

Updates in 10.4.0

In this release, we've focused on extending support for new compilers, as well as creating Compliance Packs that include new and enhanced test configurations:

- [Support for Environments](#)
- [Compliance Packs](#)
- [New and Updated Test Configurations](#)
- [New and Updated Code Analysis Rules](#)
- [Dynamic Stub Configuration with Stub Callbacks](#)
- [Enhanced Test Case Editor](#)
- [Reporting Test Execution Details](#)
- [Other Changes](#)
- [Resolved Bugs and FRs](#)

Support for Environments

New Compilers

We've added support for the following compilers:

Compiler Name	Compiler Acronym
Altium TASKING Vx-toolset for TriCore C/C++ Compiler 6.0	vxtc_6_0
Altium TASKING Vx-toolset for TriCore C/C++ Compiler 6.2	vxtc_6_2
ARM Compiler 6.6	armclang_6_6
ARM Compiler 6.9	armclang_6_9
Clang C/C++ Compiler v 3.9	clang_3_9
Clang C/C++ Compiler v 4.0	clang_4_0
Clang C/C++ Compiler v 5.0	clang_5_0
Clang C/C++ Compiler v 6.0	clang_6_0
Freescale CodeWarrior C/C++ Compiler v. 6.0 for ColdFire	cwcf_6_0
GCC for Tricore 4.9.x	tricoregcc_4_9
GNU GCC 7.x ¹	gcc_7
GNU GCC 7.x (x86_64) ¹	gcc_7-64
Green Hills Software Compiler for ARM v. 2014.1.x	ghsarm_2014_1
Green Hills Software Compiler for ARM v. 2017.5.x	ghsarm_2017_5
Green Hills Software Compiler for ARM64 v. 2014.1.x	ghsarm64_2014_1
Green Hills Software Compiler for ARM64 v. 2017.5.x	ghsarm64_2017_5
Green Hills Software Compiler for V850 v. 2013.5.x	ghsv850_2013_5
Green Hills Software Compiler for V850 v. 2014.1.x	ghsv850_2014_1
Green Hills Software Compiler for V850 v. 2017.5.x	ghsv850_2017_5
IAR Compiler for ARM v. 8.20.x	iccarm_8_20
IAR Compiler for RX v. 3.10.x	iccrx_3_10

Intel(R) C++ Compiler v 18.0	intelicc_18_0
QNX GCC 5.x	qcc_5
QNX GCC 5.x (ARM)	qccarm_5
QNX GCC 5.x (ARM64)	qccarm_5-64
QNX GCC 5.x (x86-64)	qcc_5-64
TI ARM C/C++ Compiler GNU GCC 7.x	tiarmgcc_7
TI ARM C/C++ Compiler v18.1	tiarm_18_1
TI MSP430 C/C++ Compiler GNU GCC 6.x	timsp430gcc_6
TI MSP430 C/C++ Compiler v18.1	timsp430_18_1
TI TMS320C2000 C/C++ Compiler v16.9	tic2000_16_9
TI TMS320C2000 C/C++ Compiler v18.1	tic2000_18_1
TI TMS320C6x C/C++ Compiler v8.2	tic6000_8_2

The compilers are now marked with the following support levels:

- **Extended:** Support has been validated with extended testing and is approved for use in safety-critical software development.
- **Standard:** Support has been validated with standard testing and is approved for use in non-safety critical software development.

See [Compilers](#) for details about currently supported compilers.

¹ Limited support for the C++17 standard for GNU GCC or Microsoft Visual C++ compilers
C/C++test may not be able to analyze code that uses the latest versions of GNU GCC or Microsoft Visual C++ compilers with C++17 extensions enabled. The analysis results may also be inaccurate. C/C++test does not support the following C++17 extensions:

- P0017R1 Extension to aggregate initialization
- P0091R4 Template argument deduction for class templates
- P0127R2 Declaring non-type template parameters with auto
- P0195R2 Pack expansions in using-declarations
- P0522R0 Matching of template template-arguments excludes compatible templates

New IDEs

We've added support for the following IDEs:

- Texas Instruments Code Composer Studio 7.4
- Texas Instruments Code Composer Studio 8.0

Removed Environments

See [Removed Support for Environments](#) for information about environments that are no longer supported.

