NM-CONFIG/NM-CLUSTERS. It references the related FLEXRAY-CLUSTER in COMMUNICATION-CLUSTER-REF.

If a Fibex file has been imported into the EB tresos Studio source model, the following additional FLEXRAY-NM-CLUSTER parameter values are set:

Target model property	Fibex model property
NM-CHANNEL-ID	NM-CLUSTER/CHANNEL-ID
NM-MESSAGE-TIMEOUT-TIME	NM-CLUSTER/MSG-TIMEOUT-TIME

Target model property	Fibex model property
NM-DATA-ENABLED	NM-CLUSTER/DATA-ENABLED
NM-MAIN-FUNCTION-PERIOD	max (NM-NODE/CYCLE-TIME-MAIN-FUNCTION)

For the configuration of NM-MAIN-FUNCTION-PERIOD, the maximum CYCLE-TIME-MAIN-FUNCTION value of all NM-NODES connected to the FLEXRAY-NM-CLUSTER is taken.

New parameters in the target model

NM-REMOTE-SLEEP-INDICATION-TIME, NM-REPEAT-MESSAGE-TIME, NM-REPETITION-CYCLE, NM-VOTING-CYCLE, NM-DATA-CYCLE, NM-READY-SLEEP-COUNT

NM-REMOTE-SLEEP-INDICATION-TIME, NM-REPEAT-MESSAGE-TIME, NM-REPETITION-CYCLE, NM-VOTING-CYCLE, NM-DATA-CYCLE, NM-READY-SLEEP-COUNT are determined by taking the respective parameter values of the source model FLEXRAY-CLUSTER (supported by AUTOSAR 3.1 rev 0 system description file imports only).

NM-CONTROL-BIT-VECTOR-ACTIVE

NM-CONTROL-BIT-VECTOR-ACTIVE is determined by taking the maximum NM-CONTROL-BIT-VECTOR-ENABLED parameter value of all source model FLEXRAY-COMMUNICATION-CONTROLLERS which are connected to the FLEXRAY-CLUSTER.

8.1.1.7.7. FLEXRAY-NM-NODE

For every FLEXRAY-COMMUNICATION-CONTROLLER connected to a FLEXRAY-CLUSTER in the source model, one FLEXRAY-NM-NODE entity is created. It is aggregated in FLEXRAY-NM-CLUSTER/NM-NODES.

New parameters in the target model

CONTROLLER-REF

CONTROLLER-REF is configured to reference the target model counterpart of the FLEXRAY-COMMUNICATION-CONTROLLER.

NM-NODE-ID

The parameter NM-NODE-ID is derived from the maximum NM-ADDRESS parameter value of all COMMUNICATION-CONNECTORS referencing the source model counterpart of the FLEXRAY-COMMUNICATION-CONTROLLER.

NM-IF-ECU-REF

NM-IF-ECU-REF is configured to reference the NM-ECU element which has been created for the source model ECU-INSTANCE aggregating the FLEXRAY-COMMUNICATION-CONTROLLER this FLEXRAY-NM-NODE has been created for.

RX-NM-PDU-REFS/RX-NM-PDU-REF

RX-NM-PDU-REFS/RX-NM-PDU-REF is configured to reference all NM-PDUS the FLEXRAY-COMMUNICATION-CONTROLLER is receiving.

TX-NM-PDU-REFS/TX-NM-PDU-REF

TX-NM-PDU-REFS/TX-NM-PDU-REF is configured to reference all NM-PDUS the FLEXRAY-COMMUNICATION-CONTROLLER is sending.

FLEXRAY-NM-NODE/NM-INSTANCE-ID

If a Fibex file has been imported into the EB tresos Studio source model, **FLEXRAY-NM-NODE/NM-INSTANCE-ID** is set according to the content of the Fibex parameter **NM-NODE/INSTANCE-ID**.

8.1.1.7.8. FLEXRAY-NM-CLUSTER-COUPING

FLEXRAY-NM-CLUSTER-COUPING defines parameters which have to be set consistently among two FLEXRAY-NM-CLUSTER elements which are coupled. [Section 8.1.1.7.5, “CAN-NM-CLUSTER-COUPING”](#) describes the peculiarities of the CAN cluster analogy CAN-NM-CLUSTER-COUPING.

For each of the coupled sets of FLEXRAY-NM-CLUSTER elements, and for each FLEXRAY-NM-CLUSTER not coupled to any other FLEXRAY-NM-CLUSTER, one FLEXRAY-NM-CLUSTER-COUPING element is created, aggregated in FLEXRAY-NM-CONFIG/NM-CLUSTER-COUPINGS.

If a Fibex file has been imported into the EB tresos Studio source model, the following additional FLEXRAY-NM-CLUSTER-COUPING parameter values are set (provided that they are defined consistently in the imported Fibex file):

Target model property	Fibex model property
NM-SCHEDULE-VARIANT	NM-CLUSTER-COUPINGS/NM-CLUSTER-COUPLING/SCHEDULE-VARIANT
NM-CONTROL-BIT-VECTOR-ENABLED	NM-CLUSTER-COUPINGS/NM-CLUSTER-COUPLING/CONTROL-BIT-VECTOR-ENABLED
NM-DATA-DISABLED	NM-CLUSTER-COUPINGS/NM-CLUSTER-COUPLING/NM-DATA-DISABLED

8.1.1.8. Software Component Description**8.1.1.8.1. COMPOSITION-SW-COMPONENT-TYPE**

Source model entity: COMPOSITION-TYPE

Target model entity: COMPOSITION-SW-COMPONENT-TYPE

Target model property	Source model property
POR TS/P-PORT-PROTOTYPE	POR TS/P-PORT-PROTOTYPE
POR TS/R-PORT-PROTOTYPE	POR TS/R-PORT-PROTOTYPE
PORT-GROUPS/PORT-GROUP	No source model parameter is available.
COMPONENTS/SW-COMPONENT-PROTOTYPE	COMPONENTS/COMPONENT-PROTOTYPE
CONNECTORS/ASSEMBLY-SW-CONNECTOR	CONNECTORS/ASSEMBLY-CONNECTOR-PROTOTYPE
/CONNECTORS/DELEGATION-SW-CONNECTOR	CONNECTORS/DELEGATION-CONNECTOR-PROTOTYPE
No target model parameter is available.	CONNECTORS/SERVICE-CONNECTOR-PROTOTYPE

8.1.1.8.2. APPLICATION-SW-COMPONENT-TYPE

Source model entities: APPLICATION-SOFTWARE-COMPONENT-TYPE, SENSOR-ACTUATOR-SOFT-WARE-COMPONENT-TYPE

Target model entity: APPLICATION-SW-COMPONENT-TYPE

Target model property	Source model property
POR TS/P-PORT-PROTOTYPE	POR TS/P-PORT-PROTOTYPE
POR TS/R-PORT-PROTOTYPE	POR TS/R-PORT-PROTOTYPE
PORT-GROUPS/PORT-GROUP	No source model parameter is available.

New parameters in the target model

INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR

The parameter INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR contains the SWC-INTERNAL-BEHAVIOR which corresponds to the source model INTERNAL-BEHAVIOR which references the APPLICATION-SOFTWARE-COMPONENT-TYPE via COMPONENT-REF.

8.1.1.8.3. SERVICE-SW-COMPONENT-TYPE

Source model entity: SERVICE-COMPONENT-TYPE

Target model entity: SERVICE-SW-COMPONENT-TYPE

Target model property	Source model property
POR TS/P-PORT-PROTOTYPE	POR TS/P-PORT-PROTOTYPE
POR TS/R-PORT-PROTOTYPE	POR TS/R-PORT-PROTOTYPE
PORT-GROUPS/PORT-GROUP	No source model parameter is available.
No Upgrade is currently implemented.	BSW-MODULE-DESCRIPTION-REFS/BSW-MOD-ULE-DESCRIPTION-REF

New parameters in the target model

INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR

The parameter INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR contains the SWC-INTERNAL-BEHAVIOR which corresponds to the source model INTERNAL-BEHAVIOR which references the APPLICATION-SOFTWARE-COMPONENT-TYPE via COMPONENT-REF.

8.1.1.8.4. COMPLEX-DEVICE-DRIVER-SW-COMPONENT-TYPE

Source model entity: COMPLEX-DEVICE-DRIVER-COMPONENT-TYPE

Target model entity: COMPLEX-DEVICE-DRIVER-SW-COMPONENT-TYPE

Target model property	Source model property
POR TS/P-PORT-PROTOTYPE	POR TS/P-PORT-PROTOTYPE
POR TS/R-PORT-PROTOTYPE	POR TS/R-PORT-PROTOTYPE
PORT-GROUPS/PORT-GROUP	No source model parameter is available.
No Upgrade is currently implemented.	BSW-MODULE-DESCRIPTION-REFS/BSW-MOD-ULE-DESCRIPTION-REF
No Upgrade is currently implemented.	HARDWARE-ELEMENT-REFS/HARDWARE-ELE-MENT-REF

New parameters in the target model

INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR

The parameter INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR contains the SWC-INTERNAL-BEHAVIOR which corresponds to the source model INTERNAL-BEHAVIOR which references the APPLICATION-SOFTWARE-COMPONENT-TYPE via COMPONENT-REF.

8.1.1.8.5. ECU-ABSTRACTION-SW-COMPONENT-TYPE

Source model entity: ECU-ABSTRACTION-COMPONENT-TYPE

Target model entity: ECU-ABSTRACTION-SW-COMPONENT-TYPE

Target model property	Source model property
PORTS/P-PORT-PROTOTYPE	PORTS/P-PORT-PROTOTYPE
PORTS/R-PORT-PROTOTYPE	PORTS/R-PORT-PROTOTYPE
PORT-GROUPS/PORT-GROUP	No source model parameter is available.
No upgrade is currently implemented.	BSW-MODULE-DESCRIPTION-REFS/BSW-MODULE-DESCRIPTION-REF
No upgrade is currently implemented.	HARDWARE-ELEMENT-REFS/HARDWARE-ELEMENT-REF

New parameters in the target model

INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR

The parameter INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR contains the SWC-INTERNAL-BEHAVIOR which corresponds to the source model INTERNAL-BEHAVIOR which references the APPLICATION-SOFTWARE-COMPONENT-TYPE via COMPONENT-REF.

8.1.1.8.6. PARAMETER-SW-COMPONENT-TYPE

Source model entity: CALPRM-COMPONENT-TYPE

Target model entity: PARAMETER-SW-COMPONENT-TYPE

Target model property	Source model property
PORTS/P-PORT-PROTOTYPE	PORTS/P-PORT-PROTOTYPE
PORTS/R-PORT-PROTOTYPE	PORTS/R-PORT-PROTOTYPE
PORT-GROUPS/PORT-GROUP	No source model parameter is available.
CONSTANT-MAPPING-REFS/CONSTANT-MAPPING-REF	No source model parameter is available.
DATA-TYPE-MAPPING-REFS/DATA-TYPE-MAPPING-REF	Refer to the section called “Creation of target model DATA-TYPE-MAPPING-REFS/DATA-TYPE-MAPPING-REF”.

Target model property	Source model property
INSTANTIATION-DATA-DEF-PROPS/INSTANTIATION-DATA-DEF-PROPS	No source model parameter is available.

New parameters in the target model

INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR

The parameter INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR contains the SWC-INTERNAL-BEHAVIOR which corresponds to the source model INTERNAL-BEHAVIOR which references the APPLICATION-SOFTWARE-COMPONENT-TYPE via COMPONENT-REF.

8.1.1.8.7. R-PORT-PROTOTYPE

Source model entity: R-PORT-PROTOTYPE

Target model entity: R-PORT-PROTOTYPE

Target model property	Source model property
REQUIRED-COM-SPECS/CLIENT-COM-SPEC	REQUIRED-COM-SPECS/CLIENT-COM-SPEC
REQUIRED-COM-SPECS/PARAMETER-REQUIRED-COM-SPEC	REQUIRED-COM-SPECS/PARAMETER-REQUIRED-COM-SPEC
REQUIRED-COM-SPECS/QUEUED-RECEIVER-COM-SPEC	REQUIRED-COM-SPECS/QUEUED-RECEIVER-COM-SPEC
REQUIRED-COM-SPECS/NONQUEUED-RECEIVER-COM-SPEC	REQUIRED-COM-SPECS/UNQUEUED-RECEIVER-COM-SPEC
REQUIRED-COM-SPECS/NV-REQUIRED-COM-SPEC	No source model parameter is available.
REQUIRED-INTERFACE-TREF	REQUIRED-INTERFACE-TREF

8.1.1.8.8. P-PORT-PROTOTYPE

Source model entity: P-PORT-PROTOTYPE

Target model entity: P-PORT-PROTOTYPE

Target model property	Source model property
PROVIDED-COM-SPECS/MODE-SWITCH-SENDER-COM-SPEC	PROVIDED-COM-SPECS/MODE-SWITCH-COM-SPEC
PROVIDED-COM-SPECS/NONQUEUED-SENDER-COM-SPEC	PROVIDED-COM-SPECS/UNQUEUED-SENDER-COM-SPEC

Target model property	Source model property
PROVIDED-COM-SPECS/NV-PROVIDE-COM-SPEC	No source model parameter is available.
PROVIDED-COM-SPECS/PARAMETER-PROVIDE-COM-SPEC	PROVIDED-COM-SPECS/PARAMETER-PROVIDE-COM-SPEC
PROVIDED-COM-SPECS/QUEUED-SENDER-COM-SPEC	PROVIDED-COM-SPECS/QUEUED-SENDER-COM-SPEC
PROVIDED-COM-SPECS/SERVER-COM-SPEC	PROVIDED-COM-SPECS/SERVER-COM-SPEC
PROVIDED-INTERFACE-TREF	PROVIDED-INTERFACE-TREF

8.1.1.8.9. CLIENT-SERVER-INTERFACE

Source model entity: CLIENT-SERVER-INTERFACE

Target model entity: CLIENT-SERVER-INTERFACE

Target model property	Source model property
OPERATIONS/CLIENT-SERVER-OPERATION	OPERATIONS/OPERATION-PROTOTYPE
POSSIBLE-ERRORS/APPLICATION-ERROR	POSSIBLE-ERRORS/APPLICATION-ERROR
SERVICE-KIND	No source model parameter is available.

8.1.1.8.10. ASSEMBLY-SW-CONNECTOR

Source model entity: ASSEMBLY-CONNECTOR-PROTOTYPE

Target model entity: ASSEMBLY-SW-CONNECTOR

Target model property	Source model property
PROVIDER-IREF/COMPONENT-PROTOTYPE-REF	PROVIDER-IREF/COMPONENT-PROTOTYPE-REF
PROVIDER-IREF/P-PORT-PROTOTYPE-REF	PROVIDER-IREF/P-PORT-PROTOTYPE-REF
REQUESTER-IREF/COMPONENT-PROTOTYPE-REF	REQUESTER-IREF/COMPONENT-PROTOTYPE-REF
REQUESTER-IREF/R-PORT-PROTOTYPE-REF	REQUESTER-IREF/R-PORT-PROTOTYPE-REF
No source model parameter is available.	MAPPING-REF

8.1.1.8.11. DELEGATION-SW-CONNECTOR

Source model entity: DELEGATION-CONNECTOR-PROTOTYPE

Target model entity: DELEGATION-SW-CONNECTOR

Target model property	Source model property
INNER-PORT-IREF/CONTEXT-COMPONENT-REF	INNER-PORT-IREF/COMPONENT-PROTOTYPE-REF
INNER-PORT-IREF/TARGET-P-PORT-REF	INNER-PORT-IREF/PORT-PROTOTYPE-REF
OUTER-PORT-REF	OUTER-PORT-REF
No source model parameter is available.	MAPPING-REF

8.1.1.8.12. SERVICE-CONNECTOR-PROTOTYPE

AUTOSAR 3.1.x SERVICE-CONNECTOR-PROTOTYPE elements are not upgraded because no counterpart element exists in AUTOSAR 4.0. If such SENDER-RECEIVER-INTERFACE elements are encountered, a warning is issued.

8.1.1.8.13. PARAMETER-INTERFACE

Source model entity: CALPRM-INTERFACE

Target model entity: PARAMETER-INTERFACE

Target model property	Source model property
IS-SERVICE	IS-SERVICE
CALPRM-ELEMENTS/CALPRM-ELEMENT-PROTOTYPE	PARAMETERS/PARAMETER-DATA-PROTOTYPE
SERVICE-KIND	No source model parameter is available.

8.1.1.8.14. MODE-SWITCH-INTERFACE

Source model entity: SENDER-RECEIVER-INTERFACE

Target model entity: MODE-SWITCH-INTERFACE

For every SENDER-RECEIVER-INTERFACE element in the source model that contains one or more MODE-GROUPS, but no DATA-ELEMENT, one MODE-SWITCH-INTERFACE is created. Since the target model only supports the definition of one single MODE-DECLARATION-GROUP-PROTOTYPE, only the first MODE-DECLARATION-GROUP-PROTOTYPE encountered in the source model is upgraded.

The upgrade of SENDER-RECEIVER-INTERFACE elements which in turn contain MODE-GROUPS elements and DATA-ELEMENTS at the same time is not supported. If such SENDER-RECEIVER-INTERFACE elements are encountered, none of the MODE-GROUPS is upgraded and a warning is issued.

Target model property	Source model property
IS-SERVICE	IS-SERVICE
MODE-GROUP	MODE-GROUPS/MODE-DECLARATION-GROUP-PROTOTYPE
SERVICE-KIND	No source model parameter is available.

8.1.1.8.15. MODE-GROUP

Source model entity: MODE-GROUP

Target model entity: MODE-GROUP

Target model property	Source model property
TYPE-TREF	TYPE-TREF

8.1.1.8.16. SW-COMPONENT-PROTOTYPE

Source model entity: COMPONENT-PROTOTYPE

Target model entity: SW-COMPONENT-PROTOTYPE

Target model property	Source model property
TYPE-TREF	TYPE-TREF

8.1.1.8.17. VARIABLE-DATA-PROTOTYPE

Source model entity: DATA-ELEMENT-PROTOTYPE

Target model entity: VARIABLE-DATA-PROTOTYPE

Target model property	Source model property
SW-DATA-DEF-PROPS	SW-DATA-DEF-PROPS
TYPE-TREF	TYPE-TREF
No target model parameter is available.	IS-QUEUED

Target model property	Source model property
INIT-VALUE	No source model parameter is available.

Reason for not upgraded source model parameter IS-QUEUED

IS-QUEUED

The parameter IS-QUEUED is not upgraded to the target model, the target model entity VARIABLE-DATA-PROTOTYPE does not contain it. If IS-QUEUED is set to true in the source model, the parameter SW-DATA-DEF-PROPS/SW-IMPL-POLICY is set to QUEUED.

8.1.1.8.18. ARGUMENT-DATA-PROTOTYPE

Source model entity: ARGUMENT-PROTOTYPE

Target model entity: ARGUMENT-DATA-PROTOTYPE

Target model property	Source model property
SW-DATA-DEF-PROPS	SW-DATA-DEF-PROPS
TYPE-TREF	TYPE-TREF
DIRECTION	DIRECTION
SERVER-ARGUMENT-IMPL-POLICY	No source model parameter is available.

8.1.1.8.19. PARAMETER-DATA-PROTOTYPE

Source model entity: CALPRM-ELEMENT-PROTOTYPE

Target model entity: PARAMETER-DATA-PROTOTYPE

Target model property	Source model property
SW-DATA-DEF-PROPS	SW-DATA-DEF-PROPS
TYPE-TREF	TYPE-TREF

Configuration of target model parameter INIT-VALUE

If the source model contains a LOCAL-PARAMETER-INIT-VALUE-ASSIGNMENT which references in PARAMETER-REF the source model CALPRM-ELEMENT-PROTOTYPE and in INIT-VALUE-REF an initial value, the target model PARAMETER-DATA-PROTOTYPE will reference this value in INIT-VALUE/CONSTANT-REF.

8.1.1.8.20. CLIENT-SERVER-OPERATION

Source model entity: OPERATION-PROTOTYPE

Target model entity: CLIENT-SERVER-OPERATION

Target model property	Source model property
ARGUMENTS/ARGUMENT-DATA-PROTOTYPE	ARGUMENTS/ARGUMENT-PROTOTYPE
POSSIBLE-ERROR-REFS/POSSIBLE-ERROR-REF	POSSIBLE-ERROR-REFS/POSSIBLE-ERROR-REF

8.1.1.8.21. APPLICATION-ERROR

Source model entity: APPLICATION-ERROR

Target model entity: APPLICATION-ERROR

Target model property	Source model property
ERROR-CODE	ERROR-CODE

8.1.1.8.22. NONQUEUED-RECEIVER-COM-SPEC

Source model entity: UNQUEUED-RECEIVER-COM-SPEC

Target model entity: NONQUEUED-RECEIVER-COM-SPEC

Target model property	Source model property
ALIVE-TIMEOUT	ALIVE-TIMEOUT
No target model parameter is available.	HANDLE-INVALID
INIT-VALUE/CONSTANT-REFERENCE	INIT-VALUE-REF
No target model parameter is available.	RESYNC-TIME
DATA-ELEMENT-REF	DATA-ELEMENT-REF
FILTER	FILTER
HANDLE-NEVER-RECEIVED	No source model parameter is available, the target model parameter is set to false.
ENABLE-UPDATE	No source model parameter is available, the target model parameter is set to false.
HANDLE-OUT-OF-RANGE	No source model parameter is available, the target model parameter is set to NONE.

Target model property	Source model property
HANDLE-TIMEOUT-TYPE	No source model parameter is available, the target model parameter is set to <code>NONE</code> .
NETWORK-REPRESENTATION	No source model parameter is available.
USES-END-TO-END-PROTECTION	No source model parameter is available, the target model parameter is set to <code>true</code> .

Processing of source model parameter HANDLE-INVALID

If the parameter `HANDLE-INVALID` is provided in the source model, an `INVALIDATION-POLICY` element is created in the related target model `SENDER-RECEIVER-INTERFACE`. The `INVALIDATION-POLICY` parameter `HANDLE-INVALID` is configured according to the source model `UNQUEUED-RECEIVER-COM-SPEC` parameter `HANDLE-INVALID` if all `UNQUEUED-RECEIVER-COM-SPEC` elements belonging to the source model `SENDER-RECEIVER-INTERFACE` and `DATA-ELEMENT` are providing the same `HANDLE-INVALID` value.

8.1.1.8.23. QUEUED-RECEIVER-COM-SPEC

Source model entity: `QUEUED-RECEIVER-COM-SPEC`

Target model entity: `QUEUED-RECEIVER-COM-SPEC`

Target model property	Source model property
<code>DATA-ELEMENT-REF</code>	<code>DATA-ELEMENT-REF</code>
No target model parameter is available.	<code>FILTER</code>
<code>HANDLE-OUT-OF-RANGE</code>	No source model parameter is available, the target model parameter is set to <code>NONE</code> .
<code>NETWORK-REPRESENTATION</code>	No source model parameter is available.
<code>QUEUE-LENGTH</code>	<code>QUEUE-LENGTH</code>
<code>USES-END-TO-END-PROTECTION</code>	No source model parameter is available, the target model parameter is set to <code>true</code> .

8.1.1.8.24. CLIENT-COM-SPEC

Source model entity: `CLIENT-COM-SPEC`

Target model entity: `CLIENT-COM-SPEC`

Target model property	Source model property
No target model parameter is available.	DATA-ELEMENT-REF
No target model parameter is available.	FILTER
OPERATION-REF	OPERATION-REF

8.1.1.8.25. PARAMETER-REQUIRE-COM-SPEC

Source model entity: PARAMETER-REQUIRE-COM-SPEC

Target model entity: PARAMETER-REQUIRE-COM-SPEC

Target model property	Source model property
No target model parameter is available.	DATA-ELEMENT-REF
No target model parameter is available.	FILTER
INIT-VALUE/CONSTANT-REFERENCE	INIT-VALUE-REF
PARAMETER-REF	PARAMETER-REF

Upgrade of source model INIT-VALUE

The target model contains 0 as INIT-VALUE if the source model does not provide INIT-VALUE-REF.

8.1.1.8.26. QUEUED-SENDER-COM-SPEC

Source model entity: QUEUED-SENDER-COM-SPEC

Target model entity: QUEUED-SENDER-COM-SPEC

Target model property	Source model property
DATA-ELEMENT-REF	DATA-ELEMENT-REF
TRANSMISSION-ACKNOWLEDGE	TRANSMISSION-ACKNOWLEDGE
HANDLE-OUT-OF-RANGE	No source model parameter is available, the target model parameter is set to NONE.
NETWORK-REPRESENTATION	No source model parameter is available.
USES-END-TO-END-PROTECTION	No source model parameter is available, the target model parameter is set to true.

8.1.1.8.27. NONQUEUED-SENDER-COM-SPEC

Source model entity: UNQUEUED-SENDER-COM-SPEC

Target model entity: NONQUEUED-SENDER-COM-SPEC

Target model property	Source model property
DATA-ELEMENT-REF	DATA-ELEMENT-REF
TRANSMISSION-ACKNOWLEDGE	TRANSMISSION-ACKNOWLEDGE
No target model parameter is available.	CAN-INVALIDATE
INIT-VALUE/CONSTANT-REFERENCE	INIT-VALUE-REF
HANDLE-OUT-OF-RANGE	No source model parameter is available, the target model parameter is set to NONE.
NETWORK-REPRESENTATION	No source model parameter is available.
USES-END-TO-END-PROTECTION	No source model parameter is available, the target model parameter is set to true.

8.1.1.8.28. SERVER-COM-SPEC

Source model entity: SERVER-COM-SPEC

Target model entity: SERVER-COM-SPEC

Target model property	Source model property
No target model parameter is available.	DATA-ELEMENT-REF
No target model parameter is available.	TRANSMISSION-ACKNOWLEDGE
OPERATION-REF	OPERATION-REF
QUEUE-LENGTH	QUEUE-LENGTH

8.1.1.8.29. MODE-SWITCH-SENDER-COM-SPEC

Source model entity: MODE-SWITCH-COM-SPEC

Target model entity: MODE-SWITCH-SENDER-COM-SPEC

Target model property	Source model property
MODE-GROUP-REF	MODE-GROUP-REF

Target model property	Source model property
MODE-SWITCHED-ACK	MODE-SWITCHED-ACK
QUEUE-LENGTH	QUEUE-LENGTH

8.1.1.8.30. PARAMETER-PROVIDE-COM-SPEC

Source model entity: PARAMETER-PROVIDE-COM-SPEC

Target model entity: PARAMETER-PROVIDE-COM-SPEC

Target model property	Source model property
INIT-VALUE/CONSTANT-REFERENCE	INIT-VALUE-REF
PARAMETER-REF	PARAMETER-REF

Upgrade of source model INIT-VALUE

The target model contains 0 as INIT-VALUE if the source model does not provide INIT-VALUE-REF.

8.1.1.8.31. TRANSMISSION-ACKNOWLEDGE

Source model entity: TRANSMISSION-ACKNOWLEDGE

Target model entity: TRANSMISSION-ACKNOWLEDGE

Target model property	Source model property
TIMEOUT	TIMEOUT

8.1.1.8.32. DATA-FILTER

Source model entity: TRANSMISSION-MODE-CONDITION/DATA-FILTER

Target model entity: DATA-FILTER

Target model property	Source model property
MASK	MASK
MAX	MAX

Target model property	Source model property
MIN	MIN
OFFSET	OFFSET
PERIOD	PERIOD
X	X

Configuration of target model parameter DATA-FILTER-TYPE

DATA-FILTER-TYPE is configured depending on the source model element below TRANSMISSION-MODE-CONDITION/DATA-FILTER. The following values are converted:

- ▶ ALWAYS
- ▶ MASKED-NEW-DIFFERS-MASKED-OLD
- ▶ MASKED-NEW-DIFFERS-X
- ▶ MASKED-NEW-EQUALS-MASKED-OLD
- ▶ MASKED-NEW-EQUALS-X
- ▶ NEVER
- ▶ NEW-IS-DIFFERENT
- ▶ NEW-IS-EQUAL
- ▶ NEW-IS-GREATER
- ▶ NEW-IS-GREATER-OR-EQUAL
- ▶ NEW-IS-LESS
- ▶ NEW-IS-LESS-OR-EQUAL
- ▶ NEW-IS-OUTSIDE
- ▶ NEW-IS-WITHIN
- ▶ ONE-EVERY-N

8.1.1.8.33. VARIABLE-ACCESS

The target model entity VARIABLE-ACCESS is created out of different source model entities. The source model entities and the employed parameter mapping rules are listed below.

Source model entities: DATA-READ-ACCESS, DATA-RECEIVE-POINT

Target model entity: VARIABLE-ACCESS

Target model property	Source model property
ACCESSED-VARIABLE/AUTOSAR-VARIABLE-IREF/PORT-PROTOTYPE-REF	DATA-ELEMENT-IREF/R-PORT-PROTOTYPE-REF
ACCESSED-VARIABLE/AUTOSAR-VARIABLE-IREF/TARGET-DATA-PROTOTYPE-REF	DATA-ELEMENT-IREF/DATA-ELEMENT-PROTOTYPE-REF

Source model entities: DATA-WRITE-ACCESS, DATA-SEND-POINT

Target model entity: VARIABLE-ACCESS

Target model property	Source model property
ACCESSED-VARIABLE/AUTOSAR-VARIABLE-IREF/PORT-PROTOTYPE-REF	DATA-ELEMENT-IREF/P-PORT-PROTOTYPE-REF
ACCESSED-VARIABLE/AUTOSAR-VARIABLE-IREF/TARGET-DATA-PROTOTYPE-REF	DATA-ELEMENT-IREF/DATA-ELEMENT-PROTOTYPE-REF

8.1.1.8.34. PARAMETER-ACCESS

The target model entity PARAMETER-ACCESS is created out of different source model entities. The source model entities and the employed parameter mapping rules are listed below.

Source model entity: CALPRM-ACCESS

Target model entity: PARAMETER-ACCESS

Target model property	Source model property
PARAMETER-ACCESS/ACCESSED-PARAMETER/AUTOSAR-PARAMETER-IREF/PORT-PROTOTYPE-REF	CALPRM-ACCESS-IREF/PORT-PROTOTYPE-REF
PARAMETER-ACCESS/ACCESSED-PARAMETER/AUTOSAR-PARAMETER-IREF/TARGET-DATA-PROTOTYPE-REF	CALPRM-ACCESS-IREF/CALPRM-ELEMENT-PROTOTYPE-REF

Source model entity: SHARED-CALPRM-ACCESS-REF

Target model entity: PARAMETER-ACCESS

Target model property	Source model property
PARAMETER-ACCESS/ACCESSED-PARAMETER/LOCAL-PARAMETER-REF	SHARED-CALPRM-ACCESS-REF

Source model entity: PER-INSTANCE-CALPRM-ACCESS-REF

Target model entity: PARAMETER-ACCESS

Target model property	Source model property
PARAMETER-ACCESS/ACCESSED-PARAMETER/LOCAL-PARAMETER-REF	PER-INSTANCE-CALPRM-ACCESS-REF

SHORT-NAME of target model ACCESSED-VARIABLE

Please note that for all upgrade variants, an ACCESSED-PARAMETER is created in the target model. For all variants, the ACCESSED-PARAMETER/SHORT-NAME is set to PA<index>, where <index> is an index providing for the uniqueness of ACCESSED-PARAMETER/SHORT-NAME in the target model.

8.1.1.8.35. RUNNABLE-ENTITY

Source model entity: RUNNABLE-ENTITY

Target model entity: RUNNABLE-ENTITY

Target model property	Source model property
Currently no upgrade is available.	BSW-ENTITY-REF
PARAMETER-ACCESSSS/PARAMETER-ACCESS/ACCESSED-PARAMETER	CALPRM-ACCESSSS/CALPRM-ACCESS
CAN-BE-INVOKED-CONCURRENTLY	CAN-BE-INVOKED-CONCURRENTLY
CAN-ENTER-EXCLUSIVE-AREA-REFS/CAN-ENTER-EXCLUSIVE-AREA-REF	CAN-ENTER-EXCLUSIVE-AREA-REFS/CAN-ENTER-EXCLUSIVE-AREA-REF
DATA-READ-ACCESSSS/VARIABLE-ACCESS	DATA-READ-ACCESSSS/DATA-READ-ACCESS
DATA-RECEIVE-POINT-BY-ARGUMENTS/VARIABLE-ACCESS	DATA-RECEIVE-POINTS/DATA-RECEIVE-POINT
DATA-SEND-POINTS/VARIABLE-ACCESS	DATA-SEND-POINTS/DATA-SEND-POINT
DATA-WRITE-ACCESSSS/VARIABLE-ACCESS	DATA-WRITE-ACCESSSS/DATA-WRITE-ACCESS
MINIMUM-START-INTERVAL	MINIMUM-START-INTERVAL
MODE-SWITCH-POINTS/MODE-SWITCH-POINT	MODE-SWITCH-POINTS/MODE-SWITCH-POINT
PARAMETER-ACCESSSS/PARAMETER-ACCESS/ACCESSED-PARAMETER/LOCAL-PARAMETER-REF	PER-INSTANCE-CALPRM-ACCESS-REFS/PER-INSTANCE-CALPRM-ACCESS-REF
RUNS-INSIDE-EXCLUSIVE-AREA-REFS/RUNS-INSIDE-EXCLUSIVE-AREA-REF	RUNS-INSIDE-EXCLUSIVE-AREA-REFS/RUNS-INSIDE-EXCLUSIVE-AREA-REF

Target model property	Source model property
SERVER-CALL-POINTS/ASYNCHRONOUS-SERVER-CALL-POINT	SERVER-CALL-POINTS/ASYNCHRONOUS-SERVER-CALL-POINT
SERVER-CALL-POINTS/SYNCHRONOUS-SERVER-CALL-POINT	SERVER-CALL-POINTS/SYNCHRONOUS-SERVER-CALL-POINT
PARAMETER-ACCESSSS/PARAMETER-ACCESS/ACCESSED-PARAMETER/LOCAL-PARAMETER-REF	SHARED-CALPRM-ACCESS-REFS/SHARED-CALPRM-ACCESS-REF
SYMBOL	SYMBOL
WAIT-POINTS/WAIT-POINT	WAIT-POINTS/WAIT-POINT
WRITTEN-LOCAL-VARIABLES/VARIABLE-ACCESS/ACCESSED-VARIABLE/LOCAL-VARIABLE-REF	WRITTEN-VARIABLE-REFS/WRITTEN-VARIABLE-REF
READ-LOCAL-VARIABLES/VARIABLE-ACCESS/ACCESSED-VARIABLE/LOCAL-VARIABLE-REF	READ-VARIABLE-REFS/READ-VARIABLE-REF

Migration of source model DATA-RECEIVE-POINT elements

Please note that all source model DATA-RECEIVE-POINT elements are migrated to DATA-RECEIVE-POINT-BY-ARGUMENTS/VARIABLE-ACCESS elements in the target model, i.e. there is no conversion to the target model entities DATA-RECEIVE-POINT-BY-VALUES/VARIABLE-ACCESS.

Creation of target model ASYNCHRONOUS-SERVER-CALL-RESULT-POINT elements

Please note that for every source model ASYNCHRONOUS-SERVER-CALL-RETURNS-EVENT entity which either references a RUNNABLE-ENTITY via START-ON-EVENT-REF or references a SERVER-CALL-POINT of the RUNNABLE-ENTITY via EVENT-SOURCE-REF, one ASYNCHRONOUS-SERVER-CALL-RESULT-POINTS/ASYNCHRONOUS-SERVER-CALL-RESULT-POINT is created in the target model.

8.1.1.8.36. WAIT-POINT

Source model entity: WAIT-POINT

Target model entity: WAIT-POINT

Target model property	Source model property
TIMEOUT	TIMEOUT

Target model property	Source model property
TRIGGER-REF	TRIGGER-REF

8.1.1.8.37. SWC-INTERNAL-BEHAVIOR

Source model entity: INTERNAL-BEHAVIOR

Target model entity: SWC-INTERNAL-BEHAVIOR

Target model property	Source model property
No migration is required, see Section 8.1.1.8.2, “APPLICATION-SW-COMPONENT-TYPE” .	COMPONENT-REF
EVENTS	EVENTS
EXCLUSIVE-AREAS/EXCLUSIVE-AREA	EXCLUSIVE-AREAS/EXCLUSIVE-AREA
HANDLE-TERMINATION-AND-RESTART	No source model parameter is available, the target model parameter is set to NOSUPPORT.
PER-INSTANCE-PARAMETERS/PARAMETER-DATA-PROTOTYPE	PER-INSTANCE-CALPRMS/CALPRM-ELEMENT-PROTOTYPE
PER-INSTANCE-MEMORYS/PER-INSTANCE-MEMORY	PER-INSTANCE-MEMORYS/PER-INSTANCE-MEMORY
PORT-API-OPTIONS/PORT-API-OPTION	PORT-API-OPTIONS/PORT-API-OPTION
RUNNABLES/RUNNABLE-ENTITY	RUNNABLES/RUNNABLE-ENTITY
SHARED-PARAMETERS/PARAMETER-DATA-PROTOTYPE	SHARED-CALPRMS/CALPRM-ELEMENT-PROTOTYPE
SUPPORTS-MULTIPLE-INSTANTIATION	SUPPORTS-MULTIPLE-INSTANTIATION
AR-TYPED-PER-INSTANCE-MEMORYS/VARIABLE-DATA-PROTOTYPE	No source model parameter is available.
CONSTANT-MEMORYS/PARAMETER-DATA-PROTOTYPE	No source model parameter is available.
CONSTANT-VALUE-MAPPING-REFS/CONSTANT-VALUE-MAPPING-REF	No source model parameter is available.
INCLUDED-DATA-TYPE-SETS/INCLUDED-DATA-TYPE-SET	No source model parameter is available.
INSTANTIATION-DATA-DEF-PROPSS/INSTANTIATION-DATA-DEF-PROPS	No source model parameter is available.
STATIC-MEMORYS/VARIABLE-DATA-PROTOTYPE	No source model parameter is available.

Creation of target model PARAMETER-DATA-PROTOTYPE/INIT-VALUE for INIT-VALUES/LOCAL-PARAMETER-INIT-VALUE-ASSIGNMENT

Please note that for every source model INIT-VALUES/LOCAL-PARAMETER-INIT-VALUE-ASSIGNMENT one PARAMETER-DATA-PROTOTYPE/INIT-VALUE element is created for the target model counterpart of the referenced CALPRM-ELEMENT-PROTOTYPE.

