

iTester: Java User Manual

By David Ma

iTester is an open source software performance testing framework originally developed by Dr Shiping Chen from CSIRO Australia ICT division. This is the user manual for version 1.0 of iTester on the java platform for Windows Vista. If you are using an older version please update to the new version for bugs fixes and improvements. You can check your version of your distribution by checking the description of the build.xml script in the java home directory that came with your package.

Includes:

HttpTester.java v1.0
Randomline.java v1.01
YahooTester v1.01

Java: Requirements

iTester requires the following software packages:

Apache Ant 1.7.1 or higher, <http://ant.apache.org/>

Java Development Kit 1.6.0_07 or higher <http://java.sun.com>

While it is possible to manually compile the package and it may work on a lower version of JDK, it is neither supported nor recommended.

Java: Installation

The java distribution package of iTester has a directory structure as follows:

```
...java/: build.xml, setenv.bat
.....java/data/: url.txt, list.txt
.....java/prop/: helloworld.prop, yahoo.prop, http.prop
.....java/src/: HelloTester.java, YahooTester.java, HelloTester.java, Tester.java
.....src/com/:
.....src/com/david/: randomline.java
.....src/com/shiping/: iTester Java framework library
.....src/com/yahoo/: Yahoo developer library
```

Please check you have all the required files and directories before you attempt to start installation. To install iTester you must first edit your setenv.bat file to your local computer's settings, and then simply run the following commands while in the home directory:

setenv.bat
ant clean
ant compile

If everything is successful you should get the follow outputs.

Reset path and classpath

Set Java

Set Ant

Set home

C:\iTester>ant clean

Buildfile: build.xml

clean:

[delete] Deleting directory C:\iTester\build\classes

[mkdir] Created dir: C:\iTester\build\classes

[mkdir] Created dir: C:\iTester\build\classes\logs

[mkdir] Created dir: C:\iTester\build\classes\com

BUILD SUCCESSFUL

Total time: 0 seconds

Buildfile: build.xml

compile:

[copy] Copying 1 file to C:\iTester\build\classes

[copy] Copying 1 file to C:\iTester\build\classes

[copy] Copying 54 files to C:\iTester\build\classes\com

[javac] Compiling 10 source files to C:\iTester\build\classes

BUILD SUCCESSFUL

Total time: 2 seconds

Java: Running iTester

After a successful compilation you can run iTester from the following commands.

1. To run hello test implementation

via ant : ant helloworld

An output similar to the following should appear:

```
-- listing properties --
testName=Hello
testWarmup=5
testCooldown=5
testClassName=HelloTester
testThreadNmuber=10
testRuntime=10000
Tester.run() called
testThreadNmuber = 10
```

```
To instance a test driver
Hello Thread-0 to sleep for 4000 ms
Hello Thread-1 to sleep for 6000 ms
Hello Thread-2 to sleep for 4000 ms
Hello Thread-3 to sleep for 4000 ms
Hello Thread-4 to sleep for 4000 ms
```

2. To run yahoo test implementation

via ant : ant yahoo

An output similar to the following should appear:

```
[java] - listing properties -
[java] testName=Yahoo
[java] testWarmup=5
[java] testCooldown=5
[java] testClassName=YahooTester
[java] testThreadNmuber=2
[java] testRuntime=10000
[java] Tester.run() called
[java] testThreadNmuber = 2
[java] To instance a test driver
[java]
[java] Searching for: Devin
[java] The number of query matches in the database:
33700000
[java] The number of query matches returned: 10
[java] The position of the first result in the
overall search: 1
[java] First Page title: MySpace.com - Devin the
Dude - R.I.P. PIMP C, US -
Rap / Hip Hop - www ...
[java] First Page url:
http://www.myspace.com/devinthedude
[java] First Page summary: MySpace music profile for
Devin the Dude with to
ur dates, songs, videos, pictures, blogs, band
information, downloads and more
```

