

# Assembly Coverage Tool

Parasoft provides code instrumentation to capture and report structural code coverage for statement, branch, modified condition decision coverage (MC/DC). For the most stringent safety-critical code coverage requirements, such as DO-178C Level A, Parasoft offers code coverage at the assembly level.

Parasoft's Assembly Coverage Tool (ASMTTool) satisfies the executable object code recommendations set out by DO-178B/C. With little effort, ASMTTool generates structural coverage reports from executable object code that are easy to review for insight into compiler generated code that is not directly traceable to source code statements.

ASMTTool also supports collecting structural coverage from both the software integration process (target hardware testing) and desktop development environment (Parasoft C/C++test for unit testing).

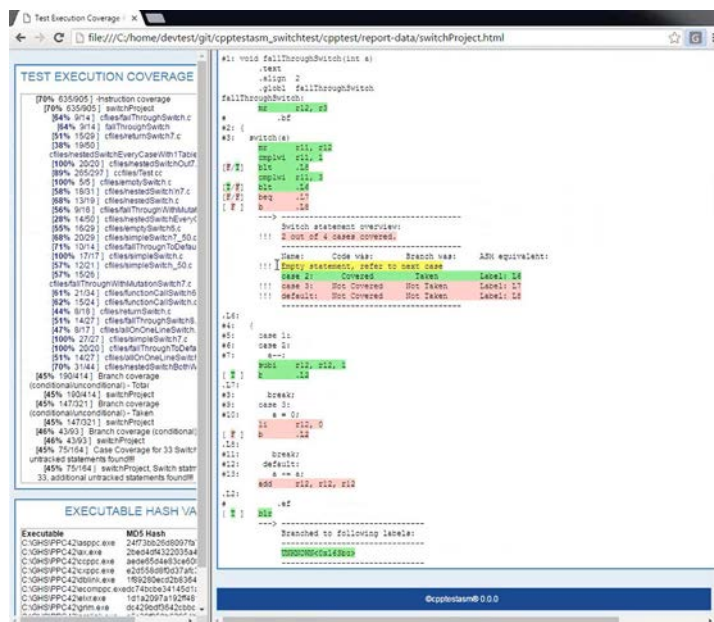
## SUPPORTED ENVIRONMENTS

## HOST PLATFORMS

- » Windows x86\_64
- » Linux x86\_64

## TOOLCHAINS

- » Green Hills Software compiler (version 4.x) for 32-bit PowerPC
- » Green Hills Software compiler (version 2017) for 32-bit PowerPC
- » Green Hills Software compiler (version 2017) for 64-bit PowerPC
- » GCC for x86\_64 (version 4.6+)
- » GCC for x86\_32 (version 4.6+)



## Parasoft ASMTTool Assembly Language Code Coverage

## TRY PARASOFT ASMT TOOL

[Schedule a demo](#) to learn how your team can satisfy object code verification for DO-178C with Parasoft's ASMTTool.

## DELIVER COMPLIANT DO-178C LEVEL A SAFETY-CRITICAL SOFTWARE

iTeams use ASMTTool to accomplish compliance goals of DO-178C software level A and B recommendations with respect to executable object code (6.4.[c,d,e] and 6.4.4.2.b).

### COVERAGE METRICS

Coverage data is stored in the tested program's memory buffers. When the scheduled tests are executed and the test executable exits, the collected coverage information is saved via the defined communication channel. Typically, directly to the file. Reporting of structural coverage includes:

- » Assembly lines executed
- » Conditional branch decisions
- » Indirect branch target addresses

The ASMTTool maps local scoped assembly labels to the lines of original source code. The resulting table data is used to simplify verification of compiler generated code sequences. Assembly coverage report is made available in html (default), xml, txt, flattxt, and csv formats.

Additional Reports				Back to Top
Test Execution Context	Assembly Coverage	Test Execution Details	Overall Status	
/stub_configuration_dwa	View Report	View Report	PASSED	

Assembly Coverage Report Table

### COMPILER CONFIDENCE

ASMTTool reporting allows easy confirmation that **testing has exercised the data and control coupling between code components**. Shared and relocatable source code structural coverage are aggregated from all invocations within scope.

### CORRELATION WITH ORIGINAL SOURCE CODE

Object level instrumentation and source code correlation are processed separately. Structural coverage reports contain data for only the original product and filter out code produced by the compiler for interleaving and debugging.

### INSTRUMENT HANDWRITTEN ASSEMBLY

Complete support for the analysis of source files written in pure assembly. Structural coverage reporting on a mix of source code languages requires no additional effort.

### FLEXIBLE TESTING

Collect cumulative and reusable results from both target hardware integration testing and desktop development unit testing within Parasoft C++test.

### ACCUMULATES COVERAGE AS SOFTWARE GROWS

ASMTTool's coverage model is additive. As library code is adopted, its structural coverage data can be collected from all downstream uses and merged to produce aggregate reports. Reducing the effort involved in producing structural coverage.

### INTEGRATE WITH EXISTING TOOLCHAIN

Easily integrates with most build systems to produce coverage reports from existing build targets.

