

JBoss Transactions 4.3.0

Installation Guide

JBTS-IG-10/3/07



Legal Notices

The information contained in this documentation is subject to change without notice.

JBoss Inc. makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. JBoss Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Java™ and J2EE is a U.S. trademark of Sun Microsystems, Inc. Microsoft® and Windows NT® are registered trademarks of Microsoft Corporation. Oracle® is a registered U.S. trademark and Oracle9™, Oracle9 Server™ Oracle9 Enterprise Edition™ are trademarks of Oracle Corporation. Unix is used here as a generic term covering all versions of the UNIX® operating system. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

Arjuna is a trademark of Hewlett-Packard and used here under licence.

Copyright

JBoss, Home of Professional Open Source Copyright 2006, JBoss Inc., and individual contributors as indicated by the @authors tag. All rights reserved.

See the copyright.txt in the distribution for a full listing of individual contributors. This copyrighted material is made available to anyone wishing to use, modify, copy, or redistribute it subject to the terms and conditions of the GNU General Public License, v. 2.0. This program is distributed in the hope that it will be useful, but WITHOUT A WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

See the GNU General Public License for more details. You should have received a copy of the GNU General Public License, v. 2.0 along with this distribution; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, * MA 02110-1301, USA.

Software Version

JBoss Transactions 4.3.0

Restricted Rights Legend

Use, duplication, or disclosure is subject to restrictions as set forth in contract subdivision (c)(1)(ii) of the Rights in Technical Data and Computer Software clause 52.227-FAR14.

© Copyright 2007 JBoss Inc.

Contents

About This Guide5

What This Guide Contains.....	5
Audience	5
Prerequisites.....	5
Organization	5
Documentation Conventions.....	5
Additional Documentation.....	6
Contacting Us	7

Installing JBossTS.....9

System installation.....	9
Preparing your system	9
ObjectStore management.....	9
Specifying the ORB to use.....	10
The interface repository	10
Loading the distribution.....	10
Installing into a non-default location	11
Additional jar requirements	11
Operating System Services	11
Installing and Uninstalling.....	11
Logging.....	15
Web Services component.....	15



About This Guide

What This Guide Contains

The Installation Guide contains information on how to use JBoss Transactions 4.3.0.

Audience

This guide is most relevant to engineers who are responsible for installing JBoss Transactions 4.3.0 installations.

Prerequisites

None.

Organization

This guide contains the following chapters:

- **Chapter 1, Installing JBossTS:** This chapter contains information on the way to install JBossTS.

Documentation Conventions

The following conventions are used in this guide:

Convention	Description
<i>Italic</i>	In paragraph text, italic identifies the titles of documents that are being referenced. When used in conjunction with the Code text described below, italics identify a variable that should be replaced by the user with an actual value.
Bold	Emphasizes items of particular importance.
Code	Text that represents programming code.
Function Function	A path to a function or dialog box within an interface. For example, “Select File Open.” indicates that you should select the Open function from the File menu.
() and	Parentheses enclose optional items in command syntax. The vertical bar separates syntax items in a list of choices. For example, any of the following three items can be entered in this syntax: <code>persistPolicy (Never OnTimer OnUpdate NoMoreOftenThan)</code>
Note:	A note highlights important supplemental information.
Caution:	A caution highlights procedures or information that is necessary to avoid damage to equipment, damage to software, loss of data, or invalid test results.

Table 1 Formatting Conventions

Additional Documentation

In addition to this guide, the following guides are available in the JBoss Transactions 4.3.0 documentation set:

- JBoss Transactions 4.3.0 *Release Notes*: Provides late-breaking information about JBoss Transactions 4.3.0.
- JBoss Transactions 4.3.0 *Administration Guide*: This guide provides instructions for administering JBoss Transactions 4.3.0.
- JBoss Transactions 4.3.0 *Programmer’s Guide*: Provides guidance for writing applications.
- JBoss Transactions 4.3.0 *Quick Start Guide*: Getting started with JBossTS; not for a novice user.
- JBoss Transactions *API Programmer’s Guide*: Provides guidance when using the JTA for building transactional applications.
- JBoss Transactions *Failure Recovery Guide*: Describes the failure recovery aspects of JBossTS.
- *TxCore Programmer’s Guide*: Describes how to write transactional applications using the non-distributed transaction engine at the heart of JBossTS.

