RTA-OS

Release Note - Version 12.0.0 (17-08-2022)

Copyright

Document: 10766-RN-12.0.0 EN-08-2022(17-08-2022)

Safety Notice

This ETAS product fulfills standard quality management requirements. If requirements of specific safety standards (e.g. IEC 61508, ISO 26262) need to be fulfilled, these requirements must be explicitly defined and ordered by the customer. Before use of the product, customer must verify the compliance with specific safety standards.

Contents

1 Introduction

RTA-OS is an AUTOSAR compliant Operating System and associated tooling.

The RTA-OS configuration and generation tools are Microsoft Windows programs that are supported on 8, 10 and 11. Full hardware and software system requirements are documented in the *Getting Started Guide*.

1.1 Version Information

This is Version 12.0.0 of RTA-OS.

RTA-OS is compliant with AUTOSAR OS (R3.0.1 -> R3.0.7, R3.1.1 -> R3.1.5, R3.2.1 -> R3.2.2, R4.0.1 -> R4.6.0 (R20-11)) scalability Class 1-4.

1.2 Installation

The installation process is covered in detail in the *Getting Started Guide*.

2 Open EHI Calls

Open issues are referred to by their call number in the ETAS Helpdesk International (EHI) system.

No EHI calls are open.

3 Change History

3.1 Version 12.0.0

Additional Features

The following features have been added to this release:

- There is a new ’Release Notes’ report that can be used to generate a machine-readable release note that contains information from both the OS Tools and the Target Port.
- When the OS option ’IOC Atomic Copy Size’ is active, the OS is able to take input from 2 optional files. The files should contain the names of IOC communications, one per line. Any communication in the ioc_atomic_exclude.txt file is excluded from the optimization. Any communication in the ioc_atomic_include.txt file is included in the optimization (regardless of its size). RTA-OS looks for the files in the ’project’ directory - which is likely to be the directory containing the first .arxml or .rtaos file passed to RTAOSGgen.
- The IOC Summary Report has been updated to show the reason for inlining optimizations.

Modified Features

The following features have been modified in this release:

- The OS tools are based on .NET 6 technology. This replaces the .NET framework technology used previously. Do not install this version over the top of earlier versions.
- The Reference Guide now lists those OS APIs that can be called from Os_Cbk_Idle().
- The MemMap section for Os_const_tasks0 has been adjusted to add _LOCAL in the single-core case for relevant AUTOSAR versions.
- An error is raised if –ioc:impl and –ioc:stub are used at the same time.
- The IOC initialization is optimized to use the memory clear function Os_ioc_memclr instead of individual assignments (may depend on target). Os_ioc_memclr has the same features as Os_ioc_memcpy.

Removed Features

No features have been removed from this release.

3.2 Version 6.2.200

Additional Features

No features have been added to this release.

Modified Features

The following features have been modified in this release:

- Remove the possible double reporting of the error message ’Status must be EXTENDED in SC3 or SC4’
- IOC internal data declarations emitted are in size order - largest first - to try to minimize fragmentation
- TrustedFunctions are declared in the MemMap section of the owning OS App
- Range checking for peripheral access APIs considers both start and end range
- Os_PeripheralAddressType size fixed at ’large integer’ size
- Core-specific task constants are placed in per-code memory sections.
- Corrected documentation example for Os_Cbk_SetMemoryAccess.
- Documentation for GetElapsedValue shows that it can return E_OS_VALUE.
- Documentation for GetActiveApplicationMode states that it may be called before StartOS().

Removed Features

No features have been removed from this release.

3.3 Version 6.2.0

Additional Features

The following features have been added to this release:

- Adds support for AUTOSAR 4.6.0 (R20-11).
- Added note that only documented OS APIs are considered to be public. Functions and data in OS header files that are not documented should not be used in non-OS code.
- Added OS option ’--incremental’. File saves will not overwrite an existing file if its contents are identical.
- Improved Configuration Summary content. Includes RESOURCE core assignments and OS object indexes.
- Improved IOC Summary content. Includes supplied IOC configuration options.
- OS options ’Omit Default Implementations’ and ’Omit Default IOC copy’ can be used to omit the default implementation of Os_ioc_memcpy from the library.
- OS option ’IOC Atomic Copy Size’ has been added to support (rare) cases where the compiler can emit code that will atomically copy values larger than integer size.

- Adds support for IOC ReceiverId that was included in AUTOSAR R20-11.
- The IOC API function IocInit() is added. It does not have to be called by user code. The OS will correctly initialize the IOC.

Modified Features

The following features have been modified in this release:

- Avoid emitting the same warning or error message more than once.
- Up to 63 events are supported per TASK. Previously 32.
- GetTaskActivationTime now returns E_OS_ILLEGAL_ADDRESS if Value references an invalid address (EXTENDED mode).
- Checking for cross-core notifications is faster and is moved to core-local code where appropriate.
- The implementation of the OS_TASKTYPE_TO_INDEX and OS_ISRTYPE_TO_INDEX macros have been changed to avoid MISRA warnings in customer code.
- Allow selection of options ’core-local data’ and ’always clear OS data’ at the same time. Previously this could cause compilation to fail.
- Do not use VAR() in declaration of structure members. e.g. terminate_jump_buf in Os_TaskDynType.
- Do not use FUNC() in typedefs. e.g. Os_CounterStartType.
- The default implementation of the TASK and ISR macros can be changed by predefining them before Os.h. (May be useful in some tests.)
- An extra interrupt lock is added to SuspendOSInterrupts() to protect against a possible race condition if it is called by Category 1 ISRs. (Category 1 ISRs never need to call SuspendOSInterrupts() because they already run at a high interrupt priority.)
- The file Os_ScheduleQ.c is omitted from the OS library unless needed because the ’Optimize Schedule()’ option is TRUE.
- The OS variable Os_lock_apperrorhook is only declared when it is used in the library.
- Documentation is updated to state that E_OS_MISSINGEND can be emitted in STANDARD mode for TerminateTask().
- Documentation is updated to state that GetAlarm only returns E_OS_ILLEGAL_ADDRESS in EXTENDED mode.
- Documentation is updated to add a caution in StartScheduleTableRel() usage.
- Documentation is updated to add a note in CallTrustedFunction() related to timing protection.

- Changed messages and reports from the OS library to use the spelling OS-Application (with the hyphen) more consistently.
- ORTI files contain correct TASK references in single-core targets.
- Corrected the schemaLocation in BSWMD reports for AUTOSAR versions 4.3.1 onwards.
- Added memory sections in BSWMD report that were previously omitted because they were local to individual C files.
- Some IOC messages have changed from warnings to information.
- The OS will not emit warnings for trusted functions created by RTA-OS to support IOC code.
- Initialization of the IOC is faster for cases where there are many cross-core notifications.

Removed Features

No features have been removed from this release.

3.4 Version 6.1.3

Additional Features

The following features have been added to this release:

- As 64 bit versions of the tools are used by default, equivalent 32 bit versions are provided for compatibility with older PCs. These are named rtaosgen32.exe and rtaoscfg32.exe.

Modified Features

The following features have been modified in this release:

- The ORTI Api ID is correctly now reset on exit from TerminateTask(). Previously it only got reset in error cases.
- An unnecessary IOC warning has been removed. The warning saying that a sender has the same OS-Application as a receiver is no longer emitted if there are multiple senders because that is a valid use-case.

Removed Features

No features have been removed from this release.

3.5 Version 6.1.2 (Preview)

Additional Features

The following features have been added to this release:

- Internal support has been added to support parallelization of compilation (where possible). Future target ports will need to be updated to take advantage of this.

Modified Features

The following features have been modified in this release:

- The engine used in the OS tools to generate the OS code has been updated. Code generation will be faster on current versions of Windows.
- 64 bit versions of the tools are used by default. These can access more memory so can cope with bigger input configurations.
- The tools no longer need .NET 3.5 to be installed on the host PC, provided that version 4.7 or above of the .NET Framework is present. (Note that Windows 10 version 1511 and lower cannot install 4.7 so will need .NET 35 support.)
- Minor adjustment in character escaping in the XML and HTML versions of the Options Report.
- VSMD files have been updated and excess information has been removed.

Removed Features

No features have been removed from this release.

3.6 Version 6.1.1

Additional Features

No features have been added to this release.

Modified Features

The following features have been modified in this release:

- The example code in the documentation for AdvanceCounter was corrected (State.Delay => State->Delay).
- The documentation for the hardware counter callback Os_Cbk_State_() was updated to reflect how it is currently used.
- Parts of the the RTA-OS tooling have been updated to use more recent releases of the .NET Framework. The tools now use .Net version 4.7.2. Experimental 64-bit versions of rtaosgen and rtaoscfg are provided, but RTA-OS targets need to be updated before they will be recognized by these tools.

Removed Features

No features have been removed from this release.

3.7 Version 6.1.0

Additional Features

The following features have been added to this release:

- Support has been added for AUTOSAR 4.5.0 (AUTOSAR R19-11).
- It has been noted that TASKs in real-world full-stack AUTOSAR systems are often not written in such a way that they can safely cope with being terminated part way through execution. If the OS sees that the configuration relates to a full-stack AUTOSAR system it will remove the ability to terminate part-completed tasks unless you set the OS option ’Allow forced TASK terminations’ to TRUE to confirm that this is safe to do. This means that errors such as timing budget overruns will result in shutdown or controlled reset of the ECU rather than task terminations. This replaces ’Allow RTE TASK terminations’.
- The OS option ’Always clear OS data’ has been added so that any OS data normally placed in NO_INIT memory sections is moved to CLEARED memory sections.
- Added E_OS_SYS_ERROR_LIMIT to mark limit of expected StatusType values.
- Os.h can be #included directly in C++ files for compilation. Note that not all target ports are currently compatible with C++ compilation.
- The TYPE-EMITTER value in an IMPLEMENTATION-DATA-TYPE is used to determine which header files to include in Ioc.h.
- OS Option ’Task Termination Hook’ can be used to cause Os_Cbk_TaskTerminated(TaskID) to get called at Task termination.
- The command-line option ’--ar_version:x.y.z’ has been added to rtaosgen.exe to specify the active AUTOSAR version. This overrides the version seen in the ARXML configuration.
- A sample file has been added to help show how to specify core IDs in AUTOSAR 4.2 and above. Use option --samples:[EcucCoreDefinition].
- The name and version of the target port get output in the rtaosgen signon message.

Modified Features

The following features have been modified in this release:

- Removed an unnecessary precautionary check in ActivateTask for multicore systems without crosscore activations.
- CheckObjectAccess did not detect being passed INVALID_OSAPPLICATION.

