JT Harness

JT Harness

The JT harness is based on Oracle's [JavaTest harness](). The JT harness is a general purpose, fully-featured, flexible, and configurable test harness very well suited for most types of unit testing. Originally developed as a test harness to run TCK test suites, it has since evolved into a general purpose test platform.

The JT harness:

- Is designed to configure, sequence, and run test suites that consist of many (100,000 or more) discrete, independent tests. It is especially good at testing APIs and compilers.
- Can be used to run tests on all of the Java platforms, from the Java Card platform, to the Java Platform, Enterprise Edition ("Java EE").
- Enables you to create test suites that are self-contained products that customers can easily configure and run.

For more information read the [JT harness whitepaper (PDF), and run the JT Harness Tutorial]().

JT harness technology is being opened in order to develop a community that will improve it, further its development, and use it to develop test suites. We encourage you to browse, download, contribute, and get involved.

News and Announcements

JTHarness 5.0 Release (May 25, 2017)

The JT Harness 5.0 is a milestone release. Release 5.0 adapts the harness for new JDK 9 changes and other features. JT Harness 5.0 changes include:

- a rewrite of the JavaTest agent
- removal of JavaHelp
- numerous bug fixes

We encourage you to browse, download, contribute, and get involved.

JTHarness 4.6 Release (October 4, 2014)
The JT harness 4.6 release is a minor release. Release 4.6 addresses the following issues:

- Support for shared agent class loader
- minor documentation updates
- [minor bug fixes]()

The JTHarness project moved to OpenJDK (March 4, 2014)

- **Source moved to OpenJDK Mercurial (Hg)!**
- [http://hg.openjdk.java.net/code-tools/jtharness]()

JTHarness 4.5 Release (March 4, 2014)
The JT harness 4.5 release is a minor release. Release 4.5 addresses the following issues:

- KFL report links are always absolute, should sometimes be relative like rest of report
- Documentation added for KFL features and plain text report
- Agent not able to handle new high concurrency limit (256)
- Remove Audit Tool from deliverables
- Max concurrency should be more than 50, now 256
- Breakdown of how execution time was spent now printed on console
- Not executed test progress stats seem to print in hex on console in Russian locale
- Agent monitor accepts negative timeout values
- NPE if exclude list file in JTI does not exist
- https links do not work in internal view, redirect to external browser
- KFL pane needed in Quick Set Configuration Editor
- NPE if report is run on command line without workdir
- Report browser doesn't refersh if file is the same file (but has changes)
- EditJTI -log option does not work
- Enhanced version info - class loader type, harness type
- Cancel button in Load Configuration can't be pressed by keyboard

Quick Links

Downloads (Provided by [Adopt OpenJDK]())

- [Tip]() (<dev>)
- [Released]()

For test suite developers:

- [Latest Release Notes 4.6]()
- [JavaTest 5.0 Harness Release Notes Addendum]()
- [JT Harness Tutorial]()
- [The JTharness whitepaper (PDF)]()
- [Additional documentation]()

For JT harness contributors:

- [Build Instructions]()
- [Repository Structure]()
- [JT Harness Developer's Guide]()
- [Source Code Description]()

Project documentation:

- [FAQ]()
- [Plans for upcoming releases]()
- [Related projects and links]()

Related projects

- [Jtreg]()
- [SigTest]()

- "Create Report" button may be disabled while there are some report types selected
- CLI documentation of -workdir subcommands missing
- Reports created on Windows with Russian locale contain ??? in date

The JTHarness project continues to expand its community, with JTHarness's introduction into the OpenJDK CodeTools project. We encourage you to browse, download, contribute, and get involved.

JTHarness 4.4.1 Maintenance Release (December 21, 2011)
The JT harness 4.4.1 release is a maintenance release. Release 4.4.1 fixes bugs and includes the following notable changes:

- Makes the "Report Converter" tool easier to use
- Improves the speed of loading the `ResultCache` file
- Shortens harness start-up time
- Allows known test execution ordering
- Adds new "Save view filter" settings
- Improves COF functionality

JT harness 4.4.1 is compatible with JT harness 4.4. For more information, see the release notes on the [d ownload page](#).

JTHarness 4.4 Feature Release (June 24, 2011)
The JT harness 4.4 release is a feature release. It fixes bugs and it contains the following notable changes:

- **Known failures lists** - The known failures list (KFL) allows you to specify one or more files that contain the names of tests that are known to result in a Failed status when executed. When you supply this list of tests to the harness, it can generate an HTML report that includes a section that can help you identify tests that report an unexpected result.
- **Case-sensitive exclude lists** - Historically the matching of test names in an exclude list is case-insensitive, although case is preserved by the harness internally. Matching can now be case sensitive if the `javatest.caseSensitiveJtx` system property is set to "true".

JT harness 4.4 is compatible with JT harness 4.3. For more information, see the release notes on the [dow nload page](#).

Related Projects and Links

This section contains links to technologies and blogs that are related to the JT harness technology.

Development Tools

[asmtools](#)	Assemblers and Disassemblers for producing Java .class files
[jcov](#)	The JCov tool is a code coverage tool, intimately tied to the JDK platform.
[sigtest](#)	The SigTest tool is used for determining platform compatibility of API changes against an API reference.

JT Harness Extensions

[ME Framew ork](#)	The ME Framework is an open source set of JT harness plugins that supports the Java ME platform. Test suite architects use the JT harness and the ME Framework to construct test suites for Java ME technologies.
[jtreg](#)	The jtreg test harness is used by the JDK test framework. This framework is intended primarily for regression tests. It can also be used for unit tests, functional tests, and even simple product tests.