

# JOTM Examples guide

Jeff Mesnil

November 12, 2002

## Abstract

This guide describes examples provided with JOTM. It explains the scenarii of the examples, how to setup and run them.

## Contents

<b>1 Available Examples</b>	<b>1</b>
<b>2 Basic Example</b>	<b>2</b>
2.1 Scenario . . . . .	2
2.2 Setup and compilation . . . . .	2
2.3 Run the example . . . . .	2
2.3.1 On RMI/JRMP . . . . .	3
2.3.2 On RMI/IIOP . . . . .	3
2.3.3 On both RMI/JRMP and RMI/IIOP . . . . .	4
2.4 Output . . . . .	4
<b>3 JMS Example</b>	<b>5</b>
3.1 Scenario . . . . .	5
3.2 Setup and compilation . . . . .	6
3.3 Run the example . . . . .	6
3.4 Output . . . . .	7
<b>4 Contacts</b>	<b>7</b>

## 1 Available Examples

For the moment, two examples are available with JOTM:

- The first one, **basic** example, is a ... basic example! But it has some interesting features such as to show how to configure JOTM to have clients accessing the same Transaction Manager on both RMI/JRMP and RMI/IIOP.

- The second one, **jms** example, shows how to use JOTM with a JMS provider (JORAM in that case) to support distributed transactional messages for your applications.

## 2 Basic Example

All Ant commands are to be executed from the `examples/basic/` directory of a JOTM *distribution* (examples won't work from JOTM *source* directory).

### 2.1 Scenario

The **basic** example is a very simple example showing how to use a Transaction Manager. The client application (`Trivial` class) looks up the `UserTransaction`. Then it makes two transactions:

- the first one is a simple begin/commit
- the second one is also a simple begin/commit but this one is rolled back due to a timeout expiration (we made slept the `Trivial` thread for longer than the transaction timeout set for this transaction)

### 2.2 Setup and compilation

To set up this example, you'll need:

- a name server (either a RMI registry or a CORBA nameserver)
- JOTM is the Transaction Manager which provides a `UserTransaction` through JNDI
- `Trivial` is the client application
- Ant tool to compile and run the example

To compile the example, in the `examples/basic/` directory, type

```
$ ant compile
```

### 2.3 Run the example

First, you've to set `JOTM_HOME` to the directory of your JOTM distribution (e.g., `.../jotm/output/dist` from CVS).

```
$ export JOTM_HOME=<JOTM_distribution_directory>
```

To run the example, you have to be in the `examples/basic/` directory. You can run the example with three different configurations for protocol communication:

- JOTM is accessible only through **RMI/JRMP** (default configuration)
- JOTM is accessible only through **RMI/IIOP**
- JOTM is accessible through *both* **RMI/JRMP** *and* **RMI/IIOP**

There are two commands to run the example:

```
$ ant run.rmi.jrmp
```

```
$ ant run.rmi.iiop
```

The first one assumes that JOTM is accessible through RMI/JRMP and that a RMI registry is running on its default port (i.e. 1099).

The second one assumes that JOTM is accessible through RMI/IIOP and that a CORBA name server is running on port 19751.

Both also assume that the `UserTransaction` object is accessible on JNDI with the name `'UserTransaction'`.

It has to be noted that these two targets are using the same class, `Trivial`. They just differ in their settings: one is for RMI/JRMP communication whereas the other is a pure RMI/IIOP client.

### 2.3.1 On RMI/JRMP

To run the example on RMI/JRMP, type

```
$ rmiregistry &
```

```
$ $JOTM_HOME/bin/jotm.sh start -u UserTransaction &
```

```
$ ant run.rmi.jrmp
```

(by default JOTM is configured to run on RMI/JRMP so you don't have to modify `../../config/rmi.properties` file to run example on it).

### 2.3.2 On RMI/IIOP

To run the example on RMI/IIOP, first change the settings of JOTM to activate RMI/IIOP support : in `../../config/rmi.properties`

- set `carol.rmi.jrmp.activate` to `false`
- set `carol.rmi.jrmp.default` to `false`
- set `carol.rmi.iiop.activate` to `true`
- set `carol.rmi.iiop.default` to `true`

(Now only RMI/IIOP is activated and is the default protocol)

Then type

```
$ tnameserv -ORBInitialPort 19751 &
$ $JOTM_HOME/bin/jotm.sh start -u UserTransaction &
$ ant run.rmi.iiop
```

### 2.3.3 On both RMI/JRMP and RMI/IIOP

To run the example on both RMI/JRMP and RMI/IIOP, first change the settings of JOTM to activate both of them.