- StartCore did not need to check for non-OS cores when all cores are OS cores.
- StartNonAutosarCore code eliminated when all cores are OS cores.
- Other small optimizations in cases where there are no Non-Autosar cores.
- Some E_OS_ACCESS checks eliminated in systems where Applications cannot be terminated.
- GetVersionInfo could only be accessed as Os_GetVersionInfo.
- Speed/memory optimization for the generation of ScheduleTables with expiries that set many Events.
- MISRA report refactored for speed. Especially relevant for systems with many MemMap sections.
- Eliminated a compiler warning in IOC stub mode for fully trusted systems.
- Eliminated a compiler warning relating to pointers to const psets that was seen on some compilers when there were more than 32 tasks on a core.
- RTA-OS GUI IOC Summary Report can now be run without rebuilding the ECU.
- Error E_OS_CORE can be passed to ErrorHook if an Alarm or ScheduleTable tries to activate a Task on a core that is in shutdown.
- Error E_OS_ACCESS can be passed to ErrorHook if an Alarm or ScheduleTable tries to activate a Task on belonging to an OsApplication that is not accessible.
- The Address field of ApplicationContext argument in the Os_Cbk_SetMemoryAccess callback now always contains the starting stack value. Previously it was zero unless StackMonitoring was enabled. (Note that the Size field will still be zero unless StackMonitoring is enabled.)
- The adjustment of the MemMap section names based on the AUTOSAR version is modified slightly to be more AUTOSAR compliant.
- The Os_BSWMD.arxml report has been enhanced based on customer feedback.
- Parts of the the RTA-OS tooling have been updated to use more recent releases of the .NET Framework. The tools now use .Net version 4.5.2.
- The version of the ETAS License Manager that is provided has been updated to v1.8.0.
- The definition of RUNNINGTASK in the ORTI file has been corrected in single-core configurations.

Removed Features

No features have been removed from this release.

3.8 Version 6.0.0

Additional Features

The following features have been added to this release:

- RTA-RTE does not support termination of its tasks part way through execution. If the OS sees that the configuration contains RTE-related information it will now remove the ability to terminate part-completed tasks unless the OS option ’Allow RTE TASK terminations’ is set to TRUE. This means that errors such as timing budget overruns will result in shutdown or controlled reset of the ECU.

Modified Features

The following features have been modified in this release:

- Small refinements and corrections to the preview IOC code in 5.91.0.
- Split internal Ioc.c file into two files to reduce memory requirements during code generation.

Removed Features

No features have been removed from this release.

3.9 Version 5.91.1 (Preview)

Additional Features

No features have been added to this release.

Modified Features

The following features have been modified in this release:

- MemMap sections containing the text ’CODE_LIB’ are changed to contain the text ’CODE’ for AUTOSAR versions 4.1 and above.
- The MemMap ’CN_CODE’ callout style introduced for AUTOSAR version 4.1 is kept to 4.1 only. It is removed from 4.2 and above. The MappingCodeLocation feature can be used for callouts above version 4.1.
- MemMap sections containing the text ’COREn_CODE’ or ’COREn_CODE_LIB’ are changed to contain the ’COREn’ part at the end.
- MemMap sections containing the text ’COREn’ have the additional modifier ’_LOCAL’ added at the end for AUTOSAR 4.3 versions and above.

Removed Features

No features have been removed from this release.

3.10 Version 5.91.0 (Preview)

Additional Features

The following features have been added to this release:

- VSMD files can be created for specific versions of OS and target plugins. This is primarily to support ETAS ISOLAR tooling.

Modified Features

The following features have been modified in this release:

- When OS option 'IOC Data' is set to 'Writer' the IOC code can optimize communications containing data types are copyable atomically - e.g. uint32. Fewer interrupt locks will be taken and some calls may become inlinable macros.
- The implementation for IOC queued communications has been changed to remove the need for interrupt and cross-core locks to manage the queue data. When OS option 'IOC Data' is set to 'Writer' some queue control data is assigned to the writer's OS Application and some to the reader's.
- To support the removal of locking in IOC queues, IOCEmptyQueue_xxx APIs are now specified such that they should not be called if corresponding read/writes could occur at the same time.
- The AUTOSAR TotalNumberOfCores value has been reimplemented as Os_TotalNumberOfCores so that all OS data starts with an Os_ prefix. A macro TotalNumberOfCores maps on to Os_TotalNumberOfCores so no code changes are needed.
- Some explanations in MISRA reports have been updated for clarity.

Removed Features

No features have been removed from this release.

3.11 Version 5.7.3

Additional Features

No features have been added to this release.

Modified Features

The following features have been modified in this release:

- No changes since 5.7.2 preview.

Removed Features

No features have been removed from this release.

3.12 Version 5.7.2 (Preview)

Additional Features

No features have been added to this release.

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.13 Version 5.7.1 (Preview)

Additional Features

No features have been added to this release.

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.14 Version 5.7.0

Additional Features

The following features have been added to this release:

- The macro OS_INDEX_FOR_taskname has been added. This can be useful in switch/case statements.
- The macro OS_INDEX_FOR_isrname has been added. This can be useful in switch/-case statements.
- Allows time intervals to be used in the Schedule Table Assistant GUI.

Modified Features

The following features have been modified in this release:

- The definition of RUNNINGTASK in the ORTI file has been modified slightly in multicore configurations to make it easier for debuggers to support TOTRACE operation.

Removed Features

No features have been removed from this release.

3.15 Version 5.6.90 (Preview)

Additional Features

The following features have been added to this release:

- Support has been added for AUTOSAR 4.4.0, including the APIs ActivateTaskAsyn and SetEventAsyn. Note that OS option 'Support ActivateTaskAsyn and SetEventAsyn' has to be used to enable them.
- Where there is ECUC partition information present in the ARXML configuration, the RTA-OS GUI allows OS Applications to select the partition that they relate to.
- MEMORY-SECTION information has been added to the Os_BSWMD.arxml report.
- Added new OS option 'Omit Default Implementations'. RTA-OS normally provides default implementations for Os_Cbk_Idle, Os_Cbk_InShutdown and Os_Cbk_StackOverrunHook in the generated library. This option can be used to omit these functions from the library.

Modified Features

The following features have been modified in this release:

- SetRelAlarm and SetAbsAlarm will return E_OS_CALLEVEL if Service Protection is enabled and they are called from Startup Hook. This is to conform to the AUTOSAR specification.
- Much larger IOC MCSD files can be processed.

Removed Features

No features have been removed from this release.

3.16 Version 5.6.4

Additional Features

No features have been added to this release.

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.17 Version 5.6.3

Additional Features

The following features have been added to this release:

- Information about interrupt vectors and target variants has been added to the Options Reference report
- In the OS GUI rtaoscfg, the interrupt addresses input field in OS Configuration / ISRs / General can have part of the name entered and it will suggest entries that contain this text. This is intended to help select interrupts more easily on larger processors.

Modified Features

The following features have been modified in this release:

- Removed duplicated line ’Os_AppOverride = INVALID_OSAPPLICATION;’ that could occur in StartOS.c and ShutdownOS.c depending on configuration. No functional change.
- Removed duplicated line ’Os_AppAccess = 1U;’ that could occur in StartOS.c depending on configuration. No functional change.

Removed Features

No features have been removed from this release.

3.18 Version 5.6.2

Additional Features

No features have been added to this release.

Modified Features

The following features have been modified in this release:

- Update ETAS address in About dialogs and documentation.

Removed Features

No features have been removed from this release.

3.19 Version 5.6.1 (Preview)

Additional Features

The following features have been added to this release:

- Supports C28xTI target port (16-bit chars).
- Reports missing data references in IOC Data Properties as an error.
- Schema and VSMD (ParamDef) added for AUTOSAR 4.3.1.
- Support reading non UTF8 MCSD files (IOC)
- MemoryMappingCodeLocationRef is supported on Hooks, OsApplication Hooks, Tasks and ISRs. This means that these objects can be located in individually configured MemMap sections.
- AllowAccess can raise E_OS_CALLEVEL
- ControlIdle can raise E_OS_CALLEVEL
- GetApplicationState can raise E_OS_CALLEVEL

Modified Features

The following features have been modified in this release:

- VSMD (ParamDef) files updated for conformance with customer tooling. e.g. alphabetical ordering
- Adjusted internal references in the supplied VSMD (ParamDef) files for each AUTOSAR 4 version.
- ActivateTask, StartScheduleTableRel, StartScheduleTableAbs, StartScheduleTableSynchron, SyncScheduleTable, Os_SyncScheduleTableRel and Os_SyncScheduleTableRel are not allowed in StartupHook (only checked when Service Protection is configured)
- Rename E_OS_XCORE_QFULL to E_OS_SYS_XCORE_QFULL
- Minor correction in IOC code to ensure that correct MemMap sections are used in declaration and implementation.

Removed Features

No features have been removed from this release.

3.20 Version 5.6.0

Additional Features

The following features have been added to this release:

- Tested on Windows 10.
- Peripheral Areas are supported. This is an AUTOSAR 4.3.0 feature. RTA-OS allows the feature to be used with any AUTOSAR version.
- Interrupt Source APIs are supported. This is an AUTOSAR 4.3.0 feature. RTA-OS allows the feature to be used with any AUTOSAR version. Your RTA-OS target port may need to be updated to support this feature.
- A new report ’Os_BSWMD.arxml’ has been added to allow the generation of a BSW Module Definition file for RTA-OS.
- RTA-OS Custom DLLs must now be enabled via an ETAS-provided key in the Auth.exe.config file.

Modified Features

The following features have been modified in this release:

- Revised BSWMD file based on feedback.
- Code generated for StartScheduleTableSynchron didn’t always check that the ScheduleTable was explicitly synchronized.
- Adjusted code and clarified documentation for GetElapsedTime and GetExecutionTime for consistent behavior in Pre and Post TASK hooks.
- Updated version of ETAS Licensing Manager is supplied. License controls are now placed in the Auth.exe.config file.
- MISRA 2012 11.4/11.6 dismissals added to OSError_ macros.
- Added check to Schedule() in case it is called from outside a TASK (not an allowed situation).
- Corrected the OSError_ macro definition for GetStackSize.
- Os_Cbk_CheckMemoryAccess was being passed the size of a pointer, not the size of the pointed-to item when used in some API calls.
- StartScheduleTableSynchron now correctly returns E_OS_ID when a non-explicitly synchronized table is passed in to it.
- ReleaseSpinlock now correctly returns E_OS_STATE when passed a NESTABLE spinlock that was not locked by the calling TASK or ISR.

Removed Features

No features have been removed from this release.

3.21 Version 5.5.11

Additional Features

No features have been added to this release.

Modified Features

The following features have been modified in this release:

- The Reference Guide now shows that E_OS_ILLEGAL_ADDRESS can be reported in both STANDARD and EXTENDED builds.
- Adjust IOC code used to copy data between structures so that bit-compatible but differently typed structures can be used.
- IOC initialization uses structure copies rather than memcpy if possible.
- ChainTask code adjusted to avoid a possible cross-core race condition that could delay the running of the chained TASK.

Removed Features

No features have been removed from this release.

3.22 Version 5.5.10

Additional Features

No features have been added to this release.

Modified Features

The following features have been modified in this release:

- Updated the documentation for Os_Cbk_CheckMemoryAccess to clarify its usage.
- The default implementation of Os_ioc_memcpy in the RTA-OS library can be replaced simply by the application code providing its own version.
- IOC code will use structure assignment rather than Os_ioc_memcpy when possible.
- The names of the memory sections created for the OS options ’IOC Data’ and ’IOC code’ has been modified since the preview.

Removed Features

No features have been removed from this release.

3.23 Version 5.5.9 (Preview)

Additional Features

The following features have been added to this release:

- The new OS option ’IOC Data’ can be used to control the MemMap data section allocation for IOC data buffers. This can be used to reduce untrusted/trusted code transitions.
- The new OS option ’IOC Code’ can be used to control the MemMap code section allocation for IOC APIs. APIs can be allocated to a code section belonging to the OS Application of the caller.
- IOC code is optimized when the OS detects extra call-pattern information from RTA-RTE. Multiple data transfers can be batched together and executed via a single API call. Preemption information can be used to reduce blocking. This behavior can be disabled using the new OS option ’Disable IOC optimizations’
- The new OS option ’IOC blocking threshold’ can be used to control the blocking duration for batched IOC data transfers.

