

Java™ Platform, Micro Edition Technology Compatibility Kit Framework



The Java™ Platform, Micro Edition (Java ME) Technology Compatibility Kit (TCK) Framework is a set of JavaTest™ harness plug-in modules that provides support for testing the Java and Java ME platforms. Test suite architects can use the JavaTest harness¹ and Java ME TCK Framework together to construct test suites for Java ME technologies.

Highlights

The Java ME TCK Framework provides:

- Assistance in the testing of resource-constrained devices with low memory, slow networking, and small displays
- Agents that simplify testing on different application models and communication protocols
- Configuration and test suite libraries that speed up test suite development
- Functionality that improves test suite performance

The Framework includes:

- Agents that assist with testing of resource-constrained devices
- Functionality that helps the test harness communicate with mobile devices that use different communication protocols and provides the means to plug in custom communication protocol implementations
- Built-in support for the different Java ME application models (Main, MIDlet, Xlet, applet)
- Test deployment bundling to assist with over-the-air (OTA) provisioning for Mobile Information Device Profile (MIDP)-based implementations. The bundling functionality:
 - Packages tests into a Java Archive (JAR) file to improve performance and satisfy MIDP requirements
 - Creates a Java Application Descriptor (JAD) file to support reliable deployment to the device
 - Supports the testing of OTA provisioning actions
- Assistance for dealing with constrained devices. For example, devices with small screens, or no class-loading or reflection facilities

Designed for test suite architects, developers, and test writers

The Java ME TCK Framework simplifies and standardizes the process of creating compatibility test suites for Java ME technologies. It is a set of JavaTest harness plug-ins and shared support libraries for Java ME test suites. The plug-ins and libraries are designed to enable fast test suite development with the focus on ease of use and rich functionality features. The Java ME TCK Framework automatically accommodates the different Java ME platform application models and optional features. While originally designed to support test suites, the Java ME TCK Framework can be used with any type of test suite. Test suite architects may also extend the Java ME TCK Framework with additional functionality to support custom test types for a specific platform. For example, architects can add support for additional unit test formats or new application models.

The figure on the following page shows some of the Java ME TCK Framework components.

¹ The JavaTest harness is a general purpose, fully-featured, flexible, and configurable test harness 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 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.

Java ME TCK Framework features

The Java ME TCK Framework includes a variety of features that extend JavaTest harness functionality to provide support for the different Java ME platform stacks, applications and security models, and device constraints.

Agents

An agent is a separate program that works in conjunction with the JavaTest harness to run tests on a system other than the one that is running the JavaTest harness. Agents are typically required when testing devices that run Java ME technologies.

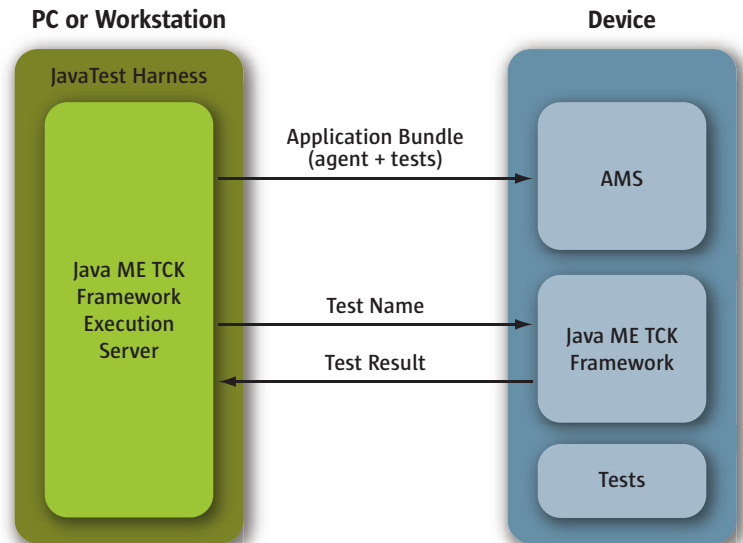
For Connected Device Configuration (CDC)-based implementations, the agent is preloaded on the device and handles test download and execution. For Connected Limited Device Configuration (CLDC)-based implementations, agents are bundled with test applications and the application management software (AMS) handles download and execution.

Interactive test framework

The interactive test framework feature supports test development and execution of interactive tests for Java ME platforms by enabling interactive tests to be run comfortably when a device has a small screen. The distributed interactive framework component has the means to create sophisticated user instruction displays and test controls.

OTA framework

The OTA framework feature provides the necessary support to test over-the-air provisioning for MIDP-based implementations. This includes obtaining, installing, and removing applications (MIDlet suites) as well as enforcing security requirements. The server-side OTA component performs the same functions for OTA tests that the agent performs for runtime tests.



