

Updates in 10.4.3

In this release, we've focused on enhancing our functional safety and security compliance solution. We've extended the coverage of the AUTOSAR C++ 14 standard to help you achieve full compliance with AUTOSAR's *required* and *automated* rules and added support for the latest release of the CWE guidelines.

- [Support for Environments](#)
- [Extended Automotive Compliance Pack](#)
- [Extended Security Compliance Pack](#)
- [New and Updated Code Analysis Rules](#)
- [New and Updated Test Configurations](#)
- [Creating Stubs that Call the Original Function](#)
- [Standalone License Server](#)
- [OpenID Connect Support](#)
- [Other Changes](#)
- [Resolved Bugs and FRs](#)

Support for Environments

New IDEs

We've added support for:

- QNX Software Development Platform 7.0

New Compilers

Compiler Name	Compiler Acronym
GNU GCC 9.x	gcc_9
GNU GCC 9.x (x86_64)	gcc_9-64
IAR Compiler for ARM v. 8.22x	iccarm_8_22
IAR Compiler for ARM v. 8.40x	iccarm_8_40
Microsoft Visual C++ 14.2	vc_14_2
Microsoft Visual C++ 14.2 (x86_64)	vc_14_2-64
Clang C/C++ Compiler v 8.0 (x86_64)	clang_8_0

Support for QNX GCC 5.x (ARM) is now extended and approved for use in safety-critical software development.

The ARM NEON extensions are now supported for GCC- and ARM-based compilers.

Deprecated Compilers

Support for the following compilers is deprecated and will be removed in future releases:

- ARM RealView 4.1
- ARM RealView 4.1 for uVision
- CodeSourcery Sourcery G++ Lite 2009q1-203
- GNU GCC 4.0.x
- GNU GCC 4.0.x (x86_64)
- GNU GCC 4.1.x
- GNU GCC 4.1.x (x86_64)
- GNU GCC 4.2.x
- GNU GCC 4.2.x (x86_64)
- GNU GCC 4.3.x
- GNU GCC 4.3.x (x86_64)
- GNU GCC 4.4.x
- GNU GCC 4.4.x (x86_64)
- IAR Compiler for ARM v. 6.1x
- IAR Compiler for ARM v. 6.3x
- IAR Compiler for MSP430 v. 5.4x
- Microsoft Visual C++ 9.0
- Microsoft Visual C++ 10.0
- TI TMS320C2000 C/C++ Compiler v6.2
- TI TMS320C6x C/C++ Compiler v7.3

- TI MSP430 C/C++ Compiler v4.0
- Vx-toolset for TriCore C/C++ Compiler 4.0
- Wind River GCC 3.4.x

Intel C++ Compiler v 18.0 is no longer supported on Windows.

Extended Automotive Compliance Pack

We've extended support for AUTOSAR C++ 14 to help you achieve compliance with the standard. All AUTOSAR rules from the "required" and "automated" categories are now fully covered to support your testing efforts in the development of automotive system architectures.

Extended Security Compliance Pack

We've added support for the newly updated 2019 Common Weakness Enumeration (CWE). C/C++test now ships with new test configurations to help you enforce compliance with the CWE Top 25 2019 and CWE Weaknesses on the Cusp guidelines; see the [New and Updated Test Configurations](#) section below.

New and Updated Code Analysis Rules

We've added new static analysis rules to extend coverage of compliance standards, with a special focus on the AUTOSAR standard C++ 14 ; see [New Rules](#) and [Updated Rules](#) for the lists of new and updated rules.

In addition, we've added a NOMCIM metric to calculate the number of function calls in functions.

New and Updated Test Configurations

We've added the following test configurations:

- CWE Top 25 2019
- CWE Top 25 + On the Cusp 2019
- OWASP Top 10 2017
- UL 2900

Deprecated Test Configurations

- CWE-SANS Top 25 Most Dangerous Programming Errors – deprecated and replaced with the CWE Top 25 2019 test configuration
- OWASP Top 10 2017 – deprecated and replaced with the new OWASP Top 10 2017 test configuration
- UL 2900 – deprecated and replaced with the new UL 2900 test configuration that includes CWE SANS Top 25 + On the Cusp 2019 and OWASP Top 10 2017 rules

The deprecated test configurations are not available by default and can only be applied as user-defined test configuration. They are now shipped with C/C++test in the following location: `[INSTALL_DIR]\configs\Deprecated`.

Creating Stubs that Call the Original Function

The Stub Callbacks mechanism has been enhanced to call the original function if no test-case specific Stub Callback Function is registered in the test case; see [Creating Stubs that Call the Original Function](#).

Standalone License Server

You can now obtain the Parasoft license from an additional instance of DTP or a standalone License Server. See [Licensing](#).

