Updates in 10.3.2

This release includes new features, as well as enhancements to the existing functionality.

Integration with Development Testing Platform

The latest release of C++test features integration with Parasoft Development Testing Platform (DTP) that builds on Parasoft's innovative approach to continuously improving software quality processes. It helps you optimize development processes by aggregating and analyzing local analysis results and converting them into actionable findings that can be imported to your IDE. Test configurations that are stored in DTP can be shared across the team to help you enforce your organization's coding policy*.

See Connecting to DTP for more information.

Support for New Compilers

- GNU GCC 6.x (Windows and Linux)
- IAR Compiler for ARM v.7.8x (Windows)
- Renesas RX C/C++ Compiler 2.5x (Windows)
- ARM Compiler 5.0 / ARM Compiler 5.0 foruVision: updated support for C++11 (–cpp11) (Windows)

Enhancements to Static Analysis

In this release, we've added new rules following the CERT C Coding Standard, as well as modern C++ standards (C++11, C++14, and C++17), see New Code Analysis Rules. The CERT C rules have been added to the CERT C Coding Standard built-in test configuration; see Updated CERT C Configuration f or information about the mapping.

Additionally, stability and accuracy of results reported by Static Analysis and FlowAnalysis have been improved; see Updated Code Analysis Rules for the list of rules that have been modified in terms of accuracy of results or documentation changes.

The RuleWizard Module has been extended with the following nodes and properties:

- C/C++ Nodes> Expressions> Miscellaneous> Lambda node
- C/C++ Nodes> General> Lambda Capture node
- HasDefaultValue property (true for routine parameter with default value)

See RuleWizard 10.3.2 User's Guide for more details.

Collecting Call Coverage

The coverage functionality has been extended to collect information about the number of defined function or method calls were executed at program runtime, see Reviewing Coverage Information.

Other changes and enhancements

- The new "Code parsing problems" option allows you to better customize how Static Analysis is performed when analyzing files with parse errors. This option has replaced the "Analyze files with parse errors" option from C++test 9.6. See Static Tab Settings - Defining How Static Analysis is Performed.
- If you use the command line mode, you can enable sending results to DTP with the `-publish` option, which in C++test 9.x was used to report results to Team Server. In C++test 10.x, publishing results to Team Server can be enabled with the `-publishteamserver` option.
- C++test has been enhanced with the new CPPTEST_ENGINE_EXTRA_ARGS option that allows you to customize advanced Static Analysis settings, such as source file encoding or the memory size; see Configuring Advanced Options.
- Windows 2000, Windows XP, Windows Vista, Windows Server 2003 are no longer supported.

New Code Analysis Rules

Rule ID	Header
BD-PB-BYTEORD	Use the correct byte ordering when transferring data between systems