Modified Features

The following features have been modified in this release:

- Each cross-core IOC communication has its own spinlock, rather than a single global one. This is done to reduce cross-core blocking.

Removed Features

No features have been removed from this release.

3.24 Version 5.5.8

Additional Features

No features have been added to this release.

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.25 Version 5.5.7 (Preview)

Additional Features

The following features have been added to this release:

- Minor fixed for changes noted in the 5.5.6 preview.
- Additional hardware information has been added to the Configuration Summary.
- Additional information about Alarm autostarting has been added to the Configuration Summary.

Modified Features

The following features have been modified in this release:

- Where a configuration has untrusted code and stack protection is used, targets that use the stack realignment feature now follow the same code path whether the AlignedAddress has been changed or not. This makes the run-time behavior consistent in each invocation and makes it easier to protect from register corruption in the untrusted code.

Removed Features

No features have been removed from this release.

3.26 Version 5.5.6 (Preview)

Additional Features

The following features have been added to this release:

- This preview release supports TrustedApplicationWithProtection for OS Applications. This can only be selected on target ports that have been updated to support it.

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.27 Version 5.5.5 (Preview)

Additional Features

The following features have been added to this release:

- The OS options ’Asynchronous TASK activation’ and ’AsyncQ’ have been added. These modify APIs that cause cross-core activation (e.g. ActivateTask, ChainTask and SetEvent) so that they queue a message for the owning core to process during a cross-core interrupt. This allows the OS to remove the internal task-related spinlock and reduce cross-core blocking. Refer to the RTA-OS Reference Guide for more information.
- The new OS option ’--ioc’ has been added. This now allows the OS and IOC to be generated in independent stages. Refer to the RTA-OS Reference Guide for more information.

Modified Features

The following features have been modified in this release:

- MISRA checks are now performed according to the MISRA C:2012 Guidelines using the PC-Lint analysis tool. This has resulted in small modifications to individual lines of code, and slightly different lint dismissals. The MISRA Deviations report has been enhanced to include additional information required by MISRA C:2012.
- Autostarted TASKs, Alarms and ScheduleTables are now started on the core that owns them. Previously the master core started all of them.

Removed Features

No features have been removed from this release.

3.28 Version 5.5.4

Additional Features

No features have been added to this release.

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.29 Version 5.5.3

Additional Features

No features have been added to this release.

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.30 Version 5.5.2

Additional Features

The following features have been added to this release:

- E_OS_VALUE is documented as an error return code for GetElapsedCounterValue. (Documentation change only.)

Modified Features

The following features have been modified in this release:

- A counter with no Alarms or ScheduleTables will wrap at its maxallowedvalue. Previously it would continue incrementing to the maximum 32-bit value before wrapping.

Removed Features

No features have been removed from this release.

3.31 Version 5.5.1

Additional Features

The following features have been added to this release:

- Adds the AUTOSAR 4.2.x ’TrustedApplicationDelayTimingViolationCall’ feature. RTA-OS allows this option to be selected for both trusted and untrusted OS Applications. RTA-OS also allows it to be used with earlier AUTOSAR versions.

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.32 Version 5.5.0

Additional Features

The following features have been added to this release:

- Supports AUTOSAR 4.2.1 and 4.2.2, apart from 'TrustedApplicationDelayTimingViolationCall' and 'TrustedApplicationWithProtection' which are not yet implemented.
- A new Options Reference Report is available, and the OS and target options are described in the RTA-OS Reference Guide.
- Addition of the OS_TPL_/OS_IMASK_ macros and the Os_GetCurrentTPL()/Os_GetCurrentIMask() APIs.

Modified Features

The following features have been modified in this release:

- The internal implementation for multicore projects has been changed to reduce cross-core interference and to improve scalability to a higher number of cores. The ready state of tasks is now owned by the appropriate core rather then being shared between all cores. Each core has an internal spinlock for protecting access to its tasks, which reduces cross-core blocking. The task dispatching code is now specific to each core, so it only depends on the number of tasks on the core rather than the whole system.
- The OS no longer uses *_least types in public APIs.
- The DelayedTask feature is core-specific. You must only add tasks from the calling core to a delayed taskset.
- DelayedTasks are only supported in systems with a maximum of 64 tasks per core.
- The OS option Timing Protection Interrupt is now a TRUE/FALSE value. Previously a Category 1 ISR had to be specified. Each core must still provide a timing protection interrupt when this option is set to TRUE, but the OS does not check it.
- OS builds that are done inside the RTA-OS GUI are now implemented by building a command-line and passing it to rtaosgen.exe for the actual build. This avoids out of memory issues that have been seen in GUI builds. Note that the command-line can be modified in the GUI before starting the build.
- The ShutdownAllCores API uses an inlined version of DisableAllInterrupts, instead of calling it directly.

Removed Features

No features have been removed from this release.

3.33 Version 5.4.4

Additional Features

The following features have been added to this release:

- OS option ’Collect OS usage metrics’ has been added. Please refer to RTA-OS User Guide
- Support for TriCoreHighTec optimizations

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.34 Version 5.4.3

Additional Features

No features have been added to this release.

Modified Features

The following features have been modified in this release:

- Multicore only: Resource lock durations specified for RES_SCHEDULER are recognized. Previously they were ignored.
- Hardware counters can be used reliably on multicore systems. The OS now ensures that the callbacks that change the counter match conditions only run on the core that owns the counter. This makes it easier to implement correct callbacks. Typically the same code can be used for both single and multi core systems.
- ORTI output is modified to change from using the pointer variables Os_OrtiRunningISRn, Os_OrtiRunningTaskn to a more explicit form. Some debuggers didn’t like having to dereference the variables.

Removed Features

No features have been removed from this release.

3.35 Version 5.4.2

Additional Features

No features have been added to this release.

Modified Features

The following features have been modified in this release:

- Fix an issue in multicore projects that broke backwards compatibility with older target ports.
- Remove declaration of unused variable setjmp_retval in cross-core ISRs.

Removed Features

No features have been removed from this release.

3.36 Version 5.4.1

Additional Features

No features have been added to this release.

Modified Features

The following features have been modified in this release:

- When ’Optimize for core-local memory’ is selected, the per-core dynamic data for ISRs is now initialized correctly during StartOS. Previously it relied on its memory being cleared during startup.
- Added protective ()’s in the macros OSMEMORY_IS_READABLE, OSMEMORY_IS_WRITEABLE, OSMEMORY_IS_EXECUTABLE and OSMEMORY_IS_STACKSPACE
- GUI supports creation of sample applications for VRTA version 3.0.13. The code did not previously support versions that ended with more than one digit.

Removed Features

No features have been removed from this release.

3.37 Version 5.4.0

Additional Features

The following features have been added to this release:

- The OS option ’Optimize for core-local memory’ has been added. This re-organizes some OS data structures so that they can be assigned to core-local memory via the MemMap mechanism. Refer to the sample Os_MemMap.h to see the section names that get allocated.

Modified Features

The following features have been modified in this release:

- The initialization of internal spinlock variables has been moved inside StartOS rather than relying on power-on clearing of their values.
- For AUTOSAR 4.0.2 and above, OS data that must be cleared during reset moves from MemMap section OS_START_SEC_VAR_POWER_ON_INIT to OS_START_SEC_VAR_CLEARED.
- For AUTOSAR 4.0.3 and above, OS data that does not need to be initialized during reset moves from MemMap section OS_START_SEC_VAR_NOINIT to OS_START_SEC_VAR_NO_INIT.
- The content of the example Os_MemMap.h file is adjusted in AUTOSAR 4.0.2 and 4.0.3 configurations to match the changes in the MemMap specification.
- For AUTOSAR 4.0 and below, switch from using OS_APPL_CODE to OS_CALLOUT_CODE in non OS-internal functions to be AUTOSAR 4.0 compliant. (OS_CALLOUT_CODE will fall back to be the same as OS_APPL_CODE for earlier versions, so no change is needed in your code.)
- For AUTOSAR 4.1, the memory mapping mechanism #includes the Os_MemMap.h file rather than MemMap.h.
- For AUTOSAR 4.1, callouts (including hooks) have their own memory mapping section named according to the pattern OS_START_SEC_CN_CODE where CN is the callback name written in uppercase letters. The Os_MemMap.h sample file can be used to see the expected names. Note that some existing target port plug-ins might not compile for AUTOSAR 4.1 because of this change. Please contact support if this is the case for you.
- VSMD files have been updated.

Removed Features

No features have been removed from this release.

3.38 Version 5.3.91 (Experimental)

Additional Features

The following features have been added to this release:

- Supports for AUTOSAR 4.1.3

Modified Features

The following features have been modified in this release:

- Adjust OSTICKDURATION macro so it can be used in the code '#if (OSTICKDURATION < 1000)'
- The file Os_MemMap.h has moved into the RTA-OS samples. Use command-line option --samples:[Includes] to generate it. This is to emphasize that it is an example that can be optionally used as a start point by a system integrator.

Removed Features

No features have been removed from this release.

3.39 Version 5.3.90 (Experimental)

Additional Features

The following features have been added to this release:

- Experimental mc_experiment option for allocating data to core-local memory
- Cross-core access to hardware counters from Alarms/ScheduleTables on different cores is not allowed unless option 'Allow cross-core HW counter access' is set. This is because it is generally not possible to eliminate race conditions with code on different cores trying to set a new counter match value at the same time.

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.40 Version 5.3.2

Additional Features

The following features have been added to this release:

- API GetElapsedValue is added for AUTOSAR 4.x applications. It is an alias to GetElapsedCounterValue.

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.41 Version 5.3.1

Additional Features

The following features have been added to this release:

- Support for AUTOSAR 4.1.x ARXML.
- IOC supports N:M unqueued communication. This is a feature of AUTOSAR 4.1.x, but RTA-OS supports it in earlier versions when configured.
- The boolean parameter OsTaskCanCallSchedule is added to the TASK configuration. This can be set to FALSE if the TASK will not call the Schedule() API call - which may reduce system stack usage. The parameter defaults to TRUE. It is ignored if global OS option 'Disallow Schedule()' is TRUE, because that implies that no TASK calls Schedule().
- Option '--debug:compact_source' can be used to emit the compact version of the source code used to build the OS. This version does not expand all include files into the C source. (Source code license needed).
- Option 'Stack Sampling' has been added to cause the Os_Cbk_CheckStackDepth callback to get called to support monitoring of overall stack usage at run time
- When you create a new project in RTAOSCfg, you can now provide an empty ECU Configuration Name. In this case no ECUC-VALUE-COLLECTION element will be placed in the generated ARXML file. Similarly, if a file is read in without an ECUC-VALUE-COLLECTION element, none will be created.
- OS Option 'Task Activation Hook' can be used to cause Os_Cbk_TaskActivated(TaskID) to get called at Task activations.