Compliance Packs

In this release, we've introduced Compliance Packs to give you instant access to test configurations that help you enforce industry-specific compliance standards and practices. Compliance Packs leverage a broad range of C/C++test's testing capabilities tailored for particular compliance domains.

The following Compliance Packs are available:

Compliance Pack	Test Configurations
-----------------	---------------------

Aerospace Pack	<ul style="list-style-type: none"> • Joint Strike Fighter • DO178C Software Level A Unit Testing • DO178C Software Level B Unit Testing • DO178C Software Level C and D Unit Testing
Automotive Pack	<ul style="list-style-type: none"> • AUTOSAR C++14 Coding Guidelines • HIS Source Code Metrics • High Integrity C++ • MISRA C 1998 • MISRA C 2004 • MISRA C 2012 • MISRA C++ 2008 • ISO26262 ASIL A Unit Testing • ISO26262 ASIL B and C Unit Testing • ISO26262 ASIL D Unit Testing
Medical Devices Pack	<ul style="list-style-type: none"> • Recommended Rules for FDA (C) • Recommended Rules for FDA (C++)
Security Pack	<ul style="list-style-type: none"> • CWE SANS Top 25 • OWASP Top 10 2017 • Payment Card Industry Data Security Standard • SEI CERT C Guidelines • Security Rules • UL 2900

Compliance Packs require dedicated license features to be activated. Contact Parasoft Support for more details on Compliance Packs licensing.

New and Updated Test Configurations

We've added the following built-in test configurations:

- AUTOSAR C++14 Coding Guidelines
- High Integrity C++
- SEI CERT C Guidelines
- UL 2900
- OWASP Top 10 2017
- DO178C Unit Testing
- ISO26262 Unit Testing

The following test configurations have been moved from the Static Analysis category to one of the Compliance Packs (see [Compliance Packs](#)):

- Joint Strike Fighter Aerospace Pack
- HIS Source Code Metrics Automotive Pack
- MISRA C 1998 Automotive Pack
- MISRA C 2004 Automotive Pack
- MISRA C 2012 Automotive Pack
- MISRA C++ 2008 Automotive Pack
- CWE SANS Top 25 Security Pack
- PCI Data Security Standard Security Pack
- Security Rules Security Pack

The rules enabled in the MISRA C 2012 built-in test configuration are now differently grouped and have new rule IDs, which may impact their previous suppressions; see [MISRA C 2012 Rules](#) for details.

Parasoft's Recommended FDA C++ Phase 1, 2, and 3 have been replaced with the following test configurations:

- Recommended Rules for FDA (C)
- Recommended Rules for FDA (C++)

The "MISRA C" test configurations have been renamed as "MISRA C 1998".

The following test configurations have been removed:

- OWASP Top 10 Security Vulnerabilities (replaced with OWASP Top 10 2017)
- CERT C Coding Standard (replaced with SEI CERT C Guidelines)
- ISO 26262 Recommended Rules
- DISA-STIG Coding Standard
- SAMATE Annex A Source Code Weaknesses
- CRules

See [Built-in Test Configurations](#) for the list of test configurations shipped with C/C++test.

New and Updated Code Analysis Rules

In this release, we've added new static analysis rules to extend coverage of compliance standards; see [New Rules](#) and [Updated Rules](#) for the lists of new and updated rules.

MISRA C 2012 Rules

The rules that enforce the MISRA C 2012 Standard now have new rule IDs and are all grouped under one category "MISRA C 2012" to facilitate reviewing results and ensure full compatibility with Parasoft's MISRA C 2012 Compliance Pack.