3. To run http test implementation

via ant : ant http

An output similar to the following should appear:

```
[java] - listing properties -
[java] testName=Http
[java] testWarmup=5
[java] testCooldown=5
[java] testClassName=HttpTester
[java] testThreadNmuber=2
[java] testRuntime=10000
[java] Tester.run() called
[java] testThreadNmuber = 2
[java] To instance a test driver
[java]
[java] Host: www.google.com Port: 80
[java] Host: www.google.com Port: 80
```

```
[java] Host: www.google.com Port: 80
[java] Host: www.yahoo.com Port: 80
[java] Host: www.java.com Port: 80
[java] Host: www.google.com Port: 80
[java] Host: www.usyd.edu.au Port: 80
[java] Host: www.usyd.edu.au Port: 80
[java] Host: www.yahoo.com Port: 80
```

At the end of each test implementation a report will be called with an output similar to:

```
Tester.report() called
testName = Hello
minValue = 0
aveValue = 3353
maxValue = 8000
20      3
2020    11
4020    15
6020    4
8020    1
```

minValue = minimal value or the fastest response time

aveValue = average value of the entire set

maxValue = maximum value or the slowest response time

Then a matrix of time interval rig against the number of occurrence will be displayed. This matrix can be used to plot graphs or for further statistical analysis.

To run your own implementation please refers to the developer manual.

iTester: Java Developer Manual

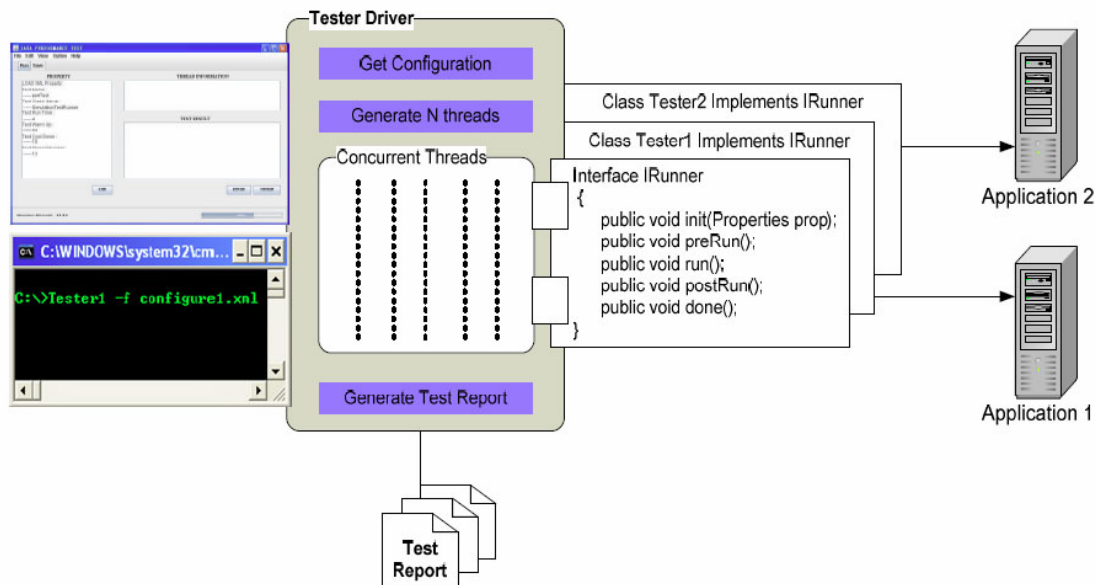
This is the developer manual for iTester version 1.0, please update to the newest version in compliance with this manual. This manual will cover standard APIs and troubleshooting for the framework. It is also highly recommended to read the original paper on the open source software performance testing framework released by CSIRO.

- S.Chen, D.Moreland, S.Nepal, J.Zic, “Yet Another Performance testing Framework”, Networking Technologies Laboratory, CSIRO ICT Centre Australia, 2008

I welcome any additional developers who would like to join us on the project. If you may be interested, please go to <http://www.sourceforge.net/> and create an account. Once an account has been made please contact *leonard_k* to be invited onto the project.

