Elektrobit

EB tresos® AutoCore Generic 8 ServiceApplTemplate documentation

Module release 1.3.7

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

Technical support

https://www.elektrobit.com/support

Legal disclaimer

Confidential information.

Table of Contents

1. Overview

Welcome to the ServiceApplTemplate product release notes and documentation.

This document provides:

- Chapter 2, “ServiceApplTemplate module release notes“: details of changes and new features in the current release
- Chapter 3, “Service Application Template Generator user's guide“: concept information and configuration instructions
- Chapter 4, “ServiceApplTemplate module references“: configuration parameters and the application programming interface

2. ServiceApplTemplate module release notes

- Module version: 1.3.7.B567464
- Supplier: Elektrobit Automotive GmbH

2.1. Change log

This chapter lists the changes between different versions.

Module version 1.3.7

2021-10-21

Module version 1.3.6

2021-01-28

- Incase of duplicate AR-PACKAGEs shortNames, generation of ServiceSwComponentTemplate file split to two files such as, - ServiceSwComponentTemplate_0.arxml :Contains infromation about Software component and its composition. - ServiceSwComponentTemplate_0_System.arxml :Contains infromation about System Otherwise generated ServiceSwComponentTemplate, contains both System and composition information in same file.

Module version 1.3.5

2020-09-24

- Support for mirroring of P-PortPrototypes implemented. Also provided an option to user to choose, generation of PortPorotypes as either P-PortPrototypes/RPortPrototypes/ChooseBoth/None for each ServiceSoftwareComponent.

Module version 1.3.4

2020-08-28

Module version 1.3.3

2020-08-25

Module version 1.3.2

2020-04-02

Module version 1.3.1

2019-11-28

Module version 1.3.0

2019-07-23

- ShortNames in generated ARXML no longer can exceed AUTOSAR's maximum ShortName length.

Module version 1.2.0

2019-06-18

- Fix for making generation of connectors optional. To achieve this, provided configuration option to include or not the AssemblyConenctors in generated ServiceApplTemplate.arxml file.
- Remove fallback output on code generation erros. Use case for it has become obsolete with the Generate Code wizard and wildcard support in system description importer.

Module version 1.1.5

2019-05-28

Module version 1.1.4

2019-04-08

- Fix invalid output ARXML when no PortPrototypes need to be mirrored. Previously, in that case, SwcInternalBehaviour.runnable list was empty which had violated its multiplicity constraint, leading to an error in Rte Editor.

Module version 1.1.3

2019-02-15

- Fix in user's guide: The generated software component contains PPortPrototypes for the service software component's RPortPrototypes.

Module version 1.1.2

2018-12-20

Module version 1.1.1

2018-06-27

- Create MemorySection in output ARXML. This prevents errors with MemMap from ACG >= 8.5.1.

Module version 1.1.0

2018-04-27

- Add support for creating a software component template for any service SwComponentPrototype. The module is renamed from DiagApplTemplate to ServiceApplTemplate.

Module version 1.0.4

2018-02-27

- Fix NoSuchElementException during generation mode `generate_swcd` when Dcm's SwComponentPrototype is located in RootSwCompositionPrototype's CompositionSwComponentType. (regression introduced in previous release)

Module version 1.0.3

2018-02-16

- Fix for handling Dcm SwComponentPrototypes not in top level CompositionSwComponentType. Previously, in this case Dcm PortPrototypes which were already connected were not recognized correctly, leading to additional counter parts in generated SWC being created.

Module version 1.0.2

2017-12-22

- Removed assumption on location of Dcm SwComponentType's prototype: The prototype previously was assumed to be located in the top level/root CompositionSwComponentType. It can now also be located anywhere in the tree of CompositionSwComponentTypes.

Module version 1.0.1

2017-09-14

- Fix error reporting: An Error Log entry is created now when fallback ARXML is generated. Previously, errors had gone lost.
- Fix XML validation problem for fallback ARXML in case of generation error: Add missing XML preamble.
- Fix XML validation problem for fallback ARXML in case of generation error: Remove superfluos characters at beginning of file.
- Improve documentation of and error message for prerequisite that a Dcm SwComponentPrototype must exist.
- Add documentation of limitations.
- Minor documentation improvements.

Module version 1.0.0

2017-05-17

- Initial release

2.2. New features

See change log.

2.3. Elektrobit-specific enhancements

This module is not part of the AUTOSAR specification.

2.4. Deviations

This module is not part of the AUTOSAR specification.

2.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

- Module generation exits with success status even on errors

  Description:

  When an error during code generation for the module occurs, the error is reported in *Error Log* view but not in *Code Generator Run Finished* dialog; module generation always reports succes status.

  Rationale:

  EB tresos Studio importers may include the generated ARXML file. When the ARXML file is deleted some time later during integration and the system model is not valid, the user can neither run the importer successfully (error due to missing ARXML file) nor generate the module successfully (invalid system model). In order to prevent this, the module always generates an ARXML that can be imported. Due to implementation limitations, in this case the error can only be reported in the *Error Log* view but not in the in *Code Generator Run Finished* dialog.

2.6. Open-source software

Open-source software information is not available for this module.

3. Service Application Template Generator user's guide

3.1. Overview

Service Application Template Generator (ServiceApplTemplate) creates an AUTOSAR software component description template suited for connection to a project's service software components (e. g. that of Dcm). Its code generator (mode `generate_swcd`)

- scans the existing connections between software components and a service SwComponentPrototype.

