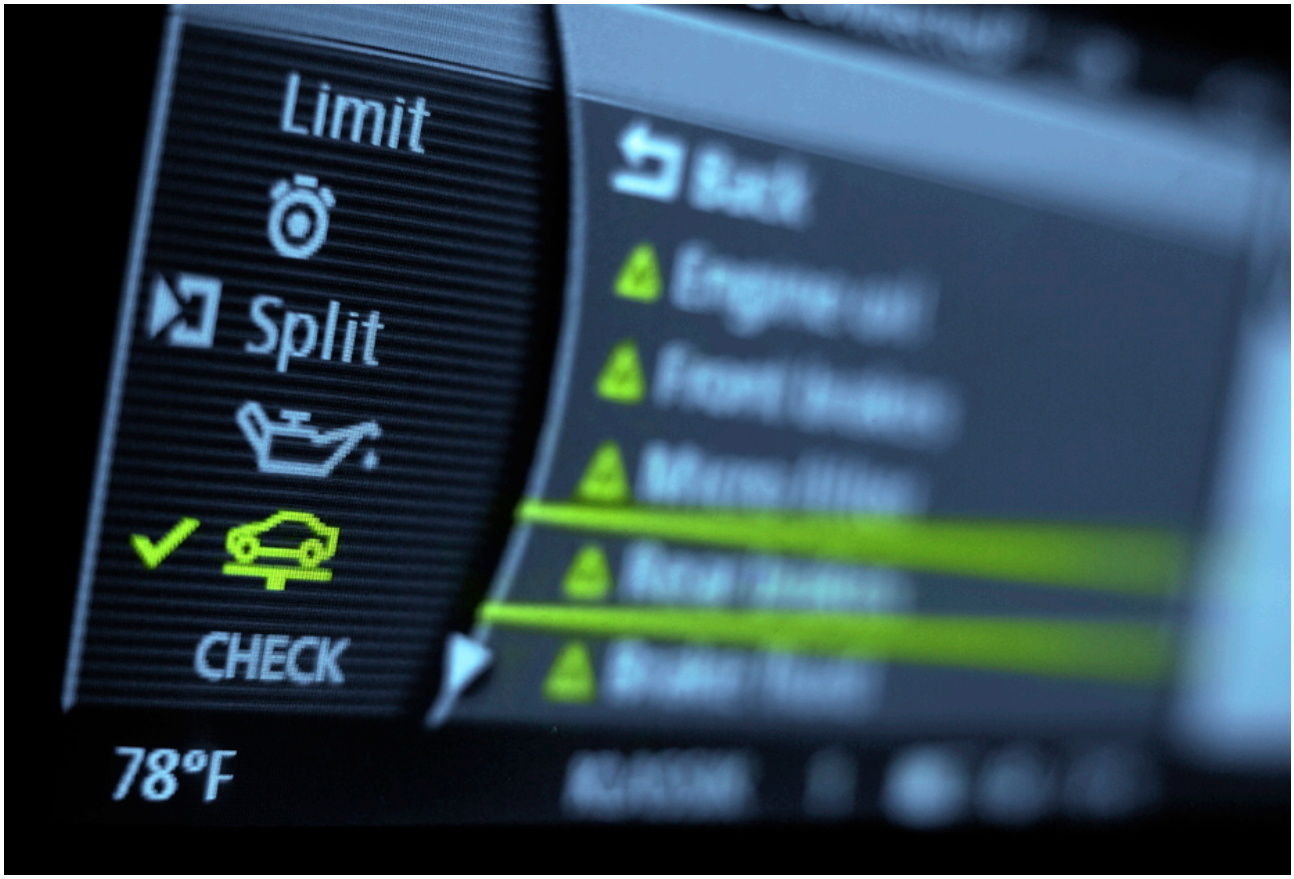
Elektrobit

EB tresos® AutoCore OS release notes
TRICORE TC38XQ

product release 6.0



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

Support URL

https://www.elektrobit.com/support

Legal notice

Table of Contents

1. Release Notes

Release:
EB tresos AutoCore OS 6.0.219 [revision 0000]

Date:
2019-02-19

Supported architecture:
TRICORE/TC38XQ

AUTOSAR version and revision:
4.0.3

AUTOSAR SWS version and revision:
5.0.0

Supplier:
Elektrobit Automotive GmbH

1.1. Supported Compilers & Hardware used for testing

For an updated list of supported compilers including supported compiler options and hardware used for testing, see the qualification report provided with the delivery.

1.2. New features

The following new features have been implemented since the last major or minor product version. See change log for further details.

- The module now implements the "Specification of Operating System V5.0.0" (AUTOSAR release R4.0 Revision 3).
- The module now provides *atomic functions*, which allow atomic and sequentially consistent memory access.

1.3. Migrating the Os module

This chapter describes how to migrate a module from one release to an other.

If you want to migrate the module through several releases, execute all migration steps between your oldest release and the latest release. To migrate, the oldest, the latest and all releases in between need to be installed on your system.

1.3.1. Migrating from EB tresos AutoCore OS 4.2 to EB tresos AutoCore OS 4.5

Import the old project. For details on how to import configuration data, see the `EB tresos Studio user's guide`.

1.3.2. Migrating from EB tresos AutoCore OS 4.1 or below to EB tresos AutoCore OS 4.2

To use the module, import the configuration as described in `EB tresos Studio user's guide`.

- If arrival rate monitoring has been used in EB tresos AutoCore OS 4.0 or below, the timestamp timer must now be configured to use this feature. The RateMonitorCounter reference has been removed.

1.3.3. Migrating from EB tresos AutoCore OS 3.1 to EB tresos AutoCore OS 4.0

Import the old project. For details on how to import configuration data, see the `EB tresos Studio user's guide`.

1.3.4. Migrating from EB tresos AutoCore OS 3.0 to EB tresos AutoCore OS 3.1

The module does not need to be migrated. Reason: the AUTOSAR release has not changed since the last EB tresos AutoCore release.

1.4. EB-specific enhancements

- Code Size and Execution Time Optimizations via

- Configuration-dependent source-level optimization
- Customizable Autosar support
- Use of a single hardware timer for all timing protection and timebase counters
- Optional small auto-incrementing software counter

- API functions for
  - Runtime measurement of tasks and ISRs
  - CPU-load measurement
- 32bit-wide hardware counters on every supported architecture
- ORTI support for debugging
- Available for various 32- and 16-bit MCU platforms

1.5. Deviations

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

- NextScheduleTable() does not continue synchronization.

  AUTOSAR 4.0 states that if OsScheduleTblSyncStrategy of the schedule tables ScheduleTableID_From and ScheduleTableID_To in a call of NextScheduleTable() is EXPLICIT and the Operating System module already synchronizes ScheduleTableID_From, NextScheduleTable() shall continue synchronization after the start of processing ScheduleTableID_To [OS505].

  The current implementation, however, follows AUTOSAR 3.1 where synchronization does not continue.
- API argument check for NULL pointers is implemented differently.

  Description:

  AUTOSAR 4.0 states that the Operating System API shall check in extended mode all pointer arguments for a NULL pointer and return OS_E_PARAM_POINTER if such an argument is NULL [OS566].

  In the current implementation, pointer arguments are checked for NULL pointers if and only if memory protection is used. The respective APIs return OS_E_ADDRESS in a case of a NULL pointer argument.
- Shutdown on restart from ProtectionHook() is not implemented.

  Description:

  If the ProtectionHook() returns with PRO_TERMINATEAPPL_RESTART, according to AUTOSAR 4.0 specification, ShutdownOS() shall be called if no OsRestartTask was configured.

In the current implementation, an unconfigured OsRestartTask will not result in a call to ShutdownOS(). Instead, the ProtectionHook() handles the return value PRO_TERMINATEAPPL_RESTART in the same way as a call to TerminateApplication() does.

- Protection hook return value restrictions for E_OS_PROTECTION_ARRIVAL not implemented.

  Description:

  If the ProtectionHook() is called with E_OS_PROTECTION_ARRIVAL, according to AUTOSAR 4.0 specification, the only valid return values shall be PRO_IGNORE or PRO_SHUTDOWN.

  In the current implementation this is not checked and other values will not result in a call to ShutdownOS(). Instead, handling of the ProtectionHook() return value follows the behavior specified in the AUTOSAR 3.-1 specification.

- GetActiveApplicationMode() not available as a Software Component service

  Description:

  The prototype of GetActiveApplicationMode follows the OSEK prototype

  ```
  AppModeType GetActiveApplicationMode(void);
  ```

  and therefore can not be called by the RTE. As a consequence, this API is not available as a service in the OS Software Component Description.

  Rationale:

  Functions called by the RTE need to have a return type of `Std_ReturnType` and to use pointer parameters for output. AUTOSAR bugzilla http://www.autosar.org/bugzilla/show_bug.cgi?id=51610 has been created to address this issue.

- Configuration of RES_SCHEDULER does not follow chapter 12.8 of the AUTOSAR SWS

  Description:

  Chapter 12.8 of the AUTOSAR SWS, "Migrating RES_SCHEDULER in AUTOSAR OS", gives hints about the configuration of the RES_SCHEDULER special resource. In contrast, AUTOSAR also specifies the OsUseResScheduler configuration parameter [OS049_Conf].

  To use RES_SCHEDULER, simply enable the OsUseResScheduler parameter in the configuration. The generator will then automatically add the RES_SCHEDULER resource.

  If you want to add a linked resource linking to RES_SCHEDULER:

  - Enable the OsUseResScheduler parameter.
  - Add a resource of type STANDARD with name RES_SCHEDULER. Task references to this resource will be ignored by the generator; it will be implicitly used by all tasks.

- Now RES_SCHEDULER can be referenced by linked resources.

- The IOC configuration follows AUTOSAR 4.0R1.

  Description:

  The IOC configuration parameters are implemented according to "Specification of Multi-Core OS Architecture, Version 1.0.0". This means that the following configuration requirements from "Specification of Operating System, Version 5.0.0" are not present: MCOS1023_Conf, MCOS1024_Conf, MCOS1035_Conf, MCOS1035_Conf.

- The configuration parameter "OsIocDataTypeRef" is not present.

  Description:

  To avoid links to another meta-model layer, this parameter has been removed. To configure the type of an IOC-channel, use the vendor-specific parameter "OsIocDataType".

- The configuration parameter "OsIocReceiverPullCB" is ignored

  Description:

  This implementation of the IOC does not support the callback feature.

- The names of the API functions of the IOC deviate from the AUTOSAR OS specification 4.0 rev 3.

  Description:

  The names of the API functions of the IOC deviate from the AUTOSAR OS specification 4.0 rev 3, instead they conform to the Multicore specification 0.3.0. This has no effect on RTE generated IOC configurations, because the IOC part of EB-RTE also expects this API naming.

- StartNonAutosarCore() not available

  Description:

  The API function StartNonAutosarCore() is not provided by the OS. AUTOSAR specifies this function as a way to start cores that are not controlled by the AUTOSAR OS [OS584] [OS682].

  Rationale:

  Starting cores that are not under the control of the OS depends on system-dependent information the OS can't deduct. If needed, this functionality has to be supplied by the system integrator; a generic implementation isn't possible.

- If the specified memory area straddles two or more defined access regions, CheckISRMemoryAccess() and CheckTaskMemoryAccess() return 0 (and additionally call the error hook if error reporting for non-StatusType services is enabled). The reasoning behind this that the service should normally be called to check the accessibility of a C variable, and a C variable by definition resides entirely within a defined access region. Furthermore, the SWS does not specify whether a memory area that sits across a stack boundary

should get the pseudo-access-right "STACK" or not. So calling the function for a "variable" that doesn't sit entirely inside one access region is an error and it doesn't seem logical to implement the complicated algorithm that would be necessary for these requirements simply to handle an error case.

- The System Services CheckObjectAccess() and CheckObjectOwnership() are defined in the SWS with an ellipsis parameter. Since this is not possible for a macro implementation and would in any case cause complications for system-call kernels by forcing the parameter onto the stack, the EB implementation defines an all-encompassing type for the single, fixed parameter that in any case is expected to be present in the ellipsis slot.
- [Autosar 2.1/3.0 OS Specification 2.1.0/3.0.0 Requirement OS083] The OS cannot permit non-trusted applications to write to ANY peripheral. Non-trusted applications run in the USER0 mode of the CPU, which does not permit any writes into the peripheral area. Even if USER1 mode were selected, the necessary fine-grained control would not be possible; it is not clear whether the memory protection registers operate in the peripheral areas, but even if they do there are not enough region registers to be able to provide the required control.

1.6. Change log

This chapter lists the changes between different versions. Only changes for the current architecture are listed. Note that a change for the current architecture may also be listed within the version entry for another architecture, if the release of this architecture coincided with the change.

EB tresos AutoCore OS version 6.0.219

TRICORE TC38XQ release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.218

ARM64 ZUXEV release

EB tresos AutoCore OS version 6.0.217

ARM S6J3200 release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.216

ARM RCARV3MCR7 release

EB tresos AutoCore OS version 6.0.215

ARM ZUXEVCR5 release

EB tresos AutoCore OS version 6.0.214

TRICORE TC39XX release

- ASCOS-5369 Fixed known issue: Error message "illegal constructor argument" when generating OS

EB tresos AutoCore OS version 6.0.213

RH850 RH850P1HC release

EB tresos AutoCore OS version 6.0.212

PA SPC58XH release

EB tresos AutoCore OS version 6.0.211

PA SPC584B release

EB tresos AutoCore OS version 6.0.210

ARM64 RCARV3H release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.209

RH850 RH850P1HC release

EB tresos AutoCore OS version 6.0.208

PA SPC584B release

EB tresos AutoCore OS version 6.0.207

CORTEXM S32K14X release

EB tresos AutoCore OS version 6.0.206

ARM RCARV3HCR7 release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.205

TRICORE TC39XX release

- ASCOS-5592 Fixed known issue: IOC spinlock configuration for Safety OS does not compile
- ASCOS-5096 Fixed known issue: OS_IsScheduleNecessary() and OS_IsScheduleWorthwhile() need a NULL-pointer check.

