Adobe

Preparing to Install LiveCycle® ES

Adobe® LiveCycle® ES

Version 8.0

July 2007

Contents

About This Document

This document is one of several resources available to help you learn about the Adobe® LiveCycle® ES (Enterprise Suite).

What's in this document?

This document provides the information you require to prepare your server environment prior to performing an automatic or manual installation and deployment of LiveCycle ES solution components on Microsoft® Windows®, Red Hat® Linux®, SUSE™ Linux, IBM® AIX®, and Sun™ Solaris™. This document contains two types of information:

- All hardware and software requirements and configurations that *must* be already in place to ensure a successful LiveCycle ES install process.
- All tasks that *can* be performed without LiveCycle ES installed and deployed but are not needed for installing or deploying.

For installations of Adobe LiveCycle Data Services ES, please refer to the documentation provided with the Data Services ES product.

Note: If you are planning to use the LiveCycle ES turnkey installation for JBoss® Application Server, you do not need to review this document. All the instructions required for preparing your environment for a turnkey installation are available in *Installing and Deploying LiveCycle ES for JBoss Using Turnkey* at www.adobe.com/go/learn_lc_installTurnkey.

After you have completed the tasks in this guide, proceed to the *Installing and Deploying LiveCycle ES* document for your application server.

Who should read this document?

This document provides information for administrators or developers responsible for preparing the application and database servers for development, staging, and production environments prior to installing, configuring, administering, or deploying LiveCycle ES. The information provided is based on the assumption that anyone reading this guide is familiar with application servers, Red Hat Linux, SUSE Linux, Windows, AIX, or Solaris operating systems, MySQL, Oracle®, DB2®, or SQL Server database servers, and web environments.

Conventions used in this document

This guide uses the following naming conventions for common file paths.

Name	Default value	Description
[appserver root]	The home directory of the application server that runs the LiveCycle ES services.	JBoss on Windows: C:\jboss
		JBoss on Linux: /opt/jboss
		WebSphere on Windows: C:\Program Files\IBM\WebSphere\AppServer
		WebSphere on Linux and UNIX: /opt/IBM/WebSphere/AppServer
		WebSphere on AIX: /usr/IBM/WebSphere/AppServer
		WebLogic on Windows: C:\bea\weblogic92\
		WebLogic on Linux and UNIX: /opt/bea/weblogic92
BEA_HOME	The install directory for WebLogic as specified for the `BEA_HOME` environment variable.	WebLogic on Windows C:\bea
		WebLogic on Linux and UNIX: /opt/bea
[appserverdomain]	The domain that you configured on WebLogic. The default domain is called "base_domain".	WebLogic on Windows: C:\bea\user_projects\domains\base_domain
		WebLogic on Linux and UNIX: /opt/bea/user_projects/domains/base_domain
[dbserver root]	The location where the LiveCycle ES database server is installed.	Depends on the database type and your specification during installation.

Most of the information about directory locations in this guide is cross-platform (all file names and paths are case-sensitive on Linux and UNIX®). Any platform-specific information is indicated as required.

Additional information

The resources in this table can help you learn about LiveCycle ES.

For information about	See
General information about LiveCycle ES and the solution components	*LiveCycle ES Overview* at www.adobe.com/go/learn_lc_overview
What's new in the Adobe LiveCycle ES (Enterprise Suite) release	www.adobe.com/go/learn_lc_whatsNew
LiveCycle ES terminology	www.adobe.com/go/learn_lc_glossary
Other services and products that integrate with LiveCycle ES	www.adobe.com/products/livecycle
Other Adobe LiveCycle ES solution components	partners.adobe.com/public/developer/main.html
Installing LiveCycle ES	*Installing and deploying for JBoss* at www.adobe.com/go/learn_lc_installJBoss *Installing and deploying WebSphere* at www.adobe.com/go/learn_lc_installWebSphere *Installing and deploying for WebLogic* at www.adobe.com/go/learn_lc_installWebLogic
Upgrading to LiveCycle ES from a previous version.	*Preparing for Upgrading to LiveCycle ES* at www.adobe.com/go/learn_lc_upgradePreparation *Upgrading to LiveCycle ES for JBoss Turnkey* at www.adobe.com/go/learn_lc_upgradeTurnkey *Upgrading to LiveCycle ES for JBoss* at www.adobe.com/go/learn_lc_upgradeJBoss *Upgrading to LiveCycle ES for WebSphere* at www.adobe.com/go/learn_lc_upgradeWebSphere *Upgrading to LiveCycle ES for WebLogic* at www.adobe.com/go/learn_lc_upgradeWebLogic
All the documentation available for LiveCycle ES	www.adobe.com/go/learn_lc_documentation
LiveCycle ES release information and last-minute changes that occur to the product	www.adobe.com/go/learn_lc_releaseNotes
Patch updates, technical notes, and additional information about this product version	www.adobe.com/support/products/enterprise/index.html

1 Introduction to Installing LiveCycle ES

This overview summarizes the order in which you will perform steps required for preparing your server environment.

1. Prepare the hardware as per your requirements (this document).

2. Install the operating system and update with all necessary patches and service packs (this document).

3. Install and configure the database server (this document).

4. Install and configure the application server (this document).

5. Install and deploy LiveCycle ES solution components (*Installing and Deploying LiveCycle ES* for your application server).

Third-party infrastructure support

This section summarizes the referenced platform and database combinations for JBoss Application Server 4.0.3 SP1, BEA WebLogic Server® 9.2, and IBM WebSphere® Application Server 6.1.0.5.

Microsoft Windows Server

LiveCycle ES supports the following Microsoft Windows operating system:

- Microsoft Windows Server® 2003, Enterprise Edition SP1, Standard Edition SP1, R2 (32 bit architectures and VMWare ESX/GSX)

The following table lists the supported application server, JDK, and database configurations for this operating system:

Application server	JDK	Database
Red Hat JBoss Application Server 4.0.3 SP1	Sun JDK 5.0 Update 11 or later updates to 5.0	- MySQL 5.0 - IBM DB2 8.2, 8.1 FixPack 7a - Oracle 9i, 10g - Microsoft SQL Server 2005 SP1
BEA WebLogic® 9.2, (Advantage & Premium)	BEA JRockit® JDK	- IBM DB2 8.2, 8.1 FixPack 7a - Oracle 9i, 10g - Microsoft SQL Server 2005 SP1
IBM WebSphere 6.1.0.5 (Base & Network Deployment Edition)	JDK supplied with application server	- IBM DB2 8.2, 8.1 FixPack 7a - Oracle 9i, 10g - Microsoft SQL Server 2005 SP1

Note: For a detailed list of the supported database editions see .

Red Hat Enterprise Linux and SUSE Linux

LiveCycle ES supports the following Red Hat Enterprise Linux and SUSE Linux operating systems:

- Red Hat Enterprise Linux AS or ES 4.0 (32 bit architectures)
- SUSE Linux Enterprise Server 9.0 (32 bit architectures)

The following table lists the supported application server, JDK, and database configurations for these operating systems:

Application server	JDK	Database
Red Hat JBoss Application Server 4.0.3 SP1	Sun JDK 5.0 Update 11 or later updates to 5.0.	• MySQL 5.0 • IBM DB2 8.2, 8.1 FixPack 7a • Oracle 9i, 10g
BEA WebLogic 9.2 (Advantage & Premium)	BEA JRockit JDK	• IBM DB2 V8.2, V8.1 FixPack 7a • Oracle 9i, 10g • Microsoft SQL Server 2005 SP1
IBM WebSphere 6.1.0.5 (Base & Network Deployment Edition)	JDK supplied with application server	• IBM DB2 8.2, 8.1 FixPack 7a • Oracle 9i, 10g • Microsoft SQL Server 2005 SP1

Note: For a detailed list of the supported database editions see "Supported software" on page 12.

IBM AIX

LiveCycle ES supports the following IBM AIX operating system:

- IBM AIX 5L 5.3

The following table lists the supported application server, JDK, and database configurations for this operating system:

Application server	JDK	Database
IBM WebSphere 6.1.0.5 (Base & Network Deployment Edition)	JDK supplied with application server	• IBM DB2 8.2, 8.1 FixPack 7a • Oracle 9i, 10g

Note: For a detailed list of the supported database editions see "Supported software" on page 12.

Sun Solaris

LiveCycle ES supports the following Sun Solaris operating systems:

● Solaris 9, 10 (SPARC® Architectures)

The following table lists the supported application server, JDK, and database configurations for these operating systems:

Application server	JDK	Database
BEA WebLogic 9.2 (Advantage & Premium)	Sun JDK supplied with application server	● Oracle 9i, 10g
IBM WebSphere 6.1.0.5 (Base & Network Deployment Edition)	JDK supplied with application server	● IBM DB2 8.2, 8.1 FixPack 7a ● Oracle 9i, 10g

Note: For a detailed list of the supported database editions see "Supported software" on page 12.

System requirements

This section provides the minimum and recommended hardware requirement for LiveCycle ES.

Minimum hardware requirements

This table provides the minimum hardware requirement supported by LiveCycle ES.

Operating system	Minimum hardware requirement
Microsoft Windows Server 2003 Enterprise Edition or Standard Edition SP1 and R2 (32-bit edition only)	Intel® Pentium® 3 or x86 equivalent, 1 GHz processor VMWare ESX and GSX RAM - 2 GB Free disk space - 2.7 GB of temporary space plus 1.7 GB for LiveCycle ES
Sun Solaris 9 and 10	Sun UltraSPARC® IIe, 650 MHz processor RAM - 2.5 GB Free disk space - 2.7 GB of temporary space plus 1.7 GB for LiveCycle ES
IBM AIX 5L 5.3	P4 pSeries 615 (Model 6C3) 7029-6C3, 1.2 GHz processor RAM - 2.5 GB Free disk space - 2.7 GB of temporary space plus 1.7 GB for LiveCycle ES

Operating system	Minimum hardware requirement
SUSE Linux Enterprise Server 9.0 i386 (32-bit edition only)	Pentium 3 or x86 equivalent, 1 GHz processor RAM - 2.5 GB Free disk space - 2.7 GB of temporary space plus 1.7 GB for LiveCycle ES
Red Hat Enterprise Linux AS or ES 4.0 (32-bit edition only)	Pentium 3 or x86 equivalent, 1 GHz processor RAM - 2.5 GB Free disk space - 2.7 GB of temporary space plus 1.7 GB for LiveCycle ES

Recommended hardware requirements

In addition to the minimum hardware requirements listed previously, here are the recommended hardware requirements for a small production environment:

Intel environments: Pentium 4, 2.8 GHz or greater. Using a dual core processor will further enhance performance.

