

# **JBossJTA 4.15.1**

## **JBossJTA**

# **Installation Guide**

**Stand-alone Installation Guide for JBossJTA**



**Mark Little**

**Jonathan Halliday**

**Andrew Dinn**

**Kevin Connor**

## JBossJTA 4.15.1 JBossJTA Installation Guide

### Stand-alone Installation Guide for JBossJTA

### Edition 1

Author	Mark Little	<a href="mailto:mlittle@redhat.com">mlittle@redhat.com</a>
Author	Jonathan Halliday	<a href="mailto:jhallida@redhat.com">jhallida@redhat.com</a>
Author	Andrew Dinn	<a href="mailto:adinn@redhat.com">adinn@redhat.com</a>
Author	Kevin Connor	<a href="mailto:kconnor@redhat.com">kconnor@redhat.com</a>
Editor	Misty Stanley-Jones	<a href="mailto:misty@redhat.com">misty@redhat.com</a>

Copyright © 2011 JBoss.org.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at <http://creativecommons.org/licenses/by-sa/3.0/>. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

All other trademarks are the property of their respective owners.

---

---

<b>Preface</b>	<b>v</b>
1. Prerequisites .....	v
2. Document Conventions .....	v
2.1. Typographic Conventions .....	v
2.2. Pull-quote Conventions .....	vi
2.3. Notes and Warnings .....	vii
3. We Need Feedback! .....	vii
<b>1. Preparing Your System</b>	<b>1</b>
<b>2. Operating System Services</b>	<b>3</b>
2.1. Microsoft Windows Server .....	3
2.2. Linux / UNIX .....	3
<b>3. Logging</b>	<b>5</b>
<b>4. Additional JAR Requirements</b>	<b>7</b>
<b>5. Setting Properties</b>	<b>9</b>
<b>A. Revision History</b>	<b>11</b>



---

# Preface

## 1. Prerequisites

JBossJTA works in conjunction with the rest of the JBoss Transactions suite. In addition to the documentation here, consult the JBossJTA documentation, which ships as part of JBossJTA and is also available on the JBoss Transaction Service website at <http://www.jboss.org/jbosstm>.

## 2. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the [Liberation Fonts](#)<sup>1</sup> set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later includes the Liberation Fonts set by default.

### 2.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

#### **Mono-spaced Bold**

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keycaps and key combinations. For example:

To see the contents of the file **my\_next\_bestselling\_novel** in your current working directory, enter the **cat my\_next\_bestselling\_novel** command at the shell prompt and press **Enter** to execute the command.

The above includes a file name, a shell command and a keycap, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from keycaps by the hyphen connecting each part of a key combination. For example:

Press **Enter** to execute the command.

Press **Ctrl+Alt+F2** to switch to the first virtual terminal. Press **Ctrl+Alt+F1** to return to your X-Windows session.

The first paragraph highlights the particular keycap to press. The second highlights two key combinations (each a set of three keycaps with each set pressed simultaneously).

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **mono-spaced bold**. For example:

File-related classes include **filesystem** for file systems, **file** for files, and **dir** for directories. Each class has its own associated set of permissions.

---

<sup>1</sup> <https://fedorahosted.org/liberation-fonts/>

## Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialog box text; labeled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

Choose **System** ☐ **Preferences** ☐ **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, click the **Left-handed mouse** check box and click **Close** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).

To insert a special character into a **gedit** file, choose **Applications** ☐ **Accessories** ☐ **Character Map** from the main menu bar. Next, choose **Search** ☐ **Find...** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit** ☐ **Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

## ***Mono-spaced Bold Italic*** or ***Proportional Bold Italic***

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

To connect to a remote machine using ssh, type **ssh *username@domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh john@example.com**.

The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount /home**.

To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above — *username*, *domain.name*, *file-system*, *package*, *version* and *release*. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

Publican is a *DocBook* publishing system.