EB tresos AutoCore OS version 6.0.203

PA SPC58XG release

EB tresos AutoCore OS version 6.0.201

TRICORE TC39XX release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.200

CORTEXM S32K14X release

EB tresos AutoCore OS version 6.0.199

RH850 RH850P1HC release

EB tresos AutoCore OS version 6.0.198

ARM64 ZUXEV release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.197

TRICORE TC39XX release

EB tresos AutoCore OS version 6.0.196

CORTEXM CYT4BF release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.195

TRICORE TC39XX release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.194

TRICORE TC277 release

EB tresos AutoCore OS version 6.0.193 (including EB tresos Safety OS support)

ARM64 ZUXEV release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.192 (including EB tresos Safety OS support)

ARM ZUXEVCR5 release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.191 (including EB tresos Safety OS support)

ARM64 RCARM3N release

EB tresos AutoCore OS version 6.0.190 (including EB tresos Safety OS support)

RH850 RH850F1H release

EB tresos AutoCore OS version 6.0.189 (including EB tresos Safety OS support)

TRICORE TC277 release

EB tresos AutoCore OS version 6.0.188

PA MPC574XG release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.187

TRICORE TC21XL release

EB tresos AutoCore OS version 6.0.186

PA MPC574XG release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.185

PA MPC574XB release

EB tresos AutoCore OS version 6.0.184

TRICORE TC23XL release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.183

PA SPC58EC release

EB tresos AutoCore OS version 6.0.182

PA SPC584C release

EB tresos AutoCore OS version 6.0.181

PA MPC574XG release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.180

CORTEXM S32K14X release

EB tresos AutoCore OS version 6.0.179

RH850 RH850F1K release

WARNING This is an untested development drop. Do not use for production.

- Removed outdated "OS Optimization Assistant"

EB tresos AutoCore OS version 6.0.178

CORTEXM S32G275M7 release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.177 (including EB tresos Safety OS support)

RH850 RH850P1HC release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.176 (including EB tresos Safety OS support)

TRICORE TC29XT release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.175 (including EB tresos Safety OS support)

TRICORE TC277 release

EB tresos AutoCore OS version 6.0.174

ARM BCM89531 release

EB tresos AutoCore OS version 6.0.173

PA SPC584B release

EB tresos AutoCore OS version 6.0.172

RH850 RH850F1L release

EB tresos AutoCore OS version 6.0.171 (including EB tresos Safety OS support)

TRICORE TC22XL release

EB tresos AutoCore OS version 6.0.170 (including EB tresos Safety OS support)

TRICORE TC277 release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.169

PA SPC584B release

- Changed the return type of OS_DiffTime32() from os_tick_t to os_uint32_t.

EB tresos AutoCore OS version 6.0.168 (including EB tresos Safety OS support)

TRICORE TC22XL release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.167

PA QUASAR3 release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.166

RH850 RH850D1X release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.165

TRICORE TC22XL release

EB tresos AutoCore OS version 6.0.164

RH850 RH850F1H release

- ASCOS-5396 Fixed known issue: AUTOSAR counters may not be incremented correctly if simple schedule tables (SSTs) are used.

EB tresos AutoCore OS version 6.0.163

CORTEXM S32K14X release

EB tresos AutoCore OS version 6.0.162

RH850 RH850F1H release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.161

CORTEXM STA1385M3 release

EB tresos AutoCore OS version 6.0.160

ARM RCARM3CR7 release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.159

CORTEXM S32K14X release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.158

RH850 RH850D1X release (including EB tresos Safety OS support)

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.157

PA MPC574XB release

EB tresos AutoCore OS version 6.0.156

PA SPC58XH release

EB tresos AutoCore OS version 6.0.155

CORTEXM STA1385M3 release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.154

ARM S32S247 release

EB tresos AutoCore OS version 6.0.153

PA MPC574XB release

EB tresos AutoCore OS version 6.0.152

PA SPC58XH release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.151

RH850 RH850F1KH release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.150

PA MPC574XB release

- Improved the output of the ORTI generator. The generated ORTI files are now compatible with the WinIDEA debugger.

EB tresos AutoCore OS version 6.0.149

TRICORE TC277 release

EB tresos AutoCore OS version 6.0.148

RH850 RH850D1X release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.147

RH850 RH850F1KM release

EB tresos AutoCore OS version 6.0.146

TRICORE TC38XQ release

- ASCOS-5307 Fixed known issue: Execution budget monitoring might corrupt kernel state.

EB tresos AutoCore OS version 6.0.145

WINDOWS WIN32X86 release

EB tresos AutoCore OS version 6.0.144

RH850 RH850F1KM release

EB tresos AutoCore OS version 6.0.143

TRICORE TC277 release

EB tresos AutoCore OS version 6.0.142

RH850 RH850D1X release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.141

PA SPC584B release

EB tresos AutoCore OS version 6.0.140

RH850 RH850F1KM release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.139

RH850 RH850F1K release

EB tresos AutoCore OS version 6.0.138

CORTEXM BCM89107 release

EB tresos AutoCore OS version 6.0.137

CORTEXM S32K14X release

EB tresos AutoCore OS version 6.0.136

ARM RCARM3CR7 release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.135

WINDOWS WIN32X86 release

EB tresos AutoCore OS version 6.0.134

TRICORE TC23XL release

EB tresos AutoCore OS version 6.0.133

TRICORE TC23XL release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.132

ARM RCARV3MCR7 release (including EB tresos Safety OS support)

- ASCOS-5226 Fixed known issue: Autostart alarms may malfunction, if underlying HW counter is also used as timestamp timer

EB tresos AutoCore OS version 6.0.131

TRICORE TC38XQ release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.130

CORTEXM S32K14X release

EB tresos AutoCore OS version 6.0.129

ARM S32S247 release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.128

PA SPC58XG release

EB tresos AutoCore OS version 6.0.127

ARM RCARV3MCR7 release

EB tresos AutoCore OS version 6.0.126

TRICORE TC38XQ release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.125

ARM RCARV3MCR7 release

WARNING This is an untested development drop. Do not use for production.

- ASCOS-5166 Fixed known issue: ReleaseSpinlock() assigns wrong priority to caller

EB tresos AutoCore OS version 6.0.124

TRICORE TC38XQ release (including EB tresos Safety OS support)

WARNING This is an untested development drop. Do not use for production.

- The macro `OS_ATOMICS_VALUE_MAX` was added to specify the maximum value, that you can store in an object of type `os_atomic_value_t`.

EB tresos AutoCore OS version 6.0.123

TRICORE TC29XT release (including EB tresos Safety OS support)

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.122

CORTEXM S32K14X release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.121

TRICORE TC277 release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.120

ARM RCARM3CR7 release (including EB tresos Safety OS support)

- ASCOS-5107 Fixed known issue: AllowAccess() chooses the wrong application when called from an ISR.

EB tresos AutoCore OS version 6.0.119

TRICORE TC39XX release

EB tresos AutoCore OS version 6.0.118

TRICORE TC39XX release

WARNING This is an untested development drop. Do not use for production.

- Changed the API of the atomic functions. A new type `os_atomic_value_t` was added, which stores the value of atomic objects, which still have the type `os_atomic_t`. Please note, that atomic objects are now *opaque*: you must not use them directly in C language expressions. To access their values, you must use the atomic functions. They return or get values with the new type `os_atomic_value_t` and you may use such values in C language expressions directly.

EB tresos AutoCore OS version 6.0.117

ARM BCM89531 release

EB tresos AutoCore OS version 6.0.116

CORTEXM BCM89107

WARNING This is an untested development drop. Do not use for production.

- Internal module improvement. Exclude additional multi-core code fragments from single-core usage.
- If a schedule table is not on the same core as its referenced counter, the generator no longer outputs a spurious error message for an alarm.
- The names of the following atomic functions were changed. Their arguments, return values and behaviors haven't changed. This means, only a search-and-replace operation is necessary to update your code.

before	**after**
OS_AtomicAdd	OS_AtomicFetchAdd
OS_AtomicSub	OS_AtomicFetchSub

before	after
OS_AtomicAnd	OS_AtomicFetchAnd
OS_AtomicOr	OS_AtomicFetchOr
OS_AtomicXor	OS_AtomicFetchXor

EB tresos AutoCore OS version 6.0.115

ARM RCARV3MCR7

EB tresos AutoCore OS version 6.0.114

PA SPC584B release

EB tresos AutoCore OS version 6.0.113

PA SPC584B release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.112

PA QUASAR2E release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.111

PA QUASAR3 release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.110

ARM BCM89531 release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.109

PA QUASAR3 release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.108

ARM BCM89531 release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.107

PA QUASAR3 release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.106

PA QUASAR3 release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.105

ARM64 RCARM3 release

EB tresos AutoCore OS version 6.0.104

ARM RCARM3CR7 release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.103

TRICORE TC29XT release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.102

WINDOWS WIN32X86 release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.101

ARM S32S247 release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.100

ARM RCARM3CR7 release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.99

PA SPC58XC release

EB tresos AutoCore OS version 6.0.98

TRICORE TC39XX release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.97

CORTEXM BCM89107 release

WARNING This is an untested development drop Do not use for production.

- Implemented *atomic functions*, which allow atomic and sequentially consistent memory access

EB tresos AutoCore OS version 6.0.96

ARM RCARV3MCR7 release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.95

ARM64 RCARV3M release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.94

ARM64 S32V234BE release

EB tresos AutoCore OS version 6.0.93

CORTEXM TDA3x release

- ASCOS-4846 Fixed known issue: On WinCore and Linux, the OS do not start when OS_INITCORE is not zero

NOTE Only WINDOWS/WIN32X86 and LINUX/LXTHREADSX86 are affected.

- ASCOS-4819 Fixed known issue: If the timestamp timer overflows, CPU load calculation never terminates
- ASCOS-4842 Fixed known issue: EB tresos Safety OS: Error in QM-OS may trigger ProtectionHook().
- Reduced default microkernel stack size (`MkKernStack`) on TRICORE, because its MPU start-up code now requires less stack space.
- ASCOS-4841 Fixed known issue: Possible undefined behaviour on ARM/ARM64 derivatives.

NOTE Only multi-core processors are affected.

EB tresos AutoCore OS version 6.0.92

TRICORE TC39XX release

WARNING This is an untested development drop Do not use for production.

EB tresos AutoCore OS version 6.0.91

PA MPC574XB release

- ASCOS-4828 Fixed known issue: WinCore and LinCore OS might crash when nested ISRs are handled

NOTE Only WINDOWS/WIN32X86 and LINUX/LXTHREADSX86 are affected.

EB tresos AutoCore OS version 6.0.90

ARM AR1642 release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.89

RH850 RH850F1H release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.88

CORTEXM MV88Q5050 release

EB tresos AutoCore OS version 6.0.87

TRICORE TC39XX release

WARNING This is an untested development drop Do not use for production.

EB tresos AutoCore OS version 6.0.86

PA SPC58XG release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.85

TRICORE TC39XX release

WARNING This is an untested development drop Do not use for production.

EB tresos AutoCore OS version 6.0.84

ARM64 RCARM3 release

WARNING This is an untested development drop Do not use for production.

EB tresos AutoCore OS version 6.0.83

ARM AR1642 release (including EB tresos Safety OS support)

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.82

PA SPC58XC release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.81

ARM S32V234AA32 release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.80

PA MPC5777C release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.79

ARM RCARM3CR7 release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.78

ARM64 RCARV3M release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.77

TRICORE TC39XX release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.76

TRICORE TC26XD release

EB tresos AutoCore OS version 6.0.75

ARM AR1642 release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.73

PA SPC58XG release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.72

CORTEXM MV88Q5050 release

Added support for ARM Compiler Toolchain Version 5 for CortexM.

- Merge AUTOSAR and ProOSEK personality headers `Os_autosar.h` and `Os_proosek.h` into one common API header. Include `Os.h` instead of the aforementioned headers.
- Since the microkernel supports its own timestamp API, a check was added to OsTimestampTimer such that it may only get enabled if OsMicrokernel is not enabled in the configuration. As the microkernel for TRICORE uses this field to select the microkernel timestamp timer the configuration for TRICORE has been left without check for OsMicrokernel.
- The Multicore Guide is now part of the generic documentation and does not exist as a separate document anymore.

EB tresos AutoCore OS version 6.0.71

ARM RCARV3MCR7 release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.70

CORTEXM MV88Q5050 release

- Since the microkernel supports its own timestamp API, a check was added to OsTimestampTimer such that it may only get enabled if OsMicrokernel is not enabled in the configuration. As the microkernel for TRICORE uses this field to select the microkernel timestamp timer the configuration for TRICORE has been left without check for OsMicrokernel.
- The Multicore Guide is now part of the generic documentation and does not exist as a separate document anymore.

EB tresos AutoCore OS version 6.0.69

PA SPC58XC release (including EB tresos Safety OS support)

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.68

PA MPC574XG release

- Merge AUTOSAR and ProOSEK personality headers `Os_autosar.h` and `Os_proosek.h` into one common API header. Include `Os.h` instead of the aforementioned headers.

EB tresos AutoCore OS version 6.0.67

ARM64 RCARV3M release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.66

ARM RCARV3MCR7 release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.65

RH850 RH850F1H release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.64

TRICORE TC29XT release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.63

CORTEXM MV88Q5050 release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.62

TRICORE TC277 release

EB tresos AutoCore OS version 6.0.61

PA XPC564XC release

EB tresos AutoCore OS version 6.0.60

TRICORE TC277 release

- ASCOS-1757 Fixed known issue: If the application error hook interrupts the start-up or shutdown hook of a non-trusted application, the Os crashes
- Removed the value PARTIAL for the OS attribute PROTECTION due to missing support by the debugger. The value PARTIAL was originally intended to help with debugging on TRICORE while memory protection was turned on.
- ASCOS-4613 Fixed known issue: Compilation errors may occur when using the IOC.

