Introduction

This installation guide is designed to be used as both a high level installation planning guide for hardware and software prerequisites, as well as detailed step-by-step install instructions for the Portfolio Manager or Lifecycle Manager application server. For upgrade instructions please see the document appropriate for your Portfolio/Lifecycle Manager version on the SOA Software Support Center. The first overview sections are intended to provide a broad picture of the server and client environments and to provide enough background so that you can plan for an install. The subsequent sections are the detailed server install instructions that will take you thru the install process step-by-step on each of the servers. The installation process will require an administrator who has experience with the target operating system, application server, and database platform.

The install of the client portion of the product is done thru the library interface by the individual users once the servers and library are installed and configured.

If you are planning a dual Portfolio Manager and Lifecycle Manager install, the versions of the two products should match. For example, Portfolio Manager 6.6 and Lifecycle Manager 6.7 should not be used together.

Use the System Admin Guide for additional ongoing maintenance, user management, debugging, and service pack information after Portfolio or Lifecycle Manager has been installed.

Step 1: Planning and Server Requirements

Overview of the servers

The application is designed based on a traditional, three-tier server architecture: database server, application server, and web server. For the Portfolio Manager or Lifecycle Manager application, these logical software servers can be installed separately, each on their own hardware server, or they can be combined on one or two hardware servers.

The recommended configuration is to install the database on one hardware server, and combine the application and web server on a second hardware server.

Hardware Requirements and Recommendations

- o These estimates do not include the base operating system, or the prerequisites such as the database product, application server product, web server product, JDK, etc.
- o These estimates are minimum requirements. Additional memory and processor speed will generally improve performance.
- o These recommendations are based on 50 users with 500 assets in the Asset Library, without using SSL encryption on the web server.

	Each logical server on separate hardware servers	**Database on one hardware server, App and Web on second hardware server**	**All logical servers on the same hardware server**
Disk	See Database Sizing + App 2GB Web 10MB	See Database Sizing + App/Web 2.1GB	See Database Sizing + 2.1GB
Processor	Sun SPARC 500mhz Intel (or equiv) 2GHz dual core	Sun SPARC 500mhz Intel (or equiv) 2GHz dual core	Sun SPARC 500mhz Intel (or equiv) 2GHz dual core

	IBM pSeries 400mhz	IBM pSeries 400mhz	IBM pSeries 400mhz
Memory	at least 2GB	at least 2GB	at least 2.5GB

Database Sizing

Tablespace Name	Approx. Size*	Notes/Comments
LOGIDEXDATA	200 MB	DATA is where user, org group, project, base asset, and other general library information is stored. The size of this tablespace is generally not affected by the size of the assets and artifacts, but does grow directly by the number of users, projects, org groups, assets etc.
LOGIDEXLONG	350 MB	LONG is where asset artifacts that are stored by value (that is, the artifact itself is stored in the database as opposed to being referenced in some other repository). The size needed will depend directly on the size of the artifacts you have in the library. This estimate assumes that there are 2 artifacts for each asset (1000 artifacts total), with each artifact 250 KB.
LOGIDEXINDX	150 MB	approx. 25% of LOGIDEXDATA + LOGIDEXLONG
LOGIDEXTEMP	40 MB	

* These sizes are the minimums. Also note that DB2 tablespace sizes are specified in pages. The amount of disk space will depend on the page size at the time the database was created. On Oracle the DATA tablespace can also increase rapidly due to Oracle TEXT index updates. On systems with heavy asset additions/modifications, we suggest Oracle jobs to periodically rebuild the indexes to keep the sizes in check. Contact support for more information.

Server Operating System Requirements

Linux distribution supported by the J2EE application server
AIX 6.1, or 7.1.
Windows 2003, 2008, 2012 for JBoss and WebSphere

Server Software Prerequisites

Portfolio Manager or Lifecycle Manager requires one LDAP server, one application server, one database server and, optionally, one web server product from the list below.

LDAP Servers Supported

Portfolio Manager and Lifecycle Manager should be able to operate against any LDAP server as long as the schema is setup to support email addresses through the mail attribute. OpenLDAP is a free, open source product that can be used if a company does not have a preexisting LDAP server.

Application Servers Supported

WebSphere Application Server - Advanced Edition or Enterprise Edition
License Purchase Required from IBM
Version: WebSphere 6.1.0.27+, 7.0.0.11+, 8.0.x, 8.5.x
Homepage: http://www.ibm.com
WebLogic Server

License Purchase Required from Oracle
Version: 9.2 Maintenance Pack 3, 10.3, 11g, 12c
Homepage: http://www.bea.com

JBoss Server
Version: AS 4.0.5, AS 4.2.3, EAP 4.3, EAP 5.1.2, AS 6.1.0-Final, EAP 6.2
Homepage: http://www.jboss.org

Database Servers Supported

Universal Database/DB2 Server - Enterprise Edition
License Purchase Required from IBM
Version: version 9.5 FP1 or higher, or 9.7
Homepage: http://www.ibm.com

Oracle 10g or 11g Database Server – Standard Edition
License Purchase Required from Oracle
Version: 10.2.0.4.3 through 11.2.0.4 (Oracle 11.2.0.3.x requires patches for Oracle bugs 16191663 and 18615607; Oracle 11.2.0.4.x requires patch for bug 18615607)
- Oracle TEXT and XML DB features
Homepage: http://www.oracle.com

Microsoft SQL Server and SQL Server Express with Advanced Services
License Purchase Required from Microsoft
Version: 2008, 2012
Homepage: http://www.microsoft.com/sqlserver/

Web Servers Supported (optional)

Apache Web Server
Available for free download from the Apache Org.
Version: 1.3.19 or greater, or 2.x
Homepage: http://httpd.apache.org

IBM HTTP Server
Available as part of WebSphere

Java Software Development Kit (JDK)

NOTE – If you are running WebLogic or WebSphere, it should not be necessary to download and install a separate JDK, the application server should have one installed already. If you are using JBoss, you'll need to install JDK 1.5 or 1.6.