Upgrade of source model INTER-RUNNABLE-VARIABLES/INTER-RUNNABLE-VARIABLE

If the source model value of INTER-RUNNABLE-VARIABLES/INTER-RUNNABLE-VARIABLE/COMMUNICATION-APPROACH is set to EXPLICIT, the content of INTER-RUNNABLE-VARIABLES/INTER-RUNNABLE-VARIABLE is upgraded to the target model parameter EXPLICIT-INTER-RUNNABLE-VARIABLES/VARIABLE-DATA-PROTOTYPE, in all other cases to IMPLICIT-INTER-RUNNABLE-VARIABLES/VARIABLE-DATA-PROTOTYPE. If the source model provides the parameter INTER-RUNNABLE-VARIABLES/INTER-RUNNABLE-VARIABLE/INIT-VALUE-REF, a corresponding parameter INTER-RUNNABLE-VARIABLES/INTER-RUNNABLE-VARIABLE/INIT-VALUE/CONSTANT-REF is set up in the target model.

Upgrade of source model SERVICE-NEEDSS

For every SERVICE-NEEDS instance below SERVICE-NEEDSS in the source model, one SWC-SERVICE-DEPENDENCY is created below SERVICE-DEPENDENCIES in the target model which in turn will contain the upgraded SERVICE-NEEDS instance.

Creation of target model DATA-TYPE-MAPPING-REFS/DATA-TYPE-MAPPING-REF

For every source model MODE-SWITCH-POINT aggregated by any RUNNABLE-ENTITY of the INTERNAL-BEHAVIOR, one DATA-TYPE-MAPPING-SET is created, which the target model INTERNAL-BEHAVIOR references in DATA-TYPE-MAPPING-REFS/DATA-TYPE-MAPPING-REF. See also [Section 8.1.1.8.82, “MODE-ACCESS-POINT”](#).

MODE-REQUEST-TYPE-MAP/IMPLEMENTATION-DATA-TYPE-REF references an IMPLEMENTATION-DATA-TYPE representing an unsigned eight bit data type. MODE-REQUEST-TYPE-MAP/MODE-GROUP-REF references the target model counterpart of the source model MODE-DECLARATION-GROUP element referenced by the MODE-DECLARATION-GROUP-PROTOTYPE, which in turn is referenced by the source model MODE-SWITCH-POINT.

8.1.1.8.38. PORT-API-OPTION

Source model entity: PORT-API-OPTION

Target model entity: PORT-API-OPTION

Target model property	Source model property
ENABLE-TAKE-ADDRESS	ENABLE-TAKE-ADDRESS
INDIRECT-API	INDIRECT-API
PORT-ARG-VALUES	PORT-ARG-VALUES
PORT-REF	PORT-REF

Upgrade of PORT-ARG-VALUES

For every source model element in PORT-ARG-VALUES, one PORT-DEFINED-ARGUMENT-VALUE is created in the target model. The source model elements are then aggregated within their target model PORT-DEFINED-ARGUMENT-VALUE elements. See [Section 8.1.1.8.49, “NUMERICAL-VALUE-SPECIFICATION”](#) and the following chapters for details.

8.1.1.8.39. DATA-SEND-COMPLETED-EVENT

Source model entity: DATA-SEND-COMPLETED-EVENT

Target model entity: DATA-SEND-COMPLETED-EVENT

Target model property	Source model property
EVENT-SOURCE-REF	EVENT-SOURCE-REF
DISABLED-MODE-IREFS/DISABLED-MODE-IREF	MODE-DEPENDENCY/DEPENDENT-ON-MODE-IREFS/DEPENDENT-ON-MODE-IREF
START-ON-EVENT-REF	START-ON-EVENT-REF

8.1.1.8.40. DATA-RECEIVED-EVENT

Source model entity: DATA-RECEIVED-EVENT

Target model entity: DATA-RECEIVED-EVENT

Target model property	Source model property
DATA-IREF	DATA-IREF
DISABLED-MODE-IREFS/DISABLED-MODE-IREF	MODE-DEPENDENCY/DEPENDENT-ON-MODE-IREFS/DEPENDENT-ON-MODE-IREF
START-ON-EVENT-REF	START-ON-EVENT-REF

8.1.1.8.41. DATA-RECEIVE-ERROR-EVENT

Source model entity: DATA-RECEIVE-ERROR-EVENT

Target model entity: DATA-RECEIVE-ERROR-EVENT

Target model property	Source model property
DATA-IREF	DATA-IREF
DISABLED-MODE-IREFS/DISABLED-MODE-IREF	MODE-DEPENDENCY/DEPENDENT-ON-MODE-IREFS/DEPENDENT-ON-MODE-IREF
START-ON-EVENT-REF	START-ON-EVENT-REF

8.1.1.8.42. ASYNCHRONOUS-SERVER-CALL-RETURNS-EVENT

Source model entity: ASYNCHRONOUS-SERVER-CALL-RETURNS-EVENT

Target model entity: ASYNCHRONOUS-SERVER-CALL-RETURNS-EVENT

Target model property	Source model property
DISABLED-MODE-IREFS/DISABLED-MODE-IREF	MODE-DEPENDENCY/DEPENDENT-ON-MODE-IREFS/DEPENDENT-ON-MODE-IREF
START-ON-EVENT-REF	START-ON-EVENT-REF

Upgrade of source model EVENT-SOURCE-REF

Please note that in the target model one ASYNCHRONOUS-SERVER-CALL-RESULT-POINT element is created for each ASYNCHRONOUS-SERVER-CALL-POINT referenced by the source model EVENT-SOURCE-REF. The target model EVENT-SOURCE-REF references the ASYNCHRONOUS-SERVER-CALL-RESULT-POINT.

8.1.1.8.43. TIMING-EVENT

Source model entity: TIMING-EVENT

Target model entity: TIMING-EVENT

Target model property	Source model property
DISABLED-MODE-IREFS/DISABLED-MODE-IREF	MODE-DEPENDENCY/DEPENDENT-ON-MODE-IREFS/DEPENDENT-ON-MODE-IREF
START-ON-EVENT-REF	START-ON-EVENT-REF
PERIOD	PERIOD

8.1.1.8.44. OPERATION-INVOKED-EVENT

Source model entity: OPERATION-INVOKED-EVENT

Target model entity: OPERATION-INVOKED-EVENT

Target model property	Source model property
DISABLED-MODE-IREFS/DISABLED-MODE-IREF	MODE-DEPENDENCY/DEPENDENT-ON-MODE-IREFS/DEPENDENT-ON-MODE-IREF
START-ON-EVENT-REF	START-ON-EVENT-REF
OPERATION-IREF	OPERATION-IREF

8.1.1.8.45. SWC-MODE-SWITCH-EVENT

Source model entity: MODE-SWITCH-EVENT

Target model entity: SWC-MODE-SWITCH-EVENT

Target model property	Source model property
DISABLED-MODE-IREFS/DISABLED-MODE-IREF	MODE-DEPENDENCY/DEPENDENT-ON-MODE-IREFS/DEPENDENT-ON-MODE-IREF
START-ON-EVENT-REF	START-ON-EVENT-REF
ACTIVATION	ACTIVATION
MODE-IREFS/MODE-IREF/CONTEXT-MODE-DECLARATION-GROUP-PROTOTYPE-REF	MODE-IREF/MODE-DECLARATION-GROUP-PROTOTYPE-REF
MODE-IREFS/MODE-IREF/TARGET-MODE-DECLARATION-REF	MODE-IREF/MODE-DECLARATION-REF
MODE-IREFS/MODE-IREF/CONTEXT-PORT-REF	MODE-IREF/R-PORT-PROTOTYPE-REF

8.1.1.8.46. MODE-SWITCHED-ACK-EVENT

Source model entity: MODE-SWITCHED-ACK-EVENT

Target model entity: MODE-SWITCHED-ACK-EVENT

Target model property	Source model property
DISABLED-MODE-IREFS/DISABLED-MODE-IREF	MODE-DEPENDENCY/DEPENDENT-ON-MODE-IREFS/DEPENDENT-ON-MODE-IREF
START-ON-EVENT-REF	START-ON-EVENT-REF

Target model property	Source model property
EVENT-SOURCE-REF	EVENT-SOURCE-REF

8.1.1.8.47. ASYNCHRONOUS-SERVER-CALL-POINT

Source model entity: ASYNCHRONOUS-SERVER-CALL-POINT

Target model entity: ASYNCHRONOUS-SERVER-CALL-POINT

Target model property	Source model property
OPERATION-IREF	OPERATION-IREFS/OPERATION-IREF
TIMEOUT	TIMEOUT

Upgrade of source model OPERATION-IREFS/OPERATION-IREF

Please note that only the first source model entity in OPERATION-IREFS/OPERATION-IREF is upgraded to the target model OPERATION-IREF since the target model allows at most one OPERATION-IREF element.

8.1.1.8.48. SYNCHRONOUS-SERVER-CALL-POINT

Source model entity: SYNCHRONOUS-SERVER-CALL-POINT

Target model entity: SYNCHRONOUS-SERVER-CALL-POINT

Target model property	Source model property
OPERATION-IREF	OPERATION-IREFS/OPERATION-IREF
TIMEOUT	TIMEOUT

Upgrade of source model OPERATION-IREFS/OPERATION-IREF

Please note that only the first source model entity in OPERATION-IREFS/OPERATION-IREF is upgraded to the target model OPERATION-IREF since the target model allows at most one OPERATION-IREF element.

8.1.1.8.49. NUMERICAL-VALUE-SPECIFICATION

Source model entities: INTEGER-LITERAL, REAL-LITERAL, BOOLEAN-LITERAL, OPAQUE-LITERAL, CHAR-LITERAL

Target model entity: NUMERICAL-VALUE-SPECIFICATION

Target model property	Source model property
SHORT-LABEL	SHORT-NAME
No target model parameter is available.	TYPE-TREF
VALUE	VALUE

Upgrade of source model VALUE

Please note that depending on the source model entity, the content of the source model `VALUE` is upgraded differently into the target model `VALUE`:

- ▶ **INTEGER-LITERALS:** No conversion is required.
- ▶ **REAL-LITERALS:** No conversion is required.
- ▶ **BOOLEAN-LITERALS:** `false` is converted to 0, `true` is converted to 1.
- ▶ **OPAQUE-LITERALS:** The opaque literal byte array is converted into an integral number. The first byte lies in the target model value MSB, the last byte in the target model LSB. Supported opaque literals are at most eight bytes long.
- ▶ **CHAR-LITERALS:** The value is converted directly. Please note that the model upgrade assumes that the source model value is provided as code which represents the character, it is not provided as character itself.

8.1.1.8.50. TEXT-VALUE-SPECIFICATION

Source model entity: STRING-LITERAL

Target model entity: TEXT-VALUE-SPECIFICATION

Target model property	Source model property
SHORT-LABEL	SHORT-NAME
No target model parameter is available.	TYPE-TREF
VALUE	VALUE

8.1.1.8.51. ARRAY-VALUE-SPECIFICATION

Source model entity: ARRAY-SPECIFICATION

Target model entity: ARRAY-VALUE-SPECIFICATION

Target model property	Source model property
SHORT-LABEL	SHORT-NAME

Target model property	Source model property
No target model parameter is available.	TYPE-TREF
ELEMENTS	ELEMENTS

8.1.1.8.52. RECORD-VALUE-SPECIFICATION

Source model entity: RECORD-SPECIFICATION

Target model entity: RECORD-VALUE-SPECIFICATION

Target model property	Source model property
SHORT-LABEL	SHORT-NAME
No target model parameter is available.	TYPE-TREF
FIELDS	ELEMENTS

8.1.1.8.53. CONSTANT-SPECIFICATION

Source model entity: CONSTANT-SPECIFICATION

Target model entity: CONSTANT-SPECIFICATION

Target model property	Source model property
SHORT-NAME	SHORT-NAME
VALUE-SPEC	VALUE

8.1.1.8.54. CONSTANT-REFERENCE

Source model entity: CONSTANT-REFERENCE

Target model entity: CONSTANT-REFERENCE

Target model property	Source model property
TYPE-TREF	TYPE-TREF
CONSTANT-REF	CONSTANT-REF

8.1.1.8.55. UNIT

Source model entity: UNIT

Target model entity: UNIT

Target model property	Source model property
FACTOR-SI-TO-UNIT	FACTOR-SI-TO-UNIT
OFFSET-SI-TO-UNIT	OFFSET-SI-TO-UNIT

No upgrade of source model DISPLAY-NAME and PHYSICAL-DIMENSION-REF

Due to internal data model restrictions, the source model parameters DISPLAY-NAME and PHYSICAL-DIMENSION-REF are not upgraded to the target model.

8.1.1.8.56. UNIT-GROUP

Due to internal data model restrictions, there is no upgrade path for UNIT-GROUP elements.

8.1.1.8.57. SW-BASE-TYPE

Source model entity: SW-BASE-TYPE

Target model entity: SW-BASE-TYPE

Target model property	Source model property
MEM-ALIGNMENT	MEM-ALIGNMENT
BASE-TYPE-SIZE	BASE-TYPE-SIZE

Upgrade of source model BASE-TYPE-ENCODING

If the source model BASE-TYPE-ENCODING contains one of the values

- ▶ 1C
- ▶ 2C
- ▶ BCD-P
- ▶ BCD-UP
- ▶ DSP-FRACTIONAL
- ▶ SM
- ▶ IEEE754

- ▶ ISO-8859-1
- ▶ ISO-8859-2
- ▶ WINDOWS-1252
- ▶ UTF-8
- ▶ UCS-2
- ▶ NONE

the value is directly copied into the target model parameter `BASE-TYPE-ENCODING`. For any other value in the source model, `NONE` will be written into the target model parameter.

Upgrade of source model `CATEGORY`

If the source model `CATEGORY` value is either `SW_FIXED_LENGTH` or `FIXED_LENGTH`, the target model `CATEGORY` parameter will contain the value `FIXED_LENGTH`. If the source model parameter value contains `SW_VARIABLE_LENGTH` or `VARIABLE_LENGTH`, the target model `CATEGORY` parameter value will contain `VARIABLE_LENGTH`.

No upgrade of source model `MAX-BASE-TYPE-SIZE` and `BASE-TYPE-REF`

Due to internal data model restrictions, the source model parameters `MAX-BASE-TYPE-SIZE` and `BASE-TYPE-REF` are not upgraded into the target model.

No upgrade of source model `BYTE-ORDER`

The source model parameter `BYTE-ORDER` is not upgraded into the target model because `BYTE-ORDER` is not a property of a (base) data type.

8.1.1.8.58. SW-DATA-DEF-PROPS

Source model entity: `SW-DATA-DEF-PROPS`

Target model entity: `SW-DATA-DEF-PROPS`

Target model property	Source model property
<code>SW-DATA-DEF-PROPS-CONDITIONAL/BASE-TYPE-REF</code>	<code>BASE-TYPE-REF</code>
<code>SW-DATA-DEF-PROPS-CONDITIONAL/COMPU-METHOD-REF</code>	<code>COMPU-METHOD-REF</code>

Target model property	Source model property
SW-DATA-DEF-PROPS-CONDITIONAL/IN-VALID-VALUE	INVALID-VALUE
SW-DATA-DEF-PROPS-CONDITIONAL/SW-CALIBRATION-ACCESS	SW-CALIBRATION-ACCESS
SW-DATA-DEF-PROPS-CONDITIONAL/SW-IMPL-POLICY	SW-IMPL-POLICY
SW-DATA-DEF-PROPS-CONDITIONAL/UNIT-REF	UNIT-REF
SW-DATA-DEF-PROPS-CONDITIONAL/IN-VALID-VALUE	INVALID-VALUE

No upgrade of source model parameters DATA-CONSTR-REF, DISPLAY-FORMAT, SW-CAL-PRM-AXIS-SET, SW-CLASS-ATTR-IMPL-REF, SW-CLASS-REF, SW-CODE-SYNTAX-REF, SW-DATA-DEPENDENCY, SW-HOST-VARIABLE, SW-POINTER, SW-RECORD-LAYOUT-REF, SW-TEXT-PROPS, SW-VALUE-BLOCK-SIZE, SW-VARIABLE-ACCESS-IMPL-POLICY

The source model parameters DATA-CONSTR-REF, DISPLAY-FORMAT, SW-CALPRM-AXIS-SET, SW-CLASS-ATTR-IMPL-REF, SW-CLASS-REF, SW-CODE-SYNTAX-REF, SW-DATA-DEPENDENCY, SW-HOST-VARIABLE, SW-POINTER, SW-RECORD-LAYOUT-REF, SW-TEXT-PROPS, SW-VALUE-BLOCK-SIZE, SW-VARIABLE-ACCESS-IMPL-POLICY are not upgraded to the target model.

No configuration of target model parameters SW-ALIGNMENT, SW-COMPARISON-VARIABLES, SW-DATA-DEPENDENCY, SW-HOST-VARIABLE, SW-INTENDED-RESOLUTION, SW-INTERPOLATION-METHOD, SW-IS-VIRTUAL, SW-POINTER-TARGET-PROPS, SW-REFRESH-TIMING, SW-TEXT-PROPS, VALUE-AXIS-DATA-TYPE-REF

The target model parameters SW-ALIGNMENT, SW-COMPARISON-VARIABLES, SW-DATA-DEPENDENCY, SW-HOST-VARIABLE, SW-INTENDED-RESOLUTION, SW-INTERPOLATION-METHOD, SW-IS-VIRTUAL, SW-POINTER-TARGET-PROPS, SW-REFRESH-TIMING, SW-TEXT-PROPS, VALUE-AXIS-DATA-TYPE-REF are not configured during the model upgrade because there is no source model counterpart.

Configuration of SW-DATA-DEF-PROPS-CONDITIONAL/SW-IMPL-POLICY

If the source model SW-DATA-DEF-PROPS element is aggregated within a DATA-ELEMENT-PROTOTYPE which has its IS-QUEUED value set to true, the upgraded SW-DATA-DEF-PROPS-CONDITIONAL/SW-IMPL-POLICY contains the value QUEUED, else it contains the SW-IMPL-POLICY value of the source model SW-DATA-DEF-PROPS counterpart. If the source model SW-DATA-DEF-PROPS value is MESSAGE, the parameter is not configured at all in the target model since it does not support this value.

Creation of SW-DATA-DEF-PROPS elements in the target model

For every target model ARGUMENT-DATA-PROTOTYPE, PARAMETER-DATA-PROTOTYPE, or VARIABLE-DATA-PROTOTYPE which has a source model counterpart which has its IS-QUEUED parameter value set to true but does not aggregate an SW-DATA-DEF-PROPS element, a dedicated SW-DATA-DEF-PROPS is created in the target model which has its SW-IMPL-POLICY value set to QUEUED.

8.1.1.8.59. IMPLEMENTATION-DATA-TYPE

Source model entities: ARRAY-TYPE, BOOLEAN-TYPE, CHAR-TYPE, INTEGER-TYPE, OPAQUE-TYPE, REAL-TYPE, RECORD-TYPE, STRING-TYPE

Target model entity: IMPLEMENTATION-DATA-TYPE

Creation of AUTOSAR 4.0 platform data types

Before any source model data type is upgraded into the target model, the AUTOSAR 4.0 platform data types are created, see [Section 8.1.4, “AUTOSAR 4.0 platform data types”](#) for details.

Configuration of target model IMPLEMENTATION-DATA-TYPE/CATEGORY platform data types

If the source model element is an ARRAY-TYPE or a STRING-TYPE, CATEGORY is set to ARRAY. For RECORD-TYPES, CATEGORY is set to STRUCTURE. In all other cases, CATEGORY is set to TYPE_REFERENCE.

Configuration of target model DATA-CONSTR

If the source model data type element contains a LOWER-LIMIT or an UPPER-LIMIT parameter, one DATA-CONSTR element is created in the target model which contains the limits. The target model IMPLEMENTATION-DATA-TYPE element references the DATA-CONSTR via SW-DATA-DEF-PROPS/SW-DATA-DEF-PROPS-VARIANTS/SW-DATA-DEF-PROPS-CONDITIONAL/DATA-CONSTR-REF.

Assignment of AUTOSAR 4.0 platform data type to target model IMPLEMENTATION-DATA-TYPE

If the source model data type element is one of BOOLEAN-TYPE, CHAR-TYPE, INTEGER-TYPE, OPAQUE-TYPE, or REAL-TYPE, a matching AUTOSAR 4.0 platform data type is determined according to the source model data type properties LOWER-LIMIT, UPPER-LIMIT, ENCODING, NUMBER-OF-BITS. The platform data type is referenced via SW-DATA-DEF-PROPS/SW-DATA-DEF-PROPS-VARIANTS/SW-DATA-DEF-PROPS-CONDITIONAL/IMPLEMENTATION-DATA-TYPE-REF

ARRAY-TYPE

If the source model data type is an ARRAY-TYPE, the following additional mapping rules apply:

Target model property	Source model property
SUB-ELEMENTS/IMPLEMENTATION-DATA-TYPE-ELEMENT/SHORT-NAME	ELEMENT/SHORT-NAME
SUB-ELEMENTS/IMPLEMENTATION-DATA-TYPE-ELEMENT/ARRAY-SIZE	ELEMENT/MAX-NUMBER-OF-ELEMENTS

The parameter SUB-ELEMENTS/IMPLEMENTATION-DATA-TYPE-ELEMENT/ARRAY-SIZE-SEMANTICS is set to FIXED-SIZE.

RECORD-TYPE

If the source model data type is a RECORD-TYPE, the following additional mapping rules apply:

Target model property	Source model property
SUB-ELEMENTS/IMPLEMENTATION-DATA-TYPE-ELEMENT	ELEMENTS/RECORD-ELEMENT

8.1.1.8.60. COMPU-METHOD

Source model entity: COMPU-METHOD

Target model entity: COMPU-METHOD

Target model property	Source model property
COMPU-INTERNAL-TO-PHYS/COMPU-SCALES/COMPU-SCALE	COMPU-INTERNAL-TO-PHYS/COMPU-SCALES/COMPU-SCALE
COMPU-PHYS-TO-INTERNAL/COMPU-SCALES/COMPU-SCALE	COMPU-PHYS-TO-INTERNAL/COMPU-SCALES/COMPU-SCALE
DISPLAY-FORMAT	DISPLAY-FORMAT
UNIT-REF	UNIT-REF

8.1.1.8.61. COMPU-SCALE

Source model entity: COMPU-SCALE

Target model entity: COMPU-SCALE

Target model property	Source model property
LOWER-LIMIT	LOWER-LIMIT
UPPER-LIMIT	UPPER-LIMIT
COMPU-CONST	COMPU-CONST
COMPU-INVERSE-VALUE	No source model parameter is available.
COMPU-RATIONAL-COEFFS / COMPU-NUMERATOR	COMPU-RATIONAL-COEFFS / COMPU-NUMERATOR
COMPU-RATIONAL-COEFFS / COMPU-DENOMINATOR	COMPU-RATIONAL-COEFFS / COMPU-DENOMINATOR
MASK	No source model parameter is available.
SHORT-LABEL	SHORT-LABEL
SYMBOL	No source model parameter is available.

8.1.1.8.62. EXCLUSIVE-AREA

Source model entity: EXCLUSIVE-AREA

Target model entity: EXCLUSIVE-AREA

Apart from the standard parameters SHORT-NAME, CATEGORY, and UUID, no further parameters are upgraded.

8.1.1.8.63. COM-MGR-USER-NEEDS

Source model entities: COM-MGR-USER-NEEDS, SWC-COM-MGR-USER-NEEDS

Target model entity: COM-MGR-USER-NEEDS

Target model property	Source model property
MAX-COMM-MODE	MAX-COMM-MODE

No upgrade of source model SWC-COM-MGR-USER-NEEDS/CALLBACK-PORTS and SWC-COM-MGR-USER-NEEDS/SERVICE-CALL-PORTS

The source model parameters SWC-COM-MGR-USER-NEEDS/CALLBACK-PORTS and SWC-COM-MGR-USER-NEEDS/SERVICE-CALL-PORTS are not upgraded into the target model because there are no corresponding parameters.

8.1.1.8.64. DIAGNOSTIC-COMMUNICATION-MANAGER-NEEDS

Source model entities: DIAGNOSTIC-COMMUNICATION-NEEDS, SWC-DIAGNOSTIC-COMMUNICATION-NEEDS

Target model entity: DIAGNOSTIC-COMMUNICATION-MANAGER-NEEDS

Apart from the standard parameters SHORT-NAME, CATEGORY, and UUID, no further parameters are upgraded.

No upgrade of source model SWC-DIAGNOSTIC-COMMUNICATION-NEEDS/CALLBACK-PORTS and SWC-DIAGNOSTIC-COMMUNICATION-NEEDS/SERVICE-CALL-PORT

The source model parameters SWC-DIAGNOSTIC-COMMUNICATION-NEEDS/CALLBACK-PORTS and SWC-DIAGNOSTIC-COMMUNICATION-NEEDS/SERVICE-CALL-PORT are not upgraded into the target model because there are no corresponding parameters.

8.1.1.8.65. DIAGNOSTIC-EVENT-NEEDS

Source model entities: DIAGNOSTIC-EVENT-NEEDS, SWC-DIAGNOSTIC-EVENT-NEEDS

Target model entity: DIAGNOSTIC-EVENT-NEEDS

Apart from the standard parameters SHORT-NAME, CATEGORY, and UUID, no further parameters are upgraded.

No upgrade of source model SWC-DIAGNOSTIC-EVENT-NEEDS/CALLBACK-PORTS and SWC-DIAGNOSTIC-EVENT-NEEDS/SERVICE-CALL-PORTS

The source model parameters SWC-DIAGNOSTIC-EVENT-NEEDS/CALLBACK-PORTS and SWC-DIAGNOSTIC-EVENT-NEEDS/SERVICE-CALL-PORTS are not upgraded into the target model because there are no corresponding parameters.

No configuration of target model parameters CONSIDER-PTO-STATUS, DIAG-EVENT-DEBOUNCE-ALGORITHM, DIAG-REQUIREMENT, DTC-KIND, DTC-NUMBER, INHIBITING-FID-REF

The target model parameters CONSIDER-PTO-STATUS, DIAG-EVENT-DEBOUNCE-ALGORITHM, DIAG-REQUIREMENT, DTC-KIND, DTC-NUMBER, INHIBITING-FID-REF are not configured during the model upgrade because there is no source model counterpart.

8.1.1.8.66. ECU-STATE-MGR-USER-NEEDS

Source model entities: ECU-STATE-MGR-USER-NEEDS, SWC-ECU-STATE-MGR-USER-NEEDS

Target model entity: ECU-STATE-MGR-USER-NEEDS

Apart from the standard parameters SHORT-NAME, CATEGORY, and UUID, no further parameters are upgraded.

No upgrade of source model SWC-ECU-STATE-MGR-USER-NEEDS/SERVICE-CALL-PORTS

The source model parameter SWC-ECU-STATE-MGR-USER-NEEDS/SERVICE-CALL-PORTS is not upgraded into the target model because there is no corresponding parameter.

No configuration of target model parameters CONSIDER-PTO-STATUS, DIAG-EVENT-DEBOUNCE-ALGORITHM, DIAG-REQUIREMENT, DTC-KIND, DTC-NUMBER, INHIBITING-FID-REF

The target model parameters CONSIDER-PTO-STATUS, DIAG-EVENT-DEBOUNCE-ALGORITHM, DIAG-REQUIREMENT, DTC-KIND, DTC-NUMBER, INHIBITING-FID-REF are not configured during the model upgrade because there is no source model counterpart.

8.1.1.8.67. FUNCTION-INHIBITION-NEEDS

Source model entities: FUNCTION-INHIBITION-NEEDS, SWC-FUNCTION-INHIBITION-NEEDS

Target model entity: FUNCTION-INHIBITION-NEEDS

Apart from the standard parameters SHORT-NAME, CATEGORY, and UUID, no further parameters are upgraded.

No upgrade of source model SWC-FUNCTION-INHIBITION-NEEDS/SERVICE-CALL-PORTS

The source model parameter SWC-FUNCTION-INHIBITION-NEEDS/SERVICE-CALL-PORTS is not upgraded into the target model because there is no corresponding parameter.

8.1.1.8.68. NV-BLOCK-NEEDS

Source model entities: BLOCK-NEEDS, SWC-NV-BLOCK-NEEDS

Target model entity: NV-BLOCK-NEEDS

Target model property	Source model property
N-DATA-SETS	(SWC-) NV-BLOCK-NEEDS/N-DATA-SETS

Target model property	Source model property
READONLY	(SWC-) NV-BLOCK-NEEDS/READONLY
RESISTANT-TO-CHANGED-SW	(SWC-) NV-BLOCK-NEEDS/RESISTANT-TO-CHANGED-SW
RESTORE-AT-START	(SWC-) NV-BLOCK-NEEDS/RESTORE-AT-START
WRITE-ONLY-ONCE	(SWC-) NV-BLOCK-NEEDS/WRITE-ONLY-ONCE
WRITING-FREQUENCY	(SWC-) NV-BLOCK-NEEDS/WRITING-FREQUENCY
WRITING-PRIORITY	(SWC-) NV-BLOCK-NEEDS/WRITING-PRIORITY

No upgrade of source model SWC-NV-BLOCK-NEEDS/CALL-BACK-PORTS, SWC-NV-BLOCK-NEEDS/DEFAULT-BLOCK-REF, SWC-NV-BLOCK-NEEDS/MIRROR-BLOCK-REF, SWC-NV-BLOCK-NEEDS/SERVICE-CALL-PORTS

The source model parameters SWC-NV-BLOCK-NEEDS/CALL-BACK-PORTS, SWC-NV-BLOCK-NEEDS/DEFAULT-BLOCK-REF, SWC-NV-BLOCK-NEEDS/MIRROR-BLOCK-REF, SWC-NV-BLOCK-NEEDS/SERVICE-CALL-PORTS are not upgraded into the target model because there are no corresponding parameters.

No configuration of target model parameters CALC-RAM-BLOCK-CRC, CHECK-STATIC-BLOCK-ID, N-ROM-BLOCKS, STORE-AT-SHUTDOWN, WRITE-VERIFICATION

The target model parameters CALC-RAM-BLOCK-CRC, CHECK-STATIC-BLOCK-ID, N-ROM-BLOCKS, STORE-AT-SHUTDOWN, WRITE-VERIFICATION are not configured during the model upgrade because there is no source model counterpart.

No configuration of target model parameter RELIABILITY

The source model parameter RELIABILITY is not upgraded to the target model parameter RELIABILITY because the parameter values are not related.

Additional configuration of ASSIGNED-DATAS/ROLE-BASED-DATA-ASSIGNMENT for SWC-NV-BLOCK-NEEDS elements

For every SWC-NV-BLOCK-NEEDS element in the source model which either contains a valid DE-FAULT-BLOCK_REF or a valid MIRROR-BLOCK-REF, one ASSIGNED-DATAS/ROLE-BASED-DATA-ASSIGNMENT element is created in the target model. The ROLE-BASED-DATA-ASSIGNMENT element is aggregated

within the target model SWC-SERVICE-DEPENDENCY element which also aggregates the target model SWC-NV-BLOCK-NEEDS element. ROLE-BASED-DATA-ASSIGNMENT/ROLE is set to ramMirror. ROLE-BASED-DATA-ASSIGNMENT/USED-PIM-REF references the target model counterpart of the source model PER-INSTANCE-MEMORY which is referenced by SWC-NV-BLOCK-NEEDS/MIRROR-BLOCK-REF. ROLE-BASED-DATA-ASSIGNMENT/USED-PARAMETER-ELEMENT aggregates a AUTOSAR-PARAMETER-REF element which in turn references in LOCAL-PARAMETER-REF the target model counterpart of the source model CALPRM-ELEMENT-PROTOTYPE which is referenced by SWC-NV-BLOCK-NEEDS/DEFAULT-BLOCK-REF.

8.1.1.8.69. OBD-CONTROL-SERVICE-NEEDS

Source model entities: OBD-CONTROL-SERVICE-NEEDS, SWC-OBD-CONTROL-SERVICE-NEEDS

Target model entity: OBD-CONTROL-SERVICE-NEEDS

Target model property	Source model property
TEST-ID	OBD-CONTROL-SERVICE-NEEDS/TEST-ID

No upgrade of source model SWC-OBD-CONTROL-SERVICE-NEEDS/CALLBACK-PORT

The source model parameter SWC-OBD-CONTROL-SERVICE-NEEDS/CALLBACK-PORT is not upgraded into the target model because there is no corresponding parameter.

No configuration of target model parameters AUDIENCES, DIAG-REQUIREMENT, SECURITY-ACCESS-LEVEL

The target model parameters AUDIENCES, DIAG-REQUIREMENT, SECURITY-ACCESS-LEVEL are not configured during the model upgrade because there is no source model counterpart.

8.1.1.8.70. OBD-INFO-SERVICE-NEEDS

Source model entities: OBD-INFO-SERVICE-NEEDS, SWC-OBD-INFO-SERVICE-NEEDS

Target model entity: OBD-INFO-SERVICE-NEEDS

Target model property	Source model property
DATA-LENGTH	OBD-INFO-SERVICE-NEEDS/DATA-LENGTH
INFO-TYPE	OBD-INFO-SERVICE-NEEDS/INFO-TYPE

No upgrade of source model SWC-OBD-INFO-SERVICE-NEEDS/CALLBACK-PORT

The source model parameter **SWC-OBD-INFO-SERVICE-NEEDS/CALLBACK-PORT** is not upgraded into the target model because there is no corresponding parameter.

No configuration of target model parameters AUDIENCES, DIAG-REQUIREMENT, SECURITY-ACCESS-LEVEL

The target model parameters **AUDIENCES, DIAG-REQUIREMENT, SECURITY-ACCESS-LEVEL** are not configured during the model upgrade because there is no source model counterpart.

8.1.1.8.71. OBD-PID-SERVICE-NEEDS

Source model entities: **OBD-PID-SERVICE-NEEDS, SWC-OBD-PID-SERVICE-NEEDS**

Target model entity: **OBD-PID-SERVICE-NEEDS**

Target model property	Source model property
DATA-LENGTH	OBD-PID-SERVICE-NEEDS/DATA-LENGTH
PARAMETER-ID	OBD-PID-SERVICE-NEEDS/PARAMETER-ID
STANDARD	OBD-PID-SERVICE-NEEDS/STANDARD

No upgrade of source model SWC-OBD-PID-SERVICE-NEEDS/CALLBACK-PORT

The source model parameter **SWC-OBD-PID-SERVICE-NEEDS/CALLBACK-PORT** is not upgraded into the target model because there is no corresponding parameter.

No configuration of target model parameters AUDIENCES, DIAG-REQUIREMENT, SECURITY-ACCESS-LEVEL

The target model parameters **AUDIENCES, DIAG-REQUIREMENT, SECURITY-ACCESS-LEVEL** are not configured during the model upgrade because there is no source model counterpart.

8.1.1.8.72. OBD-RATIO-SERVICE-NEEDS

Source model entities: **OBD-RATIO-SERVICE-NEEDS, SWC-OBD-RATIO-SERVICE-NEEDS**

Target model entity: OBD-RATIO-SERVICE-NEEDS

Target model property	Source model property
CONNECTION-TYPE	OBD-RATIO-SERVICE-NEEDS/CONNECTION-TYPE
IUMPR-GROUP	OBD-RATIO-SERVICE-NEEDS/IUMPR-GROUP
RATE-BASED-MONITORED-EVENT-REF	OBD-RATIO-SERVICE-NEEDS/RATE-BASED-MONITORED-EVENT-REF
USED-FID-REF	OBD-RATIO-SERVICE-NEEDS/USED-FID-REF

No upgrade of source model SWC-OBD-RATIO-SERVICE-NEEDS/SERVICE-CALL-PORT

The source model parameter SWC-OBD-RATIO-SERVICE-NEEDS/SERVICE-CALL-PORT is not upgraded into the target model because there is no corresponding parameter.

No configuration of target model parameters AUDIENCES, DIAG-REQUIREMENT, SECURITY-ACCESS-LEVEL

The target model parameters AUDIENCES, DIAG-REQUIREMENT, SECURITY-ACCESS-LEVEL are not configured during the model upgrade because there is no source model counterpart.

8.1.1.8.73. SUPERVISED-ENTITY-NEEDS

Source model entities: SUPERVISED-ENTITY-NEEDS, SWC-SUPERVISED-ENTITY-NEEDS

Target model entity: SUPERVISED-ENTITY-NEEDS

Target model property	Source model property
ACTIVATE-AT-START	SUPERVISED-ENTITY-NEEDS/ACTIVATE-AT-START
ENABLE-DEACTIVATION	SUPERVISED-ENTITY-NEEDS/ENABLE-DEACTIVATION
EXPECTED-ALIVE-CYCLE	SUPERVISED-ENTITY-NEEDS/EXPECTED-ALIVE-CYCLE
MAX-ALIVE-CYCLE	SUPERVISED-ENTITY-NEEDS/MAX-ALIVE-CYCLE
MIN-ALIVE-CYCLE	SUPERVISED-ENTITY-NEEDS/MIN-ALIVE-CYCLE
TOLERATED-FAILED-CYCLES	SUPERVISED-ENTITY-NEEDS/TOLERATED-FAILED-CYCLES

No configuration of target model parameters AUDIENCES, DIAG-REQUIREMENT, SECURITY-ACCESS-LEVEL

The target model parameters AUDIENCES, DIAG-REQUIREMENT, SECURITY-ACCESS-LEVEL are not configured during the model upgrade because there is no source model counterpart.

8.1.1.8.74. PER-INSTANCE-MEMORY

Source model entity: PER-INSTANCE-MEMORY

Target model entity: PER-INSTANCE-MEMORY

Target model property	Source model property
TYPE	TYPE
TYPE-DEFINITION	TYPE-DEFINITION

Configuration of target model parameter INIT-VALUE

The target model parameter INIT-VALUE is always set to 0 because there is no corresponding source model value.