OpenID Connect Support

You can now authenticate on DTP via OpenID Connect to add a layer of security to your interactions with your DTP server. See [Configuring OpenID Connect in the UI](#) and [Configuring OpenID Connect the .properties File](#) for details.

Other Changes

- The `@test` issue tracking tag is now supported by default for associating test with development artifacts; see [Indicating Code and Test Correlations](#).
- Performance of flow analysis in the incremental analysis mode has improved. We've reduced analysis times in subsequent runs.

- Connecting to Project Center is no longer supported. The Project Center module shipped with DTP/Concerto has reached its end-of-life (EOL) phase and was removed in [DTP 5.4.2](#).
- QNX Momentics IDEs older than version 7 are no longer supported.

Resolved Bugs and FRs

Bug/FR ID	Description
CPP-39554	Rule CODSTA-13 should be updated to follow MISRA2004-17_3
CPP-39913	VS Cannot enable filtered rules in Visual Studio.
CPP-42073	Add support for ARM NEON extensions
CPP-42495	Rule COMMENT-04 should not report on a function declaration when function definition is not available (visible)
CPP-42527	Improve mapping for AUTOSAR-M2_13_2-a (C++14 semantics)
CPP-42858	Improve mapping for CERT-INT31
CPP-43140	Improve mapping for MISRA2012-RULE-2_2
CPP-43141	Rule CODSTA-163_b (MISRA2012-RULE_10_3-b) reports false positive on ternary operator
CPP-43142	Rule CODSTA-CPP-59 reports false positive on #include directives excluded by __cplusplus macro
CPP-43143	Rule MISRA2004-9_2_c (AUTOSAR-M8_5_2-c) reports incorrectly on std::array and constexpr
CPP-43150	Rule GLOBAL-ONEUSEVAR (MISRA2008-0_1_4) reports false positive when static const variable is used as template argument
CPP-43413	Rule OPT-02 (OPT-03, OPT-31) reports false positive on parameters/variables captured by lambdas
CPP-43414	Parse failure reported for user-defined suffixes in templates (C++14)
CPP-43465	LSI cannot read object/library data for ARM OE toolchain
CPP-43479	Error reported when instrumenting code (Process exited with code: 137)
CPP-43523	Error reported when running unit tests: Invalid file format: Unable to read exports
CPP-43549	Custom source/header file extensions not propagated from IDE to Static Analysis engine
CPP-43558	Timeout is not deactivated when debugging test cases
CPP-43567	Symbols __once_call and __once_callable from libstdc++ are reported not found by LSI
CPP-43568	C/C++test cannot be installed if both VS2017 and VS 2019 are installed on a machine
CPP-43602	Configure gnu99 option for GHS/ARM compilers
CPP-43603	Rule FORMAT-43 reports false positive when unpaired braces are #ifdef/#ifndef
CPP-43643	Missing support for "--core" option in IAR-RL78 compiler configuration
CPP-43667	Rule OPT-05 reports false positive if const variable is used as template argument
CPP-43675	Rule PB-45 reports false positive when plain char is passed as '%c' specifier in printf/scanf function call
CPP-43688	Rules PB-45, PB-46, PB-47, PB-48, PB-49 work incorrectly for arguments of 'scanf' functions
CPP-43689	Rule PB-50 reports false positive when characters specifier is used in 'scanf' function
CPP-43706	Improve rule MISRA2004-20_5 (JSF-017): do not print line number in violation message
CPP-43744	Improve algorithm which filters duplicated violations.
CPP-43748	Rule MISRA2004-17_6_a reports false positive when address of dereferenced iterator is returned from function
CPP-43831	Compilation error on safe stubs with Microsoft Windows Kit SDK 10.0.18362.0
CPP-43837	Parse failure reported when using -endian=big with Renesas RX C++ 2.5.X compiler
CPP-43869	Rule INIT-05 reports false positive on rvalue reference
CPP-43889	Parse failure reported: initial value of reference to non-const must be an lvalue
CPP-43892	Parse failure reported: parameter pack "Indexes" was referenced but not expanded