Sun SPARC environments: UltraSPARC III or later.

Memory requirements: 4 GB of RAM.

Supported software

This table provides a summary of the application servers, web browsers, databases, database drivers, Java™ Development Kit (JDK) versions, and LDAP servers that LiveCycle ES supports.

Required software	Supported version
Operating system	• Microsoft Windows Server 2003 Enterprise Edition or Standard Edition SP1 and R2 (32-bit edition only) • (Adobe LiveCycle Workbench ES) Windows XP Service Pack 2 • Sun Solaris 9 and 10 • IBM AIX 5L 5.3 • SUSE Linux Enterprise Server 9.0 (32-bit edition only) • Red Hat Enterprise Linux AS or ES 4.0 (32-bit edition only)
Application server	• JBoss 4.0.3 SP1 • BEA WebLogic 9.2 (Advantage and Premium editions) • IBM WebSphere 6.1.05 (Base and Network Deployment editions)
Web browser	See for a complete list of web browsers.

Required software	Supported version
JDK	• JBoss on all platforms - Sun JDK 1.5.0_11 (or later version of the 1.5.0 release) available at http://.java.sun.com • WebLogic on all platforms except Solaris - BEA JRockit® SDK 1.5.0_06 • WebLogic on Solaris - Sun JDK 1.5.0_04. This version of the JDK requires the Daylight Savings Time (TZ) update to be run in North America. You can get it at http://java.sun.com/javase/tzupdater_README.html • WebSphere on all platforms except Solaris - IBM JDK 1.5.0 SR3 (embedded in application server) • WebSphere on Solaris - Sun JDK 1.5.0_09 For more information on JDKs, refer to "LDAP configuration" on page 18.
Database	• IBM DB2 8.2 or IBM DB2 8.1 Fix Pack 7 (Enterprise Edition) • Microsoft SQL Server 2005 SP1 (Enterprise Edition) • MySQL 5.0.18 (InnoDB Engine) • Oracle 9i or 10g (Standard and Enterprise Editions)
Database driver	• IBM DB2 - db2jcc.jar and db2jcc_license_cu.jar • SQL Server 2005 JDBC 1.1 - sqljdbc.jar • MySQL - mysql-connector-java-3.1.12-bin.jar (3.1.12) • Oracle 9i and 9i/10g - ojdbc14.jar, Release 2 (10.2.0.2 Thin) **Note:** All Adobe LiveCycle Business Activity Monitoring ES metadata databases use the drivers stated above.
LDAP server	• Sun ONE 5.1 • Sun ONE 5.2 • Microsoft Active Directory 2000 • Microsoft Active Directory 2003 • Novell® eDirectory 8.7 • IBM Tivoli Directory Server 6.0
Email servers	• Microsoft Exchange 2000 • Microsoft Exchange 2003 • Lotus Notes/Domino 6/7 • SendMail (included with Red Hat 4 and SUSE 9) • Novel GroupWise 6/7
Data Services ES	• Adobe Flex™ Builder™ 2.0.1 Hot Fix 2 or higher or • Flex SDK 2.0.1 Hot Fix 2 or higher

Required software	Supported version
LiveCycle ES connector	**Connector for EMC Documentum:** ● EMC Documentum Content Server 5.3 (and later service packs) In addition, on your LiveCycle ES server you must install the version of EMC Documentum Foundation Classes (DFC) that corresponds to your version of Content Server. For more information, see *Documentum Foundation Classes Installation Guide* available from EMC Documentum. **Note:** EMC Documentum Content Server is not supported on Red Hat Enterprise Linux AS or ES 4.0. **Connector for IBM FileNet:** ● IBM FileNet P8 Content Engine 3.5.x In addition, on your LiveCycle ES server you must install the version of IBM FileNet P8 Content Java API that corresponds to your version of Content Engine. For a list of the required JAR files, see *Content Java API Developer's Guide* available from IBM FileNet.
PDF client	● Adobe Acrobat® Professional and Acrobat Standard, versions 7.0 to 8.1 (for securing documents with policies and opening policy-protected documents) ● Adobe Reader®, versions 7.0 to 8.0 (for opening policy-protected documents)

Web browser support

This section outlines the supported web browsers for the LiveCycle ES user interfaces.

End-user user interface

End-user components include these solution components:

● Adobe LiveCycle Workspace ES (Flash Player required)

● Adobe LiveCycle Reader Extensions ES (Flash Player required)

● Adobe LiveCycle Rights Management ES (browser only)

● Adobe LiveCycle PDF Generator ES (browser only)

Operating system	Flash Player	Supported browser
Microsoft Windows Vista™	Flash® Player 9*	Microsoft Internet Explorer 7 or later[1] Firefox 2.0.0.1 or later[1]
Windows 2000	Flash Player 9*	Internet Explorer 6 or later[1] Firefox 2.0.0.1 or later[1]
Windows XP	Flash Player 9*	Internet Explorer 6 or later[1] Firefox 2.0.0.1 or later[1]

Operating system	Flash Player	Supported browser
Windows Server 2003	Flash Player 9*	Internet Explorer 6 or later[1] Firefox 2.0.0.1 or later[1]
OS X v 10.4.x (PowerPC)	Flash Player 9*	Firefox 2.0.0.1 or later[1] Safari 2.x or later (not for Workspace ES)[2]
OS X v 10.4.x (Intel)	Flash Player 9*	Firefox 2.0.0.1 or later[1] Safari 2.x or later (not for Workspace ES)[2]

[1]"or later" includes major revisions. For example, Internet Explorer 6.0 or later also covers Internet Explorer 7.0.

[2]Workspace ES supports Internet Explorer and Firefox from the above table but *not* Safari.

Administrator user interface

This table outlines the supported web browsers for the LiveCycle Administration Console user interface.

Operating system	Flash Player	Supported browser
Microsoft Windows Vista	N/A	Internet Explorer 7 or later[1] Firefox 2.0.0.1 or later[1]
Windows 2000	N/A	Internet Explorer 6 or later[1] Firefox 2.0.0.1 or later[1]
Windows XP	N/A	Internet Explorer 6 or later[1] Firefox 2.0.0.1 or later[1]
Windows Server 2003	N/A	Internet Explorer 6 or later[1] Firefox 2.0.0.1 or later[1]
OS X v 10.4.x (Intel)	N/A	Firefox 2.0.0.1 or later[1]

[1]"or later" includes major revisions. For example, Microsoft Internet Explorer 6.0 or later also covers Microsoft Internet Explorer 7.0

Additional requirements for Linux or UNIX operating systems

The installer for LiveCycle ES requires the bc tool to be available on the Linux or UNIX system. Normally, the bc tool is present by default. Ensure that the bc tool (an arbitrary precision calculator language) is installed on the system before you run the installer. If the tool is not present, you will experience error messages warning about lack of disk space because the bc tool is used to compute whether enough disk space is available prior to starting the installation. The bc tool may be provided by your operating system vendor, or you can get the GNU bc tool at www.gnu.org.

Note: To verify that the tool is available, execute `which bc` from the shell. If the tool is available, the path to the program is displayed.

Additional requirements for PDF Generator ES

Note: This section applies only if your configuration requires native file format conversion (for example, Microsoft Word to PDF conversion). If you are not installing PDF Generator ES, proceed to "Additional requirements for Business Activity Monitoring ES" on page 17.

User account

You must use the same user account for the following tasks:

- Installing Microsoft Office
- Installing PDF Generator ES
- Installing Acrobat 8.1 for PDF Generator ES
- Running the application server process

Acrobat

PDF Generator ES requires that a supported version of Acrobat Professional or Acrobat Standard is installed. You can choose to install Acrobat either manually prior to running the LiveCycle ES installer or as part of the LiveCycle ES install process.

Native file conversion software installation

Before installing PDF Generator ES, you must also install the software that supports the file formats for which PDF conversion support is required. PDF Generator ES can be extended to convert these additional file types to PDF files using the following applications:

- Microsoft Office 2000, XP, or 2003 (DOC, XLS, PPT)
- Microsoft Office Visio 2003 (VSD)
- Microsoft Project 2003 (MPP)
- AutoCAD 2005 (DWG)
- Corel WordPerfect 12 (WPD)
- Adobe Photoshop® CS2 (PSD)
- Adobe FrameMaker® 7.2 ((FM)
- Adobe PageMaker® 7.0 (PMD, PM6, P65, PM)
- OpenOffice 2.0 (all associated formats)

Windows environment variables

You must set the environment variables in Windows if you plan to create PDF documents from applications such as FrameMaker, Photoshop, PageMaker, WordPerfect, and Acrobat.

The names of these environment variables are listed below:

- `Acrobat_PATH`
- `FrameMaker_PATH`
- `Notepad_PATH`
- `OpenOffice_PATH`

- `PageMaker_PATH`

- `Photoshop_PATH`

- `WordPerfect_PATH`

These environment variables are optional and need to be set only if you are going to use the corresponding application to convert PDF files through PDF Generator ES. The value of the environment variable should contain the absolute path of the executable that is used to launch the corresponding application.

For example, the variable `Photoshop_PATH` might contain the value `D:\Program Files\Adobe\ Adobe Photoshop CS2\Photoshop.exe`. However, `OpenOffice_PATH` is different from others. This variable must be set to the OpenOffice installation folder (instead of the path to the executable). A typical value of `OpenOffice_PATH` on Windows would be `C:\Program Files\OpenOffice.org 2.0\`.

You do not need to set up the paths for Microsoft Office applications such as Word, PowerPoint, Excel, Visio, and Project or for AutoCAD. The Generate PDF service starts these applications automatically, if they are installed on the server.