- The layout and IDs of rules enabled in the MISRA C 2012 built-in test configuration have been updated; this may affect the way they are processed by DTP and their previous suppressions.
- The previous layout and IDs of MISRA C 2012 rules have been retained for backward compatibility as the "MISRA C 2012 (Legacy)" rule category – custom test configurations that were created to enforce the MISRA C 2012 Standard with previous versions of C/C++test automatically refer to this category and do not require updating.

The above changes do not impact rule implementations, which are identical for the "MISRA C 2012" and "MISRA C 2012 (Legacy)" rule categories.

Dynamic Stub Configuration with Stub Callbacks

The Stub Callbacks framework allows you to specify test case-specific stub behavior through the Test Case Editor interface or by manually modifying test case source code. You can define a specific stub logic that will be performed each time a stub is called during test case execution.

The Stub API for dynamic stub configuration has been deprecated and is now disabled by default.

See [Dynamic Stubs Configuration](#) for details.

Enhanced Test Case Editor

We've enhanced the Test Case Editor to facilitate modifying test suites and test cases.

- The new "Stub Configuration" and "Stub Expectations" steps allow you to configure stub behavior using the Stub Callbacks framework (see [Using Stub Callbacks](#) for details).
- The "Additional Test Suite Code" editor enables you to configure code sections that are automatically added to your test suite code.

See [Adding Test Suites and Test Cases with the Test Case Editor](#) for details.

Reporting Test Execution Details

You can now generate an additional Test Execution Details Report that is linked to your regular report. The new report includes additional information about test execution – especially useful for teams developing safety critical applications. See [Understanding Reports](#).

Other Changes

- C/C++test Professional ships with C/C++test Standard (formerly DTP Engine for C/C++). The C/C++test Standard distribution is located in the C/C++test Professional installation directory. To use C/C++test Standard, unpack the distribution to a target location other than C/C++test Professional installation directory. See [Parasoft C/C++test Standard](#) for details.
- The option for generating the coverage details report has been moved from the "Reports" preference page to the "Test Configurations" page: go to Execution> Runtime> Generate detailed coverage report for [coverage metric].
- DTP 5.4.0 is required to leverage DTP capabilities and workflows.
- Findings marked with the Do Not Show priority on your DTP no longer simulate suppressions and should be converted to true suppressions. See the following DTP documentation: [DTP 5.4.0 Release Notes](#) and [Migrating Team Server Suppression Data to DTP](#).

- The RuleWizard Module has been extended with the following nodes and properties (see the [Rule Wizard documentation](#) for details):
 - the `ExplicitTemplateInstance` node and its properties
 - the `InitializerInClass` property
 - the `ReferenceQualifier` property

Resolved Bugs and FRs

Resolved Bug/FR ID	Description
CPP-40357	Rule MRM-20 should not report violations on allocations of local pointers
CPP-40380	Fix CppUnit integration docs: use <code>CppTest_CppUnitResultsListener</code> not <code>CppTest_CppUnitResultsOutputter</code>
CPP-40384	Review CODSTA-37 rule: floating constants not detected
CPP-40388	Having an <code>#endif</code> before the first <code>#if 0</code> in a test case editor managed test case causes a <code>StringIndexOutOfBounds</code> exception to be thrown
CPP-40402	Memory leak in RS232 Common transport implementation
CPP-40405	Rule MISRA-028 (JSF-140) does not report violation on the 'register' keyword used in parameter declaration
CPP-40566	Property 'fullname' in RuleWizard returns empty string for functions with more than one parameter of the built-in ' <code>__m128i</code> ' type
CPP-40573	Rule INIT-15 reports false positive on structs with aggregate-initialized member
CPP-40574	cpptest GUI crash when <code>LD_LIBRARY_PATH</code> is not set
CPP-40578	Unwanted EDG diagnostic messages in the console when console verbosity is set to 'Normal' or 'Low'
CPP-40579	The CODSTA-63 rules doesn't trigger on enum constants
CPP-40594	<code>MisraCpp2008UnderlyingType</code> property returns incorrect type for unsigned constants expressions
CPP-40702	Rule CODSTA-CPP-78 (MISRA2008-9_3_3) detects function that can not be 'const'
CPP-40719	<code>constexpr</code> template specialization is not being reconstructed correctly in C++14 mode
CPP-40760	Block comment in CODE step may trigger a compilation error
CPP-40761	Test suite data may be corrupted if curly bracket "{" is used in test suite description
CPP-40765	parse error with <code>niclang_3_3</code> - error: expected a ";"
CPP-40780	error: "operator=" has already been declared in the current scope
CPP-40796	OPT-06 False Positives
CPP-40819	Code Analysis for <code>proj_igo9_all.cpp</code> finished with code 4 (cwc)
CPP-40907	internal error: assertion failed: <code>gen_declaration: bad entity kind on source seq list (cp_gen_be.c, line 21265)</code>
CPP-40913	cannot compile the test executable when excel data source contains "
CPP-41045	MISRA2004-14_2 (MISRA2012-RULE-2_2_a) reports false positive on casts to a void type
CPP-41047	<code>multiThread</code> option not working for Static Analysis