EB tresos AutoCore OS version 6.0.59

RH850 RH850F1K release

- ASCOS-4586 Fixed known issue: Interrupt sources QSPI2HC and QSPI3HC cannot be used on TC22XL and TC23XL

EB tresos AutoCore OS version 6.0.58

WINDOWS WIN32X86 release

- ASCOS-4567 Fixed known issue: Trusted function calls exhibit intermittent failures

NOTE Only WINDOWS/WIN32X86 and LINUX/LXTHREADSX86 are affected.

- ASCOS-4566 Fixed known issue: Possible deadlock when AUTOSAR spinlocks are used

NOTE Only WINDOWS/WIN32X86 and LINUX/LXTHREADSX86 are affected.

EB tresos AutoCore OS version 6.0.57

PA SPC58XC release

EB tresos AutoCore OS version 6.0.56

ARM64 RCARH3 release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.55

PA SPC58XG release

EB tresos AutoCore OS version 6.0.54

PA SPC58XG release

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.53

TRICORE TC29XT release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.52

PA QUASAR3 release (including EB tresos Safety OS support)

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.51

ARM S32V234AA32 release (including EB tresos Safety OS support)

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.50

ARM RCARH3CR7 development drop

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.49

PA XPC564XC release

- ASCOS-4514 Fixed known issue: Tasks may be preempted while holding a category 2 interrupt lock

EB tresos AutoCore OS version 6.0.48

RH850 RH850D1X release

EB tresos AutoCore OS version 6.0.47

RH850 RH850F1H release (including EB tresos Safety OS support)

WARNING This is an untested development drop. Do not use for production.

- Added support for AUTOSAR 4.2 parameter OsSpinlockLockMethod to EB tresos AutoCore OS.

- ASCOS-4462 Fixed known issue: Interrupt locking functions might fail to lock interrupts and might fail to unlock them afterwards.

EB tresos AutoCore OS version 6.0.46

RH850 RH850F1H release

EB tresos AutoCore OS version 6.0.45

PA MPC574XB release

- Lowered cross-core interrupt priority. SuspendAllInterrupts and DisableAllInterrupts now prevent task preemption.
- SuspendOSInterrupts now prevents task preemption. This is especially important for multi-core systems, where other cores could activate tasks.
- Allow stack sizes up to 2 GiB.
- ASCOS-4433 Fixed known issue: Spinlock state might be corrupted by forced termination during spinlock operation.

EB tresos AutoCore OS version 6.0.44

RH850 RH850F1K prototype release

- Removed the limit for the sum of all task activation limits.

EB tresos AutoCore OS version 6.0.43

RH850 RH850P1HC release

EB tresos AutoCore OS version 6.0.42

RH850 RH850P1HC prototype release

- ASCOS-4361 Fixed known issue: Safety OS IOC interrupt locks are always disabled by the generator.

EB tresos AutoCore OS version 6.0.41

ARM S32V234AA32 development drop

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.40

TRICORE TC29XT release (including EB tresos Safety OS support)

- ASCOS-4336 Fixed known issue: OS generator produces an invalid mapping of QM-OS counter ISRs to IRQs on multi-core systems.
- Removed OsSpinlockLockMethod LOCK_VIA_RESOURCES, because there is no use case for it. It is recommended to use a lock method which locks out all other threads which could take a spinlock to prevent certain kinds of deadlocks. Therefore, consider using LOCK_CAT2_INTERRUPTS or even LOCK_ALL_-INTERRUPTS instead.

EB tresos AutoCore OS version 6.0.39

PA QUASAR0 release

EB tresos AutoCore OS version 6.0.38

TRICORE TC277 release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.37

PA SPC58XC release

- Removed the API Removal feature.
- Due to limited support for the StartupHook in the Safety OS, the configuration parameters MkStar-tupHookStack, MkMemoryRegionStartupHookAccess and MkStartupHookMode were removed.
- The generator now makes use of the QMOSISR fast path provided by the EB tresos Safety OS to accel-erate hardware counter interrupt handling.

- For EB tresos Safety OS, it is now possible to configure accessible memory regions and the processor mode individually for trusted functions.

EB tresos AutoCore OS version 6.0.36

CORTEXM S32V234M4 release

EB tresos AutoCore OS version 6.0.35

WINDOWS WIN32X86 release

EB tresos AutoCore OS version 6.0.34

PA QUASAR3 release

EB tresos AutoCore OS version 6.0.33

TRICORE TC26XD release

EB tresos AutoCore OS version 6.0.32

PA QUASAR3 release

- ASCOS-4247 Fixed known issue: Stopping chained schedule tables may cause undefined behavior.
- Added a check to the OS generator for the microkernel to forbid executable regions in dynamic partitions on those derivatives for which they are not supported (currently all TRICORE derivatives and MPC574XP, XPC5777M, and SPC574S60).

EB tresos AutoCore OS version 6.0.31

TRICORE TC29XT release (including EB tresos Safety OS support)

- ASCOS-4147 Fixed known issue: Error handling of CallTrustedFunction does not conform to AUTOSAR requirements.

- ASCOS-4121 Fixed known issue: The syscall for IncrementCounter() may be missing.
- ASCOS-4199 Fixed known issue: The fast interrupt locking mechanism (OsFastInterruptLocking) does not work when used on more than one core.
- Fixed known issue: The generator aborts with an exception.

EB tresos AutoCore OS version 6.0.30

PA QUASAR3 prototype release

WARNING This is an untested prototype release. Do not use for production.

- The naming scheme of make file variables, containing generated IOC file lists, changed. Application lists are now named after the accessing applications. Pattern: `IOC_OBJS_APP(__{application-name})+`. Kernel list suffixes now contain the core indices in ascending order. Pattern: `IOC_OBJS_KERN(_{core-id})+`.
- ASCOS-4146 Fixed known issue: ShutdownOS() does not release spinlocks on the core it was called.

EB tresos AutoCore OS version 6.0.29

Renesas RH850F1H prototype release

WARNING This is an untested prototype release. Do not use for production.

- ASCOS-4073 Fixed known issue: Unintended task dispatching during ISR processing.

NOTE Only WINDOWS/WIN32X86 and LINUX/LXTHREADSX86 are affected.

EB tresos AutoCore OS version 6.0.28

TRICORE TC277 release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.27

ARM TMS570LS1227 release

EB tresos AutoCore OS version 6.0.26

CORTEXM S32V234M4 prototype release

WARNING This is an untested prototype release. Do not use for production.

EB tresos AutoCore OS version 6.0.25

TRICORE TC29XT release (including EB tresos Safety OS support)

- ASCOS-4115 Fixed known issue: Category 1 ISRs may get lost in a multi-core system.
- ASCOS-4126 Fixed known issue: Misconfiguration of STMs which are shared between two cores.
- ASCOS-4155 Fixed known issue: Deadlock in cross-core communication.

NOTE Only WINDOWS/WIN32X86 and LINUX/LXTHREADSX86 are affected.

EB tresos AutoCore OS version 6.0.24

WINDOWS WIN32X86 release

EB tresos AutoCore OS version 6.0.23

ARM TMS570LS1227 release

EB tresos AutoCore OS version 6.0.22

WINDOWS WIN32X86 release

EB tresos AutoCore OS version 6.0.21

ARM RCARH3CR7 release

- ASCOS-3994 Fixed known issue: If OsIocTargetVarsStart and OsIocTargetVarsEnd of an IOC channel are not set, the OS generator allows the channel to write everywhere.
- Parameters of Ioc calls for array types are now typed `void *` or `const void *`.

EB tresos AutoCore OS version 6.0.20

TRICORE TC29XT release (including EB tresos Safety OS support)

WARNING This is an untested development drop. Do not use for production.

- Append sender/receiver identifiers to IocRead and Ioc_ReInit, if there are multiple writers/readers.

EB tresos AutoCore OS version 6.0.19

ARM64 S32V234 release

- ASCOS-4010 Fixed known issue: Generator creates invalid values for MK_CFG_DYNREGIONS_MAX.
- ASCOS-4056 Fixed known issue: Wrong type names are generated for initialization variables of LastIsBest IOC channels.

EB tresos AutoCore OS version 6.0.18

CORTEXM SAMV71 release

- Changed naming scheme of generated IOC channel files, to reduce file name length.

EB tresos AutoCore OS version 6.0.17

TRICORE TC29XT release (including EB tresos Safety OS support)

WARNING This is an untested development drop. Do not use for production.

EB tresos AutoCore OS version 6.0.16

CORTEXM ATLAS7 prototype release

WARNING This is an untested prototype release. Do not use for production.

- ASCOS-3983 Fixed known issue: Some services do not provoke the ErrorHook for an inaccessible application.
- ASCOS-3984 Fixed known issue: OS does not define constants of ApplicationStateType.
- ASCOS-3975 Fixed known issue: Terminating a restarting application does not provoke an error.
- ASCOS-3982 Fixed known issue: Calling `ChainTask()` to a task of a terminated application results in shutdown.
- ASCOS-4005 Fixed known issue: TryToGetSpinlock() might deny service in case memory protection is NOT used.

EB tresos AutoCore OS version 6.0.15

TRICORE TC275 release

- Integrated new IOC.

EB tresos AutoCore OS version 6.0.14

ARM S6J3200 prototype release

WARNING

This is an untested prototype release. Do not use for production. See qualification report for further information.

EB tresos AutoCore OS version 6.0.13

CORTEXM TDA3X release

- Added configuration parameter OsOS/OsInitCoreId, which designates the OS start-up master core.
- Added support for configurations having unused cores within the range of used cores. A core is "unused" if it has no applications.

EB tresos AutoCore OS version 6.0.12

ARM64 S32V234 prototype release

WARNING

This is an untested prototype release. Do not use for production. See qualification report for further information.

- The setting for OsFastInterruptLocking is now always generated. It is no longer possible to define the setting outside of the configuration using the OS_USE_FAST_LOCKING preprocessor macro.
- The S-bit (Safety Task Identifier) of the processor status word for TriCore AURIX processors is now configurable through the OS generator.
- ASCOS-3917 Fixed known issue: PSW.PRS incorrectly set for tasks/ISRs sharing a memory partition in the context of fast partitions.

EB tresos AutoCore OS version 6.0.11

CORTEXM TDA3X prototype release

WARNING

This is an untested prototype release. Do not use for production. See qualification report for further information.

EB tresos AutoCore OS version 6.0.10

TRICORE TC277 development release (including EB tresos Safety OS support)

EB tresos AutoCore OS version 6.0.9

CORTEXM SAMV71 prototype release

WARNING  This is an untested prototype release. Do not use for production. See qualification report for further information.

EB tresos AutoCore OS version 6.0.8

TRICORE TC29XT development release (including EB tresos Safety OS support)

WARNING  This is a development release. See qualification report for further information.

- This change affects the EB tresos Safety OS only: Allow configuring ISRs to use the FPU.
- ASCOS-3798 Fixed known issue: NextScheduleTable returns an incorrect error code.

EB tresos AutoCore OS version 6.0.7

TRICORE TC277 prototype release (including EB tresos Safety OS support)

WARNING  This is a prototype release. Do not use for production. See qualification report for further information.

- The threads `OsHigh`, `OsLow`, `TfHigh` and `TfLow` were removed from the EB tresos Safety OS configuration. Instead the threads `Os` and `Tf` are used.

EB tresos AutoCore OS version 6.0.6

LINUX release

- ASCOS-2744 Fixed known issue: Execution budget monitoring of ISRs can get disabled if a different application is terminated by TerminateApplication().
- ASCOS-3690 Fixed known issue: Large task/ISR stack sizes may give incorrect stack lengths.
- ASCOS-3711 Fixed known issue: The generated configuration for interrupt identifiers does not satisfy the verification criteria in the Safety Manual.
- ASCOS-3712 Fixed known issue: The generated configuration for resources does not satisfy the verification criteria in the Safety Manual.

EB tresos AutoCore OS version 6.0.5

TRICORE TC277 prototype release (including EB tresos Safety OS support)

WARNING 

This is a prototype release. Do not use for production. See qualification report for further information.

EB tresos AutoCore OS version 6.0.4

TRICORE TC29XT prototype release (including EB tresos Safety OS support)

WARNING 

This is a prototype release. Do not use for production. See qualification report for further information.

EB tresos AutoCore OS version 6.0.3

TRICORE TC297 prototype release (including EB tresos Safety OS support)

WARNING 

This is a prototype release. Do not use for production.

EB tresos AutoCore OS version 6.0.2

TRICORE TC275 prototype release

WARNING This is a prototype release. Do not use for production.

EB tresos AutoCore OS version 6.0.1

TRICORE prototype release

WARNING This is an untested prototype release. Do not use for production.

- ASCOS-3493 Fixed known issue: Task/ISR execution budget may be processed with invalid values if the execution budget timer is configured with a hardware timer less than 32 bits.
- ASCOS-1665 Fixed known issue: A kernel stack overflow sometimes is not recognized by the `OS_KernelStackRemaining()` macro.
- ASCOS-2396 Fixed known issue: GetTaskID() fails with E_OS_CALLEVEL when called from the ErrorHook() which itself was raised as a result of alarm or schedule table activity.
- ASCOS-3279 Fixed known issue: Compile error: Invalid redeclaration of CounterType in Mk_autosar.h and Rte_Types.h.
- ASCOS-3540 Extend the OS generator to support microkernel simple schedule tables.

EB tresos AutoCore OS version 4.5.81

PA release

EB tresos AutoCore OS version 4.5.80

PA release

EB tresos AutoCore OS version 4.5.79

TRICORE release

EB tresos AutoCore OS version 4.5.77

TRICORE release (including support for the EB tresos Safety OS)

- ASCOS-2639 Fixed known issue: Possible compilation error in generated IOC file `Ioc_gen.c`.

EB tresos AutoCore OS version 4.5.76

ARM TMS570LS1113 release

EB tresos AutoCore OS version 4.5.75

ARM release (including support for the EB tresos Safety OS)

EB tresos AutoCore OS version 4.5.74

V850 release

- For the EB tresos Safety OS the OS generator removes duplicate memory partitions.