➤ **To set the Windows environment variables:**

1. Select **Start** > **Control Panel** > **System**.

2. Click the **Advanced** tab**.**

3. Click **Environment Variables**.

4. In the System variables section, click **New.**

5. Enter the environment variable name you need to set (for example**,** `Photoshop_PATH`). This directory is the directory containing the executable file. For example, type the following code:

   ```
   D:\Program Files\Adobe\Adobe Photoshop CS2\Photoshop.exe
   ```

Service Control Manager command line tool

Before completing an automatic installation of PDF Generator ES on Windows, ensure that the Service Control Manager command line tool, sc.exe, is installed in the Windows environment. Some Windows servers do not have this software preinstalled. By default, the sc.exe file is installed in the C:\Windows\system32 directory. Most OS installations will have this tool installed. If you do not have the tool installed, it is available in the Windows Resource Kit for your specific version of Windows.

Additional requirements for Business Activity Monitoring ES

Note: These requirements are optional and required only if you are using the email functionality of Business Activity Monitoring ES.

Business Activity Monitoring ES requires a running Simple Mail Transfer Protocol (SMTP) email server for delivering email notifications. The server is external to Business Activity Monitoring ES and is managed by your email system administrator. Contact that administrator to set up an account specifically for Business Activity Monitoring ES; you need an account and password for sending mail, an address to use in the From address field, and the name of the email server host.

LDAP configuration

Note: This configuration is optional and required only if you are using an LDAP directory to authenticate users.

Install and configure your LDAP server and database according to the vendor's documentation. See "Supported software" on page 12 for a list of supported LDAP servers. Take note of the LDAP administrator name and password to use during the LiveCycle ES configuration process. You configure LiveCycle ES to connect with the LDAP database after you have installed and deployed your LiveCycle ES services. This configuration is done using the User Manager service. See *Installing and Deploying LiveCycle ES* for your application server.

Installation considerations

This section includes considerations for an administrator when installing LiveCycle ES.

Installing from network drives

It is recommended that you only install LiveCycle ES from the installation media or a local disk drive. Attempting to install the software over a network will result in considerable delays in launching and installing.

Manual use of Acrobat restricted

If you have installed the PDF Generator ES Generate PDF service for native document conversion, use of the included Acrobat is restricted to the Generate PDF service and is not licensed for any other use.

2 | LiveCycle ES Credentials and Certificates

This chapter describes how to obtain your Reader Extensions ES Rights credential and prepare it for installation. You will install the credential in one of two ways:

- Obtaining the Reader Extensions ES Rights credential
- Obtaining digital certificates for use with Adobe LiveCycle Digital Signatures ES

Obtaining the Reader Extensions ES Rights credential

The Reader Extensions ES Rights credential is a digital certificate specific to Reader Extensions ES that enables Adobe Reader usage rights to be activated in PDF documents. If the credential is not installed, Reader Extensions ES users cannot apply usage rights to documents. You cannot use a standard digital certificate for this function; you must use the dedicated Rights credential.

The Rights credential extends the usage rights of each PDF file processed by Reader Extensions ES. It is the most important part of the software licensing and should be stored carefully in a secure environment.

The following types of Rights credentials are available:

Customer Evaluation: A credential with a limited validity period, provided to customers who want to evaluate Reader Extensions ES. Usage rights applied to documents using this credential expire when the credential expires. This type of credential is only valid for two to three months.

Production: Full-functioning credential provided to customers who have purchased the full product. Production certificates are unique to each customer but can be installed on multiple systems.

The Rights credential is delivered as a digital certificate that contains the public key, the private key, and the password used to access the credential.

If your organization orders an evaluation version of Reader Extensions ES, you receive an evaluation Rights credential, either from the sales representative through whom you ordered the product or from the website where you downloaded the evaluation product.

If your organization purchases a production version of Reader Extensions ES, the production Rights credential is delivered by Electronic Software Download (ESD). A production Rights credential is unique to your organization and can enable the specific usage rights that you require.

If you obtained Reader Extensions ES through a partner or software provider who has integrated Reader Extensions ES into their software, the Rights credential is provided to you by that partner who, in turn, receives this credential from Adobe.

Note: The Rights credential cannot be used for typical document signing or assertion of identity. For these applications, you can use a self-sign certificate or acquire an identity certificate from a Certificate Authority (CA).

Obtaining digital certificates for use with Digital Signatures ES

Digital certificates are required for use with Digital Signatures ES. Although you can configure and manage digital certificates after installing and configuring LiveCycle ES, obtaining them before installing ensures that you are ready to use LiveCycle ES when it is deployed.

Digital certificates are obtained from a Certificate Authority (CA) and sent to you by email or over the web as a certificate file. This certificate file contains the public keys (also called *certificates*) and references to private keys (also called *credentials*) used for encrypting and signing documents. Certificates do not contain actual private keys; instead, they contain a reference to the identity of the user who keeps the private key securely stored in an encrypted file or Hardware Security Module (HSM).

You can use Internet Explorer (Windows) or OpenSSL (UNIX) to export PFX, P12, and CER files for certificates stored in any compatible certificate store available on your computer. PFX files can be exported only as permitted by the certificate store or the credential itself. CER files holding the public key that corresponds to a credential can also be exported from PFX files using either Internet Explorer or OpenSSL.

Note: You can configure and manage certificates, credentials, and CRLs for use with LiveCycle ES by using Trust Store Management, which is accessible through LiveCycle Administration Console. (For more information, refer to *LiveCycle ES Trust Store Management Help*.)

The CRL distribution point describes where you can download the CRL that corresponds to a particular CER or PFX file.

The following file types are supported:

Certificates: DER-encoded X509v3 and base64 -encoded certificate (.cer) files. Certificates verifying the trust.xml file can be either DER-encoded or base64-encoded.

Credentials: RSA and DSA credentials up to 4096 bits in standard PKCS12 format (.pfx and .p12 files)

CRLs: Base64 and DER encoded .crl files

Maintaining the security of private keys (credentials) is critical to ensuring the stability of sensitive information. A physical storage device (often called a *Hardware Security Module* [HSM]) typically provides the maximum level of security for private keys. If you do not use a physical device, it is important to store highly sensitive private keys and certificates in encrypted files in a safe place.

Digital Signatures ES supports the industry-standard PKCS #11 interface to communicate with HSMs. An HSM vendor can provide the resources and tools you need to install and configure an HSM storage system.

3 | Creating the LiveCycle ES Database

This section describes how to set up the database for use with LiveCycle ES.

The database will contain these elements:

- LiveCycle ES solution component services
- LiveCycle ES run-time configurations
- LiveCycle ES process data
- Customer process definitions and templates
- Application server managed data (for example, JMS tables)

Before creating the database, you must ensure that you have read the preinstallation requirements and that you have the required software installed. (See "System requirements" on page 11.)

If this is the first installation of LiveCycle ES, you must create an empty database. All of the tables required to support LiveCycle ES are created when you run the Initialize LiveCycle ES Database task using LiveCycle Configuration Manager. See the *Installing and Deploying LiveCycle ES* document for your application server.

For information about creating the Business Activity Monitoring ES (BAM) metadata database for Adobe LiveCycle Process Management ES, see the *Installing and Deploying LiveCycle ES* document for your application server.

Database configuration requirements

This section describes special tables, indexes, and other properties required in the LiveCycle ES database that are not configured by LiveCycle Configuration Manager.

Creating a MySQL database

Note: The information contained in this document is meant for users installing MySQL manually and not for a turnkey installation. See *Installing and Deploying LiveCycle ES for JBoss Using Turnkey* at www.adobe.com/go/learn_lc_installTurnkey.

Use the MySQL tools to create a MySQL 5 database for use with LiveCycle ES and a MySQL user account that the application server can use to connect to the database. You also need to modify the MySQL database server configuration. For information about creating the database and user account, see the MySQL documentation. LiveCycle ES supports MySQL 5 with JBoss 4.0.

Tip: To set up a database schema and new users on MySQL using a graphical user interface (GUI), you must install the MySQL Administrator tool. (See the MySQL user documentation.)

When installing the MySQL database, you must specify UTF-8 character encoding.

You will need the following information when you configure the data source on JBoss:

- User name and password of the MySQL user account

- Host name or IP address of database server
- MySQL port number

MySQL user account

The MySQL user account that you create requires these privileges to access the tables in the LiveCycle ES database:

- SELECT
- INSERT
- UPDATE
- DELETE
- CREATE
- DROP
- REFERENCES
- INDEX
- ALTER
- CREATE_TMP_TABLE
- LOCK_TABLES

MySQL database server configuration

To prevent issues from occurring when uploading large files to MySQL, you must set the maximum allowed communication packet size to 25 MB. You can set this property in the MySQL my.ini file (Windows) or my.cnf file (Linux). For more information about the maximum allowed communication packet size property in MySQL, see dev.mysql.com/doc/mysql/en/packet-too-large.html.

If your MySQL installation does not include a my.ini or my.cnf file, you must create one. For information about the location of the file or how to create the file, see the MySQL documentation.

➤ **To customize the MySQL configuration:**

1. Open the my.ini (Windows) or my.cnf (Linux) file in a text editor.

2. Add the following line to the `[mysqld]` section:

   ```
   max_allowed_packet=25M
   ```

3. Save and close the file.

MySQL initial server configuration recommendations

These configuration settings are recommended for MySQL server:

- Key Buffer Size: 64 MB
- Sort Buffer Size: 1 MB
- Buffer Pool Size:100 MB
- Additional Memory Pool Size: 5 MB
- Log Buffer Size: 8 MB

- Max Packet Size: 25 MB

- Read Buffer Size: 1 MB

- Read and Buffer Size: 1 MB

Creating an Oracle database

If you prefer not to use the default database that was created when you installed Oracle 9i or Oracle 10g, create a new database using the Database Configuration Assistant tool. LiveCycle ES supports Oracle 9i and Oracle 10g with WebLogic 9.2, WebSphere 6.1.0.5, and JBoss 4.0.

You must create your Oracle database to support UTF-8 characters. Ensure that the database character set is Unicode (AL32UTF8). You must use the transaction processing template when creating the database. The connection mode for the server must be "Shared Server Mode".

You must also create a new user on the database and assign it the following system privileges:

- CREATE SEQUENCE

- CREATE VIEW

- UNLIMITED TABLESPACE

- CREATE TABLE

- CREATE CLUSTER

- CREATE SESSION

For deployments on Linux, the user name must not exceed 8 characters and, on Windows, it must not exceed 12 characters.

