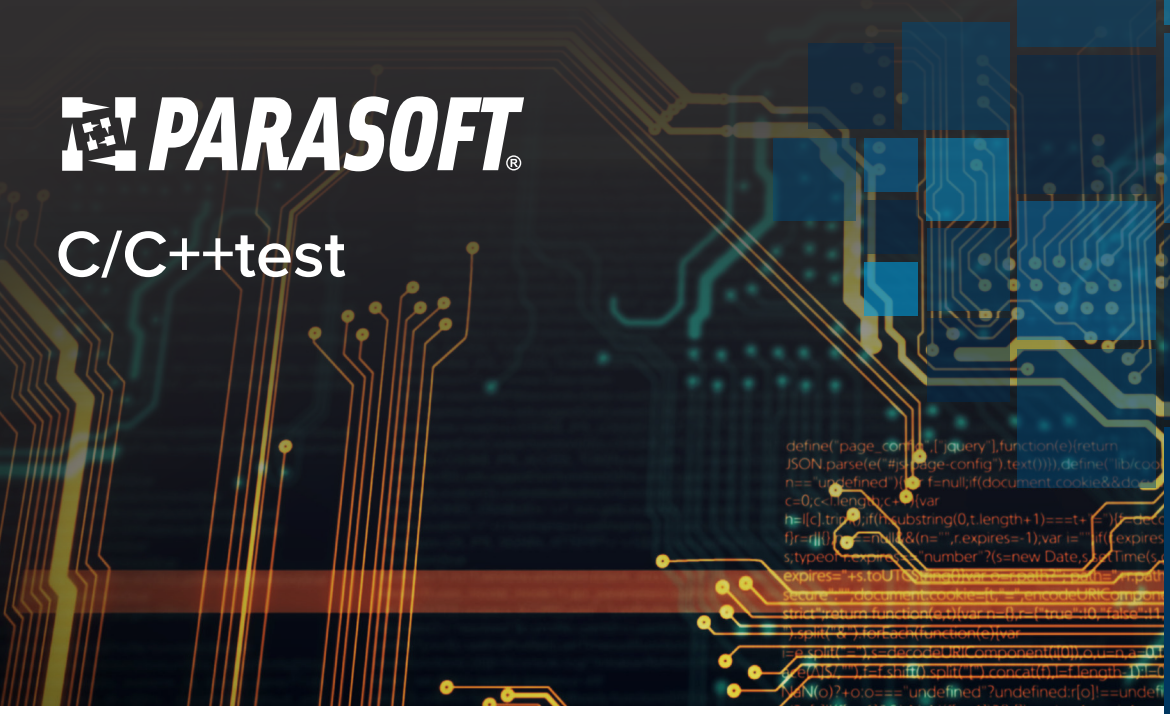
PARASOFT®

C/C++test



SAVE TIME, INNOVATE MORE, AND DELIVER WITH CONFIDENCE

Parasoft C/C++test is an integrated development testing solution for C and C++ that enables developers to quickly analyze their code for critical defects or security vulnerabilities and perform efficient unit testing. C/C++test helps organizations reduce risk, cut costs, increase productivity, and achieve industry compliance goals by automating a critical set of software testing needs. C/C++test can be used in both host-based and target-based code analysis and test flows, critical for embedded and cross-platform development.

» ELIMINATE TEST OVERHEAD

Apply a comprehensive set of best-practices that reduce testing time, testing effort, and the number of defects that reach QA.

» IMPROVE CODE QUALITY AND SECURITY

Efficiently construct, continuously execute, and effectively maintain a comprehensive regression test suite that detects whether updates break existing functionality.

» ELIMINATE RECALLS

Automate negative testing on a broad range of potential user paths to uncover problems that might otherwise surface only in “real-world” usage.

» DO MORE WITH LESS

Automatically vet known coding issues, so more time can be dedicated to tasks that require human intelligence.

» IMPROVE CODE COVERAGE

Access on-demand objective code assessments and track progress toward quality and schedule targets.

“By deploying C/C++test as the coding standard analysis tool, Mobile solution project in the SW Center of Samsung Electronics has decreased the amount of coding violations by 80%; a significant improvement on their development/testing process.”

SAMSUNG

IDENTIFY RUNTIME ERRORS WITHOUT EXECUTING SOFTWARE

Integration-time static analysis simulates feasible application execution paths and determines whether these paths could trigger specific categories of runtime errors, including:

- Using uninitialized or invalid memory
- Null pointer dereferencing
- Array and buffer overflows
- Division by zero
- Memory and resource leaks
- Various flavors of dead code

C/C++test greatly simplifies defect analysis by providing a complete highlighted path for each potential defect in the developer's IDE. Automatic cross-links to code help users quickly jump to any point in the highlighted analysis path.