EB tresos AutoCore OS version 4.5.73

V850 release

EB tresos AutoCore OS version 4.5.72

ARM TMS570LS1113 release

EB tresos AutoCore OS version 4.5.71

PA XPC574XM release

- ASCOS-3435 Fixed known issue: Incompatible macro redefinition may cause compiler errors.

EB tresos AutoCore OS version 4.5.70

V850 release

EB tresos AutoCore OS version 4.5.69

PA MPC574XP update release for EB tresos Safety OS

- ASCOS-3398 Fixed known issue: For the microkernel, interoperability problems with the delivered linker script generator occur if the application name is part of the thread stack slot names. This happens when `OsStackOptimization` is set to `WITHIN_APPLICATIONS`.

EB tresos AutoCore OS version 4.5.68

PA release

- ASCOS-2491 Fixed known issue: Unexpected memory protection fault occurs if accessing local variables.
- ASCOS-3385 Fixed known issue: Events of schedule table expiry points missing in ORTI file.

EB tresos AutoCore OS version 4.5.67

RH850 release

- ASCOS-3386 Fixed known issue: In a configuration with more than 255 ISRs some interrupts may not be initialized.

EB tresos AutoCore OS version 4.5.66

PA release (including support for the EB tresos Safety OS)

EB tresos AutoCore OS version 4.5.65

PA release

- The files shipped with the application template now take make.exe from the installed Make-plugin instead of `TRESOS_BASE/bin/make.exe`.

EB tresos AutoCore OS version 4.5.62

V850 release (including support for the EB tresos Safety OS)

EB tresos AutoCore OS version 4.5.61

ARM release (including support for the EB tresos Safety OS)

- Add new functions `OS_TimeSub64()` and `OS_DiffTime32()` and map timestamp API to the microkernel's timestamp API for the EB tresos Safety OS.

EB tresos AutoCore OS version 4.5.60

PA release

- OS_WaitGetClearEvent() (when running under the microkernel) is now defined by the OS, not directly in the microkernel. This change should be transparent to most users.
- Make handling of non-killable ISRs more coherent

EB tresos AutoCore OS version 4.5.59

TRICORE release

EB tresos AutoCore OS version 4.5.58

ARM prototype release

WARNING This is an untested prototype release. Do not use for production.

EB tresos AutoCore OS version 4.5.57

PA release

- Add support for all protection modes in the WDT on Aurix processors.

EB tresos AutoCore OS version 4.5.56 beta

PA release

WARNING This is an untested beta version for PA XPC5777M. Do not use for production code.

EB tresos AutoCore OS version 4.5.55

PA release

- ASCOS-3026 Fixed known issue: The tracing macro OS_TRACE_STATE_TASK is called from the dispatcher using an unknown task id when entering the idle loop.
- ASCOS-3053: Added license check for microkernel support.
- ASCOS-3145 Fixed known issue: In a configuration with only category 1 ISRs the lowest priority category 1 ISR may not be able to interrupt OS system services or hooks.
- Renamed some OS-specific error and panic identifiers so that they differ within the first 31 characters.
- Improve section assignments in linker script generator `genld-tasking.pl`.
- Optimize schedule table configuration: If in an expiry point several events have to be set for the same task, the events are now merged by the generator.

EB tresos AutoCore OS version 4.5.54

PA release

EB tresos AutoCore OS version 4.5.53

TRICORE release

- ASCOS-3145: New configuration parameter `OsTricoreRunningCore` has been added for Aurix derivatives. This parameter tells the OS on which core it will be executed. The configuration parameter `Os-`

`TimestampTimer` has changed for derivatives with multiple STMs and now allows you to select which STM to use as timestamp source.

EB tresos AutoCore OS version 4.5.52

ARM release

EB tresos AutoCore OS version 4.5.51

RH850 release

EB tresos AutoCore OS version 4.5.50

ARM release

- ASCOS-3124 Fixed known issue: Possible kernel stack overflow for nested ISRs (issue for ARM only). The generator adds the stack overhead for nested ISRs to the kernel stack instead of the ISR stack. This is the case for all CPU families.

NOTE

All interrupt stacks are smaller compared to previous versions. Ensure that the size of the ISR stack is still sufficient. If it is too small, increase `OsIsr/OsStacksize` as necessary.

EB tresos AutoCore OS version 4.5.49 beta

RH850 release

WARNING

This is an untested beta version for RH850 P1M. Do no use for production code.

EB tresos AutoCore OS version 4.5.48

ARM release (including support for the EB tresos Safety OS)

- ASCOS-3108: Fixed issue for the EB tresos Safety OS: Enabling `Os/OsSourceOptimization` in the configuration may result in compiler warnings or errors.

EB tresos AutoCore OS version 4.5.47

MB91 release

- ASCOS-2495: For the EB tresos Safety OS, incorrect calls to `StartOS()` are detected and an error is reported.

EB tresos AutoCore OS version 4.5.46

MB91 release

EB tresos AutoCore OS version 4.5.45

PA release (including support for the EB tresos Safety OS)

EB tresos AutoCore OS version 4.5.44

ARM release (including support for the EB tresos Safety OS)

- Updated documentation of the microkernel's user startup panic stop function.

EB tresos AutoCore OS version 4.5.43

V850 release (including support for the EB tresos Safety OS)

EB tresos AutoCore OS version 4.5.42

RH850 release

- ASCOS-2794 fixed issue: Linking the OS fails when using a software counter in a trusted application which is automatically incremented by a OsHwIncrementer feature.

EB tresos AutoCore OS version 4.5.41 beta

RH850 release

WARNING This is an untested beta version for RH850 R1L. Do no use for production code.

- ASCOS-2820 fixed issue: Calling OS_UserGetCpuLoad() or OS_UserResetPeakCpuLoad() no longer yields linker errors.

EB tresos AutoCore OS version 4.5.40

PA release

EB tresos AutoCore OS version 4.5.39

TRICORE release (including support for the EB tresos Safety OS)

EB tresos AutoCore OS version 4.5.38

PA release

EB tresos AutoCore OS version 4.5.37

PA release

EB tresos AutoCore OS version 4.5.36

PA release

EB tresos AutoCore OS version 4.5.35

V850 release (including support for the EB tresos Safety OS)

EB tresos AutoCore OS version 4.5.34

PA release

EB tresos AutoCore OS version 4.5.32

PA release

EB tresos AutoCore OS version 4.5.33

RH850 release

- ASCOS-2820 fixed issue: Calling OS_UserGetCpuLoad() or OS_UserResetPeakCpuLoad() no longer yields linker errors.

EB tresos AutoCore OS version 4.5.31

PA release

EB tresos AutoCore OS version 4.5.30 beta

RH850 release

WARNING This is an untested beta version. Do no use for production code.

EB tresos AutoCore OS version 4.5.29

ARM release (including support for the EB tresos Safety OS)

EB tresos AutoCore OS version 4.5.28

PA release

- Simplified default output directory structure for library files, allow overriding the default.

EB tresos AutoCore OS version 4.5.27

V850 release (including support for the EB tresos Safety OS)

- ASCOS-2800 EB tresos Safety OS-variant: Fixed known issue: The low-priority trusted function thread cannot be granted access to user-configured memory regions.

EB tresos AutoCore OS version 4.5.26

V850 release (including support for the EB tresos Safety OS)

- ASCOS-2757 Fixed issue: The generator creates an invalid interrupt configuration for microkernel configurations with multiple ISRs on the same interrupt level.

EB tresos AutoCore OS version 4.5.25

ARM release (including support for the EB tresos Safety OS)

EB tresos AutoCore OS version 4.5.24

V850 release (including support for the EB tresos Safety OS)

EB tresos AutoCore OS version 4.5.23

V850 release (including support for the EB tresos Safety OS)

EB tresos AutoCore OS version 4.5.22

ARM release (including support for the EB tresos Safety OS)

- ASCOS-2447 Fixed known issue: A resource that is not assigned to any task or ISR can lead to incorrect task priorities.
- The macros `Os_TimestampNsToTicks(ns)` and `Os_TimestampTicksToNs(ticks)` are now publicly available. If an OS counter is used, the macros map to the counter's conversion macros. If a CPU internal counter is used, the macros map to a conversion macro that needs to be defined in `board.h` (template is available).
- If tresos Studio is configured to use Windows line endings, the Os generator now writes a consistent set of line endings.
- ASCOS-2745 Fixed a problem where a task's execution timing could continue running after the task had been terminated by means of TerminateApplication(). The execution budget could expire during another task or even in the idle loop and cause a spurious timing protection fault to be reported.
- ASCOS-2751 Fixed known issue: If an application mode is referenced twice in the autostart configuration of a task, alarm or schedule table, incorrect data initializers are generated.

EB tresos AutoCore OS version 4.5.21

TRICORE release

- ASCOS-1937 Fixed known issue: If OsSecondsPerTick of an OsCounter is not configured in full nanoseconds, the generator throws an exception.

EB tresos AutoCore OS version 4.5.20

V850 release

EB tresos AutoCore OS version 4.5.19

ARM release

EB tresos AutoCore OS version 4.5.18

DSPIC33 release

EB tresos AutoCore OS version 4.5.17beta

ARM release

WARNING This is an untested beta version. Do no use for production code.

EB tresos AutoCore OS version 4.5.16

DSPIC33 release

EB tresos AutoCore OS version 4.5.15beta

ARM release

WARNING This is an untested beta version. Do no use for production code.

EB tresos AutoCore OS version 4.5.14

DSPIC33 release

EB tresos AutoCore OS version 4.5.13

MB91 release

EB tresos AutoCore OS version 4.5.12 Beta

DSPIC33 release

WARNING This is a beta version, do not use for production code.

EB tresos AutoCore OS version 4.5.11

TRICORE release

- Renamed parameter `OsRteUsage` to `OsGenerateSWCD` and improved description.

EB tresos AutoCore OS version 4.5.10

PA release

- ASCOS-2577 Fixed known issue: The Tricore startup code is switching to interrupt stack.

EB tresos AutoCore OS version 4.5.9

DSPIC33 beta release

EB tresos AutoCore OS version 4.5.8

PA release

- ASCOS-2531 Fixed known issue: For tasking v.4 support, the internal tasking function _c_init() is used to initialize data/bss sections.
- ASCOS-2541 Fixed known issue: Realign signatures in software component description with function signatures in RTE wrappers for the microkernel variant.
- ASCOS-2538 Fixed known issue: Change include order between `Os.h` and the generated `Os_user.h` to restore the functionality of the `OsFastInterruptLocking` configuration parameter.

EB tresos AutoCore OS version 4.5.7

PA release

EB tresos AutoCore OS version 4.5.6

TRICORE release

EB tresos AutoCore OS version 4.5.5

TRICORE release

EB tresos AutoCore OS version 4.5.4

V850 release

EB tresos AutoCore OS version 4.5.3

TRICORE release

EB tresos AutoCore OS version 4.5.2

TRICORE release

EB tresos AutoCore OS version 4.5.1

PA release with microkernel support

- ASCOS-2454 Fixed known issue: The fixed known issue ASCOS-1784 was incomplete so the Dbg module still generated wrong ids of Os tasks and ISRs for generic state tracing. This is fixed now.
- The Sidebar view action "Edit Os Optimization Settings" was not visible in an AUTOSAR 3.1 project. Furthermore, the Dbg module was not able to generate any ids for Os tasks and ISRs in conjunction with the AUTOSAR 3.1 Os module.

EB tresos AutoCore OS version 4.5.0 Studio11

PA release with microkernel support; for use with EB tresos Studio 11.0.3 only

- ASCOS-2436 Fixed known issue: The configured time constant macros of the OsTimeConstant list within counter timers are not generated.

EB tresos AutoCore OS version 4.4.21

PA release

- For the microkernel, the Os generator generates an interrupt level fixed to 1 for the os low thread (should be `MK_HWENABLEALLLEVEL`) and a wrong value for the number of memory partitions (`MK_NMEMORY-PARTITIONS`) - fixed.
- ASCOS-2442 Fixed known issue: OS generator reports "untested version, do not use for production code".

EB tresos AutoCore OS version 4.4.20

PA release

- ASCOS-2245 Fixed known issue: Stack length of trusted function is now generated under all circumstances.

EB tresos AutoCore OS version 4.4.19

PA release

- The Dbg module now generates proper ids of Os tasks and ISRs for generic state tracing.
- ASCOS-2340 Fixed known issue: OS generator creates files with CR+LF line endings.
- Macros to convert counter ticks into nano-seconds and vice versa now support CPU frequencies up to 1GHz.

EB tresos AutoCore OS version 4.4.18

PA release

EB tresos AutoCore OS version 4.4.17

TRICORE release

- ASCOS-2380 Fixed known issue: For Autosar 3.1, the API function GetElapsedCounterValue() cannot be accessed as an Rte port.

- Introduced new parameter OsSourceOptimization to replace user configurable make variable `OS_BUILD_OPTIMIZED_LIB_FROM_SOURCE`.

EB tresos AutoCore OS version 4.4.16

PIKEOS release

EB tresos AutoCore OS version 4.4.15

PA release

EB tresos AutoCore OS version 4.4.14

ARM release

EB tresos AutoCore OS version 4.4.13

V850 release

- ASCOS-1920 Fixed know issue: Os generator generates ORTI files that cannot be loaded using the Lauterbach debugger - fixed: The ORTI files use the OTM extension exactly in the way specified by Lauterbach now.

EB tresos AutoCore OS version 4.4.12

PIKEOS release

EB tresos AutoCore OS version 4.4.11

PA release

- Application example has been moved to new location. It is now stored at `$TRESOS_BASE\demos\AutoCore_OS`.

EB tresos AutoCore OS version 4.4.10

R32C release

- ASCOS-2208 Fixed known issue: The Os generator returns wrong task Ids for the Debug and Trace module.
- ASCOS-2116 Fixed known issue: Tasks entering the WAITING state are not reported to Debug&Trace interface.

EB tresos AutoCore OS version 4.4.9

ARM release

- ASCOS-2274 Fixed known issue: Generator fails with cryptic error message for a schedule table with Autostart but OsScheduleTableStartValue disabled. Fixed: OsScheduleTableStartValue is no longer optional.