CPP-43893	Improve mapping for CERT EXP45-C (remove CERT_C-EXP45-a and CERT_C-EXP45-c)
CPP-43896	Improve unit testing execution for Renesas Rx
CPP-43971	Enable <code>edg.implicit_noexcept_enabled</code> configuration option for GCC and Clang compilers
CPP-43972	C/C++test fails to read "\$NULL" value from a data source
CPP-43975	Rule CODSTA-149 (CERT_C-MS17-a) reports false positive when fall through comment is preceded by preprocessor directive
CPP-43992	TempLic*.txt files create and not cleaned up in temp folder
CPP-44001	VS IDE not responding when creating test case for <code>CMFCSampleDlg::OnPaint()</code>
CPP-44025	Rule CERT_C-INT36-a reports false positive when '0' is cast to void* type
CPP-44045	Rule OPT-06 reports false positive on local variable captured in lambda
CPP-44046	STATUS_ACCESS_VIOLATION: The thread attempts to read from or write to a virtual address for which it does not have access.
CPP-44055	VS Only first <code>-localsettings</code> parameter is handled by C/C++test (others are ignored)
CPP-44059	Report HTML - Tested functions in Test Cases have empty field
CPP-44088	Static Analysis (cwc) exits with code 3 on literal variadic templates
CPP-44225	Rule MISRA2004-12_8 (MISRA2012-RULE_12_2-a) reports false positive when double cast of the operand is used in the shift expression
CPP-44271	Parse failure reported: expression must have a constant value <code>static constexpr bool value = has_named_enum_tag<T>(0);</code>
CPP-44273	Renaming a test case actually renames the class name for that test
CPP-44274	Rule HICPP-17_2_1-a (AUTOSAR-A17_1_1-b) reports false positive on <code>#include <string></code>
CPP-44538	Add support for missing IAR atomic builtins
CPP-44576	C++test 10.4.3 BETA - Command line analysis is not licensed
FA-4617	False positives from BD-PB-DEREF on checking array variable against being null
FA-4651	BD-RES-FREE False Positive on freeing memory that was already freed as a resource of another type (e.g. pthread mutex)
FA-4998	Bogus violation for BD-RES-FREE on arithmetic operations done on closed file descriptors.
FA-7097	BD-PB-PTRARR false positive on type mismatch
FA-7105	BD-PB-OVERFWR False Positive
FA-7191	BD-RES-INVFREE false positive when working with const expression
FA-7195	BD-CO-ITOUT - false positive for container <code>end()</code> method
FA-7266	Incorrect Flow Analysis results: FA does not take into account values of the elements of the global array of consts.
FA-7291	False positives from BD-RES-INVFREE when closing resource referenced by the element of an array.
FA-7398	Flow Analysis Aggressive reports static analysis problems in C++test 10.4.2
FA-7410	False positive for BD-SECURITY-OVERFFMT when typedefs used
FA-7413	False positive of MISRA2012-RULE-19-1_c (BD-PB-OVERLAP)
FA-7441	CERT_C-ARR38-c (BD-PB-OVERFFMT) reports FP violation when specifying %*s inside string format
XT-36609	£ character in password prevents Parasoft tool from connecting to DTP
XT-36611	Publishing sim-link source code using 'min' option failed
XT-36843	Concurrent builds which use <code>cpptestcli</code> do not wait for timeout when trying to pull license
XT-36950	Update vulnerable libraries from XML Graphics Project
XT-37358	100% not being displayed in reports when achieving 100% test success

New Rules

Rule ID	header
AUTOSAR-A0_1_5-a	There shall be no unused named parameters in virtual functions
AUTOSAR-A12_1_3-a	User-defined constructors that initialize data members with the same constant values across all constructors should initialize using NSDMI instead
AUTOSAR-A12_1_6-a	Derived classes that do not need further explicit initialization and require all the constructors from the base class shall use inheriting constructors
AUTOSAR-A15_3_4-a	Avoid using catch-all exception handlers
AUTOSAR-A15_4_5-a	Checked exceptions that could be thrown from a function shall be specified in the comment directly before the function declaration
AUTOSAR-A15_5_2-c	The 'quick_exit()' and '_Exit()' functions from the 'stdlib.h' or 'cstdlib' library shall not be used
AUTOSAR-A1_1_1-a	The 'register' storage class specifier shall not be used
AUTOSAR-A1_1_1-b	A copy assignment operator should be declared when a copy constructor is declared (and vice versa)
AUTOSAR-A1_1_1-c	Both copy constructor and copy assignment operator should be declared for classes with a nontrivial destructor
AUTOSAR-A1_1_1-d	The C library shall not be used
AUTOSAR-A1_1_1-e	Prefer lambdas over std::bind, std::bind1st and std::bind2nd
AUTOSAR-A1_1_1-f	The 'binder1st' and 'binder2nd' identifiers should not be used
AUTOSAR-A1_1_1-g	Prefer to use std::unique_ptr instead of std::auto_ptr
AUTOSAR-A1_1_1-h	The 'random_shuffle' identifier should not be used
AUTOSAR-A1_1_1-i	Do not use the increment operator (++) on an operand of type 'bool'
AUTOSAR-A1_1_1-j	The 'set_unexpected' identifier should not be used
AUTOSAR-A1_1_1-k	Do not use throw exception specifications
AUTOSAR-A27_0_4-a	Don't use unsafe C functions that do write to range-unchecked buffers
AUTOSAR-A27_0_4-b	Avoid using unsafe string functions that do not check bounds
AUTOSAR-A27_0_4-c	Do not use the 'char' buffer to store input from 'std::cin'
AUTOSAR-A27_0_4-d	C-style strings shall not be used
AUTOSAR-A2_10_4-a	The identifier name of a non-member object with static storage duration shall not be reused within a namespace
AUTOSAR-A2_10_4-b	The identifier name of a non-member static function shall not be reused within a namespace
AUTOSAR-A2_7_3-a	All declarations of types, data members, and functions should be preceded by a comment annotated with the '@brief' tag
AUTOSAR-A2_7_3-b	Function parameters and return type should be documented in a comment that precedes the function declaration
AUTOSAR-A3_3_2-a	Static and thread-local objects shall be constant-initialized
AUTOSAR-A5_1_6-a	Return type of a non-void return type lambda expression should be explicitly specified
AUTOSAR-A5_1_8-a	Lambda expressions should not be defined inside another lambda expression
AUTOSAR-A5_3_1-a	The operand of the 'typeid' operator shall not contain any expression that has side effects
AUTOSAR-A5_3_1-b	The operand of the 'typeid' operator shall not contain a function call that causes side effects
AUTOSAR-A6_2_1-a	Copy assignment operators should not have side effects that could affect copying the object
AUTOSAR-A6_2_1-b	Move assignment operators should not have side effects that could affect moving the object
AUTOSAR-A6_2_2-a	Expression statements shall not be explicit calls to constructors of temporary objects only
AUTOSAR-A7_1_5-a	Do not overuse 'auto' specifier
AUTOSAR-A8_2_1-a	Use a trailing return type syntax if the return type is preceded by the 'typename' keyword
AUTOSAR-A8_4_8-a	Output parameters shall not be used
AUTOSAR-A8_5_2-a	Braced-initialization {}, without equals sign, shall be used for variable initialization
AUTOSAR-A8_5_3-a	A variable of type auto shall not be initialized using '{}' or '= {}' braced-initialization