8.1.1.8.75. PER-INSTANCE-MEMORY-SIZE

Source model entity: PER-INSTANCE-MEMORY-SIZE

Target model entity: PER-INSTANCE-MEMORY-SIZE

Target model property	Source model property
ALIGNMENT	ALIGNMENT
PER-INSTANCE-MEMORY-REF	PER-INSTANCE-MEMORY-REF
SIZE	SIZE

8.1.1.8.76. SWC-IMPLEMENTATION

Source model entity: SWC-IMPLEMENTATION

Target model entity: SWC-IMPLEMENTATION

Target model property	Source model property
BEHAVIOR-REF	BEHAVIOR-REF
PER-INSTANCE-MEMORY-SIZES/PER-INSTANCE-MEMORY-SIZE	PER-INSTANCE-MEMORY-SIZES/PER-INSTANCE-MEMORY-SIZE
REQUIRED-RTE-VENDOR	REQUIRED-RTE-VENDOR
CODE-DESCRIPTORS/CODE	CODE-DESCRIPTORS/CODE
USED-CODE-GENERATOR	CODE-GENERATOR
PROGRAMMING-LANGUAGE	PROGRAMMING-LANGUAGE
SW-VERSION	SW-MAJOR-VERSION.SW-MINOR-VERSION.SW-PATCH-VERSION
VENDOR-ID	VENDOR-ID

No upgrade of source model parameters COMPILERS/COMPILER, IMPLEMENTATION-DEPENDENCIES/DEPENDENCY-ON-FILE, IMPLEMENTATION-DEPENDENCIES/DEPENDENCY-ON-LIBRARY, LINKERS/LINKER, PROCESSOR-REFS/PROCESSOR-REF, RESOURCE-CONSUMPTION

The source model parameters COMPILERS/COMPILER, IMPLEMENTATION-DEPENDENCIES/DEPENDENCY-ON-FILE, IMPLEMENTATION-DEPENDENCIES/DEPENDENCY-ON-LIBRARY, LINKERS/LINKER, PROCESSOR-REFS/PROCESSOR-REF, RESOURCE-CONSUMPTION are not upgraded into the target model.

Upgrade of source model parameter PROGRAMMING-LANGUAGE

If the source model parameter PROGRAMMING-LANGUAGE is not defined, C is configured as default value in the target model.

Upgrade of source model parameter VENDOR-ID

If the source model parameter VENDOR-ID is not defined, 1 is configured as default value in the target model.

Creation of target model RESOURCE-CONSUMPTION

For each SWC-IMPLEMENTATION in the target model, one RESOURCE-CONSUMPTION sub element is set up.

8.1.1.8.77. CODE

Source model entity: CODE

Target model entity: CODE

Apart from the standard parameter SHORT-NAME, no further parameter is directly upgraded into the target model entity.

Configuration of source model parameter TYPE

Every target model CODE element aggregates an ARTIFACT-DESCRIPTORS/AUTOSAR-ENGINEERING-OBJECT sub element. Its SHORT-NAME is set to EngObject. The source model parameter TYPE is upgraded into ARTIFACT-DESCRIPTORS/AUTOSAR-ENGINEERING-OBJECT/CATEGORY. If the source model value is SRC, the target model value is set to SWSRC. If the source model value is OBJ, the target model value is set to SWOBJ.

8.1.1.8.78. MODE-DECLARATION

Source model entity: MODE-DECLARATION

Target model entity: MODE-DECLARATION

Apart from the standard parameters SHORT-NAME, CATEGORY, and UUID, no further parameters are upgraded.

8.1.1.8.79. MODE-DECLARATION-GROUP

Source model entity: MODE-DECLARATION-GROUP

Target model entity: MODE-DECLARATION-GROUP

Target model property	Source model property
INITIAL-MODE-REF	INITIAL-MODE-REF
MODE-DECLARATIONS/MODE-DECLARATION	MODE-DECLARATIONS/MODE-DECLARATION

8.1.1.8.80. DISABLED-MODE-IREF

Source model entity: DEPENDENT-ON-MODE-IREF

Target model entity: DISABLED-MODE-IREF

Target model property	Source model property
CONTEXT-PORT-REF	R-PORT-PROTOTYPE-REF
CONTEXT-MODE-DECLARATION-GROUP-PROTO-TYPE-REF	MODE-DECLARATION-GROUP-PROTOTYPE-REF

Target model property	Source model property
MODE-DECLARATION-REF	MODE-DECLARATION-REF

8.1.1.8.81. MODE-SWITCHED-ACK

Source model entity: MODE-SWITCHED-ACK

Target model entity: MODE-SWITCHED-ACK

Target model property	Source model property
TIMEOUT	TIMEOUT

8.1.1.8.82. MODE-ACCESS-POINT

Target model MODE-ACCESS-POINT elements are created out of source models in two scenarios.

Configuration MODE-ACCESS-POINT out of source model R-PORT-PROTOTYPE elements

For every R-PORT-PROTOTYPE referencing a SENDER-RECEIVER-INTERFACE which aggregates a MODE-DECLARATION-GROUP-PROTOTYPE element, one MODE-ACCESS-POINT element is created in every target model RUNNABLE-ENTITY element related to the target model ATOMIC-SW-COMPONENT which owns the target model counterpart of the source model R-PORT-PROTOTYPE.

In this case, MODE-ACCESS-POINT/MODE-GROUP-IREF/R-MODE-GROUP-IN-ATOMIC-SWC-INSTANCE-REF/CONTEXT-R-PORT-REF contains a reference to the target model counterpart of the source model R-PORT-PROTOTYPE and MODE-ACCESS-POINT/MODE-GROUP-IREF/R-MODE-GROUP-IN-ATOMIC-SWC-INSTANCE-REF/TARGET-MODE-GROUP-REF references the target model counterpart of the source model MODE-DECLARATION-GROUP-PROTOTYPE.

Configuration MODE-ACCESS-POINT out of source model MODE-SWITCH-POINT elements

For every source model MODE-SWITCH-POINT aggregated in a RUNNABLE-ENTITY, one MODE-ACCESS-POINT is created in the target model. The target model MODE-ACCESS-POINT is aggregated in the target model counterpart of the source model RUNNABLE-ENTITY aggregating the source model RUNNABLE-ENTITY.

In this case, MODE-ACCESS-POINT/MODE-GROUP-IREF/P-MODE-GROUP-IN-ATOMIC-SWC-INSTANCE-REF/CONTEXT-P-PORT-REF contains a reference to the target model counterpart of the source model P-PORT-PROTOTYPE referenced by the source model MODE-SWITCH-POINT. MODE-ACCESS-POINT/MODE-GROUP-IREF/P-MODE-GROUP-IN-ATOMIC-SWC-INSTANCE-REF/TARGET-MODE-GROUP-REF references

the target model counterpart of the source model MODE-DECLARATION-GROUP-PROTOTYPE referenced by the source model MODE-SWITCH-POINT.

8.1.1.8.83. MODE-SWITCH-POINT

Source model entity: MODE-SWITCH-POINT

Target model entity: MODE-SWITCH-POINT

Target model property	Source model property
MODE-GROUP-IREF/CONTEXT-P-PORT-REF	MODE-GROUP-IREF/P-PORT-PROTOTYPE-REF
MODE-GROUP-IREF/TARGET-MODE-GROUP-REF	MODE-GROUP-IREF/MODE-DECLARATION-GROUP-PROTOTYPE-REF

8.1.1.8.84. SW-ADDR-METHOD

Source model entity: SW-ADDR-METHOD

Target model entity: SW-ADDR-METHOD

Apart from the standard parameters SHORT-NAME, CATEGORY, and UUID, no further parameters are upgraded.

8.1.1.8.85. MEMORY-SECTION

Source model entity: MEMORY-SECTION

Target model entity: MEMORY-SECTION

Target model property	Source model property
ALIGNMENT	ALIGNMENT
SIZE	SIZE

Configuration of source model parameters SW-ADDR-METHOD-REFS/SW-ADDR-METHOD-REF

Since the target model MEMORY-SECTION only supports one single SW-ADDRMETHOD-REF, only the first source model SW-ADDR-METHOD-REFS / SW-ADDR-METHOD-REF is upgraded in the target model.

8.1.1.8.86. SYSTEM-MAPPING

Source model entity: SYSTEM-MAPPING

Target model entity: SYSTEM-MAPPING

Target model property	Source model property
DATA-MAPPINGS/CLIENT-SERVER-TO-SIGNAL-GROUP-MAPPING	DATA-MAPPINGS/CLIENT-SERVER-TO-SIGNAL-GROUP-MAPPING
DATA-MAPPINGS/SENDER-RECEIVER-TO-SIGNAL-GROUP-MAPPING	DATA-MAPPINGS/SENDER-RECEIVER-TO-SIGNAL-GROUP-MAPPING
DATA-MAPPINGS/SENDER-RECEIVER-TO-SIGNAL-MAPPING	DATA-MAPPINGS/SENDER-RECEIVER-TO-SIGNAL-MAPPING
SW-MAPPINGS/SWC-TO-ECU-MAPPING	SW-MAPPINGS/SWC-TO-ECU-MAPPING
SW-IMPL-MAPPINGS/SWC-TO-IMPL-MAPPING	SW-IMPL-MAPPINGS/SWC-TO-IMPL-MAPPING

No upgrade of source model parameters ECU-RESOURCE-MAPPINGS, MAPPING-CONSTRAINTS, RESOURCE-ESTIMATIONS and SIGNAL-PATH-CONSTRAINTS

Due to internal data model restrictions, the source model parameters ECU-RESOURCE-MAPPINGS, MAPPING-CONSTRAINTS, RESOURCE-ESTIMATIONS and SIGNAL-PATH-CONSTRAINTS are not upgraded to their target model counterparts.

8.1.1.8.87. SWC-TO-ECU-MAPPING

Source model entity: SWC-TO-ECU-MAPPING

Target model entity: SWC-TO-ECU-MAPPING

Target model property	Source model property
COMPONENT-IREFS/COMPONENT-IREF/CONTEXT-COMPOSITION-REF	COMPONENT-IREFS/COMPONENT-IREF/SOFT-WARE-COMPOSITION-REF
COMPONENT-IREFS/COMPONENT-IREF/CONTEXT-COMPONENT-REF	COMPONENT-IREFS/COMPONENT-IREF/COMPONENT-PROTOTYPE-REF
COMPONENT-IREFS/COMPONENT-IREF/TARGET-COMPONENT-REF	COMPONENT-IREFS/COMPONENT-IREF/TARGET-COMPONENT-PROTOTYPE-REF
ECU-INSTANCE-REF	ECU-INSTANCE-REF

No configuration of target model parameters PARTITION-REF and PROCESSING-UNIT-REF

The target model parameters PARTITION-REF and PROCESSING-UNIT-REF are not configured because there are no source model counterparts.

8.1.1.8.88. SWC-TO-IMPL-MAPPING

Source model entity: SWC-TO-IMPL-MAPPING

Target model entity: SWC-TO-IMPL-MAPPING

Target model property	Source model property
COMPONENT-IREFS/COMPONENT-IREF/CONTEXT-COMPOSITION-REF	COMPONENT-IREFS/COMPONENT-IREF/SOFT-WARE-COMPOSITION-REF
COMPONENT-IREFS/COMPONENT-IREF/CONTEXT-COMPONENT-REF	COMPONENT-IREFS/COMPONENT-IREF/COMPONENT-PROTOTYPE-REF
COMPONENT-IREFS/COMPONENT-IREF/TARGET-COMPONENT-REF	COMPONENT-IREFS/COMPONENT-IREF/TARGET-COMPONENT-PROTOTYPE-REF
COMPONENT-IMPLEMENTATION-REF	COMPONENT-IMPLEMENTATION-REF

8.1.1.8.89. SENDER-RECEIVER-TO-SIGNAL-GROUP-MAPPING

Source model entity: SENDER-RECEIVER-TO-SIGNAL-GROUP-MAPPING

Target model entity: SENDER-RECEIVER-TO-SIGNAL-GROUP-MAPPING

Target model property	Source model property
DATA-ELEMENT-IREF/CONTEXT-POSITION-REF	DATA-ELEMENT-IREF/SOFTWARE-POSITION-REF
DATA-ELEMENT-IREF/CONTEXT-COMPONENT-REF	DATA-ELEMENT-IREF/COMPONENT-PROTOTYPE-REF
DATA-ELEMENT-IREF/CONTEXT-PORT-REF	DATA-ELEMENT-IREF/PORT-PROTOTYPE-REF
DATA-ELEMENT-IREF/TARGET-DATA-PROTOTYPE-REF	DATA-ELEMENT-IREF/DATA-ELEMENT-REF
SIGNAL-GROUP-REF	SIGNAL-GROUP-REF
TYPE-MAPPING/SENDER-REC-RECORD-TYPE-MAPPING	TYPE-MAPPING/SENDER-REC-RECORD-TYPE-MAPPING
TYPE-MAPPING/SENDER-REC-ARRAY-TYPE-MAPPING	TYPE-MAPPING/SENDER-REC-ARRAY-TYPE-MAPPING

8.1.1.8.90. SENDER-RECEIVER-TO-SIGNAL-MAPPING

Source model entity: SENDER-RECEIVER-TO-SIGNAL-MAPPING

Target model entity: SENDER-RECEIVER-TO-SIGNAL-MAPPING

Target model property	Source model property
DATA-ELEMENT-IREF/CONTEXT-COM-POSITION-REF	DATA-ELEMENT-IREF/SOFTWARE-COM-POSITION-REF
DATA-ELEMENT-IREF/CONTEXT-COMPONENT-REF	DATA-ELEMENT-IREF/COMPONENT-PROTO-TYPE-REF
DATA-ELEMENT-IREF/CONTEXT-PORT-REF	DATA-ELEMENT-IREF/PORT-PROTOTYPE-REF
DATA-ELEMENT-IREF/TARGET-DATA-PROTO-TYPE-REF	DATA-ELEMENT-IREF/DATA-ELEMENT-REF
SYSTEM-SIGNAL-REF	SIGNAL-REF

8.1.1.8.91. CLIENT-SERVER-TO-SIGNAL-GROUP-MAPPING

Source model entity: CLIENT-SERVER-TO-SIGNAL-GROUP-MAPPING

Target model entity: CLIENT-SERVER-TO-SIGNAL-GROUP-MAPPING

Target model property	Source model property
APPLICATION-ERROR/SYSTEM-SIGNAL-REF	APPLICATION-ERRORS/APPLICATION-ER-ROR-MAPPING/SYSTEM-SIGNAL-REF
CLIENT-ID	CLIENT-ID
COMPOSITE-TYPE-MAPPINGS/CLIENT-SERVER-ARRAY-TYPE-MAPPING	COMPOSITE-TYPE-MAPPINGS/CLIENT-SERVER-ARRAY-TYPE-MAPPING
COMPOSITE-TYPE-MAPPINGS/CLIENT-SERVER-RECORD-TYPE-MAPPING	COMPOSITE-TYPE-MAPPINGS/CLIENT-SERVER-RECORD-TYPE-MAPPING
EMPTY-SIGNAL	EMPTY-SIGNAL
MAPPED-OPERATION-IREF/CONTEXT-COMPONENT-REF	MAPPED-OPERATION-IREF/COMPONENT-PROTO-TYPE-REF
MAPPED-OPERATION-IREF/CONTEXT-COM-POSITION-REF	MAPPED-OPERATION-IREF/COMPOSITION-PROTOTYPE-REF
MAPPED-OPERATION-IREF/TARGET-OPERATION-REF	MAPPED-OPERATION-IREF/OPERATION-REF
MAPPED-OPERATION-IREF/CONTEXT-PORT-REF	MAPPED-OPERATION-IREF/PORT-PROTOTYPE-REF
PRIMITIVE-TYPE-MAPPINGS/CLIENT-SERVER-PRIMITIVE-TYPE-MAPPING/ARGUMENT-REF	PRIMITIVE-TYPE-MAPPINGS/CLIENT-SERVER-PRIMITIVE-TYPE-MAPPING/ARGUMENT-REF

Target model property	Source model property
PRIMITIVE-TYPE-MAPPINGS/CLIENT-SERVER-PRIMITIVE-TYPE-MAPPING/SYSTEM-SIGNAL-REF	PRIMITIVE-TYPE-MAPPINGS/CLIENT-SERVER-PRIMITIVE-TYPE-MAPPING/SYSTEM-SIGNAL-REF
REQUEST-GROUP-REF	REQUEST-GROUP-REF
RESPONSE-GROUP-REF	RESPONSE-GROUP-REF
SEQUENCE-COUNTER/SYSTEM-SIGNAL-REF	SEQUENCE-COUNTER/SYSTEM-SIGNAL-REF

8.1.1.8.92. SENDER-REC-RECORD-TYPE-MAPPING

Source model entity: SENDER-REC-RECORD-TYPE-MAPPING

Target model entity: SENDER-REC-RECORD-TYPE-MAPPING

Target model property	Source model property
SENDER-REC-RECORD-TYPE-MAPPING/RECORD-ELEMENT-MAPPINGS/SENDER-REC-RECORD-ELEMENT-MAPPING	SENDER-REC-RECORD-TYPE-MAPPING/RECORD-ELEMENT-MAPPINGS/SENDER-REC-RECORD-ELEMENT-MAPPING

8.1.1.8.93. SENDER-REC-RECORD-ELEMENT-MAPPING

Source model entity: SENDER-REC-RECORD-ELEMENT-MAPPING

Target model entity: SENDER-REC-RECORD-ELEMENT-MAPPING

Target model property	Source model property
COMPLEX-TYPE-MAPPING/SENDER-REC-ARRAY-TYPE-MAPPING	COMPLEX-TYPE-MAPPING/SENDER-REC-ARRAY-TYPE-MAPPING
COMPLEX-TYPE-MAPPING/SENDER-REC-RECORD-TYPE-MAPPING	COMPLEX-TYPE-MAPPING/SENDER-REC-RECORD-TYPE-MAPPING
IMPLEMENTATION-RECORD-ELEMENT-REF	RECORD-ELEMENT-REF
SYSTEM-SIGNAL-REF	SIGNAL-REF

8.1.1.8.94. SENDER-REC-ARRAY-TYPE-MAPPING

Source model entity: SENDER-REC-ARRAY-TYPE-MAPPING

Target model entity: SENDER-REC-ARRAY-TYPE-MAPPING

Target model property	Source model property
ARRAY-ELEMENT-MAPPINGS / SENDER-REC-ARRAY-ELEMENT-MAPPING	ARRAY-ELEMENT-MAPPINGS / SENDER-REC-ARRAY-ELEMENT-MAPPING

8.1.1.8.95. SENDER-REC-ARRAY-ELEMENT-MAPPING

Source model entity: SENDER-REC-ARRAY-ELEMENT-MAPPING

Target model entity: SENDER-REC-ARRAY-ELEMENT-MAPPING

Target model property	Source model property
COMPLEX-TYPE-MAPPING / SENDER-REC-ARRAY-TYPE-MAPPING	COMPLEX-TYPE-MAPPING / SENDER-REC-ARRAY-TYPE-MAPPING
COMPLEX-TYPE-MAPPING / SENDER-REC-RECORD-TYPE-MAPPING	COMPLEX-TYPE-MAPPING / SENDER-REC-RECORD-TYPE-MAPPING
INDEXED-ARRAY-ELEMENT	INDEXED-ARRAY-ELEMENT
SYSTEM-SIGNAL-REF	SIGNAL-REF

8.1.1.8.96. END-TO-END-PROTECTION-SET

Source model entity: END-TO-END-PROTECTION-SET

Target model entity: END-TO-END-PROTECTION-SET

Target model property	Source model property
END-TO-END-PROTECTIONS / END-TO-END-PROTECTION	END-TO-END-PROTECTIONS / END-TO-END-PROTECTION

NOTE

End-to-End Protection is not part of the AUTOSAR 3.1.x standard



Since End-to-End Protection is not part of the AUTOSAR 3.1.x standard, the END-TO-END-PROTECTION-SET and its sub elements are only migrated in customer-specific models which extend AUTOSAR 3.1.x in such a way that End-to-End Protection is available.

8.1.1.8.97. END-TO-END-PROTECTION

Source model entity: END-TO-END-PROTECTION

Target model entity: END-TO-END-PROTECTION

Target model property	Source model property
END-TO-END-PROFILE/COUNTER-OFFSET	END-TO-END-PROFILE/COUNTER-OFFSET
END-TO-END-PROFILE/CRC-OFFSET	END-TO-END-PROFILE/CRC-OFFSET
END-TO-END-PROFILE/DATA-ID-MODE	END-TO-END-PROFILE/DATA-ID-MODE
END-TO-END-PROFILE/DATA-IDS/DATA-ID	END-TO-END-PROFILE/DATA-IDS/DATA-ID
END-TO-END-PROFILE/DATA-LENGTH	END-TO-END-PROFILE/DATA-LENGTH
END-TO-END-PROFILE/MAX-DELTA-COUNTER-INIT	END-TO-END-PROFILE/MAX-DELTA-COUNTER-INIT
END-TO-END-PROTECTION-I-SIGNAL-I-PDUS/ END-TO-END-PROTECTION-I-SIGNAL-I-PDU	END-TO-END-PROTECTION-I-SIGNAL-I-PDUS/ END-TO-END-PROTECTION-I-SIGNAL-I-PDU
END-TO-END-PROTECTION-VARIABLE-PROTOTYPES/END-TO-END-PROTECTION-VARIABLE-PROTOTYPE	END-TO-END-PROTECTION-DATA-ELEMENT-PROTOTYPES/END-TO-END-PROTECTION-DATA-ELEMENT-PROTOTYPE

8.1.1.8.98. END-TO-END-PROTECTION-I-SIGNAL-I-PDU

Source model entity: END-TO-END-PROTECTION-I-SIGNAL-I-PDU

Target model entity: END-TO-END-PROTECTION-I-SIGNAL-I-PDU

Target model property	Source model property
DATA-OFFSET	DATA-OFFSET
I-SIGNAL-GROUP-REF	I-SIGNAL-GROUP-REF
I-SIGNAL-I-PDU-REF	I-SIGNAL-I-PDU-REF

8.1.1.8.99. END-TO-END-PROTECTION-VARIABLE-PROTOTYPE

Source model entity: END-TO-END-PROTECTION-DATA-ELEMENT-PROTOTYPE

Target model entity: END-TO-END-PROTECTION-VARIABLE-PROTOTYPE

Target model property	Source model property
RECEIVER-IREFS/RECEIVER-IREF/CONTEXT-COMPONENT-REF	RECEIVER-IREFS/RECEIVER-IREF/COMPONENT-PROTOTYPE-REF

Target model property	Source model property
RECEIVER-IREFS/RECEIVER-IREF/CONTEXT-PORT-REF	RECEIVER-IREFS/RECEIVER-IREF/PORT-PROTOTYPE-REF
RECEIVER-IREFS/RECEIVER-IREF/TARGET-DATA-PROTOTYPE-REF	RECEIVER-IREFS/RECEIVER-IREF/TARGET-DATA-PROTOTYPE-REF
SENDER-IREF/CONTEXT-COMPONENT-REF	SENDER-IREF/COMPONENT-PROTOTYPE-REF
SENDER-IREF/CONTEXT-PORT-REF	SENDER-IREF/PORT-PROTOTYPE-REF
SENDER-IREF/TARGET-DATA-PROTOTYPE-REF	SENDER-IREF/TARGET-DATA-PROTOTYPE-REF

8.1.2. Upgrading a system model from AUTOSAR 3.2.x to AUTOSAR 4.0.3

This section describes how an AUTOSAR 3.2.x system model is upgraded to AUTOSAR 4.0.3.

Since the AUTOSAR 3.1.x and 3.2.x source models and the destination system models 4.0.2 and 4.0.3 are similar, this chapter only describes those parts of the upgrade that differ from the 3.1.x to 4.0.2 upgrade. For all other parts, please refer to [Section 8.1.1, “Upgrading a system model from AUTOSAR 3.1.x to AUTOSAR 4.0.2”](#).

If not stated explicitly, all entities in the 4.0.x target model are located in the same package as they were in the 3.1.x source model.

If not specified explicitly otherwise, the parameters `SHORT-NAME`, `CATEGORY`, and `UUID` of `Identifiable` entities are upgraded directly into their counterparts in the target model.

8.1.2.1. Topology

8.1.2.1.1. SYSTEM

Source model entity: `SYSTEM`

Target model entity: `SYSTEM`

Target model property	Source model property
<code>SYSTEM-VERSION</code>	<code>SYSTEM-VERSION</code>
<code>ECU-EXTRACT-VERSION</code>	<code>ECU-EXTRACT-VERSION</code>

Target model property	Source model property
FIBEX-ELEMENT-REFS	FIBEX-ELEMENT-REFS
PNC-VECTOR-LENGTH	PNC-VECTOR-LENGTH
PNC-VECTOR-OFFSET	PNC-VECTOR-OFFSET
MAPPINGS/SYSTEM-MAPPING	MAPPING
SYSTEM/ROOT-SOFTWARE-COMPOSITIONS/ROOT-SW-COMPOSITION-PROTOTYPE	SYSTEM/SOFTWARE-COMPOSITION

Changed or moved parameters

FIBEX-ELEMENT-REFS

In FIBEX-ELEMENT-REFS, references to the following types of source model entities are upgraded to their target model counterparts: CAN-CLUSTER, COMMUNICATION-CLUSTER, DCM-I-PDU, ECU-INSTANCE, FLEXRAY-CLUSTER, FRAME, GATEWAY, I-PDU-GROUP, I-SIGNAL, LIN-CLUSTER, MULTIPLEXED-I-PDU, N-PDU, NM-PDU, PDUR-I-PDU-GROUP, SIGNAL-I-PDU, SUBSTITUTION-FRAME, USER-DEFINED-I-PDU, USER-DEFINED-PDU.

8.1.2.1.2. ECU-INSTANCE

Source model entity: ECU-INSTANCE

Target model entity: ECU-INSTANCE

Target model property	Source model property
ASSOCIATED-COM-I-PDU-GROUP-REFS/ASSOCIATED-COM-I-PDU-GROUP-REF	ASSOCIATED-I-PDU-GROUP-REFS/ASSOCIATED-I-PDU-GROUP-REF
ASSOCIATED-PDUR-I-PDU-GROUP-REFS/ASSOCIATED-PDUR-I-PDU-GROUP-REF	ASSOCIATED-PDUR-I-PDU-GROUP-REFS/ASSOCIATED-PDUR-I-PDU-GROUP-REF
COM-CONFIGURATION-GW-TIME-BASE	COM-PROCESSING-PERIOD-GW
COM-CONFIGURATION-RX-TIME-BASE	COM-PROCESSING-PERIOD-RX
COM-CONFIGURATION-TX-TIME-BASE	COM-PROCESSING-PERIOD
COMM-CONTROLLERS	COMM-CONTROLLERS
CONNECTORS	CONNECTORS
DIAGNOSTIC-ADDRESS	DIAGNOSTIC-ADDRESS
SLEEP-MODE-SUPPORTED	SLEEP-MODE-SUPPORTED
WAKE-UP-OVER-BUS-SUPPORTED	WAKE-UP-OVER-BUS-SUPPORTED

Target model property	Source model property
No target model parameter is available.	COM-CONFIGURATION-ID
No target model parameter is available.	PDU-R-CONFIGURATION-ID
Not upgraded.	RESPONSE-ADDRESSS/RESPONSE-ADDRESS

Reason for not upgraded source model parameter

RESPONSE-ADDRESSS/RESPONSE-ADDRESS

The parameter RESPONSE-ADDRESSS/RESPONSE-ADDRESS is not upgraded to the target model since the contained diagnostic addresses cannot be uniquely assigned to concrete interfaces.

8.1.2.1.3. CAN-CLUSTER

Source model entity: CAN-CLUSTER

Target model entity: CAN-CLUSTER

Target model property	Source model property
CAN-CLUSTER-CONDITIONAL/PHYSICAL-CHANNELS	PHYSICAL-CHANNELS
CAN-CLUSTER-CONDITIONAL/PROTOCOL-NAME	PROTOCOL-NAME
CAN-CLUSTER-CONDITIONAL/PROTOCOL-VERSION	PROTOCOL-VERSION
CAN-CLUSTER-CONDITIONAL/SPEED	SPEED/1000
No target model parameter is available.	MAX-FRAME-LENGTH
No target model parameter is available.	NM-REPEAT-MESSAGE-SUPPORT

Changed or moved parameters

NM-NODE-DETECTION-ENABLED and NM-NODE-ID-ENABLED

The source model parameters NM-NODE-DETECTION-ENABLED and NM-NODE-ID-ENABLED are configured in the target model entity NM-ECU, see [Section 8.1.2.7.2, "NM-ECU"](#).

SPEED

In the target model, the parameter SPEED is given in kbit/s, whereas the source model provides it in bit/s. If the source model parameter is not divisible by 1000 without rest, it cannot be upgraded at all.

NM-BUS-LOAD-REDUCTION-ACTIVE, NM-WAIT-BUS-SLEEP-TIME, NM-IMMEDIATE-NM-CYCLE-TIME, NM-IMMEDIATE-NM-TRANSMISSIONS, NM-MSG-CYCLE-TIME, and NM-REMOTE-SLEEP-INDICATION-TIME

The parameters NM-BUS-LOAD-REDUCTION-ACTIVE, NM-WAIT-BUS-SLEEP-TIME, NM-IMMEDIATE-NM-CYCLE-TIME, NM-IMMEDIATE-NM-TRANSMISSIONS, NM-MSG-CYCLE-TIME, and NM-REMOTE-SLEEP-INDICATION-TIME are configured as NM-BUSLOAD-REDUCTION-ACTIVE, NM-WAIT-BUS-SLEEP-TIME, NM-IMMEDIATE-NM-CYCLE-TIME, NM-IMMEDIATE-NM-TRANSMISSIONS, NM-MSG-CYCLE-TIME, and NM-REMOTE-SLEEP-INDICATION-TIME in the target model entity CAN-NM-CLUSTER, see [Section 8.1.2.7.3, "CAN-NM-CLUSTER"](#).

NM-BUS-LOAD-REDUCTION-ENABLED

The parameter NM-BUS-LOAD-REDUCTION-ENABLED is configured as NM-BUSLOAD-REDUCTION-ENABLED in the target model entity CAN-NM-CLUSTER-COUPLING, see [Section 8.1.1.7.8, "FLEXRAY-NM-CLUSTER-COUPLING"](#).

NM-LOWER-CAN-ID and NM-UPPER-CAN-ID

The parameters NM-LOWER-CAN-ID and NM-UPPER-CAN-ID are configured in the target model entity CAN-NM-RANGE-CONFIG parameters LOWER-CAN-ID and UPPER-CAN-ID, which is aggregated in CAN-NM-NODE, see [Section 8.1.1.7.4, "CAN-NM-NODE"](#).

8.1.2.1.4. CAN-COMMUNICATION-CONTROLLER

Source model entity: CAN-COMMUNICATION-CONTROLLER

Target model entity: CAN-COMMUNICATION-CONTROLLER

Target model property	Source model property
CATEGORY	CATEGORY
CAN-COMMUNICATION-CONTROLLER-CONDITION-AL/CAN-CONTROLLER-ATTRIBUTES/CAN-CONTROLLER-CONFIGURATION/SYNC-JUMP-WIDTH	CAN-CONTROLLER-ATTRIBUTES/CAN-CONTROLLER-CONFIGURATION/SYNC-JUMP-WIDTH
CAN-COMMUNICATION-CONTROLLER-CONDITION-AL/CAN-CONTROLLER-ATTRIBUTES/CAN-CONTROLLER-CONFIGURATION/TIME-SEG-1	CAN-CONTROLLER-ATTRIBUTES/CAN-CONTROLLER-CONFIGURATION/TIME-SEG-1
CAN-COMMUNICATION-CONTROLLER-CONDITION-AL/CAN-CONTROLLER-ATTRIBUTES/CAN-CONTROLLER-CONFIGURATION/TIME-SEG-2	CAN-CONTROLLER-ATTRIBUTES/CAN-CONTROLLER-CONFIGURATION/TIME-SEG-2
No target model parameter is available.	CAN-CONTROLLER-ATTRIBUTES/CAN-CONTROLLER-CONFIGURATION/NUMBER-OF-SAMPLES
No target model parameter is available.	CAN-CONTROLLER-ATTRIBUTES/CAN-CONTROLLER-CONFIGURATION/PROPAGATION-DELAY

Target model property	Source model property
CAN-COMMUNICATION-CONTROLLER/CAN-CONTROLLER-CONFIGURATION-REQUIREMENTS	No source model parameter is available.

Changed or moved parameters

NM-MSG-CYCLE-OFFSET and NM-MSG-REDUCED-TIME

The source model parameters NM-MSG-CYCLE-OFFSET and NM-MSG-REDUCED-TIME are configured in the target model entity NM-NODE, see [Section 8.1.2.7.2, "NM-ECU"](#).

NM-USER-DATA-ENABLED

The source model parameter NM-USER-DATA-ENABLED is configured in the target model entity NM-ECU, see [Section 8.1.2.7.2, "NM-ECU"](#).

8.1.2.1.5. CAN-COMMUNICATION-CONNECTOR

Source model entity: CAN-COMMUNICATION-CONNECTOR

Target model entity: CAN-COMMUNICATION-CONNECTOR

Target model property	Source model property
CATEGORY	CATEGORY
COMM-CONTROLLER-REF	COMM-CONTROLLER-REF
ECU-COMM-PORT-INSTANCES	ECU-COMM-PORT-INSTANCES
PNC-WAKEUP-CAN-ID	PNC-WAKEUP-CAN-ID
PNC-WAKEUP-CAN-ID-EXTENDED	PNC-WAKEUP-CAN-ID-EXTENDED
PNC-WAKEUP-CAN-ID-MASK	PNC-WAKEUP-CAN-ID-MASK
PNC-WAKEUP-DATA-MASK	PNC-WAKEUP-DATA-MASK
PNC-WAKEUP-DLC	PNC-WAKEUP-DLC
PNC-GATEWAY-TYPE	PNC-GATEWAY-TYPE

Changed or moved parameters

CHANNEL-REF

The source model parameter CHANNEL-REF has been replaced by the target model parameter COMM-CONNECTORS/COMMUNICATION-CONNECTOR-REF-CONDITIONAL of the target model entity CAN-PHYSICAL-CHANNEL.

NM-ADDRESS

The source model parameter NM-ADDRESS is configured in the target model entity CAN-NM-NODE, see [Section 8.1.2.7.4, “CAN-NM-NODE”](#).

8.1.2.1.6. CAN-PHYSICAL-CHANNEL

Source model entities: PHYSICAL-CHANNEL, COMMUNICATION-CONNECTOR

Target model entity: CAN-PHYSICAL-CHANNEL

Target model property	Source model property
CATEGORY	CATEGORY
FRAME-TRIGGERINGS	FRAME-TRIGGERINGSS
PDU-TRIGGERINGS	I-PDU-TRIGGERINGS
I-SIGNAL-TRIGGERINGS	I-SIGNAL-TRIGGERINGS

Changed or moved parameters

COMM-CONNECTORS

The target model parameter COMM-CONNECTORS is configured by retrieving all target model COMMUNICATION-CONNECTOR entities that meet the following condition: The source model counterpart of the COMMUNICATION-CONNECTOR references the source model counterpart of the upgraded CAN-PHYSICAL-CHANNEL via CHANNEL-REF.

CAN-TP-CONNECTION-CHANNEL

The CAN-TP-CONNECTION-CHANNEL entities aggregated in TP-CHANNELS are upgraded to CAN-TP-CONNECTION and CAN-TP-CHANNEL entities in the target model, aggregated in CAN-TP-CONFIG, see [Section 8.1.2.4.1, “CAN-TP-CONFIG”](#).

TP-ADDRESS

The TP-ADDRESS entities aggregated in TP-ADDRESSSS are upgraded to CAN-TP-ADDRESS entities in the target model, aggregated in CAN-TP-CONFIG, see [Section 8.1.2.4.1, “CAN-TP-CONFIG”](#).

CAN-TP-NODE

The CAN-TP-NODE entities aggregated in TP-NODES are upgraded to CAN-TP-NODE entities in the target model, which are aggregated in CAN-TP-CONFIG, see [Section 8.1.2.4.1, “CAN-TP-CONFIG”](#).

8.1.2.1.7. FLEXRAY-CLUSTER

Source model entity: FLEXRAY-CLUSTER

Target model entity: FLEXRAY-CLUSTER/FLEXRAY-CLUSTER-CONDITIONAL

Target model property	Source model property
PHYSICAL-CHANNELS	PHYSICAL-CHANNELS
PROTOCOL-NAME	PROTOCOL-NAME
PROTOCOL-VERSION	PROTOCOL-VERSION
SPEED	SPEED/1000
ACTION-POINT-OFFSET	ACTION-POINT-OFFSET
BIT	BIT
CAS-RX-LOW-MAX	CAS-RX-LOW-MAX
COLD-START-ATTEMPTS	COLD-START-ATTEMPTS
CYCLE	CYCLE
CYCLE-COUNT-MAX	CYCLE-COUNT-MAX
DYNAMIC-SLOT-IDLE-PHASE	DYNAMIC-SLOT-IDLE-PHASE
LISTEN-NOISE	LISTEN-NOISE
MACRO-PER-CYCLE	MACRO-PER-CYCLE
MACROTICK-DURATION	MACROTICK-DURATION
MAX-WITHOUT-CLOCK-CORRECTION-FATAL	MAX-WITHOUT-CLOCK-CORRECTION-FATAL
MAX-WITHOUT-CLOCK-CORRECTION-PASSIVE	MAX-WITHOUT-CLOCK-CORRECTION-PASSIVE
MINISLOT-ACTION-POINT-OFFSET	MINISLOT-ACTION-POINT-OFFSET
MINISLOT-DURATION	MINISLOT-DURATION
NETWORK-IDLE-TIME	NETWORK-IDLE-TIME
NETWORK-MANAGEMENT-VECTOR-LENGTH	NETWORK-MANAGEMENT-VECTOR-LENGTH
NUMBER-OF-MINISLOTS	NUMBER-OF-MINISLOTS
NUMBER-OF-STATIC-SLOTS	NUMBER-OF-STATIC-SLOTS
OFFSET-CORRECTION-START	OFFSET-CORRECTION-START
PAYLOAD-LENGTH-STATIC	PAYLOAD-LENGTH-STATIC
SAMPLE-CLOCK-PERIOD	SAMPLE-CLOCK-PERIOD
STATIC-SLOT-DURATION	STATIC-SLOT-DURATION
SYMBOL-WINDOW	SYMBOL-WINDOW
SYMBOL-WINDOW-ACTION-POINT-OFFSET	SYMBOL-WINDOW-ACTION-POINT-OFFSET
SYNC-FRAME-ID-COUNT-MAX	SYNC-FRAME-ID-COUNT-MAX
TRANSMISSION-START-SEQUENCE-DURATION	TRANSMISSION-START-SEQUENCE-DURATION

Target model property	Source model property
WAKEUP-RX-IDLE	WAKEUP-RX-IDLE
WAKEUP-RX-LOW	WAKEUP-RX-LOW
WAKEUP-RX-WINDOW	WAKEUP-RX-WINDOW
WAKEUP-TX-ACTIVE	WAKEUP-TX-ACTIVE
WAKEUP-TX-IDLE	WAKEUP-TX-IDLE
No target model parameter is available.	MAX-FRAME-LENGTH
No target model parameter is available.	NM-REPEAT-MESSAGE-SUPPORT
No target model parameter is available.	BUS-GUARDIAN-ENABLE-PART
No target model parameter is available.	CAS-RX-LOW-MIN
No target model parameter is available.	MACRO-INITIAL-OFFSET
No target model parameter is available.	MAX-INITIALISATION-ERROR
No target model parameter is available.	MAX-PROPAGATION-DELAY
No target model parameter is available.	MIN-PROPAGATION-DELAY
No target model parameter is available.	OFFSET-CORRECTION-MAX
DETECT-NIT-ERROR	No source model parameter is available.
IGNORE-AFTER-TX	IGNORE-AFTER-TX
SAFETY-MARGIN	No source model parameter is available.

