



Elektrobit

EB tresos[®] AutoCore Generic 8

Base documentation

release notes update for the Base module

product release 8.5.1



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

Europe

Phone: +49 9131 7701 6060

Japan

Phone: +81 3 5577 6110

USA

Phone: +1 888 346 3813

Support URL

<https://www.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.

Copyright 2018, Elektrobit Automotive GmbH.



Table of Contents

- 1. Overview 4
- 2. Base module release notes 5
 - 2.1. Change log 5
 - 2.2. New features 10
 - 2.3. EB-specific enhancements 11
 - 2.4. Deviations 11
 - 2.5. Limitations 12



1. Overview

This document provides you with the release notes to accompany an update to the `Base` module. Refer to the changelog [Section 2.1, “Change log”](#) for details of changes made for this update.

2. Base module release notes

- ▶ Module version: 5.0.23.B216231
- ▶ Supplier: Elektrobit Automotive GmbH

2.1. Change log

This chapter lists the changes between different versions.

Module version 5.0.23

2018-09-27

- ▶ ASCBASE-1988 Fixed known issue: Added include of header file `TS_Compiler.h` (that defines macro `TS_CAST`) to header file `ComStack_Helpers.h`

Module version 5.0.22

2018-07-11

- ▶ Added `uint32` to supported range of `PduLengthType`
- ▶ Added header file `Base_Version.h` for the version number of the Base plugin
- ▶ Added support for core scope option of MemMap to Base SWCD by introducing additional software addressing methods (`SwAddrMethod`)

Module version 5.0.21

2018-03-07

- ▶ Removed bundles `BaseSelectableWizardAsc`, `BaseXGenerator`, `BaseXPathAsc`, and `SW Description Updater` which are now released by the plugin `ToolExtensions`
- ▶ Removed macros `TS_AtomicSetBit` and `TS_AtomicClearBit` as they are now provided by the plugin `Atomics`
- ▶ ASCBASE-1939 Fixed known issue: Fixed generation error for non-EB complex device drivers
- ▶ Added support for providing custom memory copy function and overriding default implementation

Module version 5.0.20

2017-10-18

- ▶ Added parameters for mapping of AUTOSAR base types to C data types to the Base module configuration to be used by the `typedef` definitions in the generated `Platform_Types.h`
- ▶ Added support for non-EB modules (without `_TS_` infix) in generator `Base_Modules.h` for generation of module specific enable macros
- ▶ ASCBASE-1892 Fixed known issue: Added software addressing method (`SwAddrMethod`) for `VAR_NO_INIT` as specified by AUTOSAR 4.0.3

Module version 5.0.19

2017-04-19

- ▶ Added datatype definitions for `EthTrcv_PhyLoopbackModeType`, `EthTrcv_PhyTxModeType`, `EthTrcv_PhyTestModeType` and `EthTrcv_CableDiagResultType` to `Eth_GeneralTypes.h` according to AUTOSAR 4.3.0
- ▶ Added return value definitions for `E_NO_DATA` to `TransformerTypes.h` according to AUTOSAR 4.3.0
- ▶ Added datatype definitions for `EthSwt_PortMirrorCfgType` and `EthSwt_PortMirrorStateType` to `Eth_GeneralTypes.h` according to AUTOSAR 4.3.0
- ▶ Added macro `TS_CAST` to suppress excessive warnings from some compilers while keeping the code readable and efficient for well-behaving compilers
- ▶ ASCBASE-1893 Fixed known issue: Fixed XPath functions for size and alignment of base types to return correct values for 64bit data types

Module version 5.0.18

2016-11-23

- ▶ Added generator mode `export_asr_config` for writing ARXML files with ecu parameter configuration for each module
- ▶ Added application programming interface (API) to module references
- ▶ Generate definition of data type `Eth_BufIdxType` dependent on `EthIf` config parameter
- ▶ ASCBASE-1857 Fixed known issue: Fixed value of lower limit and upper limit for AUTOSAR 64bit base types (`uint64`, `sint64`) in the Base SWCD
- ▶ ASCBASE-1845 Fixed known issue: Added support for module generation output directories to XGenerator plugin

Module version 5.0.17

2016-06-15

- ▶ Added definition of `TS_STATIC_ASSERT` to `TSAutosar.h`
- ▶ Added tooling to calculate checksums on installed plugins and verify against provided checksum