AUTOSAR-M15_3_7-a	Where multiple handlers are provided in a single 'try-catch' statement or 'function-try-block', any ellipsis (catch-all) handler shall occur last
AUTOSAR-M18_0_3-b	The 'exit()' function from the 'stdlib.h' or 'cstdlib' library shall not be used
AUTOSAR-M18_0_3-c	The 'system()' function from the 'stdlib.h' or 'cstdlib' library shall not be used
AUTOSAR-M18_0_3-d	The 'getenv()' function from the 'stdlib.h' or 'cstdlib' library shall not be used
BD-RES-INSUFMEM	Allocate sufficient memory to hold an object of a given type
BD-SECURITY-XXEXRC	Disable resolving XML external entities (XXE) in libxerces-c
CERT_C-ERR04-b	The 'exit()' function from the 'stdlib.h' or 'cstdlib' library shall not be used
CERT_C-ERR04-c	The 'quick_exit()' and '_Exit()' functions from the 'stdlib.h' or 'cstdlib' library shall not be used
CERT_C-ERR05-b	The 'exit()' function from the 'stdlib.h' or 'cstdlib' library shall not be used
CERT_C-ERR05-c	The 'quick_exit()' and '_Exit()' functions from the 'stdlib.h' or 'cstdlib' library shall not be used
CERT_C-INT31-o	Avoid integer overflows
CERT_CPP-ERR50-n	The 'quick_exit()' and '_Exit()' functions from the 'stdlib.h' or 'cstdlib' library shall not be used
CERT_CPP-EXP52-d	The operand of the 'typeid' operator shall not contain any expression that has side effects
CERT_CPP-EXP52-e	The operand of the 'typeid' operator shall not contain a function call that causes side effects
CODSTA-204	Functions declared as 'noreturn' shall have the 'void' return type
CODSTA-CPP-103	Output parameters shall not be used
CODSTA-CPP-104_b	The operand of the 'typeid' operator shall not contain a function call that causes side effects
CODSTA-CPP-104	The operand of the 'typeid' operator shall not contain any expression that has side effects
CODSTA-MCPP-07_b	The 'binder1st' and 'binder2nd' identifiers should not be used
CODSTA-MCPP-13_b	The 'std::forward' function shall be used to forward universal references
CODSTA-MCPP-32	Static and thread-local objects shall be constant-initialized
CODSTA-MCPP-37	Derived classes that do not need further explicit initialization and require all the constructors from the base class shall use inheriting constructors
CODSTA-MCPP-38	Braced-initialization {}, without equals sign, shall be used for variable initialization
CODSTA-MCPP-39	A variable of type auto shall not be initialized using '{}' or '= {}' braced-initialization
CODSTA-MCPP-40	Do not overuse 'auto' specifier
CODSTA-MCPP-41	The 'random_shuffle' identifier should not be used
CODSTA-MCPP-42	Do not use the increment operator (++) on an operand of type 'bool'
CODSTA-MCPP-43	The 'set_unexpected' identifier should not be used
CODSTA-MCPP-44	Lambda expressions should not be defined inside another lambda expression
CODSTA-MCPP-45	Return type of a non-void return type lambda expression should be explicitly specified
CODSTA-MCPP-46	Include a parameter list in every lambda expression
COMMENT-04_b	Document functions in comments that precede function declarations
COMMENT-14_b	Function parameters and return type should be documented in a comment that precedes the function declaration
COMMENT-14	All declarations of types, data members, and functions should be preceded by a comment annotated with the '@brief' tag
CWE-119-a	Avoid accessing arrays out of bounds
CWE-119-b	Avoid accessing arrays and pointers out of bounds
CWE-119-c	Avoid buffer overflow due to defining incorrect format limits
CWE-119-d	Avoid overflow when reading from a buffer
CWE-119-e	Avoid overflow when writing to a buffer
CWE-119-f	Avoid tainted data in array indexes