## 2.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```
books      Desktop  documentation  drafts  mss    photos  stuff  svn
books_tests Desktop1  downloads      images  notes  scripts svgs
```

Source-code listings are also set in **mono-spaced roman** but add syntax highlighting as follows:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
    public static void main(String args[])
        throws Exception
    {
        InitialContext iniCtx = new InitialContext();
        Object          ref    = iniCtx.lookup("EchoBean");
        EchoHome         home   = (EchoHome) ref;
        Echo             echo   = home.create();

        System.out.println("Created Echo");

        System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
    }
}
```

## 2.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.



### Note

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.



### Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled 'Important' will not cause data loss but may cause irritation and frustration.



### Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

## 3. We Need Feedback!

If you find a typographical error in this manual, or if you have thought of a way to make this manual better, we would love to hear from you! Please submit a report in the JBoss Issue Tracker: <https://jira.jboss.org/> against the product **JBossJTA**.

When submitting a bug report, be sure to mention the manual's identifier:  
*JBossJTA\_Installation\_Guide*

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.



# Preparing Your System

## Procedure 1.1. Pre-Installation Steps

Before installing the JBossJTA software, we recommend the following administrative steps be taken, assuming a default configuration for JBossJTA.

1. **Install the distribution into the required location.**

Typically, the distribution is extracted from a **.ZIP** file.

2. **Specify the Location for the Object Store**

JBossJTA requires a minimum object store for storing the outcome of transactions in the event of system crashes. The location of this should be specified in the properties file using the `ObjectStoreEnvironmentBean.objectStoreDir` key or by using environment variable:

```
java -DObjectStoreEnvironmentBean.objectStoreDir =C:\temp foo.
```

3. **Optional: Specify the sub-directory within the Object Store root.**

By default, all object states will be stored within the **defaultStore/** sub-directory of the object store root. For instance, if the object store root is **/usr/local/Arjuna/TransactionService/ObjectStore**, the subdirectory **/usr/local/Arjuna/TransactionService/ObjectStore/defaultStore/** is used.

To change this subdirectory, set the `ObjectStoreEnvironmentBean.localOSRoot` or `com.arjuna.ats.arjuna.objectstore.localOSRoot` property variable accordingly.



# Operating System Services

## 2.1. Microsoft Windows Server

Four scripts, located in the **Services\bin\windows** folder, install and uninstall the recovery manager and transaction server services.

Installation Scripts for Microsoft Windows

Recovery Manager Service

**InstallRecoveryManagerService-NT.bat**

Transaction Server

**InstallTransactionServiceService-NT.bat**

Uninstallation Scripts for Microsoft Windows

Recovery Manager Service

**UninstallRecoveryManagerService-NT.bat**

Transaction Server

**UninstallTransactionServerService-NT.bat**



### Note

Each of the scripts requires administrative privileges.

After running any of the scripts, a status message indicates success or failure.

## 2.2. Linux / UNIX

Procedure 2.1. Installing Services in Linux / UNIX

1. **Log into the system with root privileges.**  
The installer needs these privileges to create files in **/etc**.
2. **Change to *JBOSS\_HOME*/services/installer directory.**  
**JBOSS\_HOME** refers to the directory where you extracted JBossJTA.
3. **Set the *JAVA\_HOME* variable, if necessary.**  
Set the *JAVA\_HOME* variable to the *base directory* of the JVM the service will use. The base directory is the directory above **bin/java**.
  - a. Bash: **export JAVA\_HOME="/opt/java"**
  - b. CSH: **setenv JAVA\_HOME="/opt/java"**
4. **Run the installer script.**  
**./install\_service.sh**
5. **The start-up and shut-down scripts are installed.**  
Information similar to the output below is displayed.

```
Adding $JAVA_HOME (/opt/java) to $PATH in
/opt/arjuna/ats-3.2/services/bin/solaris/recoverymanagerservice.sh
Adding $JAVA_HOME (/opt/java) to $PATH in
/opt/arjuna/ats-3.2/services/bin/solaris/transactionsserverservice.sh
Installing shutdown scripts into /etc/rcS.d:
K01recoverymanagerservice
K00transactionsserverservice
Installing shutdown scripts into /etc/rc0.d:
K01recoverymanagerservice
K00transactionsserverservice
Installing shutdown scripts into /etc/rc1.d:
K01recoverymanagerservice
K00transactionsserverservice
Installing shutdown scripts into /etc/rc2.d:
K01recoverymanagerservice
K00transactionsserverservice
Installing startup scripts into /etc/rc3.d:
S98recoverymanagerservice
S99transactionsserverservice
```