Changed or moved parameters

NM-NODE-DETECTION-ENABLED and NM-NODE-ID-ENABLED

The source model parameters NM-NODE-DETECTION-ENABLED and NM-NODE-ID-ENABLED are configured in the target model entity NM-ECU, see [Section 8.1.2.7.2, “NM-ECU”](#).

SPEED

In the target model, the parameter SPEED is given in kBit/s, whereas the source model provides it in Bit/s. If the source model parameter is not divisible by 1000 without rest, it cannot be upgraded at all.

NM-DATA-CYCLE, NM-READY-SLEEP-COUNT, NM-REMOTE-SLEEP-INDICATION-TIME, NM-REPETAT-MESSAGE-TIME, NM-REPETITION-CYCLE, NM-VOTING-CYCLE

The source model parameters NM-DATA-CYCLE, NM-READY-SLEEP-COUNT, NM-REMOTE-SLEEP-INDICATION-TIME, NM-REPETAT-MESSAGE-TIME, NM-REPETITION-CYCLE, and NM-VOTING-CYCLE are configured in the target model entity FLEXRAY-NM-CLUSTER, see [Section 8.1.2.7.6, “FLEXRAY-NM-CLUSTER”](#).

Reason for not upgraded source model parameter

MAX-FRAME-LENGTH, BUS-GUARDIAN-ENABLE-PART, CAS-RX-LOW-MIN, CAS-RX-LOW-MIN, MACRO-INITIAL-OFFSET, MAX-INITIALISATION-ERROR, MAX-PROPAGATION-DELAY, MIN-PROPAGATION-DELAY, OFFSET-CORRECTION-MAX, NM-REPEAT-MESSAGE-SUPPORT

The parameters MAX-FRAME-LENGTH, BUS-GUARDIAN-ENABLE-PART, CAS-RX-LOW-MIN, CAS-RX-LOW-MIN, MACRO-INITIAL-OFFSET, MAX-INITIALISATION-ERROR, MAX-PROPAGATION-DELAY, MIN-PROPAGATION-DELAY, OFFSET-CORRECTION-MAX, NM-REPEAT-MESSAGE-SUPPORT are not upgraded to the target model, because they do not contribute any data required for configuring an ECU. Therefore, they have been removed from the target model.

8.1.2.1.8. FLEXRAY-COMMUNICATION-CONTROLLER

Source model entity: FLEXRAY-COMMUNICATION-CONTROLLER

Target model entity: FLEXRAY-COMMUNICATION-CONTROLLER/FLEXRAY-COMMUNICATION-CONTROLLER-CONDITIONAL

Target model property	Source model property
CATEGORY	CATEGORY
ACCEPTED-STARTUP-RANGE	ACCEPTED-STARTUP-RANGE
ALLOW-HALT-DUE-TO-CLOCK	ALLOW-HALT-DUE-TO-CLOCK
ALLOW-PASSIVE-TO-ACTIVE	ALLOW-PASSIVE-TO-ACTIVE
CLUSTER-DRIFT-DAMPING	CLUSTER-DRIFT-DAMPING
DECODING-CORRECTION	DECODING-CORRECTION
DELAY-COMPENSATION-A	DELAY-COMPENSATION-A
DELAY-COMPENSATION-B	DELAY-COMPENSATION-B
EXTERN-OFFSET-CORRECTION	EXTERN-OFFSET-CORRECTION
EXTERN-RATE-CORRECTION	EXTERN-RATE-CORRECTION
KEY-SLOT-ID	KEY-SLOT-ID
KEY-SLOT-ONLY-ENABLED	KEY-SLOT-ONLY-ENABLED
KEY-SLOT-USED-FOR-START-UP	KEY-SLOT-USED-FOR-START-UP
KEY-SLOT-USED-FOR-SYNC	KEY-SLOT-USED-FOR-SYNC
LATEST-TX	LATEST-TX
LISTEN-TIMEOUT	LISTEN-TIMEOUT
MACRO-INITIAL-OFFSET-A	MACRO-INITIAL-OFFSET-A
MACRO-INITIAL-OFFSET-B	MACRO-INITIAL-OFFSET-B

Target model property	Source model property
MAXIMUM-DYNAMIC-PAYLOAD-LENGTH	MAXIMUM-DYNAMIC-PAYLOAD-LENGTH
MICRO-INITIAL-OFFSET-A	MICRO-INITIAL-OFFSET-A
MICRO-INITIAL-OFFSET-B	MICRO-INITIAL-OFFSET-B
MICRO-PER-CYCLE	MICRO-PER-CYCLE
MICROTICK-DURATION	MICROTICK-DURATION
OFFSET-CORRECTION-OUT	OFFSET-CORRECTION-OUT
RATE-CORRECTION-OUT	RATE-CORRECTION-OUT
SAMPLES-PER-MICROTICK	SAMPLES-PER-MICROTICK
WAKE-UP-PATTERN	WAKE-UP-PATTERN
No target model parameter is available.	DYNAMIC-SEGMENT-ENABLE
No target model parameter is available.	MICRO-PER-MACRO-NOM
No target model parameter is available.	START-UP-NODE
No target model parameter is available.	SYNC-SLOT
EXTERNAL-SYNC	EXTERNAL-SYNC
FALL-BACK-INTERNAL	FALL-BACK-INTERNAL
FLEXRAY-FIFOS/FLEXRAY-FIFO-CONFIGURATION	FLEXRAY-FIFOS/FLEXRAY-FIFO-CONFIGURATION
NM-VECTOR-EARLY-UPDATE	NM-VECTOR-EARLY-UPDATE
SECOND-KEY-SLOT-ID	SECOND-KEY-SLOT-ID
TWO-KEY-SLOT-MODE	TWO-KEY-SLOT-MODE
No target model parameter is available.	NM-SOURCE-NODE-IDENTIFIER-ENABLED

Changed or moved parameters

NM-CONTROL-BIT-VECTOR-ENABLED

The source model parameter NM-CONTROL-BIT-VECTOR-ENABLED is configured in the target model entity FLEXRAY-NM-CLUSTER, see [Section 8.1.2.7.6, “FLEXRAY-NM-CLUSTER”](#).

NM-NODE-DETECTION-ENABLED, NM-USER-DATA-ENABLED

The source model parameters NM-NODE-DETECTION-ENABLED and NM-USER-DATA-ENABLED are configured in the target model entity NM-ECU, see [Section 8.1.2.7.2, “NM-ECU”](#).

Reason for not upgraded source model parameter

DYNAMIC SEGMENT-ENABLE, MICRO-PER-MACRO-NOM, START-UP-NODE, SYNC-SLOT

The parameters DYNAMIC SEGMENT-ENABLE, MICRO-PER-MACRO-NOM, START-UP-NODE, SYNC-SLOT are not upgraded to the target model, because they do not contribute any data required for configuring an ECU or are redundant with respect to other parameters. Therefore, they have been removed from the target model.

8.1.2.1.9. FLEXRAY-FIFO-CONFIGURATION

Source model entity: FLEXRAY-FIFO-CONFIGURATION

Target model entity: FLEXRAY-FIFO-CONFIGURATION

Target model property	Source model property
ADMIT-WITHOUT-MESSAGE-ID	ADMIT-WITHOUT-MESSAGE-ID
BASE-CYCLE	BASE-CYCLE
CHANNEL-REF	CHANNEL-REFS/CHANNEL-REF
CYCLE-REPETITION	CYCLE-REPETITION
FIFO-DEPTH	FIFO-DEPTH
FIFO-RANGES/FLEXRAY-FIFO-RANGE	FIFO-RANGES/FLEXRAY-FIFO-RANGE
MSG-ID-MASK	MSG-ID-MASK
MSG-ID-MATCH	MSG-ID-MATCH

Changed or moved parameters

CHANNEL-REFS/CHANNEL-REF

If multiple CHANNEL-REFS/CHANNEL-REF exist in the source model, a copy of FLEXRAY-FIFO-CONFIGURATION is created for each CHANNEL-REFS/CHANNEL-REF.

8.1.2.1.10. FLEXRAY-FIFO-RANGE

Source model entity: FLEXRAY-FIFO-RANGE

Target model entity: FLEXRAY-FIFO-RANGE

Target model property	Source model property
RANGE-MAX	RANGE-MAX
RANGE-MIN	RANGE-MIN

8.1.2.1.11. FLEXRAY-COMMUNICATION-CONNECTOR

Source model entity: FLEX-RAY-COMMUNICATION-CONNECTOR

Target model entity: FLEXRAY-COMMUNICATION-CONNECTOR

Target model property	Source model property
CATEGORY	CATEGORY
COMM-CONTROLLER-REF	COMM-CONTROLLER-REF
ECU-COMM-PORT-INSTANCES	ECU-COMM-PORT-INSTANCES
WAKE-UP-CHANNEL	WAKE-UP-CHANNEL
PNC-GATEWAY-TYPE	PNC-GATEWAY-TYPE

Changed or moved parameters

ECU-COMM-PORT-INSTANCES

In ECU-COMM-PORT-INSTANCES, the aggregated source model entities are upgraded to their target model counterparts: FRAME-PORT, I-PDU-PORT, SIGNAL-PORT.

CHANNEL-REF

The source model parameter CHANNEL-REF has been replaced by the target model parameter COMM-CONNECTORS/COMMUNICATION-CONNECTOR-REF-CONDITIONAL of the target model entity FLEXRAY-PHYSICAL-CHANNEL.

NM-ADDRESS

The source model parameter NM-ADDRESS is configured in the target model entity FLEXRAY-NM-NODE, see [Section 8.1.2.7.7, “FLEXRAY-NM-NODE”](#).

8.1.2.1.12. FLEXRAY-PHYSICAL-CHANNEL

Source model entities: PHYSICAL-CHANNEL, FLEX-RAY-COMMUNICATION-CONNECTOR

Target model entity: FLEXRAY-PHYSICAL-CHANNEL

Target model property	Source model property
CATEGORY	CATEGORY
FRAME-TRIGGERINGS	FRAME-TRIGGERINGSS
PDU-TRIGGERINGS	I-PDU-TRIGGERINGS
I-SIGNAL-TRIGGERINGS	I-SIGNAL-TRIGGERINGS
CHANNEL-NAME	CHANNEL-NAME

Changed or moved parameters

COMM-CONNECTORS

The target model parameter COMM-CONNECTORS is configured by retrieving all target model FLEXRAY-COMMUNICATION-CONNECTOR entities that meet the following condition: The source model counterpart of the FLEXRAY-COMMUNICATION-CONNECTOR references the source model counterpart of the upgraded FLEXRAY-PHYSICAL-CHANNEL via CHANNEL-REF.

TP-ADDRESS

The TP-ADDRESS entities aggregated in TP-ADDRESSSS are upgraded to TP-ADDRESS entities, aggregated in the target model parameter FLEXRAY-TP-CONFIG/TP-ADDRESSSS, see [Section 8.1.2.5.1, "FLEXRAY-TP-CONFIG"](#).

FLEXRAY-ISO-TP-CONNECTION

The FLEXRAY-ISO-TP-CONNECTION entities aggregated in TP-CHANNELS are upgraded to FLEXRAY-TP-NODE entities, which are aggregated in the target model parameter FLEXRAY-TP-CONFIG/TP-NODES, see [Section 8.1.2.5.1, "FLEXRAY-TP-CONFIG"](#).

FLEXRAY-ISO-TP-NODE

The FLEXRAY-ISO-TP-NODE entities aggregated in TP-NODES are upgraded to FLEXRAY-TP-CONNECTION entities, aggregated in the target model parameter FLEXRAY-TP-CONFIG, see [Section 8.1.2.5.1, "FLEXRAY-TP-CONFIG"](#).

FLEXRAY-ISO-TP-PDU-POOL

The FLEXRAY-ISO-TP-PDU-POOL entities aggregated in PDU-POOLS are upgraded to FLEXRAY-TP-PDU-POOL entities, aggregated in the target model parameter FLEXRAY-TP-CONFIG, see [Section 8.1.2.5.1, "FLEXRAY-TP-CONFIG"](#).

FLEXRAY-ISO-TP-CONNECTION-CONTROL

The FLEXRAY-ISO-TP-CONNECTION-CONTROL entities aggregated in TP-CONNECTION-CONTROLS are upgraded to FLEXRAY-TP-CONNECTION-CONTROL entities, aggregated in the target model parameter FLEXRAY-TP-CONFIG/TP-CONNECTION-CONTROLS, see [Section 8.1.2.5.1, "FLEXRAY-TP-CONFIG"](#).

FLEXRAY-ISO-TP-ECU

The FLEXRAY-ISO-TP-ECU entities aggregated in TP-ECUSS are upgraded to FLEXRAY-TP-ECU entities, aggregated in the target model parameter FLEXRAY-TP-CONFIG/TP-ECUS, see [Section 8.1.2.5.1, "FLEXRAY-TP-CONFIG"](#).

8.1.2.1.13. LIN-CLUSTER

The LIN-CLUSTER is upgraded as described in [Section 8.1.1.11, "LIN-CLUSTER"](#).

8.1.2.1.14. LIN-MASTER

The LIN-MASTER is upgraded as described in [Section 8.1.1.12, "LIN-MASTER"](#).

8.1.2.1.15. LIN-SLAVE

The LIN-SLAVE is upgraded as described in [Section 8.1.1.1.13, “LIN-SLAVE”](#).

8.1.2.1.16. LIN-ERROR-RESPONSE

The LIN-ERROR-RESPONSE is upgraded as described in [Section 8.1.1.1.14, “LIN-ERROR-RESPONSE”](#).

8.1.2.1.17. LIN-COMMUNICATION-CONNECTOR

Source model entity: COMMUNICATION-CONNECTOR

Target model entity: LIN-COMMUNICATION-CONNECTOR

Target model property	Source model property
CATEGORY	CATEGORY
COMM-CONTROLLER-REF	COMM-CONTROLLER-REF
ECU-COMM-PORT-INSTANCES	ECU-COMM-PORT-INSTANCES
LIN-CONFIGURABLE-FRAMES/LIN-CONFIGURABLE-FRAME	No source model parameter is available.
PNC-GATEWAY-TYPE	PNC-GATEWAY-TYPE

Changed or moved parameters

ECU-COMM-PORT-INSTANCES

In ECU-COMM-PORT-INSTANCES, the aggregated source model entities are upgraded into their target model counterparts: FRAME-PORT, I-PDU-PORT, SIGNAL-PORT.

CHANNEL-REF

The source model parameter CHANNEL-REF has been replaced by the target model parameter COMM-CONNECTORS/COMMUNICATION-CONNECTOR-REF-CONDITIONAL of the target model entity LIN-PHYSICAL-CHANNEL.

INITIAL-NAD

The target model parameter INITIAL-NAD is configured by taking the CONFIGURED-NAD parameter value of the LIN-SLAVE which is referenced via COMM-CONTROLLER-REF.

8.1.2.1.18. LIN-PHYSICAL-CHANNEL

Source model entities: PHYSICAL-CHANNEL, COMMUNICATION-CONNECTOR

Target model entity: LIN-PHYSICAL-CHANNEL

Target model property	Source model property
CATEGORY	CATEGORY
FRAME-TRIGGERINGS	FRAME-TRIGGERINGSS
PDU-TRIGGERINGS	I-PDU-TRIGGERINGS
I-SIGNAL-TRIGGERINGS	I-SIGNAL-TRIGGERINGS

Changed or moved parameters

COMM-CONNECTORS

The target model parameter COMM-CONNECTORS is configured by retrieving all target model LIN-COMMUNICATION-CONNECTOR entities that meet the following condition: The source model counterpart of the LIN-COMMUNICATION-CONNECTOR references the source model counterpart of the upgraded LIN-PHYSICAL-CHANNEL via CHANNEL-REF.

TP-ADDRESS

The TP-ADDRESS entities aggregated in TP-ADDRESSSS are upgraded to TP-ADDRESS entities, aggregated in the target model parameter LIN-TP-CONFIG, see [Section 8.1.2.6.1, “LIN-TP-CONFIG”](#).

LIN-TP-CHANNEL entities aggregated in TP-CHANNELS

The LIN-TP-CHANNEL entities aggregated in TP-CHANNELS are upgraded to LIN-TP-CONNECTION entities, which are aggregated in the target model parameter LIN-TP-CONFIG, see [Section 8.1.2.6.1, “LIN-TP-CONFIG”](#).

LIN-TP-NODE

The LIN-TP-NODE entities aggregated in TP-NODES are upgraded to LIN-TP-NODE entities, aggregated in the target model parameter LIN-TP-CONFIG, see [Section 8.1.2.6.1, “LIN-TP-CONFIG”](#).

LIN-PHYSICAL-CHANNEL

The target model entity LIN-PHYSICAL-CHANNEL aggregates in its parameter SCHEDULE-TABLES all LIN-SCHEDULE-TABLE entities the source model entity LIN-CLUSTER aggregates in its corresponding parameter.

8.1.2.2. Communication

8.1.2.2.1. CAN-FRAME-TRIGGERING

Source model entity: CAN-FRAME-TRIGGERING

Target model entity: CAN-FRAME-TRIGGERING

Target model property	Source model property
CAN-ADDRESSING-MODE	CAN-ADDRESSING-MODE
CATEGORY	CATEGORY
IDENTIFIER	IDENTIFIER
FRAME-PORT-REFS / FRAME-PORT-REF	FRAME-PORT-REFS / FRAME-PORT-REF
FRAME-REF	FRAME-REF
PDU-TRIGGERINGS / PDU-TRIGGERING-REF-CONDITIONAL	I-PDU-TRIGGERING-REFS / I-PDU-TRIGGERING-REF
RX-IDENTIFIER-RANGE / LOWER-CAN-ID	RX-IDENTIFIER-RANGE / LOWER-CAN-ID
RX-IDENTIFIER-RANGE / UPPER-CAN-ID	RX-IDENTIFIER-RANGE / UPPER-CAN-ID

8.1.2.2.2. FLEXRAY-FRAME-TRIGGERING

The FLEXRAY-FRAME-TRIGGERING is upgraded as described in [Section 8.1.1.2.2, “FLEXRAY-FRAME-TRIGGERING”](#).

8.1.2.2.3. FLEXRAY-ABSOLUTELY-SCHEDULED-TIMING

The FLEXRAY-ABSOLUTELY-SCHEDULED-TIMING is upgraded as described in [Section 8.1.1.2.3, “FLEXRAY-ABSOLUTELY-SCHEDULED-TIMING”](#).

8.1.2.2.4. LIN-FRAME-TRIGGERING

Source model entity: LIN-FRAME-TRIGGERING

Target model entity: LIN-FRAME-TRIGGERING

Target model property	Source model property
CATEGORY	CATEGORY
FRAME-PORT-REFS / FRAME-PORT-REF	FRAME-PORT-REFS / FRAME-PORT-REF
FRAME-REF	FRAME-REF
IDENTIFIER	IDENTIFIER
LIN-CHECKSUM	LIN-CHECKSUM
PDU-TRIGGERINGS / PDU-TRIGGERING-REF-CONDITIONAL	I-PDU-TRIGGERING-REFS / I-PDU-TRIGGERING-REF

Changed or moved parameters

RELATIVELY-SCHEDULED-TIMING

The source model RELATIVELY-SCHEDULED-TIMING entities aggregated in LIN-FRAME-TRIGGERING/RELATIVELY-SCHEDULED-TIMINGS are now aggregated in the target model entity LIN-SCHEDULE-TABLE, see [Section 8.1.2.2.10, “LIN-SCHEDULE-TABLE”](#).

8.1.2.2.5. CAN-FRAME

CAN-FRAME is upgraded as described in [Section 8.1.1.2.5, “CAN-FRAME”](#).

8.1.2.2.6. FLEXRAY-FRAME

FLEXRAY-FRAME is upgraded as described in [Section 8.1.1.2.6, “FLEXRAY-FRAME”](#).

8.1.2.2.7. LIN-UNCONDITIONAL-FRAME

LIN-UNCONDITIONAL-FRAME is upgraded as described in [Section 8.1.1.2.7, “LIN-UNCONDITIONAL-FRAME”](#).

8.1.2.2.8. LIN-SPORADIC-FRAME

LIN-SPORADIC-FRAME is upgraded as described in [Section 8.1.1.2.8, “LIN-SPORADIC-FRAME”](#).

8.1.2.2.9. LIN-EVENT-TRIGGERED-FRAME

LIN-EVENT-TRIGGERED-FRAME is upgraded as described in [Section 8.1.1.2.9, “LIN-EVENT-TRIGGERED-FRAME”](#).

8.1.2.2.10. LIN-SCHEDULE-TABLE

LIN-SCHEDULE-TABLE is upgraded as described in [Section 8.1.1.2.10, “LIN-SCHEDULE-TABLE”](#).

8.1.2.2.11. ASSIGN-FRAME-ID

ASSIGN-FRAME-ID is upgraded as described in [Section 8.1.1.2.11, “ASSIGN-FRAME-ID”](#).

8.1.2.2.12. UNASSIGN-FRAME-ID

UNASSIGN-FRAME-ID is upgraded as described in [Section 8.1.1.2.12, “UNASSIGN-FRAME-ID”](#).

8.1.2.2.13. ASSIGN-NAD

ASSIGN-NAD is upgraded as described in [Section 8.1.1.2.13, “ASSIGN-NAD”](#).

8.1.2.2.14. FREE-FORMAT

FREE-FORMAT is upgraded as described in [Section 8.1.1.2.14, “FREE-FORMAT”](#).

8.1.2.2.15. PDU-TRIGGERING

PDU-TRIGGERING is upgraded as described in [Section 8.1.1.2.15, “PDU-TRIGGERING”](#).

8.1.2.2.16. I-SIGNAL-TRIGGERING

I-SIGNAL-TRIGGERING is upgraded as described in [Section 8.1.1.2.16, “I-SIGNAL-TRIGGERING”](#).

8.1.2.2.17. DCM-I-PDU

DCM-I-PDU is upgraded as described in [Section 8.1.1.2.17, “DCM-I-PDU”](#).

8.1.2.2.18. I-SIGNAL-I-PDU

I-SIGNAL-I-PDU is upgraded as described in [Section 8.1.1.2.18, “I-SIGNAL-I-PDU”](#).

8.1.2.2.19. MULTIPLEXED-I-PDU

Source model entity: MULTIPLEXED-I-PDU

Target model entity: MULTIPLEXED-I-PDU

Target model property	Source model property
CATEGORY	CATEGORY
DYNAMIC-PARTS/DYNAMIC-PART	DYNAMIC-PART

Target model property	Source model property
DYNAMIC-PART-ALTERNATIVES/DYNAMIC-PART-ALTERNATIVE	DYNAMIC-PART-ALTERNATIVES/DYNAMIC-PART-ALTERNATIVE
LENGTH	LENGTH
SELECTOR-FIELD-BYTE-ORDER	SELECTOR-FIELD-BYTE-ORDER
SELECTOR-FIELD-LENGTH	SELECTOR-FIELD-LENGTH
SELECTOR-FIELD-START-POSITION	SELECTOR-FIELD-START-POSITION
STATIC-PARTS/STATIC-PART	STATIC-PART
TRIGGER-MODE	TRIGGER-MODE
UNUSED-BIT-PATTERN	UNUSED-BIT-PATTERN

Changed or moved parameters

LENGTH

During the upgrade, the parameter LENGTH's value given in units of bits in the source model is converted to bytes as required by the target model.

8.1.2.2.20. DYNAMIC-PART

Source model entity: DYNAMIC-PART

Target model entity: DYNAMIC-PART

Target model property	Source model property
DYNAMIC-PART-ALTERNATIVES/DYNAMIC-PART-ALTERNATIVE	DYNAMIC-PART-ALTERNATIVES/DYNAMIC-PART-ALTERNATIVE
SEGMENT-POSITIONS/SEGMENT-POSITION	SEGMENT-POSITIONS/SEGMENT-POSITION

8.1.2.2.21. DYNAMIC-PART-ALTERNATIVE

Source model entity: DYNAMIC-PART-ALTERNATIVE

Target model entity: DYNAMIC-PART-ALTERNATIVE

Target model property	Source model property
I-PDU-REF	I-PDU-REF
SELECTOR-FIELD-CODE	SELECTOR-FIELD-CODE

Target model property	Source model property
INITIAL-DYNAMIC-PART	INITIAL-DYNAMIC-PART

8.1.2.2.22. STATIC-PART

Source model entity: STATIC-PART

Target model entity: STATIC-PART

Target model property	Source model property
I-PDU-REF	I-PDU-REF
SEGMENT-POSITIONS/SEGMENT-POSITION	SEGMENT-POSITIONS/SEGMENT-POSITION

8.1.2.2.23. SEGMENT-POSITION

Source model entity: SEGMENT-POSITION

Target model entity: SEGMENT-POSITION

Target model property	Source model property
SEGMENT-BYTE-ORDER	SEGMENT-BYTE-ORDER
SEGMENT-LENGTH	SEGMENT-LENGTH
SEGMENT-POSITION	SEGMENT-POSITION

8.1.2.2.24. N-PDU

Source model entity: N-PDU

Target model entity: N-PDU

Target model property	Source model property
CATEGORY	CATEGORY
LENGTH	LENGTH

Changed or moved parameters

LENGTH

During the upgrade process, the parameter LENGTH's value given in units of bits in the source model is converted to bytes as required by the target model.

8.1.2.2.25. NM-PDU

Source model entity: NM-PDU

Target model entity: NM-PDU

Target model property	Source model property
CATEGORY	CATEGORY
LENGTH	LENGTH
I-SIGNAL-TO-I-PDU-MAPPINGS/I-SIGNAL-TO-I-PDU-MAPPING	I-SIGNAL-TO-I-PDU-MAPPINGS/I-SIGNAL-TO-I-PDU-MAPPING
NM-VOTE-INFORMATION	No source model parameter is available.

Changed or moved parameters

LENGTH

During the upgrade process, the parameter LENGTH's value given in units of bits in the source model is converted to bytes as required by the target model.

NM-CBV-POSITION, NM-NID-POSITION, and NM-USER-DATA-LENGTH

The source model parameters NM-CBV-POSITION, NM-NID-POSITION, and NM-USER-DATA-LENGTH are used to configure parameter values of the target model entity CAN-NM-CLUSTER, see [Section 8.1.2.7.3, “CAN-NM-CLUSTER”](#).

NM-DATA- INFORMATION

The target model parameter NM-DATA- INFORMATION is set to true, if the source model parameter NM-USER-DATA-LENGTH is greater than zero or at least one of the optional parameters NM-CBV-POSITION, NM-NID-POSITION is present. Otherwise NM-DATA- INFORMATION is set to false.

8.1.2.2.26. USER-DEFINED-PDU

Source model entity: USER-DEFINED-PDU

Target model entity: USER-DEFINED-PDU

Target model property	Source model property
CDD-TYPE	CDD-TYPE

8.1.2.2.27. USER-DEFINED-I-PDU

Source model entity: USER-DEFINED-I-PDU

Target model entity: USER-DEFINED-I-PDU

Target model property	Source model property
CDD-TYPE	CDD-TYPE
LENGTH	LENGTH

Changed or moved parameters

LENGTH

During the upgrade process, the parameter LENGTH's value given in units of bits in the source model is converted to bytes as required by the target model.

8.1.2.2.28. PDU-TO-FRAME-MAPPING

PDU-TO-FRAME-MAPPING is upgraded as described in [Section 8.1.1.2.26, “PDU-TO-FRAME-MAPPING”](#).

8.1.2.2.29. I-PDU-TIMING

Source model entity: I-PDU-TIMING-SPECIFICATION

Target model entity: I-PDU-TIMING

Target model property	Source model property
MINIMUM-DELAY	MINIMUM-DELAY
TRANSMISSION-MODE-DECLARATION	TRANSMISSION-MODE-DECLARATION
No target model parameter is available.	REQUEST-CONTROLLED-TIMING

8.1.2.2.30. I-SIGNAL-I-PDU-GROUP

I-SIGNAL-I-PDU-GROUP is upgraded as described in [Section 8.1.1.2.28, “I-SIGNAL-I-PDU-GROUP”](#).

8.1.2.2.31. PDUR-I-PDU-GROUP

Source model entity: PDUR-I-PDU-GROUP

Target model entity: PDUR-I-PDU-GROUP

Target model property	Source model property
COMMUNICATION-MODE	COMMUNICATION-MODE
I-PDUS/PDU-TRIGGERING-REF-CONDITION-AL/PDU-TRIGGERING-REF	I-PDU-REFS/I-PDU-REF

8.1.2.2.32. I-SIGNAL-TO-I-PDU-MAPPING

I-SIGNAL-TO-I-PDU-MAPPING is upgraded as described in [Section 8.1.1.2.29, “I-SIGNAL-TO-I-PDU-MAPPING”](#).

8.1.2.2.33. I-SIGNAL

I-SIGNAL is upgraded as described in [Section 8.1.1.2.30, “I-SIGNAL”](#).

8.1.2.2.34. I-SIGNAL-GROUP

I-SIGNAL-GROUP is upgraded as described in [Section 8.1.1.2.31, “I-SIGNAL-GROUP”](#).

8.1.2.2.35. SYSTEM-SIGNAL

SYSTEM-SIGNAL is upgraded as described in [Section 8.1.1.2.32, “SYSTEM-SIGNAL”](#).

8.1.2.2.36. I-SIGNAL-PORT

Source model entity: SIGNAL-PORT

Target model entity: I-SIGNAL-PORT

Target model property	Source model property
CATEGORY	CATEGORY
COMMUNICATION-DIRECTION	COMMUNICATION-DIRECTION
TIMEOUT	TIMEOUT
DATA-FILTER	DATA-FILTER

8.1.2.2.37. FRAME-PORT

FRAME-PORT is upgraded as described in [Section 8.1.1.2.34, “FRAME-PORT”](#).

8.1.2.2.38. I-PDU-PORT

I-PDU-PORT is upgraded as described in [Section 8.1.1.2.35, “I-PDU-PORT”](#).

8.1.2.2.39. TRANSMISSION-MODE-DECLARATION

Source model entity: TRANSMISSION-MODE-DECLARATION

Target model entity: TRANSMISSION-MODE-DECLARATION

Target model property	Source model property
TRANSMISSION-MODE-CONDITIONS / TRANSMISSION-MODE-CONDITION	TRANSMISSION-MODE-CONDITIONS / TRANSMISSION-MODE-CONDITION
TRANSMISSION-MODE-FALSE-TIMING	TRANSMISSION-MODE-FALSE-TIMING

Changed or moved parameters

TRANSMISSION-MODE-TRUE-TIMING

All source model timing elements residing in I-PDU-TIMING-SPECIFICATION of the source model SIG-NAL-I-PDU are moved into the target model TRANSMISSION-MODE-TRUE-TIMING element.

8.1.2.2.40. TRANSMISSION-MODE-CONDITION

TRANSMISSION-MODE-CONDITION is upgraded as described in [Section 8.1.1.2.37, “TRANSMISSION-MODE-CONDITION”](#).

8.1.2.2.41. TRANSMISSION-MODE-FALSE-TIMING

TRANSMISSION-MODE-FALSE-TIMING is upgraded as described in [Section 8.1.1.2.38, “TRANSMISSION-MODE-FALSE-TIMING”](#).

8.1.2.2.42. EVENT-CONTROLLED-TIMING

EVENT-CONTROLLED-TIMING is upgraded as described in [Section 8.1.1.2.39, “EVENT-CONTROLLED-TIMING”](#).

8.1.2.2.43. CYCLIC-TIMING

CYCLIC-TIMING is upgraded as described in [Section 8.1.1.2.40, “CYCLIC-TIMING”](#).

8.1.2.2.44. TIME-RANGE-TYPE

TIME-RANGE-TYPE is upgraded as described in [Section 8.1.1.2.41, “TIME-RANGE-TYPE”](#).

8.1.2.2.45. ABSOLUTE-TOLERANCE

ABSOLUTE-TOLERANCE is upgraded as described in [Section 8.1.1.2.42, “ABSOLUTE-TOLERANCE”](#).

8.1.2.2.46. RELATIVE-TOLERANCE

RELATIVE-TOLERANCE is upgraded as described in [Section 8.1.1.2.43, “RELATIVE-TOLERANCE”](#).

8.1.2.3. Gateway

8.1.2.3.1. FRAME-MAPPING

FRAME-MAPPING is upgraded as described in [Section 8.1.1.3.1, “FRAME-MAPPING”](#).

8.1.2.3.2. I-PDU-MAPPING

Source model entity: I-PDU-MAPPING

Target model entity: I-PDU-MAPPING

Target model property	Source model property
PDUR-TP-CHUNK-SIZE	PDUR-TP-CHUNK-SIZE
SOURCE-I-PDU-REF	SOURCE-I-PDU/SOURCE-I-PDU-REF
TARGET-I-PDU/TARGET-I-PDU-REF	TARGET-I-PDU/TARGET-I-PDU-REF
No target model parameter is available.	SOURCE-I-PDU-REF/PDU-MAPPINGS-DE-FAULT-VALUE

8.1.2.3.3. TARGET-I-PDU-REF

TARGET-I-PDU-REF is upgraded as described in [Section 8.1.1.3.3, “TARGET-I-PDU-REF”](#).

8.1.2.3.4. PDU-MAPPING-DEFAULT-VALUE

PDU-MAPPING-DEFAULT-VALUE is upgraded as described in [Section 8.1.1.3.4, “PDU-MAPPING-DEFAULT-VALUE”](#).

8.1.2.3.5. DEFAULT-VALUE-ELEMENT

DEFAULT-VALUE-ELEMENT is upgraded as described in [Section 8.1.1.3.5, “DEFAULT-VALUE-ELEMENT”](#).

8.1.2.3.6. I-SIGNAL-MAPPING

I-SIGNAL-MAPPING is upgraded as described in [Section 8.1.1.3.6, “I-SIGNAL-MAPPING”](#).

8.1.2.4. CAN Transport Layer

8.1.2.4.1. CAN-TP-CONFIG

For each CAN-CLUSTER in the source model that has at least one CAN-TP-CONNECTION-CHANNEL aggregated in its PHYSICAL-CHANNEL, one CAN-TP-CONFIG element is created in the target model. The target model element CAN-TP-CONFIG resides in the same AR-PACKAGE as the related CAN-CLUSTER.

New parameters in the target model

CAN-TP-ECU

For each ECU-INSTANCE in the source model that is attached to the PHYSICAL-CHANNEL, one CAN-TP-ECU element is created in the target model. The target model element CAN-TP-ECU references the target model counterpart of the source model ECU-INSTANCE in ECU-INSTANCE-REF. All created CAN-TP-ECU elements are aggregated in CAN-TP-CONFIG/TP-ECUS.

Moved parameters in the target model

CAN-TP-NODE

For each source model CAN-TP-NODE aggregated in a PHYSICAL-CHANNEL of a CAN-CLUSTER for which a CAN-TP-CONFIG has been created in the target model, one CAN-TP-NODE is added. CAN-TP-NODE is

aggregated in CAN-TP-CONFIG/TP-NODES. [Section 8.1.2.4.2, “CAN-TP-NODE”](#) describes the upgrade in detail.

CAN-TP-ADDRESS

For each source model TP-ADDRESS aggregated in a PHYSICAL-CHANNEL of a CAN-CLUSTER for which a CAN-TP-CONFIG has been created in the target model, one CAN-TP-ADDRESS is added. CAN-TP-ADDRESS is aggregated in CAN-TP-CONFIG/TP-ADDRESSSS. [Section 8.1.2.4.3, “CAN-TP-ADDRESS”](#) describes the upgrade in detail.

CAN-TP-CONNECTION

For each source model CAN-TP-CONNECTION-CHANNEL aggregated in a PHYSICAL-CHANNEL of a CAN-CLUSTER for which a CAN-TP-CONFIG has been created in the target model, one CAN-TP-CHANNEL is created and aggregated in CAN-TP-CONFIG/TP-CHANNELS. Depending on the CHANNEL-MODE-TYPE of the CAN-TP-CONNECTION, one or two CAN-TP-CONNECTION-CHANNELS are created in the target model and aggregated in CAN-TP-CONFIG/TP-CONNECTIONS. If the CHANNEL-MODE-TYPE of the source model CAN-TP-CONNECTION is HALF-DUPLEX-MODE, one CAN-TP-CONNECTION-CHANNEL is created in the target model. If the CHANNEL-MODE-TYPE of the source model CAN-TP-CONNECTION is FULL-DUPLEX-MODE, two CAN-TP-CONNECTION-CHANNELS are created. [Section 8.1.2.4.5, “CAN-TP-CONNECTION”](#) and [Section 8.1.2.4.4, “CAN-TP-CHANNEL”](#) describes the upgrade in detail.

8.1.2.4.2. CAN-TP-NODE

Source model entity: CAN-TP-NODE

Target model entity: CAN-TP-NODE

Target model property	Source model property
CATEGORY	CATEGORY
CONNECTOR-REF	CONNECTOR-REFS/CONNECTOR-REF
TP-ADDRESS-REF	TP-ADDRESS-REF
MAX-FC-WAIT	RX-WFT-MAX
TIMEOUT-AR	TIMEOUT-AR

New parameters in the target model

TIMEOUT-AS

The target model parameter TIMEOUT-AS is set to the minimum TIMEOUT-AS value of all source model CAN-TP-CONNECTION-CHANNELS that reference the CAN-TP-NODE as sending node.

ST-MIN

The target model parameter ST-MIN is set to the maximum MINIMUM-SEPARATION-TIME value of all source model CAN-TP-CONNECTION-CHANNELS that reference the CAN-TP-NODE as receiving node.