Database Requirements and DBA Authority

The base install requires a database instance created and available prior to install. One of the first steps in the install process is to create the initial database objects (user, tablespaces, and access rights) in this database instance. Both of these steps require your DBA, or DBA authority.

After this initial setup of the database instance and schema creation the database objects will be done via the product which uses standard JDBC access. The DBA is responsible for ongoing and normal DBA maintenance (backup, disk allocation, monitoring, etc).

Overview of the clients

The Portfolio and Lifecycle Manager product is accessed via a standard Web browser and through SOAP-based clients. Interactive tooling brings key features of Portfolio/Lifecycle Manager to your Desktop through a standalone Windows client or through an IDE plug-in.

Browsers

Portfolio and Lifecycle Manager support the following browsers:

- Deprecate Internet Explorer 8
- Internet Explorer 9, 10, 11
- Firefox
- Chrome
- Deprecate Safari (on mac) 5.x
- Safari (on mac) 6.x, 7.x

The browser must have the following security settings enabled:

- Cookies
- Active Scripting (IE) or JavaScript (Firefox)

There are no specific hardware requirements for the standard browser interface to Portfolio or Lifecycle Manager. Performance of the client PC will be constrained by memory, processor speed, and network connections.

Client PC Operating System and Prerequisite Requirements

The client tooling is supported on the following platforms:

- Deprecate Windows XP
- Windows 7 & 8

With the following prerequisites:

- JRE 1.5 or greater installed on the client

Client PC Hardware Requirements and Recommendations

- 1.4 GHz processor, minimum 512MB memory, 1GB memory suggested
- 250MB disk space

Step 2: Pre-Install Checklist and Prerequisites

Notation Used in this Document

Notation for Different Operating Systems

In some cases the instructions vary slightly by operating system or application server version. Those differences are denoted by:

> Operating system specific instructions:
> RHL (Red Hat Linux)
> AIX (AIX)
> Windows (WIN)

Notation for the Configuration Properties

In these install instructions you will need to use several configuration properties from the lib.conf file (created below as part of the Application Server install). In the instructions below these configuration properties are bracketed with "{" and "}" to identify them and make it easier to search on and replace these values.

> e.g. app.server.host is the property in the lib.conf file, and in the instructions below it is referred to as {app.server.host}

The configuration properties are described in the configuration file (lib.conf).

Notation for Commands and File Updates

Throughout this document you will be instructed to run certain commands and update various files. These commands and file updates are denoted with a shaded background, such as below:

> **Do not execute the following instructions. It is only an example of what executable commands look like.**
> **groupadd logidex**
> **useradd -g logidex logidex**

1. General Issues

Portfolio Manager and Lifecycle Manager use dates and times for various display and computation purposes. If the application server and database server are installed on separate servers or you will be running in a clustered setup it is suggested that a time synchronization utility, such as NTPD, is installed.

2. Install the JDK

You can install a new JDK or use an existing JDK that is already on your system. You can install the JDK in any directory on your system. Make sure the JDK you install is the correct version (see above).

During later steps in the install process you will need to provide the path to the JDK.

Install the JDK on the server where your Application Server will be (or is) installed.

3. Install or Configure a Database Product

If you do not already have an existing installation of one of the Database Server products (such as DB2 UDB, or Oracle) then you will need to install that product now. Install the Database Server according to the basic instructions that are provided by the vendor.

An initial database instance needs to be created. It is recommended that you create a separate database instance, but you can also use an existing database instance. Refer to the UDB or Oracle documentation for instructions on creating a database instance for that database product. For Oracle, the instance is the "database" that the schema will be created in. For MSSQL server, the database will be created by the installation scripts. The Oracle SID must have the Oracle Text feature installed and must have Oracle XML DB support enabled. DB2 installations must have DB2 Text Search enabled on the instance. SQL Server Express must be the version with "Advanced Services."

On UDB, a userid also needs to be created on the database server that will be used to access the database (standard UDB setup process). This userid will need to be entered in the configuration file as part of the install.

4. Install an Application Server Product

If you do not already have an existing installation of one of the Application Server products (such as WebSphere, WebLogic, etc) then you will need to install that product now. Install the Application Server according to the basic instructions provided by the vendor (WebSphere or WebLogic or JBoss)

JBoss: A JDK is not bundled with JBoss. Therefore, you will need to install a JDK.

The application servers support the concept of running the application as a separate user (other than root). If you want to run the application as a separate userid, you should create that userid (and group) now and then install the application server while logged in as that user. During the install, the userid that will be running the Portfolio Manager or Lifecycle Manager web app processes is referred to as {logidex.user}, which will be a Unix userid (such as portman).

Here is an example of the commands to create this userid:

> **groupadd repoman (or portman)**
> **useradd -g repoman repoman**

WebSphere Note: WebSphere defaults to running all application servers as root, if you would like to run Portfolio Manager or Lifecycle Manager or any application server as a different user, several additional steps are needed. Pay attention to these specific instructions which will be marked as '**WebSphere Note'**.

5. Install a Web Server Product

If you wish to use a Web server in front of your J2EE application server and do not already have one of the Web servers listed above, then you will need to install that product now. Install the Web server according to the basic instructions provided by the vendor.