You need the following information when you configure the data source on the application server:

- SID (Service ID)

- User name and password of the Oracle user account

- Host name or IP address of database server

- Oracle port number (default is 1521)

For information about using Oracle 9i or Oracle 10g, see the appropriate user documentation.

Creating a DB2 database

Create a DB2 database by running the script provided in this section. The script is tuned for a system that will use 1 GB of memory for the database. If your system has a smaller amount of memory dedicated for the database, see the DB2 documentation for details about configuring your system settings. LiveCycle ES supports DB2 with WebSphere 6.1.0.5 and JBoss 4.0.

You must also create a user with SYSADM and DBADM privileges that can be used when configuring the data source on the application server. For information about creating a user, see the DB2 documentation. For deployments on Linux and UNIX, the user name must not exceed 8 characters and, on Windows, it must not exceed 12 characters.

You need the following information when you configure the data source on the application server:

- Database name
- User name and password of the DB2 user account
- Host name or IP address of database server
- DB2 port number

➤ **To create a DB2 database:**

1. On the computer that hosts DB2, create a new text file that includes the following DB2 script:

   ```
   create database dbname using codeset utf-8 territory default;
   connect to dbname;
   CREATE BUFFERPOOL "BP8K" SIZE 50000 PAGESIZE 8192 NOT EXTENDED STORAGE;
   connect reset;
   connect to dbname;
   CREATE TEMPORARY TABLESPACE DBNAME_TEMP_8K IN DATABASE PARTITION GROUP
   IBMTEMPGROUP PAGESIZE 8192 MANAGED BY SYSTEM USING
   ('DB2_root\DBNAME_TEMP')
   EXTENTSIZE 32 PREFETCHSIZE 16 BUFFERPOOL BP8K;
   CREATE REGULAR TABLESPACE DBNAME_DATA_8K IN DATABASE PARTITION GROUP
   IBMDEFAULTGROUP PAGESIZE 8192 MANAGED BY DATABASE USING
   (FILE'DB2_root\DBNAME_DATA'64000) EXTENTSIZE 16 PREFETCHSIZE 16
   BUFFERPOOL
   BP8K;
   commit work;
   connect reset;
   connect to dbname;
   alter bufferpool ibmdefaultbp immediate size 96000;
   alter bufferpool bp8k immediate size 32000;
   commit work;
   connect reset;
   update db cfg for dbname using dbheap 4000;
   update db cfg for dbname using logbufsz 2048;
   update db cfg for dbname using locklist 2000;
   update db cfg for dbname using chngpgs_thresh 40;
   update db cfg for dbname using logfilsiz 4000;
   deactivate database dbname;
   activate database dbname;
   ```

2. Make the following changes to the script:

   - Replace the instances of *dbname* and *DBNAME* with the name that you want for the LiveCycle ES database.
   - Replace *DB2_root* with the path to the root directory where DB2 is installed.
   - Ensure that no commands include line breaks and each command is terminated by a semicolon (";").
   - Change `9000` in the following line based on your database size:

     ```
     (FILE'DB2_root\DBNAME_DATA'9000)
     ```

     This number specifies the minimum number of pages required to initialize the database. You can also change this number by using the DB2 administration tools after initializing the database.

3. Save the text file in a location that DB2 Command Line Processor can access.

4. Open a DB2 command prompt and type the following command to run the script:

```
db2 -tf <path_to_script_file>/<script_file_name>
```

➤ **To add a user to the new database:**

1. Log in to the IBM Control Center.

2. Click *[database_name]* > **User and Group Objects** > **DB Users** and then select **Add User**.

3. Select the user you want to use.

4. Under Authorities, select **Connect to database**, **Create tables**, and **Database administration authority**.

5. Click **OK**.

Creating a SQL Server database

You can create a SQL Server database that LiveCycle ES will use to store run-time and configuration data. For information about creating a SQL Server database, refer to the SQL Server documentation. LiveCycle ES supports SQL Server 2005 SP1 with JBoss 4.0, BEA WebLogic 9.2, and IBM WebSphere 6.1.0.5.

Create a SQL Server database and create a user with DB_OWNER privileges that can be used when configuring the data source on the application server. For information about creating the database and user, see the SQL Server documentation.

You need the following information when you configure the data source on the application server:

- Database name
- User name and password of the SQL Server user account
- Host name or IP address of database server
- SQL Server port number

The SQL Server database can be configured with the Windows or SQL Server authentication types. For LiveCycle ES, the authentication type must be set to SQL Server.

Install the SQL Server database driver:

- If you have not done so already, download the SQL Server 2005 JDBC Driver 1.1 from the following location on the Microsoft website:

  www.microsoft.com/downloads/details.aspx?familyid=6d483869-816a-44cb-9787-a866235efc7c&displaylang=en

  Follow the instructions on the website for downloading and installing the driver. Take note of the directory location where you install the driver on your system.

Setting up SQL Server for LiveCycle ES

Before you create the LiveCycle ES database, optimize SQL Server by changing these settings.

Increasing memory

The default SQL Server settings do not aggressively allocate memory. This situation significantly impacts performance on most deployments of a SQL Server database.

➤ **To increase the memory for SQL Server:**

1. Using Microsoft SQL Server Management Studio, connect to the database server where you want to host the LiveCycle ES database.

2. Right-click the database connection and select **Properties**.

3. Select the **Memory** page and enter a size in the **Minimum Server Memory (in MB)** box that is equal to the size of the free memory you have on the server.

Setting processor priority

On dedicated database servers (recommended for production installations of LiveCycle ES), the SQL Server process is configured to prevent over-consumption of the system CPU resources.

➤ **To set the processor priority:**

1. Using Microsoft SQL Server Management Studio, connect to the database server where you want to host the LiveCycle ES database.

2. Right-click the database connection and select **Properties**.

3. Select the **Processors** page and select **Boost SQL Server Priority**.

Increasing the recovery interval

This setting specifies the amount of time the deployment waits for recovery after a crash. The SQL Server default setting is 1 minute. Increasing this setting to a larger value improves performance because it allows the server to be more relaxed in writing changes from the database log to the database files. This setting does not compromise the transactional behavior; however, it does affect the size of the log file that is replayed on startup.

➤ **To increase the recovery interval:**

1. Using Microsoft SQL Server Management Studio, connect to the database server where you want to host the LiveCycle ES database.

2. Right-click the database connection and select **Properties**.

3. Select the **Database Settings** page and type 5 in the **Recovery Interval (Minutes)** box.

Note: After you change these settings, restart the SQL Server database to ensure that the settings take effect.

Integrated Security

If you are using SQL Server 2005 integrated security, you must set your SQL Server 2005 database to Mixed Mode for WebSphere or WebLogic. For JBoss (Windows) you can set your SQL Server 2005 database to either Mixed Mode or Windows Authentication Mode. If you set your SQL Server 2005 database to Windows Authentication Mode for JBoss (Windows), you must follow the steps in "Configuring Integrated Security on Windows" on page 49.

Sizing your SQL Server database

The default database sizes that SQL Server provides are too small for LiveCycle ES. Even if the database is set to auto-grow, unintended consequences can occur, such as reduced performance when the database grows or the size growth begins to fragment the disk. Because of these effects, it is best to preallocate the database size at creation to reflect your deployment requirements:

Medium size deployments: Environments where the LDAP directory has around 100,000 users and around 10,000 groups. Set Database Data Initial Size to 1 GB, and set autogrowth to 250 MB.

Large size deployments: Environments where the LDAP directory has around 350,000 users and more than 10,000 groups. Set Database Data Initial Size to 2 GB, and set autogrowth to 1 GB.

Note: Database growth is always restricted to a certain size. Administrators should monitor the resource usage of the LiveCycle ES database to ensure that it does not run out of restricted space or the space available on the disks where the database resides.

Creating the LiveCycle ES database

After you have optimized the SQL Server, you are ready to create the LiveCycle ES database.

➤ **To create the LiveCycle ES database:**

1. Using Microsoft SQL Server Management Studio, click **Server**, and then right-click **Database** and select **New Database.**

2. In the New Database dialog box, enter the database name of your choice, *database_name*.

3. Ensure that **Use Full-Text Indexing** is deselected.

4. In the **Database Data Initial Size MB** box, enter the appropriate value:
   - For small development or small production system: 200 MB.
   - For larger systems: see "Sizing your SQL Server database" on page 27.

5. In the **Database Data Autogrowth** box, enter 50%.

6. In the **Database Log Initial Size** box, enter the appropriate value:
   - For small development or small production system: 20MB
   - For larger systems: see "Sizing your SQL Server database" on page 27.

7. In the **Database Log Autogrowth** box, enter 50%.

Creating the LiveCycle ES user in SQL Server

In the following procedure, *database_name* represents the name you specified when you created your database, and *database_username* represents the name you specify for the new user.

➤ **To create the LiveCycle ES user:**

1. Using Microsoft SQL Server Management Studio, connect to the database server where you created the LiveCycle ES database.

2. Click **Server** > **Security**, and then right-click **Logins** and select **New Login**.

3. In the New Login dialog box, enter the login name *database_username*.

4. Select **SQL Server Authentication** and type a new password.

5. Deselect **Enforce Password Expiration**. **User must change password on next login** is also deselected.

6. Leave the default database as **Master**, and click **OK**.

7. Click **Server** > **Databases** > *database_name* > **Security**, and then right-click **Schemas** and select **New Schema**.

8. In the **Schema Name** box, type *database_username*, and then click **OK**.

9. Click **Server** > **Databases** > *database_name* > **Security**, right-click **Users**, and then select **New User**.

10. In the New User dialog box, enter the login name and user name *database_username*.

11. Set the default schema to *database_username*.

    Note: The schema name should be the same as the database_username.

12. In the Schemas Owned By This User area, select the schema ***database_username.***

13. In the Database Role Membership area, select **db_owner**.

Associating the LiveCycle ES user with the database

After you have created the LiveCycle ES user, associate it with the LiveCycle ES database.