CWE-119-g	Prevent buffer overflows from tainted data
CWE-119-h	Avoid buffer read overflow from tainted data
CWE-119-i	Avoid buffer write overflow from tainted data
CWE-119-j	Suspicious use of 'strcpy' without checking size of source buffer
CWE-125-a	Avoid accessing arrays out of bounds
CWE-125-b	Avoid accessing arrays and pointers out of bounds
CWE-125-c	Avoid overflow when reading from a buffer
CWE-125-d	Avoid buffer read overflow from tainted data
CWE-190-a	Avoid integer overflows
CWE-190-b	Avoid possible integer overflow in expressions in which the result is cast to a wider integer type
CWE-190-c	Avoid possible integer overflow in expressions in which the result is assigned to a variable of a wider integer type
CWE-190-d	Avoid possible integer overflow in expressions in which the result is compared to an expression of a wider integer type
CWE-190-e	Integer overflow or underflow in constant expression in '+', '-', '*' operator
CWE-190-f	Integer overflow or underflow in constant expression in '<<' operator
CWE-190-g	Evaluation of constant unsigned integer expressions should not lead to wrap-around
CWE-20-a	Avoid tainted data in array indexes
CWE-20-b	Protect against integer overflow/underflow from tainted data
CWE-20-c	Avoid passing unvalidated binary data to log methods
CWE-20-d	Protect against command injection
CWE-20-e	Avoid printing tainted data on the output console
CWE-20-f	Protect against environment injection
CWE-20-g	Exclude unsanitized user input from format strings
CWE-20-h	Protect against SQL injection
CWE-20-i	Protect against file name injection
CWE-20-j	Untrusted data is used as a loop boundary
CWE-200-a	Do not print potentially sensitive information, resulting from an application error into exception messages
CWE-22-a	Protect against file name injection
CWE-269-a	Observe correct revocation order while relinquishing privileges
CWE-269-b	Ensure that privilege relinquishment is successful
CWE-287-a	Do not use weak encryption functions
CWE-326-a	Do not use weak encryption functions
CWE-362-a	Usage of functions prone to race is not allowed
CWE-362-b	Avoid race conditions while accessing files
CWE-362-c	Use locks to prevent race conditions when modifying bit fields
CWE-362-d	Avoid race conditions when using fork and file descriptors
CWE-362-e	Do not use global variable with different locks set
CWE-400-a	Do not create variables on the stack above the defined limits
CWE-415-a	Do not use resources that have been freed
CWE-416-a	Do not use resources that have been freed
CWE-416-b	Do not point to a wrapped object that has been freed
CWE-416-c	Freed memory shouldn't be accessed under any circumstances