6. Install an LDAP Server

If you do not have an LDAP server you can install OpenLDAP. If you would like to use native LDAP mode, where users are managed within Lifecycle Manager, ensure you use OpenLDAP 2.0.

7. Update System Settings and Reboot Application Server (Solaris Only)

Update the kernel parameters to increase the maximum file descriptors per process and system wide.

This step is not necessary for Red Hat.

Add these settings (or increase the value up to at least this number) to /etc/system:

```
set rlim_fd_max = 1024
set rlim_fd_cur = 1024
```

NOTE: The system must be rebooted for these changes to take effect.

Step 3: Install Portfolio or Lifecycle Manager on the Application Server

1. Sign-on to the application server as root / administrator

Under Windows be sure to launch your command window with Administrator privileges.

2. Create the Portfolio or Lifecycle Manager user (AIX / Linux)

You will need to create a Unix userid and Unix group on the server where the application server will be (or is) installed. This userid and group will be used to run the Portfolio Manager or Lifecycle Manager WebApp and will own the directories and files.

This is the same userid/groupid that you will be running the application server as.

WebSphere Note: See step 2, section 3 for special instructions on running as a non-root user.

```
SOL and RHL
  groupadd {logidex.group}
  useradd -g {logidex.group} {logidex.user}
AIX
  mkgroup {logidex.group}
  mkuser pgrp={logidex.group} {logidex.user}
```

3. Update {logidex.user} profile script

AIX / Linux:

Update the profile script for the user so that these environment variables (JDK_HOME and JAVA_HOME) are always set.

The instructions below are for the Bourne shell - you can adjust the statements below for your default shell.

In these instructions **/path-to-jdk-home** is the absolute directory path where the prerequisite JDK was installed (e.g. /usr/jdk142).

Switch to the user and add the appropriate lines to the profile script.

```
su – {logidex.user}
vi .profile     (use the editor of your choice and edit the appropriate shell profile script for your shell)

JDK_HOME=/path-to-jdk-home
export JDK_HOME

JAVA_HOME=/path-to-jdk-home
export JAVA_HOME
```

Return to root

> **exit**

Windows:

The location to the JDK home should be set as an environment variable in Windows. Follow these instructions to do so:

1. Launch the Control Panel
2. Open the System settings
3. Click the Advanced tab
4. Select the Environment Variables button
5. Add a new System Variable for JDK_HOME. For example:

    JDK_HOME=c:\java\jdk1.5
    JAVA_HOME=c:\java\jdk1.5

4. Create the Portfolio Manager or Lifecycle Manager root directory

AIX / Linux:

> **mkdir -p /opt/portfoliomanager (or mkdir –p /opt/lifecyclemanager)**
> **chown {logidex.user}:{logidex.group} /opt/portfoliomanager**

Windows:

Use Windows Explorer to create the application directory (e.g. c:\portfoliomanager or c:\lifecyclemanager)

5. Switch User to {logidex.user} (AIX / Linux)

The majority of the installation must be done running as the {logidex.user} (except where noted below to return to root).

> **su – {logidex.user}**

6. Unpack the Install Package from CD or Download

AIX / Linux:

Execute the installation file appropriate for your operating system. When prompted, enter a directory name that will be the *ApplicationRoot* where you will configure the application for deployment. Download and/or copy the the javaclients.zip and netclients.zip to the *ApplicationRoot* directory you specified above. The javaclients.zip and netclients.zip provide the thick-client installables for the Java and .NET platforms available from the download center within Portfolio Manager or Lifecycle Manager. Though these files are optional, it is suggested that you install them for full functionality.

> **cp /cdrom/cdrom0/v6.0-javaclients.zip** *ApplicationRoot***/javaclients.zip**
> cp /cdrom/cdrom0v5.6-netclients.zip *ApplicationRoot*/netclients.zip
>
> **(If you have downloaded the files be sure they are named 'javaclients.zip' and 'netclients.zip', not 'v6.1-javaclients.zip' for example.)**

Prevent others from reading the conf and bin directories

```
chmod o-rwx ./conf ./bin
```

Windows:

Execute the server installation file and specify the *ApplicationRoot* as the destination.

Copy the javaclients.zip and netclients.zip (if available) to the *ApplicationRoot* directory

7. Install the License file

A file named **license.xml** was supplied to you by SOA Software.

Copy the license.xml file to *ApplicationRoot*/conf/license.xml

8. Update the Configuration Parameters

The install requires a number of configuration parameters. These parameters include server names and IP addresses, directory names where the EAR will be installed, passwords, database information, etc.

These configuration parameters are described in the configuration file.

Note: If backslashes are used, in Windows paths for example, they need to be escaped with a backslash. So c:\logidex, should be entered as c:\\logidex.

Database creation notes: One decision you need to make while setting the lib.conf parameters is what character set you will be using to create the database. The character set determines what characters can be stored (without conversion) in the database character columns. Potential character sets include ISO8859-1 through ISO8859-15, UTF-8, and others. If you choose a multi-byte character set (UTF-8) then Portfolio Manager or Lifecycle Manager needs to create wider columns than if you choose a single byte character set (ISO8859). The database.expanded.columns parameter in the lib.conf file determines whether or not columns are created with more room for possible multi-byte characters. If you anticipate users creating assets with many extended characters you should probably set this value to true. If the users will likely only be using ASCII characters or very few extended characters, then it should be alright to set this value to false. If you created the database with a single byte character set, such as ISO8859-1, then this parameter should be set to false, as it only increases wasted space.

Oracle: On Oracle the character set is specified when you create an instance. Depending on what character set was specified when the database instance was created, you may need to adjust the database.expanded.columns parameter.