Modified Features

The following features have been modified in this release:

- The memmap section for certain constants has moved from CONST_UNSPECIFIED to CONST_FAST_UNSPECIFIED. The constants affected are Os_const_coreconfiguration, Os_const_applications, Os_const_isrs, Os_const_tasks, Os_const_resources, Os_const_alarms, Os_const_counters, Os_const_scheduletables and TotalNumberOfCores.
- Os_Cbk_StackOverrunHook: OS_ECC_START, OS_ECC_RESUME and OS_ECC_WAIT can occur independently of whether Stack Monitoring is configured.
- Updates/corrections to the documentation for Os_Cbk_StackOverrunHook.
- Changes license check scheme.
- Removed E_OS_ACCESS checks from SyncScheduleTable/SyncScheduleTableRel when in STANDARD status.

- For configurations created for AUTOSAR 4.0.3 onwards, references in the ARXML change from /ETAS_RTAOS to /AUTOSAR_Os/EcucModuleDefs
- When creating a new project in the GUI, the Package Heirarchy can be used to create nested packages. e.g. a/b/c/EcucModuleConfigurationValuess.
- Only for multicore systems using delayed tasks: the task dispatcher is modified to avoid a race condition involving changing the delayed task set from a different core.
- Structure of VSMD files for AUTOSAR 4.0.3+ has changed slightly, based on customer feedback.

Removed Features

No features have been removed from this release.

3.42 Version 5.3.0

Additional Features

No features have been added to this release.

Modified Features

The following features have been modified in this release:

- Correction to max execution time calculation for ECC tasks.
- Correct service protection checks for APIs that can run during Startup Hook

Removed Features

No features have been removed from this release.

3.43 Version 5.2.9 (beta)

Additional Features

The following features have been added to this release:

- When the GUI RTAOSCfg creates new containers with an upper-multiplicity of 1, it now standardizes on giving then a SHORT-NAME that is the same as the AUTOSAR name. e.g. OsHooks, OsScheduleTblAdjustableExpPoint
- Added option to control checking for Activation of ECC tasks that are in the WAITING state. By default the check is not done, because ECC tasks typically are only activated once.
- The Os_SyncScheduleTableRel() has been added as an alternative for the AUTOSAR SyncScheduleTable() API where you supply a drift value rather than an absolute value.

- Option ’Enable Elapsed Time Recording’ has been added to provide support for measuring total time spent in Idle, Tasks and Category 2 ISRs through a new set of APIs
- Option ’Single Memory Protection Zone’ has been added to optimize setting up memory protection where all untrusted code can use a single set of protection settings..
- Option ’Stack Only Memory Protection’ has been added to optimize setting up memory protection where only the stack protection range needs to be changed.
- Option ’Optimize Schedule()’ is now also effective when there are more than 64 tasks.
- If it exists, the file (projectname)_postsamples.bat gets run during the build process. It runs after any sample files get created, so can be used to copy them to the correct build location.
- The StackFaultHook option is taken account of correctly.
- The version number of the target port is added to the header of all source files.
- The command-line option --os_option has been added. This acts like --target_option, but for the OS-specific options.
- Stack realignment of ECC tasks is possible within Os_Cbk_SetMemoryAccess().
- OS_INDEX_TO_TASKTYPE has been added as a partner to OS_TASKTYPE_TO_INDEX
- The macro OS_FAST_TASK_TERMINATION is defined when option ’Fast Terminate’ results in TerminateTask() being implemented as a simple return statement.

Modified Features

The following features have been modified in this release:

- Update to the code used to support Elapsed Time Recording to fix issue with ISRs that activate tasks.
- In ARXML for AUTOSAR 4.0.3, OsResources do not belong to an OsApplication.
- Clarifications in documentation about when spinlocks can be used.

Removed Features

No features have been removed from this release.

3.44 Version 5.2.2 (beta)

Additional Features

The following features have been added to this release:

- Prevents a spurious call to Os_Cbk_StackOverrunHook that could occur with Stack monitoring. This could only occur when using TerminateTask to leave an ECC task, and then only if an interrupt occurred within a small execution window.

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.45 Version 5.2.1 (beta)

Additional Features

No features have been added to this release.

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.46 Version 5.2.0

Additional Features

The following features have been added to this release:

- Spinlocks can be configured with a LockMethod, based on the AUTOSAR 4.1.1 proposed specification and with extra options.
- Spinlocks can record usage statistics at run time. See the GetSpinlockInfo API.
- The option --debug:build_info may be available, depending on your license. This causes a .bat file to be emitted that shows the build steps that RTA-OS uses.

Modified Features

The following features have been modified in this release:

- RTAOSCfg emits Alarm Actions within ECUC-CHOICE-CONTAINER-DEF rater than ECUC-PARAM-CONF-CONTAINER-DEF

Removed Features

No features have been removed from this release.

3.47 Version 5.1.3

Additional Features

The following features have been added to this release:

- Added optional callbacks Os_Cbk_TaskStart and Os_Cbk_TaskEnd. These act like Pre/Post Task hooks but don't get called when the task is pre-empted.
- Added optional callbacks Os_Cbk_ISRStart and Os_Cbk_ISREnd hooks. These get called when an Category 2 ISR starts/ends. They don't get called when the ISR is pre-empted.

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.48 Version 5.1.2

Additional Features

The following features have been added to this release:

- Added 2 new macros: OS_ISRTYPE_TO_INDEX and OS_TASKTYPE_TO_INDEX. These convert an ISRType or TaskType to an index value in the range 0 to the number of ISRs/Tasks.

Modified Features

The following features have been modified in this release:

- Fixes issue introduced in 5.1.1 where a system with untrusted functions, but no untrusted tasks could make Os_Cbk_SetMemoryAccess for the wrong element when leaving CallTrustedFunction()
- Multicore configurations only: fix issue where TASK execution time monitoring could record incorrect values if preempted by a cross-core ISR.

Removed Features

No features have been removed from this release.

3.49 Version 5.1.1

Additional Features

The following features have been added to this release:

- Support BAS licensing scheme.

Modified Features

The following features have been modified in this release:

- Delayed task execution APIs have run-time checks for validity.

Removed Features

No features have been removed from this release.

3.50 Version 5.1.0

Additional Features

The following features have been added to this release:

- Release.

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.51 Version 5.0.95

Additional Features

The following features have been added to this release:

- Partial support in GUI for OsAppEcucPartitionRef. The full path has to be entered manually. No checking is done.

Modified Features

The following features have been modified in this release:

- Fixes issue in GUI where deleting a TASK could result in deletion of OS Applications
- Fixes issue in GUI where window could start on a display that was no longer connected
- Fixes issue in GUI where --env did not affect the build path
- Fixes issue in GUI where reloading a project would result in target configuration information being ’greyed out’
- Fixes issue in GUI where relative paths are specified for include files. Such paths are relative to the project base directory.
- Revised wording in GUI for Expiry Point advance/retard to match the AUTOSAR 4.0.3 form

Removed Features

No features have been removed from this release.

3.52 Version 5.0.94

Additional Features

No features have been added to this release.

Modified Features

The following features have been modified in this release:

- Caters for the changed IOC configuration format for AUTOSAR 4.0.3
- Expands the documentation and examples around memory protection and Os_Cbk_SetMemoryAccess

Removed Features

No features have been removed from this release.

3.53 Version 5.0.93

Additional Features

The following features have been added to this release:

- Support for AUTOSAR 3.2.2
- Delayed Task Execution APIs
- ’Only Terminate Untrusted Applications’ optimization

Modified Features

The following features have been modified in this release:

- Slight update to AUTOSAR references in VSMD files

Removed Features

No features have been removed from this release.

3.54 Version 5.0.92

Additional Features

The following features have been added to this release:

- Supports 4.0.2 and 4.0.3 changes in XML (IOC / ScheduleTables)
- Supports nested packages in the XML

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.55 Version 5.0.91

Additional Features

No features have been added to this release.

Modified Features

The following features have been modified in this release:

- The VSMD files have been regenerated for each AUTOSAR version, using the ETAS_RTAOS root identifier
- Generated XML files reference ETAS_RTAOS rather than /AUTOSAR, /AUTOSAR/EcucDefs or /RTAOS/EcucDefs
- Generated XML files VALUE-REFs corrected (AUTOSAR 4)
- Generated XML files boolean values are emitted as NUMERICAL values (AUTOSAR 4)

Removed Features

No features have been removed from this release.

3.56 Version 5.0.90

Additional Features

No features have been added to this release.

Modified Features

The following features have been modified in this release:

- Fixes a possible race condition in multicore versions of WaitEvent
- The code in WaitEvent that checks for OS_ECC_WAIT error was testing the task's worst-case stack usage rather than the actual usage at the point of the call. This could make the stack allocation needed overly pessimistic.
- The algorithm used for license checking has been modified to reduce the number of requests to the ETAS License Manager - particularly for licenses that are *usually* not found.

Removed Features

No features have been removed from this release.

3.57 Version 5.0.0

Additional Features

The following features have been added to this release:

- Release.

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.58 Version 4.94.0

Additional Features

The following features have been added to this release:

- Initial cleanup and optimizations, specifically to reduce the number of calls to get core id in multicore applications.

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.59 Version 4.93.0

Additional Features

The following features have been added to this release:

- Functionally complete AUTOSAR 4.0 and multicore. Not yet optimized.

Modified Features

The following features have been modified in this release:

- Ensure that calls to Os_Cbk_SetMemoryAccess are done with interrupts locked.

Removed Features

No features have been removed from this release.

3.60 Version 4.92.3

Additional Features

The following features have been added to this release:

- Extended tasks work with multiple cores
- RES_SCHEDULER works across multiple cores

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.61 Version 4.92.2

Additional Features

The following features have been added to this release:

- Register sets work with multiple cores. A register set can include tasks and ISRs on different cores. The OS will use independent save areas for each core. Note that register sets cannot be used to protect data that is shared across cores - you will still need to use spinlocks for this.
- Register sets preserve state across an ECC task WaitEvent call.
- Experimental support for lightweight tasks in multicore if SC1 is selected.
- Increased test coverage on single-core code.

Modified Features

The following features have been modified in this release:

- Significant refactoring of PC-lint/MISRA dismissals (work in progress). PC-lint version 9 is now being used.

Removed Features

No features have been removed from this release.

3.62 Version 4.92.1

Additional Features

The following features have been added to this release:

- Early access development release for RTA-OS5.0.
- Refinements to target sample applications for multicore.
- Minor change in GUI for the number of cores offered for assignment to OS applications.
- Show OS Size Info report if selected after a build.
- Show license manager on Help menu (if installed).
- Remove restriction on max period for Schedule Table Assistant.
- Updated splash/about screens.

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.63 Version 4.91.0

Additional Features

The following features have been added to this release:

- Early access development release for RTA-OS5.0.
- Preliminary support for AUTOSAR MultiCore OS, including: Starting cores, StartOS, BCC Tasks and ISRs running on different cores, Spinlocks, Resources, Reports. MultiCore elements NOT necessarily supported: ECC tasks, IOC, forced termination, RegisterSets, Trusted functions, Shutdown OS.

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.64 Version 4.2.2

Additional Features

No features have been added to this release.

Modified Features

The following features have been modified in this release:

- Ensure that calls to Os_Cbk_SetMemoryAccess are done with interrupts locked.

Removed Features

No features have been removed from this release.

3.65 Version 4.2.0

Additional Features

No features have been added to this release.

Modified Features

The following features have been modified in this release:

- Beta release of RTA-OS 4.2. FunctionParams field added to structure passed to Os_Cbk_SetMemoryAccess. It gets used when calling an untrusted function.