EB tresos AutoCore OS version 4.4.8 Beta

ARM release

WARNING

This ia s beta version, do not use for production code because some test cases are not running. Caution this beta version has the following restrictions: System calls must not be called in the hook functions. The error, startup, shutdown hook and the ISRs must be configured to be directly called by the OS.

- ASCOS-2214 Fixed known issue: If an IOC system call is performed from a non-trusted ISR, the IOC fails to correctly verify the permissions of the pointers passed as arguments.
- ASCOS-2165 Fixed known issue: The OS runs out of kernel stack when no tasks are executing. Fixed: The idle loop is now coded in assembly language to ensure that no stack is used and that a switch to the kernel stack takes place anyway on the next interrupt.
- If alarm callbacks are used in scalability class 2 and above, a warning is now issued instead of an error message.
- ASCOS-2085 Fixed known issue: OS generator fails if not all tasks belong to OS applications and stack optimization is configured to WITHIN_APPLICATIONS.

EB tresos AutoCore OS version 4.4.7

V850 release

EB tresos AutoCore OS version 4.4.6

PA release

- ASCOS-2222 Fixed known issue: Configuration parameter OsStackMonitoring is ignored. Fixed: Removed vendor specific parameter OsStackCheck which was duplicated by AUTOSAR parameter OsStackMonitoring. The generator now uses the parameter OsStackMonitoring to enable/disable stack checking.
- Align documentation and generation of version information macros (like OS_AUTOSAROS_VER) to coding style.

EB tresos AutoCore OS version 4.4.5

ARM release

- ASCOS-2105 Fixed known issue: Tasks which return from their main function without calling `TerminateTask()` or `ChainTask()` will now be killed even if OsErrorHandling is set to MINIMAL (instead of resulting in an endless loop).

EB tresos AutoCore OS version 4.4.4

TRICORE release

EB tresos AutoCore OS version 4.4.3

PA release

- ASCOS-2130 Fixed known issue: The IOC now uses const qualifiers for all input parameters that are passed to the `Ioc_Send_*` and `Ioc_Write_*` functions.

EB tresos AutoCore OS version 4.4.2

ARM release

- Allow calling OS API in ErrorHook when OsErrorHandling is set to MINIMAL

EB tresos AutoCore OS version 4.4.1

PA release

EB tresos AutoCore OS version 4.4.0

WINDOWS release

- Generator-warning OS_702 (schedule table attached to SW counter) shall be removed
- Update configuration schema and module registration to Autosar 4.0

EB tresos AutoCore OS version 4.2.27

PA release

EB tresos AutoCore OS version 4.2.26

TRICORE release

WARNING

Caution is necessary when enabling the XBAR_SRI interrupt. See the errata document `CPU_TC_H007_EPN_v1_0_02081A.pdf` provided by Infineon.

EB tresos AutoCore OS version 4.2.25

TRICORE release

- ASCOS-2060 Fixed known issue: Implicit typecast from `(T const * *)` to `(void * *)` in `Ioc_gen.c`.

EB tresos AutoCore OS version 4.2.24

V850 release

EB tresos AutoCore OS version 4.2.23

TRICORE release

EB tresos AutoCore OS version 4.2.22

PA release

EB tresos AutoCore OS version 4.2.21

ARM release

EB tresos AutoCore OS version 4.2.20

TRICORE release

EB tresos AutoCore OS version 4.2.19

V850 release

EB tresos AutoCore OS version 4.2.18

ARM release

EB tresos AutoCore OS version 4.2.17

PA release

- For scalability classes 3 and above, setting OsOS/OsStatus to STANDARD now only leads to a warning message instead of an error.

EB tresos AutoCore OS version 4.2.16

ARM release

EB tresos AutoCore OS version 4.2.15

PA release

- Input parameters of Ioc calls for array types are now typed `(void *)`. It is now compatible with both AUTOSAR 3.1 and 4.0 RTE.
- Updated internal API for IOC system calls. All pointers that are not written to by the kernel are now qualified as `const`.

EB tresos AutoCore OS version 4.2.14

PA release

EB tresos AutoCore OS version 4.2.13

PA release

- Increase maximum number of expiry entries in a schedule table from 255 (uint8) to 65535 (uint16).
- ASCOS-1377 Fixed known issue: Removed internal resource OS_RESSCHEDULER to which RES_SCHEDULER has been mapped. Instead, RES_SCHEDULER will now be provided directly by the OS generator.

EB tresos AutoCore OS version 4.2.12 Beta

PA release

EB tresos AutoCore OS version 4.2.11

XC2000 release

EB tresos AutoCore OS version 4.2.10

PA release

EB tresos AutoCore OS version 4.2.9

PA release

EB tresos AutoCore OS version 4.2.8

PA release

EB tresos AutoCore OS version 4.2.7

XC2000 release

- ASCOS-1850 Fixed known issue: An empty string in the parameter OsIocInitValueSymbol leads to a compilation error. Fixed: Added check that OsIocInitValueSymbol is not empty.
- In the STM timer driver the number of times registers are read back to flush the pipelines has been reduced in order to improve performance. According to Infineon this will not affect the functionality in any way.
- ASCOS-1793 Fixed known issue: ProtectionHook can be called twice when an execution budget expires. Fixed: The order of processing in the macro OS_CAT2PREEMPTEXECTIMING() has been changed to avoid the possibility of the same timing-protection violation being detected twice.

EB tresos AutoCore OS version 4.2.6

PA release

EB tresos AutoCore OS version 4.2.5

WINDOWS release

EB tresos AutoCore OS version 4.2.4

PA release

EB tresos AutoCore OS version 4.2.3

PA release

EB tresos AutoCore OS version 4.2.2

PA release

EB tresos AutoCore OS version 4.2.1

WINDOWS release

EB tresos AutoCore OS version 4.2.0

WINDOWS release

- The new API functions OS_IsScheduleNecessary() and OS_IsScheduleWorthwhile() and the macros OS_ScheduleIfNecessary() and OS_ScheduleIfWorthwhile() have been added to help improve the performance of non- and mixed-premptive systems.

EB tresos AutoCore OS version 4.1.25

SH4 (system call) alpha version

EB tresos AutoCore OS version 4.1.24

SH4 (non system call) release

EB tresos AutoCore OS version 4.1.23

V850 release

- The new API functions OS_IsScheduleNecessary() and OS_IsScheduleWorthwhile() and the macros OS_ScheduleIfNecessary() and OS_ScheduleIfWorthwhile() have been added to help improve the performance of non- and mixed-premptive systems.

EB tresos AutoCore OS version 4.1.22

TRICORE release

EB tresos AutoCore OS version 4.1.21

TRICORE release

EB tresos AutoCore OS version 4.1.20

XC2000 release

EB tresos AutoCore OS version 4.1.19

ARM release

- The private data of tasks, ISRs and applications now remain uninitialized if the address of the ROM image of the .data sections is not NULL.In earlier versions, a NULL pointer caused the whole region to be set to zero following the uninitialized variables conventions for C. The new version permits access to be granted to "noinit" regions, preinitialized regions and memory-mapped peripherals. Bss conventions are still supported by setting the base and limit addresses of the data image to the same (non-NULL) value, which is what the ldscript generators do anyway.
- The execution timer is now stopped when the OS gets shut down. This prevents any unexpected awakening when the timer expires.

EB tresos AutoCore OS version 4.1.18

V850 release

EB tresos AutoCore OS version 4.1.17

V850 release

EB tresos AutoCore OS version 4.1.16

PA release

- ASCOS-1568 Fixed known issue: The OS generators now provides macros for separate application hook stacks (startup, shutdown and error hook). If an application hook is configured, the kernel switches to a separate stack area before starting the hook with a size defined in the application's hook stack size parameters. This is supported e.g. for the PA and the ARM architecture.
- OS_CanWrite() no longer dereferences a NULL pointer.
- Restore the original interrupt status in all error branches of OS_KernStartOs()
- Corrected OS Generator to allow category 1 interrupts belonging to trusted applications in scalability classes 3 and 4. All other scenarios will lead to a generation error in correspondence with Autosar requirement OS361.

EB tresos AutoCore OS version 4.1.15

PA release

EB tresos AutoCore OS version 4.1.14

PA release

EB tresos AutoCore OS version 4.1.13

V850 release

EB tresos AutoCore OS version 4.1.12

PA release

- ASCOS-1593 Fixed known issue: The code for the API functions `OS_GetCurrentStackArea()`, `OS_GetIsrMaxRuntime()` and `OS_GetTaskMaxRuntime()` has been moved to the kernel library. With this

change, the references to the internal data structures get updated upon recompile when the configuration has changed and optimized libraries are used.

EB tresos AutoCore OS version 4.1.11

PA release

- The constant data structures containing the configuration of the OS objects are now configuration dependent. All elements of a data structure that are not used for a given configuration are removed to reduce the size of these objects.
- The RUNNINGTASK entry in the ORTI file has been changed to point to OS_taskCurrent instead of OS_taskCurrent->dynamic. Some debuggers cannot evaluate complex entries for this keyword.

EB tresos AutoCore OS version 4.1.10

XC2000 release

- The macros OS_IsTimeStampTimer() and OS_GetLastTimeStampValue() have been removed since they are no longer needed. The one remaing call to OS_GetLastTimeStampValue() has been replaced with the variable itself, and the function body is conditionally removed depending on use of generic timestamp.

EB tresos AutoCore OS version 4.1.9

V850 release

EB tresos AutoCore OS version 4.1.8 BETA

PA release

- The System Services that return info via a referenced variable now return the information correctly when called from a non-trusted ISR using a reference to a stack variable.

EB tresos AutoCore OS version 4.1.7

V850 release

EB tresos AutoCore OS version 4.1.6

V850 release

- ASCOS-1485 Fixed known issue: The generator now correctly handles the OsErrorHookForVoidApi attribute in the OsAutosarCustomization object.
- The generator now correctly generates the OS_LCFG_TASKQUEUE_SMALL macro, which means that the priority lookup in the task queue handler happens in a single step when there are 32 or fewer priorities (16 on 16-bit systems).
- Implement a simplified version of the error handling that simply calls the error hook for every error, then returns to the caller. OSEK service ID and parameter information are also optionally supported.
- The linked-list method of managing the task queue has been reinstated as a build option because this method can be faster in some configurations. The method can be selected by a macro in Os_cfg.mak.

EB tresos AutoCore OS version 4.1.5

PA release

EB tresos AutoCore OS version 4.1.4

PA release

- ASCOS-1475 Fixed known issue: In ShutdownOS() it is now guaranteed that all interrupts are locked (including the timing protection interrupt) before calling the shutdown hooks and entering the endless loop.

EB tresos AutoCore OS version 4.1.3

V850 release

- Removed version restrictions for EB tresos Studio plugins in Manifest.MF to provide compatibility with future Studio version.

EB tresos AutoCore OS version 4.1.2

V850 release

- ASCOS-1450 Fixed known issue: After an ISR, memory protection settings are now always restored to prevent the incorrect memory protection boundaries from being restored after nested interrupts.
- ASCOS-1453 Fixed known issue: New Os internal macro OS_TimerSub() added. Calculates time differences including overflow handling like needed in timer driver start function.

EB tresos AutoCore OS version 4.1.1

V850 release

- OsCounterTimer and OsCounterIrqLevel warning "Removing ENABLE attribute for * because it may not be calculated for an optional node" within tresos 2010 fixed.

EB tresos AutoCore OS version 4.1.0

WINDOWS release

- Generated OS files are not rewritten anymore when the configuration does not change between generator runs.
- Add OS Optimization Assistant.
- Re-enable OIL-importer by default.
- The task priority queue algorithm has been rewritten to provide better performance, and an almost-constant task activation time. In addition, the new algorithm offers optimizations for low conformance classes (BCC1/ECC1) and systems with few tasks and priority levels.
- The alarm wrappers for ActivateTask, SetEvent and IncrementCounter now call internal kernel functions directly rather than via the API function. This eliminates a layer of error checking that can never fail and should improve performance of alarm and schedule table handling.
- An automatically-ticked software counter has been added to the range of time-generation features in the OS. A ticked counter may bring performance benefits on systems with regular timing requirements and no synchronization with global time. The availability of the ticker is subject to hardware features of the microcontroller and availability of drivers.
- A new error-reporting function has been introduced for internal errors from which no recovery is possible. OS_Panic() attempts to report the error using standard error reporting in OS_Error(), which should caused a shutdown. However, if OS_Error() returns for any reason (for example, if error handling has been disabled), the kernel shuts down. If a breakpoint is placed at the first statement of OS_Panic(), a meaningful code should be available in the debugger. OS_Panic() is now called for all errors that are defined map to the StatusType value of E_OS_PANIC.
- Consistency checks of the generated configuration have been implemented. If enabled, the checks run at startup and report errors by calling OS_Panic().

- The timer handling has been unified so that the execution timer can also be used as the timestamp timer.
- The interrupt handing functions for hardware counters and for the execution timer have been diversified so that the Generator can select an appropriate function, thus eliminating the test to determine whether the timer is the timestamp timer.
- The OS now extends the execution timer to 32-bits to avoid problems when tasks need to have a long execution budget specified. Intermediate interrupts are inserted when the budget exceeds the range of the physical hardware, in a way similar to that of the extended hardware counter that has been available for some time.
- Arrival rate montioring now uses the 64-bit timestamp to determine the time between two arrivals, eliminating potential problems with timer wraparound when configuring arrival rate limits for interrupts or task activations that normally happen infrequently.

EB tresos AutoCore OS version 4.0.48

R32C release

- ASCOS-1363 Fixed known issue: When OsStackOptimization is activated, the generator might place tasks that preempt each other on the same stack.
- ASCOS-1361 Fixed known issue: A non-premptive task can get preempted after calling Schedule() if an application gets terminated from an interrupt at the same time.
- Re-introduced OsIsrCountLimit and OsTaskCountLimit configuration parameters. With these parameters, the number of interrupt or task arrivals within a given time frame can be configured.