UDB: On DB2 UDB the character set is specified in the SQL scripts that are generated in the following steps. You'll need to ensure that the database.expanded.columns parameter is set according to the character set specified in the database.codeset parameter.

Copy the lib.conf file

AIX / Linux

> **cp conf/lib.conf.orig conf/lib.conf**

Windows

> Use Windows Explorer to copy conf\lib.conf.orig to conf\lib.conf

Edit the lib.conf file

> **Update the configuration properties (the format of each configuration property is a simple shell script assignment statement). See the Configuration File for the description of each configuration parameter.**

9. Create the App Root Directory (AIX / Linux)

Note: This step may need to be run while logged in as root.

mkdir -p {logidex.app.root}
chown {logidex.user}:{logidex.group} {logidex.app.root}

10. Create the Database Objects

For Oracle Installations

- Download the Oracle JDBC driver from Oracle's website. *Oracle 11.2.0.x installations below the 11.2.0.4 PSU require the Oracle 11.1 JDBC (11.1.0.7) driver. If your Oracle installation is at or above the 11.2.0.4 PSU, you can use the JDBC driver from 11.2.0.4.*
- Copy the jar file to *ApplicationRoot*/lib
- **Warning:** if you wish to use internationalization features, or use extended (non-ASCII) characters, then the instance you create, or the existing instance you use, should have a Unicode character set (UTF-8, for example).
- Still on the App Server as {logidex.user}, run the database setup:

> **AIX / Linux / Solaris**
> **ApplicationRoot/bin/install dbsetup**
> **Windows**
> **ApplicationRoot\bin\install dbsetup**

- dbsetup creates an sql script named *ApplicationRoot*/bin/{db.instance.name}.setup.sql (or *ApplicationRoot*\bin\{db.instance.name}.setup.sql) . Copy this file to the database server as /tmp/{db.instance.name}.setup.sql.
- Your DBA might want to review this setup script. Make sure the tablespace sizes and management parameters are appropriate.
- Login to the database server as root
- Switch user (or login directly) to the Oracle system with a userid with authority to create database objects. Generally this will be the "oracle" userid.

> **su - oracle**

- Database Setup
  NOTE: ensure that ORACLE_BASE and ORACLE_HOME are set first:

  ```
  echo $ORACLE_BASE
  echo $ORACLE_HOME
  ```

- Run the setup script:

  ```
  export ORACLE_SID={db.instance.name}
  sqlplus /nolog

  SQL> connect / as sysdba
  SQL> @ /tmp/{db.instance.name}.setup.sql;
  SQL> exit
  ```

- logoff the database server and return to the application server

For UDB Installations

- You need to copy the JDBC driver and license jars (db2jcc.jar, db2jcc_license_cisuz.jar, and db2jcc_license_cu.jar) located in /home*/instance*/sqllib/java to /opt/logidex/lib on the application server.
- Still on the App Server as {logidex.user}, run the database setup:

  ```
  AIX / Linux
          ApplicationRoot/bin/install dbsetup
  Windows
          ApplicationRoot\bin\install dbsetup
  ```

- dbsetup creates an sql script named *ApplicationRoot*/bin/{db.instance.name}.setup.sql (or *ApplicationRoot*\bin\{db.instance.name}.setup.sql) . Copy this file to the database server as /tmp/{db.instance.name}.setup.sql.
- Your DBA might want to review these setup scripts. Make sure the values for the tablespace sizes are fine and there is enough space on the database server. The sizes listed in the setup script are in pages and each page size is 4k by default.
- Login to the database server as root
- A user is needed to connect to the database. This user is listed as {logidex.db.user} in the lib.conf file.

  ```
  SOL and RHL
    groupadd {logidex.db.user}
    useradd -m -g {logidex.db.user} -c "Database Application User" {logidex.db.user}
  AIX
    mkgroup {logidex.db.user}
    mkuser pgrp={logidex.db.user} gecos="Database Application User" {logidex.db.user}
  ```

- Set the password for the user. This password is used to connect to the database and is listed as {logidex.password}. When asked to enter the password for the {logidex.db.user} enter {logidex.password}.

  ```
  passwd {logidex.db.user}
  ```

- To create the database, switch to a user who belongs to the SYSADM_GROUP, usually the instance owner. This user must have authority to create database objects and have write access to the {db.data.file.dir} directory.

  ```
  su – UDBinstanceUser
  ```

- Run the setup script:

  ```
  db2  -tvf  /tmp/{db.instance.name}.setup.sql
  ```

- There are three DB2 instance variables (DB2_EVALUNCOMMITTED, DB2_SKIPDELETED, and DB2_SKIPINSERTED) that should be set if this is a dedicated instance. If Portfolio Manager and Lifecycle Manager are not in their own instances, you should analyze the impact of setting these variables on the other databases that are running within the instance. If these variables are not set, performance issues involving SQL timeouts may arise as the number of concurrent users increase.

  ```
  db2set DB2_SKIPDELETED=ON
  db2set DB2_SKIPINSERTED=ON
  db2set DB2_EVALUNCOMMITTED=ON
  ```

- Return to root:

  ```
  exit
  ```

For Microsoft SQL Installations

- Download the **jtds-1.2.*x*-dist.zip** JDBC zip file from <u>SourceForge's website</u>.
- Extract the jtds-1.2.*x*.jar file from the zip and copy it to *ApplicationRoot*/lib/jtds.jar
- Copy the jtds-1.2.x.jar file to *ApplicationRoot*/lib
- Still on the App Server as {logidex.user}, run the database setup:

  ```
  AIX / Linux
          ApplicationRoot/bin/install dbsetup
  Windows
          ApplicationRoot\bin\install dbsetup
  ```