Module version 5.0.16

2016-02-24

- ▶ Added support for Debug & Trace with custom header file configurable via parameter `BaseDbgHeaderFile`
- ▶ Added definition of `Can_HwType` to `Can_GeneralTypes.h`

Module version 5.0.15

2015-11-25

- ▶ Changed the generator template `Base_Modules.h` to appropriately support complex driver and transformer modules
- ▶ Added support for 64bit CPUs

Module version 5.0.14

2015-08-07

- ▶ Provided type definitions for `VoidPtr` and `ConstVoidPtr`

Module version 5.0.13

2015-07-08

- ▶ Added error codes for E2E Transformer based on proposal from RfC #67553
- ▶ Provided configuration parameter to enable or disable the generation of AUTOSAR 64bit base types (`uint64`, `sint64`)

Module version 5.0.12

2015-03-11

- ▶ Provided configuration parameters for post-build-selectable variants (intermediate solution)
- ▶ Added support for AUTOSAR base types `uint64` and `sint64`
- ▶ Removed adding of Service Component Prototypes to EcuExtract by SwdUpdater as the prototypes shall be added manually by the user in the composition editor and in the hierarchical model

Module version 5.0.11

2015-01-21

- ▶ Added configuration scheme to enable custom include files to be included via `Platform_Types.h`
- ▶ Support of custom defined `Std_VersionInfoType` via a configuration macro (`STD_VERSION_INFO_TYPE_DEFINED`)
- ▶ ASCBASE-1742 Fixed known issue: Changed the generator template `Base_Modules.h` to appropriately support modules that allow multiple instantiation

Module version 5.0.10

2014-10-02

- ▶ ASCBASE-1683 Fixed known issue: Binary Code generation uses wrong pointer type for non-relocatable configuration
- ▶ ASCBASE-1701 Fixed known issue: Changed the merge algorithm invoked by SwdUpdater in EB tresos Studio to generate a valid service component description
- ▶ Added getting of alignment and size also for derived data types to custom XPath functions

Module version 5.0.9

2014-04-25

- ▶ Added defines for module header file names respecting `VendorId` and `VendorApiInfix` from `CommonPublishedInformation` in `Base_Modules.h`

Module version 5.0.8

2014-02-04

- ▶ ASCBASE-1656 Fixed known issue: Binary Code generation does not support big-endian platforms
- ▶ ASCBASE-1662 Fixed known issue: Binary Code generation does not support relocatable configuration for address offsets different from 0

- ▶ Changed adaptation of ComM's initialization config pointers (`BASE_COMM_CONFIG_PTR`) to support both post-build capable and non post-build capable ComM variants
- ▶ Improved the XGenerator to handle nested structures
- ▶ Added support for signed data types for binary code generation
- ▶ Improved sorting of module specific macros in `Base_Modules.h` in alphabetical order
- ▶ Added support for non-AUTOSAR data types in arrays of post build configurations

Module version 5.0.7

2013-10-21

- ▶ Added explicit cast to `TS_AtomicSet/ClearBit()` in order to prevent compile warnings/errors in case `TS_AtomicSet/ClearBit_x()` are implemented as a functions
- ▶ Added BSW Module Description for Base (including memory section for CODE)
- ▶ ASCBASE-1438 Fixed known issue: Changed the macros to set or get a bit to fix the undefined behavior on 16bit platforms when the index of the bit to get or set is larger than 15
- ▶ Improved the handling of disabled modules in the configuration project, so that these are also disabled in the header file `Base_Modules.h` by defining the macro `BASE_{module}_ENABLED` as `STD_OFF`
- ▶ Implemented support for the AutoCore binary code generation to create a post-generation step to convert XGEN files

Module version 5.0.6

2013-06-14

- ▶ Added developer's guide for XPath functions
- ▶ Changed memory routines to use 32-bit parameter for length parameter
- ▶ Added macro definition checks and optimized structure type definition in header file `Fr_GeneralTypes.h`
- ▶ Added macro to check if a bit set in a variable (without returning its value)

Module version 5.0.5

2013-02-11

- ▶ Added datatype definitions for `CanTrcv_PNActivationType` and `CanTrcv_TrcvFlagStateType` to `Can_GeneralTypes.h`
- ▶ Provided version numbers and module IDs of Platform Types, Compiler Abstraction, Standard Types, and Communication Stack Types

Module version 5.0.4

2012-10-12

- ▶ Adapted top level structure for model elements for which the AUTOSAR specification already defines a standardized name. The package structure has been changed from `/AUTOSAR/{module}` to `/AUTOSAR_{module}`