CPP-41048	Lots of false warnings reported by C++test parsewr during instrumentation of Tasking code
CPP-41178	Incorrect value of cpptestDebug configuration property may lead to linker errors when debugging test cases
CPP-41239	Rule JSF-138_b should not report violations on definitions of const objects
CPP-41246	MISRA2012-RULE-21_6 (CODSTA-110) reports false negatives on most of stdio.h API
CPP-41332	-fail command for C++test Engines
FA-5903	Inaccurate message from BD-SECURITY-INTOVERF rule (incorrect expression reported as possibly overflowed)
FA-5906	BD-SECURITY-INTOVERF inaccurate presentation/messages when reporting use of the ptr that results from a pointer arithmetic using potentially overflowed value
FA-5994	Cannot define constructor as null not accepting method for BD.EXCEPT.NP
FA-6279	BD-PB-CC False Positive
FA-6280	False Negatives for rule BD-RES-FREE
FA-6402	FA does not recognize iccarm's errno pattern
FA-6140	Not all paths are counted when reporting flowanalysis.output.performance.info for some of the rules.

Removed Support for Environments

IDEs

The following IDEs are no longer supported:

- Eclipse IDE for C/C++ Developers 3.2, 3.3, 3.4, 3.5, 3.6, 3.7
- ARM Workbench IDE for RVDS 3.0, 3.1, 4.0, 4.1
- QNX Momentics IDE 4.x
- Texas Instruments Code Composer Studio 4.x
- Wind River Workbench 2.6, 3.0, 3.1, 3.2
- Visual Studio 2005

Compilers

The following compilers are no longer supported (see [Compilers](#) for details about currently supported compilers):

Compiler Name	Compiler Acronym
ARM Developer Suite 1.2	ads_1_2
Borland C++ Compiler 5.6.x for Win32	bcc32_5_6
CodeGear C++ Compiler 5.9.x for Win32	bcc32_5_9
Wind River Diab 5.0	diab_5_0
Wind River Diab 5.5.x	diab_5_5
Wind River Diab 5.6.x	diab_5_6
eCosCentric GCC 3.4.x	ecosgcc_3_4