- dbsetup creates an sql script named *ApplicationRoot*/bin/{db.instance.name}.setup.sql (or *ApplicationRoot*\bin\{db.instance.name}.setup.sql) . Copy this file to the database server.
- Your DBA might want to review this setup script. Make sure the tablespace sizes and management parameters are appropriate.
- Log into the database server and launch the Microsoft SQL Server Management Studio. Open the {db.instance.name}.setup.sql script.
- Execute the script. You shouldn't see any errors. Verify that the {db.instance.name} database has been created.
- Enable "Snapshot Isolation" mode:
  - ALTER DATABASE {db.instance.name}
      SET READ_COMMITTED_SNAPSHOT ON;
  - ALTER DATABASE {db.instance.name}
      SET ALLOW_SNAPSHOT_ISOLATION ON;

11. Install Portfolio Manager or Lifecycle Manager and Configure the Application Server

For WebLogic Installations:

- Make sure you are logged on to the App Server as {logidex.user}
- Update the WebLogic start scripts:

- **WebLogic 9, 10, 11g, or 12c**
  - Edit the script {weblogic.domain.root}/startWebLogic.sh – if this is not the script normally used to start WebLogic, please modify the associated scripts and make the changes according to the following instructions.
    - **UDB Note:** if you are configuring WebLogic against a DB2/UDB server, copy the three db2jcc jars to the machine the appserver is running on.
      - Several bugs and problems in WebLogic necessitate adding certain JARs to WebLogic's classpath.
      - The appropriate JDBC drivers must be added to the classpath.
      - You must add Portfolio Manager or Lifecycle Manager's saaj.jar to the classpath. This is due to a bug in WebLogic where it always uses its own SAAJ classes instead of those in an application.
      - The URL handlers jar must be available by the system classloader, so it also needs to be added to the classpath.
      - The following resembles the changes that need to be made. Of course, you'll need to modify paths accordingly.
        - **WebLogic 9, 10, 11g, or 12c:** Add the appropriate EXT_PRE_CLASSPATH entry to the user's profile script. Be sure to source this file after you have made the changes.

          Oracle sample:
          EXT_PRE_CLASSPATH="/opt/logidex/lib/ojdbc14.jar:/usr/bea/user_projects/domains/my domain/applications/logidex/saaj.jar:/usr/bea/user_projects/domains/mydomain/applications /logidex/urlhandlers.jar"
          export EXT_PRE_CLASSPATH

          UDB sample:
          EXT_PRE_CLASSPATH="/opt/logidex/lib/db2jcc.jar:/opt/logidex/lib/db2jcc_license_cisuz.j ar:/opt/logidex/lib/db2jcc_license_cu.jar:/usr/bea/user_projects/domains/mydomain/applicat ions/logidex/saaj.jar:/usr/bea/user_projects/domains/mydomain/applications/logidex/urlhan dlers.jar"
          export EXT_PRE_CLASSPATH

          MSSQL sample:
          EXT_PRE_CLASSPATH="/opt/logidex/lib/jtds.jar:usr/bea/user_projects/domains/mydoma in/applications/logidex/saaj.jar:/usr/bea/user_projects/domains/mydomain/applications/logid ex/urlhandlers.jar"
          export EXT_PRE_CLASSPATH

- Start the WebLogic server (adjusting the /home/bea/user_projects/… path to the WebLogic domain root). Note: you must be in the domain directory otherwise you will create another WebLogic domain in the directory you are currently in.

  cd /home/bea/user_projects/domains/mydomain
  ./startWebLogic.sh

- Start the managed server if you are deploying to one.

- The above command does not start WebLogic in the background. You may need to put it in the background once it starts up: Ctrl+Z then bg.

- Install the web application

  `ApplicationRoot/bin/install install`

- Review the *ApplicationRoot*/install.log file for any errors or warnings.
- **Recommended** - Update the JTA Timeout Value
  The JTA timeout value should be increased from the default 30 seconds. There are some upgrade and configuration commands that can exceed the default timeout, depending on the size of your library (number of assets, number of users, size and speed of your hardware, etc).

  - Start the WebLogic Server Console in a browser

    http://{app.server.host}:7001/console

  - Navigate to *domainname***> Services> JTA**
  - Click the "Lock & Edit" button if using WebLogic 9
  - Click on "JTA" Tab

    Timeout Seconds: 600

  - Click Apply/Save
  - Click on the "Activate Changes" button if using WebLogic 9

For WebSphere Installations:

- If running Portfolio Manager or Lifecycle Manager as a non-root user, {logidex.user}, complete the next series of steps, otherwise skip to the next bulleted step:

  You may see an error after changing the WebSphere application server user, such as: ejbdeploy.sh: line 7: ulimit: open files: cannot modify limit: Operation not permitted.   You can ignore this warning.

  Make sure you are logged in as {logidex.user} and generate the script to set permissions on the WebSphere directories to allow Portfolio Manager or Lifecycle Manager to run as a different user.

  `ApplicationRoot/bin/install generate-websphere-permissions`

  The file *ApplicationRoot*/bin/setupperms has been created, which modifies the WebSphere file permissions to allow application servers to run as non-root users according to the WebSphere 6 documentation. Any such application server that does run as a different group or user, needs to use the same group (logidex.group).

  If you are using a Network Deployment install, follow the WebSphere documentation in the preceding paragraph to modify your installation to run as non-root, then continue with the next bulleted step. Otherwise, if you are installing into a WebSphere Base install, then continue with these steps.

Switch to the user WebSphere is running as.