Contacting Us

Questions or comments about JBoss Transactions 4.3.0 should be directed to our support team.

Installing JBossTS

System installation

Before *JBossTS* applications can be developed and executed on your network, there are several steps that you or your system/network administrator must perform.

Preparing your system

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

- Although this is not a requirement to successfully run JBossTS, it is recommended that you first create a user (and group) *arjuna* using system administration procedures appropriate to your installation. This user should be used to run the OTS transaction manager. This ensures that all transaction logs and object store states are owned by that user and cannot be arbitrarily deleted.
- Install the distribution into the required location.
- *JBossTS* 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 `<jbts_root>/etc/jbossjts-properties.xml` properties file using the `com.arjuna.ats.arjuna.objectstore.objectStoreDir` environment variable, e.g., `java -Dcom.arjuna.ats.arjuna.objectstore.objectStoreDir=C:\temp foo`. Alternatively, the location may be added to the CLASSPATH, and *must* have *ObjectStore* within the path.
- By default, all object states will be stored within the `defaultStore` subdirectory of the object store root, e.g., `/usr/local/Arjuna/TransactionService/ObjectStore/defaultStore`. However, this subdirectory can be changed by setting the `com.arjuna.ats.arjuna.objectstore.localOSRoot` property variable accordingly.
- If you will be using the JBossTS transactional JDBC driver or JTA with the JTS then you must change the JTA implementation to use the JTS. You should consult the Programmer's Guide or the JTA manuals on how to accomplish this.

ObjectStore management

The JBossTS layer requires an object store for transaction management logs. This is the same object store that TxCore and JBossJTA use, so careful configuration and management of that is necessary.

Within the transaction service installation, object store is updated regularly whenever transactions are created, or when *Transactional Objects for Java* is used. In a failure free environment, the only object states which should reside within the object store are those representing objects created with the *Transactional Objects for Java* API. However, if failures occur, transaction logs may remain in the object store until crash recovery facilities have resolved the transactions they represent. As such it is very important that the contents of the object store are not deleted without due care and attention, as this will make it impossible to resolve in doubt transactions. In addition, if multiple users share the same object store it is important that they realize this and do not simply delete the contents of the object store assuming it is an exclusive resource.

Specifying the ORB to use

JDK releases from 1.2.2 onwards include a minimum ORB implementation from Sun. If using such a JDK in conjunction with another ORB it is necessary to tell the JVM which ORB to use. This happens by specifying the `org.omg.CORBA.ORBClass` and `org.omg.CORBA.ORBSingletonClass` properties. In earlier versions of the *JBossTS* it was necessary to specify these properties explicitly, either on the command line or in the properties file. However, it is no longer a requirement to do this, as the ORB Portability classes will ensure that these properties are automatically set when required. Of course it is still possible to specify these values explicitly (and necessary if not using the ORB initialization methods).

The interface repository

JBossTS comes with two idl files which are required to build and run applications: `CosTransactions.idl` and `ArjunaOTS.idl`. These files may be registered with your interface repository prior to running any applications. See your ORB manual for instructions on how to do this.

Loading the distribution

The software may be installed in `/usr/local/JBoss/jbts-4.2` on Unix machines, and `C:\Program Files\JBoss\jbts-4.2` on Windows.

The distribution is split into several Java modules (packages), located within the `lib` subdirectory, as documented below:

- `com.arjuna.ats.arjuna`: this is *TxCore*, providing the transactions, persistence and concurrency control and many of the basic types, e.g., `Uid`.
- `com.arjuna.orbportability`: this package provides many useful utility classes along with classes which provide the ORB portability harness.
- `org.omg.CosTransactions`: this package provides the raw OTS interfaces.
- `com.arjuna.ats.jts`: this is the OTS component of *JBossTS*, providing support for the `ArjunaOTS.idl` and the *JBossTS* implementation of the OTS.
- `com.arjuna.ats.jta`: this package contains support for the JTA API defined by Sun Microsystems.

- `com.arjuna.ats.jdbc`: this package contains support for JDBC 2.0.