Microsoft Embedded Visual C++ 4.0	evc_4_0
Microsoft Visual C++ 8.0 for Windows Mobile	evc_8_0
Microsoft Visual C++ 9.0 for Windows Mobile	evc_9_0
GNU GCC 2.9.x	gcc_2_9
GNU GCC 3.2.x	gcc_3_2
GNU GCC 3.3.x	gcc_3_3
GNU GCC 3.3.x (x86_64)	gcc_3_3-64
GNU GCC 3.4.x	gcc_3_4
GNU GCC 3.4.x (x86_64)	gcc_3_4-64
Green Hills Software Compiler for V850 v. 3.4	ghs_3_4
Green Hills Software Compiler Native v. 4.0.x	ghs_4_0
IAR Compiler for MSP430 v. 4.2x	icc430_4_2
IAR Compiler for MSP430 v. 5.3x	icc430_5_3
IAR Compiler for ARM v. 5.3x	iccarm_5_3
IAR Compiler for ARM v. 5.4x	iccarm_5_4
IAR Compiler for ARM v. 5.5x	iccarm_5_5
Altera Nios II 5.1 b73 GCC 3.4.x	nios2gcc_3_4
Altera Nios GCC 2.9	niosgcc_2_9
QNX GCC 2.9.x	qcc_2_9
QNX GCC 3.3.x	qcc_3_3
ARM RealView 2.2	rvct_2_2
ARM RealView 3.0	rvct_3_0
ARM RealView 3.1	rvct_3_1
ARM RealView 3.1 for uVision	rvct_3_1_uV
ARM RealView 4.0	rvct_4_0
ARM RealView 4.0 for uVision	rvct_4_0_uV
STMicroelectronics ST20	st20_2_2
STMicroelectronics ST40	st40_3_1
TI TMS470 C/C++ Compiler v4.9.x	tiarm_4_9
TI ARM C/C++ Compiler v5.0.x	tiarm_5_0
TI TMS320C2000 C/C++ Compiler v4.1	tic2000_4_1
TI TMS320C2000 C/C++ Compiler v5.2	tic2000_5_2
TI TMS320C2000 C/C++ Compiler v6.0	tic2000_6_0
TI TMS320C54x C/C++ Compiler v4.2	tic54x_4_2
TI TMS320C55x C/C++ Compiler v4.3	tic55x_4_3
TI TMS320C6x C/C++ Compiler v5.1	tic6000_5_1
TI TMS320C6x C/C++ Compiler v6.0	tic6000_6_0
TI TMS320C6x C/C++ Compiler v6.1	tic6000_6_1

TI TMS320C6x C/C++ Compiler v7.0	tic6000_7_0
TI TMS320C6x C/C++ Compiler v7.2	tic6000_7_2
TI MSP430 C/C++ Compiler v3.2	tmsp430_3_2
Microsoft Visual C++ 6.0	vc_6_0
Microsoft Visual C++ 7.0	vc_7_0
Microsoft Visual C++ 7.1	vc_7_1
Microsoft Visual C++ 8.0	vc_8_0
Microsoft Visual C++ 8.0 (x64)	vc_8_0-64
Altium TASKING Vx-toolset for TriCore C/C++ Compiler 2.5	vxtc_2_5
Altium TASKING Vx-toolset for TriCore C/C++ Compiler 3.3	vxtc_3_3
Altium TASKING Vx-toolset for TriCore C/C++ Compiler 3.4	vxtc_3_4
Altium TASKING Vx-toolset for TriCore C/C++ Compiler 3.5	vxtc_3_5
Wind River EGCS 2.9	wregcs_2_9
Wind River GCC 2.9	wrgcc_2_9

New Rules

The following rules have been added:

Rule ID	Header
BD-PB-DNMPTR	Do not modify the alignment of objects by calling realloc()
BD-PB-EXITHAN	Properly define exit handlers
BD-PB-FGETS	Reset strings on fgets() or fgetws() failure
BD-PB-FSETPOS	Only use values for fsetpos() that are returned from fgetpos()
BD-PB-SIGHAN	Properly define signal handlers
BD-PB-SIGRETURN	Do not return from a computational exception signal handler
BD-PB-STDEXC	Always throw created std::exception object
BD-PB-STREAMINOUT	Do not alternately input and output from a stream without an intervening flush or positioning call
BD-PB-SUBSEQFRWD	Do not subsequently use the argument to std::forward
BD-PB-VLASIZE	Ensure the size of the variable length array is in valid range
BD-SECURITY-LOG	Avoid passing unvalidated binary data to log methods
BD-TRS-FRC	Avoid race conditions while accessing files
BD-TRS-JOINDETACH	Do not join or detach a thread that was previously joined or detached
CODSTA-191	Wrap functions that can spuriously wake up in a loop
CODSTA-192	The final member of a structure should not be an array of size '0' or '1'
CODSTA-193	Allocate structures containing a flexible array member dynamically
CODSTA-194	Wrap functions that can fail spuriously in a loop