EB tresos AutoCore OS version 4.0.47

TRICORE release

EB tresos AutoCore OS version 4.0.46

PA release

EB tresos AutoCore OS version 4.0.45

V850 release

- Os.h now includes board.h so that the time-to-ticks and ticks-to-time macros are fully resolved without having to include a vendor-specific header file.

- Support for Derivate TC1782 added. [ASCOS-1309]

EB tresos AutoCore OS version 4.0.43

V850 release

EB tresos AutoCore OS version 4.0.42

TRICORE release

- ASCOS-1321 Fixed known issue: The execution timer interrupt now uses the highest available priority of all interrupts. Before this change, the priority of the execution timer was incorrectly calculated if hardware counters with a priority higher than all other ISRs existed. In this case, the execution timer interrupt could have been blocked with a SuspendOsInterrupts() API call causing interrupt lock timing protection to fail.
- If an application with a running restart task is restarted (e.g. due to an exception), the task is now terminated and started again. Previously, the restart task continued to run.
- If OsCat1DirectCall is enabled, the prototype of the ISR can now be prefixed by defining the macro OS_-INTERRUPT_KEYWORD prior to including Os.h. This allows to pass a toolchain specific keyword (e.g. __interrupt) to the prototype of the ISR.

EB tresos AutoCore OS version 4.0.41

PA release

EB tresos AutoCore OS version 4.0.40 BETA

PA release

EB tresos AutoCore OS version 4.0.39

PA release

- Introduced "Enable Rte usage" configuration parameter. Service software component description is not generated if Rte usage is switched off.

EB tresos AutoCore OS version 4.0.38

PA release

- ASCOS-1270 Fixed known issue: The automatically assigned priority of the execution timer interrupt is now only checked if an execution timer is actually configured.
- Update configuration schema and module registration to Autosar 3.1.
- ASCOS-1270 Fixed known issue: The automatically assigned priority of the execution timer interrupt is now only checked if an execution timer is actually configured.

EB tresos AutoCore OS version 4.0.37

SH4 release

EB tresos AutoCore OS version 4.0.36

PA release

EB tresos AutoCore OS version 4.0.35 BETA

TRICORE release

EB tresos AutoCore OS version 4.0.34

PA release

EB tresos AutoCore OS version 4.0.33

PA release

EB tresos AutoCore OS version 4.0.30

SH4 release

EB tresos AutoCore OS version 4.0.29

V850 release

- The fast interrupt locking mechanism can now be enabled via the configurator as an optional parameter. The parameter is named OsFastInterruptLocking and located in the OsOS/OsAutosarCustomization container.
- ASCOS-1207 Fixed known issue: If a software counter is driven by a Gpt driver, the inclusion of Gpt.-h is now encapsulated by an "#ifndef OS_ASM" statement to prevent the inclusion of C statements into assembler files.

EB tresos AutoCore OS version 4.0.26

PA release

- The timestamp driver for TRICORE, using the 56-bit hardware timer in the STM has been implemented. The idle-exit code in the interrupt handler is implemented in assembly language very early in the interrupt handler, thus making the measurement very accurate.

EB tresos AutoCore OS version 4.0.25 BETA

SH4 release

EB tresos AutoCore OS version 4.0.24

PA release

EB tresos AutoCore OS version 4.0.23 BETA

SH4 release

EB tresos AutoCore OS version 4.0.22

PA release

EB tresos AutoCore OS version 4.0.21

PA release

- The API function OS_GetCurrentStackArea has been added.
- ASCOS-1100 Fixed known issue: If a software counter is driven by a Gpt driver, the reference path to the configured GPT channel is now removed in the generated configuration, which lead to uncompilable code. In addition, the Os configuration header file now also includes Gpt.h for such a configuration.
- The OS generator now correctly converts event ids to a hexadecimal string prior to prefixing them with "0x".

EB tresos AutoCore OS version 4.0.20

PA release

EB tresos AutoCore OS version 4.0.19

V850 release

- The API function OS_GetCurrentStackArea has been added.
- ASCOS-1100 Fixed known issue: If a software counter is driven by a Gpt driver, the reference path to the configured GPT channel is now removed in the generated configuration, which lead to uncompilable code. In addition, the Os configuration header file now also includes Gpt.h for such a configuration.
- The OS generator now correctly converts event ids to a hexadecimal string prior to prefixing them with "0x".

EB tresos AutoCore OS version 4.0.17

PA release

EB tresos AutoCore OS version 4.0.17

PA release

- ASCOS-1084 Fixed known issue: When generating the counter configuration, the generator now checks the status of the OsDriver container before evaluating OsGptChannelRef.

- ASCOS-1064 Fixed known issue: MMU support is now active for XPC563XM Monaco derivate. A special function RemapMmuEntries has been implemented for this derivate to mark selected entries as OS or user entries. The OS memory protection may use all but the OS entries of non-trusted applications, tasks or ISRs. The OS needs a minimum of 3 entries, which allows non-trusted applications, tasks or ISRs to use up to 5 TLB entries to access private memory areas.
- ASCOS-1092 Fixed known issue: The generator check for synchronizable schedule tables that expiry point offset plus MaxAdvance must not exceed the duration has been changed. It is now checked that MaxAdvance does not exceed the schedule table's duration. This is a deviation to Autosar 3.0 requirement OS437, but necessary for certain OS configurations (see Autosar Bugzilla entry http://www.autosar.org/bugzilla/show_bug.cgi?id=37670).

EB tresos AutoCore OS version 4.0.16

V850 release

EB tresos AutoCore OS version 4.0.15

V850 release

EB tresos AutoCore OS version 4.0.14

PA release

EB tresos AutoCore OS version 4.0.12

V850 release

EB tresos AutoCore OS version 4.0.11

PA release

EB tresos AutoCore OS version 4.0.10 BETA

PA release

- ASCOS-1007 Fixed known issue: When using the optimized library build, syscall/non-syscall (OsTrappingKernel option) and tracing/non-tracing libraries have their own bitmask to avoid linking errors.
- Support for TC1387 derivative added.

EB tresos AutoCore OS version 4.0.9

PA release

EB tresos AutoCore OS version 4.0.8

WINDOWS release

- The type for StatusType has been changed. It was previously 16-bit, and is now 8-bit to be compatible with the Autosar general specifications and the OSEK Binding Specification. All the error codes that were outside the 8-bit range are now in the range 0-31. Unfortunately this destroys the conformance with the Binding Specification in another area, but that is unavoidable.

EB tresos AutoCore OS version 4.0.7

PA release

EB tresos AutoCore OS version 4.0.6

PA release

- ASCOS-977 Fixed known issue: Removed NullPointerException in generator when references to other OS objects are invalid.
- The alignment size of the stack arrays is now independent of the size of a stack element. On most architectures it remains identical with the stack element size, however, some architectures require different alignment boundaries e.g. for memory protection. The value is adjustable via the configuration database parameter STACK_ALIGNMENT_SIZE.
- The test for a terminating task in syscall, cat2 and exception exit has been improved. It now uses 1 instruction to compare with OS_TS_MAX_TERMINATING instead of 3 separate equality comparisons.
- ASCOS-395 Fixed known issue: Resource lock timing fails when no execution budget is configured.
- ASCOS-718 Fixed known issue: Max runtime calculation for tasks/ISRs not working.

- ASCOS-782 Fixed known issue: Interrupt lock timing (Task) fails if task has no execution budget

EB tresos AutoCore OS version 4.0.5

PA release

- The OsTrappingKernel parameter has been moved into the OsOS container as optional parameter. If the optional parameter is disabled, the setting will be determined automatically based on the Os configuration.
- ASCOS-969 Fixed known issue: Removed NullPointerException in the generator if the selected timebase equaled the timer of a hardware counter.

EB tresos AutoCore OS version 4.0.4

V850 release

- The type of the "clean" field in the os_stackinfo_t structure has been changed to os_size_t, and the stack reporting functions have had their return types changed to this type too. This avoids the need for (possibly unnecessary and possibly incorrect) casting in the API functions - including the SALSA API.
- I ASCOS-202 Fixed known issue: In the local time calculation in SyncScheduleTable(), the time remaining to the next expiry point is coerced into the range 0 .. (duration-1) using a modulo operation. This prevents an underflow in the local time calculation which in turn could result in incorrect synchronization if SyncScheduleTable() is called when the next expiry point is more than the schedule table's duration in the future, and is subsequently not called again.

EB tresos AutoCore OS version 4.0.3

SH2 release

- Task and ISR stacksize with 0 (zero) possible in configuration.

EB tresos AutoCore OS version 4.0.2

PA release

EB tresos AutoCore OS version 4.0.1

V850 release

EB tresos AutoCore OS version 4.0.0

WINDOWS release

- A configuration importer for OS configurations using the Autosar 2.1 XML configuration schema (AUTOSAR OS 3.0/3.1) has been added. Most configuration options can be imported into an Autosar 3.0 project.

EB tresos AutoCore OS version 3.0.1

- ASCOS-560 Fixed known issue: Kernel interrupts are now locked at the correct level inside the "All" interrupt lock/unlock functions when timing protection is in use.
- ASCOS-559 Fixed known issue: The macro OS_STARTGPTTICKERS is now defined correctly. This means that the file Os_startgpttickers.c will compile without errors when a GPT-driven software counter is configured. Additionally, the optional "optimized" version of the GPT notification function now uses the correct internal OS function.
- ASCOS-605 Fixed known issue: The macros OS_TICKS2US_HWCounter, OS_TICKS2MS_HWCounter and OS_TICKS2SEC_HWCounter have been corrected.
- The PLL settings for the TC1797 in the application template have been changed. The previous settings operated the VCO at 360 MHz, which appears to be outside the operating range. The result was that the CP frequency was not accurate. The new settings operate the VCO at 720 MHz, which appears to be inside the operating range.
- ASCOS-615 Fixed known issue: An extra test has been added to OS_FindTaskSp() (TRICORE-findtasksp.c) so that upper-CSAs whose PSW.IS flag is set are ignored when tracing back along the task's CSA list for the stack pointer. This is a workaround for an apparent hardware problem on the TC1797 whereby the CPU accepts an interrupt after a SYSCALL instruction has been executed.
- ASCOS-628 Fixed known issue: The Generator has been modified so that it emits the correct running priority for ISRs that are configured with the same level. Before this change only ISRs at the highest configured level would get the correct running priority; all others with equal configured level would get different running priorities. The unexpected nesting could cause stack overflow or free context-list depletion.
- ASCOS-634 Fixed known issue: The Generator now selects the correct entry in a choice container. Before this the generator selected always the first entry, which makes it hard to use the GUI for proper ALARM action configuration.
- The Generator has been modified so that it removes the temporary directory after generating the files.
- ASCOS-641 Fixed known issue: Tooltip text updated for attributes that are using a time unit.
- The file TRICORE-interruptvectors.asm is now updated if changing the interrupt nodes in the configuration. This was leading to code which fits not to the configuration or old configuration.

- ASCOS-744 Fixed known issue: The problem with SyncScheduleTable() in an ISR has been fixed. The problem caused the OS to shut down spuriously if the ISR interrupted the processing of an expiry point on the schedule table.
- ASCOS-717 Fixed known issues: TC1797 BSP: PLL configuration change to give the expected divide-by-72 in the VCO feedback loop.
- ASCOS-615 Fixed known issue: TC1797: A full workaround has been implemented for the silicon problem that permits an interrupt to be serviced between a trap being acknowledged and the first instruction of the trap being executed. The workaround adds 4 extra instructions to each interrupt vector, 2 of which are executed every time an interrupt occurs. The extra instructions detect whether the interrupt has occurred incorrectly and if so they branch to a short handler that retriggers the interrupt and then returns to the interrupted context without handling the interrupt. The workaround is only present for the TC1797 (and other derivatives that use the same core). If the workaround is not considered necessary it can be inhibited by defining the macro OS_INCLUDE_BF11_WORKAROUND to be 0, either on the command line or in some header file that gets included by TRICORE-interruptvectors.s. If an interrupt that is not known to the kernel occurs in the critical window, it will not get retriggered because the OS does not know which service request register to use. The correctness of this workaround depends on the vector table being filled from priority 1 upwards with no gaps, so that unknown interrupt priorities are always higher than the number of ISRs configured.
- ASCOS-745 Fixed known issue: If a task defines a resource locking time and no ISR that owns the same resource is available a java.lang.NullPointerException occurs while verifying/generating.
- Update to Autosar OS SWS specification version 3.0.1 (revision 0003). This update brings the following changes:
  - The official configuration language is now XML, not OIL. The XML schema has several changes over the previous unofficial schema.
  - The second parameter to StartScheduleTableSynchron() and the data types GlobalTimeTickType and UnitType no longer exist.
  - SCHEDULETABLE_NOT_STARTED is now SCHEDULETABLE_STOPPED. Affects all APIs that use the status of a schedule table.
  - StartScheduleTableAbs() and StartScheduleTableRel() now start the schedule table so that its "start-of-round" (not its first expiry point) occurs at the specified counter value.
  - Schedule tables con now be configured as implicitly synchronized. This means that they run on a counter that is automatically synchronized to global time. Such schedule tables can only be started with StartScheduleTableAbs(), and cannot be the target of SyncScheduleTable() or SetScheduleTableAsync(). Use with NextScheduleTable() is also restricted. The original synchronization type is now called "explicit" synchronization.
  - Automatically-started schedule tables and alarms can now be started with the "Abs" method as well as the "Rel" method. For schedule tables, "auto-start synchronously" is also available.