➤ **To associate the LiveCycle ES user with the database:**

1. Click **Security** > **Logins**, and then right-click ***database_username*** and select **Properties**.

2. In the Login Properties, in the General page, set the user's default database to ***database_name***.

3. Select the **User Mapping** page. In the Users Mapped To This Login section, verify that ***database_name*** is selected, **User** is set to **dbo**, and **Default Schema** is set to **dbo**.

4. Ensure that ***database_name*** is selected in the Users Mapped To This Login table, and ensure that **db_owner** and **public** are selected in the Database Role Membership For *database_name* table.

Setting the isolation level for the LiveCycle ES database

SQL Server 2005 provides a number of new isolation levels. LiveCycle ES requires a specific isolation level to manage deadlocking when long-running transactions occur at the same time as numerous shorter reads.

➤ **To set the isolation level:**

1. Click **Databases**, and then right-click *database_name* and select **New Query**.

    Note: *database_name* represents the name you specified when you created your database.

2. In the Query panel, type the following text:

    ```
    ALTER DATABASE database_name
    SET READ_COMMITTED_SNAPSHOT ON
    GO
    ```

3. Click **Execute**. A response is displayed in the messages panel.

4 | Configuring a JBoss Application Server

When using a JBoss Application Server, you have three installation options:

● Turnkey

● Adobe preconfigured JBoss

● Manual JBoss configuration (draft content for the Preview Release)

This section covers the use of the JBoss Application Server when you are not using the turnkey installation option. You use the non-turnkey method under these conditions:

● You are using a Linux operating system.

● You are preparing a cluster deployment on Windows or Linux.

● You are using DB2, Oracle, or SQL Server as your database.

● You need advanced control of the JBoss configurations.

LiveCycle ES runs on JBoss on Windows Server 2003 (Enterprise Edition or Standard Edition), Red Hat Linux ES/AS 4.0, and SUSE Linux ES 9 platforms. (See "Third-party infrastructure support" on page 9.)

Note: If you are planning to use the LiveCycle ES turnkey installation for JBoss, you do not need to review this document. All the instructions required for preparing your environment for a turnkey installation are available in *Installing and Deploying LiveCycle ES for JBoss Using Turnkey* at www.adobe.com/go/learn_lc_installTurnkey.

This chapter uses the following conventions.

Name	Default value
[appserver root]	(Windows) C:\jboss
	(Linux) /opt/jboss

Installing the J2SE SDK

You must download and install the Sun J2SE SDK version 1.5.0_11 or later update of version 1.5.0 from www.java.sun.com.

You must create or set the JAVA_HOME environment variable to point to the location where the Java SDK is installed.

➤ **To set the JAVA_HOME environment variable (Windows):**

1. Select **Start** > **Control Panel > System**.

2. Click the **Advanced** tab**.**

3. Click **Environment Variables**.

4. In the System Variables area, click **New.**

5. Enter JAVA_HOME as the variable name, and enter the directory where you installed the Java SDK. This directory is the directory containing the /bin subdirectory. For example, type the following line:
`C:\Program Files\Java\jdk1.5.0_11`

Note: To verify your JAVA_HOME environment variable, open a command prompt and run the following command:

```
%JAVA_HOME%\bin\java -version
```

You should receive a response beginning with Java version `1.5.0_11` (or a later version of 1.5.0).

➤ **To set the JAVA_HOME environment variable (UNIX and Linux):**

● It is recommended that you set the JAVA_HOME variable for Borne and Bash shells as shown in the following example:

```
JAVA_HOME=/usr/java
export JAVA_HOME
```

➤ **To set the PATH environment variable (UNIX and Linux):**

● It is recommended that you set the PATH variable for Borne and Bash as shown in the following example:

```
PATH=$JAVA_HOME/bin:$PATH
export PATH
```

Installing JBoss

Use one of the following two ways to proceed with the deployment of JBoss for a LiveCycle ES installation. Select the option that is most appropriate:

● Use the Adobe preconfigured JBoss software that is delivered as part of the LiveCycle ES media. In most cases, this option is the simplest. Advanced knowledge of JBoss is not required.

● Download the default JBoss software from the Internet. This option should be considered for advanced installations only. Advanced knowledge of JBoss is typically required.

Using Adobe preconfigured JBoss

Adobe downloads the standard JBoss 4.0.3 SP1 archive from the JBoss website and configures it with the options required to run LiveCycle ES in a stand-alone configuration. You can review these configurations in the next section.

The preconfigured JBoss software is located on the installation media or ESD download in the /third_party directory. Copy the JBoss directory and its contents to the location where you intend to install JBoss. This JBoss installation has all of the configuration steps completed as described in the "Modifying JBoss configuration files" and "Configuring MySQL" sections except for the steps detailed in "Configuring MySQL for Adobe preconfigured JBoss" on page 37, which you must complete if you plan to use MySQL. If you do not intend to use a MySQL database, you must complete the remaining database configuration steps.

Manually installing JBoss

If you are manually configuring JBoss for running LiveCycle ES, download and unpackage
JBoss Application Server 4.0.3 SP1.

➤ **To download and unpackage JBoss:**

1. In a browser, go to labs.jboss.com/portal/jbossas/download.

2. Locate the entry for JBoss Application Server 4.0.3 SP1 and click the **Download** link beside it.

3. On the download page, select the appropriate file for your operating system:
   - (Windows) jboss-4.0.3SP1.zip
   - (Linux) jboss-4.0.3SP1.tar.gz

4. Extract the file to the directory of your choice on your server.

Starting and stopping JBoss

Several procedures in this chapter require you to stop and start the instance of JBoss where you want to
deploy the product.

Note: These procedures apply to both the Adobe preconfigured JBoss and the manually installed
JBosssoftware.

➤ **To start JBoss:**

1. From a command prompt, navigate to *[appserver root]*/bin.

2. Start the application server by typing the following command:
   - (Windows) `run.bat -c all`
   - (Linux) `./run.sh -c all`

➤ **To stop JBoss:**

1. From a command prompt, navigate to *[appserver root]*/bin.

2. Stop the application server by typing the following command:
   - (Windows) `shutdown.bat -S`
   - (Linux) `./shutdown.sh -S`

Modifying the JBoss configuration files

Note: This procedure is required only if you are not using the Adobe preconfigured JBoss software
provided on the installation media.

Modify the following areas on the instance of JBoss that hosts LiveCycle ES:
   - Update the xalan version.
   - Update the jacorb.properties file.
   - Modify the EAR and WAR file class-loading isolation.

- Remove the JavaServer™ Faces (JSF) libraries.
- Disable cluster service (for single node installations only).
- Remove the tests directory.
- Create the security policy file.
- Modify the run.bat file (Windows only).
- Modify the run.sh file (Linux only).
- Modify run.conf.
- Modify log4J.xml.
- Remove the JMX Console and Web Console.
- Create the Java Message Service (JMS) topics and queues.

➤ **To update the xalan version:**

1. Download the xalan-j 2.7.0 download package from the following website:

   apache.seekmeup.com/xml/xalan-j/

2. Download the required file:
   - (Windows) xalan-j_2_7_0-bin.zip
   - (Linux) xalan-j_2_7_0-bin.tar.gz

3. Extract the xalan.jar file and the serializer.jar file from the downloaded archive file.

4. From the *[appserver root]*/lib/endorsed directory, delete the old xalan.jar file.

5. Copy the new xalan.jar file and the serializer.jar files to the *[appserver root]*/lib/endorsed directory.

➤ **To update the jacorb.properties file:**

1. Open the *[appserver root]*/server/all/conf/jacorb.properties file in a text editor.

2. Locate the `jacorb.poa.thread_pool_max` setting and change the value to `16`.

➤ **To modify WAR file class-loading isolation:**

1. Open the *[appserver root]*/server/all/deploy/jbossweb-tomcat55.sar/META-INF/jboss-service.xml file in an editor.

2. Locate `<attribute name="Java2ClassLoadingCompliance">` and change the value to `true`.

3. Locate `<attribute name="UseJBossWebLoader">` and ensure that the value is `false`.

4. Save and close the file.

➤ **To modify EAR file class-loading isolation:**

1. Open the *[appserver root]*/server/all/conf/jboss-service.xml file in an editor.

2. Locate `<attribute name="CallByValue">` and change the value to `true`.

3. Save and close the file.

4. Open the *[appserver root]*/server/all/deploy/ear-deployer.xml file in an editor.

5. Locate `<attribute name="Isolated">` and change the value to `true`.

6. Locate `<attribute name="CallByValue">` and change the value to `true`.

7. Save and close the file.

➤ **To remove the JSF libraries:**

● Navigate to the *[appserver root]*/server/all/deploy/jbossweb-tomcat55.sar directory and delete the jsf-libs subdirectory and all its files.

➤ **To disable the cluster service:**

1. Open the *[appserver root]*/server/all/deploy/cluster-service.xml file in an editor, and locate and modify the following attribute values:

   ● `mcast_port="34374"`
   ● `ip_ttl="1"`

2. Save and close the file.

3. Open the *[appserver root]*/server/all/deploy/tc5-cluster-service.xml file in an editor.

4. Locate `ip_mcast` and change it to `false`.

5. Save and close the file.

6. Open the *[appserver root]*/server/all/deploy/jms/hajndi-jms-ds.xml file in an editor.

7. Locate `jnp.disableDiscovery` and change the value to `true`.

8. Save and close the file.

➤ **To remove the tests directory:**

● Delete the *[appserver root]*/docs/tests directory.

