Built-in Test Configurations

The following tables include the test configurations shipped in the [INSTALL]/configs/builtin directory.

Static Analysis

This group includes universal static analysis test configurations. See Compliance Packs for test configurations that enforce coding standards.

Built-in Test Configuration	Description
Effective C++	Checks rules from Scott Meyers' "Effective C++" book. These rules check the efficiency of C++ programs.
Effective STL	Checks rules from Scott Meyers' "Effective STL" book.
Find Duplicated Code	Applies static code analysis rules that report duplicate code. Duplicate code may indicate poor application design and lead to maintainability issues.
Find Unused Code	Includes rules for identifying unused/dead code.
Flow Analysis Standard	Detects complex runtime errors without requiring test cases or application execution. Defects detected include using uninitialized or invalid memory, null pointer dereferencing, array and buffer overflows, division by zero, memory and resource leaks, and dead code. This requires a special Flow Analysis license option.
Flow Analysis Aggressive	Includes rules for deep flow analysis of code. A significant amount of time may be required to run this configuration.
Flow Analysis Fast	Includes rules for shallow depth of flow analysis, which limits the number of potentially acceptable defects from being reported.
Global Analysis	Checks the Global Static Analysis rules.
Metrics	Computes values for several code metrics.
Modern C++ (11, 14 and 17)	Checks rules that enforce best practices for modern C++ standards (C++11, C++14, C++17).
Recommended Rules	The default configuration of recommended rules. Covers most Severity 1 and Severity 2 rules. Includes rules in the Flow Analysis Fast configuration.
Sutter-Alexandrescu	Checks rules based on the book "C++ Coding Standards," by Herb Sutter and Andrei Alexandrescu.
The Power of Ten	Checks rules based on Gerard J. Holzmann's article "The Power of Ten - Rules for Developing Safety Critical Code." http://spinroot.com/gerard/pdf/Power_of_Ten.pdf

Compliance Packs

Compliance Packs include test configurations tailored for particular compliance domains to help you enforce industry-specific compliance standards and practices. See Compliance Packs Rule Mapping for information how the standards are mapped to C/C++test's rules.

Displaying compliance results on DTP

Some test configurations in this category have a corresponding "Compliance" extension on DTP, which allows you to view your security compliance status, generate compliance reports, and monitor the progress towards your security compliance goals. These test configurations require dedicated license features to be activated. Contact Parasoft Support for more details on Compliance Packs licensing.

See the "Extensions for DTP" section in the DTP documentation for the list of available extensions, requirements, and usage.

Aerospace Pack

Built-in Test Configuration	Description
Joint Strike Fighter	Checks rules that enforce the Joint Strike Fighter (JSF) program coding standards.

Automotive Pack

Built-in Test Configuration	Description
AUTOSAR C++14 Coding Guidelines	Checks rules that enforce the AUTOSAR C++ Coding Guidelines (Adaptive Platform, version 19.03). This test configuration is part of Parasoft Compliance Pack solution that allows you to monitor compliance with industry standards using the "Compliance" extensions on DTP. It requires dedicated license features to be activated. Contact your Parasoft representative for details.
HIS Source Code Metrics	Checks metrics required by the Herstellerinitiative Software (HIS) group.
High Integrity C++	Checks rules that enforce the High Integrity C++ Coding Standard.
MISRA C 1998	Checks rules that enforce the MISRA C coding standards
MISRA C 2004	Checks rules that enforce the MISRA C 2004 coding standard.
MISRA C 2012	Checks rules that enforce the MISRA C 2012 coding standard. This test configuration is part of Parasoft Compliance Pack solution that allows you to monitor compliance with industry standards using the "Compliance" extensions on DTP. It requires dedicated license features to be activated. Contact your Parasoft representative for details.
MISRA C++ 2008	Checks rules that enforce the MISRA C++ 2008 coding standards.

Medical Devices Pack

Built-in Test Configuration	Description
Recommended Rules for FDA (C)	Checks rules recommended for complying with the FDA General Principles for Software Validation (test configuration for the C language).
Recommended Rules for FDA (C++)	Checks rules recommended for complying with the FDA General Principles for Software Validation (test configuration for the C++ language).

Security Pack

Built-in Test Configuration	Description
CWE Top 25 2019	Includes rules that find issues classified as Top 25 Most Dangerous Programming Errors of the CWE standard. This test configuration is part of Parasoft Compliance Pack solution that allows you to monitor compliance with industry standards using the "Compliance" extensions on DTP.
CWE Top 25 2019 + On the Cusp	Includes rules that find issues classified as Top 25 Most Dangerous Programming Errors of the CWE standard or included on the CWE Weaknesses On the Cusp list. This test configuration is part of Parasoft Compliance Pack solution that allows you to monitor compliance with industry standards using the "Compliance" extensions on DTP.
OWASP Top 10 2017	Includes rules that find issues identified in OWASP's Top 10 standard This test configuration is part of Parasoft Compliance Pack solution that allows you to monitor compliance with industry standards using the "Compliance" extensions on DTP. It requires dedicated license features to be activated. Contact your Parasoft representative for details.
Payment Card Industry Data Security Standard	Includes rules that find issues identified in PCI Data Security Standard
SEI CERT C Guidelines	Checks rules and recommendations for the SEI CERT C Coding Standard. This standard provides guidelines for secure coding. The goal is to facilitate the development of safe, reliable, and secure systems by, for example, eliminating undefined behaviors that can lead to undefined program behaviors and exploitable vulnerabilities.

