

Develop Compliant DO-178C Software for Airborne Systems

Create, execute, and maintain tests with Parasoft's unified, fully integrated, and automated testing solution for C/C++ software development.



STREAMLINE DO-178C COMPLIANCE CHALLENGES

Software compliance for airborne systems starts with software engineering standards and best practices, complemented by a TÜV certified tool for verification and validation of DO-178B/C objectives that help ensure FAA and EASA certification. Parasoft C/C++test offers broad solutions for avionics software including:

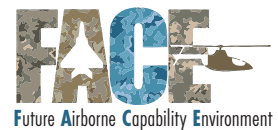
- » Requirements traceability
- » Unit, integration, system testing
- » Coding standards compliance
- » On-target hardware testing
- » Structural code coverage
- » Object code verification
- » Tool qualification kit
- » Compliance management

ACCELERATE FEEDBACK LOOPS & DEFECT REMEDIATION

Developers can run static analysis in their IDE with Parasoft C/C++test and have the ability to create, execute, and review unit testing results along with corresponding code coverage. This accelerates the feedback loop to development and enables faster remediation of defects, preventing a rise in program costs associated with the discovery of late stage defects.

GET OUT-OF-THE-BOX QUALITY STANDARD SUPPORT

The C/C++ static analysis powered by Parasoft offers out-of-the-box support for quality standards (Effective C++, Effective STL, Modern C++, Sutter-Alexandrescu, the Power of Ten, and so on), security standards (CWE Top 15, CWE Cusp, DISA-ASD-STIG, OWASP top 10, SEI CERT C, SEI CERT C++, UL 2900, and more), and includes support for industry standard (such as MISRA C, MISRA C++, High Integrity C++, AUTOSAR, JSF, Recommended Rules for FDA C & C++), often used for safety-critical programs.

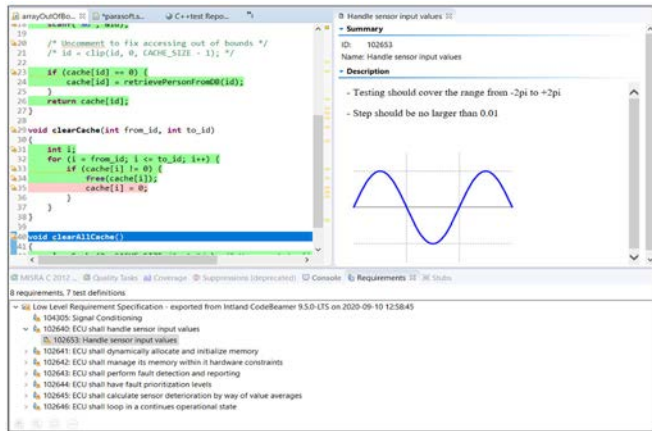


TRY PARASOFT C/C++TEST

[Schedule a demo](#) to learn how your team can streamline DO-178C compliance challenges to deliver compliant safety-critical software at speed.

PARASOFT'S SOLUTIONS FOR AIRBORNE SYSTEMS

Requirements traceability. Bidirectional traceability with ALM solutions like Jama Connect, codebeamer, Polarion, and Jira.



Access ALM requirements from IDE, and build tests with traceability to requirements and code.

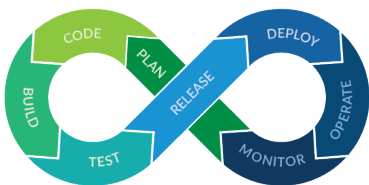


Automated code analysis. The policy-based static code analysis ensures that industry coding standards and development policies are being followed. Users can define custom rule sets or use the built-in rules to identify potential coding errors and policy violations.

Flow analysis. Simulate possible execution paths to determine whether these paths could trigger specific categories of runtime errors. This is especially useful for systems in which detailed runtime analysis is ineffective or impossible, such as embedded systems.

Unit, integration, and system testing with coverage analysis. Automatically run unit tests generated by open frameworks and report results, including assembly coverage data. You can also generate complete unit tests, including test drivers and test cases for individual functions, to validate functional behavior. Automatically generated test cases also check function responses to unexpected inputs.

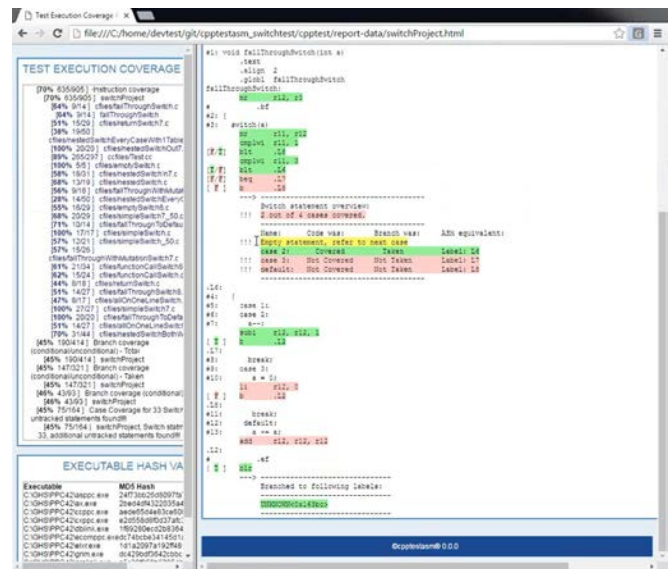
DevOps/DevSecOps. CI/CD automation, reliability, compatibility, and speed.



Reporting dashboard. Generate compliance reporting, traceability of all project artifacts, requirements, defects/enhancements, and tasks with Parasoft DTP. Enforce policies related to compliance and process standardization. This development testing platform identifies inefficiencies by tracking and analyzing software development metrics and progress, so development teams can build on the increased productivity.

Configurable reports. Generate HTML, PDF, and custom format reports that cover which files were tested and analyzed in addition to providing test results and code coverage. Automatically email reports using role-based filters, distribute data directly to developers responsible for the code, as well as send summary reports to managers and team leads.

Code review. DTP Change Explorer enables convenient analyses of source code deltas between specific milestones or points in development. Overlapping code delta information with static analysis or unit testing results elevates the traditional code review process to completely new level.



Assembly language code coverage shown in the Parasoft ASMTool.

Tool qualification. Reduce the potential for human error and the time it takes to perform tool qualification as required by DO-178B/C and DO-330. Automate the process of creating the supporting documentation required for tool qualification of static analysis, unit testing, and coverage requirements with Parasoft's Qualification Kit.

