

Testing C Code with Polyspace Test

OEM: MathWorks • Code: MW-TESTING-C-CODE-WITH-POLYSPACE-TEST

COURSE DESCRIPTION

This two-day, hands-on course discusses the use of Polyspace Test™ to dynamically verify C software in host and target environments against expected behaviors and code coverage objectives. Authoring and executing simple and complex graphical tests Measuring code coverage of a set of test cases and generating test cases for missing code coverage Using stubs and mocks to substitute for function calls in code under test Authoring, building, and running test suites and test cases using the C xUnit API Registering a provided target support package and running test cases on an external target Generating reports from Polyspace Test results

AUDIENCE PROFILE

- IT professionals and technical staff seeking hands-on training in this course's subject area.

PREREQUISITES

- Strong knowledge of C or C++

COURSE OBJECTIVES

By the end of this course, participants will be able to:

- Become familiar with the Polyspace Platform environment and author and execute simple graphical tests.
- Author graphical tests for code that contains various data formats and scopes.
- Author tests with multiple function calls and perform parameter sweep tests.
- Run tests with instrumentation for various levels of code coverage, execution time profiling, and stack profiling.
- Add stub functions for callees without an implementation, and write mocks for callees that need to be replaced in testing.
- Become familiar with simple tests in the xUnit C API, and compile and run tests directly with an external toolchain.
- Author tests using the xUnit C API that are organized into suites, parameter sweep tests, and tests that assess data of custom types.
- Cross-compile and run existing test cases on an external target and profile for performance and code coverage.

COURSE OUTLINE

Module 1: Day 1 of 2

Authoring Graphical Tests

- Creating a Polyspace Platform project
- Configuring the test environment
- Authoring simple graphical tests
- Debugging tests
- Understanding generated artifacts
- Generating test reports
- Defining test cases outside a Polyspace Platform project
- Linking test cases to requirements
- Automating test execution from the command line

Handling Complex Data

- Assessing floating-point values
- Initializing and assessing global variables
- Initializing and assessing pointer targets
- Initializing and assessing structures
- Initializing and assessing arrays

Authoring Complex Tests

- Authoring multi-step tests
- Authoring parametrized tests
- Authoring scripted tests

Profiling, Coverage, and Test Generation

- Measuring code coverage
- Generating tests for missing code coverage
- Profiling execution time
- Profiling memory usage
- Automating code coverage and profiling on the command line

Module 2: Day 2 of 2

Using Stubs and Mocks

- Stubs and mocks overview
- Stubbing undefined global variables
- Stubbing undefined functions
- Mocking defined functions
- Using mock variables

Authoring Tests with xUnit C API

- xUnit C API overview

- Building, running, and reviewing xUnit-based tests
- Using assessment macros
- Exporting test results
- Importing xUnit tests into Polyspace projects
- Linking imported xUnit test cases to requirements

Authoring Complex Tests with xUnit C API

- Configuring test suites
- Authoring parametrized tests
- Comparing custom data type objects
- Measuring code coverage
- Measuring execution time and memory usage

Running Tests on Target

- On-target testing overview
- Registering targets
- Running tests on target
- Profiling on target