Start the Admin Server

{websphere.profile.root}/bin/startServer.sh server1 (for WebSphere Base)

Navigate to the WebSphere administration pages, usually at: http://{app.server.host}:9090/admin.

Login and select Servers ->Application Servers. For each of the application servers listed, change the group and umask by selecting: *servername* -> Java Process and Management -> Process Execution and change the following:

UMASK: 022 (for WebSphere)
Run As Group: {logidex.group}

If you would like to use the Portfolio/Lifecycle Manager start and stop scripts to bring WebSphere up and down as well, you can set Run As User to {logidex.user}. If Portfolio/Lifecycle Manager will be running with other applications, it is probably best not to do this, as it may break other applications. Select OK, and click save twice to save the changes.

Now stop WebSphere to pick up the changes.

{websphere.profile.root}/bin/stopServer.sh server1

As root, change the permissions on the WebSphere directory to allow Portfolio or Lifecycle Manager to run as different user

{logidex.root}/bin/setupperms

- Switch to the {logidex.user}
- Start the base WebSphere server (adjusting the /opt/WebSphere/... path to be the path where you installed WebSphere):

   WebSphere Base
      /opt/IBM/WebSphere/AppServer/profiles/default/bin/startServer.sh server1

- Configure WebSphere for Portfolio Manager or Lifecycle Manager

   ***ApplicationRoot*/bin/install setup-websphere**

   If WebSphere runs as a different user than {logidex.user}, then switch that that user and run the following as that user:
   su - <user-that-WebSphere-runs-as>
   ***ApplicationRoot*/bin/websphere.setup**

- Check the WebSphere administration page to see that it all installed/deployed correctly using the following URL:
  http://{app.server.host}:9090/admin
- Install the web application

   su - {logidex.user}
   ***ApplicationRoot*/bin/install install**
   Windows:
   ***ApplicationRoot*\bin\install.bat install**

- Review the *ApplicationRoot*/install.log file for any errors or warnings.
- Set the transaction timeout:

- Web Admin console->Servers->Application Servers->logidex->Container Services->Transaction Service
- Set the Total transaction lifetime timeout to 600 seconds
- Click OK, save from the top navigation bar, and save again.

WebSphere custom setting for Internet Explorer clients when SSL is enabled:

- Application servers > logidex > Web container transport chains >

- WCInboundDefault > HTTP inbound channel (HTTP_2) > Custom Properties > New

- Name: CookiesConfigureNoCache

- Value: false

For JBoss Installations:

- Copy the JDBC drivers you will be using to the *JBOSS_HOME*/server/default/lib (adjust as necessary). This is not necessary for JBoss 6.2 EAP, as the deployment scripts will install the JDBC driver from the location specified in the lib.conf file.

- **JBoss 4.0 only:** Logidex uses certain JARs that JBoss ships with as well. This causes conflicts where JBoss loads the incorrect JARs. To workaround this issue, edit the *JBOSS_HOME*/server/default/conf/jboss-service.xml file.

```
Remove the line:
<classpath codebase="${jboss.server.lib.url:lib}" archives="*"/>
And replace with the following lines
<classpath codebase="${jboss.server.lib.url:lib}" archives="activation.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="antlr-2.7.6.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="autonumber-plugin.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="bcel.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="bindingservice-plugin.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="bsf.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="bsh-1.3.0.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="bsh-deployer.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="cglib.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="commons-collections.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="commons-logging.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="dom4j.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="hibernate3.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="hsqldb.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="hsqldb-plugin.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="javassist.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="javax.servlet.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="javax.servlet.jsp.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="jaxen.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="jboss-backport-concurrent.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="jboss-common-jdbc-wrapper.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="jboss-hibernate.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="jboss-j2ee.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="jboss.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="jboss-jaxrpc.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="jboss-jca.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="jboss-jsr77.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="jboss-jsr88.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="jboss-management.jar"/>
```

```
<classpath codebase="${jboss.server.lib.url:lib}" archives="jboss-monitoring.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="jbossmq.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="jboss-remoting-int.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="jboss-remoting.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="jbossretro-rt.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="jboss-saaj.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="jboss-serialization.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="jboss-srp.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="jbosssx.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="jboss-transaction.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="jmx-adaptor-plugin.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="jnpserver.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="joesnmp.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="jpl-pattern.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="jpl-util.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="log4j.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="log4j-snmp-appender.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="mail.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="mail-plugin.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="properties-plugin.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="scheduler-plugin-example.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="scheduler-plugin.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="wsdl4j.jar"/>
<classpath codebase="${jboss.server.lib.url:lib}" archives="xmlentitymgr.jar"/>
```

Also add a line for the JDBC driver you are using:
Oracle:
```
        <classpath codebase="${jboss.server.lib.url:lib}" archives="ojdbc14.jar"/>
```
DB2:
```
        <classpath codebase="${jboss.server.lib.url:lib}" archives="db2jcc.jar"/>
        <classpath codebase="${jboss.server.lib.url:lib}" archives="db2jcc_license_cu.jar"/>
```
MSSQL:
```
        <classpath codebase="${jboss.server.lib.url:lib}" archives="jtds.jar"/>
```

For installations with Lifecycle Manager and Portfolio Manager in the same JBoss instance, also change the following attribute to replace the default value of "false" with "true"
```
<attribute name="CallByValue">true</attribute>
```

- Only JBoss 4.0.5 requires all of the changes to the jboss-service.xml in the block above, however dual RM-PortM installations require the "CallByValue" change specified above.

- JBoss uses a flat classloading scheme by default, which causes applications to conflict with one another. To enable isolation, edit the {app.server.root}/server/default/deploy/ear-deployer.xml.