CWE-426-a	Use care to ensure that LoadLibrary() will load the correct library
CWE-476-a	Avoid null pointer dereferencing
CWE-476-b	Do not check for null after dereferencing
CWE-611-a	Disable resolving XML external entities (XXE) in libxerces-c
CWE-617-a	Do not use assertions
CWE-704-a	Conversions shall not be performed between a pointer to a function and any other type than pointer to function
CWE-704-b	Conversions shall not be performed between non compatible pointer to a function types
CWE-704-c	Conversions shall not be performed between a pointer to an incomplete type and any other type
CWE-704-d	A cast shall not be performed between a pointer to object type and a pointer to a different object type
CWE-704-e	A conversion should not be performed between a pointer to object type and an integer type other than 'uintptr_t' or 'intptr_t'
CWE-704-f	A conversion should not be performed from pointer to void into pointer to object
CWE-704-g	A cast shall not be performed between pointer to void and an arithmetic type
CWE-704-h	An implicit conversion shall not be performed between pointer to void and an arithmetic type
CWE-704-i	A cast shall not be performed between pointer to object and a non-integer arithmetic type
CWE-704-j	Implicit conversions from wider to narrower integral type which may result in a loss of information shall not be used
CWE-704-k	Implicit conversions from integral to floating type which may result in a loss of information shall not be used
CWE-704-l	Implicit conversions from integral constant to floating type which may result in a loss of information shall not be used
CWE-732-a	Call 'umask' before calling 'mkstemp'
CWE-732-b	Specify the access permission bits if a file is created using the 'open' or 'openat' system call
CWE-770-a	Ensure resources are freed
CWE-772-a	Ensure resources are freed
CWE-772-b	Define a virtual destructor in classes used as base classes which have virtual functions
CWE-78-a	Protect against command injection
CWE-787-a	Avoid accessing arrays out of bounds
CWE-787-b	Avoid accessing arrays and pointers out of bounds
CWE-787-c	Avoid buffer overflow due to defining incorrect format limits
CWE-787-d	Avoid overflow when writing to a buffer
CWE-787-e	Prevent buffer overflows from tainted data
CWE-787-f	Avoid buffer write overflow from tainted data
CWE-798-a	Do not hard code string literals
CWE-835-a	Avoid infinite loops
CWE-863-a	Do not use 'cuserid' function
CWE-89-a	Protect against SQL injection
EXCEPT-22	Checked exceptions that could be thrown from a function shall be specified in the comment directly before the function declaration
EXCEPT-23	Do not use throw exception specifications
EXCEPT-24	Where multiple handlers are provided in a single 'try-catch' statement or 'function-try-block', any ellipsis (catch-all) handler shall occur last
EXCEPT-25	Do not leave 'catch' blocks empty
EXCEPT-26	Avoid using catch-all exception handlers
GLOBAL-REUSEDQUALGLOBVAR	The identifier name of a non-member object with static storage duration shall not be reused within a namespace

GLOBAL-REUSEDQUALSTATF UN	The identifier name of a non-member static function shall not be reused within a namespace
HICPP-17_2_1-b	The error indicator 'errno' shall not be used
HICPP-5_1_6-e	The operand of the 'typeid' operator shall not contain any expression that has side effects
HICPP-5_1_6-f	The operand of the 'typeid' operator shall not contain a function call that causes side effects
INIT-17	User-defined constructors that initialize data members with the same constant values across all constructors should initialize using NSDMI instead
JSF-024_b	The library function 'exit' of <stdlib.h> shall not be used
JSF-024_c	The library function 'getenv' of <stdlib.h> shall not be used
JSF-024_d	The library function 'system' of <stdlib.h> shall not be used
JSF-134_b	Document functions in comments that precede function declarations
MISRA2004-20_11_b	The 'exit()' function from the 'stdlib.h' or 'cstdlib' library shall not be used
MISRA2004-20_11_c	The 'getenv()' function from the 'stdlib.h' or 'cstdlib' library shall not be used
MISRA2004-20_11_d	The 'system()' function from the 'stdlib.h' or 'cstdlib' library shall not be used
MISRA2008-18_0_3_b	The library function 'exit' of <stdlib.h> shall not be used
MISRA2008-18_0_3_c	The library function 'getenv' of <stdlib.h> shall not be used
MISRA2008-18_0_3_d	The library function 'system' of <stdlib.h> shall not be used
MISRA2012-RULE-21_8_b	The library function 'exit' of <stdlib.h> shall not be used
MISRA2012-RULE-21_8_c	The library function 'getenv' of <stdlib.h> shall not be used
MISRA2012-RULE-21_8_d	The library function 'system' of <stdlib.h> shall not be used
MISRA2012-RULE-2_2_b	Avoid unused values
MISRAC2012-RULE_21_8-b	The 'exit()' function from the 'stdlib.h' or 'cstdlib' library shall not be used
MISRAC2012-RULE_21_8-c	The 'system()' function from the 'stdlib.h' or 'cstdlib' library shall not be used
MISRAC2012-RULE_21_8-d	The 'getenv()' function from the 'stdlib.h' or 'cstdlib' library shall not be used
MISRAC2012-RULE_2_2-b	Avoid unused values
MRM-56	Copy assignment operators should not have side effects that could affect copying the object
MRM-57	Move assignment operators should not have side effects that could affect moving the object
OPT-42	There shall be no unused named parameters in virtual functions
OWASP2017-A1-a	Avoid passing unvalidated binary data to log methods
OWASP2017-A1-b	Protect against command injection
OWASP2017-A1-c	Avoid printing tainted data on the output console
OWASP2017-A1-d	Protect against environment injection
OWASP2017-A1-e	Exclude unsanitized user input from format strings
OWASP2017-A1-f	Protect against SQL injection
OWASP2017-A10-a	All exceptions should be rethrown or logged with standard logger
OWASP2017-A2-a	Do not use weak encryption functions
OWASP2017-A3-a	Properly seed pseudorandom number generators

