JOTM Test Suite Guide

Jeff Mesnil

March 11, 2003

Abstract

This guide describes how to install and run JOTM Test Suite

Contents

1 Introduction

This guide describes how to install and run JOTM Test Suite.
JOTM (Java Open Transaction Manager) is a fully functional Open Source implementation of JTA (Java Transaction API).
It is a mature project since it has been used for several years in the JOnAS application server project as its own transaction manager.

JOTM provides a test suite which tests its compliance to transaction standards such as JTA.

If you have any questions or comments on JOTM or its test suite, do no hesitate to let us know (`mailto:jotm@objectweb.org`).

2 Installing Test Suite

To run JOTM Test Suite, you'll need a *distribution* of JOTM as well as Ant and Junit.
You'll also need to get the test suite from CVS.

See "JOTM Installation Guide" which explains all these steps: `http://www.objectweb.org/jotm/doc/`.

3 Ant Commands

JOTM and its test suite rely on Ant for its build process.
All Ant commands are to be typed in the `tests/` directory (i.e. in the same directory than the `build.xml` file).
To have a list and descriptions of all Ant available targets for JOTM Test suite. type:

```
$ ant -projecthelp
```

3.1 Compile and build JOTM Test Suite

If you've retrieved JOTM from a *source* package or from CVS, you'll need first to create a *distribution* of JOTM. To do so, type

```
$ cd <JOTM source directory>
$ ant dist
```

This will create a *distribution* of JOTM in the `output/dist/` directory.
Then you can compile the test suite: in the `test/` directory (which in CVS is at the same level as the `jotm/` directory), type

```
$ cd <JOTM Test Suite directory>
$ ant dist
```

JOTM Test suite expects to find a JOTM distribution in the `../jotm/output/dist/` directory (which is the case if you're using CVS). If you got JOTM from a *source* or a *distribution* package, you may need to set the correct path in the `build.properties` file with the `jotm.dist` property.
If Ant can't find a JOTM distribution thanks to the `jotm.dist`, it'll alert you by a message and do nothing:

```
...
 [echo] JOTM distribution directory has not been found!
 [echo] In build.properties, it has been set to <jotm.dist property value>
 [echo] Maybe you've set it to an incorrect directory
 [echo] or you've not built a JOTM distribution yet.
...
```

In that case, you'll have to double check your settings before building and running the test suite.

3.2 Run JOTM Test Suite

Once you have a *distribution* of the Test Suite, you can run the tests by typing in the `test/` directory

```
$ ant runtest
```

All tests will be run both on RMI/JRMP and RMI/IIOP.
At the end of the test run, reports will be written in the `output/dist/reports/` directory (see section 6).

In the output generated by Ant, you'll notice the following message:

```
 [java] Timeout: killed the sub-process
 [java] Java Result: -113
```

This message is not an error message: it comes from the fact that the name server (either **rmiregistry** or **tnameserv**) is started from Ant and there is no other way to stop it than to tell Ant to destroy it after the expiration of a given timeout.

3.3 Generate Javadoc

To generate Test Suite Javadoc, in the `test/` directory, type

```
$ ant jdoc
```

Generated Javadoc will be put into the `output/dist/jdoc/` directory.

3.4 Generate Documentation

JOTM Test Suite documentation is written in LaTeX.
We use **pdflatex** tool to generate PDF files and **latex2html** to generate HTML files. Since these two tools may not be installed on your system, it's up to you to inform Ant that it'll have the tools to perform document generation.
PDF (resp. HTML) generation is triggered by `pdflatex` (resp. `latex2html`)

property on the command line. What's more, for HTML generation a shell script, **doc2html**, is used. So for the moment, you can generate HTML documentation only from Linux. Sorry... (Anyway, documentation is still available online at `http://www.objectweb.org/jotm/doc/` in both PDF and HTML format).

- If you've **pdflatex** on your system, you can ask Ant to generate PDF documentation by typing

  ```
  $ ant doc -Dpdflatex=1
  ```

- If you've **latex2html** and you're on *Linux*, you can ask Ant to generate HTML documentation by typing

  ```
  $ and doc -Dlatex2html=1
  ```

Of course, you can also do both

```
$ and doc -Dpdflatex=1 -Dlatex2html=1
```

Generated documentation will be put into the `output/dist/doc` directory.

3.5 Clean JOTM Test Suite

To remove files generated during compilation or build process, type:

```
$ ant clean
```

You'll start from a clean working directory again.