The start-up and shut-down scripts are installed for each run-level. Depending on your specific operating system, you may need to explicitly enable the services for automatic start-up.

### Procedure 2.2. Uninstalling Services in Linux / UNIX

**1. Log into the system with root privileges.**

The installer needs these privileges to delete files in **/etc**.

**2. Change to *JBoss\_HOME*/services/installer directory.**

***JBoss\_HOME*** refers to the directory where you extracted JBossJTA.

**3. Run the installation script with the -u option.**

***./install\_services.sh -u***

**4. The start-up and shut-down scripts are removed.**

Messages like the ones below indicate that the start-up and shut-down scripts have been removed successfully.

```
Removing startup scripts from /etc/rc3.d:
S98recoverymanagerservice
S99transactionsserverservice
Removing shutdown scripts from /etc/rcS.d:
K01recoverymanagerservice
K00transactionsserverservice
Removing shutdown scripts from /etc/rc0.d:
K01recoverymanagerservice
K00transactionsserverservice
Removing shutdown scripts from /etc/rc1.d:
K01recoverymanagerservice
K00transactionsserverservice
Removing shutdown scripts from /etc/rc2.d:
K01recoverymanagerservice
K00transactionsserverservice
```

# Logging

The recovery manager and the transaction server services produce log files which are located in the **services/logs/** directory. Two log files are created per service.

## ***service-name-service.log***

Contains information regarding whether the service is stopped, started, restarted, or in another state.

## ***service-name.log***

Contains information logged from the actual service.

To configure what information is logged in these files, edit the appropriate LOG4J configuration files located in **services/config/**.



## Additional JAR Requirements

To use all of the facilities available within JBossJTA, you need to add all of the JAR files contained in the **lib/** directory of the distribution to the CLASSPATH .





# Setting Properties

JBossJTA has been designed to be highly configurable at runtime through the use of various property attributes. Although these attributes can be provided at runtime on the command line, it may be more convenient to specify them through a single properties file or via `setter` methods on the beans. At runtime, JBossJTA looks for the file **jbossjta-properties.xml**, in a specific search order.

1. A location specified by a system property, allowing the normal search path to be overridden.
2. The directory from which the application was executed.
3. The home directory of the user that launched JBossJTA.
4. `java.home`
5. The CLASSPATH, which normally includes the installation's **etc/** directory.
6. A default set of properties embedded in the **JAR** file.

Where properties are defined in both the system properties by using the `-D` switch, and in the properties file, the value from the system property takes precedence. This facilitates overriding individual properties easily on the command line.

The properties file uses **java.util.Properties** XML format, for example:

```
<entry key="CoordinatorEnvironmentBean.asyncCommit">NO</entry>
<entry key="ObjectStoreEnvironmentBean.objectStoreDir">/var/ObjectStore</entry>
```

You can override the name of the properties file at runtime by specifying a new file using the `com.arjuna.ats.arjuna.common.propertiesFile` attribute variable.



## Note

Unlike earlier releases, there is no longer one properties file name per module. This properties file name key is now global for all JBoss Transaction Service components in the JVM.



---

# Appendix A. Revision History

**Revision 0**      **Wed Sep 1 2010**

Conversion to Docbook

Misty Stanley-Jones [misty@redhat.com](mailto:misty@redhat.com)

**Revision 1**      **Wed Apr 13 2010**

Taken from installation guide

Tom Jenkinson

[tom.jenkinson@redhat.com](mailto:tom.jenkinson@redhat.com)