Removed Features

No features have been removed from this release.

3.66 Version 4.1.0

Additional Features

No features have been added to this release.

Modified Features

The following features have been modified in this release:

- Release of RTA-OS 4.1. No changes post 4.0.91.

Removed Features

No features have been removed from this release.

3.67 Version 4.0.91

Additional Features

No features have been added to this release.

Modified Features

The following features have been modified in this release:

- Beta release of RTA-OS 4.1. GUI handling of sample applications fixed (SERN 20110125-DJL1). GUI code generation switch from EXTENDED to STANDARD status fixed (SERN 20101229-DJL2).

Removed Features

No features have been removed from this release.

3.68 Version 4.0.90

Additional Features

No features have been added to this release.

Modified Features

The following features have been modified in this release:

- First Beta release of RTA-OS 4.1

Removed Features

No features have been removed from this release.

3.69 Version 4.0.5

Additional Features

No features have been added to this release.

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.70 Version 4.0.4

Additional Features

No features have been added to this release.

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.71 Version 4.0.3

Additional Features

The following features have been added to this release:

- IOC single-core

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.72 Version 4.0.2

Additional Features

The following features have been added to this release:

- For each build, the tool will generate a log file projectname.log that contains a copy of the text that the tool and compiler output to the screen.
- A batch file named projectname_prebuild.bat can be used to run user-specified code just before the start of a build.
- A batch file named projectname_postbuild.bat can be used to run user-specified code just after the end of a build.
- Where the command-line options - -debug:source and - -nobuild are used together, the source code will be generated without the tool requiring a compiler to be available.
- The GUI can be directed to store only relative paths in the .rtaos file.

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.73 Version 4.0.1

Additional Features

The following features have been added to this release:

- AUTOSAR 4.0 API is supported in addition to AUTOSAR 3.x. The API selected depends upon the AUTOSAR XML version used to configure the project. AUTOSAR 4.0 tightens some previously unspecified behaviors and adds state to OS-Applications. TerminateApplication gains the ability to terminate a specified OS-Application.

Modified Features

The following features have been modified in this release:

- Hooks and callbacks are declared using FUNC(void, OS_CALLOUT_CODE) Os_Cbk_... rather than FUNC(void, OS_APPL_CODE) Os_Cbk_... to be compatible with AUTOSAR 4.0. For compatibility, OS_CALLOUT_CODE is defined as OS_APPL_CODE unless it already exists in Os_Compiler_Cfg.h

Removed Features

No features have been removed from this release.

3.74 Version 4.0.0

Additional Features

The following features have been added to this release:

- Support for AUTOSAR 4.0 XML, but not the 4.0 API changes. These are planned for the next release.
- Split Configuration is supported in rtaoscfg.exe.
- Function pointers can be created to most of the OS APIs. Not StartOS/Terminate-Task/ChainTask.
- Added API GetTaskActivationTime()

Modified Features

The following features have been modified in this release:

- Moved default installation location to c:/etas/rta-os

Removed Features

No features have been removed from this release.

4 Fixed EHI Calls

Bugs that have been fixed are referred to by their call number in the ETAS Helpdesk International (EHI) system.

4.1 Version 12.0.0

EHI 694630

Status:	Fixed
Title:	Possible compilation failure using –ioc:impl option.
Description:	In cases where the IOC used trusted functions to implement callbacks, there might be a compile error in –ioc:impl mode if the configuration specified that these functions should be implemented in the memory section of the owning OS Application.

EHI 695076

Status:	Fixed
Title:	Too many cross-core trusted function calls in –ioc:impl mode.
Description:	In cases where the IOC uses trusted functions to implement cross-core callbacks, the cross-core check code might attempt to run functions on the wrong cores (in addition to running them on the correct ones). The faulty calls can be expected to be rejected (E_OS_ACCESS) but they will waste execution time.

EHI 695544

Status:	Fixed
Title:	RTA-OS BSWMD Report fails with –ioc:impl mode.
Description:	The RTA-OS tool would fail if the BSWMD report was requested in a build where –ioc:impl mode was also selected.

EHI 696513

Status:	Fixed
Title:	Task activation count was limited to 255.
Description:	The number of activations per task was limited to 255, but is now 4294967295.

4.2 Version 6.2.200

EHI 674918

Status:	Fixed
Title:	Compilation fails on targets that support Enhanced Isolation without selecting full Memory Protection.
Description:	The OS now forces full memory protection to be used whenever Enhanced Isolation is selected.

EHI 674921

Status: Fixed

Title: Multi-core initialization to INVALID_OSAPPLICATION with AUTOSAR version 4.5+.

Description: It was possible for one of the OS-internal variables to skip initialization to INVALID_OSAPPLICATION in some multi-core configurations. This was only an issue when the INVALID_OSAPPLICATION value was non-zero, as specified for AUTOSAR 4.5+. For a workaround the OS option ’ApplicationType starts at zero’ could be set to FALSE.

EHI 674927

Status: Fixed

Title: IOC compilation failure with array types smaller than "IOC Atomic Copy Size"

Description: Array types are now excluded from the "IOC Atomic Copy Size" optimization. The C language does not suppport array assignment copies.

EHI 675636

Status: Fixed

Title: Possible compilation failure connected to OS-Application Error Hooks

Description: On some targets using stack realignment and per OS-Application Error Hooks, a compilation error might occur in configurations where not all untrusted or Trusted-With-Protection OS-Applications enabled the Error Hook.

EHI 677325

Status: Fixed

Title: Error in BSWMD for Os_TimingFaultDetected element

Description: BSWMD EXECUTION-CONTEXT for Os_TimingFaultDetected changed from CAT1 to INTERRUPT-CAT-1

EHI 680886

Status: Fixed

Title: IOC ReceiverIDs not respected in Optimized IOC mode.

Description: IOC ReceiverIDs operate correctly in AUTOSAR IOC mode, but not when the IOC Optimization mode is enabled. The code has been updated so that ReceiverIDs work in both modes.

EHI 683412

Status:	Fixed
Title:	Possible lockup in GUI build page
Description:	The GUI could lock up if an odd number of quote characters were entered manually in the build command line.

EHI 683413

Status:	Fixed
Title:	Possible GUI fault loading ARXML
Description:	The GUI would show a fault dialog changing the target device if OsRTATargetName was missing from the originally loaded ARXML configuration.

EHI 685538

Status:	Fixed
Title:	Incorrect reference paths in VSMD files
Description:	ETAS extensions to AUTOSAR ECUC file content are described in the VSMD files that ship with RTA-OS. The files for some versions of the VSMD files had incorrect references, and these are now fixed.

4.3 Version 6.2.0

EHI 666885

Status:	Fixed
Title:	Compilation fails on targets that support Enhanced Isolation if the Single Memory Protection Zone is selected.
Description:	Selecting Single Memory Protection Zone would remove one of the C types needed to support the Enhanced Isolation code.

4.4 Version 6.1.3

EHI 654189

Status:	Fixed
Title:	ApplicationType and value of INVALID_OSAPPLICATION
Description:	AUTOSAR version 4.5.0 (19-11) requires ApplicationType to start at 0 and INVALID_OSAPPLICATION to be above the range of valid ApplicationTypes. RTA-OS has previously used 0 for INVALID_OSAPPLICATION. To avoid breaking existing code, the new OS option ’ApplicationType starts at zero’ should be used if you want this behavior.

EHI 658926

Status:	Fixed
Title:	RTA-OS GUI choice of XSD when creating a new project.
Description:	When using the GUI to create a new project, the schemaLocation in the first line did not use the updated AUTOSAR schema ending 00044, 00046 and 00048 for AUTOSAR versions 4.3.1, 4.4.0 and 4.5.0 respectively. This has been corrected.

EHI 658934

Status:	Fixed
Title:	GUI support for editing OsIsrPeriod and OsTaskPeriod
Description:	These were added in AUTOSAR 4.5.0 for use by the RTE. The OS does not use them. The RTA-OS GUI now provides support to configure them.

EHI 658935

Status:	Fixed
Title:	MemoryStartAddressType definition change.
Description:	AUTOSAR version 4.5.0 (19-11) requires MemoryStartAddressType to be of type void *. RTA-OS has previously used uint8 * because that makes it easy to add a MemorySizeType to it. Adding MemorySizeType to a void * is not possible without a cast in portable C. To avoid breaking existing code, the new OS option 'MemoryStartAddressType as void pointer' should be used if you want MemoryStartAddressType to be of type void *.

4.5 Version 6.1.2 (Preview)

EHI 655022

Status:	Fixed
Title:	BSWMD report did not include IOC-generated memory sections.
Description:	Sections created as part of IOC generation are now included.

EHI 658291

Status:	Fixed
Title:	Possible fault in generation of OS_INDEX_FOR_CAT2-ISR-NAME
Description:	In the case where a Category 1 and Category 2 ISR share the same priority, the value of OS_INDEX_FOR_CAT2-ISR-NAME would be wrong if the name of the Category 1 ISR was alphabetically earlier.

EHI 658315

Status: Fixed

Title: Optimized IOC generation crash with unexpected input.

Description: The code generation could fail with an IOC configuration containing an OsIocWriteGroup without any OsIocWriteGroupParams (no data to transfer). The generation now succeeds and a warning is emitted.

4.6 Version 6.1.1

EHI 647895

Status: Fixed

Title: Issues in BSWMD file

Description: In some configurations variable and constants could get emitted as if they were in CODE memory sections. There was also a capitalization error in ResConsumption.

EHI 650575

Status: Fixed

Title: IOC declaration for ReceiverPullCB might be in wrong memory section.

Description: In IOC configurations with receive callbacks and OS option ’ioc_code’ set to ’Caller’ the extern for the trusted function did not move to the caller’s memory section.

EHI 650908

Status: Fixed

Title: Possible compilation error when a hardware counter is attached only to alarms that set events.

Description: In a multi-core system, some compilers could report an error in systems with hardware counters that only had event-setting alarms attached.

EHI 651205

Status: Fixed

Title: RTA-OS GUI fails to display Application CoreRef in cases where the CoreRef was previously undefined.

Description: When the GUI created a new CoreRef entry for an OS Application, it would not correctly update the UI to show it. The reference was stored in the ARXML, so saving the file and reloading would correct the issue. Also there was a slight error in the reference definition for the added CoreRef.

EHI 651424

Status: Fixed

Title: Compilation error in CallTrustedFunction.c with --ioc:stub option.

Description: There could be a compilation error in CallTrustedFunction.c for some targets when using the --ioc:stub command-line option if the configuration did not include some trusted functions. Note that this fix has not been tested on all previously released target versions, so for completeness you should ensure that your testing adequately checks the correct operation of trusted functions if you use the --ioc:stub configuration.

4.7 Version 6.1.0

EHI 637422

Status: Fixed

Title: Race condition when using SuspendOSInterrupts in a CAT1 ISR

Description: If a CAT1 ISR preempted the SuspendOSInterrupts API at a particular point, and then itself called SuspendOSInterrupts, it was possible for it to disrupt the final call to ResumeOSInterrupts such that the interrupt priority would not be restored. Although there is no reason for a Category 1 interrupt to call SuspendOSInterrupts, AUTOSAR allows it. The code is now modified when there are CAT1 ISRs in the configuration so that this issue cannot occur.