➤ **To create the security policy file:**

1. In a text editor, open a new document and add the following content:

   ```
   grant {
      // Allow everything for now
      permission java.security.AllPermission;
   };
   ```

2. Save the file as *[appserver root]*/server/all/conf/server.policy.

➤ **To modify the run.bat file (Windows only):**

1. Open the *[appserver root]*/bin/run.bat file in an editor.

2. Locate `rem Sun JVM memory allocation pool parameters` and insert the following text immediately before that string:

   ```
   rem for Java 2 security
   set JAVA_OPTS=%JAVA_OPTS% -Djava.security.manager
   set JAVA_OPTS=%JAVA_OPTS% -Djava.security.policy=../server/all/conf/
     server.policy
   set JAVA_OPTS=%JAVA_OPTS% -Djava.security.debug=failure
   ```

3. Modify the line immediately following `rem Sun JVM memory allocation pool parameters` to read as follows:

   ```
   set JAVA_OPTS=%JAVA_OPTS% -XX:MaxPermSize=256m -Xms256m -Xmx1024m
   -Djava.net.preferIPv4Stack=true
   ```

4. Modify the line immediately following `rem Setup JBoss specific properties` to set encoding to UTF-8:

   ```
   set JAVA_OPTS=%JAVA_OPTS% -Dadobeidp.serverName=server1
   -Dfile.encoding=utf8 -Dprogram.name=%PROGNAME%
   ```

5. Save and close the file.

➤ **To modify the run.sh file (Linux only):**

1. Open the *[appserver root]*/bin/run.sh file in an editor.

2. Modify the line immediately following `Setup JBoss sepecific properties` to set encoding to UTF-8 and security manager:

   ```
   JAVA_OPTS="$JAVA_OPTS -Dadobeidp.serverName=server1 -Dfile.encoding=utf8
   -Dprogram.name=$PROGNAME -Djava.security.manager
   -Djava.security.policy=../server/all/conf/server.policy
   -Djava.security.debug=failure"
   ```

Note: Ensure that this entry appears as a single line in the run.sh file.

3. Save and close the file.

➤ **To modify the run.conf file:**

1. Open the *[appserver root]*/bin/run.conf file in an editor.

2. Locate `JAVA_OPTS="-server -Xms128m -Xmx128m"` and modify it to read as follows:

   ```
   JAVA_OPTS="-server -XX:MaxPermSize=256m -Xms256m -Xmx1024m"
   ```

Note: If you have Linux, you must also add the `"-Djava.net.preferIPv4Stack=true"` flag to remove the warning "[HANamingService] Failed to start AutomaticDiscovery".

3. Save and close the file.

➤ **To modify the log4j.xml file to increase the logging level from DEBUG to INFO:**

1. Open the *[appserver root]*/server/all/conf/log4j.xml file in an editor.

2. Locate the following text in the `FILE appender` section and add the line that appears in bold:

```
<appender name="FILE" class="org.jboss.logging.appender.DailyRollingFileAppe
nder">
   <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
   <param name="File" value="${jboss.server.home.dir}/log/cluster.log"/>
   <param name="Threshold" value="INFO"/>
   <param name="Append" value="false"/>
```

3. Locate the following text in the `CONSOLE appender` section:

```
<appender name="CONSOLE" class="org.apache.log4j.ConsoleAppender">
   <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
   <param name="Target" value="System.out"/>
   <param name="Threshold" value="INFO"/>
```

Modify the last line to read as follows:

```
<param name="Threshold" value="WARN"/>
```

4. Add the com.adobe category, and then find the `Limit categories` section and add the following lines:

```
<category name="com.adobe">
   <priority value="INFO"/>
</category>
```

5. Add the following category under the `com.adobe` category:

```
<category name="org.apache.xml.security.signature.Reference">
   <priority value="WARN"/>
</category>
```

6. Save and close the file.

➤ **To remove the JMX Console and Web Console:**

● To remove the default insecure web pages from JBoss, delete the following directories:

```
[appserver root]/server/all/deploy/jmx-console.war
[appserver root]/server/all/deploy/management
```

➤ **To create JMS topics and queues:**

1. Create a new file called adobe-service.xml and store it in the *[appserver root]*/server/all/deploy/jms directory.

2. Add the following content to the file to define the topics and queues:

```
<?xml version="1.0" encoding="UTF-8"?>
<server>
   <mbean code="org.jboss.mq.server.jmx.Queue"
name="jboss.mq.destination:service=Queue,name=adobe_PECommandQueue">
   <depends optional-attributename="
DestinationManager">jboss.mq:service=DestinationManager</depends>
   </mbean>
   <mbean code="org.jboss.mq.server.jmx.Queue"
```

```
name="jboss.mq.destination:service=Queue,name=adobe_PEDCommandQueue">
    <depends optional-attributename="
DestinationManager">jboss.mq:service=DestinationManager</depends>
    </mbean>
    <mbean code="org.jboss.mq.server.jmx.Queue"
name="jboss.mq.destination:service=Queue,name=adobe_PEInteractionQueue">
    <depends optional-attributename="
DestinationManager">jboss.mq:service=DestinationManager</depends>
    </mbean>
    <mbean code="org.jboss.mq.server.jmx.Queue"
name="jboss.mq.destination:service=Queue,name=adobe_JobManagerQueue">
    <depends optional-attributename="
DestinationManager">jboss.mq:service=DestinationManager</depends>
    </mbean>
    <mbean code="org.jboss.mq.server.jmx.Topic"
name="jboss.mq.destination:service=Queue,name=adobe_TaskEventTopic">
    <depends optional-attributename="
DestinationManager">jboss.mq:service=DestinationManager</depends>
    </mbean>
</server>
```

3. Save and close the file.

Configuring the LiveCycle ES database connectivity

To configure the LiveCycle ES database connectivity, you must complete the following tasks:

● Configure the LiveCycle ES data source.

● Configure JBoss to use your database as the default data source.

You must install database drivers to the installation directories of the application server. Drivers are required to enable LiveCycle Configuration Manager and the application server to connect to the LiveCycle ES database. You must install the drivers for the type of database that you use for the database.

You must configure the data source to connect to the database. For JBoss, you can configure a MySQL, SQL Server, Oracle, or DB2 data source.

Configuring MySQL for Adobe preconfigured JBoss

To enable Adobe preconfigured JBoss to connect to the MySQL database that stores LiveCycle ES data, you must complete these tasks:

● Obtain and copy the MySQL JDBC driver to the instance of JBoss where you will deploy LiveCycle ES.

● Create a data source file and deploy it to the instance of JBoss where you will deploy LiveCycle ES.

● Configure JBoss to use MySQL for JMS.

If you are using a manually installed JBoss, see .

Configuring the MySQL data source

Before configuring the MySQL data source, you must have already created the database on MySQL. (See "Creating a MySQL database" on page 21.)

➤ **To install the MySQL database driver:**

- Copy the mysql-connector-java-3.1.12-bin.jar driver file from the *[DVD_root]*/third_party/jdbc directory on the installation DVD to the *[appserver root]*/server/all/lib directory.

➤ **To create the MySQL data source file:**

1. Copy the adobe-ds-jboss-mysql.xml file from the *[DVD_root]*/third_party/datasources directory on the installation DVD to the *[appserver root]*/server/all/deploy directory.

2. Open the adobe-ds-jboss-mysql.xml file in a text editor and locate this line:

   ```
   <connection-url>jdbc:mysql://localhost:3306/adobe</connection-url>
   ```

3. Replace the following text with values that are specific to your database:
   - **`localhost`**: The name, IP address, or fully-qualified path of the computer that hosts the database. The default is `localhost`.
   - **`3306`**: The port used to access the database. The default port is `3306`.
   - **`adobe`**: The name of the database that stores the LiveCycle ES data. You will need to update the default value, `adobe`, with your database name.
   - **`database_username`** and **`password`**: The user name and password that the application server uses to access the database. You will need to update the default values, `adobe` and `adobe`, with the credentials for your database.

4. Complete step 3 for the `IDP_DS`, `EDC_DS` and `com.celequest.metadata.metaDatasource` elements.

   Note: The `com.celequest.metadata.metaDatasource` element is required only if you are using Business Activity Monitoring ES

5. Save the file as adobe-ds.xml in the *[appserver root]*/server/all/deploy directory.

6. Restart JBoss.

Setting MySQL as the JMS data source

If you are running LiveCycle ES with a MySQL database, you must set MySQL as the default data source for JBoss. (JBoss is configured by default to use Hypersonic as the data source.)

This procedure assumes that the MySQL JDBC driver is installed in the *[appserver root]*/server/all/lib directory.

➤ **To set MySQL as the JMS data source:**

1. Open the *[appserver root]*/server/all/deploy/mysql-ds.xml file in a text editor and modify the <local-tx-datasource> element with your MySQL connection settings:

   ```
   <jndi-name>DefaultDS</jndi-name>
   ```

```
<connection-url>jdbc:mysql://yourHostname:yourPort/
   yourDatabaseName/</connection-url>
<driver-class>com.mysql.jdbc.Driver</driver-class>
<user-name>yourUserName</user-name>
<password>yourPassword</password>
```

2. Replace the bold text with values that are specific to your database:

   yourHostname, **yourPort**, **yourDatabaseName**, **youruserName**, and **yourPassword**: The database values the application server uses to access the database.

3. Save and close the file.

4. Open the *[appserver root]*/server/all/conf/login-config.xml file in a text editor and add the following code within the `<policy>` element:

```
<application-policy name="MySqlDbRealm">
   <authentication>
      <login-module
      code="org.jboss.resource.security.ConfiguredIdentityLoginModule" flag
      = "required">
      <module-option name="principal">database_name</module-option>
      <module-option name="userName">database_username</module-option>
      <module-option name="password">database_password</module-option>
      <module-option
      name="managedConnectionFactoryName">jboss.jca:service=LocalTxCM,
      name=Default DS </module-option>
      </login-module>
   </authentication>
</application-policy>
```

5. Replace the bold text with values that are specific to your database:

   database_name, **database_username**, and **database_password**: The user name and password that the application server uses to access the database.

6. Save and close the file.

7. Restart JBoss.

Configuring MySQL for manually installed JBoss

To enable JBoss to connect to the MySQL database that stores LiveCycle ES data, you must complete these tasks.

- Obtain and copy the MySQL JDBC driver to the instance of JBoss where you will deploy LiveCycle ES.

- Create a data source file and deploy it to the instance of JBoss where you will deploy LiveCycle ES.

- Configure JBoss to use MySQL for JMS.

If you are using Adobe preconfigured JBoss, see .

Configuring the MySQL data source

Before configuring the MySQL data source, you must have already created the database on MySQL. (See "Creating a MySQL database" on page 21.)

➤ **To install the MySQL database driver:**

● Copy the mysql-connector-java-3.1.12-bin.jar driver file from the *[DVD_root]*/third_party/jdbc directory on the installation DVD to the *[appserver root]*/server/all/lib directory.

