

# Source Layout

## JT Harness Source Code Description

The JT harness source code is divided into areas based on their responsibility or whether they are general versus or specific. The core of the harness is in one package, the generic GUI framework in another, while sub-packages contain more specific GUI subsystems that plug into the generic framework. An implication of this is that there are clearly defined interfaces between packages because the Java™ programming language does not have a "friend" mechanism.

### Packages

Located in the `src/` directory under the root of the Mercurial project tree:

#### `com.sun.javatest`

The core of the JT harness. Many base classes and interfaces are defined here, for example: `TestFinder`, `Script`, and `TestEnvironment`. Most sub-packages reference classes from this package. In order to prevent the initialization of those libraries when running in CLI mode, it is important that no class in this package reference AWT or Swing.

#### `com.sun.javatest.util`

The core of the JT harness. Many base classes and interfaces are defined here, for example: `TestFinder`, `Script`, and `TestEnvironment`. Most sub-packages reference classes from this package. In order to prevent the initialization of those libraries when running in CLI mode, it is important that no class in this package reference AWT or Swing. `com.sun.javatest.util`

#### `com.sun.javatest.lib`

Contains concrete implementations of plug-in classes (`TestFinder`, `Script`, `Test`). Do not add GUI code to this package.

#### `com.sun.javatest.finder`

Similar to `com.sun.javatest.lib`, but for `TestFinder` classes. Contains many implementations and most test suites use these either directly or as a sub-class. Do not add GUI code to this package.

#### `com.sun.javatest.report`

Contains all implementations of the reporting code, both HTML and plain text. Future report types may be located here or in a sub-package.

#### `com.sun.javatest.tool`

This is the core GUI framework for the JT harness GUI interface. The GUI interface is defined by the `Desktop` at the top level, with `Tools` below that. The GUI in which you execute tests is a `Tool` whose implementation is not in this package. This package controls things such as menu organization, top-level window arrangement (desktop style), and the preference dialog. This package also contains the entry point for the JT harness: `com.sun.javatest.tool.Main`.

#### `com.sun.javatest.exec`

The "Exec Tool" is also known as the "Test Manager." This is the GUI used for test execution functionality. Code for the Configuration Editor is also contained here. This is what most people think of as the core the JT harness GUI functionality.

#### `com.sun.javatest.exec.template`

Support classes for GUI template handling.

#### `com.sun.javatest.batch`

Command-line interface support classes.

#### `com.sun.interview`

Generic interview API.

#### `com.sun.interview.wizard`

GUI rendering components for the interview question types.

#### `com.sun.javatest.interview`

Concrete implementations of the interview classes. Also contains the base class that all JT harness interviews must implement: `com.sun.javatest.interview.BasicInterviewParameters`.

#### `com.sun.javatest.agent`

The classes for the standard JT harness agent used for remote execution and result reporting. Used by the JCK and CDC TCKs.

#### `com.sun.javatest.servlets`

A few utility servlet classes provided for use with the JT harness. Not much work occurs here.

`com.sun.javatest.audit`

Tool for auditing test results. This class is distributed but not currently in active use or development.