EHI 639138

Status: Fixed

Title: Possible compilation warning in 6.0.0

Description: In configurations that prevent forced task terminations there might be a compiler warning in ProtectionSupport.c because the the local variable ’Application’ was initialized but not used. Depending on compiler options this might be interpreted as a compilation failure.

EHI 641560

Status: Fixed

Title: Possible compilation error when a software counter is attached only to alarms that set events.

Description: In a multi-core system, some compilers could report an error in systems with software counters that only had event-setting alarms attached.

4.8 Version 6.0.0

EHI 636262

Status: Fixed

Title: Possible deadlock in IOC

Description: There was a small race condition where preemption of an IOC write by higher priority IOC-accessing code might result in a spinlock-related deadlock. The issue only applied when preempting an IOC write where there was a cross-core callback function that was triggered by the write and the data type was primitive and unqueued.

4.9 Version 5.91.0 (Preview)

EHI 626361

Status: Fixed

Title: Incorrect return from Os_GetCurrentTPL()

Description: The return value from Os_GetCurrentTPL() did not take account of the raising of the task priority when a resource lock was in effect. It could also produce incorrect results in systems with shared task priorities.

EHI 627128

Status: Fixed

Title: Possible RES_SCHEDULER failure in multicore systems

Description: In systems with a mixture of AUTOSAR and non-AUTOSAR cores, locking RES_SCHEDULER (when enabled) would fail on any AUTOSAR core that had an ID greater than any non-AUTOSAR core.

4.10 Version 5.7.2 (Preview)

EHI 623346

Status: Fixed

Title: Compilation error with --ioc:impl

Description: In multicore systems with cross-core IOC Rx callbacks, the code in Ioc.c could fail to compile or, depending on the compiler, call an illegal function.

4.11 Version 5.7.1 (Preview)

EHI 622904

Status: Fixed

Title: Incorrect re-initialization of some OS data during a warm reset.

Description: The OS-internal Os_AnyCoreInfo data structure that is used in multi-core systems was in a ’NO INIT’ memory section but should have been in a ’CLEARED’ memory section. It would normally be cleared correctly on a first-time reset by the C startup code but in systems with a warm reset capability it was not getting reinitialized to zero.

EHI 623635

Status: Fixed

Title: Incorrect scheduling with ECC2 tasks and cross-core SetEvent.

Description: If SetEvent was called for an ECC2 task on a different core at the point that it was entering the Waiting state, the dispatcher could try to start an erroneous task.

EHI 623907

Status: Fixed

Title: No autostarted Alarms or ScheduleTables when Service-Protection configured.

Description: The checks of caller legality were getting wrongly applied in the OS startup code, causing autostarted Alarms and ScheduleTables to get rejected.

4.12 Version 5.7.0

EHI 621257

Status: Fixed

Title: Possible compilation error in a Custom DLL

Description: In a customer-specific DLL it was possible to see a compilation error in some situations unless ORTI generation was enabled.

4.13 Version 5.6.90 (Preview)

EHI 615868

Status: Fixed

Title: Task continuing to run after TerminateApplication

Description: TerminateApplication might fail to fully terminate lower priority tasks if a cross-core activated tasks was already running on the terminating core.

EHI 616116

Status: Fixed

Title: XML identifiers containing underscores

Description: For AUTOSAR versions 4.0.x to 4.2.1, identifiers in ARXML were not allowed to have 2 consecutive underscores. However from 4.2.2 they are allowed. RTA-OS previously did not allow 2 consecutive underscores in version 4.2.2 and above.

EHI 617003

Status: Fixed

Title: Incorrect XML references created when adding objects in the RTA-OS GUI.

Description: The tooling could use the path to a non-OS ECUC-MODULE-DEF element if one appeared in a later ARXML file.

EHI 619930

Status: Fixed

Title: Incorrect handling of some references in optimized IOC ReadGroup and WriteGroup containers.

Description: The decoding of references to ReceiverProperties and SenderProperties previously assumed unique SHORT-NAMEs. The decoding now correctly uses the full path name, so duplicate SHORTNAMEs can be used.

4.14 Version 5.6.4

EHI 605608

Status: Fixed

Title: Custom DLL options show in the RTA-OS GUI even when the Custom DLL is NOT activated.

Description: Fixed. These options now only appear when the Custom DLL is activated.

EHI 606702

Status: Fixed

Title: Compilation failure in IocStubs.c.

Description: The code in IocStubs.c would not compile in the case where option --ioc:impl was selected and there was an Rx Callback for a communication that was single core.

EHI 606708

Status: Fixed

Title: IOC MCSD file processing was slow.

Description: On large input files, the MSCD file processing was inefficient. This has been corrected.

EHI 608082

Status: Fixed

Title: Incorrect MPU handling with ECC tasks when using Trusted-With-Protection.

Description: The MPU was not being switched out on entry to the code that dispatched ECC tasks. This has been corrected.

EHI 608594

Status: Fixed

Title: Sequential TerminateApplication calls might fail

Description: Where application code called TerminateApplication for several different OS Applications in sequence, calls after the first would fail with E_OS_CALLEVEL if Service Protection was enabled.

4.15 Version 5.6.3

EHI 596758

Status: Fixed

Title: Tools reject ARXML files that do not include OS container

Description: Tools warn about input ARXML files that do not include OS container, but allow them now. This is because in some cases they might contain references to data that is needed to complete the configuration; typically core assignments.

EHI 598729

Status: Fixed

Title: Possible failure in cross-core SetRelAlarm

Description: Where a software counter has a maximum value less than 0xffffffff, calling SetRelAlarm from a core different from the one running the counter could result in the alarm match time being set to a value that would never be reached. Workaround: Use full modulus counter or use SetAbsAlarm.

EHI 602306

Status: Fixed

Title: Possible code loop with hardware counters

Description: The OS-internal function Os_HwQFlush might execute an infinite loop in systems with more than one hardware counter.

EHI 602612

Status: Fixed

Title: GetTaskID not always callable from Os_Cbk_CheckStackDepth

Description: With ServiceProtection turned on, calling GetTaskID() from within the Os_Cbk_CheckStackDepth callback would be refused if the callback was raised for an autostarted task.

EHI 604120

Status: Fixed

Title: Possible compilation failure with Enhanced Isolation

Description: On some targets there could be a compile error in TerminateApplication.c because the variable ’calling_application’ was declared but not used in some configurations.

EHI 604121

Status: Fixed

Title: Possible compilation failure in cross-core interrupts with metrics enabled

Description: On some targets there could be a compile error in cross-core interrupt code. A variable was declared but not used in some configurations.

4.16 Version 5.6.1 (Preview)

EHI 586153

Status: Fixed

Title: Incorrect values returned from GetSpinlockInfo.

Description: GetSpinlockInfo could return an incorrect (large) value for the MaxSpinTime value. The value would be effectively the current stopwatch time. CurrentLockTime could contain a non-zero value when unlocked.

4.17 Version 5.6.0

EHI 565195

Status: Fixed

Title: Improve command-line error messages

Description: Extra clarification has been added to some command-line errors to be more specific about the cause of the error. Previously messages such as ’Illegal characters in path’ were emitted without indicating which option was affected.

EHI 566503

Status: Fixed

Title: Could not release spinlocks with interrupts disabled when Service Protection was enabled.

Description: GetSpinlock, ReleaseSpinlock, TryToGetSpinlock and TerminateApplication can now be called with interrupts disabled.

EHI 566622

Status: Fixed

Title: Added brackets around some macro parameters in Os.h

Description: Some macros with parameters did not wrap them in brackets when used, contrary to MISRA 2012 rule 20.7. This was not detected by our source code checker.

EHI 574522

Status: Fixed

Title: Incorrect MPU settings after preemption by a trusted-with-protection TASK

Description: On return from preemption by a trusted-with-protection TASK to an untrusted TASK, the Os_Cbk_SetMemoryAccess callback was not getting called to reset the MPU to the appropriate values for the untrusted TASK.

EHI 581901

Status: Fixed

Title: Incorrect calculation of RegisterSet preemption depth in multicore systems

Description: Applied only on multicore where OS option ’Disallow Schedule()’ was set and Tasks had same priorities or used internal resources. The size needed for the RegisterSet buffers could be underestimated. Workaround: remove ’Disallow Schedule()’.

4.18 Version 5.5.11

EHI 565986

Status: Fixed

Title: User Guide.

Description: Clarified the description of Stack overheads for ISR and Stack overheads for ISR activation.

4.19 Version 5.5.9 (Preview)

EHI 550970

Status: Fixed

Title: ECC TASKs running on the wrong core.

Description: It was possible for an ECC TASK allocated to one core to be (always) started on a different core. This situation would be quickly noticed in testing. This issue occurred only when more than one core had ECC TASKs, and depended on TASK priorities. Workaround: Ensure that all ECC TASKs on core N have priorities greater than those on core N-1.

EHI 556745

Status: Fixed

Title: Compilation error in Os_Cfg_Counters.c

Description: In a multicore system with a Software Counter that only drives Alarms, and where none of the Alarms activate a TASK, a compile error could occur because of a missing reference to os_current_controlled_core.

EHI 558338

Status: Fixed

Title: Insufficient resource locking for TASKs that share a priority.

Description: If a resource is used by a TASK that shares a priority with another TASK, then it should lock to at least the base priority of all TASKs that share that priority. Previously it was possible for the lock to omit some of these shared-priority TASKs. This is only a problem if the TASKs use multiple activation, because the implementation of the resulting FIFO activation mechanism meant that the activation of one of the TASKs with higher base priority could allow a lower base priority TASK to be popped from the FIFO and run, even though the resource was locked. A workaround is to configure the system so that any resource that is used by a TASK with a shared priority is also used by all other TASKs at that priority.

EHI 559562

Status: Fixed

Title: Incorrect return from GetCurrentApplicationID()

Description: When calling GetCurrentApplicationID() from an ISR or TASK that was preempting a Trusted Function, the return value would be the ID of the OS Application for the Trusted Function.

4.20 Version 5.5.8

EHI 528612

Status: Fixed

Title: Number of Cores incorrect in the RTA-OS GUI

Description: When a configuration with OsNumberOfCores equal to 1 was first loaded in to the RTA-OS GUI, it would increase the number to match the total number of cores supported by the target. Subsequent loads of the same file did not have this behavior.

EHI 548986

Status: Fixed

Title: Imbalance in MemMap start/stop directives in systems with no Counters, Alarms or ScheduleTables.

Description: In a system where no code needed to be emitted in for Counters, Alarms or ScheduleTables, the internal source file Os_Cfg_Counters.c could contain a OS_STOP_SEC_CONST_FAST_UNSPECIFIED without a preceding OS_START_SEC_CONST_FAST_UNSPECIFIED. There is no effect on the generated object code.

4.21 Version 5.5.7 (Preview)

EHI 537764

Status: Fixed

Title: Failure to build OS (out of memory)

Description: Migration of core tools to use .NET version 4 has improved memory performance.

EHI 546474

Status: Fixed

Title: Calculation of remaining time is incorrect when a trusted function is terminated with an exception.

Description: The remaining time is now decremented correctly.

EHI 546475

Status: Fixed

Title: Calculation of remaining time is incorrect when a trusted function is terminated with resource lock timeout AND there is a timing protection interrupt.

Description: The remaining time is now decremented correctly.

EHI 546476

Status: Fixed

Title: Reference bases in IOC

Description: The IOC component in RTA-OS now takes account of Reference Bases in the ARXML.