Change false to true in the following the line:
```
    <attribute name="Isolated">false</attribute>
```
It should look like this:
```
    <attribute name="Isolated">true</attribute>
```

- To ensure JBoss clears its cache of compiled JSP files properly, change the "DeleteWorkDirOnContextDestroy" attribute to "true"

JBoss 4.0.5
Edit the file *JBOSS_HOME*/server/default/deploy/jbossweb-tomcat55.sar/META-INF/jboss-service.xml

JBoss 4.2.3
Edit the file *JBOSS_HOME*/server/default/deploy/jboss-web.deployer/META-INF/jboss-service.xml

```
<attribute name="DeleteWorkDirOnContextDestroy">true</attribute>
```

- Modify JBoss Java options

  AIX / Linux / Solaris
  > **Edit the file {app.server.root}/bin/run.conf**
  > **Find the JAVA_OPTS line and modify as follows (this is all on one line):**
  JAVA_OPTS="-Xms128m -Xmx1024m -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000 -XX:PermSize=128m -XX:MaxPermSize=128m"

  Windows
  > **Notepad {app.server.root}\bin\run.bat**
  > **Find the JAVA_OPTS line and set as follows:**
  set JAVA_OPTS=%JAVA_OPTS% -Xms256m -Xmx1024m -XX:PermSize=128m -XX:MaxPermSize=128m

- The install will create configuration files and copy them to the JBoss deploy directory. When JBoss is started, it will read these configuration files to determine which resources to create.   **JBoss EAP 6.2 Note:** JBoss 6.2 EAP needs to be running before executing the install command below, as this will connect to JBoss to deploy the JDBC drivers and datasource configuration.

-

  AIX / Linux
  > *ApplicationRoot*/bin/install install
  Windows
  > *ApplicationRoot*\bin\install.bat install

1.   Clustering setup

- This numbered step is only if you would like to configure Lifecycle Manager and the application server for clustering. Skip to the next numbered step if this does not apply. Follow the next set of instructions specific to your application server version to setup clustering. If your application server isn't listed, clustering Portfolio Manager / Lifecycle Manager is not supported on that platform.

  - Perform the following on each host in the cluster.

    - Create the *ApplicationRoot* directory on each of the members in the cluster, remember to substitute the actual value of *ApplicationRoot* in the following commands, or set it prior to running the following

      ```
      mkdir –p ApplicationRoot/logs/webapp
      ```

    - Change the ownership of the Portfolio Manager or Lifecycle Manager directory to whomever the application server is running as (possibly {logidex.user}:{logidex.group})

      ```
      chown –R {logidex.user}:{logidex.group} ApplicationRoot
      ```

  - **WebLogic**

    - Login to the administration console. Click the "Lock & Edit" button

    - Undeploy Portfolio Manager or Lifecycle Manager from the targeted server; from the WebLogic console select *domainname* -> Deployments -> logidex. Select the Targets tab. Place a check next to the top-level Portfolio Manager or Lifecycle Manager component and select "Change Targets". Uncheck the server and select "Yes".

- Create a cluster according to the documentation at http://edocs.bea.com/wls/docs91/index.html

- From the WebLogic console, target the Connection Pool to the cluster by selecting: *domainname ->* Services -> JDBC -> Data Sources ->LogidexDataSource. Click on the Targets tab, select the cluster, and click Save.

- Click the "Activate Changes" button.

- Edit the lib.conf file and set the app.server.cluster.name to the name of the cluster Portfolio Manager or Lifecycle Manager is deployed on (i.e. app.server.cluster.name=clustername).

- Restart all the affected managed servers.

- Run install upgrade (each clustered server should be running)

  ```
  ApplicationRoot/bin/install upgrade
  ```

- **WebSphere Network Deployment**
  - Perform the following on each host in the cluster.

  - Copy the *ApplicationRoot*/lib directory to each system so they have access to the JDBC drivers

    ```
    scp -r root@{app.server.host}:/ApplicationRoot/lib /ApplicationRoot/lib
    ```

  - Change the ownership of the logs directory to whatever user Portfolio Manager or Lifecycle Manager is running as

    ```
    chown -R {logidex.user}:{logidex.group} ApplicationRoot
    ```

  - Create a cluster based on the application server that was just installed. See the WebSphere specific documentation for instructions on how to create a cluster based on an existing application server (http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1/topic/com.ibm.websphere.nd.doc/info/ae/ae/crun_srvgr p.html).

  - Make sure to save configuration changes in the WebSphere admin console or scripting environment

  - Regenerate the WebSphere plugin using the WebSphere admin console or scripting environment

  - Edit the lib.conf file and set the app.server.cluster.name property to the name of the cluster created above.

  - Make sure the app.server.cluster.name property propagated to the configuration files

    ```
    ApplicationRoot/bin/install configure
    ```

  - Wait for the nodes of the cluster to pick up the configuration changes and deploy Portfolio Manager or Lifecycle Manager. You can verify this by checking the contents of the $WAS_ROOT/installedApps/{websphere.cell.name}/logidex.ear directory.

  - Again on each member in the cluster, create a symlink from the logidex webapp to the logging directory

    ```
    ln -s ApplicationRoot/logs/webapp \
      /opt/WebSphere/AppServer/installedApps/{websphere.cell.name}/logidex.ear/logidex.war/logs
    ```

- ▪ Start the cluster

```
ApplicationRoot/bin/libctl start cluster
```

Clustering Notes:

- ▪ If the Global Definition Template (GDT) or Logidex Process Configuration (LPC) are updated in a clustered environment, then the application server will need to be restarted.

2.   Start the LDAP server

```
ApplicationRoot/bin/slapd start          (if in native LDAP mode)
```

- ▪ If you are using a different LDAP server, ensure it is started.