- Schedule tables are now specified by an adjustment (maxIncrease and maxDecrease) that is specified per expiry point. The old concept of SMOOTH vs. HARD synchronization has gone, as has the concept of different max inc/dec values for synchronous and asynchronous schedule tables.
- If a schedule table activates a task and then sets an event for the same task at the same time ("exipry point"), the OS now guarantees that the activation will take place before the event gets set, so that the SetEvent action doesn't result in an error.
- The "previous time" parameter to GetElapsedCounterValue() is now a reference, and the previous value is updated when the API is successfully called.
- The APIs EnableInterruptSource() and DisableInterruptSource() no longer exist. However, the EB APIs OS_UserEnableInterruptSource() and OS_UserDisableInterruptSource() still exist, although OS_UserEnableInterruptSource() no longer makes reference to the rate-limit status of the ISR.
- Explicit permission must now be given for all accessing applications. Previously it was assumed that trusted applications could access all other objects without restrictions. The result of this is that the total number of applications (both trusted and non-trusted) that is supported by the OS is governed by the width of the permissions word (normally 32). Previously the limit was 32 non-trusted applications and a total of 255 applications. A failure due to lack of permission now results in E_OS_ACCESS (was E_OS_ID).
- If an ISR blocks interrupts using an API function and returns without calling the corresponding un-blocking function, the OS calls the error hook with E_OS_DISABLEDINT (was E_OS_MISSINGEND).
- If an ISR takes a resource with GetResource() and returns without freeing the resource with ReleaseResource(), the OS calls the error hook with E_OS_RESOURCE (was E_OS_MISSINGEND).
- Timing protection has reverted to the SWS2.0 scheme of a single-shot execution budget per task and ISR, and a limit on the number of "arrivals" in a time frame. For strict Autosar conformance the arrival limit is 1 per time-frame, but EB implements a configurable limit. One significant difference is that successfully waiting for an event that is already pending is now considered to be an arrival and restarts the execution time limit.
- If a protection violation occurs and no protection hook is configured, the OS now shuts down as per Autosar requirement. An EB extension can be enabled to inhibit this behavior.
- The action if ProtectionHook returns KILLTASKISR for a protection error in an application hook has been changed. The action now is always to kill the application, as per OS243.

- The output of the generated files is now split up in src,include, make, orti and info. This change increases the conformance to EB AutoCore BSW modules.
- ASCOS-762 Fixed known issue: The documentation of the stack fill pattern is now in sync with the implementation. In chapter 1.7.6 of the User's Guide the value of the end-of-stack marker is incorrectly defined as 0xED. The value used in reality is the same as the stack-fill value (0xEE).

EB tresos AutoCore OS version 3.0.0

- Numerous bugs fixed
- Many new features implemented
- Generator accepts OIL files to full Autosar 1.3 spec.
- Calling context and interrupts disabled checks now fully implemented.
- Trusted functions now run in non-kernel environment
- Rate monitoring for task activations and ISRs is now implemented.
- Resource and interrupt lock timing for tasks and ISRs is now implemented.
- Nontrusted ISRs now fully implemented
- ISRs can now be killed individually. If an application gets killed at ISR level, ALL of that application's ISRs are killed.
- Multiple activations of tasks now run in order of activation
- Schedule tables can have multiple action points with the same offset value.
- Return from task (no TerminateTask() is now handled better; fully inside the kernel, so cannot be defeated by a badly-behaved nontrusted task.
- StopScheduleTable() now returns E_OS_NOFUNC if the schedule table is not running.
- Partial support for TC1775B and TC1920A added; GPTU driver supports hardware counters but not run-time/locktime monitoring.
- Kernel error table now only has one STATUS mode (EXTENDED or STANDARD). One of two tables is selected during configuration. This reduces the amount of ROM needed for the table.
- Execution time exceeded handler can now differentiate between execution-time budget and lock-time.
- Nesting-prevention in error hooks is now more selective.
- ShutdownOS() can no longer be called from a nontrusted app
- Several error return codes modified in the error database
- GetActiveApplicationMode now returns a NULL value in error cases
- GetEvent() can now be called from all autosar-specified contexts.
- ChainScheduleTable() now works when the second ST is on a different counter
- Schedule tables with an event at offset 0 now work correctly.
- KERNELSEMANTICS=[AUTOSAR,OSEK] can be used to relax some of the overstrict calling-context and interrupts-disabled checking required by Autosar.
- The 2 global-time parameters to OSEKMP_KernSyncScheduleTable() and therefore also SyncScheduleTable() etc. were reversed; now in correct order as per Autosar 1.3
- Definition of TrustedFunctionIndexType is now correct.

- Definitions of IS_READABLE() etc. are now correct
- Now supports gcc as well as tasking
- Checker is now fully implemented
- PROTECTION=PARTIAL is now supported. Using this mode, memory protection is turned on but the kernel does not modify the contents of the CPM register nor any of the CPR register sets. This should permit limited debugging (breakpoints, single step) while still retaining protection against erroneous writes, provided that the debugger also plays fair with the protection registers.
- AUTOSAR-OS v1.5 implemented
- Incorrect text in error message for CYCLETIME < MINCYCLE for an AUTOSTARTed alarm corrected.
- Error message for unused synchronization parameters for a hardware timer changed to a warning.
- Alarms can now be autostarted with cycle time of zero (non-cyclic).
- The checker now ensures that the times between consecutive expiry points in a schedule table are within the limits imposed by the associated counter's MAXALLOWEDVALUE.
- The generator now accepts the value PARTIAL for the OS attribute PROTECTION.
- Support for TC1796 B-step (fixed STM) added. Startup code detects which version the STM is and selects the bugfix if necessary. If you are using an older TC1796 with the STM's interrupt connections twisted, AND you have developed your own startup code, ensure that you select the workaround AFTER the global data sections have been initialized. See boards/TriboardTC1796/board.c for details.
- Fixed a bug in OSEKMP_SetIsrResourceLevel that caused the incorrect interrupt level to be set when a resource that is shared by one or more ISRs was taken or released.
- The TriboardTC1766 linker script for the tasking compiler has been fixed to use a reserved portion of the internal DSRAM as data RAM. The previous version incorrectly used the overlay RAM. The size of the reserved region has been corrected to match the values specified in board.h.
- GetScheduleTableStatus() now returns SCHEDULETABLE_ASYNCHRONOUS for a schedule table that is configured for SYNCHRONOUS startup but is not running because global time is not available.
- Several minor modifications have been made to avoid complaints from the MISRA-2004 conformance checker.
- An error in SyncScheduleTable() that caused incorrect synchronization when the service was called between the last expiry point and the end-of-round of the schedule table has been fixed.
- The parameters to SyncScheduleTable() are now passed to the error handler in the correct order. This means that they appear as expected in the OSEKMP_errorStatus structure and in the defined OSEK access macros when used from the error hook.
- A schedule table that starts synchronously is now marked SYNCHRONOUS as soon as global time is provided.
- A possible counter overflow when a schedule table first gets global time has been avoided. This would not affect TRICORE because all its timers can measure the full range of the TickType variable.

- The synchronization parameters (STARTUP, MAX_INCREASE, MAX_DECREASE and PRECISION) are no longer in the COUNTER.TYPE attribute. Instead, they are in a new attribute SCHEDULETABLE.EXPLICIT_SYNCHRONIZATION. OIL files for systems that use synchronization will need modification.
- The template file created by the generator now has a protection hook if one is configured. However, application-specific hook functions are not present.
- The --iso=90 option is now used for compiling with Tasking. C99 constructs (long long and initialization of specific parts of constant unions) have been removed.
- A warning when compiling osekmp-configuration.c has been eliminated.
- The user's guide specifies the units (nanoseconds) for the execution budget, locking times and rate-monitor timelimits
- The kernel now supports up to 32 events per task (was previously 16)
- The unused interrupt vectoring should now work correctly.
- The spurious "Incorrect nesting" error that could occur if a task's execution budget expired while it was in ResumeOSInterrupts should no longer occur.
- The generator should now work correctly for STACKOPTIMIZATION=NO and STACKOPTIMIZATION=WITHIN_APPLICATIONS.
- The return route from the application-specific error hook is now implemented correctly.
- SetRelAlarm() now always returns E_OS_VALUE when the increment parameter is zero, even when STATUS=STANDARD.
- GetApplicationID() should now return the expected value in all contexts.
- A non-trusted application no longer needs access rights to a counter to attach alarms or schedule tables to the counter, as this would also grant the application the right to increment the counter (in the case of a software counter).
- A schedule table in a non-trusted application can now activate tasks in another application provided that the task lists the schedule table's application as an ACCESSING_APPLICATION.
- A system with no alarms but one or more counters will now compile correctly.
- Interrupts get re-enabled correctly in OSEKMP_KernSyncScheduleTable()
- The generator now emits the correct arbitration-cycles value.
- The Users Guide now refers to INVALID_OSAPPLICATION.
- In the configuration database files, the array indices that were specified with a leading zero (thus causing the compiler to treat them as octal numbers) are now correctly specified without the leading zero. (RT-Id: 5303)
- The default value of the STACKOPTIMIZATION attribute of the OS object is now GLOBAL.
- The application-specific error hook now returns correctly via the trap handler when KERNELSEMANTICS=AUTOSAR. (RT-Id: 5266)

- The schedule synchronization sub-attributes (STARTUP, MAX_INCREASE, MAX_DECREASE and PRECISION) have been moved from the COUNTER.TYPE attribute to a new attribute EXPLICIT_SYNCHRONIZATION of the SCHEDULE_TABLE as specified in AutosarOS v1.5. (RT-Id: 5254)
- All internal symbols have been renamed: OSEKMP_xxx and OSEKAS_xxx are now OS_xxx, osekmp_xxx and osekas_xxx are now os_xxx. The system reserves any symbol beginning with "OS_-", "os_-", "Os_-" or "oS_-" for its own use: do not declare symbols with these prefixes in any user files (including OIL files).
- ORTI files are now generated for debugging support.
- Online help is now fully integrated in the eclipse framework.
- The OIL implementation file that is included at the start of every user's OIL file has changed its name. The new name is <ARCHITECTURE>_<DERIVATIVE>_AutosarOS.oil, where <ARCHITECTURE> and <DERIVATIVE> depend on your chosen microcontroller. For example: TRICORE_TC1796_AutosarOS.oil. If you have old OIL files, they will need editing by hand before they can be loaded into tresos. The recommended way is to first generate a new OIL file to discover the correct name for the include file.
- The OS can now be delivered as source code. If the source code is available an optimized library can be built, which can result in a much smaller and faster kernel, depending on the configuration.
- The new system services EnableInterruptSource() and DisableInterruptSource() are available in all scalability classes. This is because otherwise there is no safe way for an application to enable and disable interrupts; doing so directly might conflict with the kernel's management of mask registers etc., and in any case might be forbidden for non-trusted applications due to the processor's protection mechanisms.
- The kernel now supports the use of the Autosar MemMap specification if the compiler macro OS_MEMMAP is set to 1 on the compiler command line. However use of the MemMap feature is not recommended as it may conflict with other memory layout requirements such as those for memory-protected systems.
- The MAXALLOWEDVALUE for hardware counters is now taken from the derivative configuration file. The value in the OIL file is not used; if it is different from the configured value a warning is given.
- The OIL semantics have changed significantly in Autosar 2.0. For a full description of the OIL see the Autosar 2.0 specification and the Users' Guide. The main changes are listed below:
- In Autosar 2.0 OIL the OIL version is 3.0.
- In Autosar 2.0 OIL, in the APPLICATION object, the list of TRUSTED_FUNCTIONs is a sub-attribute of the TRUSTED attribute and is only present when TRUSTED is TRUE. The TRUSTED_FUNCTION list is a list of booleans with further subattributes on the TRUE values. The name of the trusted function is in the NAME sub-sub-attribute. 3SOFT has added a further implementation-specific sub-sub-attribute called STACKSIZE.
- In Autosar 2.0 OIL all the timing protection attributes of the TASK and ISR objects are gathered together as sub-attributes of a boolean attribute called TIMING_PROTECTION, and are only present when TIMING_PROTECTION is TRUE.
- In Autosar 2.0 OIL the TIMELIMIT and COUNTLIMIT attributes of the TASK object have been replaced by TIMEFRAME and TIMELIMIT attributes to specify the new accumulated execution-time limit. These are not used; see "Deviations" below.

- In Autosar 2.0 OIL the LOCKINGTIME INTERRUPTLOCK value of the TASK and ISR objects each have an additional parameter ALLINTERRUPTLOCKTIME. This is not used; see "Deviations".
- In Autosar 2.0 OIL the COUNTER object has a new attribute, UNIT, which specifies the units for the times given in other OIL objects associated with this counter. The UNITS attribute does not affect the parameters to system calls such as SetRelAlarm() etc. These parameters are defined in ticks.
- In Autosar 2.0 OIL the SCHEDULETABLE object has an AUTOSTART attribute that is similar to the AUTOSTART attribute for the ALARM object of standard OSEK.
- In Autosar 2.0 OIL the EXPLICIT_SYNCHRONIZATION attribute of the SCHEDULETABLE object has been renamed to LOCAL_TO_GLOBAL_TIME_SYNCHRONIZATION. The STARTUP sub-attribute has been removed but is still present as an implementation-specific attribute. New sub-attributes SYNC_STRATEGY, MAX_INCREASE_ASYNC and MAX_DECREASE_ASYNC have been added.
- In Autosar 2.0 OIL the PERIOD attribute of the SCHEDULETABLE object has been replaced by a LENGTH attribute and a boolean PERIODIC attribute. The LENGTH attribute defines the length of the schedule table; for periodic schedule tables it is the period, for non-periodic schedule tables it is the length of time after the notional zero-point for which the schedule table is considered to be running even though it has no more actions to perform.
- The Autosar 2.0 specification introduces several changes to the syntax and semantics of system services. See the Autosar specification and the Users' Guide for full details of the new syntax and semantics.
- In Autosar 2.0, the specification no longer forbids two or more schedule tables to be active simultaneously on the same counter.
- In Autosar 2.0, StartScheduleTable() has been renamed StartScheduleTableRel(), and a new service, StartScheduleTableAbs(), has been added.
- In Autosar 2.0, StartScheduleTableRel() does not accept zero as the 'offset' parameter.
- In Autosar 2.0, SyncScheduleTable() no longer has a localtime parameter. The caller must coerce the raw global time into a form suitable for the schedule table.
- In Autosar 2.0, smooth synchronization happens throughout the schedule table by adjusting the time of every event. Smooth synchronization implies asynchronous startup unless the implementation-specific STARTUP attribute specifies otherwise.
- In Autosar 2.0, hard synchronization happens at the end of the round and will always adjust in one step unless the MAX_INCREASE/MAX_DECREASE attributes specify otherwise (implementation-specific semantics). Hard synchronization implies synchronous startup unless the implementation-specific STARTUP attribute specifies otherwise.
- In Autosar 2.0, the MAX_INCREASE_ASYNC/MAX_DECREASE_ASYNC parameters are used when the schedule table is asynchronous. The "shortest route" synchronization method is only used when the schedule table is determined to be asynchronous to avoid forcing a synchronous schedule table into asynchronism.
- In Autosar 2.0, the synchronous/asynchronous state of a schedule table is determined exclusively by the SycnScheduleTable() service. This means that a schedule table that is determined to be asynchronous

will remain so even if the amount of adjustment remaining reduces to less than the PRECISION (or even to zero), unless SyncScheduleTable() is called again.