➤ **To create the MySQL data source file:**

1. Copy the adobe-ds-jboss-mysql.xml file from the *[DVD_root]*/third_party/datasources directory on the installation DVD to the *[appserver root]*/server/all/deploy directory.

2. Open the adobe-ds-jboss-mysql.xml file in a text editor and locate this line:

   `<connection-url>jdbc:mysql://`***localhost:3306***`/`***adobe***`</connection-url>`

3. Replace the following text with values that are specific to your database:
   ● ***localhost***: The name, IP address, or fully-qualified path of the computer that hosts the database. The default is `localhost`.
   ● ***3306***: The port used to access the database. The default port is `3306`.
   ● ***adobe***: The name of the database that stores the LiveCycle ES data. You will need to update the default value, `adobe`, with your database name.
   ● ***database_username*** and ***password***: The user name and password that the application server uses to access the database. You will need to update the default values, `adobe` and `adobe`, with the credentials for your database.

4. Complete step 3 for the `IDP_DS`, `EDC_DS` and `com.celequest.metadata.metaDatasource` elements.

   Note: The `com.celequest.metadata.metaDatasource` element is required only if you are using Business Activity Monitoring ES

5. Save the file as adobe-ds.xml in the *[appserver root]*/server/all/deploy directory.

6. Restart JBoss.

Setting MySQL as the JMS data source

If you are running LiveCycle ES with a MySQL database, you must set MySQL to be the default data source for JBoss. (JBoss is configured by default to use Hypersonic for the data source.)

This procedure assumes that the MySQL JDBC driver is installed in the *[appserver root]*/server/all/lib directory.

➤ **To set MySQL as the JMS data source:**

1. Navigate to the *[appserver root]*/server/all/deploy directory and delete the hsqldb-ds.xml file.

2. Navigate to the *[appserver root]*/docs/examples/jca directory and copy the mysql-ds.xml file to the *[appserver root]*/server/all/deploy directory.

3. Open the *[appserver root]*/server/all/deploy/mysql-ds.xml file in a text editor and modify the `<local-tx-datasource>` element with your MySQL connection settings:

```
<jndi-name>MySqlDS</jndi-name>
<connection-url>jdbc:mysql://yourHostname:yourPort/
yourDatabaseName</connection-url>
<driver-class>com.mysql.jdbc.Driver</driver-class>
<user-name>yourUserName</user-name>
<password>yourPassword</password>
```

4. Replace the bold text with values that are specific to your database:

   - **MySqlDS** must be changed to `DefaultDS`.

   - **yourHostname**, **yourPort**, **yourDatabaseName**, **youruserName**, and **yourPassword** are the database values the application server uses to access the database.

5. Save and close the file.

6. Open the *[appserver root]*/server/all/conf/standardjaws.xml file and change the `<type-mapping>` element to MySQL:

```
<type-mapping>mySQL</type-mapping>
```

7. Save and close the file.

8. Open the standardjbosscmp-jdbc.xml file and change the following elements (not necessarily contiguous):

```
<datasource>java:/DefaultDS</datasource>
<datasource-mapping>mySQL</datasource-mapping>
<fk-constraint>true</fk-constraint>
```

9. Remove the text: `<!—optional since 4.0 and ..`" from the `<datasource-mapping>` tag to uncomment it.

10. Save and close the file.

11. Open the login-config.xml file in a text editor and add the following text within the `<policy>` element:

```
<application-policy name="MySqlDbRealm">
   <authentication>
     <login-module
     code="org.jboss.resource.security.ConfiguredIdentityLoginModule" flag
     = "required">
     <module-option name="principal">database_name</module-option>
     <module-option name="userName">database_username</module-option>
     <module-option name="password">database_password</module-option>
     <module-option
     name="managedConnectionFactoryName">jboss.jca:service=LocalTxCM,
     name=DefaultDS </module-option>
     </login-module>
   </authentication>
</application-policy>
```

12. Replace the bold text with values that are specific to your database:

   database_name, **database_username**, and **database_password**: The user name and password that the application server uses to access the database.

13. Save and close the file.

14. Replace the hsql-jdbc2-service.xml file in the *[appserver root]*/server/all/deploy-hasingleton/jms directory with the mysql-jdbc2-service.xml file from the *[appserver root]*/docs/examples/jms directory.

15. Open the *[appserver root]*/server/all/deploy-hasingleton/jms/mysql-jdbc2-service.xml file and change `MySqlDS` to `DefaultDS`:

```
<mbean code="org.jboss.mq.pm.jdbc2.PersistenceManager"
name="jboss.mq:service=PersistenceManager">
<depends optional-attributename="ConnectionManager">
jboss.jca:service=DataSourceBinding,name=DefaultDS
</depends>
<attribute name="SqlProperties">
...
```

16. In the same file, mysql-jdbc2-service.xml, modify the `SELECT_MAX_TX` line to the following text:

```
SELECT_MAX_TX = SELECT MAX(TXID) FROM JMS_MESSAGES
```

17. Rename the hsqldb-jdbc-state-service.xml file to mysql-jdbc-state-service.xml.

18. Restart JBoss.

Configuring Oracle for Adobe preconfigured JBoss

To enable JBoss to connect to the Oracle database that stores LiveCycle ES data, you must complete the following tasks if you are manually deploying LiveCycle ES:

● Obtain and copy the Oracle JDBC driver to the instance of JBoss where you will deploy LiveCycle ES.

● Create a data source file and deploy it to the instance of JBoss where you will deploy LiveCycle ES.

● Configure JBoss to use Oracle for JMS.

Configuring the Oracle data source

Before configuring the Oracle data source, you must have already created the database on Oracle. (See "Creating an Oracle database" on page 23.)

➤ **To install the Oracle 10g database driver:**

1. You can access the Oracle 10g driver file from the *[LiveCycleES root]*/lib/db/oracle directory. You can also download the Oracle 10g Release 2 (10.2.0.2 Thin) driver from this location:

   www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html

2. Copy the ojdbc14.jar driver file to the *[appserver root]*/server/all/lib directory.

➤ **To create the Oracle data source file:**

1. Copy the adobe-ds-jboss-oracle.xml file from the *[DVD_root]*/third_party/datasources directory on the installation DVD to the *[appserver root]*/server/all/deploy directory.

2. Open the adobe-ds-jboss-oracle.xml file in a text editor and locate this line:

```
<connection-url>jdbc:oracle:thin:@//localhost:1521/adobe</connection-url>
```

3. Replace the following text with values that are specific to your database:

   - `localhost`: The name, IP address, or fully-qualified path of the computer that hosts the database. The default is `localhost`.

   - `1521`: The port used to access the database. The default port is `1521`.

   - `adobe`: The name of the database that stores the LiveCycle ES data. You will need to update the default value, `adobe`, with your database name.

   - `database_username` and `password`: The user name and password that the application server uses to access the database. You will need to update the default values, `adobe` and `adobe`, with the credentials for your database.

4. Complete step <u>3</u> for the `IDP_DS`, `EDC_DS` and `com.celequest.metadata.metaDatasource` elements.

   Note: The `com.celequest.metadata.metaDatasource` element is required only if you are using Business Activity Monitoring ES

5. Save the file as adobe-ds.xml in the *[appserver root]*/server/all/deploy directory.

6. Restart JBoss.

Setting Oracle as the JMS data source

If you are running LiveCycle ES with a Oracle database, you must set Oracle to be the default data source for JBoss. (JBoss is configured by default to use Hypersonic for the data source.)

This procedure assumes that the Oracle JDBC driver is installed in the *[appserver root]*/server/all/lib directory.

➤ **To set Oracle to be the JMS data source:**

1. Navigate to the *[appserver root]*/server/all/deploy directory and delete the mysql-ds.xml file.

2. Navigate to the *[appserver root]*/docs/examples/jca directory and copy the oracle-ds.xml to the *[appserver root]*/server/all/deploy directory.

3. Open the *[appserver root]*/server/all/deploy/oracle-ds.xml file in a text editor and modify the `<local-tx-datasource>` element with your Oracle connection settings:

   ```
   <jndi-name>OracleDS</jndi-name>
   <connection-url>jdbc:oracle:thin:@youroraclehost:1521:yoursid
   </connectionurl>
   <driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
   <user-name>x</user-name>
   <password>y</password>
   ```

4. Replace the bold text with values that are specific to your database:

   - `OracleDS`: Change this value to `DefaultDS`.

   - `youroraclehost`: Replace this value with the host name of your Oracle server.

   - `1521`: If Oracle is not using the default port, replace this value with the appropriate port number.

   - `yoursid`: Replace this value with your Oracle System Identifier.

- Replace **x** with the user name to access the Oracle database and **y** with the password associated with the user. The application server uses these credentials to access the database.

5. Save and close the file.

6. Open the *[appserver root]*/server/all/conf/standardjaws.xml file and change the `<type-mapping>` element to Oracle:

   ```
   <type-mapping>oracle8</type-mapping>
   ```

7. Save and close the file.

8. Open the *[appserver root]*/server/all/conf/standardjbosscmp-jdbc.xml file and change the following elements (not necessarily contiguous):

   ```
   <datasource>java:/DefaultDS</datasource>
   <datasource-mapping>oracle8</datasource-mapping>
   <fk-constraint>true</fk-constraint>
   ```

9. Save and close the file.

10. Open the *[appserver root]*/server/all/conf/login-config.xml file in a text editor and add the following text within the `<policy>` element:

    ```
    <application-policy name = "OracleDbRealm">
       <authentication>
          <login-module code =
          "org.jboss.resource.security.ConfiguredIdentityLoginModule" flag =
          "required">
          <module-option name = "principal">database_name </module-option>
          <module-option name = "userName">database_username</module-option>
          <module-option name = "password">password</module-option>
          <module-option name ="managedConnectionFactoryName">
          jboss.jca:service=LocalTxCM,name=DefaultDS</module-option>
          </login-module>
       </authentication>
    </application-policy>
    ```

11. Replace the bold text with values that are specific to your database:

    database_name, database_username and **password**: The user name and password that the application server uses to access the database.

12. Save and close the file.

13. Delete the mysql-jdbc2-service.xml file from the *[appserver root]*/server/all/deploy-hasingleton/jms directory.

14. Copy the oracle-jdbc2-service.xml file from the *[appserver root]*/docs/examples/jms directory to the *[appserver root]*/server/all/deploy-hasingleton/jms directory.