SEI CERT C Rules	Checks rules for the SEI CERT C Coding Standard. This standard provides guidelines for secure coding. The goal is to facilitate the development of safe, reliable, and secure systems by, for example, eliminating undefined behaviors that can lead to undefined program behaviors and exploitable vulnerabilities. This test configuration is part of Parasoft Compliance Pack solution that allows you to monitor compliance with industry standards using the "Compliance" extensions on DTP. It requires dedicated license features to be activated. Contact your Parasoft representative for details.
SEI CERT C++ Rules	Checks rules for the SEI CERT C++ Coding Standard. This standard provides guidelines for secure coding. The goal is to facilitate the development of safe, reliable, and secure systems by, for example, eliminating undefined behaviors that can lead to undefined program behaviors and exploitable vulnerabilities. This test configuration is part of Parasoft Compliance Pack solution that allows you to monitor compliance with industry standards using the "Compliance" extensions on DTP. It requires dedicated license features to be activated. Contact your Parasoft representative for details.
Security Rules	General test configuration that finds security issues
UL 2900	Includes rules that find issues identified in the UL-2900 standard.

Runtime Analysis

Built-in Test Configuration	Description
Coverage	Generates the code coverage report.
Unit Testing	Analyzes CppUnit or CppUTest test results collected with C/C++test's connector (see Integrating with CppUnit and CppUtest)

Compliance Packs Rule Mapping

This section includes rule mapping for the CWE standard. The mapping information for other standards is available in the PDF rule mapping files shipped with Compliance Packs.

CWE Top 25 Mapping

CWE ID	CWE Name	Parasoft rule ID(s)
CWE-119	Improper Restriction of Operations within the Bounds of a Memory Buffer	• CWE-119-a • CWE-119-b • CWE-119-c • CWE-119-d • CWE-119-e • CWE-119-f • CWE-119-g • CWE-119-h • CWE-119-i • CWE-119-j
CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	N/A
CWE-20	Improper Input Validation	• CWE-20-a • CWE-20-b • CWE-20-c • CWE-20-d • CWE-20-e • CWE-20-f • CWE-20-g • CWE-20-h • CWE-20-i • CWE-20-j
CWE-200	Information Exposure	• CWE-200-a

CWE-125	Out-of-bounds Read	• CWE-125-a • CWE-125-b • CWE-125-c • CWE-125-d
CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	• CWE-89-a
CWE-416	Use After Free	• CWE-416-a • CWE-416-b • CWE-416-c
CWE-190	Integer Overflow or Wraparound	• CWE-190-a • CWE-190-b • CWE-190-c • CWE-190-d • CWE-190-e • CWE-190-f • CWE-190-g
CWE-352	Cross-Site Request Forgery (CSRF)	N/A
CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	• CWE-22-a
CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	• CWE-78-a
CWE-787	Out-of-bounds Write	• CWE-787-a • CWE-787-b • CWE-787-c • CWE-787-d • CWE-787-e • CWE-787-f
CWE-287	Improper Authentication	• CWE-287-a
CWE-476	NULL Pointer Dereference	• CWE-476-a • CWE-476-b
CWE-732	Incorrect Permission Assignment for Critical Resource	• CWE-732-a • CWE-732-b
CWE-434	Unrestricted Upload of File with Dangerous Type	N/A
CWE-611	Improper Restriction of XML External Entity Reference	• CWE-611-a
CWE-94	Improper Control of Generation of Code ('Code Injection')	N/A
CWE-798	Use of Hard-coded Credentials	• CWE-798-a
CWE-400	Uncontrolled Resource Consumption	• CWE-400-a
CWE-772	Missing Release of Resource after Effective Lifetime	• CWE-772-a • CWE-772-b

CWE-426	Untrusted Search Path	• CWE-426-a
CWE-502	Deserialization of Untrusted Data	N/A
CWE-269	Improper Privilege Management	• CWE-269-a • CWE-269-b
CWE-295	Improper Certificate Validation	N/A

CWE Weaknesses On the Cusp Mapping

CWE ID	CWE Name	Parasoft rule ID(s)
CWE-835	Loop with Unreachable Exit Condition ('Infinite Loop')	• CWE-835-a
CWE-522	Insufficiently Protected Credentials	N/A
CWE-704	Incorrect Type Conversion or Cast	• CWE-704-a • CWE-704-b • CWE-704-c • CWE-704-d • CWE-704-e • CWE-704-f • CWE-704-g • CWE-704-h • CWE-704-i • CWE-704-j • CWE-704-k • CWE-704-l
CWE-362	Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')	• CWE-362-a • CWE-362-b • CWE-362-c • CWE-362-d • CWE-362-e
CWE-918	Server-Side Request Forgery (SSRF)	N/A
CWE-415	Double Free	• CWE-415-a
CWE-601	URL Redirection to Untrusted Site ('Open Redirect')	N/A
CWE-863	Incorrect Authorization	• CWE-863-a
CWE-862	Missing Authorization	N/A
CWE-532	Inclusion of Sensitive Information in Log Files	• CWE-532-a
CWE-306	Missing Authentication for Critical Function	N/A
CWE-384	Session Fixation	N/A
CWE-326	Inadequate Encryption Strength	• CWE-326-a
CWE-770	Allocation of Resources Without Limits or Throttling	• CWE-770-a

CWE-617	Reachable Assertion	• CWE-617-a