4.22 Version 5.5.6 (Preview)

EHI 541022

Status: Fixed

Title: Os_ShutdownAllCores() fails in untrusted mode when ORTI is switched on

Description: Os_ShutdownAllCores may only be called by trusted code, so it returns without shutting down when called from untrusted code. Previously the ORTI variable Os_OrtiApiID was written before checking the trust state, possibly provoking a memory protection fault.

4.23 Version 5.5.4

EHI 531809

Status: Fixed

Title: Cross-core IocWrite must be implemented as a FUNCTION if there is a receive callback

Description: RTA-OS will try to implement IocWrite as a MACRO for simple types so that the code can be inlined for efficiency. However prior to this fix, the implementation would not signal a receiver callback to run.

4.24 Version 5.5.3

EHI 522355

Status: Fixed

Title: Build-related command-line options to GUI

Description: The --target_include option could be passed to the GUI, but it did not get passed automatically to rtaosgen for building, although it could be entered manually and stored in the .rtaos file. The code now passes through GUI options that relate to the build. Affected versions: 5.5.0 and 5.5.1.

4.25 Version 5.5.2

EHI 516364

Status: Fixed

Title: Issue with IocEmptyQueue APIs on PPC target ports

Description: When a queued communication is used with IOC between exactly 2 different cores, the code generated for the IocEmptyQueue_xxx API should only be called from the receiver. If it is called by the sender then it might change the interrupt priority of the core of the receiver. This only affects multicore PowerPC target ports.

EHI 520507

Status: Fixed

Title: Fault in multicore code for Delayed Tasks

Description: The code generated for Delayed Tasks in a multicore configuration might only deal with one core. Affected versions: 5.5.0 and 5.5.1.

EHI 522340

Status: Fixed

Title: Compilation error in Os_Cfg.c with shared resources

Description: A compilation error would happen in Os_Cfg.c if the following 2 conditions occurred: (a) a resource was used by an interrupt and (b) the number of tasks on two cores differed by 32 or more. Affected versions: 5.5.0 and 5.5.1.

4.26 Version 5.5.1

EHI 511179

Status: Fixed

Title: Race condition affecting per-application error hooks

Description: For multicore configurations using the RH850GHS, PPCWR, PPCHT, PPCGHS and ZynqRVDS targets, it was possible for one core to raise the wrong application-specific error hook if (1) two such hooks were raised at the same time on different cores and (2) both belonged to an untrusted OS-Application and (3) untrusted stack re-alignment was configured.

4.27 Version 5.5.0

EHI 472745

Status: Fixed

Title: Documentation

Description: The User Documentation adds Windows 8.1 to the list of supported platforms

EHI 474886

Status: Fixed

Title: Sample Applications

Description: Sample applications contain the version and processor variant in the sample XML, so that they can be more easily read back into the GUI.

EHI 475944

Status: Fixed

Title: GUI build option

Description: The RTA-OS GUI option "Keep source code" was greyed out even when there was a valid licence.

EHI 477315

Status:	Fixed
Title:	GUI import
Description:	There was a discrepancy in the ISR information retained when opening a configuration file via the GUI versus the command line.

EHI 483338

Status:	Fixed
Title:	RTA-OS User Guide
Description:	The description of ECC Wait Stack size is clarified.

EHI 485189

Status:	Fixed
Title:	Tools crash
Description:	The tools could crash when loading a configuration where an OS Application did not have a valid core assignment.

EHI 485193

Status:	Fixed
Title:	Sample code
Description:	The default implementations of Os_Cbk_SetTimeLimit and Os_Cbk_SuspendTimeLimit are multicore aware.

EHI 485504

Status:	Fixed
Title:	Ordering in GUI
Description:	In the navigation tree, ISRs can be sorted by Name, Priority, Address or Category. Tasks can be sorted by Name or Priority.

EHI 487821

Status:	Fixed
Title:	Hardware Counters
Description:	It was possible to miss the activation of an alarm if a new alarm was being started and the hardware compare match occurred just after the OS had checked if the compare interrupt was pending.

EHI 487825

Status:	Fixed
Title:	Timing protection restoration after nested preemption
Description:	It was possible for a nested interrupt to allocate more execution time to a preempted TASK than was correct. Only the first preemption should adjust its time.

EHI 492200

Status:	Fixed
Title:	RTA-OS User Guide
Description:	Additional explanation of which APIs can be called when interrupts are disabled.

4.28 Version 5.4.3

EHI 417525

Status:	Fixed
Title:	OS variables read permission with untrusted code
Description:	Just after switching to untrusted mode to enter an untrusted TASK, ISR or function, the OS was reading from an OS variable or constant that held the entry address. This meant that OS data had to be readable (though not writable) from untrusted code. Although this is a common way to set up the memory protection, some customers might prefer to prevent read access to OS data by untrusted code. The implementation has been adjusted to help support this.

EHI 451106

Status:	Fixed
Title:	ShutdownAllCores with untrusted OS Applications
Description:	Calling ShutdownAllCores could fail to complete if code on another core was running untrusted code at the time of the call.

EHI 452252

Status:	Fixed
Title:	Check-only build option in GUI
Description:	For some targets (e.g. TriCore Tasking), a check-only build of the OS from within the OS GUI would fail if the compiler was not on the path.

EHI 456736

Status:	Fixed
Title:	Compilation warning in CallTrustedFunction.c
Description:	A compilation warning could occur in CallTrustedFunction.c for the case where memory protection is enabled and there are untrusted TASKs, but no untrusted Functions.

EHI 456737

Status:	Fixed
Title:	Compilation failure in Os_Wrapper.c
Description:	If the OS option 'Only Terminate Untrusted Applications' was true and there were no Untrusted Applications, this could provoke a compilation error. This is corrected and now for this case the OS does not emit any code to support termination of Applications.

EHI 458456

Status:	Fixed
Title:	Compilation failure in Ioc.c
Description:	A compilation error could occur in Ioc.c for elements taking 2 or more values where the FunctionImplementationKind was MACRO.

EHI 458502

Status:	Fixed
Title:	Write to ORTI variable from untrusted code
Description:	Some 'readonly' APIs such as GetISRID() were not switching to trusted mode before updating the OS internal ORTI variables. This occurred when ORTI was enabled and the option 'Untrusted code can read OS data' was TRUE.

EHI 458506

Status:	Fixed
Title:	Compilation failure in TimeProtection.c
Description:	A compilation error could occur in TimeProtection.c when using Timing Protection with a timing interrupt in a multi-core configuration.

EHI 473819

Status:	Fixed
Title:	Race condition in ShutdownAllCores
Description:	If a cross-core interrupt occurred during a call to ShutdownAllCores, it was possible that the other cores would not be informed about the shutdown immediately. In some situations the other cores might not detect the shutdown. This issue is fixed. As a workaround for earlier versions, call DisableAllInterrupts() before calling ShutdownAllCores().

4.29 Version 5.4.1

EHI 432574

Status: Fixed

Title: Untrusted functions

Description: RTA-OS allows untrusted OS Applications to declare 'Trusted Functions'. These can be used in the same way as AUTOSAR trusted functions, but they run untrusted. A build warning has been added to make it clear when this is happening.

EHI 442493

Status: Fixed

Title: GUI crash

Description: Malformed ARXML missing VALUE-REFs could cause the GUI to crash when reading in the data.

4.30 Version 5.3.91 (Experimental)

EHI 416358

Status: Fixed

Title: VSMD REFINED-MODULE-DEF-REF

Description: The REFINED-MODULE-DEF-REF section was in the wrong place in the VSMD

EHI 420784

Status: Fixed

Title: VSMD reference in OsRegSetRef

Description: The DESTINATION-REF for OsRegSet has a wrong path

EHI 421442

Status: Fixed

Title: Missed events from alarm

Description: Corrected an issue that could result in alarms not setting events under these specific conditions: IF (you have alarms that set events) AND there are alarms on the same counter that activate tasks but *don’t* set events AND EITHER ((number of event bits < 9) AND ((number of tasks + number of alarm callbacks + number of event bits) > 8)) OR ((number of event bits < 17) AND ((number of tasks + number of alarm callbacks + number of event bits) > 16)) THEN One or more events will not get set when the alarm fires. The task may resume from wait, but will not see the event set.

Note: ’number of event bits’ is normally the maximum number of events that any task waits for. It is not the total number of events. i.e. you can have 30 events in the system, but if no task waits for more than 2 events then the number of event bits is 2. This only changes if you allocate specific mask values to alarms because the number of event bits may have to increase to account for the value you set.

Workaround: Select an event and force its mask to bit 9 or 17 such that the condition above is no longer met.

EHI 422921

Status: Fixed

Title: Software counters do not increment unless they contain Alarms or ScheduleTables

Description: (Introduced in version 5.3.1) Software counters that contain no Alarms or ScheduleTables returned zero from GetCounterValue().

EHI 425490

Status: Fixed

Title: User Guide

Description: Section 10.1 of the RTA-OS User Guide wrongly stated that a Counter could only drive one ScheduleTable.

4.31 Version 5.3.2

EHI 403473

Status: Fixed

Title: In version 5.3.1, the VSMD files for AUTOSAR 3.x were generated incorrectly and included some AUTOSAR 4.x syntax.

Description:

EHI 412197

Status: Fixed

Title: In version 5.3.1 an internal crash could occur when calculating stack usage. This affected unusual multicore configurations where the OsTaskCanCallSchedule feature was set to FALSE on a TASK.

Description:

4.32 Version 5.3.1

EHI 390500

Status: Fixed

Title: Compilation error in CallTrustedFunction.c with a rare set of multicore configuration options.

Description:

EHI 392097

Status: Fixed

Title: In SC3/4, PostTaskHook was not getting called when a Task terminated through ChainTask.

Description:

EHI 399847

Status: Fixed

Title: In the Beta2 GUI, sorting of some columns in the TASK and ISR summary caused the application to crash.

Description:

4.33 Version 5.2.9 (beta)

EHI 369431

Status: Fixed

Title: In RTAOSCfg, saving with an empty expiry point selected could cause unhandled exception

Description:

EHI 369432

Status:	Fixed
Title:	In RTAOsCfg, checking is done that the correct number of stack values are entered. Affects TriCore targets.
Description:	

4.34 Version 5.2.2 (beta)

EHI 369431

Status:	Fixed
Title:	RTAOSCfg: Saving the project with the cursor in an empty cell on the ScheduleTable/Expiry Points pane could cause an unhandled exception
Description:	

EHI 369432

Status:	Fixed
Title:	RTAOSCfg: The validation of stack values for multi-stack targets (TriCore) did not reject entries with fewer stack values than required. e.g. 16 instead of 16,64.
Description:	

4.35 Version 5.2.1 (beta)

EHI 359725

Status:	Fixed
Title:	When starting an ECC task, there was a small window where an ISR could interrupt and cause the stack overrun hook to be called when it was not necessary. Only affected systems with stack monitoring enabled.
Description:	

EHI 999999

Status:	Fixed
Title:	Fixes a small window with ECC tasks on multicore systems where setting an Event for a Task on a different core might not be detected immediately.
Description:	

4.36 Version 5.2.0

EHI 354972

Status: Fixed

Title: When using hardware counters and starting alarms or schedule tables relative to the current time, it was sometimes possible for to skip the next counter interrupt if it was already due earlier than the interval being requested in this call.

Description: Use software counters or absolute start times.