3.   Restart the App Server Processes

AIX / Linux
  For unclustered systems:
  ApplicationRoot/bin/libctl stop appserver
  ApplicationRoot/bin/libctl start appserver

  For clustered systems:
  ApplicationRoot/bin/libctl stop cluster
  ApplicationRoot/bin/libctl start cluster

Windows
  For unclustered systems:
  ApplicationRoot\bin\libctl stop appserver
  ApplicationRoot\bin\libctl start appserver

4.   Return to root

  exit

5.   Install license files

Access the application's Installation Administration console and install the license.key and license.properties using the commands SetLicenseKey and SetLicenseProperties as detailed in the System Administration guide.

6.   (Optional) Setup Portfolio Manager or Lifecycle Manager to startup/shutdown when the system is booted or shutdown (AIX / Linux)

SOL
 ln -s *ApplicationRoot*/bin/libctl /etc/rc3.d/S85logidex
 ln -s *ApplicationRoot*/bin/libctl /etc/rc3.d/K35logidex
 ln -s *ApplicationRoot*/bin/slapd /etc/rc3.d/S84slapd   **(if in native LDAP mode)**
 ln -s *ApplicationRoot*/bin/slapd /etc/rc3.d/K25slapd   **(if in native LDAP mode)**

```
RHL and AIX
 ln -s ApplicationRoot/bin/libctl /etc/rc.d/rc3.d/S85logidex
 ln -s ApplicationRoot/bin/libctl /etc/rc.d/rc3.d/K35logidex
 ln -s ApplicationRoot/bin/slapd /etc/rc.d/rc3.d/S84slapd          (if in native LDAP mode)
 ln -s ApplicationRoot/bin/slapd /etc/rc.d/rc3.d/K25slapd          (if in native LDAP mode)
```

7.   (Optional) Precompile the .jsp files

This is an optional step that will improve initial performance. The .jsp files can be precompiled so they won't be compiled at run-time at first touch.

```
AIX / Linux / Solaris
   ApplicationRoot/bin/libctl jspcompile
```

Step 4: Create Asset Libraries

1. Bootstrap the setup (if in native LDAP mode – ldap.ownership.mode=native)

Due to the fact that there are no users initially in native mode, Portfolio Manager or Lifecycle Manager must be "bootstrapped". Super user access is meant for recovery purposes and not recommended for steady-state operations. An installation administrator user and library administrator user should be created. The steps are as follows:

1. Set the superuser.password in lib.conf.
2. Set admin.users to 1 or more account names you will create later.
3. If you modified anything in the lib.conf file, you will need to update the application (see the System Administration Guide, section "The Configuration File," for more detailed instructions).
4. Create an initial library by following the "Create a Library for Your Assets" which follows. Return back to this step when finished.
5. After you have created the library, log off the installation administration pages. You should be redirected to a login page with a link to request an account. Click on that link and create an installation administrator account (one you specified in admin.users in the lib.conf file). When finished you should be redirected into the library or notified of pending access, depending on the Controlled Access setting. You'll need to login as the superuser to grant appropriate privileges to the new account. Log out and, if required, request another account to be the library administrator for future libraries you create.
6. At this point you now have a valid account you can use to login to the installation administration pages. Try logging into the installation administration pages with your new account. If this is successful, it is recommended that you comment out the superuser.password in lib.conf to disable the super user.

```
http://{web.server.url.host}/lm/admin/mainMenu.do
```

2. Create a Library for Your Assets

o   Open one of the following URLs in a browser, depending on your setup. If you are using native LDAP mode and have not bootstrapped the setup according to the above instructions, use the superuser login page and use the configured {superuser.password}. Otherwise use the installation administration login page and specify a valid LDAP user configured in {admin.users} with a valid password.

http://{web.server.url.host}/lm/admin/mainMenu.do   (installation administration login page
http://{web.server.url.host}/lm/application/access/suLogin.do (superuser login page)

o   Select Create Library
o   Fill in the information for your library:

Field	Description
Basic Tab	
Library Name	This is the name for your Asset Library. It will be used in the URL when accessing the library via the web with the thick clients. You will want this to be reasonably short, but at the same time make it something that will be meaningful to your users. Naming Rules: the library name must start with an alpha character and white space is not allowed. Valid characters are: A-Z, a-z, 0-9,_ (underscore)
Description	This is more descriptive text about your Asset Library.
Sender Email	This email address will be used as the "from" email address on notifications that go out from this library to end users. The format is: username@domain (e.g. internalsupport@example.com)
Controlled Access	If the library is setup for controlled access, then all user account requests must be approved by a usage controller before they are allowed access. If the library is setup for uncontrolled access, any user who already has an account or requests an account is allowed access. If bootstrapping (in native LDAP mode) it is recommended that you say no to this option.
Enable Project Manager Authority to Create Users	If this option is selected then the Project Manager has authority to create users from LDAP. If this options is not selected then only a Usage Controller can perform this task.
Library Administrator Information	
Account	This is the library login account name for the initial library administrator. It must already be an existing user (in LDAP if using guest LDAP mode, or an existing account if using native LDAP mode). If this is the first library created on a new installation in native LDAP mode, there are no accounts yet, and this property should be left blank. To assign a library administrator later, you must login to the library using the super user.
Feedback Account Information	
First Name, Middle Name, Last Name, email	This user will receive email from your end users when they click on the "Feedback" link on the top navigation bar of the library interface. It can be any user and email account that you want (it does not have to be a user that has a library account). It is recommended that you use the same user info as the Support Account info from above, or the name of the Usage Controller for this library.