BD-PB-INVENV	Do not rely on an environment pointer following an operation that may invalidate it
BD-PB-PUTENV	Donotcallputenv() with a pointer to an automatic variable as the argument
BD-SECURITY-RAND	Properly seed pseudorandom number generators
BD-TRS-ARG	Declare objects shared between POSIX threads with appropriate storage durations
BD-TRS-BITLOCK	Use locks to prevent race conditions when modifying bit fields
BD-TRS-DSTRLOCK	Do not destroy another thread's mutex
BD-TRS-FORKFILE	Avoid race conditions when using fork and file descriptors
BD-TRS-REVLOCK	Do not release a lock that has not been acquired
BD-TRS-SYMLINK	Avoid race conditions while checking for the existence of a symbolic link
CODSTA-127_b	A conversion should not be performed between a pointer to object type and an integer type other than 'uintptr_t' or 'intptr_t'
CODSTA-150_c	Avoid side effects in arguments to unsafe macros
CODSTA-187_a	Cast characters to unsigned char before assignment to larger integer sizes
CODSTA-187_b	An expression of the 'signed char' type should not be used as an array index
CODSTA-187_c	Cast characters to unsigned char before converting to larger integer sizes
CODSTA-188	Do not confuse narrow and wide character strings and functions
CODSTA-189	Do not add or subtract a scaled integer to a pointer
CODSTA-190	Do not use object representations to compare floating-point values
CODSTA-MCPP-10_a	Prefer const iterators to iterators
CODSTA-MCPP-10_b	Prefertousecbegin(),crbegin,cend(),crend() functions
CODSTA-MCPP-13	Use std::move() on rvalue references and std::forward() on forwarding references
CODSTA-MCPP-15_a	Avoid default capture modes
CODSTA-MCPP-15_b	Use the 'this' pointer explicitly in lambdas with default by-reference capture
CODSTA-MCPP-16_a	Prefer smart pointer members over raw pointer members
CODSTA-MCPP-16_b	Prefer smart pointers over raw pointers for arrays or STL containers
CODSTA-MCPP-16_c	Prefer 'std::make_shared' to the direct use of new
CODSTA-MCPP-16_d	Prefer to use std::unique_ptr instead of std::auto_ptr
CODSTA-MCPP-17	Never return lambdas that capture local objects by reference
CODSTA-MCPP-18_a	Avoid unnecessary default capture modes in lambda expressions
CODSTA-MCPP-18_b	Avoid unnecessary lambda captures
GLOBAL-CONDMUTEXVAR	Do not use more than one mutex for concurrent waiting operations on a condition variable
SECURITY-02_b	Do not use the rand() function for generating pseudorandom numbers
SECURITY-43	The function 'pthread_setcanceltype()' should not be called with 'PTHREAD_CANCEL_ASYNCHRONOUS' argument
SECURITY-44	Observe correct revocation order while relinquishing privileges
SECURITY-45	Ensure that privilege relinquishment is successful
SECURITY-46	A pointer to a structure should not be passed to a function that can copy data to the user space
SECURITY-47	Use correct integer precisions when checking the right-hand operand of the shift operator
SECURITY-48	Do not call system()

Updated Code Analysis Rules

The verbose mode has been added for the BD-PB-ZERO rule.

Severity levels for MISRA C 2012 rules have been updated:

- Mandatory Severity 1
- Required Severity 2
- Advisory Severity 4

The following rules have been modified:

- BD-PB-ARRAY, BD-PB-ERRNO, BD-PB-EXCEPT, BD-PB-INTOVERF, BD-PB-NOTINIT, BD-PB-NZTS, BD-PB-SWITCH, BD-TRS-TSHL
- CODSTA-144*,CODSTA-145*,CODSTA-150*, CODSTA-150_b*, CODSTA-161_a, CODSTA-CPP-04, CODSTA-CPP-28, CODSTA-CPP-53
- INIT-06, INIT-07, INIT-15
- JSF-071_b, JSF-118, JSF-174_b, JSF-177_b, JSF-180_f, JSF-180_g, JSF-209_b
- MISRA-043_c, MISRA-043_d, MISRA-107_b, MISRA2004-6_3_b, MISRA2004-16_7, MISRA2008-7_1_1, MISRA2008-7_1_2_a, MISRA2012-DIR-4_1_a, MISRA2012-DIR-4_6_b, MISRA2012-RULE-8_13_a, MISRA2012RULE-10_1_a, MISRA2012-RULE-12_4_a*, MISRA2012-RULE-14_3_zd, MISRA2012-RULE-18_1_a
- OOP-01
- PB-66_a*
- SECURITY-10, SECURITY-12*, SECURITY-25

* documentation changes

Updated CERT C Configuration

The CERT C Coding Standard built-in test configuration has been updated with new Parasoft rules according to the following rule mapping:

CERT ID	Parasoft ID
CERT-ARR39-C	CODSTA-189
CERT-CON33-C	SECURITY-25
CERT-DCL39-C	SECURITY-46
CERT-ENV31-C	BD-PB-INVENV
CERT-ENV33-C	SECURITY-48
CERT-ERR30-C	BD-PB-ERRNO
CERT-FIO42-C	BD-RES-LEAKS
CERT-FLP32-C	BD-API-VALPARAM
CERT-FLP36-C	MISRA-043_c, MISRA-043_d
CERT-FLP37-C	CODSTA-190
CERT-INT30-C	PB-66_a, BD-PB-INTOVERF
CERT-INT35-C	SECURITY-47
CERT-INT36-C	CODSTA-127_b
CERT-MSC30-C	SECURITY-02_b
CERT-MSC32-C	BD-SECURITY-RAND
CERT-POS30-C	CODSTA-144,CODSTA-145, BD-PB-OVERFNZT
CERT-POS33-C	SECURITY-10
CERT-POS34-C	BD-PB-PUTENV
CERT-POS35-C	BD-TRS-SYMLINK
CERT-POS36-C	SECURITY-44
CERT-POS37-C	SECURITY-45
CERT-POS38-C	BD-TRS-FORKFILE
CERT-POS39-C	BD-PB-BYTEORD
CERT-POS47-C	SECURITY-43
CERT-POS48-C	BD-TRS-DSTRLOCK, BD-TRS-REVLOCK
CERT-POS49-C	BD-TRS-BITLOCK

CERT-POS50-C	BD-TRS-ARG
CERT-POS51-C	BD-TRS-ORDER
CERT-POS52-C	BD-TRS-TSHL
CERT-POS53-C	GLOBAL-CONDMUTEXVAR
CERT-PRE31-C	CODSTA-150, CODSTA-150_b, CODSTA-150_c
CERT-STR31-C	BD-PB-ARRAY, BD-PB-OVERFWR, BD-SECURITY-BUFWRITE, BD-SECURITY-OVERFWR, SECURITY-12
CERT-STR34-C	CODSTA-187_a, CODSTA-187_b, CODSTA-187_c
CERT-STR38-C	CODSTA-188

Fixed PRs and FRs

FR/PR ID	Description
CPP-36398	Problem with matching function definition with its declaration when using Eigen library
CPP-36530	/STACK option being ignored by C++test (VS 2010)
CPP-36850	Add support for C++11 inarmcc5.06
CPP-36861	Support for Renesas RX 2.05 C/C++ compiler (renrx_2_5)
CPP-36892	The rule MISRA-107_b reports false positives when pointer is assigned and checked in the same line
CPP-36896	Support for GNU GCC 6 (native/host-based compiler)
CPP-36965	CODSTA-CPP-53 false positive, "Declare local variable 'x' as const"
CPP-36966	OOP-01 (Sutter Rule 54) false positive: "Base class copy constructor should be protected or public with smart pointer as parameter"
CPP-36967	CODSTA-CPP-28 (Sutter Rule 27) false positive: "When binary arithmetic operators are defined, assignment versions should be provided too"
CPP-37009	Rule INIT-06 reports violation even if member is initialized directly in class [C++11]
CPP-37011	The rule MISRA2004-6_3_b (MISRA2012-DIR-4_6_b, JSF-209_b) does not report violation when implicit signed/unsigned type is used
CPP-37018	The rule CODSTA-CPP-04 should not report violations on move constructors [C++11]
CPP-37019	Segmentation fault when using cwc.bin
CPP-37021	The rule MISRA2004-16_7 (MISRA2012-RULE-8_13_a) reports false positives
CPP-37022	QA: Suppression comment is garbled when the multibyte character is used to suppression comment
CPP-37023	[VS2015] Parse problem with _Buffer_descriptor() restrict(amp,cpu)
CPP-37782	Source paths not correctly scanned inBDFfile

* Parasoft Concerto (4.9.4 and later) is now deprecated. You can still connect to the Project Center and Team Server modules, but future releases of C++test will replaceConterto-related workflows with DTP workflows.