Java ME TCK Framework Components

Distributed test framework

The distributed test framework is very important feature of the Java ME TCK Framework. It is used as a base for all network and interactive tests, both during development and execution. This feature enables the creation and execution of remote test components that are used for the network tests. The framework is able to send and receive test messages and, for interactive tests, differentiate a test screen from a test description and navigation screen.

Base test suite classes and configuration interview questions

The Java ME TCK Framework provides base classes and interview components that greatly simplify new test suite development. In addition, it automatically adds appropriate questions to configuration interviews.

The configuration interview extension library is easy to use and provides a straightforward way to develop configuration interviews for use in the JavaTest harness configuration editor. This library greatly simplifies development of complex configuration interviews.

Servers

The Java ME TCK Framework provides a number of custom servers that support the execution of networking tests, OTA, and regular runtime tests.

CLDC-specific features

The following features assist in testing CLDC-based devices.

Performance

- **Test bundling** — Packages several tests into one downloadable test application bundle. This significantly reduces the number of application downloads as well as the quantity of code downloaded. It also provides full control over the number and size of the test bundles.

- **Parallel test execution** — Enables several test devices to be connected to a JavaTest harness host. To reduce overall test suite execution time, automated tests run concurrently on all connected devices as part of a single JavaTest harness session.

Preinstalled agent — Enables shared agent and communication client classes to be pre-installed on the device before the test run. This reduces the amount of application code that is downloaded to run each test bundle.

Debugging support

- **Test export** — Creates standalone versions of automated tests that can be run without a harness-agent setup. This can greatly simplify test debugging.
- **MIDP 2.0 security** — Provides functionality to handle MIDP security features, such as:
 - Permissions
 - Certificates
 - Java Archive(JAR) file signing

Test support

- **Custom manifest and JAD add-ons** — Adds custom (test-specific) attributes to the JAR manifest and JAD files. This feature is used to test for the correct behavior of different manifest and JAD content states.

Application models

The Java ME TCK Framework understands different application models, enabling it to provide test execution support for the CDC and ClDC stacks. In addition, the Java ME TCK Framework supports optional packages that define specific application models such as the

Java TV™ javax.tv.xlet API. The Java ME TCK Framework also includes test execution agents that support the following application models:

- Main
- MIDlet
- Xlet (javax.microedition.xlet.Xlet)
- Applet

In addition, test suite architects and developers can add support for other application models.

Types of supported tests

The Java ME TCK Framework supports the following types of tests:

- Simple automated tests
- Network distributed tests
- Interactive tests
- Interactive distributed tests
- Dual-duty tests for security and other constraints (for example, small screens)
- OTA download tests for MIDP-based implementations

Open Source

The cqME™ project (cqme.dev.java.net/), is the home for Java ME platform compatibility and quality testing. The goal of the cqME project is to develop and improve the tools used to test Java ME technologies. The ME Framework is a sub-project of the cqME project — it is the open source version of the commercial Java TCK ME Framework. Through the cqME project you can:

- Explore the ME Framework source code
- Download binaries versions of the ME Framework libraries
- Contribute to the development of the project
- Correspond with ME Framework developers

Future open source testing technologies are also expected to find a home here. You can use these technologies to create test suites, including:

- Technology compatibility kit (TCK) test suites that test Java ME technologies
- Test suites that test application behavior on different platforms

Test suites contributed by the community can be shared with other members of the community. The cq3G test suite is a good example of such a test suite.

The Java technology community is the one of the largest software development communities in the world. The Java ME platform represents the ideal development technology for the creation and deployment of mobile data services.

The technologies that make up the cqME open source project are being developed to help ensure the quality and integrity of the Java platforms. Compatibility testing helps to reduce the fragmentation of Java technologies and helps ensure that applications run consistently across implementations of these technologies.

The ME Framework and JT harness (the open source version of the JavaTest harness at jtharness.dev.java.net/) technologies have been opened in order to develop communities that will improve them, further their development, and use them to develop their test suites. We encourage you to browse, download, contribute, and get involved.

About Sun

A singular vision, The Network is the Computer™, drives Sun in delivering industry-leading technologies that focus on the whole system — where computers, software, storage, and services combine. With a proven history of sharing, building communities, and innovation, Sun solutions create opportunities, both social and economic, around the world. You can learn more about Sun at sun.com.

For more information

For more information on the JavaTest harness, visit java.sun.com/javame/javatest.

For more information on the cqME open source project visit <https://cqme.dev.java.net/>.

For more information on the JT harness open source project visit <https://jtharness.dev.java.net/>.

Learn More

To learn more about the Java ME TCK Framework, go to java.sun.com/javame/meframework. You can download the Java ME TCK Developer's Guide from java.sun.com/javame/reference/apis.jsp#meframework.