OWASP2017-A4-a	Disable resolving XML external entities (XXE) in libxerces-c
OWASP2017-A5-a	Protect against file name injection
OWASP2017-A5-b	Observe correct revocation order while relinquishing privileges
OWASP2017-A5-c	Ensure that privilege relinquishment is successful
OWASP2017-A6-a	Where multiple handlers are provided in a single try-catch statement or function-try-block for a derived class and some or all of its bases, the handlers shall be ordered most-derived to base class
OWASP2017-A6-b	Do not leave 'catch' blocks empty
OWASP2017-A6-c	Properly use errno value
PB-75_b	The 'exit()' function from the 'stdlib.h' or 'cstdlib' library shall not be used
PB-75_c	The 'quick_exit()' and '_Exit()' functions from the 'stdlib.h' or 'cstdlib' library shall not be used
PB-76	C-style strings shall not be used
PB-77	Expression statements shall not be explicit calls to constructors of temporary objects only
SECURITY-48_b	The 'system()' function from the 'stdlib.h' or 'cstdlib' library shall not be used
SECURITY-51	Do not use the 'char' buffer to store input from 'std::cin'
SECURITY-52	The 'getenv()' function from the 'stdlib.h' or 'cstdlib' library shall not be used
TEMPL-17	Use a trailing return type syntax if the return type is preceded by the 'typename' keyword

Updated Rules

We've updated following static analysis rules to improve analysis results:

Rule Category	Rule IDs
AUTOSAR C++14 Coding Guidelines	AUTOSAR-A0_1_4-a, AUTOSAR-A12_1_1-b, AUTOSAR-A12_8_4-a, AUTOSAR-A13_5_4-b, AUTOSAR-A15_4_1-a, AUTOSAR-A15_5_2-b, AUTOSAR-A18_0_1-a, AUTOSAR-A18_9_2-a, AUTOSAR-A27_0_1-d, AUTOSAR-A27_0_2-b, AUTOSAR-A2_8_1-a, AUTOSAR-A3_8_1-a, AUTOSAR-A3_8_1-b, AUTOSAR-A5_1_3-a, AUTOSAR-A5_2_5-c, AUTOSAR-A8_4_5-a, AUTOSAR-A8_4_6-a, AUTOSAR-A8_5_0-a, AUTOSAR-M0_1_3-a, AUTOSAR-M0_1_3-b, AUTOSAR-M0_3_1-b, AUTOSAR-M0_3_1-c, AUTOSAR-M0_3_1-h, AUTOSAR-M0_3_1-j, AUTOSAR-M18_0_3-a, AUTOSAR-M2_13_2-a, AUTOSAR-M5_0_16-b, AUTOSAR-M5_0_18-a, AUTOSAR-M5_0_2-e, AUTOSAR-M5_3_4-c, AUTOSAR-M5_8_1-a, AUTOSAR-M7_5_1-a, AUTOSAR-M8_5_2-c
Flow Analysis	BD-CO-ITOUT, BD-PB-CC, BD-PB-CHECKRET, BD-PB-DEREF, BD-PB-NOTINIT, BD-PB-OVERFFMT, BD-PB-OVERFWR, BD-PB-OVERLAP, BD-PB-PTRARR, BD-PB-SIGHAN, BD-RES-FREE, BD-RES-INVFREE, BD-SECURITY-OVERFFMT, BD-TRS-MLOCK, BD-TRS-ORDER
SEI CERT C	CERT_C-API01-a, CERT_C-ARR38-b, CERT_C-ARR38-c, CERT_C-CON31-b, CERT_C-CON31-c, CERT_C-DCL10-a, CERT_C-DCL11-a, CERT_C-DCL11-b, CERT_C-DCL11-c, CERT_C-DCL11-d, CERT_C-DCL11-e, CERT_C-DCL11-f, CERT_C-DCL18-b, CERT_C-DCL30-a, CERT_C-ENV01-c, CERT_C-ERR04-a, CERT_C-ERR05-a, CERT_C-EXP33-a, CERT_C-EXP39-d, CERT_C-EXP44-b, CERT_C-FIO46-a, CERT_C-FIO47-a, CERT_C-FIO47-b, CERT_C-FIO47-c, CERT_C-FIO47-d, CERT_C-FIO47-e, CERT_C-FIO47-f, CERT_C-INT31-i, CERT_C-MEM00-d, CERT_C-MEM01-a, CERT_C-MEM30-a, CERT_C-MEM34-a, CERT_C-MS313-a, CERT_C-MS314-a, CERT_C-MS315-a, CERT_C-MS324-b, CERT_C-POS51-a, CERT_C-SIG30-a, CERT_C-SIG31-a, CERT_C-SIG34-a, CERT_C-STR31-b
SEI CERT C++	CERT_CPP-CON53-a, CERT_CPP-ERR50-l, CERT_CPP-EXP52-c, CERT_CPP-EXP53-a, CERT_CPP-EXP54-a, CERT_CPP-EXP54-b, CERT_CPP-MEM50-a, CERT_CPP-MS354-a, CERT_CPP-STR50-c
Coding Conventions	CODSTA-102, CODSTA-103, CODSTA-116, CODSTA-13, CODSTA-163_b, CODSTA-22
Coding Conventions for C++	CODSTA-CPP-59, CODSTA-CPP-86
Coding Conventions for Modern C++	CODSTA-MCPP-10_a CODSTA-MCPP-13
Comments	COMMENT-04
Formatting	FORMAT-43
High	