PARASOFT
C/C++test

STATIC CODE ANALYSIS

Reduce risk at each stage of development:

- » Integration-time analysis
- » Continuous analysis
- » Edit-time static analysis
- » Runtime static analysis

Graphical rule editor for creating custom coding rules

UNIT TESTING

Automated generation and execution of unit & component level tests

Flexible stub framework

Full support for regression testing

RUNTIME ERROR DETECTION

Expose hard-to-find errors, such as memory leaks, null pointers, uninitialized memory, and buffer overflows

Increase test-result accuracy through execution in a real target environment

COVERAGE ANALYSIS

Code coverage analysis with code highlighting

HTML, PDF, and custom format reports

MONITOR AND ELIMINATE RUN-TIME ERRORS

Runtime error detection constantly monitors for coding issues, and provides results immediately after the test session is finished. The reported problems are presented in the developer's IDE along with details about how to fix the errors (including memory block size, array index, allocation/deallocation stack trace, etc.). Coverage metrics are collected during application execution. These can be used to see what part of the application was tested and to fine tune the set of regression unit tests (complementary to functional testing).

AUTOMATE CODE ANALYSIS FOR MONITORING COMPLIANCE

C/C++test automates the enforcement of your policy by analyzing code and reporting errors directly in developers' IDE when code deviates from the standards prescribed in your programming policy. Hundreds of built-in rules—including implementations of MISRA, MISRA C++, FDA, Scott Meyers' Effective C++, Effective STL, and other established sources—help identify bugs, highlight undefined or unspecified C/C++ language usage, enforce best practices, and improve code maintainability and reusability. Development managers can use the built-in rules or create rules and configurations specific to their group or organization. For safety- and quality-critical applications, such as avionics, medical, automotive, transportation, and industrial automation, C/C++test enables efficient and auditable quality processes with complete visibility into compliance efforts.

UNIT AND INTEGRATION TESTING WITH COVERAGE ANALYSIS

C/C++test automatically generates complete tests, including test drivers and test cases for individual functions, purely in C or C++ code in a CppUnit-like format. Auto-generated tests, with or without modifications, are used for initial validation of the functional behavior of the code. By using corner case conditions, the test cases also check function responses to unexpected inputs, exposing potential reliability problems. Specific GUI widgets simplify test creation and management and a graphical Test Case Wizard enables developers to rapidly create black-box functional tests for selected functions without having to worry about their inner workings or embedded data dependencies. A Data Source Wizard helps parameterize test cases and stubs, enabling increased test scope and coverage with minimal effort; the Stub View allows users to create stubs for any functions not available in the test scope; and the Test Case Explorer centralizes data to provide a clear pass/fail status.

AUTOMATED REGRESSION TESTING

C/C++test facilitates the development of robust regression test suites that detect if incremental code changes break existing functionality. With a large legacy code base, a small piece of just-completed code, or anything in between, C/C++test can generate tests that capture the existing software behavior, via test assertions produced by automatically recording the runtime test results. As the code base evolves, C/C++test reruns these tests and compares the current results with those from the originally captured "golden set." It can easily be configured to use different execution settings, test cases, and stubs to support testing in different contexts. This type of regression testing is especially critical for supporting agile development and short release cycles, and ensures the continued functionality of constantly-evolving and difficult-to-test applications.

TEST ON THE HOST, SIMULATOR, AND TARGET

C/C++test automates the complete test execution flow, including test case generation, cross-compilation, deployment, execution, and loading results (including coverage metrics) back into the GUI. Testing can be driven interactively from the GUI or from the command line for automated test execution or batch regression testing. In the interactive mode, users can run tests individually or in selected groups for easy debugging or validation. In addition to using the built-in test automation, users can incorporate custom test scripts and shell commands to fit the tool into their specific build and test environment. A customizable workflow allows users to test code as it's developed, then use the same tests to validate functionality/reliability in target environments. All test artifacts of C/C++test are source code, and therefore completely portable.

Supported Host Platforms
Windows
Linux
Solaris UltraSPARC

Supported Tool Chains / Environments
ARM
Eclipse IDE for C/C++ Developers
GreenHills
IAR
Kiel
Microsoft
QNX
Renasas
Texas Instruments
WindRiver

Build Management
GNU make
Sun make
Microsoft nmake
ElectricAccelerator

Continuous Integration
Hudson
Jenkins
Electric Accelerator

Source Control
AccuRev SCM
Borland StarTeam
CVS
Git
IBM Rational ClearCase
IBM Rational Synergy
Mercurial
Microsoft Team Foundation Server
Microsoft Visual SourceSafe
Perforce SCM
Serena Dimensions
Subversion (SVN)

Coverage Metric Generation
Function
Call
Line
Statement
Block
Path
Decision
Simple Condition
MCDC

Parasoft Corporation www.parasoft.com | Sales: 1-888-305-0041 | International Sales: +1-626-256-3680