

JavaTest™ Harness

A General-purpose Test Platform

Highlights

- Configures, sequences, and executes tests
- Provides a rich and highly configurable environment for viewing test results
- Delivers an extensible environment for specifying test execution
- Executes tests through either a graphical user interface (GUI) or a command-line interface



The JavaTest™ harness is a general-purpose, full-featured, flexible, configurable test harness that is well suited for most types of unit testing. Originally developed as a test harness to run Technology Compatibility Kit (TCK) test suites, the JavaTest harness has since evolved into a general-purpose test platform upon which test suite developers can build product quality test suites that can run a wide variety of tests on a wide variety of platforms.

- The JavaTest harness is an excellent tool for configuring, sequencing, and running test suites that consist of large numbers of discrete, independent tests.
- The JavaTest harness user interface provides a convenient interface for configuring and running simple or extremely complex test suites.
- The JavaTest harness is not a test development tool, however, it is shipped with a library of classes that make it easier to write tests.

JavaTest harness features

Rich configuration experience

The JavaTest harness configuration editor enables a test suite architect to create wizard-based configuration interviews that hide much of the complexity of unbounded configuration from the tester. Using a configuration interview, the tester answers questions only about the platforms involved and is not required to know anything about the syntax or format required for specifying the information to the JavaTest harness. In addition, the configuration editor contains a help pane that provides examples and information about each question.

The configuration editor supports the use of configuration templates. These templates make it possible for a site administrator to fill in the configuration information common to a given company or site. This can greatly reduce the amount of configuration each user is responsible for.

Extensible GUI

Using the JavaTest harness, a test suite architect can customize the GUI to meet the needs of the test suite. For example, an architect can add a menu item to show the test suite's custom documentation or may add custom toolbars and menus. The test suite architect can also customize the preferences dialog to augment the standard preference options.

Extensible test execution engine

The test execution engine is a plug-in component that can be adapted to many different test styles. For example, plug-in components can:

- Execute tests that conform to any application programming interface (API)
- Execute tests written in most programming languages
- Annotate or modify the result of a test before reporting it to the harness
- Execute tests on constrained devices such as cell phones and PDAs. The Java ME TCK Framework is a good example of such a component.

Accommodates diverse test formats

The JavaTest harness has a public API that the test suite architect can use to identify discrete test entities and present them to the user. The architect can create unique test names and display the test results in a hierarchical tree.

Test suite size

The JavaTest harness is designed to configure, sequence, and run test suites that consist of many — 100,000 or more — discrete independent tests.

Sophisticated GUI

With the JavaTest harness, the entire testing process — from configuration to analysis and reporting — can be done without leaving the graphical interface. The graphical interface provides the means to set configuration values, select sets of tests to run, and monitor the progress of a test run. A high-level tree-based interface is provided to enable the user to "drill" down into the results to determine the cause of any problem.

Fully implemented command-line interface

The JavaTest harness command-line interface provides access to virtually all harness functionality. It enables users to run test suites from build scripts or other automated test programs. The command-line interface allows users to change parameters, conduct multiple test runs, monitor test run progress, and create reports.

Reports

The JavaTest harness can generate reports in a variety of formats, including the HyperText Markup Language (HTML) and plain text. Because the reporting system is extensible, test suite architects can use the API to generate custom reports. Reports from multiple test runs can also be merged into a single report. The merge feature allows users to distribute test suite execution over different machines and then produce a unified report for the entire test run.

View filtering of test results

JavaTest harness users can choose and create view filters that change the way test results are displayed in the GUI. The user can choose an applicable filter when they configure a test run and a different filter while they watch the test run or analyze the results.

Extensive online help

The JavaTest harness has an extensive online help system that helps users learn to use the harness. The online help includes context-sensitive help, Help buttons in dialogs, a printable guide, and a command-line reference.

Requirements

The JavaTest harness runs on the Java™ Platform, Standard Edition (Java SE) version 5 or higher.

Learn More

Get the inside story on the trends and technologies shaping the future of computing by signing up for the Sun Inner Circle program. You'll receive a monthly newsletter packed with information, plus access to a wealth of resources. Register today at sun.com/joinic.

Learn more

To learn more about the JavaTest harness, visit java.sun.com/javame/javatest.

About Sun

A singular vision, "The Network is the Computer", drives Sun in delivering industry-leading technologies that focus on the whole system — where hardware, software, and services combine. With a proven history of sharing, building communities, and innovation, Sun creates opportunities, both social and economic, around the world.