# ArjunaCore Installation Guide

# Contents

ArjunaCore Installation Guide........3

About This Guide........4

Preparing your system........5

Operating System Services........6

- Installing and uninstalling services - Windows........6
- Installing services - UNIX........7
- Uninstalling a service - UNIX........8
- Logging........8

ObjectStore Management........9

Additional jar requirements........10

Setting Properties........11

Licensing........12

# ArjunaCore Installation Guide

Instructions for installing ArjunaCore 4.12.0 on Linux, UNIX, and Windows servers.

**Authors**

- Mark Little
- Andrew Dinn
- Jonathan Halliday

# About This Guide

### What This Guide Contains

The Installation Guide contains information on how to use ArjunaCore 4.12.0.

### Audience

This guide is most relevant to engineers who are responsible for installing ArjunaCore 4.12.0.

### Prerequisites

None

### Document Conventions

The following conventions are used in this guide:

| **Convention** | **Description** |
|---|---|
| *Italic* | 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 program code. |
| **File** > **Open** | A path to a function or dialog box within an interface. For example, **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 (pipe) separates syntax items in a list of choices. For example, any of the following three items can be entered in this syntax: persistPolicy (Never \| OnTimer \| OnUpdate \| NoMoreOftenThan) |
| 👉 **Note:** or **:** | A note highlights important supplemental information. A warning highlights procedures or information that is necessary to avoid damage to equipment, damage to software, loss of data, or invalid test results. |

### Additional Documentation

- ArjunaCore 4.12.0 Release Notes: Provides latebreaking information about ArjunaCore 4.12.0.
- ArjunaCore 4.12.0 Administration Guide: This guide provides instructions for administering ArjunaCore 4.12.0.
- ArjunaCore 4.12.0 Programmer’s Guide: Provides guidance for writing applications.

### Contacting Us

Questions or comments about ArjunaCore 4.12.0 should be directed to our support team.

# Preparing your system

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

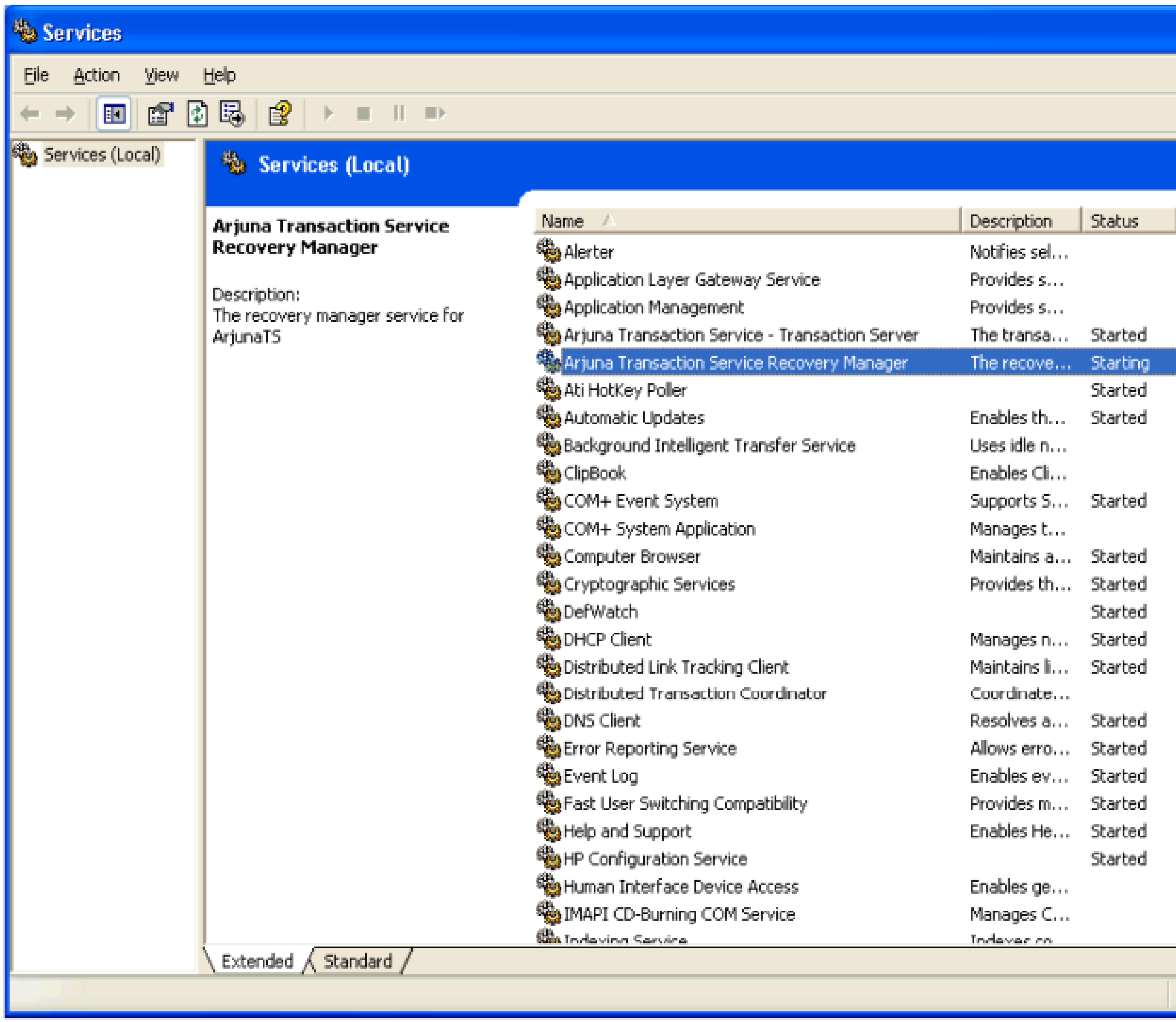
1. Install the distribution into the required location.
2. `ArjunaCore` requires a minimum object store for storing the outcome of transactions in the event of system crashes. The location of this should be specified in the properties file using the *com.arjuna.ats.arjuna.objectstore.objectStoreDir* environment variable, e.g., `java -Dcom.arjuna.ats.arjuna.objectstore.objectStoreDir=C:\temp foo`.
3. 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