Integrity C++	HICPP-12_4_2-a, HICPP-17_2_1-a, HICPP-17_3_2-a, HICPP-18_2_2-a, HICPP-18_3_2-a, HICPP-1_2_1-i, HICPP-1_3_1-a, HICPP-1_3_3-a, HICPP-1_3_5-a, HICPP-2_5_2-a, HICPP-3_4_1-a, HICPP-3_5_1-d, HICPP-4_2_2-a, HICPP-5_1_3-a, HICPP-5_1_5-a, HICPP-5_1_6-c, HICPP-7_1_7-a, HICPP-8_4_1-a, HICPP-8_4_1-b
Initialization	INIT-05, INIT-06
Joint Strike Fighter	JSF-024, JSF-060_b, JSF-071_b, JSF-077, JSF-085_a, JSF-111, JSF-117_b, JSF-134, JSF-139, JSF-143_a, JSF-149, JSF-164, JSF-166_c, JSF-171, JSF-181_a, JSF-203, JSF-204_a, JSF-204_b
MISRA C 1998	MISRA-027, MISRA-044, MISRA-051
MISRA C 2004	MISRA2004-12_1_e, MISRA2004-12_3_c, MISRA2004-12_8, MISRA2004-17_3, MISRA2004-17_6_a, MISRA2004-20_11, MISRA2004-7_1_a, MISRA2004-9_2_c
MISRA C++ 2008	MISRA2008-0_1_11, MISRA2008-0_1_3_a, MISRA2008-0_1_3_b, MISRA2008-0_3_1_d, MISRA2008-0_3_1_f, MISRA2008-0_3_1_h, MISRA2008-0_3_1_j, MISRA2008-18_0_1, MISRA2008-18_0_3, MISRA2008-2_13_2_a, MISRA2008-5_0_16_b, MISRA2008-5_0_18, MISRA2008-5_0_2_e, MISRA2008-5_3_4_c, MISRA2008-5_8_1, MISRA2008-7_5_1, MISRA2008-7_5_2_a, MISRA2008-8_5_2_c
MISRA C 2012	MISRAC2012-DIR_4_1-d, MISRAC2012-DIR_4_1-f, MISRAC2012-DIR_4_1-h, MISRAC2012-DIR_4_1-j, MISRAC2012-DIR_4_13-b, MISRAC2012-DIR_4_13-c, MISRAC2012-DIR_4_14-i, MISRAC2012-DIR_4_7-a, MISRAC2012-RULE_10_3-b, MISRAC2012-RULE_12_1-a, MISRAC2012-RULE_12_2-a, MISRAC2012-RULE_14_3-ac, MISRAC2012-RULE_16_1-g, MISRAC2012-RULE_16_5-a, MISRAC2012-RULE_18_1-c, MISRAC2012-RULE_18_3-a, MISRAC2012-RULE_18_6-a, MISRAC2012-RULE_19_1-c, MISRAC2012-RULE_1_3-b, MISRAC2012-RULE_1_3-c, MISRAC2012-RULE_1_3-e, MISRAC2012-RULE_1_3-m, MISRAC2012-RULE_21_17-b, MISRAC2012-RULE_21_8-a, MISRAC2012-RULE_22_2-a, MISRAC2012-RULE_22_2-b, MISRAC2012-RULE_22_6-a, MISRAC2012-RULE_7_1-a, MISRAC2012-RULE_9_1-a
Memory and Resource Management	MRM-41
Naming Conventions	NAMING-32
Optimization	OPT-02, OPT-02, OPT-03, OPT-05, OPT-06, OPT-29, OPT-31
Possible Bugs	PB-11, PB-18, PB-22, PB-23, PB-45, PB-46, PB-47, PB-48, PB-49, PB-50, PB-73, PB-75
Security	SECURITY-14

Removed Rules

The following rules have been removed:

- AUTOSAR-A13_5_4-a
- AUTOSAR-A17_1_1-b
- CERT_C-ENV33-b
- CERT_C-EXP45-a
- CERT_C-EXP45-c
- CERT_C-FLP37-a
- CERT_C-FLP37-b
- CERT_C-INT36-a