- In Autosar 2.0, when chaining schedule tables using NextScheduleTable(), the two schedule tables must be attached to the same counter, and the 'current' schedule table cannot be in the STOPPED or NEXT states.
- In Autosar 2.0, when chaining using NextScheduleTable(), the new schedule table that is to be chained replaces any other schedule table that is already chained after the 'current' schedule table. The previous behavior in this case was not defined, and the 3SOFT implementation chose to return an error code.
- In Autosar 2.0, GetScheduleTable() returns five possible values to better describe the state of the schedule table.
- In Autosar 2.0, the TerminateApplication() system service accepts a RESTART option.
- In Autosar 2.0, new system services EnableInterruptSource() and DisableInterruptSource() are available.
- Hardware counters now have no inherent limits on the number of alarms they can support simultaneously. This means that multiple simultaneous schedule tables on the same alarm are supported, at the same time as one or more normal alarms.
- Because a hardware counter can now be used alone as a system counter, the generator must generate the macros OSMAXALLOWEDVALUE, OSMINCYCLE, OSTICKSPERBASE and OSTICKDURATION based on a hardware counter. To maintain compatibility with existing OIL files, if there is a "system-counter-chain" consisting of an autostarted alarm that is attached to a hardware counter and incrementing a software counter on expiry, the first such chain is used as the system counter. Otherwise the first hardware counter is assumed to be the system counter. The standard OSEK warning also applies here: if there is more than one counter that could be regarded as a system counter, it is not specified which counter shall be chosen. Therefore it is always safer to use the counter-specific macros (OSMAXALLOWEDVALUE_<countername> etc.) instead.
- The generator now defines the macros OS_TicksToNs_<countername> and OS_NsToTicks_<countername> for every hardware counter. These macros can be used whenever a time value must be converted to ticks, and vice-versa. The macros will give the best possible accuracy without using multiple-precision or floating-point arithmetic, and will not overflow internally, although it is possible that the final result may overflow in the TicksToNs case.
- The linker scripts for the Tasking compiler have been greatly simplified. At the same time, the selection of memory regions for the CSA list has been moved entirely into the linker script - it was previously in the linker script AND in board.h, which meant that the two had to be kept in sync. The disadvantage is that there can be no compile-time check that there are sufficient CSAs allocated. A startup-time check would be possible.
- When an interrupt gets disabled because it exceeds its arrival rate, the kernel now re-enables it when the time-frame has elapsed. To achieve this, several visible changes have been made:
  - The alarm ID type is now 16-bit (was 8-bit)
  - An extra OIL attribute in the OS (RATE_MONITOR_COUNTER) has been added. This specifies the COUNTER object that will be used for rate monitoring. The counter must be a HARDWARE

counter and will have an alarm attached for each ISR with a rate-limit. The designated RATE_MONTOR_COUNTER can also have normal user-objects attached (alarms, schedule-tables).

  - To avoid confusion, the architecture-specific <arch>_PROTECTION_TIMER attribute has been renamed to <arch>_EXECUTION_TIMER. The timer cannot be associated with a hardware counter.
  - Because the rate-monitor counter and the execution timer are now associated with different hardware timers (and therefore perhaps different frequencies), there are now different NsToTicks() macros for the two purposes.

- The TRICORE_PROTECTION_TIMER attribute (OS object) has been renamed to TRICORE_EXECUTION_TIMER to make it explicit that this timer is only used for the execution-time aspects of timing protection.
- The bug that resulted in a task or ISR being able to extend its execution budget by calling GetResource/ReleaseResource or SuspendOSInterrupts/ResumeOSInterrupts nested inside GetResource/ReleaseResource has been fixed.
- Aggregate execution-time limit in specified time-frame has been implemented for tasks. This replaces the previous rate-monitor.
- Worst-case execution-time measurement for tasks and ISRs has been implemented. See the MEASURE_MAX_RUNTIME attributes for TASK and ISR objects, and the API functions OS_GetTaskMaxRuntime() and OS_GetIsrMaxRuntime().
- A bug in the Generator has been fixed. This bug caused the generator to produce uncompilable code under certain circumstances - certain OIL objects with long names.
- A problem with the OSEK API definition has been fixed. The macros DeclareAlarm(), DeclareEvent() and DeclareResource() were previously defined as empty, but some compilers complain about the resulting lone semicolon on strict warning/error settings. The new macros simply declare a harmless function prototype.
- ASCOS-175 Fixed known issue: The corrupted task state that occurred if and event was set in an ISR that occurred during the WaitEvent() call for that event has been fixed.
- Prototypes for the following functions added to Os_proosek.h: OS_GetUnusedIsrStack(), OS_GetUsedIsrStack(), OS_GetUnusedTaskStack(), OS_GetUsedTaskStack()
- ASCOS-194 Fixed known issue: The Generator now generates the correct BISRPRIO macros for all category 2 and kernel interrupts. This means that nested Cat2 interrupts can no longer discard the lower priority interrupt before the ISR is called.
- Several changes have been made to the architecture-independent code with the aim of speeding up the kernel, especially when an optimized kernel is built.
- ASCOS-189 Fixed known issue: The IS bit in the PSW is now set to the same value as in the calling context when a trusted function is called. This prevents spurious stack errors when stack checking is enabled.
- When a hardware counter is configured with units of NANOSECONDS, the various times that are specified in nanoseconds are no longer compared against the timer attributes MAXALLOWEDVALUE and MINCY-

CLE, which are still specified in ticks. This means that at the moment there are no checks on the validity of such times. Be careful to ensure that all times are within the capabilities of the counter.

- The amount of stack required for the kernel and interrupts is now calculated depending on the potential nesting depth of the error handler, which in turn depends on various configuration parameters. The end result is that the kernel/interrupt stack is smaller.
- The optimization macro OS_EXCLUDE_TIMINGPROTECTION is now correctly dependent on whether execution-timing is in use. Previously it depended simply on the scalability class, which meant that the timing protection functions were called, but no timer driver was present.
- There is now a prototype for OS_UserSetScheduleTableAsync(), which eliminates a compiler warning whenever SetScheduleTableAsync() is used.
- The Os generator is now based on the Datamodel. This includes reading of .oil, .epc and .xdm files.
- The make-environment is updated to work together with asc_Make plugin.
- The plugin makefiles are renamed from AutosarOS_ to Os_-.
- A new version of the Datamodel is included. Default values for many Parameters are defined.
- Added License support.
- ASCOS-192 Fixed known issue: Rate monitor checks have been implemented in ChainTask().
- ASCOS-230 Fixed known issue: The kernel function OS_KernGetStackInfo() now gives results for the kernel stack when the requested object is an ISR. This means that the library functions getUsedIsrStack() and getUnusedIsrStack() should now function correctly.
- The PostTaskHook() function is now called when a running task is killed. This means (among other things) that the PostTaskHook() is correctly called when a task returns from its top-level function without calling TerminateTask() or equivalent.
- ASCOS-284 Fixed known issue: The function OS_RequeueUp() (lib_src/kernel/kern-requeueup.c) has been rewritten to avoid corrupting the task queue if the current task is no longer at the head of the ready list during GetResource(). This can only occur if the task is enqueued in an ISR the interrupts the first part of the GetResource() service.
- ASCOS-302 Fixed known issue: The global accounting structure is now cleared down correctly on return from an ISR if the interrupted context had no timing protection active.
- ASCOS-300 Fixed known issue: The interval for cyclic alarms is now correct. When requeueing a cyclic alarm, OS_RunCounter() uses a new version of OS_EnqueueAlarm() that does not take the counter's current error term into account.
- ASCOS-348 Fixed known issue: Os_configuration.c now determines correctly whether the Increment-Counter system call should be omitted. Previously the presence of a protection timer was not taken into account, and so the service was omitted if there was only one counter of type SOFTWARE.
- ASCOS-14 Fixed known issue: When an ISR causes an error and the resulting application-error-hook then causes a protection error, the ISR now gets killed properly.

- The coupling between the MAXALLOWEDVALUE of a hardware counter and the physical timer's wrap-around value has been removed. It is now possible to set a MAXALLOWEDVALUE of any size up to the range of the TickType variable (os_tick_t, 32-bit on all current architectures). When a duration of more than half of the timer's wrap value must be timed from one alarm to the next, it is done in steps of half of the timer's wrap value. This value is chosen to minimize the risk of losing a full timer round under extreme circumstances.
- ASCOS-183 Fixed known issue: TerminateApplication(RESTART) now works correctly when called from a category 2 ISR belonging to an application. Previously the application got terminated but the restart task didn't get activated.
- ASCOS-184 Fixed known issue: TerminateApplication() now works correctly when called from an error hook. From the global ErrorHook() it results in an error (E_OS_CALLLEVEL). From an application error hook, the application gets correctly terminated and the service does not return to the caller.
- ASCOS-397 Fixed known issue: Every task's stack is now initialized during StartOS(), even for tasks that share a stack with another task that has already had its stack initialized. There are two reasons for this: 1. If the shared stacks are of different sizes only the smaller might get initialized, and 2. The behavior after a warm-start is not consistent.
- ASCOS-185 Fixed known issue: The macro for detecting if an ISR was killed did not work because the ISR calling mechanism cleared down the stack limit. This macro has now been implemented in a architecture-independent way independently of the stack-limit variable to avoid the problem. Error notification for ISRs that terminate without releasing resources or re-enabling interrupts now works correctly.
- ASCOS-188 Fixed known issue: Pointers passed to system services like GetEvent() are now checked for alignment as well as writeability. This means that it should no longer be possible for non-trusted code to cause an exception in the kernel on systems with memory-protection enabled. Trusted code can still manage it on some systems (for example by passing an address that causes another type of exception like non-existent memory or protected flash).
- ASCOS-414 Fixed known issue: When a schedule table action SETEVENT is used the generator now takes the correct event from the configuration. Before only the first event was used. Unknown events used in schedule tables are now reported by the checker.
- ASCOS-415 Fixed known issue: OS_KernSuspendOsInterrupts(), OS_KernResumeOsInterrupts() and OS_LeaveKernel() now use OS_IntDisableConditional() to avoid inadvertently lowering the interrupt level when trying to disable interrupts for the kernel.
- The prototypes of alarm callbacks are now generated in os.h instead of in os-config.h.
- The feature to remove selected API functions now conforms to the OS naming convention: os_api_remove.h has been renamed to os-remove-api.h and the name of the make option in Os_cfg.mak has been changed from OS_USE_ADVANCED_API_REMOVE to OS_USE_API_REMOVAL.
- Changed all parameters with multiplicity 0..1 to optional parameters, which improves usability during configuration.
- Tresos Workspace for os application template has been added.

- ASCOS-435 Fixed known issue: The OS Generator no longer changes the internal configuration when sorting the tasks. This issue improves interoperability with EB RTE Generator.
- ASCOS-468 Fixed known issue: The validity of the RestartOption parameter to TerminateApplication() is now the first thing that is checked in OS_KernTerminateApplication(). This means that in the case of a double error (e.g. an TerminateApplication() called from an invalid context with an invalid parameter), the error code will be different.
- The COUNTER API has been changed to conform with the Autosar OS v2.1. specification. Two new system services, GetCounterValue() and GetElapsedCounterValue(), have been implemented; a GPT channel can be used to tick a software counter automatically, but not to drive a hardware counter (see: Deviations); PhysicalTimeType and UnitType datatypes are now available; the required OS_TICKS2uuu_ccc macros are available (where uuu is unit type and ccc is counter name).
- The synchronization API has been changed to conform with the Autosar OS v2.1 specification. Synchronous startup is no longer available as a configuration option; the system service StartScheduleTableSynchron() has been implemented; synchronization attributes MAX_CORRECTION_SYNC and MAX_CORRECTION_ASYNC have been implemented. The old synchronization attributes MAX_INCREASE and MAX_DECREASE (and their _ASYNC counterparts) have been retained as implementation-specific attributes to maintain compatibility with systems designed for OS v2.0 and with older OSEK/Time behavior. If the old parameters are specified, they are used in preference to the new ones; however, their values must lie within the limits specified by the new parameters.
- System services that do not return a StatusType now no longer call the ErrorHook by default, in accordance with Autosar OS v2.1. However, it is possible to enable the ErrorHook using an EB-specific configuration parameter. Application-specific error hooks are now never called for these System Services.
- A lot of new optimization parameters have been introduced. Some of them replace the old KERNEL_SEMANTICS option so that finer control over non-conformant optimizations can be achieved. Others are new optimizations. WARNING: Use of the optimizations results in a kernel that is not 100% Autosar-compliant.
- The method of calling the initialization functions at startup has been changed. The array length is now variable (configuration-dependent). This results in a slightly quicker startup for smaller systems and uses a little less ROM.
- Interrupt lock timing protection has now been implemented for SuspendAllInterrupts() and DisableAllInterrupts(). All three suspend/disable services are now implemented by a single OS API OS_UserSuspendInterrupts(), and all three resume/enable services are implemented by a single OS API OS_UserResumeInterrupts(). On system-call architectures this results in fewer function entries in the system-call table. Apart from interrupt lock timing the semantics are the same.
- The configuration data for previous OS releases is included into the OS plugin, allowing the Tresos configuration importer to import old configurations into new projects that use the current OS release.