For backward compatibility the old top level structure (`/AUTOSAR/{module}`) is still supported, but considered deprecated and will be removed in future releases

Module version 5.0.3

2012-06-15

- ▶ Added configuration parameter index macros (`FR_CIDX_*`) in `Fr_GeneralTypes.h` according to AUTOSAR R4.0r3
- ▶ Removed type definition `Fr_configType` from `Fr_GeneralTypes.h` according to AUTOSAR R4.0r3 FlexRay Driver SWS
- ▶ Added the SW Description Updater to generate and import the service software component descriptions and basic software module descriptions

Module version 5.0.2

2011-09-30

- ▶ Updated `Fr_POCSStatusType`, `Fr_RxLPduStatusType`, `Fr_SlotModeType`, `Fr_StartupStateType` and `FrTrcv_TrcvWUReasonType` in `Fr_GeneralTypes.h` to AUTOSAR R4.0r2

Module version 5.0.1

2011-09-02

- ▶ Initial AUTOSAR 4.0 version

2.2. New features

- ▶ Base supports the range `uint16/uint32` of the AUTOSAR EcuC specification for the EcuC configuration parameter `PduLengthType`.

The mapping of AUTOSAR base types to C data types can be defined by parameters in the Base module configuration, container `BaseTypes`. The mapping is used to generate the corresponding `typedef` definitions in the generated header file `Platform_Types.h`.

If Platforms also provides mappings, the definitions from Platforms are used as default values for the corresponding parameters of Base. The mappings defined in Base overwrite the mappings of Platforms when `Platform_Types.h` is generated.

- ▶ Base supports 64bit CPUs by providing appropriate macros and functions, especially optimized memory operations (copy, move, compare, set) for 64bit types.
- ▶ The module contains a configuration scheme. The configuration scheme enables a user to configure custom include files which are included via `Platform_Types.h`.

This include mechanism can be used to provide types, macros and function declarations at a central location to all modules in a project.

- ▶ The provision of the type `Std_VersionInfoType` in the file `Std_Types.h` is controlled by a macro. If the macro `STD_VERSION_INFO_TYPE_DEFINED` is set to `STD_ON`, the user must provide a definition of the type `Std_VersionInfoType` in a custom include file.
- ▶ The default implementation of the memory copy function can be overridden by a custom implementation.
- ▶ Base supplies a layer of common operations that are not standardized in C90. Since they are required by several ComStack modules, Base provides the layer of abstraction with provisions to predefine them from Platforms or Compiler packages, effectively overriding them with a more targeted version using either non-standard compiler-known-functions, C or inline assembly. The default implementation is a portable default.

2.3. EB-specific enhancements

This module is not part of the AUTOSAR specification.

2.4. Deviations

This chapter lists the deviations of the module from the AUTOSAR standard.

- ▶ The type of `PduLengthType` that is defined in header file `ComStack_Cfg.h` is fixed to `uint16`.
Rationale:
Refer to deviations of EcuC.
- ▶ The type of `PduIdType` that is defined in header file `ComStack_Cfg.h` is fixed to `uint16`.

Rationale:

Refer to deviations of EcuC.

- ▶ The header file `Eth_GeneralTypes.h` does not include the header file `Eth_Cfg.h`.

Rationale:

`Eth_Cfg.h` contains the configuration specific parameters for the Ethernet controller driver and is provided by the Eth plugin. The inclusion of the header file introduces a dependency between Base and Eth which is not intended as Base is used by different other plugins.

- ▶ The type of `PduLengthType` is limited to `uint16` and `uint32`.

Rationale:

Refer to deviations of EcuC.

- ▶ The type of `PduIdType` is limited to `uint16`.

Rationale:

Refer to deviations of EcuC.

- ▶ The type of `PduInfoType` is not compliant to AUTOSAR 4.3.0.

Rationale:

No support of Metadata.

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.

- ▶ Dependency to header file `stddef.h` of C standard library

For the compilation of Base, the header file `stddef.h` provided by the C standard library is required. From this header file, only `offsetof()` is used which usually is implemented as macro (ISO-C:90).

Rationale:

`offsetof()` is used by the implementation of the post-build configuration to compute a structure member's offset from the beginning of a structure. The usage of `offsetof()` violates MISRA-C:2012 rule 1.3, but an own implementation for `offsetof()` would create other MISRA-C:2012 violations. As no library from the C standard library is required for linking, MISRA-C:2012 directive 1.1 is not applicable.