In `../../config/rmi.properties`

- set `carol.rmi.jrmp.activate` to true
- set `carol.rmi.jrmp.default` to false
- set `carol.rmi.iiop.activate` to true
- set `carol.rmi.iiop.default` to true

(Now both RMI/IIOP and RMI/JRMP are activated and RMI/JRMP is the default protocol)

Then type

```
$ tnameserver -ORBInitialPort 19751 &
$ rmiregistry &
$ java -jar ../../lib/jotm.jar -u UserTransaction &
```

Finally you can access JOTM on both RMI/JRMP or RMI/IIOP

```
$ ant run.rmi.jrmp
$ ant run.rmi.iiop
$ ant run.rmi.jrmp
$ ...
```

## 2.4 Output

Whatever configuration, you have chosen, the output of the example is still the same : something like

```
$ ...
$
$ [java] create initial context
$ [java] lookup UserTransaction at : UserTransaction
$
$ [java] a simple transaction wich is committed:
$ [java]       - initial status : STATUS_NO_TRANSACTION
$ [java]       - after begin status : STATUS_ACTIVE
$ [java]       - after commit status : STATUS_NO_TRANSACTION
```

```

$
$ [java] a simple transaction which is rolled back.
$ [java] we set a transaction timeout to 1 second, begin the
$ [java] transaction, and wait 5 seconds before committing it:
$ [java]         - initial status : STATUS_NO_TRANSACTION
$ [java]         - after begin status : STATUS_ACTIVE
$ [java]         - wait for 5 seconds
$ [java]         - after rollback status : STATUS_NO_TRANSACTION
$
$ [java] Trivial example is OK.
$
$ ...

```

If you have the message `Trivial example is OK.`, the example is working. If it's not the case, double check your JOTM settings. Most of the time, troubles come from incorrect settings (clients try to access JOTM through RMI/JRMP whereas there is no RMI registry but a CORBA name server,...).

### 3 JMS Example

JOTM can be used with any JMS (Java Message Service) provider to gain advantage of both *message-oriented architecture* and *distributed transactions*.

This example uses JORAM (<http://www.objectweb.org/joram/>) as its JMS provider.

All Ant commands are to be executed from the `examples/jms/` directory of a JOTM *distribution* (examples won't work from JOTM *source* directory).

#### 3.1 Scenario

The `jms` example shows how to use JOTM with a JMS provider (in our case, JORAM) to provide distributed transactional messages.

- a RMI registry is started
- JOTM is started with `UserTransaction` and `TransactionManager` objects accessible through JNDI
- An application (`SimpleJmsXa` class) starts JORAM, setups the JMS objects (`Queue`, `Session`, `ConnectionFactory`) and registers them in JOTM as XA resources

- This application then starts a message sender, `SimpleSender`, and a message receiver, `SimpleReceiver`

On one hand, `SimpleSender` sends 4 messages on a JMS queue:

- one is outside a transaction
- one is inside a transaction with a commit result
- one is inside a transaction with a rollback result
- and the last one with a special text to stop `SimpleReceiver`

On the other hand, `SimpleReceiver` receives 3 messages from the same JMS queue:

- the one which was outside a transaction
- the one which was inside a transaction with a commit result
- the last one (with the special text)

(`SimpleReceiver` does not receive the 3rd sent message because it has been rolled back.)

### 3.2 Setup and compilation

To compile the example, type

```
$ ant compile
```

### 3.3 Run the example

First, you've to set `JOTM_HOME` to the directory of your JOTM distribution (e.g., `.../jotm/output/dist` from CVS).

```
$ export JOTM_HOME=<JOTM_distribution_directory>
```

To run the example, type

```
$ rmiregistry &
$ $JOTM_HOME/bin/jotm.sh start -u UserTransaction -m
TransactionManager &
$ ant run.jms
```

Since the client application of the `jms` example is a simple RMI/JRMP client, you've to use default protocol configuration for JOTM (i.e RMI/JRMP) in `../../config/rmi.properties` file.

### 3.4 Output

the output of the **jms** example should be something like

```
$ ...
$
$ [java] [SimpleJmsXa] lookup the TransactionManager.
$ [java] [SimpleJmsXa] start the JMS server.
$ [java] [SimpleJmsXa] JMS server started.
$ [java] [SimpleJmsXa] create JMS objects, register them in JOTM and bind them.
$ [java] [SimpleJmsXa] JMS objects available.
$ [java] [SimpleJmsXa] start simple sender.
$ [java] [SimpleSender] send : non transactional message
$ [java] [SimpleSender] send : transactional message with commit
$ [java] [SimpleSender] send : transactional message with rollback
$ [java] [SimpleSender] send : LAST message
$ [java] [SimpleJmsXa] start simple receiver.
$ [java] [SimpleReceiver] received: non transactional message
$ [java] [SimpleReceiver] received: transactional message with commit
$ [java] [SimpleReceiver] received: LAST message
$ [java] [SimpleJmsXa] JMS server stopped
$
$ ...
```

If `SimpleSender` has effectively sent 4 messages and `SimpleReceiver` has effectively received only 3 messages, then the **jms** example is working!

## 4 Contacts

If you have some trouble to make the examples work or want to contribute to JOTM, do not hesitate to contact us (<mailto:jotm@objectweb.org>).