Furthermore our repository can be found at: <http://itester.cvs.sourceforge.net/>

Java: Implementation



To create your own implementations, please first examine the example implementations. From the above diagram there are two key components to an implementation.

- The test class which implements IRunner.
- The configuration script for the test class.

The IRunner interface is shown below in the API section. Using the helloworld test implementation as a template is recommended as it is a very basic example. Create a new test class and put the source in the src directory. Ant will take care of all the compilation work.

After a test class has been created, a configuration script must be created in the prop directory. Again, using the helloworld configuration as a template is highly recommended.

Finally executing the test via Ant, remember to fork the argument values in the ant file build.xml, along with the other implementation examples.

Java: APIs

IRunner

IRunner is iTester's test script implementation interface. Its standard functions are:

- **Init (Properties prop)** – Init () function acts to initialise any required properties or actions needed to perform the test. Example for this may be

setting up a server connection to a database. Each thread will call init () initially however, each test will only call it once.

- **Run ()** – is the heart of the test script, the run () function will have the specified code for which the test threads are to perform. Each thread will repeatedly call the run () function to perform what is desired by the test engineer. This allows for a wide degree of flexibility as the test engineer may program whatever actions he chooses.
- **Pre and post Run ()** – are the preparation and clean up methods, it provides the option and possibility for even greater detail tests. For example, random sleeps to simulate the typing delay of users.
- **Done ()** – Called at the completion of all the tests, it allows for final closure and freeing up of resources and any other house keeping that may be required.

Randomline

Random line is iTester's primary workload generator and execution mechanism. The default home directory for which files are read from is in *build/classes/*

Function	Parameters	Description
countLines	Input: (String) filename to open Output : (int) number of lines in that file	This function uses a while loop to count how many lines are in the file
getLineNumber	Input: (String) filename to open and line number to get Output: (String) content for that given line number	This function uses a while loop to run through the file and return back the content for that specific line
getLine	Input: (String) filename to open Output: (String) content for a random line	This function will get how many lines are in a file, and then generate a random number between 1 – max number of lines, then feed that number to the getLineNumber function in order to extract the content for a random line and then returns it.

Java: Troubleshooting

Profile	[java] java.lang.IllegalAccessException: Class org.apache.tools.ant.taskdefs.ExecuteJava can not access a member of class Tester with modifiers "public static" [java] at sun.reflect.Reflection.ensureMemberAccess(Reflection.java:65)
Reason	JVM requires a fork to start.

Solution	To start a new JVM by setting fork as follows: <pre><java dir="\${build.dir}" fork="yes" failOnError="true" className="Tester"></pre>
-----------------	--

Profile	[java] java.io.FileNotFoundException: test.prop (The system cannot find the file specified)
Reason	Ant cannot located the argument
Solution	Make sure argument location is clearly defined in build.xml at the line <pre><arg value=" "/></pre>

Profile	[java] Exception in thread "main" java.lang.NoClassDefFoundError: com/shiping/test/TestManager Or [java] Exception in thread "main" java.lang.NoClassDefFoundError: com/david Or [java] Exception in thread "main" java.lang.NoClassDefFoundError: com/yahoo
Reason	Com folder cannot be found
Solution	Make sure src/com was successfully copied to build/classes/com

Profile	[java] java.io.FileNotFoundException: .\logs\www.google.com.au_Thread-1_94 (The system cannot find the path specified) Or Similar
Reason	iTester could not find logs folder to store its downloads
Solution	Make sure the folder build/classes/logs has been created.

Profile	[java] java.io.FileNotFoundException: list.txt (The system cannot find the file specified)
Reason	Ant cannot located the file list.txt
Solution	Make sure the file data/list.txt has been successfully copied to build/classes/list.txt

Profile	[java] java.io.FileNotFoundException: url.txt (The system cannot find the file specified)
Reason	Ant cannot located the file url.txt
Solution	Make sure the file data/url.txt has been successfully copied to build/classes/url.txt