

# JMeterTesting

Navigation trail: [JMeterProjectPages](#) - [JMeterDevelopment](#)

---

JMeter uses [JUnit](#) for testing of its code. The JUnit tests can be run from the command line, or using an IDE such as Eclipse.

This page has some details on how to use these test methods.

## Command line testing using Ant

The JMeter build file includes various targets which can be used to run tests on the code.

- test - runs all the tests using the default settings. This is the target that is invoked by Gump builds.
- test-headless - runs the tests with -Djava.awt.headless=true
- test-headed - runs the tests with -Djava.awt.headless=false
- junit - runs JUnit on a single class, which is specified using the test.case property, for example:

```
ant junit -Dtest.case=org.apache.jmeter.junit.JMeterTest (this is the default)
```

At present (build.xml version 1.147) the junit target does not work unless both optional.jar and junit.jar are on the classpath. One way to do this is to put junit.jar into ANT\_HOME/lib (where optional.jar should be already).

## Testing using Eclipse

In theory, just select a file to be tested, and choose Run As ... /JUnit Test and the test will be run using the Graphical JUnit add-in.

In practise this does not work, as the launch configuration that is generated is incomplete.

To fix the JUnit launch configuration after it has been created, select Run > Run... and click on the configuration to be changed.

Select the Arguments Tab, deselect "Use default working directory" and set the working directory to the bin directory in the workspace.

Now select the [ClassPath](#) / User classes tab. Deselect "Use default classpath" and add bin/ApacheJMeter.jar as well as all the jars in lib/ext to the list.

Hopefully, this will allow the test to be run.

At the time of writing (Dec 12, 2003) there are some classes that cannot yet be tested this way. See the Warning below for one of the possible reasons.

## WARNING

JMeter tests are normally run by the org.apache.jorphan.test.AllTests class, which performs some common setup prior to starting the tests.

Some tests will fail if this common setup has not been performed, so a change was made to org.apache.jmeter.junit.JMeterTestCase to do some basic setup, if it has not been done already.

Only a few of the JMeter test classes use [JMeterTestCase](#) at present; there are likely to be some that will fail if they are run outside of the [AllTests](#) class.

There may be other items that need to be set up to allow every class to be tested in isolation.