8.1.2.4.3. CAN-TP-ADDRESS

Source model entity: TP-ADDRESS

Target model entity: CAN-TP-ADDRESS

Target model property	Source model property
CATEGORY	CATEGORY
TP-ADDRESS	TP-ADDRESS

8.1.2.4.4. CAN-TP-CHANNEL

Source model entity: CAN-TP-CONNECTION-CHANNEL

Target model entity: CAN-TP-CHANNEL

Target model property	Source model property
CHANNEL-MODE	CHANNEL-MODE-TYPE

8.1.2.4.5. CAN-TP-CONNECTION

Source model entity: CAN-TP-CONNECTION-CHANNEL

Target model entity: CAN-TP-CONNECTION

Target model property	Source model property
ADDRESSING-FORMAT	ADDRESSING-FORMAT
MAX-BLOCK-SIZE	BLOCK-SIZE
DATA-PDU-REF	DATA-PDU-REF
FLOW-CONTROL-PDU-REF	FLOW-CONTROL-PDU-REF
MAX-BLOCK-SIZE	BLOCK-SIZE
PADDING-ACTIVATION	PADDING-ACTIVATION
TRANSMITTER-REF	SOURCE-REF
TA-TYPE	TA-TYPE
RECEIVER-REFS/RECEIVER-REF	TARGET-REFS/TARGET-REF
TIMEOUT-BR	TIMEOUT-BR

Target model property	Source model property
TIMEOUT-BS	TIMEOUT-BS
TIMEOUT-CR	TIMEOUT-CR
TIMEOUT-CS	TIMEOUT-CS
TP-SDU-REF	TP-SDU-REF
No target model parameter is available.	MULTICAST-ADDRESSING
MULTICAST-REF	MULTICAST-REF

New parameters in the target model

CAN-TP-CHANNEL-REF

Every created CAN-TP-CONNECTION references the CAN-TP-CHANNEL it belongs to in CAN-TP-CHAN-NEL-REF.

Changed or moved parameters

MINIMUM-SEPARATION-TIME

The parameter MINIMUM-SEPARATION-TIME is upgraded to the receiver CAN-TP-NODE, see [Section 8.1.2.4.2, “CAN-TP-NODE”](#).

TIMEOUT-AS

The parameter TIMEOUT-AS is upgraded to the sender CAN-TP-NODE, see [Section 8.1.2.4.2, “CAN-TP-NODE”](#).

8.1.2.4.6. Second CAN-TP-CONNECTION for FULL-DUPLEX-MODE

Source model entity: CAN-TP-CONNECTION-CHANNEL

Target model entity: CAN-TP-CONNECTION

The second target model parameter CAN-TP-CONNECTION is only created for CAN-TP-CONNECTION-CHAN-NELS in FULL-DUPLEX-MODE.

Target model property	Source model property
ADDRESSING-FORMAT	ADDRESSING-FORMAT
MAX-BLOCK-SIZE	BLOCK-SIZE

Target model property	Source model property
FLOW-CONTROL-PDU-REF	DATA-PDU-REF
DATA-PDU-REF	FLOW-CONTROL-PDU-REF
MAX-BLOCK-SIZE	BLOCK-SIZE
PADDING-ACTIVATION	PADDING-ACTIVATION
RECEIVER-REFS/RECEIVER-REF	SOURCE-REF
TA-TYPE	TA-TYPE
TRANSMITTER-REF	TARGET-REFS/TARGET-REF
TIMEOUT-BR	TIMEOUT-BR
TIMEOUT-BS	TIMEOUT-BS
TIMEOUT-CR	TIMEOUT-CR
TIMEOUT-CS	TIMEOUT-CS
No target model parameter is available.	MULTICAST-ADDRESSING
MULTICAST-REF	MULTICAST-REF

8.1.2.5. FlexRay ISO Transport Layer

NOTE



Non-ISO FlexRay Transport Layer upgrade is not supported

The system model upgrade only supports the upgrade of the FlexRay ISO Transport Layer.
The upgrade of the Non-ISO FlexRay Transport Layer variant is *not* supported.

8.1.2.5.1. FLEXRAY-TP-CONFIG

For each FLEXRAY-CLUSTER in the source model that contains at least one FLEXRAY-ISO-TP-CONNECTION in one of its PHYSICAL-CHANNELS, one FLEXRAY-TP-CONFIG element is created in the target model. The target model element FLEXRAY-TP-CONFIG resides in the same AR-PACKAGE as the related FLEXRAY-CLUSTER.

New parameters in the target model

FLEXRAY-TP-ECU

For each ECU-INSTANCE in the source model that is attached to one of the PHYSICAL-CHANNELS, one FLEXRAY-TP-ECU element is created in the target model. In ECU-INSTANCE-REF the FLEXRAY-TP-

ECU element references the target model counterpart of the source model ECU-INSTANCE. All created FLEXRAY-TP-ECU elements are aggregated in FLEXRAY-TP-CONFIG/TP-ECUS. [Section 8.1.2.5.2, “FLEXRAY-TP-ECU”](#) describes the upgrade in detail.

Moved parameters in the target model

FLEXRAY-TP-NODE

For each source model FLEXRAY-ISO-TP-NODE that is attached to any PHYSICAL-CHANNEL of a FLEXRAY-CLUSTER, and for which a FLEXRAY-TP-CONFIG has been created in the target model, one FLEXRAY-TP-NODE is added. FLEXRAY-TP-NODE is aggregated in FLEXRAY-TP-CONFIG/TP-NODES. [Section 8.1.2.5.3, “FLEXRAY-TP-NODE”](#) describes the upgrade in detail.

TP-ADDRESS

For each source model TP-ADDRESS attached to any PHYSICAL-CHANNEL of a FLEXRAY-CLUSTER for which a FLEXRAY-TP-CONFIG has been created in the target model, one FLEXRAY-TP-NODE is added. TP-ADDRESS is aggregated in FLEXRAY-TP-CONFIG/TP-ADDRESS. [Section 8.1.2.5.4, “TP-ADDRESS”](#) describes the upgrade in detail.

FLEXRAY-TP-CONNECTION

For each source model FLEXRAY-ISO-TP-CONNECTION attached to any PHYSICAL-CHANNEL of a FLEXRAY-CLUSTER for which a FLEXRAY-TP-CONFIG has been created in the target model, one FLEXRAY-TP-CONNECTION is added. FLEXRAY-TP-CONNECTION is aggregated in FLEXRAY-TP-CONFIG/TP-CONNECTIONS. [Section 8.1.2.5.5, “FLEXRAY-TP-CONNECTION”](#) describes the upgrade in detail.

FLEXRAY-TP-CONNECTION-CONTROL

For each source model FLEXRAY-ISO-TP-CONNECTION-CONTROL attached to any PHYSICAL-CHANNEL of a FLEXRAY-CLUSTER for which a FLEXRAY-TP-CONFIG has been created in the target model, one FLEXRAY-TP-CONNECTION-CONTROL is added. FLEXRAY-TP-CONNECTION-CONTROL is aggregated in FLEXRAY-TP-CONFIG/TP-CONNECTION-CONTROLS. [Section 8.1.2.5.6, “FLEXRAY-TP-CONNECTION-CONTROL”](#) describes the upgrade in detail.

FLEXRAY-TP-PDU-POOL

For each source model FLEXRAY-ISO-TP-PDU-POOL attached to any PHYSICAL-CHANNEL of a FLEXRAY-CLUSTER for which a FLEXRAY-TP-CONFIG has been created in the target model, one FLEXRAY-TP-PDU-POOL is added. FLEXRAY-TP-PDU-POOL is aggregated in FLEXRAY-TP-CONFIG/PDU-POOLS. [Section 8.1.2.5.7, “FLEXRAY-TP-PDU-POOL”](#) describes the upgrade in detail.

8.1.2.5.2. FLEXRAY-TP-ECU

Source model entity: ECU-INSTANCE

Target model entity: FLEXRAY-TP-ECU

Target model property	Source model property
CYCLE-TIME-MAIN-FUNCTION	CYCLE-TIME-MAIN-FUNCTION
FULL-DUPLEX-ENABLED	FULL-DUPLEX-ENABLED

8.1.2.5.3. FLEXRAY-TP-NODE

Source model entity: FLEXRAY-ISO-TP-NODE

Target model entity: FLEXRAY-TP-NODE

Target model property	Source model property
CONNECTOR-REFS/CONNECTOR-REF	CONNECTOR-REFS/CONNECTOR-REF
TP-ADDRESS-REF	TP-ADDRESS-REF

8.1.2.5.4. TP-ADDRESS

Source model entity: TP-ADDRESS

Target model entity: TP-ADDRESS

Target model property	Source model property
TP-ADDRESS	TP-ADDRESS

8.1.2.5.5. FLEXRAY-TP-CONNECTION

Source model entity: FLEXRAY-ISO-TP-CONNECTION

Target model entity: FLEXRAY-TP-CONNECTION

Target model property	Source model property
BANDWIDTH-LIMITATION	BANDWIDTH-LIMITATION
DIRECT-TP-SDU-REF	DIRECT-TP-SDU-REF
MULTICAST-REF	MULTICAST-REF
RECEIVER-REFS/RECEIVER-REF	RECEIVER-REFS/RECEIVER-REF
REVERSED-TP-SDU-REF	REVERSED-TP-SDU-REF
RX-PDU-POOL-REF	RX-PDU-POOL-REF
TP-CONNECTION-CONTROL-REF	TP-CONNECTION-CONTROL-REF

Target model property	Source model property
TRANSMITTER-REF	TRANSMITTER-REF
TX-PDU-POOL-REF	TX-PDU-POOL-REF

8.1.2.5.6. FLEXRAY-TP-CONNECTION-CONTROL

Source model entity: FLEXRAY-ISO-TP-CONNECTION-CONTROL

Target model entity: FLEXRAY-TP-CONNECTION-CONTROL

Target model property	Source model property
ACK-TYPE	ACK-TYPE
MAX-AR	MAX-AR
MAX-AS	MAX-AS
MAX-BUFFER-SIZE	MAX-BUFFER-SIZE
MAX-FC-WAIT	MAX-FC-WAIT
MAX-FR-IF	MAX-FR-IF
MAX-NUMBER-OF-NPDU-PER-CYCLE	MAX-NUMBER-OF-NPDU-PER-CYCLE
MAX-RETRIES	MAX-RETRIES
SEPARATION-CYCLE-EXPONENT	SEPARATION-CYCLE-EXPONENT
TIME-BR	TIME-BR
TIME-BUFFER	TIME-BUFFER
TIME-FR-IF	TIME-FR-IF
TIMEOUT-AR	TIMEOUT-AR
TIMEOUT-AS	TIMEOUT-AS
TIMEOUT-BS	TIMEOUT-BS
TIMEOUT-CR	TIMEOUT-CR

The source model ACK-TYPE parameter value ACK-WITHOUT-RT is not supported in the target model. If this source model parameter value is detected during model upgrade, the corresponding destination parameter is left undefined.

8.1.2.5.7. FLEXRAY-TP-PDU-POOL

Source model entity: FLEXRAY-ISO-TP-PDU-POOL

Target model entity: FLEXRAY-TP-PDU-POOL

Target model property	Source model property
N-PDU-REFS/N-PDU-REF	N-PDU-REFS/N-PDU-REF

8.1.2.6. LIN Transport Layer

8.1.2.6.1. LIN-TP-CONFIG

For each LIN-CLUSTER in the source model that contains at least one LIN-TP-CHANNEL in its PHYSICAL-CHANNEL, one LIN-TP-CONFIG element is created in the target model. LIN-TP-CONFIG resides in the same AR-PACKAGE as the related LIN-CLUSTER.

Moved parameters in the target model

LIN-TP-NODE

For each source model LIN-TP-NODE attached to the PHYSICAL-CHANNEL of a LIN-CLUSTER for which a LIN-TP-CONFIG has been created in the target model, one LIN-TP-NODE is added. LIN-TP-NODE is aggregated in LIN-TP-CONFIG/TP-NODES. [the section called “LIN-TP-NODE”](#) describes the upgrade in detail.

TP-ADDRESS

For each source model TP-ADDRESS attached to the PHYSICAL-CHANNEL of a LIN-CLUSTER for which a LIN-TP-CONFIG has been created in the target model, one TP-ADDRESS is added. TP-ADDRESS is aggregated in LIN-TP-CONFIG/TP-ADDRESSES. [the section called “TP-ADDRESS”](#) describes the upgrade in detail.

LIN-TP-CONNECTION

For each source model LIN-TP-CHANNEL attached to the PHYSICAL-CHANNEL of a LIN-CLUSTER for which a LIN-TP-CONFIG has been created in the target model, one LIN-TP-CONNECTION is added. LIN-TP-CONNECTION is aggregated in LIN-TP-CONFIG/TP-CONNECTIONS. [the section called “LIN-TP-CONNECTION”](#) describes the upgrade in detail.

LIN-TP-NODE

Source model entity: LIN-TP-NODE

Target model entity: LIN-TP-NODE

Target model property	Source model property
CONNECTOR-REF	CONNECTOR-REFS/CONNECTOR-REF
TP-ADDRESS-REF	TP-ADDRESS-REF
MAX-NUMBER-OF-RESP-PENDING-FRAMES	No source model parameter is available.
P-2-MAX	No source model parameter is available.
P-2-TIMING	No source model parameter is available.

TP-ADDRESS

Source model entity: TP-ADDRESS

Target model entity: TP-ADDRESS

Target model property	Source model property
TP-ADDRESS	TP-ADDRESS

LIN-TP-CONNECTION

Source model entity: LIN-TP-CHANNEL

Target model entity: LIN-TP-CONNECTION

Target model property	Source model property
DATA-PDU-REF	DATA-PDU-REF
DROP-NOT-REQUESTED-NAD	DROP-NOT-REQUESTED-NAD
FLOW-CONTROL-REF	FLOW-CONTROL-REF
LIN-TP-N-SDU-REF	LIN-TP-N-SDU-REF
MULTICAST-REF	MULTICAST-REF
TRANSMITTER-REF	SOURCE-REF
RECEIVER-REFS/RECEIVER-REF	TARGET-REFS/TARGET-REF

8.1.2.7. Network Management

8.1.2.7.1. NM-CONFIG

NM-CONFIG is upgraded as described in [Section 8.1.1.7.1, “NM-CONFIG”](#).

8.1.2.7.2. NM-ECU

For every ECU-INSTANCE in the source model, one NM-ECU element is created in the target model. The related ECU-INSTANCE is referenced in ECU-INSTANCE-REF. The NM-ECU is aggregated in NM-CONFIG/NM-IF-ECUS.

New parameters in the target model

NM-NODE-DETECTION-ENABLED

NM-NODE-DETECTION-ENABLED is configured by taking the maximum value of NM-NODE-DETECTION-ENABLED of any FLEXRAY-COMMUNICATION-CONTROLLER of the ECU-INSTANCE. If none of these FLEXRAY-COMMUNICATION-CONTROLLER elements has the value set, NM-NODE-DETECTION-ENABLED's maximum value is taken of all COMMUNICATION-CLUSTER elements to which the ECU-INSTANCE is connected.

NM-NODE-ID-ENABLED

NM-NODE-ID-ENABLED is determined by taking NM-NODE-ID-ENABLED's maximum value of all COMMUNICATION-CLUSTER elements to which the ECU-INSTANCE is connected.

NM-USER-DATA-ENABLED

NM-USER-DATA-ENABLED is configured by taking NM-USER-DATA-ENABLED's maximum value of all CAN-COMMUNICATION-CONTROLLERS and FLEXRAY-COMMUNICATION-CONTROLLERS to which the ECU-INSTANCE is connected.

8.1.2.7.3. CAN-NM-CLUSTER

Source model entity: CAN-CLUSTER

Target model entity: CAN-NM-CLUSTER

Target model property	Source model property
NM-BUSLOAD-REDUCTION-ACTIVE	NM-BUS-LOAD-REDUCTION-ACTIVE
NM-NETWORK-TIMEOUT	NM-TIMEOUT-TIME
NM-REPEAT-MESSAGE-TIME	NM-REPEAT-MESSAGE-STATE-TIME
NM-REMOTE-SLEEP-INDICATION-TIME	NM-REMOTE-SLEEP-INDICATION-TIME
NM-WAIT-BUS-SLEEP-TIME	NM-WAIT-BUS-SLEEP-TIME
NM-CHANNEL-SLEEP-MASTER	No source model parameter is available.
NM-IMMEDIATE-NM-CYCLE-TIME	NM-IMMEDIATE-NM-CYCLE-TIME
NM-MSG-CYCLE-TIME	NM-MSG-CYCLE-TIME

Target model property	Source model property
NM-REMOTE-SLEEP-INDICATION-TIME	NM-REMOTE-SLEEP-INDICATION-TIME

The created CAN-NM-CLUSTER is aggregated in NM-CONFIG/NM-CLUSTERS. It references the related CAN-CLUSTER in COMMUNICATION-CLUSTER-REF.

New parameters in the target model

NM-CBV-POSITION, NM-NID-POSITION

NM-CBV-POSITION, NM-NID-POSITION are calculated by taking NM-CBV-POSITION, NM-NID-POSITION's maximum value of any NM-PDU transmitted over the CAN-CLUSTER in the source model, and then dividing the result by eight.

NM-USER-DATA-LENGTH

NM-USER-DATA-LENGTH is calculated by taking NM-USER-DATA-LENGTH's maximum value of any NM-PDU transmitted over the CAN-CLUSTER.

8.1.2.7.4. CAN-NM-NODE

For every CAN-COMMUNICATION-CONTROLLER connected to a CAN-CLUSTER in the source model, one CAN-NM-NODE entity is created. CAN-NM-NODE is aggregated in CAN-NM-CLUSTER/NM-NODES.

New parameters in the target model

NM-RANGE-CONFIG

If the source model COMMUNICATION-CLUSTER to which the CAN-COMMUNICATION-CONTROLLER is connected defines values for both NM-LOWER-CAN-ID and NM-UPPER-CAN-ID, one NM-RANGE-CONFIG element is added to the CAN-NM-NODE. NM-RANGE-CONFIG/LOWER-CAN-ID is set to the value of NM-LOWER-CAN-ID, and NM-RANGE-CONFIG/UPPER-CAN-ID is set to the value of NM-UPPER-CAN-ID.

CONTROLLER-REF

CONTROLLER-REF is configured to reference the target model counterpart of the CAN-COMMUNICATION-CONTROLLER.

NM-NODE-ID

The parameter NM-NODE-ID is derived from the NM-ADDRESS parameter value of COMMUNICATION-CONNECTOR that references the source model counterpart of the CAN-COMMUNICATION-CONTROLLER.

NM-IF-ECU-REF

NM-IF-ECU-REF is configured to reference the NM-ECU element, which has been created for the source model ECU-INSTANCE that aggregates the CAN-COMMUNICATION-CONTROLLER this CAN-NM-NODE has been created for.

RX-NM-PDU-REFS/RX-NM-PDU-REF

RX-NM-PDU-REFS/RX-NM-PDU-REF is configured to reference all NM-PDUS the CAN-COMMUNICATION-CONTROLLER is receiving.

TX-NM-PDU-REFS/TX-NM-PDU-REF

TX-NM-PDU-REFS/TX-NM-PDU-REF is configured to reference all NM-PDUS the CAN-COMMUNICATION-CONTROLLER is sending.

NM-MSG-REDUCED-TIME

NM-MSG-REDUCED-TIME is configured by CAN-COMMUNICATION-CONTROLLER/NM-MSG-REDUCED-TIME.

NM-MSG-CYCLE-OFFSET

NM-MSG-CYCLE-OFFSET is configured by CAN-COMMUNICATION-CONTROLLER/NM-MSG-CYCLE-OFFSET.

8.1.2.7.5. CAN-NM-CLUSTER-COUPLED

CAN-NM-CLUSTER-COUPLED defines parameters which have to be set consistently among two coupled CAN-NM-CLUSTER elements. [Section 8.1.1.7.5, “CAN-NM-CLUSTER-COUPLED”](#) describes the peculiarities of the CAN-NM-CLUSTER-COUPLED.

For each of the coupled sets of CAN-NM-CLUSTER elements, and for each CAN-NM-CLUSTER that is not coupled to any other CAN-NM-CLUSTER, one element is created, aggregated in CAN-NM-CONFIG/NM-CLUSTER-COUPLED.

New parameters in the target model

COUPLED-CLUSTER-REFS/COUPLED-CLUSTER-REF

COUPLED-CLUSTER-REFS/COUPLED-CLUSTER-REF is configured to reference all CAN-NM-CLUSTERS that are coupled.

NM-BUSLOAD-REDUCTION-ENABLED

If the NM-BUS-LOAD-REDUCTION-ENABLED parameters of the CAN-CLUSTER counterparts of the coupled CAN-NM-CLUSTERS yield a consistent value, the NM-BUSLOAD-REDUCTION-ENABLED of the CAN-NM-CLUSTER-COUPLED is configured accordingly.

8.1.2.7.6. FLEXRAY-NM-CLUSTER

One FLEXRAY-NM-CLUSTER is created in the target model for each FLEXRAY-CLUSTER in the source model. FLEXRAY-NM-CLUSTER is aggregated in NM-CONFIG/NM-CLUSTERS and references the related FLEXRAY-CLUSTER in COMMUNICATION-CLUSTER-REF.

New parameters in the target model

NM-REMOTE-SLEEP-INDICATION-TIME, NM-REPEAT-MESSAGE-TIME, NM-REPETITION-CYCLE, NM-VOTING-CYCLE, NM-DATA-CYCLE, NM-READY-SLEEP-COUNT
 NM-REMOTE-SLEEP-INDICATION-TIME, NM-REPEAT-MESSAGE-TIME, NM-REPETITION-CYCLE, NM-VOTING-CYCLE, NM-DATA-CYCLE, NM-READY-SLEEP-COUNT are defined by taking the respective parameter values of the source model FLEXRAY-CLUSTER.

NM-CONTROL-BIT-VECTOR-ACTIVE

NM-CONTROL-BIT-VECTOR-ACTIVE is defined by taking the maximum NM-CONTROL-BIT-VECTOR-ENABLED parameter value of all source model FLEXRAY-COMMUNICATION-CONTROLLERS that are connected to the FLEXRAY-CLUSTER.

8.1.2.7.7. FLEXRAY-NM-NODE

For every FLEXRAY-COMMUNICATION-CONTROLLER connected to a FLEXRAY-CLUSTER in the source model, one FLEXRAY-NM-NODE entity is created. FLEXRAY-NM-NODE is aggregated in FLEXRAY-NM-CLUSTER/NM-NODES.

New parameters in the target model

CONTROLLER-REF

CONTROLLER-REF is configured to reference the target model counterpart of the FLEXRAY-COMMUNICATION-CONTROLLER.

NM-NODE-ID

The parameter NM-NODE-ID is derived from NM-ADDRESS's maximum value of all COMMUNICATION-CONNECTORS that reference the source model counterpart of the FLEXRAY-COMMUNICATION-CONTROLLER.

NM-IF-ECU-REF

NM-IF-ECU-REF references the target model NM-ECU element that has been created for the source model ECU-INSTANCE to which the source model FLEXRAY-COMMUNICATION-CONTROLLER belongs. A source model FLEXRAY-COMMUNICATION-CONTROLLER belongs to a source model ECU-INSTANCE, if the ECU-INSTANCE aggregates the FLEXRAY-COMMUNICATION-CONTROLLER in COMM-CONTROLLERS.

RX-NM-PDU-REFS/RX-NM-PDU-REF

RX-NM-PDU-REFS/RX-NM-PDU-REF is configured to reference all NM-PDUS the FLEXRAY-COMMUNICATION-CONTROLLER is receiving.

TX-NM-PDU-REFS/TX-NM-PDU-REF

TX-NM-PDU-REFS/TX-NM-PDU-REF is configured to reference all NM-PDUS the FLEXRAY-COMMUNICATION-CONTROLLER is sending.

8.1.2.7.8. FLEXRAY-NM-CLUSTER-COUPING

FLEXRAY-NM-CLUSTER-COUPING defines parameters which have to be set consistently among two FLEXRAY-NM-CLUSTER elements which are coupled. [Section 8.1.1.7.5, “CAN-NM-CLUSTER-COUPING”](#) describes the peculiarities of the CAN cluster analogy CAN-NM-CLUSTER-COUPING.

For each of the coupled sets of FLEXRAY-NM-CLUSTER elements, and for each FLEXRAY-NM-CLUSTER not coupled to any other FLEXRAY-NM-CLUSTER, one FLEXRAY-NM-CLUSTER-COUPING element is created, aggregated in FLEXRAY-NM-CONFIG/NM-CLUSTER-COPLINGS.

8.1.2.8. Software Component Description

8.1.2.8.1. COMPOSITION-SW-COMPONENT-TYPE

The COMPOSITION-SW-COMPONENT-TYPE is upgraded as described in [Section 8.1.1.8.1, “COMPOSITION-SW-COMPONENT-TYPE”](#).

8.1.2.8.2. APPLICATION-SW-COMPONENT-TYPE

Source model entities: APPLICATION-SOFTWARE-COMPONENT-TYPE, SENSOR-ACTUATOR-SOFT-WARE-COMPONENT-TYPE

Target model entity: APPLICATION-SW-COMPONENT-TYPE

Target model property	Source model property
PORTS/P-PORT-PROTOTYPE	PORTS/P-PORT-PROTOTYPE
PORTS/R-PORT-PROTOTYPE	PORTS/R-PORT-PROTOTYPE
PORT-GROUPS/PORT-GROUP	PORT-GROUPS/PORT-GROUP

New parameters in the target model

INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR

The parameter INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR contains the SWC-INTERNAL-BEHAVIOR which corresponds to the source model INTERNAL-BEHAVIOR which references the APPLICATION-SOFTWARE-COMPONENT-TYPE via COMPONENT-REF.

8.1.2.8.3. SERVICE-SW-COMPONENT-TYPE

Source model entity: SERVICE-COMPONENT-TYPE

Target model entity: SERVICE-SW-COMPONENT-TYPE

Target model property	Source model property
PORTS/P-PORT-PROTOTYPE	PORTS/P-PORT-PROTOTYPE
PORTS/R-PORT-PROTOTYPE	PORTS/R-PORT-PROTOTYPE
PORT-GROUPS/PORT-GROUP	PORT-GROUPS/PORT-GROUP
No upgrade is currently implemented.	BSW-MODULE-DESCRIPTION-REFS/BSW-MODULE-DESCRIPTION-REF

New parameters in the target model

INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR

The parameter INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR contains the SWC-INTERNAL-BEHAVIOR which corresponds to the source model INTERNAL-BEHAVIOR which references the APPLICATION-SOFTWARE-COMPONENT-TYPE via COMPONENT-REF.

8.1.2.8.4. COMPLEX-DEVICE-DRIVER-SW-COMPONENT-TYPE

Source model entity: COMPLEX-DEVICE-DRIVER-COMPONENT-TYPE

Target model entity: COMPLEX-DEVICE-DRIVER-SW-COMPONENT-TYPE

Target model property	Source model property
PORTS/P-PORT-PROTOTYPE	PORTS/P-PORT-PROTOTYPE
PORTS/R-PORT-PROTOTYPE	PORTS/R-PORT-PROTOTYPE
PORT-GROUPS/PORT-GROUP	PORT-GROUPS/PORT-GROUP
No upgrade is currently implemented.	BSW-MODULE-DESCRIPTION-REFS/BSW-MODULE-DESCRIPTION-REF
No upgrade is currently implemented.	HARDWARE-ELEMENT-REFS/HARDWARE-ELEMENT-REF

New parameters in the target model

INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR

The parameter INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR contains the SWC-INTERNAL-BEHAVIOR which corresponds to the source model INTERNAL-BEHAVIOR which references the APPLICATION-SOFTWARE-COMPONENT-TYPE via COMPONENT-REF.

8.1.2.8.5. ECU-ABSTRACTION-SW-COMPONENT-TYPE

Source model entity: ECU-ABSTRACTION-COMPONENT-TYPE

Target model entity: ECU-ABSTRACTION-SW-COMPONENT-TYPE

Target model property	Source model property
PORTS/P-PORT-PROTOTYPE	PORTS/P-PORT-PROTOTYPE
PORTS/R-PORT-PROTOTYPE	PORTS/R-PORT-PROTOTYPE
PORT-GROUPS/PORT-GROUP	PORT-GROUPS/PORT-GROUP
No upgrade is currently implemented.	BSW-MODULE-DESCRIPTION-REFS/BSW-MODULE-DESCRIPTION-REF
No upgrade is currently implemented.	HARDWARE-ELEMENT-REFS/HARDWARE-ELEMENT-REF

New parameters in the target model

INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR

The parameter INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR contains the SWC-INTERNAL-BEHAVIOR which corresponds to the source model INTERNAL-BEHAVIOR which references the APPLICATION-SOFTWARE-COMPONENT-TYPE via COMPONENT-REF.

8.1.2.8.6. PARAMETER-SW-COMPONENT-TYPE

Source model entity: CALPRM-COMPONENT-TYPE

Target model entity: PARAMETER-SW-COMPONENT-TYPE

Target model property	Source model property
PORTS/P-PORT-PROTOTYPE	PORTS/P-PORT-PROTOTYPE
PORTS/R-PORT-PROTOTYPE	PORTS/R-PORT-PROTOTYPE
PORT-GROUPS/PORT-GROUP	PORT-GROUPS/PORT-GROUP
CONSTANT-MAPPING-REFS/CONSTANT-MAPPING-REF	No source model parameter is available.
DATA-TYPE-MAPPING-REFS/DATA-TYPE-MAPPING-REF	Refer to the section called “Creation of target model DATA-TYPE-MAPPING-REFS/DATA-TYPE-MAPPING-REF”.
INSTANTIATION-DATA-DEF-PROPS/INSTANTIATION-DATA-DEF-PROPS	No source model parameter is available.

New parameters in the target model

INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR

The parameter INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR contains the SWC-INTERNAL-BEHAVIOR which corresponds to the source model INTERNAL-BEHAVIOR which references the APPLICATION-SOFTWARE-COMPONENT-TYPE via COMPONENT-REF.

8.1.2.8.7. R-PORT-PROTOTYPE

R-PORT-PROTOTYPE is upgraded as described in [Section 8.1.1.8.7, “R-PORT-PROTOTYPE”](#).

8.1.2.8.8. P-PORT-PROTOTYPE

P-PORT-PROTOTYPE is upgraded as described in [Section 8.1.1.8.8, “P-PORT-PROTOTYPE”](#).

8.1.2.8.9. CLIENT-SERVER-INTERFACE

CLIENT-SERVER-INTERFACE is upgraded as described in [Section 8.1.1.8.9, “CLIENT-SERVER-INTERFACE”](#).

8.1.2.8.10. DELEGATION-SW-CONNECTOR

DELEGATION-SW-CONNECTOR is upgraded as described in [Section 8.1.1.8.11, “DELEGATION-SW-CONNECTOR”](#).

8.1.2.8.11. ASSEMBLY-SW-CONNECTOR

ASSEMBLY-SW-CONNECTOR is upgraded as described in [Section 8.1.1.8.10, “ASSEMBLY-SW-CONNECTOR”](#).

8.1.2.8.12. SERVICE-CONNECTOR-PROTOTYPE

AUTOSAR 3.2.x SERVICE-CONNECTOR-PROTOTYPE elements are not upgraded because no counterpart element exists in AUTOSAR 4.0. If such SENDER-RECEIVER-INTERFACE elements are encountered, a warning is issued.

8.1.2.8.13. PARAMETER-INTERFACE

PARAMETER-INTERFACE is upgraded as described in [Section 8.1.1.8.13, “PARAMETER-INTERFACE”](#).

8.1.2.8.14. MODE-SWITCH-INTERFACE

MODE-SWITCH-INTERFACE is upgraded as described in [Section 8.1.1.8.14, “MODE-SWITCH-INTERFACE”](#).

8.1.2.8.15. MODE-GROUP

Source model entity: MODE-GROUP

Target model entity: MODE-GROUP

Target model property	Source model property
TYPE-TREF	TYPE-TREF
SW-CALIBRATION-ACCESS	No source model parameter is available.

8.1.2.8.16. SW-COMPONENT-PROTOTYPE

SW-COMPONENT-PROTOTYPE is upgraded as described in [Section 8.1.1.8.16, “SW-COMPONENT-PROTOTYPE”](#).

8.1.2.8.17. VARIABLE-DATA-PROTOTYPE

VARIABLE-DATA-PROTOTYPE is upgraded as described in [Section 8.1.1.8.17, “VARIABLE-DATA-PROTOTYPE”](#).

8.1.2.8.18. ARGUMENT-DATA-PROTOTYPE

ARGUMENT-DATA-PROTOTYPE is upgraded as described in [Section 8.1.1.8.18, “ARGUMENT-DATA-PROTOTYPE”](#).

8.1.2.8.19. PARAMETER-DATA-PROTOTYPE

PARAMETER-DATA-PROTOTYPE is upgraded as described in [Section 8.1.1.8.19, “PARAMETER-DATA-PROTOTYPE”](#).

8.1.2.8.20. CLIENT-SERVER-OPERATION

CLIENT-SERVER-OPERATION is upgraded as described in [Section 8.1.1.8.20, “CLIENT-SERVER-OPERATION”](#).

8.1.2.8.21. APPLICATION-ERROR

APPLICATION-ERROR is upgraded as described in [Section 8.1.1.8.21, “APPLICATION-ERROR”](#).

8.1.2.8.22. NONQUEUED-RECEIVER-COM-SPEC

Source model entity: UNQUEUED-RECEIVER-COM-SPEC

Target model entity: NONQUEUED-RECEIVER-COM-SPEC

Target model property	Source model property
ALIVE-TIMEOUT	ALIVE-TIMEOUT
No target model parameter is available.	HANDLE-INVALID
INIT-VALUE/CONSTANT-REFERENCE	INIT-VALUE-REF
No target model parameter is available.	RESYNC-TIME
DATA-ELEMENT-REF	DATA-ELEMENT-REF
FILTER	FILTER
HANDLE-NEVER-RECEIVED	HANDLE-NEVER-RECEIVED if the source model parameter value is available, else false.
ENABLE-UPDATE	No source model parameter is available, the target model parameter is set to false.
HANDLE-OUT-OF-RANGE	No source model parameter is available, the target model parameter is set to NONE.
HANDLE-TIMEOUT-TYPE	No source model parameter is available, the target model parameter is set to NONE.
NETWORK-REPRESENTATION	No source model parameter is available.
USES-END-TO-END-PROTECTION	USES-END-TO-END-PROTECTION
EXTERNAL-REPLACEMENT-REF	No source model parameter is available.
MAX-DELTA-COUNTER-INIT	No source model parameter is available.

Processing of source model parameter HANDLE-INVALID

If the parameter HANDLE-INVALID is provided in the source model, a INVALIDATION-POLICY element is created in the related target model SENDER-RECEIVER-INTERFACE. The INVALIDATION-POLICY parameter HANDLE-INVALID is configured according to the source model UNQUEUED-RECEIVER-COM-SPEC parameter HANDLE-INVALID if all UNQUEUED-RECEIVER-COM-SPEC elements belonging to the source model SENDER-RECEIVER-INTERFACE and DATA-ELEMENT are providing the same HANDLE-INVALID value.

8.1.2.8.23. QUEUED-RECEIVER-COM-SPEC

Source model entity: QUEUED-RECEIVER-COM-SPEC

Target model entity: QUEUED-RECEIVER-COM-SPEC

Target model property	Source model property
DATA-ELEMENT-REF	DATA-ELEMENT-REF
No target model parameter is available.	FILTER
HANDLE-OUT-OF-RANGE	No source model parameter is available, the target model parameter is set to NONE.
HANDLE-OUT-OF-RANGE-STATUS	No source model parameter is available.
NETWORK-REPRESENTATION	No source model parameter is available.
QUEUE-LENGTH	QUEUE-LENGTH
USES-END-TO-END-PROTECTION	No source model parameter is available, the target model parameter is set to false.
MAX-DELTA-COUNTER-INIT	No source model parameter is available.

8.1.2.8.24. CLIENT-COM-SPEC

CLIENT-COM-SPEC is upgraded as described in [Section 8.1.1.8.24, “CLIENT-COM-SPEC”](#).

8.1.2.8.25. PARAMETER-REQUIRE-COM-SPEC

PARAMETER-REQUIRE-COM-SPEC is upgraded as described in [Section 8.1.1.8.25, “PARAMETER-REQUIRE-COM-SPEC”](#).

8.1.2.8.26. QUEUED-SENDER-COM-SPEC

Source model entity: QUEUED-SENDER-COM-SPEC

Target model entity: QUEUED-SENDER-COM-SPEC

Target model property	Source model property
DATA-ELEMENT-REF	DATA-ELEMENT-REF
TRANSMISSION-ACKNOWLEDGE	TRANSMISSION-ACKNOWLEDGE
HANDLE-OUT-OF-RANGE	No source model parameter is available, the target model parameter is set to NONE.

Target model property	Source model property
NETWORK-REPRESENTATION	No source model parameter is available.
USES-END-TO-END-PROTECTION	USES-END-TO-END-PROTECTION

8.1.2.8.27. NONQUEUED-SENDER-COM-SPEC

Source model entity: UNQUEUED-SENDER-COM-SPEC

Target model entity: NONQUEUED-SENDER-COM-SPEC

Target model property	Source model property
DATA-ELEMENT-REF	DATA-ELEMENT-REF
TRANSMISSION-ACKNOWLEDGE	TRANSMISSION-ACKNOWLEDGE
No target model parameter is available.	CAN-INVALIDATE
INIT-VALUE/CONSTANT-REFERENCE	INIT-VALUE-REF
HANDLE-OUT-OF-RANGE	No source model parameter is available, the target model parameter is set to NONE.
NETWORK-REPRESENTATION	No source model parameter is available.
USES-END-TO-END-PROTECTION	USES-END-TO-END-PROTECTION

8.1.2.8.28. SERVER-COM-SPEC

SERVER-COM-SPEC is upgraded as described in [Section 8.1.1.8.28, “SERVER-COM-SPEC”](#).

8.1.2.8.29. MODE-SWITCH-SENDER-COM-SPEC

Source model entity: MODE-SWITCH-COM-SPEC

Target model entity: MODE-SWITCH-SENDER-COM-SPEC

Target model property	Source model property
MODE-GROUP-REF	MODE-GROUP-REF
MODE-SWITCHED-ACK	MODE-SWITCHED-ACK
QUEUE-LENGTH	QUEUE-LENGTH
ENHANCED-MODE-API	No source model parameter is available.