CODSTA-195	Do not refer to an atomic variable twice in an expression
CODSTA-196	Do not access an array in the result of a function call
CODSTA-198	The value of a complex expression of floating type may only be cast to a narrower floating type
CODSTA-MCPP-19	Declare assignment operators with the ref-qualifier &
CODSTA-MCPP-20	Prefer smart pointers over raw local pointers
CODSTA-MCPP-21	Do not call lock() directly on a mutex
GLOBAL-ONEDEFINLINE	An inline function that is used in multiple translation units shall be defined in one and only one file
GLOBAL-ONEDEFTEMPL	A function template that is used in multiple translation units shall be defined in one and only one file
GLOBAL-ONEDEFTYPE	A type that is used in multiple translation units shall be defined in one and only one file
INIT-16	The initializer for an aggregate or union shall be enclosed in braces
MRM-52	Use RAII to prevent resource leaks
PB-71	Do not copy instances of structures containing a flexible array member
PB-72	Do not call va_arg with an argument of the incorrect type
PB-73	Evaluation of constant unsigned integer expressions should not lead to wrap-around
PREPROC-20	Match the filename in a #include directive to the one on the filesystem
PREPROC-21	Avoid token concatenation that may produce universal character names
SECURITY-49	Use the 'cnd_signal()' function with a unique condition variable
TEMPL-15	Declare 'extern' an explicitly instantiated template

New rules have also been added to the following compliance categories:

Rule ID Prefix	Category
AUTOSAR	AUTOSAR C++14 Coding Guidelines
CERT_C	SEI CERT C
HICPP	High Integrity C++
MISRAC2012	MISRA C 2012

Updated Rules

Rule Category	Rule IDs
Flow Analysis	BD-API-CTYPE, BD-EXCEPT-NP, BD-PB-CC, BD-PB-EXCEPT, BD-PB-INTOVRF, BD-PB-SWITCH, BD-PB-VOVR, BD-RES-FREE, BD-RES-LEAKS, BD-SECURITY-INTOVRF, BD-TRS-ARG, BD-TRS-LOCK, BD-TRS-MLOCK, BD-TRS-ORDER
Coding Conventions	CODSTA-110, CODSTA-166_a, CODSTA-37, CODSTA-47, CODSTA-63, CODSTA-CPP-78
Initialization	INIT-15
Joint Strike Fighter	JSF-138_a, JSF-138_b, JSF-140, JSF-187

MISRA C 1998	MISRA-024, MISRA-028
MISRA C 2004	MISRA2004-14_2, MISRA2004-8_11
MISRA C++ 2008	MISRA2008-0_1_3_b, MISRA2008-0_1_6, MISRA2008-0_1_9, MISRA2008-5_0_21, MISRA2008-9_3_3
MISRA C 2012	MISRA2012-DIR-4_13_a, MISRA2012-DIR-4_13_b, MISRA2012-DIR-4_13_d, MISRA2012-DIR-4_14_b, MISRA2012-RULE-14_3_zc, MISRA2012-RULE-14_3_zd, MISRA2012-RULE-1_3_c, MISRA2012-RULE-21_13, MISRA2012-RULE-21_6, MISRA2012-RULE-22_1, MISRA2012-RULE-22_2_a, MISRA2012-RULE-22_5_a, MISRA2012-RULE-22_6, MISRA2012-RULE-2_2_a, MISRA2012-RULE-8_8
Memory and Resource Management	MRM-09, MRM-20
Object Oriented	OOP-07
Optimization	OPT-06