All of these packages appear in the `lib` directory of the installation, and should be added to the programmer's `CLASSPATH`..:

Installing into a non-default location

To install in a location other than the default, it is necessary to perform the following:

- specify the location of the OTS server configuration file, using the `com.arjuna.orbportability.initialReferencesRoot` property variable. See the Orb Portability manual for further details.
- specify the location of the object store, using the `com.arjuna.ats.arjuna.objectstore.objectStoreDir` property variable.
- update your `CLASSPATH` to reflect the location of the various *JBossTS* jars in `<jbts_root>/lib` and `<jbts_root>/lib/ext`.

Additional jar requirements

In order to fully utilize all of the facilities available within *JBossTS*, it will be necessary to add the following jar files to your `CLASSPATH`: `jta-spec1_0_1.jar`, `jdbc2_0-stdext.jar` and `jndi.jar`.

Note: if using JDBC then you will need to obtain one of the supported JDBC drivers from the manufacturers web site.

Operating System Services

It is possible to run the recovery manager and transaction server as operating system services.

Installing and Uninstalling

This section explains how to install and uninstall these operating system services for the two main types of operating system: Windows and UNIX (Solaris, Linux and HP-UX).

Note: It is necessary to edit the `recovery.service.conf` and `transaction.service.conf` files and replace `PUT_JACORB_HOME_HERE` with the location of your JacORB installation.

Windows

The directory `<jbts_root>\services\bin\windows` contains the files required to install the transaction and recovery manager services as Windows NT services. The administrator can directly invoke four of these scripts to install/uninstall the recovery manager and transaction server services:

- Recovery Manager Service
 - InstallRecoveryManagerService-NT.bat – running this script will install the Recovery Manager as a Windows service.
 - UninstallRecoveryManagerService-NT.bat – running this script will uninstall the Recovery Manager as a Windows service.
- Transaction Server
 - InstallTransactionServerService-NT.bat – running this script will install the Transaction Manager as a Windows service.
 - UninstallTransactionServerService-NT.bat – running this script will uninstall the Transaction Server as a Windows service.

Please note that you must either be logged on as an administrator or have administrator privileges to install/uninstall a Windows service.

Once you have run an install script you should see the following message:

```
wrapper | JBoss Transaction Service <service name> installed.
```

Note: On some systems this message may appear briefly if launched via the Explorer.

This indicates that the service has been installed successfully. The service should also be visible in the services list from the Control Panel (see Figure 1 Services list from the Control Panel)

