

ArjunaCore 4.2.3

Installation Guide

AC-IG-4/13/07





**1801 Varsity Drive
Raleigh, NC 27606-2072
USA
Phone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701
PO Box 13588
Research Triangle Park, NC 27709
USA**

Documentation-Administration

Copyright © 2007 by Red Hat, Inc. This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, V1.0 or later (the latest version is presently available at <http://www.opencontent.org/openpub/>).

Distribution of substantively modified versions of this document is prohibited without the explicit permission of the copyright holder.

Distribution of the work or derivative of the work in any standard (paper) book form for commercial purposes is prohibited unless prior permission is obtained from the copyright holder.

Red Hat and the Red Hat "Shadow Man" logo are registered trademarks of Red Hat, Inc. in the United States and other countries.

All other trademarks referenced herein are the property of their respective owners.

The GPG fingerprint of the security@redhat.com key is:

CA 20 86 86 2B D6 9D FC 65 F6 EC C4 21 91 80 CD DB 42 A6 0E

Contents

About This Guide 4

What This Guide Contains.....	4
Audience	4
Prerequisites.....	4
Organization	4
Documentation Conventions.....	4
Additional Documentation.....	5
Contacting Us	5

Installing ArjunaCore..... 7

Preparing your system	7
Operating System Services	7
Installing and Uninstalling.....	7
Logging.....	10
ObjectStore management.....	10
Additional jar requirements	11
Setting properties	11
Licensing.....	11

About This Guide

What This Guide Contains

The Installation Guide contains information on how to use ArjunaCore 4.2.3.

Audience

This guide is most relevant to engineers who are responsible for installing ArjunaCore 4.2.3 installations.

Prerequisites

None.

Organization

This guide contains the following chapters:

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

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 ArjunaCore 4.2.3 documentation set:

- ArjunaCore 4.2.3 *Release Notes*: Provides late-breaking information about ArjunaCore 4.2.3.
- ArjunaCore 4.2.3 *Administration Guide*: This guide provides instructions for administering ArjunaCore 4.2.3.
- ArjunaCore 4.2.3 *Programmer’s Guide*: Provides guidance for writing applications.

Contacting Us

Questions or comments about ArjunaCore 4.2.3 should be directed to our support team.

Installing ArjunaCore

Preparing your system

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

- Install the distribution into the required location.
- *ArjunaCore* 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 `com.arjuna.ats.arjuna.objectstore.objectStoreDir` environment variable, e.g., `java -Dcom.arjuna.ats.arjuna.objectstore.objectStoreDir=C:\temp foo`.
- 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.

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).

Windows

There are four scripts provided which will install/uninstall the recovery manager and transaction server services, these files can be found in `Services\bin\windows`:

- 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.
```

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)

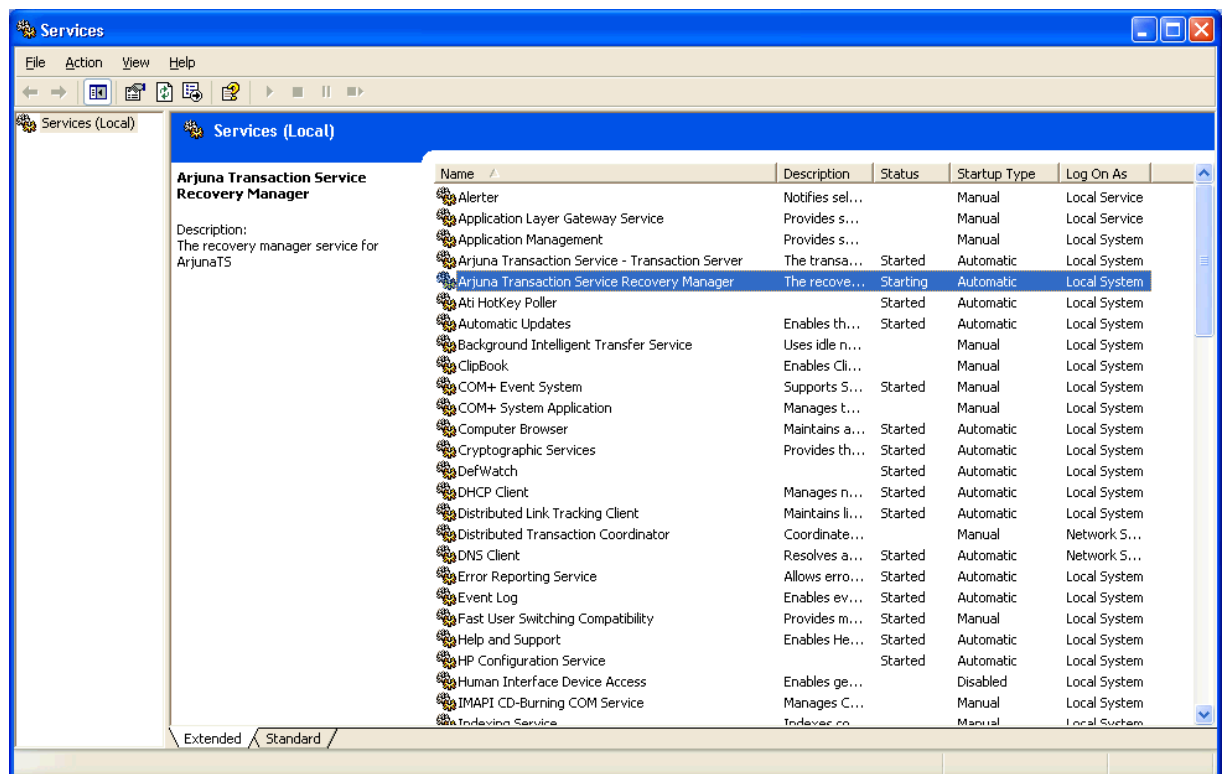


Figure 1 Services list from the Control Panel

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 `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/transactionserver.service.sh
Installing shutdown scripts into /etc/rcS.d:
    K01recoverymanagerservice
    K00transactionserver.service
Installing shutdown scripts into /etc/rc0.d:
    K01recoverymanagerservice
    K00transactionserver.service
Installing shutdown scripts into /etc/rc1.d:
    K01recoverymanagerservice
    K00transactionserver.service
Installing shutdown scripts into /etc/rc2.d:
    K01recoverymanagerservice
    K00transactionserver.service
Installing startup scripts into /etc/rc3.d:
    S98recoverymanagerservice
    S99transactionserver.service
```

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
  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. 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 `services/config/`.

ObjectStore management

Within the transaction service installation, the 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 realise this and do not simply delete the contents of the object store assuming it is an exclusive resource.

Additional jar requirements

In order to fully utilize all of the facilities available within *ArjunaCore*, it will be necessary to add all of the jar files contained in the lib directory of the distribution.

Setting properties

ArjunaCore has been designed to be highly configurable at runtime through the use of various property attributes, which will be described in subsequent sections. Although these attributes can be provided at runtime on the command line, it is possible (and may be more convenient) to specify them through a single properties file. At runtime *ArjunaCore* looks for the file `jbossts-properties.xml` in the following order:

1. the current working directory, i.e., where the application was executed from.
2. the user's home directory.
3. the `etc` directory of the *ArjunaCore* installation; this must be placed in your CLASSPATH.

If found, all entries within this file will be added to the system properties. Obviously non-*ArjunaCore* specific properties can also be specified in this file. For example:

```
<property
  name="com.arjuna.ats.arjuna.coordinator.asyncCommit"
  value="NO"/>
<property
  name="com.arjuna.ats.arjuna.objectstore.objectStoreDir"
  value="c:\temp\ObjectStore"/>
```

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

Licensing

ArjunaCore no longer require a license key in order to operate.