8.1.2.8.30. PARAMETER-PROVIDE-COM-SPEC

PARAMETER-PROVIDE-COM-SPEC is upgraded as described in [Section 8.1.1.8.30, “PARAMETER-PROVIDE-COM-SPEC”](#).

8.1.2.8.31. TRANSMISSION-ACKNOWLEDGE

TRANSMISSION-ACKNOWLEDGE is upgraded as described in [Section 8.1.1.8.31, “TRANSMISSION-ACKNOWLEDGE”](#).

8.1.2.8.32. DATA-FILTER

Source model entities: SIGNAL-PORT/DATA-FILTER and TRANSMISSION-MODE-CONDITION/DATA-FILTER

Target model entity: DATA-FILTER

Target model property	Source model property
MASK	MASK
MAX	MAX
MIN	MIN
OFFSET	OFFSET
PERIOD	PERIOD
X	X

Configuration of target model parameter DATA-FILTER-TYPE

DATA-FILTER-TYPE is configured depending on the source model element below SIGNAL-PORT/DATA-FILTER and TRANSMISSION-MODE-CONDITION/DATA-FILTER. The following values are converted:

- ▶ ALWAYS
- ▶ MASKED-NEW-DIFFERS-MASKED-OLD
- ▶ MASKED-NEW-DIFFERS-X
- ▶ MASKED-NEW-EQUALS-X
- ▶ NEVER
- ▶ NEW-IS-OUTSIDE

- ▶ NEW-IS-WITHIN
- ▶ ONE-EVERY-N

8.1.2.8.33. VARIABLE-ACCESS

VARIABLE-ACCESS is upgraded as described in [Section 8.1.1.8.33, “VARIABLE-ACCESS”](#).

8.1.2.8.34. PARAMETER-ACCESS

PARAMETER-ACCESS is upgraded as described in [Section 8.1.1.8.34, “PARAMETER-ACCESS”](#).

8.1.2.8.35. RUNNABLE-ENTITY

Source model entity: RUNNABLE-ENTITY

Target model entity: RUNNABLE-ENTITY

Target model property	Source model property
ARGUMENTS/RUNNABLE-ENTITY-ARGUMENT	No source model parameter is available.
Currently no upgrade is available.	BSW-ENTITY-REF
PARAMETER-ACCESSS/PARAMETER-ACCESS/ACCESSED-PARAMETER	CALPRM-ACCESSS/CALPRM-ACCESS
CAN-BE-INVOKED-CONCURRENTLY	CAN-BE-INVOKED-CONCURRENTLY
CAN-ENTER-EXCLUSIVE-AREA-REFS/CAN-ENTER-EXCLUSIVE-AREA-REF	CAN-ENTER-EXCLUSIVE-AREA-REFS/CAN-ENTER-EXCLUSIVE-AREA-REF
DATA-READ-ACCESSS/VARIABLE-ACCESS	DATA-READ-ACCESSS/DATA-READ-ACCESS
DATA-RECEIVE-POINT-BY-ARGUMENTS/VARIABLE-ACCESS	DATA-RECEIVE-POINTS/DATA-RECEIVE-POINT
DATA-SEND-POINTS/VARIABLE-ACCESS	DATA-SEND-POINTS/DATA-SEND-POINT
DATA-WRITE-ACCESSS/VARIABLE-ACCESS	DATA-WRITE-ACCESSS/DATA-WRITE-ACCESS
MINIMUM-START-INTERVAL	MINIMUM-START-INTERVAL
MODE-SWITCH-POINTS/MODE-SWITCH-POINT	MODE-SWITCH-POINTS/MODE-SWITCH-POINT
PARAMETER-ACCESSS/PARAMETER-ACCESS/ACCESSED-PARAMETER/LOCAL-PARAMETER-REF	PER-INSTANCE-CALPRM-ACCESS-REFS/PER-INSTANCE-CALPRM-ACCESS-REF
RUNS-INSIDE-EXCLUSIVE-AREA-REFS/RUNS-INSIDE-EXCLUSIVE-AREA-REF	RUNS-INSIDE-EXCLUSIVE-AREA-REFS/RUNS-INSIDE-EXCLUSIVE-AREA-REF

Target model property	Source model property
SERVER-CALL-POINTS/ASYNCHRONOUS-SERVER-CALL-POINT	SERVER-CALL-POINTS/ASYNCHRONOUS-SERVER-CALL-POINT
SERVER-CALL-POINTS/SYNCHRONOUS-SERVER-CALL-POINT	SERVER-CALL-POINTS/SYNCHRONOUS-SERVER-CALL-POINT
PARAMETER-ACCESSSS/PARAMETER-ACCESS/ACCESSED-PARAMETER/LOCAL-PARAMETER-REF	SHARED-CALPRM-ACCESS-REFS/SHARED-CALPRM-ACCESS-REF
SYMBOL	SYMBOL
WAIT-POINTS/WAIT-POINT	WAIT-POINTS/WAIT-POINT
WRITTEN-LOCAL-VARIABLES/VARIABLE-ACCESS/ACCESSED-VARIABLE/LOCAL-VARIABLE-REF	WRITTEN-VARIABLE-REFS/WRITTEN-VARIABLE-REF
READ-LOCAL-VARIABLES/VARIABLE-ACCESS/ACCESSED-VARIABLE/LOCAL-VARIABLE-REF	READ-VARIABLE-REFS/READ-VARIABLE-REF

Migration of source model DATA-RECEIVE-POINT elements

Please note that all source model DATA-RECEIVE-POINT elements are migrated to DATA-RECEIVE-POINT-BY-ARGUMENTS/VARIABLE-ACCESS elements in the target model, i.e. there is no conversion to the target model entities DATA-RECEIVE-POINT-BY-VALUES/VARIABLE-ACCESS.

Creation of target model ASYNCHRONOUS-SERVER-CALL-RESULT-POINT elements

Please note that for every source model ASYNCHRONOUS-SERVER-CALL-RETURNS-EVENT entity which either references a RUNNABLE-ENTITY via START-ON-EVENT-REF or references a SERVER-CALL-POINT of the RUNNABLE-ENTITY via EVENT-SOURCE-REF, one ASYNCHRONOUS-SERVER-CALL-RESULT-POINTS/ASYNCHRONOUS-SERVER-CALL-RESULT-POINT is created in the target model.

8.1.2.8.36. WAIT-POINT

WAIT-POINT is upgraded as described in [Section 8.1.1.8.36, “WAIT-POINT”](#).

8.1.2.8.37. SWC-INTERNAL-BEHAVIOR

SWC-INTERNAL-BEHAVIOR is upgraded as described in [Section 8.1.1.8.37, “SWC-INTERNAL-BEHAVIOR”](#).

8.1.2.8.38. PORT-API-OPTION

PORT-API-OPTION is upgraded as described in [Section 8.1.1.8.38, “PORT-API-OPTION”](#).

8.1.2.8.39. DATA-SEND-COMPLETED-EVENT

DATA-SEND-COMPLETED-EVENT is upgraded as described in [Section 8.1.1.8.39, “DATA-SEND-COMPLETED-EVENT”](#).

8.1.2.8.40. DATA-RECEIVED-EVENT

DATA-RECEIVED-EVENT is upgraded as described in [Section 8.1.1.8.40, “DATA-RECEIVED-EVENT”](#).

8.1.2.8.41. DATA-RECEIVE-ERROR-EVENT

DATA-RECEIVE-ERROR-EVENT is upgraded as described in [Section 8.1.1.8.41, “DATA-RECEIVE-ERROR-EVENT”](#).

8.1.2.8.42. ASYNCHRONOUS-SERVER-CALL-RETURNS-EVENT

ASYNCHRONOUS-SERVER-CALL-RETURNS-EVENT is upgraded as described in [Section 8.1.1.8.42, “ASYNCHRONOUS-SERVER-CALL-RETURNS-EVENT”](#).

8.1.2.8.43. TIMING-EVENT

TIMING-EVENT is upgraded as described in [Section 8.1.1.8.43, “TIMING-EVENT”](#).

8.1.2.8.44. OPERATION-INVOKED-EVENT

OPERATION-INVOKED-EVENT is upgraded as described in [Section 8.1.1.8.44, “OPERATION-INVOKED-EVENT”](#).

8.1.2.8.45. SWC-MODE-SWITCH-EVENT

SWC-MODE-SWITCH-EVENT is upgraded as described in [Section 8.1.1.8.45, “SWC-MODE-SWITCH-EVENT”](#).

8.1.2.8.46. MODE-SWITCHED-ACK-EVENT

MODE-SWITCHED-ACK-EVENT is upgraded as described in [Section 8.1.1.8.46, “MODE-SWITCHED-ACK-EVENT”](#).

8.1.2.8.47. ASYNCHRONOUS-SERVER-CALL-POINT

ASYNCHRONOUS-SERVER-CALL-POINT is upgraded as described in [Section 8.1.1.8.47, “ASYNCHRONOUS-SERVER-CALL-POINT”](#).

8.1.2.8.48. SYNCHRONOUS-SERVER-CALL-POINT

SYNCHRONOUS-SERVER-CALL-POINT is upgraded as described in [Section 8.1.1.8.48, “SYNCHRONOUS-SERVER-CALL-POINT”](#).

8.1.2.8.49. NUMERICAL-VALUE-SPECIFICATION

NUMERICAL-VALUE-SPECIFICATION is upgraded as described in [Section 8.1.1.8.49, “NUMERICAL-VALUE-SPECIFICATION”](#).

8.1.2.8.50. TEXT-VALUE-SPECIFICATION

TEXT-VALUE-SPECIFICATION is upgraded as described in [Section 8.1.1.8.50, “TEXT-VALUE-SPECIFICATION”](#).

8.1.2.8.51. ARRAY-VALUE-SPECIFICATION

ARRAY-VALUE-SPECIFICATION is upgraded as described in [Section 8.1.1.8.51, “ARRAY-VALUE-SPECIFICATION”](#).

8.1.2.8.52. RECORD-VALUE-SPECIFICATION

RECORD-VALUE-SPECIFICATION is upgraded as described in [Section 8.1.1.8.52, “RECORD-VALUE-SPECIFICATION”](#).

8.1.2.8.53. CONSTANT-SPECIFICATION

CONSTANT-SPECIFICATION is upgraded as described in [Section 8.1.1.8.53, “CONSTANT-SPECIFICATION”](#).

8.1.2.8.54. CONSTANT-REFERENCE

CONSTANT-REFERENCE is upgraded as described in [Section 8.1.1.8.54, “CONSTANT-REFERENCE”](#).

8.1.2.8.55. UNIT

Source model entity: UNIT

Target model entity: UNIT

Target model property	Source model property
FACTOR-SI-TO-UNIT	FACTOR-SI-TO-UNIT
OFFSET-SI-TO-UNIT	OFFSET-SI-TO-UNIT
PHYSICAL-DIMENSION-REF	PHYSICAL-DIMENSION-REF

No upgrade of source model DISPLAY-NAME

Due to internal data model restrictions, the source model parameter DISPLAY-NAME is not upgraded to the target model.

8.1.2.8.56. UNIT-GROUP

Due to internal data model restrictions, there is no upgrade path for UNIT-GROUP elements.

8.1.2.8.57. PHYSICAL-DIMENSION

Source model entity: PHYSICAL-DIMENSION

Target model entity: PHYSICAL-DIMENSION

Target model property	Source model property
CURRENT-EXP	CURRENT-EXP
LENGTH-EXP	LENGTH-EXP
LUMINOUS-INTENSITY-EXP	LUMINOUS-INTENSITY-EXP
MASS-EXP	MASS-EXP
MOLAR-AMOUNT-EXP	MOLAR-AMOUNT-EXP
TEMPERATURE-EXP	TEMPERATURE-EXP
TIME-EXP	TIME-EXP

8.1.2.8.58. SW-BASE-TYPE

Source model entity: SW-BASE-TYPE

Target model entity: SW-BASE-TYPE

Target model property	Source model property
MEM-ALIGNMENT	MEM-ALIGNMENT
BASE-TYPE-SIZE	BASE-TYPE-SIZE
MAX-BASE-TYPE-SIZE	MAX-BASE-TYPE-SIZE

Upgrade of source model BASE-TYPE-ENCODING.

If the source model BASE-TYPE-ENCODING contains one of the values

- ▶ 1C
- ▶ 2C
- ▶ BCD-P
- ▶ BCD-UP
- ▶ DSP-FRACTIONAL
- ▶ SM
- ▶ IEEE754
- ▶ ISO-8859-1
- ▶ ISO-8859-2
- ▶ WINDOWS-1252
- ▶ UTF-8
- ▶ UCS-2
- ▶ NONE

the value is directly copied into the target model parameter BASE-TYPE-ENCODING. For any other value in the source model, NONE will be written into the target model parameter.

Upgrade of source model CATEGORY

If the source model CATEGORY value is either SW_FIXED_LENGTH or FIXED_LENGTH, the target model CATEGORY parameter will contain the value FIXED_LENGTH. If the source model parameter value contains SW_VARIABLE_LENGTH or VARIABLE_LENGTH, the target model CATEGORY parameter value will contain VARIABLE_LENGTH.

No upgrade of source model **BASE-TYPE-REF**

Due to internal data model restrictions, the source model parameter **BASE-TYPE-REF** is not upgraded into the target model.

No upgrade of source model **BYTE-ORDER**

The source model parameter **BYTE-ORDER** is not upgraded into the target model because **BYTE-ORDER** is not a property of a (base) data type.

8.1.2.8.59. SW-DATA-DEF-PROPS

Source model entity: **SW-DATA-DEF-PROPS**

Target model entity: **SW-DATA-DEF-PROPS**

Target model property	Source model property
SW-DATA-DEF-PROPS-CONDITIONAL/BASE-TYPE-REF	BASE-TYPE-REF
SW-DATA-DEF-PROPS-CONDITIONAL/COMPU-METHOD-REF	COMPU-METHOD-REF
SW-DATA-DEF-PROPS-CONDITIONAL/INVALID-VALUE	INVALID-VALUE
SW-DATA-DEF-PROPS-CONDITIONAL/SW-CALIBRATION-ACCESS	SW-CALIBRATION-ACCESS
SW-DATA-DEF-PROPS-CONDITIONAL/SW-IMPL-POLICY	SW-IMPL-POLICY
SW-DATA-DEF-PROPS-CONDITIONAL/UNIT-REF	UNIT-REF
SW-DATA-DEF-PROPS-CONDITIONAL/INVALID-VALUE	INVALID-VALUE
SW-DATA-DEF-PROPS-CONDITIONAL/SW-VALUE-BLOCK-SIZE	SW-VALUE-BLOCK-SIZE/V
SW-DATA-DEF-PROPS-CONDITIONAL/DISPLAY-FORMAT	DISPLAY-FORMAT
SW-DATA-DEF-PROPS-CONDITIONAL/DATA-CONSTR-REF	DATA-CONSTR-REF
SW-DATA-DEF-PROPS-CONDITIONAL/SW-POINTER-TARGET-PROPS	SW-POINTER

No upgrade of source model parameters `SW-CALPRM-AXIS-SET`, `SW-CLASS-ATTR-IMPL-REF`, `SW-CLASS-REF`, `SW-CODE-SYNTAX-REF`, `SW-DATA-DEPENDENCY`, `SW-HOST-VARIABLE`, `SW-POINTER`, `SW-RECORD-LAYOUT-REF`, `SW-TEXT-PROPS`, `SW-VARIABLE-ACCESS-IMPL-POLICY`

The source model parameters `SW-CALPRM-AXIS-SET`, `SW-CLASS-ATTR-IMPL-REF`, `SW-CLASS-REF`, `SW-CODE-SYNTAX-REF`, `SW-DATA-DEPENDENCY`, `SW-HOST-VARIABLE`, `SW-POINTER`, `SW-RECORD-LAYOUT-REF`, `SW-TEXT-PROPS`, `SW-VARIABLE-ACCESS-IMPL-POLICY` are not upgraded to the target model.

No configuration of target model parameters `SW-ALIGNMENT`, `SW-COMPARISON-VARIABLES`, `SW-DATA-DEPENDENCY`, `SW-HOST-VARIABLE`, `SW-INTENDED-RESOLUTION`, `SW-INTERPOLATION-METHOD`, `SW-IS-VIRTUAL`, `SW-REFRESH-TIMING`, `SW-TEXT-PROPS`, `VALUE-AXIS-DATA-TYPE-REF`

The target model parameters `SW-ALIGNMENT`, `SW-COMPARISON-VARIABLES`, `SW-DATA-DEPENDENCY`, `SW-HOST-VARIABLE`, `SW-INTENDED-RESOLUTION`, `SW-INTERPOLATION-METHOD`, `SW-IS-VIRTUAL`, `SW-REFRESH-TIMING`, `SW-TEXT-PROPS`, `VALUE-AXIS-DATA-TYPE-REF` are not configured during the model upgrade because there is no source model counterpart.

Configuration of `SW-DATA-DEF-PROPS-CONDITIONAL`/`SW-IMPL-POLICY`

If the source model `SW-DATA-DEF-PROPS` element is aggregated within a `DATA-ELEMENT-PROTOTYPE` which has its `IS-QUEUED` value set to `true`, the upgraded `SW-DATA-DEF-PROPS-CONDITIONAL`/`SW-IMPL-POLICY` contains the value `QUEUED`, else it contains the `SW-IMPL-POLICY` value of the source model `SW-DATA-DEF-PROPS` counterpart. If the source model `SW-DATA-DEF-PROPS` value is `MESSAGE`, the parameter is not configured at all in the target model since it does not support this value.

Creation of `SW-DATA-DEF-PROPS` elements in the target model

For every target model `ARGUMENT-DATA-PROTOTYPE`, `PARAMETER-DATA-PROTOTYPE`, or `VARIABLE-DATA-PROTOTYPE` which has a source model counterpart which has its `IS-QUEUED` parameter value set to `true` but does not aggregate an `SW-DATA-DEF-PROPS` element, a dedicated `SW-DATA-DEF-PROPS` is created in the target model which has its `SW-IMPL-POLICY` value set to `QUEUED`.

8.1.2.8.60. DATA-CONSTR

Source model entity: `DATA-CONSTR`

Target model entity: `DATA-CONSTR`

Target model property	Source model property
DATA-CONSTR-RULES/DATA-CONSTR-RULE	DATA-CONSTR-RULES/DATA-CONSTR-RULE
SHORT-NAME-PATTERN	No source model parameter is available.

8.1.2.8.61. DATA-CONSTR-RULE

Source model entity: DATA-CONSTR-RULE

Target model entity: DATA-CONSTR-RULE

Target model property	Source model property
INTERNAL-CONSTRS	INTERNAL-CONSTRS
PHYS-CONSTRS	PHYS-CONSTRS

8.1.2.8.62. PHYS-CONSTRS

Source model entity: PHYS-CONSTRS

Target model entity: PHYS-CONSTRS

Target model property	Source model property
LOWER-LIMIT	LOWER-LIMIT
UPPER-LIMIT	UPPER-LIMIT
UNIT-REF	UNIT-REF

No configuration of target model parameters MAX-DIFF, MAX-GRADIENT, MONOTONY, SCALE-CONSTRS

The target model parameters MAX-DIFF, MAX-GRADIENT, MONOTONY, SCALE-CONSTRS are not configured during the model upgrade because there is no source model counterpart.

Configuration of target model parameter UNIT-REF

If any of the source model PHYS-CONSTRS has no valid UNIT-REF defined, a dedicated UNIT is created in the target model, called NoDimensionUnit. A target model PHYS-CONSTRS is referencing this UNIT via UNIT-REF if its source model counterpart does not provide UNIT-REF.

8.1.2.8.63. INTERNAL-CONSTRS

Source model entity: INTERNAL-CONSTRS

Target model entity: INTERNAL-CONSTRS

Target model property	Source model property
LOWER-LIMIT	LOWER-LIMIT
UPPER-LIMIT	UPPER-LIMIT

No configuration of target model parameters MAX-DIFF, MAX-GRADIENT, MONOTONY, SCALE-CONSTRS

The target model parameters MAX-DIFF, MAX-GRADIENT, MONOTONY, SCALE-CONSTRS are not configured during the model upgrade because there is no source model counterpart.

8.1.2.8.64. IMPLEMENTATION-DATA-TYPE

IMPLEMENTATION-DATA-TYPE is upgraded as described in [Section 8.1.1.8.59, “IMPLEMENTATION-DATA-TYPE”](#).

8.1.2.8.65. COMPU-METHOD

COMPU-METHOD is upgraded as described in [Section 8.1.1.8.60, “COMPU-METHOD”](#).

8.1.2.8.66. COMPU-SCALE

Source model entity: COMPU-SCALE

Target model entity: COMPU-SCALE

Target model property	Source model property
LOWER-LIMIT	LOWER-LIMIT
UPPER-LIMIT	UPPER-LIMIT
COMPU-CONST	COMPU-CONST
COMPU-INVERSE-VALUE	No source model parameter is available.
COMPU-RATIONAL-COEFFS/COMPU-NUMERATOR	COMPU-RATIONAL-COEFFS/COMPU-NUMERATOR
COMPU-RATIONAL-COEFFS/COMPU-DENOMINATOR	COMPU-RATIONAL-COEFFS/COMPU-DENOMINATOR

Target model property	Source model property
MASK	No source model parameter is available.
SHORT-LABEL	SHORT-LABEL
SYMBOL	SYMBOL

8.1.2.8.67. EXCLUSIVE-AREA

EXCLUSIVE-AREA is upgraded as described in [Section 8.1.1.8.62, “EXCLUSIVE-AREA”](#).

8.1.2.8.68. COM-MGR-USER-NEEDS

COM-MGR-USER-NEEDS is upgraded as described in [Section 8.1.1.8.63, “COM-MGR-USER-NEEDS”](#).

8.1.2.8.69. DIAGNOSTIC-COMMUNICATION-MANAGER-NEEDS

DIAGNOSTIC-COMMUNICATION-MANAGER-NEEDS is upgraded as described in [Section 8.1.1.8.64, “DIAGNOSTIC-COMMUNICATION-MANAGER-NEEDS”](#).

8.1.2.8.70. DIAGNOSTIC-EVENT-NEEDS

DIAGNOSTIC-EVENT-NEEDS is upgraded as described in [Section 8.1.1.8.65, “DIAGNOSTIC-EVENT-NEEDS”](#).

8.1.2.8.71. ECU-STATE-MGR-USER-NEEDS

ECU-STATE-MGR-USER-NEEDS is upgraded as described in [Section 8.1.1.8.66, “ECU-STATE-MGR-USER-NEEDS”](#).

8.1.2.8.72. FUNCTION-INHIBITION-NEEDS

FUNCTION-INHIBITION-NEEDS is upgraded as described in [Section 8.1.1.8.67, “FUNCTION-INHIBITION-NEEDS”](#).

8.1.2.8.73. NV-BLOCK-NEEDS

NV-BLOCK-NEEDS is upgraded as described in [Section 8.1.1.8.68, “NV-BLOCK-NEEDS”](#).

8.1.2.8.74. OBD-CONTROL-SERVICE-NEEDS

OBD-CONTROL-SERVICE-NEEDS is upgraded as described in [Section 8.1.1.8.69, “OBD-CONTROL-SERVICE-NEEDS”](#).

8.1.2.8.75. OBD-PID-SERVICE-NEEDS

OBD-PID-SERVICE-NEEDS is upgraded as described in [Section 8.1.1.8.71, “OBD-PID-SERVICE-NEEDS”](#).

8.1.2.8.76. OBD-RATIO-SERVICE-NEEDS

OBD-RATIO-SERVICE-NEEDS is upgraded as described in [Section 8.1.1.8.72, “OBD-RATIO-SERVICE-NEEDS”](#).

8.1.2.8.77. SUPERVISED-ENTITY-NEEDS

SUPERVISED-ENTITY-NEEDS is upgraded as described in [Section 8.1.1.8.73, “SUPERVISED-ENTITY-NEEDS”](#).

8.1.2.8.78. PER-INSTANCE-MEMORY

Source model entity: PER-INSTANCE-MEMORY

Target model entity: PER-INSTANCE-MEMORY

Target model property	Source model property
TYPE	TYPE
TYPE-DEFINITION	TYPE-DEFINITION

Configuration of target model parameter INIT-VALUE

The target model parameter INIT-VALUE is always set to 0 because there is no corresponding source model value.

Creation of target model parameter SW-DATA-DEF-PROPS

For each PER-INSTANCE-MEMORY element created in the target model, one SW-DATA-DEF-PROPS sub element is set up.

8.1.2.8.79. PER-INSTANCE-MEMORY-SIZE

PER-INSTANCE-MEMORY-SIZE is upgraded as described in [Section 8.1.1.8.75, “PER-INSTANCE-MEMORY-SIZE”](#).

8.1.2.8.80. SWC-IMPLEMENTATION

SWC-IMPLEMENTATION is upgraded as described in [Section 8.1.1.8.76, “SWC-IMPLEMENTATION”](#), except for the SHORT-NAME of the created RESOURCE-CONSUMPTION element, see [the section called “Creation of target model RESOURCE-CONSUMPTION”](#).

8.1.2.8.81. CODE

CODE is upgraded as described in [Section 8.1.1.8.77, “CODE”](#).

8.1.2.8.82. MODE-DECLARATION

MODE-DECLARATION is upgraded as described in [Section 8.1.1.8.78, “MODE-DECLARATION”](#).

8.1.2.8.83. MODE-DECLARATION-GROUP

MODE-DECLARATION-GROUP is upgraded as described in [Section 8.1.1.8.79, “MODE-DECLARATION-GROUP”](#).

8.1.2.8.84. DISABLED-MODE-IREF

DISABLED-MODE-IREF is upgraded as described in [Section 8.1.1.8.80, “DISABLED-MODE-IREF”](#).

8.1.2.8.85. MODE-SWITCHED-ACK

MODE-SWITCHED-ACK is upgraded as described in [Section 8.1.1.8.81, “MODE-SWITCHED-ACK”](#).

8.1.2.8.86. MODE-ACCESS-POINT

MODE-ACCESS-POINT is upgraded as described in [Section 8.1.1.8.82, “MODE-ACCESS-POINT”](#).

8.1.2.8.87. MODE-SWITCH-POINT

MODE-SWITCH-POINT is upgraded as described in [Section 8.1.1.8.83, “MODE-SWITCH-POINT”](#).

8.1.2.8.88. SW-ADDR-METHOD

SW-ADDR-METHOD is upgraded as described in [Section 8.1.1.8.84, “SW-ADDR-METHOD”](#).

8.1.2.8.89. MEMORY-SECTION

MEMORY-SECTION elements are not required by the RTE and are therefore not upgraded into the target model.

8.1.2.8.90. SYSTEM-MAPPING

Source model entity: MAPPING

Target model entity: SYSTEM-MAPPING

Target model property	Source model property
DATA-MAPPINGS/CLIENT-SERVER-TO-SIGNAL-GROUP-MAPPING	DATA-MAPPINGS/CLIENT-SERVER-TO-SIGNAL-GROUP-MAPPING
DATA-MAPPINGS/SENDER-RECEIVER-TO-SIGNAL-GROUP-MAPPING	DATA-MAPPINGS/SENDER-RECEIVER-TO-SIGNAL-GROUP-MAPPING
DATA-MAPPINGS/SENDER-RECEIVER-TO-SIGNAL-MAPPING	DATA-MAPPINGS/SENDER-RECEIVER-TO-SIGNAL-MAPPING
PNC-MAPPINGS/PNC-MAPPING	PNC-MAPPINGS/PNC-MAPPING
SW-MAPPINGS/SWC-TO-ECU-MAPPING	SW-MAPPINGS/SWC-TO-ECU-MAPPING
SW-IMPL-MAPPINGS/SWC-TO-IMPL-MAPPING	SW-IMPL-MAPPINGS/SWC-TO-IMPL-MAPPING

No upgrade of source model parameters ECU-RESOURCE-MAPPINGS, MAPPING-CONSTRAINTS, RESOURCE-ESTIMATIONS and SIGNAL-PATH-CONSTRAINTS

Due to internal data model restrictions, the source model parameters ECU-RESOURCE-MAPPINGS, MAPPING-CONSTRAINTS, RESOURCE-ESTIMATIONS and SIGNAL-PATH-CONSTRAINTS are not upgraded to their target model counterparts.

8.1.2.8.91. SWC-TO-ECU-MAPPING

SWC-TO-ECU-MAPPING is upgraded as described in [Section 8.1.1.8.87, “SWC-TO-ECU-MAPPING”](#).

8.1.2.8.92. SWC-TO-IMPL-MAPPING

SWC-TO-IMPL-MAPPING is upgraded as described in [Section 8.1.1.8.88, “SWC-TO-IMPL-MAPPING”](#).

8.1.2.8.93. SENDER-RECEIVER-TO-SIGNAL-GROUP-MAPPING

SENDER-RECEIVER-TO-SIGNAL-GROUP-MAPPING is upgraded as described in [Section 8.1.1.8.89, “SENDER-RECEIVER-TO-SIGNAL-GROUP-MAPPING”](#).

8.1.2.8.94. SENDER-RECEIVER-TO-SIGNAL-MAPPING

SENDER-RECEIVER-TO-SIGNAL-MAPPING is upgraded as described in [Section 8.1.1.8.90, “SENDER-RECEIVER-TO-SIGNAL-MAPPING”](#).

8.1.2.8.95. CLIENT-SERVER-TO-SIGNAL-GROUP-MAPPING

CLIENT-SERVER-TO-SIGNAL-GROUP-MAPPING is upgraded as described in [Section 8.1.1.8.91, “CLIENT-SERVER-TO-SIGNAL-GROUP-MAPPING”](#).

8.1.2.8.96. SENDER-REC-RECORD-TYPE-MAPPING

SENDER-REC-RECORD-TYPE-MAPPING is upgraded as described in [Section 8.1.1.8.92, “SENDER-REC-RECORD-TYPE-MAPPING”](#).

8.1.2.8.97. SENDER-REC-RECORD-ELEMENT-MAPPING

SENDER-REC-RECORD-ELEMENT-MAPPING is upgraded as described in [Section 8.1.1.8.93, “SENDER-REC-RECORD-ELEMENT-MAPPING”](#).

8.1.2.8.98. SENDER-REC-ARRAY-TYPE-MAPPING

SENDER-REC-ARRAY-TYPE-MAPPING is upgraded as described in [Section 8.1.1.8.94, “SENDER-REC-ARRAY-TYPE-MAPPING”](#).

8.1.2.8.99. SENDER-REC-ARRAY-ELEMENT-MAPPING

SENDER-REC-ARRAY-ELEMENT-MAPPING is upgraded as described in [Section 8.1.1.8.95, “SENDER-REC-ARRAY-ELEMENT-MAPPING”](#).

8.1.2.8.100. PNC-MAPPING

Source model entity: PNC-MAPPING

Target model entity: PNC-MAPPING

Target model property	Source model property
PNC-GROUP-REFS / PNC-GROUP-REF	PNC-GROUP-REFS / PNC-GROUP-REF
PNC-IDENTIFIER	PNC-IDENTIFIER
VFC-IREFS / VFC-IREF	VFC-IREFS / VFC-IREF
WAKEUP-FRAME-REFS / WAKEUP-FRAME-REF	WAKEUP-FRAME-REFS / WAKEUP-FRAME-REF

8.1.2.8.101. END-TO-END-PROTECTION-SET

END-TO-END-PROTECTION-SET is upgraded as described in [Section 8.1.1.8.96, “END-TO-END-PROTECTION-SET”](#).

8.1.2.8.102. END-TO-END-PROTECTION

END-TO-END-PROTECTION is upgraded as described in [Section 8.1.1.8.97, “END-TO-END-PROTECTION”](#).

8.1.2.8.103. END-TO-END-PROTECTION-I-SIGNAL-I-PDU

END-TO-END-PROTECTION-I-SIGNAL-I-PDU is upgraded as described in [Section 8.1.1.8.98, “END-TO-END-PROTECTION-I-SIGNAL-I-PDU”](#).

8.1.2.8.104. END-TO-END-PROTECTION-VARIABLE-PROTOTYPE

END-TO-END-PROTECTION-VARIABLE-PROTOTYPE is upgraded as described in [Section 8.1.1.8.99, “END-TO-END-PROTECTION-VARIABLE-PROTOTYPE”](#).

8.1.3. Upgrading a system model from AUTOSAR 3.2.2 to AUTOSAR 4.0.3

This section describes how an AUTOSAR 3.2.2 system model is upgraded to AUTOSAR 4.0.3.

Since the AUTOSAR 3.2.1 and 3.2.2 source models are similar, this chapter only describes those parts of the upgrade that differ from the 3.2.1 to 4.0.3 upgrade. For all other parts, please refer to [Section 8.1.2, “Upgrading a system model from AUTOSAR 3.2.x to AUTOSAR 4.0.3”](#).

If not stated explicitly, all entities in the 4.0.3 target model are located in the same package as they were in the 3.2.2 source model.

If not specified explicitly otherwise, the parameters `SHORT-NAME`, `CATEGORY`, and `UUID` of Identifiable entities are upgraded directly into their counterparts in the target model.

8.1.3.1. Topology

8.1.3.1.1. CAN-CLUSTER

Source model entity: CAN-CLUSTER

Target model entity: CAN-CLUSTER

Target model property	Source model property
CAN-CLUSTER-CONDITIONAL/PHYSICAL-CHANNELS	PHYSICAL-CHANNELS
CAN-CLUSTER-CONDITIONAL/PROTOCOL-NAME	PROTOCOL-NAME
CAN-CLUSTER-CONDITIONAL/PROTOCOL-VERSION	PROTOCOL-VERSION
CAN-CLUSTER-CONDITIONAL/SPEED	SPEED/1000
No target model parameter is available.	MAX-FRAME-LENGTH
No target model parameter is available.	NM-REPEAT-MESSAGE-SUPPORT
No target model parameter is available.	NM-COORDINATOR-SYNC-PRIO

Changed or moved parameters

`NM-NODE-DETECTION-ENABLED` and `NM-NODE-ID-ENABLED`

The source model parameters `NM-NODE-DETECTION-ENABLED` and `NM-NODE-ID-ENABLED` are configured in the target model entity `NM-ECU`, see [Section 8.1.2.7.2, “NM-ECU”](#).

`SPEED`

In the target model, the parameter `SPEED` is given in kbit/s, whereas the source model provides it in bit/s. If the source model parameter is not divisible by 1000 without rest, it cannot be upgraded at all.

`NM-BUS-LOAD-REDUCTION-ACTIVE`, `NM-WAIT-BUS-SLEEP-TIME`, `NM-IMMEDIATE-NM-CYCLE-TIME`, `NM-IMMEDIATE-NM-TRANSMISSIONS`, `NM-MSG-CYCLE-TIME`, and `NM-REMOTE-SLEEP-INDICATION-TIME`

The parameters `NM-BUS-LOAD-REDUCTION-ACTIVE`, `NM-WAIT-BUS-SLEEP-TIME`, `NM-IMMEDIATE-NM-CYCLE-TIME`, `NM-IMMEDIATE-NM-TRANSMISSIONS`, `NM-MSG-CYCLE-TIME`, and `NM-RE-`

MOTE-SLEEP-INDICATION-TIME are configured as NM-BUSLOAD-REDUCTION-ACTIVE, NM-WAIT-BUS-SLEEP-TIME, NM-IMMEDIATE-NM-CYCLE-TIME, NM-IMMEDIATE-NM-TRANSMISSIONS, NM-MSG-CYCLE-TIME, and NM-REMOTE-SLEEP-INDICATION-TIME in the target model entity CAN-NM-CLUSTER, see [Section 8.1.2.7.3, “CAN-NM-CLUSTER”](#).

NM-BUS-LOAD-REDUCTION-ENABLED

The parameter NM-BUS-LOAD-REDUCTION-ENABLED is configured as NM-BUSLOAD-REDUCTION-ENABLED in the target model entity CAN-NM-CLUSTER-COUPLING, see [Section 8.1.1.7.8, “FLEXRAY-NM-CLUSTER-COUPLING”](#).

NM-LOWER-CAN-ID and NM-UPPER-CAN-ID

The parameters NM-LOWER-CAN-ID and NM-UPPER-CAN-ID are configured in the target model entity CAN-NM-RANGE-CONFIG parameters LOWER-CAN-ID and UPPER-CAN-ID, which is aggregated in CAN-NM-NODE, see [Section 8.1.1.7.4, “CAN-NM-NODE”](#).

8.1.3.1.2. CAN-COMMUNICATION-CONNECTOR

Source model entity: CAN-COMMUNICATION-CONNECTOR

Target model entity: CAN-COMMUNICATION-CONNECTOR

Target model property	Source model property
CATEGORY	CATEGORY
COMM-CONTROLLER-REF	COMM-CONTROLLER-REF
ECU-COMM-PORT-INSTANCES	ECU-COMM-PORT-INSTANCES
PNC-WAKEUP-CAN-ID	PNC-WAKEUP-CAN-ID
PNC-WAKEUP-CAN-ID-EXTENDED	PNC-WAKEUP-CAN-ID-EXTENDED
PNC-WAKEUP-CAN-ID-MASK	PNC-WAKEUP-CAN-ID-MASK
PNC-WAKEUP-DATA-MASK	PNC-WAKEUP-DATA-MASK
PNC-WAKEUP-DLC	PNC-WAKEUP-DLC
PNC-GATEWAY-TYPE	PNC-GATEWAY-TYPE
No target model parameter is available.	NM-ENABLED

Changed or moved parameters

CHANNEL-REF

The source model parameter CHANNEL-REF has been replaced by the target model parameter COMM-CONNECTORS/COMMUNICATION-CONNECTOR-REF-CONDITIONAL of the target model entity CAN-PHYSICAL-CHANNEL.

NM-ADDRESS

The source model parameter NM-ADDRESS is configured in the target model entity CAN-NM-NODE, see [Section 8.1.2.7.4, “CAN-NM-NODE”](#).