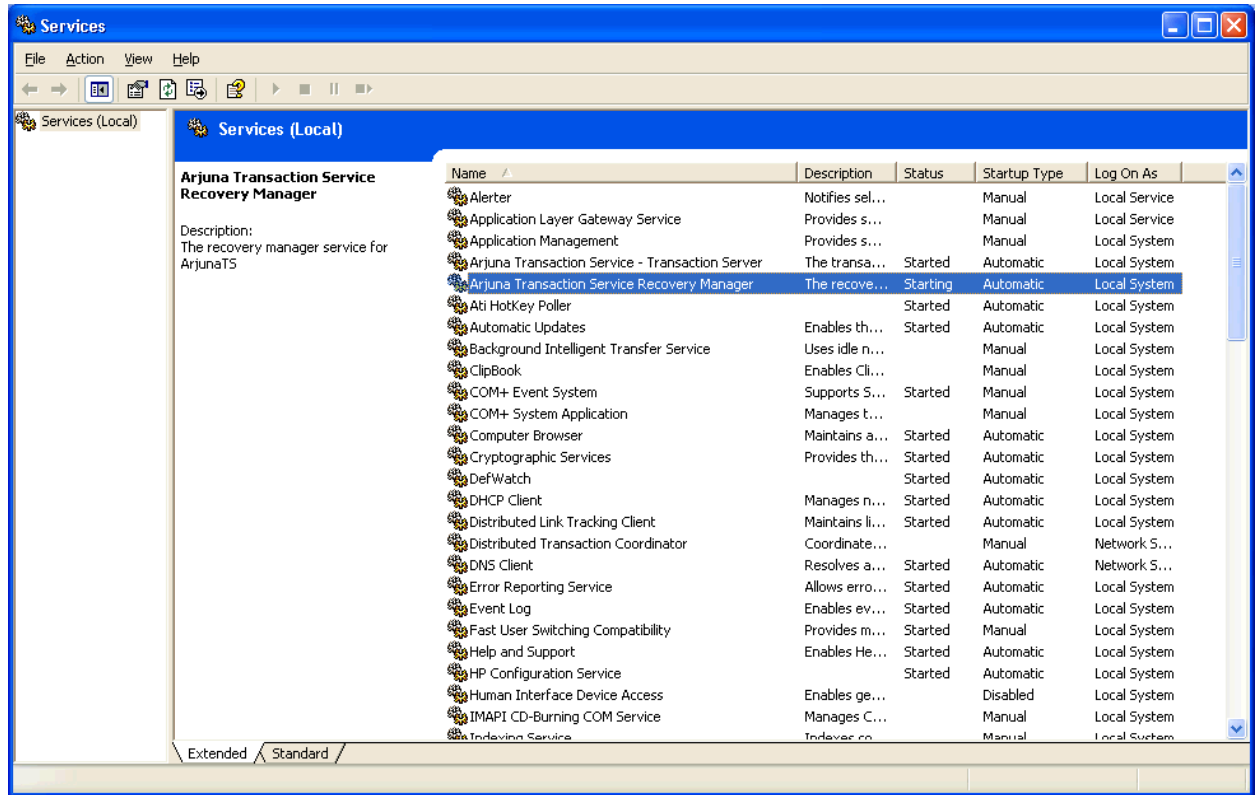


Figure 1 Services list from the Control Panel

Once you have run an uninstall script you should see the following message:

```
wrapper | JBoss Transaction Service <service name> removed.
```

This indicates that the service has been uninstalled successfully.

UNIX

To install the services on Unix machines perform the following steps:

1. Make sure that you are logged in as the user `root`, as the installer needs to create files under the directory `/etc`.
2. Change directory to `<jbts_root>/services/installer`.
3. Make sure the environment variable `JAVA_HOME` is set to the home directory of the JVM you wish the services to be run as. For example:

```
JAVA_HOME=/opt/java; export JAVA_HOME      (for sh)
```

```
setenv JAVA_HOME /opt/java                  (for csh)
```

4. Run the installer script:

```
./install_services.sh
```

The following is example output from a computer running the Solaris operating system:

```
# ./install_services.sh

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/transactionserverservice.sh
Installing shutdown scripts into /etc/rcS.d:
    K01recoverymanagerservice
    K00transactionserverservice
Installing shutdown scripts into /etc/rc0.d:
    K01recoverymanagerservice
    K00transactionserverservice
Installing shutdown scripts into /etc/rc1.d:
    K01recoverymanagerservice
    K00transactionserverservice
Installing shutdown scripts into /etc/rc2.d:
    K01recoverymanagerservice
    K00transactionserverservice
Installing startup scripts into /etc/rc3.d:
    S98recoverymanagerservice
    S99transactionserverservice
```

Upon restarting the computer the transaction server and recovery manager service should be started. To uninstall the services on a Unix machine simply perform the following steps:

1. Make sure that you are logged in as the user `root`, as the installer needs to remove files under the directory `/etc`.

2. Change directory to `services/installer`.

3. Run the installer script with the "-u" argument:

```
./install_services.sh -u
```

The following is example output from a computer running the Solaris operating system:

```
# ./install_services.sh

Removing startup scripts from /etc/rc3.d:
    S98recoverymanagerservice
    S99transactionserverservice
Removing shutdown scripts from /etc/rcS.d:
    K01recoverymanagerservice
    K00transactionserverservice
Removing shutdown scripts from /etc/rc0.d:
    K01recoverymanagerservice
    K00transactionserverservice
```

```
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 `<jbts_root>/services/logs/` directory. There are two log files per service one called `<service name>-service.log` (e.g. `recovery-manager-service.log`), which contains information regarding the state of the service (e.g. started, stopped, restarted etc). The other is called `<services-name>.log` (e.g. `recovery-manager.log`) this contains information logged from the actual service (e.g. application level logging). To configure what information is logged in these files please edit the appropriate LOG4J configuration files which are located in `<jbts_root>/services/config/`.

Web Services component

For the demo application of the Web Services transactions component and standalone coordinator, configuration is performed by editing the appropriate `build.xml` file before running `ant`. Consult the trail map accompanying these components for details.

When running within JBoss we recommend running the application server in the 'all' server configuration. This can be achieved by using `'run.sh -c all'` on unix platforms or `'run.bat -c all'` on Windows.