  Note: If a SwComponentPrototype for a basic software module does not exist yet, you can add one for an existing SwComponentType by using EB tresos Studio's *Connection Editor*. Please refer to EB tresos Studio documentation user's guide section 8.2 *Using the Connection Editor*, especially section 8.2.4.4 *Adding a component prototype*.
- creates a software component (type and prototype) which has matching ports for all unconnected ports of the service SwComponentPrototype
- creates PortPrototypes of type ClientServerInterface and SenderReceiverInterface based on the user selection
- creates AssemblyConnectors between the generated SwComponentPrototype and the service SwComponentPrototype

3.2. Configuration and software component description generation

In order to run the ServiceApplTemplate generator, the following steps are necessary:

1. Configure the AUTOSAR basic software module for whose service software component prototype you want to create a template according to the project's needs.
2. Connect the service software component's ports for which you already have destination software component ports in your project. This can be done using EB tresos Studio's *Connection Editor* or by importing AUTOSAR system description files containing that information.
3. Adapt ServiceApplTemplate's module configuration to your needs. Different PortPrototypes can be selected based on the interface types. The attribute Period lets the user to configure the cycle time for Timing Event.

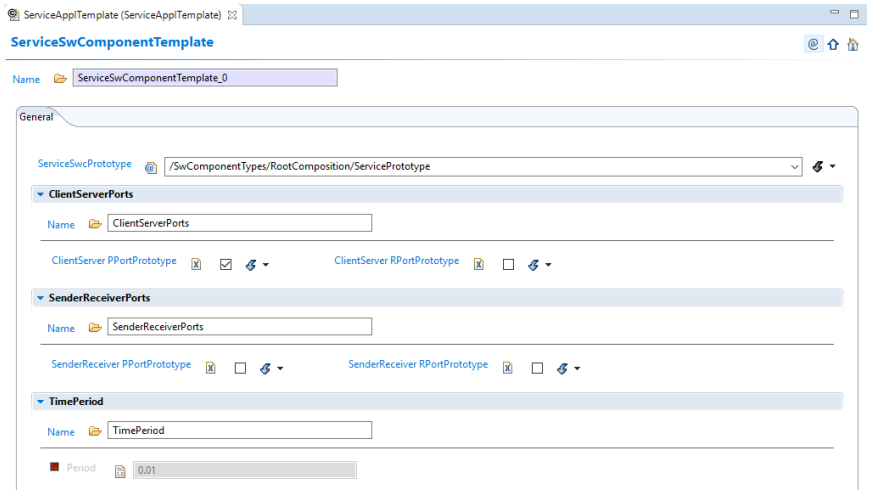

Figure 3.1. Selecting ports to be generated

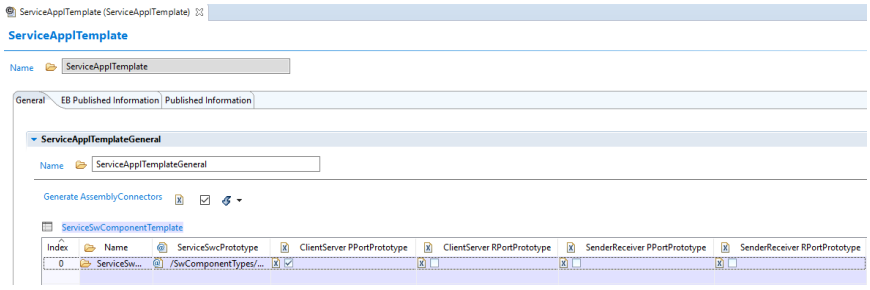

Figure 3.2. Selecting ports and configuring Period

4. Run the module's code generator in mode `generate_swcd` (Figure 3.3, “Running ServiceApplTemplate generator.”). This creates a software component description ARXML file.

5. Add the generated file to your project's software component description importer and run it. If you only want to use the generated file as an ARXML fragment template for your own software component, copy the relevant parts to yours and add that to the importer.

By using the Rte module's code generator, you can also obtain a template C file in folder `output/generated/src_appl`.

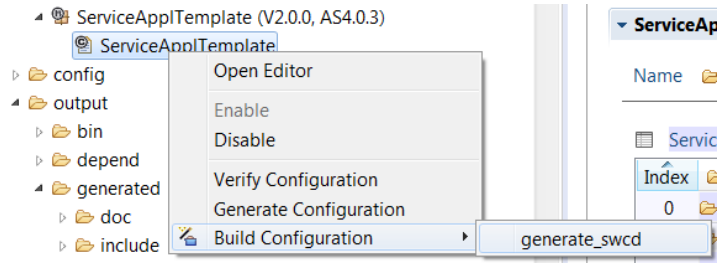

Figure 3.3. Running ServiceApplTemplate generator.

3.3. Updating ServiceApplTemplate's template after changes to a service software component

In case one of the service software components gets a new port (e. g. for Dcm after configuration of a new diagnostic service), you can update your generated software component template as follows:

1. Run your service module's software component description generator (generator mode `generate_swcd` when EB tresos Studio is used).
2. Import the generated ARXML service software component description file.
3. Run ServiceApplTemplate's software component description generator.
4. Import ServiceApplTemplate's generated software component description file.

3.4. Third party license information

This product includes software developed by the Ant-Contrib project (http://sourceforge.net/projects/ant-contrib). The full license information can be found in the plugin's installation directory.

4. ServiceApplTemplate module references

ServiceApplTemplate configuration parameter reference is not available.

ServiceApplTemplate API reference is not available.

4.1. Integration notes

4.1.1.1. ServiceApplTemplate dependencies

The usage of the module depends on the following other modules

- OemApiLib

4.1.1.2. WorkFlow Requirement

Execute ECUExtractCreation wizard before ServiceApplTemplate