8.1.3.1.3. FLEXRAY-CLUSTER

Source model entity: FLEXRAY-CLUSTER

Target model entity: FLEXRAY-CLUSTER/FLEXRAY-CLUSTER-CONDITIONAL

Target model property	Source model property
PHYSICAL-CHANNELS	PHYSICAL-CHANNELS
PROTOCOL-NAME	PROTOCOL-NAME
PROTOCOL-VERSION	PROTOCOL-VERSION
SPEED	SPEED/1000
ACTION-POINT-OFFSET	ACTION-POINT-OFFSET
BIT	BIT
CAS-RX-LOW-MAX	CAS-RX-LOW-MAX
COLD-START-ATTEMPTS	COLD-START-ATTEMPTS
CYCLE	CYCLE
CYCLE-COUNT-MAX	CYCLE-COUNT-MAX
DYNAMIC-SLOT-IDLE-PHASE	DYNAMIC-SLOT-IDLE-PHASE
LISTEN-NOISE	LISTEN-NOISE
MACRO-PER-CYCLE	MACRO-PER-CYCLE
MACROTICK-DURATION	MACROTICK-DURATION
MAX-WITHOUT-CLOCK-CORRECTION-FATAL	MAX-WITHOUT-CLOCK-CORRECTION-FATAL
MAX-WITHOUT-CLOCK-CORRECTION-PASSIVE	MAX-WITHOUT-CLOCK-CORRECTION-PASSIVE
MINISLOT-ACTION-POINT-OFFSET	MINISLOT-ACTION-POINT-OFFSET
MINISLOT-DURATION	MINISLOT-DURATION
NETWORK-IDLE-TIME	NETWORK-IDLE-TIME
NETWORK-MANAGEMENT-VECTOR-LENGTH	NETWORK-MANAGEMENT-VECTOR-LENGTH
NUMBER-OF-MINISLOTS	NUMBER-OF-MINISLOTS
NUMBER-OF-STATIC-SLOTS	NUMBER-OF-STATIC-SLOTS
OFFSET-CORRECTION-START	OFFSET-CORRECTION-START
PAYLOAD-LENGTH-STATIC	PAYLOAD-LENGTH-STATIC

Target model property	Source model property
SAMPLE-CLOCK-PERIOD	SAMPLE-CLOCK-PERIOD
STATIC-SLOT-DURATION	STATIC-SLOT-DURATION
SYMBOL-WINDOW	SYMBOL-WINDOW
SYMBOL-WINDOW-ACTION-POINT-OFFSET	SYMBOL-WINDOW-ACTION-POINT-OFFSET
SYNC-FRAME-ID-COUNT-MAX	SYNC-FRAME-ID-COUNT-MAX
TRANSMISSION-START-SEQUENCE-DURATION	TRANSMISSION-START-SEQUENCE-DURATION
WAKEUP-RX-IDLE	WAKEUP-RX-IDLE
WAKEUP-RX-LOW	WAKEUP-RX-LOW
WAKEUP-RX-WINDOW	WAKEUP-RX-WINDOW
WAKEUP-TX-ACTIVE	WAKEUP-TX-ACTIVE
WAKEUP-TX-IDLE	WAKEUP-TX-IDLE
No target model parameter is available.	MAX-FRAME-LENGTH
No target model parameter is available.	NM-REPEAT-MESSAGE-SUPPORT
No target model parameter is available.	BUS-GUARDIAN-ENABLE-PART
No target model parameter is available.	CAS-RX-LOW-MIN
No target model parameter is available.	MACRO-INITIAL-OFFSET
No target model parameter is available.	MAX-INITIALISATION-ERROR
No target model parameter is available.	MAX-PROPAGATION-DELAY
No target model parameter is available.	MIN-PROPAGATION-DELAY
No target model parameter is available.	OFFSET-CORRECTION-MAX
DETECT-NIT-ERROR	No source model parameter is available.
IGNORE-AFTER-TX	IGNORE-AFTER-TX
SAFETY-MARGIN	No source model parameter is available.
No target model parameter is available.	TRANSCEIVER-STANDBY-DELAY
No target model parameter is available.	NM-COORDINATOR-SYNC-PRIO

Changed or moved parameters

NM-NODE-DETECTION-ENABLED and NM-NODE-ID-ENABLED

The source model parameters NM-NODE-DETECTION-ENABLED and NM-NODE-ID-ENABLED are configured in the target model entity NM-ECU, see [Section 8.1.2.7.2, “NM-ECU”](#).

SPEED

In the target model, the parameter SPEED is given in kbit/s, whereas the source model provides it in bit/s. If the source model parameter is not divisible by 1000 without rest, it cannot be upgraded at all.

NM-DATA-CYCLE, NM-READY-SLEEP-COUNT, NM-REMOTE-SLEEP-INDICATION-TIME, NM-REPEAT-MESSAGE-TIME, NM-REPETITION-CYCLE, NM-VOTING-CYCLE

The source model parameters NM-DATA-CYCLE, NM-READY-SLEEP-COUNT, NM-REMOTE-SLEEP-INDICATION-TIME, NM-REPEAT-MESSAGE-TIME, NM-REPETITION-CYCLE, and NM-VOTING-CYCLE are configured in the target model entity FLEXRAY-NM-CLUSTER, see [Section 8.1.2.7.6, “FLEXRAY-NM-CLUSTER”](#).

Reason for not upgraded source model parameter

MAX-FRAME-LENGTH, BUS-GUARDIAN-ENABLE-PART, CAS-RX-LOW-MIN, CAS-RX-LOW-MIN, MACRO-INITIAL-OFFSET, MAX-INITIALISATION-ERROR, MAX-PROPAGATION-DELAY, MIN-PROPAGATION-DELAY, OFFSET-CORRECTION-MAX, NM-REPEAT-MESSAGE-SUPPORT

The parameters MAX-FRAME-LENGTH, BUS-GUARDIAN-ENABLE-PART, CAS-RX-LOW-MIN, CAS-RX-LOW-MIN, MACRO-INITIAL-OFFSET, MAX-INITIALISATION-ERROR, MAX-PROPAGATION-DELAY, MIN-PROPAGATION-DELAY, OFFSET-CORRECTION-MAX, NM-REPEAT-MESSAGE-SUPPORT are not upgraded to the target model, because they do not contribute any data required for configuring an ECU. Therefore, they have been removed from the target model.

8.1.3.1.4. FLEXRAY-COMMUNICATION-CONNECTOR

Source model entity: FLEX-RAY-COMMUNICATION-CONNECTOR

Target model entity: FLEXRAY-COMMUNICATION-CONNECTOR

Target model property	Source model property
CATEGORY	CATEGORY
COMM-CONTROLLER-REF	COMM-CONTROLLER-REF
ECU-COMM-PORT-INSTANCES	ECU-COMM-PORT-INSTANCES
WAKE-UP-CHANNEL	WAKE-UP-CHANNEL
PNC-GATEWAY-TYPE	PNC-GATEWAY-TYPE
No target model parameter is available.	NM-READY-SLEEP-TIME
No target model parameter is available.	NM-ENABLED

Changed or moved parameters

ECU-COMM-PORT-INSTANCES

In ECU-COMM-PORT-INSTANCES, the aggregated source model entities are upgraded to their target model counterparts: FRAME-PORT, I-PDU-PORT, SIGNAL-PORT.

CHANNEL-REF

The source model parameter CHANNEL-REF has been replaced by the target model parameter COMM-CONNECTORS/COMMUNICATION-CONNECTOR-REF-CONDITIONAL of the target model entity FLEXRAY-PHYSICAL-CHANNEL.

NM-ADDRESS

The source model parameter NM-ADDRESS is configured in the target model entity FLEXRAY-NM-NODE, see [Section 8.1.2.7.7, “FLEXRAY-NM-NODE”](#).

8.1.3.1.5. LIN-CLUSTER

Source model entity: LIN-CLUSTER

Target model entity: LIN-CLUSTER/LIN-CLUSTER-CONDITIONAL

Target model property	Source model property
PHYSICAL-CHANNELS	PHYSICAL-CHANNELS
PROTOCOL-NAME	PROTOCOL-NAME
PROTOCOL-VERSION	PROTOCOL-VERSION
SPEED	SPEED/1000
No target model parameter is available.	MAX-FRAME-LENGTH
No target model parameter is available.	NM-REPEAT-MESSAGE-SUPPORT
No target model parameter is available.	NM-COORDINATOR-SYNC-PRIO

Changed or moved parameters

SPEED

In the target model, the parameter SPEED is given in kbit/s, whereas the source model provides it in bit/s. If the source model parameter is not divisible by 1000 without rest, it cannot be upgraded at all.

LIN-SCHEDULE-TABLE entities of the source model parameter SCHEDULE-TABLES

The LIN-SCHEDULE-TABLE entities aggregated in the source model in SCHEDULE-TABLES are moved to the corresponding LIN-PHYSICAL-CHANNEL in the target model, see [Section 8.1.1.16, “LIN-PHYSICAL-CHANNEL”](#).

8.1.3.1.6. LIN-COMMUNICATION-CONNECTOR

Source model entity: COMMUNICATION-CONNECTOR

Target model entity: LIN-COMMUNICATION-CONNECTOR

Target model property	Source model property
CATEGORY	CATEGORY
COMM-CONTROLLER-REF	COMM-CONTROLLER-REF
ECU-COMM-PORT-INSTANCES	ECU-COMM-PORT-INSTANCES
LIN-CONFIGURABLE-FRAMES / LIN-CONFIGURABLE-FRAME	No source model parameter is available.
PNC-GATEWAY-TYPE	PNC-GATEWAY-TYPE
No target model parameter is available.	NM-ENABLED

Changed or moved parameters

ECU-COMM-PORT-INSTANCES

In ECU-COMM-PORT-INSTANCES, the aggregated source model entities are upgraded into their target model counterparts: FRAME-PORT, I-PDU-PORT, SIGNAL-PORT.

CHANNEL-REF

The source model parameter CHANNEL-REF has been replaced by the target model parameter COMM-CONNECTORS / COMMUNICATION-CONNECTOR-REF-CONDITIONAL of the target model entity LIN-PHYSICAL-CHANNEL.

INITIAL-NAD

The target model parameter INITIAL-NAD is configured by taking the CONFIGURED-NAD parameter value of the LIN-SLAVE which is referenced via COMM-CONTROLLER-REF.

8.1.3.1.7. LIN-SLAVE-CONFIG

The LIN-SLAVE-CONFIG entity has been introduced in AUTOSAR 3.2.2 to model LIN-SLAVE parameters without having to add a complete ECU-INSTANCE. Since the target model AUTOSAR 4.0.x does not support LIN-SLAVE-CONFIG, the source model LIN-SLAVE-CONFIG entity is translated into one ECU-INSTANCE, one LIN-SLAVE, and one LIN-COMMUNICATION-CONNECTOR in the target model.

New parameters in the target model

ECU-INSTANCE

A new ECU-INSTANCE is added to the target model for each source model LIN-SLAVE-CONFIG. Its SHORT-NAME parameter is set to LinSlave_<configured_nad>, where <configured_nad> is the CONFIGURED-NAD parameter value of the source model LIN-SLAVE-CONFIG. One LIN-SLAVE element is added to its COMM-CONTROLLERS parameter, one LIN-COMMUNICATION-CONNECTOR element is added to its CONNECTORS parameter.

LIN-SLAVE

A new LIN-SLAVE is added to the target model for each source model LIN-SLAVE-CONFIG. Its SHORT-NAME parameter is set to LinSlave_<configured_nad>_CommController, where <configured_nad> is the CONFIGURED-NAD parameter value of the source model LIN-SLAVE-CONFIG. The source model LIN-SLAVE-CONFIG/LIN-ERROR-RESPONSE parameter is upgraded to LIN-SLAVE/LIN-SLAVE-VARIANTS/LIN-SLAVE-CONDITIONAL/LIN-ERROR-RESPONSE in the target model. The source model parameters LIN-SLAVE-CONFIG/CONFIGURED-NAD and LIN-SLAVE-CONFIG/PROTOCOL-VERSION are upgraded to LIN-SLAVE/LIN-SLAVE-VARIANTS/LIN-SLAVE-CONDITIONAL/CONFIGURED-NAD and LIN-SLAVE/LIN-SLAVE-VARIANTS/LIN-SLAVE-CONDITIONAL/PROTOCOL-VERSION.

LIN-COMMUNICATION-CONNECTOR

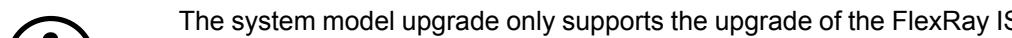
A new LIN-COMMUNICATION-CONNECTOR is added to the target model for each source model LIN-SLAVE-CONFIG. Its SHORT-NAME parameter is set to LinSlave_<configured_nad>_CommConnector, where <configured_nad> is the CONFIGURED-NAD parameter value of the source model LIN-SLAVE-CONFIG. LIN-COMMUNICATION-CONNECTOR/COMM-CONTROLLER-REF references the target model LIN-SLAVE element that has been created for the source model LIN-SLAVE-CONFIG. Moreover, a COMMUNICATION-CONNECTOR-REF-CONDITIONAL parameter referencing this LIN-COMMUNICATION-CONNECTOR is added to all LIN-PHYSICAL-CHANNELS which also reference any of the LIN-COMMUNICATION-CONNECTORS belonging to the target model LIN-MASTER counterpart of the source model LIN-MASTER element which aggregates the LIN-SLAVE-CONFIG.

8.1.3.2. FlexRay ISO Transport Layer

NOTE

Non-ISO FlexRay Transport Layer upgrade is not supported

The system model upgrade only supports the upgrade of the FlexRay ISO Transport Layer. The upgrade of the Non-ISO FlexRay Transport Layer variant is *not* supported.



8.1.3.2.1. FLEXRAY-TP-ECU

Source model entity: ECU-INSTANCE

Target model entity: FLEXRAY-TP-ECU

Target model property	Source model property
CYCLE-TIME-MAIN-FUNCTION	CYCLE-TIME-MAIN-FUNCTION
FULL-DUPLEX-ENABLED	FULL-DUPLEX-ENABLED
No target model parameter is available.	TRANSMIT-CANCELLATION

8.1.3.3. Network Management

8.1.3.3.1. NM-COORDINATOR

In the following all entities used to create an NM-COORDINATOR are located in the target model if not stated otherwise.

To every NM-ECU, which is referenced by at least two NM-NODES of different NM-CLUSTERS, a NM-COORDINATOR is added. The NM-CLUSTERS must also reference COMMUNICATION-CLUSTERS which have an NM-GLOBAL-COORDINATOR-TIME defined in the corresponding COMMUNICATION-CLUSTER of the source model. NM-GLOBAL-COORDINATOR-TIME of the NM-COORDINATOR is then set to the maximum of all NM-GLOBAL-COORDINATOR-TIME values found in these COMMUNICATION-CLUSTERS.

For each NM-NODE referencing the NM-ECU to which an NM-COORDINATOR has been added, a NM-NODE-REF referencing the NM-NODE is added to the NM-COORDINATOR's NM-NODE-REFS.

8.1.3.4. Software Component Description

8.1.3.4.1. APPLICATION-SW-COMPONENT-TYPE

Source model entities: APPLICATION-SOFTWARE-COMPONENT-TYPE, SENSOR-ACTUATOR-SOFT-WARE-COMPONENT-TYPE

Target model entity: APPLICATION-SW-COMPONENT-TYPE

Target model property	Source model property
PORTS/P-PORT-PROTOTYPE	PORTS/P-PORT-PROTOTYPE
PORTS/R-PORT-PROTOTYPE	PORTS/R-PORT-PROTOTYPE
PORT-GROUPS/PORT-GROUP	PORT-GROUPS/PORT-GROUP
SYMBOL-PROPS/SHORT-NAME	SYMBOL-PROPS/SHORT-NAME
SYMBOL-PROPS/SYMBOL	SYMBOL-PROPS/SYMBOL

New parameters in the target model

INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR

The parameter INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR contains the SWC-INTERNAL-BEHAVIOR which corresponds to the source model INTERNAL-BEHAVIOR which references the APPLICATION-SOFTWARE-COMPONENT-TYPE via COMPONENT-REF.

8.1.3.4.2. SERVICE-SW-COMPONENT-TYPE

Source model entity: SERVICE-COMPONENT-TYPE

Target model entity: SERVICE-SW-COMPONENT-TYPE

Target model property	Source model property
PORTS/P-PORT-PROTOTYPE	PORTS/P-PORT-PROTOTYPE
PORTS/R-PORT-PROTOTYPE	PORTS/R-PORT-PROTOTYPE
PORT-GROUPS/PORT-GROUP	PORT-GROUPS/PORT-GROUP
SYMBOL-PROPS/SHORT-NAME	SYMBOL-PROPS/SHORT-NAME
SYMBOL-PROPS/SYMBOL	SYMBOL-PROPS/SYMBOL
No upgrade is currently implemented.	BSW-MODULE-DESCRIPTION-REFS/BSW-MODULE-DESCRIPTION-REF

New parameters in the target model

INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR

The parameter INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR contains the SWC-INTERNAL-BEHAVIOR which corresponds to the source model INTERNAL-BEHAVIOR which references the APPLICATION-SOFTWARE-COMPONENT-TYPE via COMPONENT-REF.

8.1.3.4.3. COMPLEX-DEVICE-DRIVER-SW-COMPONENT-TYPE

Source model entity: COMPLEX-DEVICE-DRIVER-COMPONENT-TYPE

Target model entity: COMPLEX-DEVICE-DRIVER-SW-COMPONENT-TYPE

Target model property	Source model property
PORTS/P-PORT-PROTOTYPE	PORTS/P-PORT-PROTOTYPE
PORTS/R-PORT-PROTOTYPE	PORTS/R-PORT-PROTOTYPE
PORT-GROUPS/PORT-GROUP	PORT-GROUPS/PORT-GROUP

Target model property	Source model property
SYMBOL-PROPS/SHORT-NAME	SYMBOL-PROPS/SHORT-NAME
SYMBOL-PROPS/SYMBOL	SYMBOL-PROPS/SYMBOL
No upgrade is currently implemented.	BSW-MODULE-DESCRIPTION-REFS/BSW-MODULE-DESCRIPTION-REF
No upgrade is currently implemented.	HARDWARE-ELEMENT-REFS/HARDWARE-ELEMENT-REF

New parameters in the target model

INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR

The parameter INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR contains the SWC-INTERNAL-BEHAVIOR which corresponds to the source model INTERNAL-BEHAVIOR which references the APPLICATION-SOFTWARE-COMPONENT-TYPE via COMPONENT-REF.

8.1.3.4.4. ECU-ABSTRACTION-SW-COMPONENT-TYPE

Source model entity: ECU-ABSTRACTION-COMPONENT-TYPE

Target model entity: ECU-ABSTRACTION-SW-COMPONENT-TYPE

Target model property	Source model property
PORTS/P-PORT-PROTOTYPE	PORTS/P-PORT-PROTOTYPE
PORTS/R-PORT-PROTOTYPE	PORTS/R-PORT-PROTOTYPE
PORT-GROUPS/POR-T-GROUP	PORT-GROUPS/POR-T-GROUP
SYMBOL-PROPS/SHORT-NAME	SYMBOL-PROPS/SHORT-NAME
SYMBOL-PROPS/SYMBOL	SYMBOL-PROPS/SYMBOL
No upgrade is currently implemented.	BSW-MODULE-DESCRIPTION-REFS/BSW-MODULE-DESCRIPTION-REF
No upgrade is currently implemented.	HARDWARE-ELEMENT-REFS/HARDWARE-ELEMENT-REF

New parameters in the target model

INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR

The parameter INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR contains the SWC-INTERNAL-BEHAVIOR which corresponds to the source model INTERNAL-BEHAVIOR which references the APPLICATION-SOFTWARE-COMPONENT-TYPE via COMPONENT-REF.

8.1.3.4.5. NV-BLOCK-SW-COMPONENT-TYPE

Source model entity: NV-BLOCK-SW-COMPONENT-TYPE

Target model entity: NV-BLOCK-SW-COMPONENT-TYPE

Target model property	Source model property
POR TS/P-PORT-PROTOTYPE	POR TS/P-PORT-PROTOTYPE
POR TS/R-PORT-PROTOTYPE	POR TS/R-PORT-PROTOTYPE
PORT-GROUPS/PORT-GROUP	PORT-GROUPS/PORT-GROUP
SYMBOL-PROPS/SHORT-NAME	SYMBOL-PROPS/SHORT-NAME
SYMBOL-PROPS/SYMBOL	SYMBOL-PROPS/SYMBOL
NV-BLOCK-DESCRIPTORS/NV-BLOCK-DESCRIP-TOR	NV-BLOCK-DESCRIPTORS/NV-BLOCK-DESCRIP-TOR

New parameters in the target model

INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR

The parameter INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR contains the SWC-INTERNAL-BEHAVIOR which corresponds to the source model INTERNAL-BEHAVIOR which references the APPLICATION-SOFTWARE-COMPONENT-TYPE via COMPONENT-REF.

8.1.3.4.6. NV-BLOCK-DESCRIPTOR

Source model entity: NV-BLOCK-DESCRIPTOR

Target model entity: NV-BLOCK-DESCRIPTOR

Target model property	Source model property
CLIENT-SERVER-PORTS/ROLE-BASED-PORT-ASSIGNMENT	CLIENT-SERVER-PORTS/ROLE-BASED-PORT-ASSIGNMENT
NV-BLOCK-DATA-MAPPINGS/NV-BLOCK-DA-TA-MAPPING	NV-BLOCK-DATA-MAPPINGS/NV-BLOCK-DA-TA-MAPPING
NV-BLOCK-NEEDS	NV-BLOCK-NEEDS, SWC-NV-BLOCK-NEEDS
RAM-BLOCK	RAM-BLOCK
RAM-BLOCK/INIT-VALUE/CONSTANT-REFERENCE	RAM-BLOCK-INIT-VALUE-REF
ROM-BLOCK	ROM-BLOCK

Target model property	Source model property
ROM-BLOCK/INIT-VALUE/CONSTANT-REFERENCE	ROM-BLOCK-INIT-VALUE-REF

No configuration of target model parameters CONSTANT-VALUE-MAPPING-REFS , DATA-TYPE-MAPPING-REFS , INITIATION-DATA-DEF-PROPS

The target model parameters CONSTANT-VALUE-MAPPING-REFS , DATA-TYPE-MAPPING-REFS , INITIATION-DATA-DEF-PROPS are not configured because there are no source model counterparts.

New parameters in the target model

INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR

The parameter INTERNAL-BEHAVIORS/SWC-INTERNAL-BEHAVIOR contains the SWC-INTERNAL-BEHAVIOR which corresponds to the source model INTERNAL-BEHAVIOR which references the APPLICATION-SOFTWARE-COMPONENT-TYPE via COMPONENT-REF.

8.1.3.4.7. NV-BLOCK-DATA-MAPPING

Source model entity: NV-BLOCK-DATA-MAPPING

Target model entity: NV-BLOCK-DATA-MAPPING

Target model property	Source model property
No target model parameter is available.	NV-RAM-BLOCK-ELEMENT-REF/ARRAY-INDEX
NV-RAM-BLOCK-ELEMENT/LOCAL-VARIABLE-REF	NV-RAM-BLOCK-ELEMENT-REF/SUB-ELEMENT-REF
READ-NV-DATA/AUTOSAR-VARIABLE-IREF/PORT-PROTOTYPE-REF	READ-NV-DATA/DATA-PROTOTYPE-IREF/PORT-PROTOTYPE-REF
READ-NV-DATA/AUTOSAR-VARIABLE-IREF/TARGET-DATA-PROTOTYPE-REF	READ-NV-DATA/DATA-PROTOTYPE-IREF/DATA-PROTOTYPE-REF
WRITTEN-NV-DATA/AUTOSAR-VARIABLE-IREF/PORT-PROTOTYPE-REF	WRITTEN-NV-DATA/DATA-PROTOTYPE-IREF/PORT-PROTOTYPE-REF
WRITTEN-NV-DATA/AUTOSAR-VARIABLE-IREF/TARGET-DATA-PROTOTYPE-REF	WRITTEN-NV-DATA/DATA-PROTOTYPE-IREF/DATA-PROTOTYPE-REF

8.1.3.4.8. ROLE-BASED-PORT-ASSIGNMENT

Source model entities: ROLE-BASED-P-PORT-ASSIGNMENT , ROLE-BASED-R-PORT-ASSIGNMENT

Target model entity: ROLE-BASED-PORT-ASSIGNMENT

Target model property	Source model property
PORT-PROTOTYPE-REF	P-PORT-PROTOTYPE-REF, R-PORT-PROTOTYPE-REF
ROLE	ROLE

8.1.3.4.9. P-PORT-PROTOTYPE

Source model entity: P-PORT-PROTOTYPE

Target model entity: P-PORT-PROTOTYPE

Target model property	Source model property
PROVIDED-COM-SPECS/MODE-SWITCH-SEN-DER-COM-SPEC	PROVIDED-COM-SPECS/MODE-SWITCH-COM-SPEC
PROVIDED-COM-SPECS/NONQUEUED-SEN-DER-COM-SPEC	PROVIDED-COM-SPECS/UNQUEUED-SENDER-COM-SPEC
PROVIDED-COM-SPECS/NV-PROVIDE-COM-SPEC	PROVIDED-COM-SPECS/NV-PROVIDE-COM-SPEC
PROVIDED-COM-SPECS/PARAMETER-PROVIDE-COM-SPEC	PROVIDED-COM-SPECS/PARAMETER-PROVIDE-COM-SPEC
PROVIDED-COM-SPECS/QUEUED-SENDER-COM-SPEC	PROVIDED-COM-SPECS/QUEUED-SENDER-COM-SPEC
PROVIDED-COM-SPECS/SERVER-COM-SPEC	PROVIDED-COM-SPECS/SERVER-COM-SPEC
PROVIDED-INTERFACE-TREF	PROVIDED-INTERFACE-TREF

8.1.3.4.10. NONQUEUED-RECEIVER-COM-SPEC

Source model entity: UNQUEUED-RECEIVER-COM-SPEC

Target model entity: NONQUEUED-RECEIVER-COM-SPEC

Target model property	Source model property
ALIVE-TIMEOUT	ALIVE-TIMEOUT
No target model parameter is available.	HANDLE-INVALID
INIT-VALUE/CONSTANT-REFERENCE	INIT-VALUE-REF
No target model parameter is available.	RESYNC-TIME

Target model property	Source model property
DATA-ELEMENT-REF	DATA-ELEMENT-REF
FILTER	FILTER
HANDLE-NEVER-RECEIVED	HANDLE-NEVER-RECEIVED if the source model parameter value is available, else false.
ENABLE-UPDATE	ENABLE-UPDATE
HANDLE-OUT-OF-RANGE	No source model parameter is available, the target model parameter is set to NONE.
HANDLE-TIMEOUT-TYPE	No source model parameter is available, the target model parameter is set to NONE.
NETWORK-REPRESENTATION	No source model parameter is available.
USES-END-TO-END-PROTECTION	USES-END-TO-END-PROTECTION
EXTERNAL-REPLACEMENT-REF	No source model parameter is available.
MAX-DELTA-COUNTER-INIT	MAX-DELTA-COUNTER-INIT
No target model parameter is available.	MAX-NO-NEW-OR-REPEATED-DATA
No target model parameter is available.	SYNC-COUNTER-INIT

Processing of source model parameter HANDLE-INVALID

If the parameter HANDLE-INVALID is provided in the source model, a INVALIDATION-POLICY element is created in the related target model SENDER-RECEIVER-INTERFACE. The INVALIDATION-POLICY parameter HANDLE-INVALID is configured according to the source model UNQUEUED-RECEIVER-COM-SPEC parameter HANDLE-INVALID if all UNQUEUED-RECEIVER-COM-SPEC elements belonging to the source model SENDER-RECEIVER-INTERFACE and DATA-ELEMENT are providing the same HANDLE-INVALID value.

8.1.3.4.11. QUEUED-RECEIVER-COM-SPEC

Source model entity: QUEUED-RECEIVER-COM-SPEC

Target model entity: QUEUED-RECEIVER-COM-SPEC

Target model property	Source model property
DATA-ELEMENT-REF	DATA-ELEMENT-REF
No target model parameter is available.	FILTER
HANDLE-OUT-OF-RANGE	No source model parameter is available, the target model parameter is set to NONE.

Target model property	Source model property
HANDLE-OUT-OF-RANGE-STATUS	No source model parameter is available.
NETWORK-REPRESENTATION	No source model parameter is available.
QUEUE-LENGTH	QUEUE-LENGTH
USES-END-TO-END-PROTECTION	No source model parameter is available, the target model parameter is set to false.
MAX-DELTA-COUNTER-INIT	MAX-DELTA-COUNTER-INIT
No target model parameter is available.	MAX-NO-NEW-OR-REPEATED-DATA
No target model parameter is available.	SYNC-COUNTER-INIT

8.1.3.4.12. NV-PROVIDE-COM-SPEC

Source model entity: NV-PROVIDE-COM-SPEC

Target model entity: NV-PROVIDE-COM-SPEC

Target model property	Source model property
VARIABLE-REF	NV-DATA-ELEMENT-REF
RAM-BLOCK-INIT-VALUE/CONSTANT-REFERENCE	RAM-BLOCK-INIT-VALUE-REF
ROM-BLOCK-INIT-VALUE/CONSTANT-REFERENCE	ROM-BLOCK-INIT-VALUE-REF

8.1.3.4.13. RUNNABLE-ENTITY

Source model entity: RUNNABLE-ENTITY

Target model entity: RUNNABLE-ENTITY

Target model property	Source model property
ARGUMENTS/RUNNABLE-ENTITY-ARGUMENT	ARGUMENTS/RUNNABLE-ENTITY-ARGUMENT
Currently no upgrade is available.	BSW-ENTITY-REF
PARAMETER-ACCESSSS/PARAMETER-ACCESS/ACCESSED-PARAMETER	CALPRM-ACCESSSS/CALPRM-ACCESS
CAN-BE-INVOKED-CONCURRENTLY	CAN-BE-INVOKED-CONCURRENTLY
CAN-ENTER-EXCLUSIVE-AREA-REFS/CAN-ENTER-EXCLUSIVE-AREA-REF	CAN-ENTER-EXCLUSIVE-AREA-REFS/CAN-ENTER-EXCLUSIVE-AREA-REF
DATA-READ-ACCESSSS/VARIABLE-ACCESS	DATA-READ-ACCESSSS/DATA-READ-ACCESS

Target model property	Source model property
DATA-RECEIVE-POINT-BY-ARGUMENTS/VARIABLE-ACCESS	DATA-RECEIVE-POINTS/DATA-RECEIVE-POINT
DATA-SEND-POINTS/VARIABLE-ACCESS	DATA-SEND-POINTS/DATA-SEND-POINT
DATA-WRITE-ACCESSSS/VARIABLE-ACCESS	DATA-WRITE-ACCESSSS/DATA-WRITE-ACCESS
MINIMUM-START-INTERVAL	MINIMUM-START-INTERVAL
MODE-SWITCH-POINTS/MODE-SWITCH-POINT	MODE-SWITCH-POINTS/MODE-SWITCH-POINT
PARAMETER-ACCESSSS/PARAMETER-ACCESS/ACCESSED-PARAMETER/LOCAL-PARAMETER-REF	PER-INSTANCE-CALPRM-ACCESS-REFS/PER-INSTANCE-CALPRM-ACCESS-REF
RUNS-INSIDE-EXCLUSIVE-AREA-REFS/RUNS-INSIDE-EXCLUSIVE-AREA-REF	RUNS-INSIDE-EXCLUSIVE-AREA-REFS/RUNS-INSIDE-EXCLUSIVE-AREA-REF
SERVER-CALL-POINTS/ASYNCHRONOUS-SERVER-CALL-POINT	SERVER-CALL-POINTS/ASYNCHRONOUS-SERVER-CALL-POINT
SERVER-CALL-POINTS/SYNCHRONOUS-SERVER-CALL-POINT	SERVER-CALL-POINTS/SYNCHRONOUS-SERVER-CALL-POINT
PARAMETER-ACCESSSS/PARAMETER-ACCESS/ACCESSED-PARAMETER/LOCAL-PARAMETER-REF	SHARED-CALPRM-ACCESS-REFS/SHARED-CALPRM-ACCESS-REF
SYMBOL	SYMBOL
WAIT-POINTS/WAIT-POINT	WAIT-POINTS/WAIT-POINT
WRITTEN-LOCAL-VARIABLES/VARIABLE-ACCESS/ACCESSED-VARIABLE/LOCAL-VARIABLE-REF	WRITTEN-VARIABLE-REFS/WRITTEN-VARIABLE-REF
READ-LOCAL-VARIABLES/VARIABLE-ACCESS/ACCESSED-VARIABLE/LOCAL-VARIABLE-REF	READ-VARIABLE-REFS/READ-VARIABLE-REF

Migration of source model DATA-RECEIVE-POINT elements

Please note that for all source model DATA-RECEIVE-POINT elements are migrated to DATA-RECEIVE-POINT-BY-ARGUMENTS/VARIABLE-ACCESS elements in the target model, i.e. there is no conversion to the target model entities DATA-RECEIVE-POINT-BY-VALUES/VARIABLE-ACCESS.

Creation of target model ASYNCHRONOUS-SERVER-CALL-RESULT-POINT elements

Please note that for every source model ASYNCHRONOUS-SERVER-CALL-RETURNS-EVENT entity which either references a RUNNABLE-ENTITY via START-ON-EVENT-REF or references a SERVER-CALL-POINT

of the RUNNABLE-ENTITY via EVENT-SOURCE-REF, one ASYNCHRONOUS-SERVER-CALL-RESULT-POINTS/ASYNCHRONOUS-SERVER-CALL-RESULT-POINT is created in the target model.

8.1.3.4.14. R-PORT-PROTOTYPE

Source model entity: R-PORT-PROTOTYPE

Target model entity: R-PORT-PROTOTYPE

Target model property	Source model property
REQUIRED-COM-SPECS/CLIENT-COM-SPEC	REQUIRED-COM-SPECS/CLIENT-COM-SPEC
REQUIRED-COM-SPECS/PARAMETER-REQUIRE-COM-SPEC	REQUIRED-COM-SPECS/PARAMETER-REQUIRE-COM-SPEC
REQUIRED-COM-SPECS/QUEUED-RECEIVER-COM-SPEC	REQUIRED-COM-SPECS/QUEUED-RECEIVER-COM-SPEC
REQUIRED-COM-SPECS/NONQUEUED-RECEIVER-COM-SPEC	REQUIRED-COM-SPECS/UNQUEUED-RECEIVER-COM-SPEC
REQUIRED-COM-SPECS/NV-REQUIRE-COM-SPEC	REQUIRED-COM-SPECS/NV-REQUIRE-COM-SPEC
REQUIRED-INTERFACE-TREF	REQUIRED-INTERFACE-TREF

8.1.3.4.15. NV-REQUIRE-COM-SPEC

Source model entity: NV-REQUIRE-COM-SPEC

Target model entity: NV-REQUIRE-COM-SPEC

Target model property	Source model property
VARIABLE-REF	NV-DATA-ELEMENT-REF
INIT-VALUE/CONSTANT-REFERENCE	INIT-VALUE-REF

8.1.3.4.16. NV-DATA-INTERFACE

Source model entity: NV-DATA-INTERFACE

Target model entity: NV-DATA-INTERFACE

Target model property	Source model property
NV-DATAS/VARIABLE-DATA-PROTOTYPE	NV-DATAS/DATA-ELEMENT-PROTOTYPE

Target model property	Source model property
IS-SERVICE	IS-SERVICE
SERVICE-KIND	No source model parameter is available.

8.1.3.4.17. COMPU-SCALE

Source model entity: COMPU-SCALE

Target model entity: COMPU-SCALE

Target model property	Source model property
LOWER-LIMIT	LOWER-LIMIT
UPPER-LIMIT	UPPER-LIMIT
COMPU-CONST	COMPU-CONST
COMPU-INVERSE-VALUE	COMPU-INVERSE-VALUE
COMPU-RATIONAL-COEFFS/COMPU-NUMERATOR	COMPU-RATIONAL-COEFFS/COMPU-NUMERATOR
COMPU-RATIONAL-COEFFS/COMPU-DENOMINATOR	COMPU-RATIONAL-COEFFS/COMPU-DENOMINATOR
MASK	No source model parameter is available.
SHORT-LABEL	SHORT-LABEL
SYMBOL	SYMBOL

8.1.3.4.18. NV-BLOCK-NEEDS

Source model entities: BLOCK-NEEDS, SWC-NV-BLOCK-NEEDS

Target model entity: NV-BLOCK-NEEDS

Target model property	Source model property
N-DATA-SETS	NV-BLOCK-NEEDS/N-DATA-SETS
READONLY	NV-BLOCK-NEEDS/READONLY
RESISTANT-TO-CHANGED-SW	NV-BLOCK-NEEDS/RESISTANT-TO-CHANGED-SW
RESTORE-AT-START	NV-BLOCK-NEEDS/RESTORE-AT-START
STORE-AT-SHUTDOWN	NV-BLOCK-NEEDS/STORE-AT-SHUTDOWN
WRITE-ONLY-ONCE	NV-BLOCK-NEEDS/WRITE-ONLY-ONCE
WRITING-FREQUENCY	NV-BLOCK-NEEDS/WRITING-FREQUENCY

Target model property	Source model property
WRITING-PRIORITY	NV-BLOCK-NEEDS/WRITING-PRIORITY

No upgrade of source model SWC-NV-BLOCK-NEEDS/CALL-BACK-PORTS, SWC-NV-BLOCK-NEEDS/NV-DATA-PORT-PROTOTYPE-REFS, SWC-NV-BLOCK-NEEDS/DEFAULT-BLOCK-REF, SWC-NV-BLOCK-NEEDS/MIRROR-BLOCK-REF, SWC-NV-BLOCK-NEEDS/SERVICE-CALL-PORTS

The source model parameters SWC-NV-BLOCK-NEEDS/CALL-BACK-PORTS, SWC-NV-BLOCK-NEEDS/NV-DATA-PORT-PROTOTYPE-REFS, SWC-NV-BLOCK-NEEDS/DEFAULT-BLOCK-REF, SWC-NV-BLOCK-NEEDS/MIRROR-BLOCK-REF, SWC-NV-BLOCK-NEEDS/SERVICE-CALL-PORTS are not upgraded into the target model because there are no corresponding parameters.

No upgrade of source model NV-BLOCK-NEEDS/RAM-BLOCK-STATUS-CONTROL

The source model parameter NV-BLOCK-NEEDS/RAM-BLOCK-STATUS-CONTROL is not upgraded into the target model because there is no corresponding parameter.

No configuration of target model parameters CALC-RAM-BLOCK-CRC, CHECK-STATIC-BLOCK-ID, N-ROM-BLOCKS, WRITE-VERIFICATION

The target model parameters CALC-RAM-BLOCK-CRC, CHECK-STATIC-BLOCK-ID, N-ROM-BLOCKS, WRITE-VERIFICATION are not configured during model upgrade because there is no source model counterpart.

No configuration of target model parameter RELIABILITY

The source model parameter RELIABILITY is not upgraded to the target model parameter RELIABILITY because the parameter values are not related.