## Installing and uninstalling services - 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`. You must be logged in as an administrator or have administrator privileges to install and uninstall services.

1. Recovery Manager
   a) `InstallRecoverymanagerService-NT.bat` - running this script will install the Recovery manager as a Windows service.
   b) `UninstallRecoveryManagerService-NT.bat` - running this script will uninstall the Recovery Manager as a Windows service
2. Transaction Service
   a) `InstallTransactionServerService-NT.bat` - running this script will install the Transaction Manager as a Windows service.
   b) `UninstallTransactionServerService-NT.bat` - running this script will uninstall the Transaction Server as a Windows service.
3. After running one of the scripts, you should see a message that indicates its success or failure.
4. The service should also be visible in the services list from the Control Panel.



**Arjuna Transaction Service Recovery Manager**

Description:
The recovery manager service for ArjunaTS

| Name | Description | Status |
|---|---|---|
| Alerter | Notifies sel... | |
| Application Layer Gateway Service | Provides s... | |
| Application Management | Provides s... | |
| Arjuna Transaction Service - Transaction Server | The transa... | Started |
| Arjuna Transaction Service Recovery Manager | The recove... | Starting |
| Ati HotKey Poller | | Started |
| Automatic Updates | Enables th... | Started |
| Background Intelligent Transfer Service | Uses idle n... | |
| ClipBook | Enables Cli... | |
| COM+ Event System | Supports S... | Started |
| COM+ System Application | Manages t... | |
| Computer Browser | Maintains a... | Started |
| Cryptographic Services | Provides th... | Started |
| DefWatch | | Started |
| DHCP Client | Manages n... | Started |
| Distributed Link Tracking Client | Maintains li... | Started |
| Distributed Transaction Coordinator | Coordinate... | |
| DNS Client | Resolves a... | Started |
| Error Reporting Service | Allows erro... | Started |
| Event Log | Enables ev... | Started |
| Fast User Switching Compatibility | Provides m... | Started |
| Help and Support | Enables He... | Started |
| HP Configuration Service | | Started |
| Human Interface Device Access | Enables ge... | |
| IMAPI CD-Burning COM Service | Manages C... | |
| Indexing Service | Indexes co... | |

## Installing services - UNIX

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.
   a) `JAVA_HOME=/opt/java; export JAVA_HOME` (for sh)
   b) `setenv JAVA_HOME /opt/java` (for csh)
4. Run the installer script: `./install_services.sh`
5. The following is example output from a computer running the Solaris operating system:
   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.

## Uninstalling a service - UNIX

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`
4. The following is example output from a computer running the Solaris operating system:
   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 K00transactionserverservice Removing shutdown scripts from /etc/rc2.d: K01recoverymanagerservice K00transactionserverservice

## 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 realize 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.