4.37 Version 4.0.5

EHI 249323

Status: Fixed

Title: Build size report in GUI

Description: If a non-default output location for .html files was specified, post build reports would not open in the browser, though they would be created correctly.

4.38 Version 4.0.4

EHI 249320

Status: Fixed

Title: Pessimistic stack size calculation for ISRs with the same priority

Description: With ISRs at the same priority, the maximum stack calculation was pessimistic. It assumed that all ISRs could preempt each other. This only showed on systems where Schedule() was allowed.

4.39 Version 4.0.3

EHI 249317

Status: Fixed

Title: Tracing with compact IDs and extended time

Description: When tracing with compact ids and extended time, certain targets would read the stopwatch twice per record (once for each 16 bits). Where the 2 reads straddled a 0x1_0000 boundary, the time recorded appeared to jump backwards.

EHI 249318

Status: Fixed

Title: Large trace buffers and serial links

Description: Serial links are restricted to 64K frames by RTA-TRACE. The RTA-OS code was trying to limit transmission to 64K, but not including follow-on frames where the buffer size was larger than the frame. It could also mis-calculate the size and crash RTA-TRACE Server.

4.40 Version 4.0.2

EHI 249311

Status: Fixed

Title: SetEvent

Description: If 2 events were set for a waiting ECC task before it was able to resume, then the task could appear to be reactivated twice. This only happened on systems where any task had multiple activations, or where tasks shared a priority.

EHI 249312

Status: Fixed

Title: Register sets

Description: If an ISR was saving or restoring a register set and a higher prio isr fired then the higher prio isr could save over the same set that the lower one was using. Only affected some of the simpler configurations. Would not occur with stack monitoring, time monitoring or memory protection present.

4.41 Version 4.0.0

EHI 249284

Status: Fixed

Title: Terminating Applications

Description: If a task activated by an ISR faulted and the protection hook was told to kill the OS Application, a null reference could occur in the interrupt wrapper.

EHI 249288

Status: Fixed

Title: NextScheduleTable

Description: A repeating ScheduleTable X, that had been ’nexted’ to ScheduleTable Y and then ’nexted’ from Y would run once and then try to ’next’ to Y rather than repeat X.

5 Limitations

5.1 Installer

There are no known limitations.

5.2 General

There are the following limitations for this tool:

Limitation RTA-OS does not currently support optional data lengths in IOC functions (AUTOSAR 4.3.0 [SWS_Os_00805]).

Workaround Please contact support if you need this behavior.

Limitation RTA-OS does not currently supply a specification of the Ports and Port Interfaces.

Workaround Please contact support if you need this behavior.

Limitation RTA-OS does not implement the following AUTOSAR 4.* requirement because of the impact on performance at run-time: "Reaction to timing protection can be defined to terminate the OSApplication. If a task is inside CallTrustedFunction() and task rescheduling takes place within the same OSApplication, the newly running higher priority task may cause timing protection and terminate the OSApplication, thus indirectly aborting the trusted function. To avoid this, the scheduling of other Tasks which belong to the same OS-Application as the caller needs to be restricted, as well as the availability of interrupts of the same OS-Application." Note that RTA-OS will suspend the timing measurement of the calling code if the function has its own timing budget, so the issue of timing overruns can be controlled. The TrustedApplicationDelayTimingViolationCall configuration option can be used to prevent timing protection from terminating code part-way through a Trusted Function.

Workaround Please contact support if you need this behavior.

Limitation RTA-OS does not expect to handle more than one <AR-PACKAGE> containing an OS definition per input XML file.

Workaround Merge the OS definitions in the file or put each one into a separate file using an XML/text editor.

5.3 rtaoscfg

There are the following limitations for this tool:

Limitation [File modified indicator is unreliable.] In particular, opening a file sets the modification indicator, even when no changes have been made and making modifications to a saved file does not set the indicator.

Workaround None.

Limitation [Project Save] When a project is saved all constituent files are also saved irrespective of whether or not they have been modified.

Workaround None.

5.4 rtaosgen

There are the following limitations for this tool:

Limitation [AC31039] Libraries generated for the same configuration and compiler will result in the same target code each time. However a binary comparison of 2 libraries generated at different times may differ because the compilations take place in differently named temporary directories and some toolchains embed the build path in the object and/or library file.

Workaround None.

Limitation Prior to AUTOSAR4.0, all Service IDs were specified to have the form OSServiceId_api_name, and their values were implementation dependent. In the AUTOSAR 4.0 SWS, the AUTOSAR-specific APIs (but not the OSEK compatible ones) are assigned numeric values from 0x00-0x0d and 0x0f-0x14. This is thought to be a flaw in the generation of the document rather than a deliberate change. By contrast the MultiCore OS SWS still uses OSServiceId_api_name rather than numbers. For this reason, RTA-OS stays with the original implementation (OSServiceId_api_name) and defines OSServiceId_None to have the value zero. Additionally Service IDs are not allocated in Hook calls, despite the 4.0 SWS suggesting that they should all be assigned a value 0x00.

Workaround None.

6 Standards and Norms

RTA-OS implements AUTOSAR OS (R3.0.1 -> R3.0.7, R3.1.1 -> R3.1.5, R3.2.1 -> R3.2.2, R4.0.1 -> R4.6.0 (R20-11)) Scalability Class 1-4.

6.1 Deviations

6.1.1 OsDriver

Each hardware counter declared in the configuration can take an optional `OsDriver` configuration parameter that indicate how AUTOSAR interacts with the driver.

If the configuration parameter is omitted then the interaction is defined to be 'INTERNAL'. How interaction occurs is undefined by the AUTOSAR standard. This model is supported by RTA-OS and interaction occurs using the Hardware Counter Driver Interface described in Section 6.3.1.

If the `OsDriver` is specified then it must define the driver source as AUTOSAR Gpt channel. The OS requires that the Gpt provide free-running functionality and a match interrupt. However, the Gpt can be free running *or* provide a match interrupt but cannot provide both from the same Gpt channel and therefore cannot be used by the OS. This is a known bug with the AUTOSAR Gpt specification that may be fixed in later releases of AUTOSAR OS.

Due to this problem in the standard, RTA-OS does not support the `OsDriver` configuration attribute.

If your input configuration includes and `OsDriver` then this is ignored and you will be expected to use the RTA-OS the Hardware Counter Driver Interface described in Section 6.3.1 to interact with the Gpt.

6.2 Clarifications

Some parts of the AUTOSAR OS specifications are ambiguous or otherwise poorly defined. This section identifies such ambiguities and states what interpretations of these parts of the specification have been used when implementing RTA-OS. If you are migrating to RTA-OS from another AUTOSAR implementation these clarifications can be used to identify and resolve portability issues as other vendors may have different interpretations.

6.2.1 OsAppModeId

AUTOSAR OS R3.0 (not R3.1 or later) has a configuration parameter for `OsAppMode` called `OsAppModeId`. The parameter is mandatory in the XML configuration language but its use is specified as "internal" in the AUTOSAR standard. This means it is an internal parameter that is externally configurable.

When dealing with R3.0 configurations, RTA-OS ignores any configured value in this parameter and allocates internal identifiers to each OsAppMode.

6.3 Extensions

Extensions are functionality that is provided by RTA-OS in addition to that specified by the AUTOSAR OS standards. Further details about extensions are provided in the *RTA-OS User Guide*.

6.3.1 Hardware Counter Driver Interface

The interface between hardware counter drivers and AUTOSAR OS is not specified by AUTOSAR. RTA-OS provides a simple but highly flexible hardware counter interface that can be used to interface any type of peripheral with the OS. The interface provides an API called `Os_AdvanceCounter()` that does two things:

1. It tells RTA-OS that a hardware counter trigger has occurred.
2. In return, RTA-OS programs the number of ticks of the counter that must elapse before for the next trigger occurs through a user-supplied callback function.

The API is provided in both dynamic (counter to advance passed in as a parameter) and static (counter to advance is bound into a generated API call name) forms just like the software counter interface.

There are also four simple user callbacks that RTA-OS needs to control the peripheral.

This simple model is useful for a wide variety of peripherals hardware counter sources and allows very precise control of the OS from external triggers.

6.3.2 Extended Stack Monitoring Features

AUTOSAR OS provides stack monitoring. When a stack fault occurs, the specification states that the OS should be shutdown via a call to `ShutdownOS()`. This is not especially useful if you are trying to debug stack problems at development time because you have no way to identify the cause of the problem. RTA-OS provides an optional callback function that provides you access to information about the actual fault. When returning from this callback, the standard AUTOSAR behavior occurs.

6.3.3 Time Monitoring

AUTOSAR OS provides a time monitoring and protection mechanism in SC2 and 4. RTA-OS provides a lightweight and efficient time monitoring system available in SC1 that can be used to measure execution times and detect timing overruns in the application. When enabled, time monitoring logs the execution time of tasks and Category 2 ISRs and checks it against pre-configured budgets for overrun. Similarly, critical sections protected by resource or interrupt locks can be measured.

Unlike timing protection in AUTOSAR OS SC2/4, time monitoring does not require any access to a peripheral timer interrupt, only access to a free running counter, making it useful on even the most resource constrained systems. Timing overruns do not result in the termination of tasks or Category 2 ISRs, they simply invoke a user supplied callback routine, allowing the error to be logged or the appropriate fault recovery strategy to be invoked. This means that time monitoring requires significantly less resources at runtime - making it appropriate when resources are severely constrained or when you want to identify timing problems but do not want to significantly change the behavior of the OS at runtime.

6.3.4 Fast Software Counter Driver Interface

RTA-OS provides a static interface to software counters in addition to the dynamic interface required by AUTOSAR OS. This means a standard call like:

```
IncrementCounter(YourCounter);
```

may also be made in RTA-OS as:

```
Os_IncrementCounter_YourCounter();
```

This is provided because many increment counter calls are made from interrupts where speed is important. By statically binding the counter name into the API, RTA-OS avoids the need to pass a parameter on the stack and decode that parameter internally.

There are no restrictions on the use of the different versions - both the static and dynamic versions of the call can be mixed in application code.

7 Contacting ETAS

7.1 Technical Support

Technical support is available to all users with a valid support contract. If you do not have a valid support contract, please contact your regional sales office (see below).

The best way to get technical support is by email. Any problems or questions about the use of the product should be sent to:

rta.hotline@etas.com

If you prefer to discuss your problem with the technical support team, you call the support hotline on:

+44 (0)1904 562624.

The hotline is available during normal office hours (0900-1730 GMT/BST).

In either case, it is helpful if you can provide technical support with the following information:

- Your support contract number
- The version of the ETAS tools you are using
- The version of the compiler tool chain you are using
- The command line (or reproduction of steps) that result in an error message
- The error messages or return codes you received (if any)
- Your `.xml`, `.arxml` and `.rtaos` files
- The file `Diagnostic.dmp` if it was generated

7.2 General Enquiries

7.2.1 ETAS Global Headquarters

ETAS GmbH		
Borsigstrasse 24	Phone:	+49 711 3423-0
70469 Stuttgart	Fax:	+49 711 3423-2106
Germany	WWW:	www.etas.com

7.2.2 ETAS Local Sales & Support Offices

Contact details for your local sales office and local technical support team (where available) can be found on the ETAS web site:

ETAS subsidiaries	www.etas.com/en/contact.php
ETAS technical support	www.etas.com/en/hotlines.php