Additional configuration of ASSIGNED-DATAS/ROLE-BASED-DATA-ASSIGNMENT for SWC-NV-BLOCK-NEEDS elements

For every SWC-NV-BLOCK-NEEDS element in the source model which either contains a valid DEFAULT-BLOCK_REF or a valid MIRROR-BLOCK-REF, one ASSIGNED-DATAS/ROLE-BASED-DATA-ASSIGNMENT element is created in the target model. The ROLE-BASED-DATA-ASSIGNMENT element is aggregated within the target model SWC-SERVICE-DEPENDENCY element which also aggregates the target model SWC-

NV-BLOCK-NEEDS **element**. ROLE-BASED-DATA-ASSIGNMENT/ROLE is set to ramMirror. ROLE-BASED-DATA-ASSIGNMENT/USED-PIM-REF references the target model counterpart of the source model PER-INSTANCE-MEMORY which is referenced by SWC-NV-BLOCK-NEEDS/MIRROR-BLOCK-REF. ROLE-BASED-DATA-ASSIGNMENT/USED-PARAMETER-ELEMENT aggregates a AUTOSAR-PARAMETER-REF element which in turn references in LOCAL-PARAMETER-REF the target model counterpart of the source model CALPRM-ELEMENT-PROTOTYPE which is referenced by SWC-NV-BLOCK-NEEDS/DEFAULT-BLOCK-REF.

8.1.3.4.19. END-TO-END-PROTECTION

Source model entity: END-TO-END-PROTECTION

Target model entity: END-TO-END-PROTECTION

Target model property	Source model property
END-TO-END-PROFILE/COUNTER-OFFSET	END-TO-END-PROFILE/COUNTER-OFFSET
END-TO-END-PROFILE/CRC-OFFSET	END-TO-END-PROFILE/CRC-OFFSET
END-TO-END-PROFILE/DATA-ID-MODE	END-TO-END-PROFILE/DATA-ID-MODE
END-TO-END-PROFILE/DATA-IDS/DATA-ID	END-TO-END-PROFILE/DATA-IDS/DATA-ID
END-TO-END-PROFILE/DATA-LENGTH	END-TO-END-PROFILE/DATA-LENGTH
END-TO-END-PROFILE/MAX-DELTA-COUNTER-INIT	END-TO-END-PROFILE/MAX-DELTA-COUNTER-INIT
No target model parameter is available.	END-TO-END-PROFILE/MAX-NO-NEW-OR-REPEATED-DATA
No target model parameter is available.	END-TO-END-PROFILE/SYNC-COUNTER-INIT
END-TO-END-PROTECTION-I-SIGNAL-I-PDUS/ END-TO-END-PROTECTION-I-SIGNAL-I-PDU	END-TO-END-PROTECTION-I-SIGNAL-I-PDUS/ END-TO-END-PROTECTION-I-SIGNAL-I-PDU
END-TO-END-PROTECTION-VARIABLE-PROTOTYPES/END-TO-END-PROTECTION-VARIABLE-PROTOTYPE	END-TO-END-PROTECTION-DATA-ELEMENT-PROTOTYPES/END-TO-END-PROTECTION-DATA-ELEMENT-PROTOTYPE

8.1.3.5. BSW Module Description

8.1.3.5.1. BSW-MODULE-DESCRIPTION

Source model entities: BSW-MODULE-DESCRIPTION

Target model entity: BSW-MODULE-DESCRIPTION

Target model property	Source model property
BSW-MODULE-DEPENDENCIES/BSW-MODULE-DEPENDENCY	BSW-MODULE-DEPENDENCIES/BSW-MODULE-DEPENDENCY
MODULE-ID	MODULE-ID
OUTGOING-CALLBACKS/BSW-MODULE-ENTRY-TRY-REF-CONDITIONAL	OUTGOING-CALLBACK-REFS/OUTGOING-CALLBACK-REF
PROVIDED-ENTRIES/BSW-MODULE-ENTRY-REF-CONDITIONAL	PROVIDED-ENTRY-REFS/PROVIDED-ENTRY-REF
BSW-MODULE-DOCUMENTATIONS/SW-COMPONENT-DOCUMENTATION	No source model parameter is available.
PROVIDED-MODE-GROUPS/MODE-DECLARATION-GROUP-PROTOTYPE	No source model parameter is available.
RELEASED-TRIGGERS/TRIGGER	No source model parameter is available.
REQUIRED-MODE-GROUPS/MODE-DECLARATION-GROUP-PROTOTYPE	No source model parameter is available.
REQUIRED-TRIGGERS/TRIGGER	No source model parameter is available.

New parameters in the target model

INTERNAL-BEHAVIORS/BSW-INTERNAL-BEHAVIOR

The parameter INTERNAL-BEHAVIORS/BSW-INTERNAL-BEHAVIOR contains the BSW-INTERNAL-BEHAVIOR which corresponds to the source model BSW-BEHAVIOR which references the BSW-MODULE-DESCRIPTION via MODULE-REF.

8.1.3.5.2. BSW-MODULE-ENTRY

Source model entities: BSW-MODULE-ENTRY

Target model entity: BSW-MODULE-ENTRY

Target model property	Source model property
CALL-TYPE	CALL-TYPE
IS-SYNCHRONOUS	IS-SYNCHRONOUS
SERVICE-ID	SERVICE-ID
ARGUMENTS/SW-SERVICE-ARG	SW-SERVICE-ARGS/SW-SERVICE-ARG
RETURN-TYPE	SW-SERVICE-RETURN
ROLE	No source model parameter is available.

Configuration of target model parameter EXECUTION-CONTEXT

The target model parameter `EXECUTION-CONTEXT` is configured using the source model parameter `EXE-CUTION-CONTEXT` according to the following table:

Target model value	Source model value
TASK	TASK
INTERRUPT-CAT-1	INTERRUPT
UNSPECIFIED	UNSPECIFIED

Configuration of target model parameter IS-REENTRANT

The target model parameter `IS-REENTRANT` is configured using the source model parameter `SW-SERVICE-PROPS/SW-SERVICE-REENTRANCE`. If the source model parameter is available and has its value set to `REENTRANCE`, `IS-REENTRANT` is set to `true`, in all other cases to `false`.

Configuration of target model parameter SW-SERVICE-IMPL-POLICY

The target model parameter `SW-SERVICE-IMPL-POLICY` is configured using the source model parameter `SW-SERVICE-PROPS/SW-IMPL-POLICY`. If the source model parameter is not available, the target value is set to `STANDARD`.

8.1.3.5.3. SW-SERVICE-ARG

Source model entities: `SW-SERVICE-ARG`

Target model entity: `SW-SERVICE-ARG`

Target model property	Source model property
<code>SW-ARRAYSIZE</code>	<code>SW-ARRAYSIZE</code>
<code>SW-DATA-DEF-PROPS</code>	<code>SW-DATA-DEF-PROPS</code>
<code>DIRECTION</code>	No source model parameter is available.

8.1.3.5.4. RETURN-TYPE

Source model entities: `SW-SERVICE-RETURN`

Target model entity: `RETURN-TYPE`

The RETURN-TYPE is upgraded to SW-SERVICE-RETURN in the same way as described in [Section 8.1.3.5.3, "SW-SERVICE-ARG"](#).

8.1.3.5.5. BSW-MODULE-DEPENDENCY

Source model entities: BSW-MODULE-DEPENDENCY

Target model entity: BSW-MODULE-DEPENDENCY

Target model property	Source model property
EXPECTED-CALLBACKS/BSW-MODULE-EN-TRY-REF-CONDITIONAL	EXPECTED-CALLBACK-REFS/EXPECTED-CALLBACK-REF
REQUIRED-ENTRYS/BSW-MODULE-ENTRY-REF-CONDITIONAL	REQUIRED-ENTRY-REFS/REQUIRED-ENTRY-REF
TARGET-MODULE-ID	TARGET-MODULE-ID

Configuration of SERVICE-NEEDS parameters in BSW-MODULE-DEPENDENCY/SERVICE-ITEMS

The target model SERVICE-NEEDS elements in BSW-MODULE-DEPENDENCY/SERVICE-ITEMS are configured using their source model counterparts in BSW-MODULE-DEPENDENCY/SERVICE-ITEMS.

8.1.3.5.6. BSW-INTERNAL-BEHAVIOR

Source model entities: BSW-BEHAVIOR

Target model entity: BSW-INTERNAL-BEHAVIOR

Target model property	Source model property
ENTITYS/BSW-INTERRUPT-ENTITY	ENTITYS/BSW-INTERRUPT-ENTITY
ENTITYS/BSW-CALLED-ENTITY	ENTITYS/BSW-MODULE-ENTITY
ENTITYS/BSW-SCHEDULABLE-ENTITY	ENTITYS/BSW-SCHEDULABLE-ENTITY
EVENTS/BSW-TIMING-EVENT	EVENTS/BSW-CYCLIC-EVENT
EVENTS/BSW-INTERNAL-TRIGGER-OC-CURRED-EVENT	EVENTS/BSW-SPORADIC-EVENT
EXCLUSIVE-AREAS/EXCLUSIVE-AREA	EXCLUSIVE-AREAS/EXCLUSIVE-AREA
CONSTANT-MEMORYS/PARAMETER-DATA-PROTO-TYPE	No source model parameter is available.

Target model property	Source model property
CONSTANT-VALUE-MAPPING-REFS/CONSTANT-VALUE-MAPPING-REF	No source model parameter is available.
DATA-TYPE-MAPPING-REFS/DATA-TYPE-MAPPING-REF	No source model parameter is available.
INTERNAL-TRIGGERING-POINTS/BSW-INTERNAL-TRIGGERING-POINT	No source model parameter is available.
MODE-RECEIVER-POLICYS/BSW-MODE-RECEIVER-POLICY	No source model parameter is available.
MODE-SENDER-POLICYS/BSW-MODE-SENDER-POLICY	No source model parameter is available.
PER-INSTANCE-PARAMETERS/PARAMETER-DATA-PROTOTYPE	No source model parameter is available.
SCHEDULER-NAME-PREFIXS/BSW-SCHEDULER-NAME-PREFIX	No source model parameter is available.
SERVICE-DEPENDENCYS/BSW-SERVICE-DEPENDENCY	No source model parameter is available.
STATIC-MEMORYS/VARIABLE-DATA-PROTOTYPE	No source model parameter is available.
TRIGGER-DIRECT-IMPLEMENTATIONS/BSW-TRIGGER-DIRECT-IMPLEMENTATION	No source model parameter is available.

No configuration of BSW-BACKGROUND-EVENT, BSW-EXTERNAL-TRIGGER-OCCURRED-EVENT, BSW-MODE-SWITCH-EVENT, BSW-MODE-SWITCHED-ACK-EVENT in target model

The target model parameters EVENTS/BSW-BACKGROUND-EVENT, EVENTS/BSW-EXTERNAL-TRIGGER-OCCURRED-EVENT, EVENTS/BSW-MODE-SWITCH-EVENT, and EVENTS/BSW-MODE-SWITCHED-ACK-EVENT are not configured since there are no source model counterparts.

Configuration of source model parameter MODULE-REF

The target model counterpart of the source model BSW-MODULE-DESCRIPTION referenced via MODULE-REF contains the upgraded BSW-INTERNAL-BEHAVIOR in the target model.

8.1.3.5.7. BSW-INTERRUPT-ENTITY

Source model entities: BSW-INTERRUPT-ENTITY

Target model entity: BSW-INTERRUPT-ENTITY

Target model property	Source model property
ACTIVATION-POINTS/BSW-INTERNAL-TRIGGERING-POINT-REF-CONDITIONAL/BSW-INTERNAL-TRIGGERING-POINT-REF	ACTIVATION-POINT-REFS/ACTIVATION-POINT-REF
CALLED-ENTRYS/BSW-MODULE-ENTRY-REF-CONDITIONAL/BSW-MODULE-ENTRY-REF	CALLED-ENTRY-REFS/CALLED-ENTRY-REF
CAN-ENTER-EXCLUSIVE-AREA-REFS/CAN-ENTER-EXCLUSIVE-AREA-REF	CAN-ENTER-EXCLUSIVE-AREA-REFS/CAN-ENTER-EXCLUSIVE-AREA-REF
No target model parameter is available.	CANCELLATION-POINT-REFS/CANCELLATION-POINT-REF
IMPLEMENTED-ENTRY-REF	IMPLEMENTED-ENTRY-REF
INTERRUPT-CATEGORY	INTERRUPT-CATEGORY
INTERRUPT-SOURCE	INTERRUPT-SOURCE
ACCESSED-MODE-GROUPS/MODE-DECLARATION-GROUP-PROTOTYPE-REF-CONDITIONAL	No source model parameter is available.
ISSUED-TRIGGERS/TRIGGER-REF-CONDITIONAL	No source model parameter is available.
MANAGED-MODE-GROUPS/MODE-DECLARATION-GROUP-PROTOTYPE-REF-CONDITIONAL	No source model parameter is available.
MINIMUM-START-INTERVAL	No source model parameter is available, set to 0.
RUNS-INSIDE-EXCLUSIVE-AREA-REFS/RUNS-INSIDE-EXCLUSIVE-AREA-REF	No source model parameter is available.
SCHEDULER-NAME-PREFIX-REF	No source model parameter is available.
SW-ADDR-METHOD-REF	No source model parameter is available.

8.1.3.5.8. BSW-SCHEDULABLE-ENTITY

Source model entities: BSW-SCHEDULABLE-ENTITY

Target model entity: BSW-SCHEDULABLE-ENTITY

Target model property	Source model property
ACTIVATION-POINTS/BSW-INTERNAL-TRIGGERING-POINT-REF-CONDITIONAL/BSW-INTERNAL-TRIGGERING-POINT-REF	ACTIVATION-POINT-REFS/ACTIVATION-POINT-REF

Target model property	Source model property
CALLED-ENTRYS/BSW-MODULE-ENTRY-REF-CONDITIONAL/BSW-MODULE-ENTRY-REF	CALLED-ENTRY-REFS/CALLED-ENTRY-REF
CAN-ENTER-EXCLUSIVE-AREA-REFS/CAN-ENTER-EXCLUSIVE-AREA-REF	CAN-ENTER-EXCLUSIVE-AREA-REFS/CAN-ENTER-EXCLUSIVE-AREA-REF
No target model parameter is available.	CANCELLATION-POINT-REFS/CANCELLATION-POINT-REF
IMPLEMENTED-ENTRY-REF	IMPLEMENTED-ENTRY-REF
ACCESSED-MODE-GROUPS/MODE-DECLARATION-GROUP-PROTOTYPE-REF-CONDITIONAL	No source model parameter is available.
ISSUED-TRIGGERS/TRIGGER-REF-CONDITIONAL	No source model parameter is available.
MANAGED-MODE-GROUPS/MODE-DECLARATION-GROUP-PROTOTYPE-REF-CONDITIONAL	No source model parameter is available.
MINIMUM-START-INTERVAL	No source model parameter is available, set to 0.
RUNS-INSIDE-EXCLUSIVE-AREA-REFS/RUNS-INSIDE-EXCLUSIVE-AREA-REF	No source model parameter is available.
SCHEDULER-NAME-PREFIX-REF	No source model parameter is available.
SW-ADDR-METHOD-REF	No source model parameter is available.

8.1.3.5.9. BSW-CALLED-ENTITY

Source model entities: BSW-MODULE-ENTITY

Target model entity: BSW-CALLED-ENTITY

Target model property	Source model property
ACTIVATION-POINTS/BSW-INTERNAL-TRIGGERING-POINT-REF-CONDITIONAL/BSW-INTERNAL-TRIGGERING-POINT-REF	ACTIVATION-POINT-REFS/ACTIVATION-POINT-REF
CALLED-ENTRYS/BSW-MODULE-ENTRY-REF-CONDITIONAL/BSW-MODULE-ENTRY-REF	CALLED-ENTRY-REFS/CALLED-ENTRY-REF
CAN-ENTER-EXCLUSIVE-AREA-REFS/CAN-ENTER-EXCLUSIVE-AREA-REF	CAN-ENTER-EXCLUSIVE-AREA-REFS/CAN-ENTER-EXCLUSIVE-AREA-REF
No target model parameter is available.	CANCELLATION-POINT-REFS/CANCELLATION-POINT-REF

Target model property	Source model property
IMPLEMENTED-ENTRY-REF	IMPLEMENTED-ENTRY-REF
ACCESSED-MODE-GROUPS/MODE-DE-CLARATION-GROUP-PROTOTYPE-REF-CONDITIONAL	No source model parameter is available.
ISSUED-TRIGGERS/TRIGGER-REF-CONDITIONAL	No source model parameter is available.
MANAGED-MODE-GROUPS/MODE-DE-CLARATION-GROUP-PROTOTYPE-REF-CONDITIONAL	No source model parameter is available.
MINIMUM-START-INTERVAL	No source model parameter is available, set to 0.
RUNS-INSIDE-EXCLUSIVE-AREA-REFS/RUNS-INSIDE-EXCLUSIVE-AREA-REF	No source model parameter is available.
SCHEDULER-NAME-PREFIX-REF	No source model parameter is available.
SW-ADDR-METHOD-REF	No source model parameter is available.

8.1.3.5.10. BSW-TIMING-EVENT

Source model entities: BSW-CYCLIC-EVENT

Target model entity: BSW-TIMING-EVENT

Target model property	Source model property
PERIOD	CYCLE-TIME
STARTS-ON-EVENT-REF	STARTS-ON-EVENT-REF
DISABLED-IN-MODE-IREFS/DISABLED-IN-MODE-IREF	No source model parameter is available.

8.1.3.5.11. BSW-INTERNAL-TRIGGER-OCCURRED-EVENT

Source model entities: BSW-SPORADIC-EVENT

Target model entity: BSW-INTERNAL-TRIGGER-OCCURRED-EVENT

Target model property	Source model property
No target model parameter is available.	DELAY-TIME
STARTS-ON-EVENT-REF	STARTS-ON-EVENT-REF

Target model property	Source model property
DISABLED-IN-MODE-IREFS/DISABLED-IN-MODE-IREF	No source model parameter is available.

Creation of target model element BSW-INTERNAL-BEHAVIOR/INTERNAL-TRIGGERING-POINTS/BSW-INTERNAL-TRIGGERING-POINT

For every source model element BSW-BEHAVIOR/EVENTS/BSW-SPORADIC-EVENT one additional target model element BSW-INTERNAL-BEHAVIOR/INTERNAL-TRIGGERING-POINTS/BSW-INTERNAL-TRIGGERING-POINT is created, its SHORT-NAME is set to ITP_<BswEventName>, where <BswEventName> is the SHORT-NAME of the source model BSW-SPORADIC-EVENT. The target model BSW-SPORADIC-EVENT references the BSW-INTERNAL-TRIGGERING-POINT via EVENT-SOURCE-REF.

8.1.3.5.12. BSW-IMPLEMENTATION

Source model entity: BSW-IMPLEMENTATION

Target model entity: BSW-IMPLEMENTATION

Target model property	Source model property
AR-RELEASE-VERSION	AR-MAJOR-VERSION.AR-MINOR-VERSION.AR-PATCH-VERSION
CODE-DESCRIPTORS/CODE	CODE-DESCRIPTORS/CODE
USED-CODE-GENERATOR	CODE-GENERATOR
DEBUG-INFOS/BSW-DEBUG-INFO	No source model parameter is available.
GENERATED-ARTIFACTS/DEPENDENCY-ON-ARTIFACT	No source model parameter is available.
MC-SUPPORT	No source model parameter is available.
PROGRAMMING-LANGUAGE	PROGRAMMING-LANGUAGE
REQUIRED-ARTIFACTS/DEPENDENCY-ON-ARTIFACT	No source model parameter is available.
REQUIRED-GENERATOR-TOOLS/DEPENDENCY-ON-ARTIFACT	No source model parameter is available.
SW-VERSION	SW-MAJOR-VERSION.SW-MINOR-VERSION.SW-PATCH-VERSION
USED-CODE-GENERATOR	No source model parameter is available.

Target model property	Source model property
VENDOR-ID	VENDOR-ID
BEHAVIOR-REF	BEHAVIOR-REF
VENDOR-API-INFIX	VENDOR-API-INFIX

No upgrade of source model parameters COMPILERS/COMPILER, IMPLEMENTATION-DEPENDENCIES/DEPENDENCY-ON-FILE, IMPLEMENTATION-DEPENDENCIES/DEPENDENCY-ON-LIBRARY, LINKERS/LINKER, PROCESSOR-REFS/PROCESSOR-REF, RESOURCE-CONSUMPTION, PRECONFIGURED-CONFIGURATION-REF, RECOMMENDED-CONFIGURATION-REF, REQUIRED-HW-REFS/REQUIRED-HW-REF, VENDOR-SPECIFIC-MODULE-DEF-REF

The source model parameters COMPILERS/COMPILER, IMPLEMENTATION-DEPENDENCIES/DEPENDENCY-ON-FILE, IMPLEMENTATION-DEPENDENCIES/DEPENDENCY-ON-LIBRARY, LINKERS/LINKER, PROCESSOR-REFS/PROCESSOR-REF, RESOURCE-CONSUMPTION, PRECONFIGURED-CONFIGURATION-REF, RECOMMENDED-CONFIGURATION-REF, REQUIRED-HW-REFS/REQUIRED-HW-REF, VENDOR-SPECIFIC-MODULE-DEF-REF are not upgraded into the target model.

Creation of target model RESOURCE-CONSUMPTION

For each SWC-IMPLEMENTATION in the target model, one RESOURCE-CONSUMPTION sub element is set up. Its SHORT-NAME is the SHORT-NAME of the SWC-IMPLEMENTATION plus the suffix `_resourceConsumption`.

Configuration of target model SWC-BSW-MAPPING-REF

If a SWC-BSW-MAPPING to which the BSW-IMPLEMENTATION is related, see [Section 8.1.3.5.13, “SWC-BSW-MAPPING”](#), then this SWC-BSW-MAPPING is referenced in SWC-BSW-MAPPING-REF.

Upgrade of source model parameter PROGRAMMING-LANGUAGE

If the source model parameter PROGRAMMING-LANGUAGE is not defined, C is configured as default value in the target model.

Upgrade of source model parameter VENDOR-ID

If the source model parameter VENDOR-ID is not defined, 1 is configured as default value in the target model.

8.1.3.5.13. SWC-BSW-MAPPING

If the source model contains an INTERNAL-BEHAVIOR aggregating at least one RUNNABLE-ENTITY which references a BSW-MODULE-ENTITY/BSW-SCHEDULABLE-ENTITY/BSW-INTERRUPT-ENTITY via BSW-ENTITY-REF, a SWC-BSW-MAPPING element is created in the target model.

Its SHORT-NAME is SwcBswMapping_<SWCBhShortName>_<BSWBhShortName>. <SWCBhShortName> is the SHORT-NAME of the INTERNAL-BEHAVIOR aggregating the source model RUNNABLE-ENTITY and <BSWBhShortName> is the SHORT-NAME of the BSW-BEHAVIOR aggregating the source model BSW-MODULE-ENTITY/BSW-SCHEDULABLE-ENTITY/BSW-INTERRUPT-ENTITY.

BSW-BEHAVIOR-REF references the target model counterpart of the source model BSW-BEHAVIOR, SWC-BEHAVIOR-REF references the target model counterpart of the source model INTERNAL-BEHAVIOR.

For each RUNNABLE-ENTITY referencing a BSW-MODULE-ENTITY/BSW-SCHEDULABLE-ENTITY/BSW-INTERRUPT-ENTITY, one RUNNABLE-MAPPINGS/SWC-BSW-RUNNABLE-MAPPING parameter is set up. Its BSW-ENTITY-REF references the target model counterpart of the source model BSW-MODULE-ENTITY/BSW-SCHEDULABLE-ENTITY/BSW-INTERRUPT-ENTITY, SWC-RUNNABLE-REF references the target model counterpart of the source model RUNNABLE-ENTITY.

8.1.4. AUTOSAR 4.0 platform data types

During a model upgrade run, the AUTOSAR 4.0 *platform* datatypes are created according to the table below. Each line corresponds to one IMPLEMENTATION-DATA-TYPE element referencing one SW-BASE-TYPE element via SW-DATA-DEF-PROPS/W-DATA-DEF-PROPS-VARIANTS/SW-DATA-DEF-PROPS-CONDITIONAL/BASE-TYPE-REF and one DATA-CONSTR element via SW-DATA-DEF-PROPS/W-DATA-DEF-PROPS-VARIANTS/SW-DATA-DEF-PROPS-CONDITIONAL/DATA-CONSTR-REF. The IMPLEMENTATION-DATA-TYPE are created in the AR-PACKAGE path /AUTOSAR_Platform/Implementation-Datatypes, the SW-BASE-TYPE are situated in /AUTOSAR_Platform/BaseTypes, the DATA-CONSTR are created in /AUTOSAR_Platform/DataConstrs. If UPPER-LIMITS and/or LOWER-LIMITS are available, they are considered "closed" limits, i.e. the limit is included by the value range of the data type. SHORT-NAME refers to the SHORT-NAME parameters of the created IMPLEMENTATION-DATA-TYPE, SW-BASE-TYPE, and DATA-CONSTR elements. LOWER-LIMIT and UPPER-LIMIT refer to the respective DATA-CONSTR parameters. BASE-TYPE-ENCODING and BASE-TYPE-SIZE refer to the respective parameters in SW-BASE-TYPE.

For the IMPLEMENTATION-DATA-TYPE boolean, an IMPLEMENTATION-DATA-TYPE reference to uint8 is set up, therefore no SW-BASE-TYPE and DATA-CONSTR are required.

For the IMPLEMENTATION-DATA-TYPES float32 and float64, no dedicated limits are provided, therefore no dedicated DATA-CONSTR entries exist for them.

SHORT-NAME	LOWER-LIMIT	UPPER-LIMIT	BASE-TYPE-ENCODING	BASE-TYPE-SIZE
boolean	n/a	n/a	n/a	n/a

SHORT-NAME	LOWER-LIMIT	UPPER-LIMIT	BASE-TYPE-ENCODING	BASE-TYPE-SIZE
uint8	0	255	NONE	8
uint16	0	65535	NONE	16
uint32	0	4294967295	NONE	32
sint8	-128	127	2C	8
sint16	-32768	32767	2C	16
sint32	-2147483648	2147483647	2C	32
float32	n/a	n/a	IEEE754	32
float64	n/a	n/a	IEEE754	64

If the AUTOSAR 4.0 platform data types are configured during the upgrade of an AUTOSAR 3.2 model into a 4.0.3 target model, every platform data type additionally has its TYPE-EMITTER parameter value set to BSW.

Bibliography

- [1] *AUTOSAR Layered Software Architecture*. Version 2.1.0, Revision 0014. Publisher: AUTOSAR.
- [2] *OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection*. Author: Zimmermann. Pages: 425 to 432.
- [3] *AUTOSAR - Specification of ECU Configuration*. Document number 087. Document version 2.0.1, Revision 0002, Part of release 3.0. Publisher: AUTOSAR.
- [4] *AUTOSAR - Specification of ECU Configuration*. Document number 087. Document version 3.2.0, Revision 3, Part of release 4.0. Publisher: AUTOSAR.
- [5] *AUTOSAR Model Persistence Rules for XML*. Version 2.1.2, Release 3.0, Revision 0001. Publisher: AUTOSAR.
- [6] *LIN Specification Package*. URL: <http://www.lin-subbus.org/>. Publisher: Lin Consortium.

Appendix A. Third party license

This product includes third party components that require the following notices. For respective license terms refer to the subfolder /licenses

- ▶ This product includes Java(TM) 2 Runtime Environment 1.6.0_24.

Java(TM) 2 Runtime Environment 1.6.0_24 is licensed under the Oracle Corporation Binary Code License Agreement. For additional Licenses Terms according to Section G of the Oracle Corporation Binary Code License Agreement please refer to "jre/THIRDPARTYLICENSEREADME.txt".

Use of the Commercial Features for any commercial or production purpose requires a separate license from Oracle. "Commercial Features" means those features identified Table 1-1 (Commercial Features In Java SE Product Editions) of the Java SE documentation accessible at <http://www.oracle.com/technetwork/java/javase/documentation/index.html>

- ▶ This product includes Java(TM) 2 Runtime Environment 1.6.0_35.

Java(TM) 2 Runtime Environment 1.6.0_35 is licensed under the Oracle Corporation Binary Code License Agreement. For additional Licenses Terms according to Section G of the Oracle Corporation Binary Code License Agreement please refer to "jre/THIRDPARTYLICENSEREADME.txt".

- ▶ The product is based in part on software developed for eclipse.org licensed under the EPL 1.0. (Eclipse runtime environment 3.8.0)

Source code for these components is available from <http://subclipse.tigris.org/> and www.eclipse.org, under the Eclipse Public License ("EPL", see <http://www.eclipse.org/org/documents/epl-v10.php>).

As a requirement of the EPL, Elektrobit hereby:

(1) disclaims on behalf of all Eclipse.org Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

(2) excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits; and

(3) states that any provisions which differ from the Eclipse Public License are offered by Elektrobit alone and not by any other party.

- ▶ The product is based in part on software developed for eclipse.org licensed under the EPL 1.0. (Eclipse Modeling Framework 2.8.0)

Source code for these components is available from <http://subclipse.tigris.org/> and www.eclipse.org, under the Eclipse Public License ("EPL", see <http://www.eclipse.org/org/documents/epl-v10.php>).

As a requirement of the EPL, Elektrobit hereby:

(1) disclaims on behalf of all Eclipse.org Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

(2) excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits; and

(3) states that any provisions which differ from the Eclipse Public License are offered by Elektrobit alone and not by any other party.

- ▶ This product contains a modified version of the GNU make version 3.80 (default shell is always set to "command.com") which is available from <http://www.gnu.org/software/make/>. This program may be freely copied and distributed under the terms of the GNU General Public License.

Please refer to <http://www.gnu.org/licenses/old-licenses/gpl-2.0.html> for details.

The source code for this program is available and you can order it from us by writing to us at 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 There will be a small charge to cover materials and postage.

- ▶ This product includes jaxen 1.1-beta-7.

Copyright 2003-2006 The Werken Company. All Rights Reserved.

- ▶ This product includes commons-jxpath.Apache Commons.

JXPath Copyright 2001-2008 The Apache Software Foundation.

This product includes software developed by The Apache Software Foundation (<http://www.apache.org/>).

- ▶ This product includes java cup 0.10k.

Copyright 1996 by Scott Hudson, Frank Flannery, C. Scott Ananian

- ▶ This product includes log4j 1.2.7.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

Copyright (C) The Apache Software Foundation. All rights reserved.

- ▶ This product includes software developed by the JDOM Project (<http://www.jdom.org/>).

Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin. All rights reserved.

- ▶ This product includes poi 2.5.1.

This product includes software developed by The Apache Software Foundation (<http://www.apache.org/>).

- ▶ This product includes sqlite-jdbc 3.6.19.

sqlite-jdbc Copyright 2013 The Apache Software Foundation.

This product includes software developed by The Apache Software Foundation (<http://www.apache.org/>).

- ▶ This product includes Xerces-J 2.8.0.

This product includes software developed by The Apache Software Foundation (<http://www.apache.org/>).

Apache Xerces Java, Copyright 1999-2007 The Apache Software Foundation

Portions of this software were originally based on the following:

- software copyright (c) 1999, IBM Corporation., <http://www.ibm.com>.
- software copyright (c) 1999, Sun Microsystems., <http://www.sun.com>.
- voluntary contributions made by Paul Eng on behalf of the Apache Software Foundation that were originally developed at iClick, Inc., software copyright (c) 1999.

- ▶ This product includes Xerces 3.1.1

This product includes software developed by The Apache Software Foundation (<http://www.apache.org/>).

Portions of this software were originally based on the following:

- software copyright (c) 1999, IBM Corporation., <http://www.ibm.com>.

- ▶ This product includes trang version 20091111.

Copyright (c) 2002, 2003, 2008 Thai Open Source Software Center Ltd

- ▶ The product is based in part on software developed for eclipse.org licensed under the EPL 1.0. (subclipse 1.8.18)

Source code for these components is available from <http://subclipse.tigris.org/> and www.eclipse.org, under the Eclipse Public License ("EPL", see <http://www.eclipse.org/org/documents/epl-v10.php>).

As a requirement of the EPL, Elektrobit hereby:

(1) disclaims on behalf of all Eclipse.org Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

(2) excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits; and

(3) states that any provisions which differ from the Eclipse Public License are offered by Elektrobit alone and not by any other party.

- ▶ The product is based in part on software developed for eclipse.org licensed under the EPL 1.0. (collabnet merge 2.1)

Source code for these components is available from <http://subclipse.tigris.org/> and www.eclipse.org, under the Eclipse Public License ("EPL", see <http://www.eclipse.org/org/documents/epl-v10.php>).

As a requirement of the EPL, Elektrobit hereby:

(1) disclaims on behalf of all Eclipse.org Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

(2) excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits; and

(3) states that any provisions which differ from the Eclipse Public License are offered by Elektrobit alone and not by any other party.

- ▶ This product includes javaHL 1.7.8.

This product includes software developed by CollabNet (<http://www.Collab.Net/>). Copyright (c) 2000-2007 CollabNet. All rights reserved.

- ▶ This product includes software developed by Computing Services at Carnegie Mellon University (<http://www.cmu.edu/computing/>).

- ▶ This product includes jsch 0.1.

Copyright (c) 2002-2012 Atsuhiko Yamanaka, JCraft, Inc.

- ▶ This product includes antlr 3.5.

Copyright (c) 2013 Terence Parr All rights reserved.

Index

A

about EB tresos Studio, 70
 autoconfiguration wizards (see unattended wizards)
 AUTOSAR format
 module configuration, 45
 system description, 46

B

building
 projects, 299
 Bundle Registry, 127

C

CAN network, 348
 collaboration (see team collaboration)
 command line interface, 376
 command shell, 376
 compare editor, 228
 configuration
 configuration class
 dependency to AUTOSAR exporter, 319
 dependency to AUTOSAR importer, 317
 enabling/disabling parameters, 105
 setting in your project, 151
 verifying the, 296
 what is this?, 43
 configuration time
 dependency to code generators, 298
 enabling/disabling parameters, 105
 verifying the, 296
 what is this?, 43
 configuration variant
 setting the, 270
 what is this?, 43
 configuration and generation tool
 EB tresos Studio, 37
 configuration data
 export, 314
 import, 314

content type, 44, 314

CVS
 version control system
 working with CVS in EB tresos Studio, 198

D

DAD files, 345
 datamodel, 229
 DBC files, 345
 DBC format, 47
 DBC Importer, 339
 attaching attributes, 338
 attributes, 338
 I-Pdu Multiplexer, 354
 messages, 338
 multiplexed messages, 338
 NM messages, 338
 signals, 338
 DBC parser
 grammar, 350, 350
 differ, 228

E

EB tresos Details dialog, 123
 EB tresos Studio verifier, 94
 ECU Configuration Editor, 267
 Bulk Change dialog, 278
 opening the, 269
 ECU Extract
 importing, 320
 editing parameter values
 unattended wizards, 280
 Editor view, 84, 86, 268
 Bulk Change dialog, 278
 overview, 81
 page name, 81
 Search dialog, 289
 tables
 changing rows simultaneously, 278
 tabs, 82
 unattended wizards, 280
 configuring the, 280

- Element Description view, 89
- Element Information view
- parameter meta information, 87
- Element Outline view, 268
- Error Log view, 90, 268
- filtering messages, 92
 - log entries, 92
- export
- AUTOSAR, 314
 - system descriptions (Sys-D), 332
- extending
- EB tresos Studio
 - with your own features, 145
- F**
- features (see extending)
- Fibex format, 48
- Fibex Importer, 364
- file type, 44
- FlexLM licenses, 139
- floating network licenses, 139
- functional safety
- suitability, 34
- G**
- generating code
- of your project, 296
- generating source code
- from projects, 298
- graphical user interface, 54
- GUI, 54
- configuring a module, 269
- ECU Configuration Editor, 269
- Editor view
- comments, 271
- Element Description view
- parameter, 89
- Element Information view
- parameter meta information, 87
- Generic Configuration Editor, 103
- choices, 110
 - optional elements, 114
- references, 108
- tables, 111
- H**
- Help menu, 70
- I**
- import
- AUTOSAR, 314
 - importing system descriptions (Sys-D)
 - adding a System Description Importer, 321
- L**
- LDF files, 359
- LDF format, 48, 359
- LDF Importer, 359, 359
- licenses, 139
- LIN configuration, 359
- linking
- external folders, 145
- log files, 42
- M**
- Module Registry, 126
- N**
- network licenses, 139
- Node Outline view, 268
- appearance of displayed items, 86
 - editor contents, 84
 - filtering displayed items, 84
- O**
- OIL files, 335
- OIL format, 335
- OIL Importer, 335
- Os configurations, 335, 335
- Os module, 335, 335
- P**
- parameter, 89
- (see also Element Description view)

- parameter meta information, 87
 (see also Element Information view)
- Plugin Registry, 127
- Problems view, 94, 268
 filtering messages, 97
- project
 building a, 299
 convert to newer version, 263
 delete, 172
 export, 305
 generate code, 296
 generating source code, 298
 import, 301
 verifying, 296
- project configuration data
 exporting, 308
 importing, 308
- Project Explorer
 hiding layers, 74
- Project Explorer view, 170, 171, 171, 172, 268
- project properties, 176
 creation of child elements, 181
 generation path, 179
 module configurations, 182
- Project Wizard, 146
- proprietary format, 350
- ## S
- safe use of
 EB tresos Studio, 34
- Search
 dialog, 289
- searching for parameters
 Editor view, 289
- source code
 building from projects, 299
 generating from projects, 298
- storing projects as compressed archive, 307
- SVN
 version control system
 working with SVN in EB tresos Studio, 208
- synchronizing Project Explorer with editor, 80
- Sys-D format (see system description)
- system description
 exporting, 331
 adding a System Description Exporter, 332
 importing, 320
- ## T
- TDB Importer, 370
- team collaboration, 228
 using, 198
- Third party license, 632
- tresosDB format, 47
- ## U
- unattended wizards, 280
 Calculate Derivable Values
 configuring the, 287
 Calculate Handle IDs
 configuring the, 288
 configuring the, 280
- Unix line endings, 179
- use of OIL Importer, 338
- ## V
- variant resolving, 432
- vendor-specific module definition, 409, 415, 417
- verifying
 project, 296
- version control system, 198
- ## W
- workspace, 42
- ## X
- XDM files, 228