4 Project Structure

The structure of JOTM Test Suite is the following one

- `build.xml` - the main Ant build file used to build and run the test suite
- `build.properties` - configuration of Ant properties are done in this file
- `doc` - all test suite documentation sources are in this directory (e.g. `tests.tex` from which that guide was generated)
- `externals` - This directory contains one file, `junit.jar`, which is only used for Javadoc generation (Ant uses the `junit.jar` file which was put in its `lib/` directory)
- `archive` - manifest file of JOTM test suite jar file is in this directory

- `conform` - this directory contains an Ant build file which run conformance tests
- `ext` - the extensions used by the test suite. This directory contains properties files for various communication protocol configuration (RMI/JRMP, RMI/IIOP,...)
- `orb_plugins` - this directory contains an Ant build file which creates Stubs and Skeletons of the remote objects of the test suite
- `src` - the source code of the test suite

5 Customize Test Suite

By default, the test suite tests all the available test cases on all the available communication protocol.
However, you may need to be interested by only passing several times the same test while modifying JOTM until the test pass and *then* passing all the other tests.
Unfortunately, for the moment, there are no convenient way to do so...
The only way to customize test run is to modify the `jotm_tests.xml` Ant file in the `conform` directory.
You can customize two things in this file:

- the test case which are to be run
- the communication protocol you want to use

5.1 Test Cases

To choose which test cases you want to run, comment all but the one test cases you don't want to run in the `batchtest` tag inside the `run.generic.tests` target:

```
 <target name="run.generic.tests">
  ...
  <junit ...>
  ...
    <batchtest fork="yes" todir="${test.dist.reports}/${protocol.name}">
      <fileset dir="${test.src}">
        <!-- test only remote client test case -->
<!--
        <include
    name="org/objectweb/jotm/jtests/conform/LocalTestCase.java"/>
        <include
    name="org/objectweb/jotm/jtests/conform/ClientTestCase.java"/>
```

```
-->
        <include
    name="org/objectweb/jotm/jtests/conform/RemoteTestCase.java"/>
      </fileset>
    </batchtest>
  </junit>
...
```

In that example, only the remote test case will be run.

5.2 Communication protocol

If you're interested to test on only one specific communication protocol (e.g. RMI/JRMP or RMI/IIOP), in the `run.tests` target, you can comment all but the one protocol you don't want to use:

```
  ...
  <target name="run.tests">
    <condition property="interceptor.classes.presents">
      <available classname="org.omg.PortableInterceptor.ORBInitInfo">
        <classpath refid="test.classpath"/>
        <classpath refid="test.iiop.classpath"/>
      </available>
    </condition>
    <!-- test only on RMI/IIOP -->
<!--
    <antcall target="run.jrmp.tests"/>
-->
    <antcall target="run.iiop.tests"/>
  </target>
  ...
```

In that example, tests will be run only on RMI/IIOP.

5.3 Log configuration

JOTM now uses Monolog as its log system.
Log configuration is stored in a file, `trace.properties`, located in the `config/` directory of the test suite distribution.
By default, logs are stored in a file, `jotm.log`, which will be created at the root of the test suite distribution.

For more information on Monolog and Log configuration, see Monolog documentation.

6 Test reports

While you run the tests, you have a short summary from Ant output:

```
[junit] Running org.objectweb.jotm.jtests.conform.ClientTestCase
[junit] Tests run: 5, Failures: 0, Errors: 0, Time elapsed: 40.167 sec
[junit] Running org.objectweb.jotm.jtests.conform.LocalTestCase
[junit] Tests run: 17, Failures: 0, Errors: 0, Time elapsed: 32.747 sec
[junit] Running org.objectweb.jotm.jtests.conform.RemoteTestCase
[junit] Tests run: 11, Failures: 0, Errors: 0, Time elapsed: 49.914 sec
```

So you can quickly check if you're interested only by some tests.
Once all the tests are finished, a Javadoc-like HTML version of the test report will be created and put into the `output/dist/reports/` subdirectories.
There is one subdirectory for each communication protocol used during the tests (by default, there will be a `rmi_jrmp/html` and a `rmi_iiop/html` subdirectories).
Test reports are then browsable in the same way that standard Javadoc and provides global informations on the test suite, such as:

```
 ------------------------------------------------
| Tests | Failures | Errors | Success | rateTime |
 ------------------------------------------------
|    33 |        0 |      2 |  93.94% |  120.105 |
 ------------------------------------------------
```

as well as information for every test cases and test method.s

7 Contacts

If you have some trouble to install or run JOTM Test Suite, if you've any questions or if you want to contribute, do not hesitate to contact us (`mailto:jotm@objectweb.org`).