15. In the oracle-jdbc2-service.xml file in the *[appserver root]*/server/all/deploy-hasingleton/jms directory, change `OracleDS` to `DefaultDS`:

    ```
    <mbean code="org.jboss.mq.pm.jdbc2.PersistenceManager"
    name="jboss.mq:service=PersistenceManager">
    <depends optional-attributename="ConnectionManager">
    jboss.jca:service=DataSourceBinding,name=DefaultDS
    </depends>
    ```

```
<attribute name="SqlProperties">
...
```

16. Save and close the file.

17. Rename the *[appserver root]*/server/all/deploy-hasingleton/jms/mysqldb-jdbc-state-service.xml file to oracle-jdbc-state-service.xml.

18. Restart JBoss.

Configuring Oracle for manually installed JBoss

To enable JBoss to connect to the Oracle database that stores LiveCycle ES data, you must complete the following tasks if you are manually deploying LiveCycle ES:

● Obtain and copy the Oracle JDBC driver to the instance of JBoss where you will deploy LiveCycle ES.

● Create a data source file and deploy it to the instance of JBoss where you will deploy LiveCycle ES.

● Configure JBoss to use Oracle for JMS.

Configuring the Oracle data source

Before configuring the Oracle data source, you must have already created the database on Oracle. (See "Creating an Oracle database" on page 23.)

➤ **To install the Oracle 10g database driver:**

● Copy the ojdbc14.jar driver file from the *[LiveCycleES root]*/lib/db/oracle directory to the *[appserver root]*/server/all/lib directory. In addition, you can download the Oracle 10g Release 2 (10.2.0.2 Thin) driver from this location:

http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html

➤ **To create the Oracle data source file:**

1. Copy the adobe-ds-jboss-oracle.xml file from the *[DVD_root]*/third_party/datasources directory to the *[appserver root]*/server/all/deploy directory.

2. Open the adobe-ds-jboss-oracle.xml file in a text editor and locate this line:

```
<connection-url>jdbc:oracle:thin:@//localhost:1521/adobe</connection-url
```

3. Replace the following text with values that are specific to your database:

● **localhost**: The name, IP address, or fully-qualified path of the computer that hosts the database. The default is localhost.

● **1521**: The port used to access the database. The default port is 1521.

● **adobe**: The name of the database that stores the LiveCycle ES data. You will need to update the default value, adobe, with your database name.

● **database_username** and **password**: The user name and password that the application server uses to access the database. You will need to update the default values, adobe and adobe, with the credentials for your database.

4. Complete step 3 for the IDP_DS, EDC_DS and com.celequest.metadata.metaDatasource elements.

Note: The `com.celequest.metadata.metaDatasource` element is required only if you are using Business Activity Monitoring ES

5. Save the file as adobe-ds.xml in the *[appserver root]*/server/all/deploy directory.

6. Restart JBoss.

Setting Oracle as the JMS data source

If you are running LiveCycle ES with a Oracle database, you must set Oracle to be the default data source for JBoss. (JBoss is configured by default to use Hypersonic for the data source.)

This procedure assumes that the Oracle JDBC driver is installed in the *[appserver root]*/server/all/lib directory.

➤ **To set Oracle to be the JMS data source:**

1. Navigate to the *[appserver root]*/server/all/deploy directory and delete the hsqldb-ds.xml file.

2. Copy the oracle-ds.xml file from the *[appserver root]*/docs/examples/jca directory to the *[appserver root]*/server/all/deploy directory.

3. Open the *[appserver root]*/server/all/deploy/oracle-ds.xml file in a text editor and modify the `<local-tx-datasource>` element with your Oracle connection settings:

```
<jndi-name>OracleDS</jndi-name>
<connection-url>jdbc:oracle:thin:@youroraclehost:1521:yoursid
</connectionurl>
<driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
<user-name>x</user-name>
<password>y</password>
```

4. Replace the bold text with values that are specific to your database:

- *OracleDS*: Change this value to `DefaultDS`.

- *youroraclehost*: Replace this value with the host name of your Oracle server.

- *1521*: If Oracle is not using the default port, replace this value with the appropriate port number.

- *yoursid*: Replace this value with your Oracle System Identifier.

- Replace *x* with the user name to access the Oracle database and *y* with the password associated with the user. The application server uses these credentials to access the database.

5. Save and close the file.

6. Open the *[appserver root]*/server/all/conf/standardjaws.xml file and change the `<type-mapping>` element to Oracle:

```
<type-mapping>oracle8</type-mapping>
```

7. Save and close the file.

8. Open the *[appserver root]*/server/all/conf/standardjbosscmp-jdbc.xml file and change the following elements (not necessarily contiguous):

```
<datasource>java:/DefaultDS</datasource>
<datasource-mapping>oracle8</datasource-mapping>
<fk-constraint>true</fk-constraint>
```

9. Remove the text: `<!—optional since 4.0 and ..` from the `<datasource-mapping>` tag to uncomment it.

10. Save and close the file.

11. Open the *[appserver root]*/server/all/conf/login-config.xml file in a text editor and add the following text within the `<policy>` element:

```
<application-policy name = "OracleDbRealm">
   <authentication>
      <login-module code =
      "org.jboss.resource.security.ConfiguredIdentityLoginModule" flag =
      "required">
      <module-option name = "principal">database_name </module-option>
      <module-option name = "userName">database_username</module-option>
      <module-option name = "password">password</module-option>
      <module-option name ="managedConnectionFactoryName">
      jboss.jca:service=LocalTxCM,name=DefaultDS</module-option>
      </login-module>
   </authentication>
</application-policy>
```

12. Replace the bold text with values that are specific to your database:

   database_name, database_username and ***password***: The user name and password that the application server uses to access the database.

13. Save and close the file.

14. Delete the hsql-jdbc2-service.xml file from the *[appserver root]*/server/all/deploy-hasingleton/jms directory.

15. Copy the oracle-jdbc2-service.xml file from the *[appserver root]*/docs/examples/jms directory to the *[appserver root]*/server/all/deploy-hasingleton/jms directory.

16. Open the *[appserver root]*/server/all/deploy-hasingleton/jms/oracle-jdbc2-service.xml file and change `OracleDS` to `DefaultDS`:

```
<mbean code="org.jboss.mq.pm.jdbc2.PersistenceManager"
name="jboss.mq:service=PersistenceManager">
<depends optional-attributename="ConnectionManager">
jboss.jca:service=DataSourceBinding,name=DefaultDS
</depends>
<attribute name="SqlProperties">
...
```

17. Save and close the file.

18. Rename the hsqldb-jdbc-state-service.xml file located in the *[appserver root]*/server/all/deploy-hasingleton/jms directory to oracle-jdbc-state-service.xml.

19. Restart JBoss.

Configuring SQL Server

To enable JBoss to connect to the SQL Server database that stores LiveCycle ES data, you must complete the following tasks:

- Obtain and copy the SQL Server JDBC driver files to the instance of JBoss where you will deploy LiveCycle ES.

- Create a SQL Server data source file and deploy it to the instance of JBoss where you will deploy LiveCycle ES.

- Configure JBoss to use SQL Server for JMS.

Configuring the SQL Server data source

Before configuring the SQL Server data source, you must have already created the LiveCycle ES database on SQL Server. (See "Creating a SQL Server database" on page 25.)

➤ **To install the SQL database driver:**

1. Obtain the SQL Server 2005 JDBC 1.1 database driver from the Microsoft website.

2. (Windows) Download the *.exe file and run the file, and then extract the files to a temporary directory (referred to as the *[SQL_root]* directory in the remainder of this section).

3. (Linux) Extract the *.tar.gz files to a temporary directory (referred to as the *[SQL_root]* directory in the remainder of this section).

4. Copy the sqljdbc.jar file from the *[SQL_root]*/sqljdbc_1.1/enu directory to the *[appserver root]*/server/all/lib directory.

➤ **To create the SQL Server data source file:**

1. Copy the adobe-ds-jboss-mssql.xml file from the *[DVD_root]*/third_party/datasources directory to the *[appserver root]*/server/all/deploy directory.

2. Open the adobe-ds-jboss-mssql.xml file in a text editor and locate this line:

```
<connection-url>jdbc:sqlserver://localhost:1433;DatabaseName=adobe</connecti
on-url>
```

3. Replace the following text with values that are specific to your database:

   - *localhost*: The name, IP address, or fully-qualified path of the computer that hosts the database. The default is `localhost`.

   - *1433*: The port used to access the database. The default port is `1433`.

   - *adobe*: The name of the database that stores the LiveCycle ES data. You will need to update the default value, `adobe`, with your database name.

   - *database_username* and *password*: The user name and password that the application server uses to access the database. You will need to update the default values, `adobe` and `adobe`, with the credentials for your database.

4. Complete step 3 for the `IDP_DS`, `EDC_DS` and `com.celequest.metadata.metaDatasource` elements.

> **Note:** The `com.celequest.metadata.metaDatasource` element is required only if you are using Business Activity Monitoring ES

5. Save the file as adobe-ds.xml in the *[appserver root]*/server/all/deploy directory.

6. Restart JBoss.

➤ **Configuring Integrated Security on Windows**

This steps provide instructions on how to use integrated security to make a trusted connection with SQL Server.

1. Modify the adobe-ds.xml file, located in *[appserver root]*\server\all\deploy, to add `integratedSecurity=true` to the connection URL. For example:

    ```
    jdbc:sqlserver://<serverhost>:<port>;databaseName=<dbname>;integratedSec
    urity=true.
    ```

2. Add the sqljdbc_auth.dll file to the Windows systems path (C:\Windows) on the computer that is running JBoss. The sqljdbc_auth.dll file is located with the Microsoft SQL JDBC 1.1 driver installation (default is <InstallDir>/sqljdbc_1.1/enu/auth/x86).

3. Modify the JBoss Windows service (JBoss for Adobe LiveCycle ES v8.0) property for *Log On As* from Local System to a good login account. This change is not required if you are running JBoss from the command line.

4. Set SQL Server's Security from Mixed mode to Windows Authentication only.

Configuring JMS with Microsoft SQL Server

This section applies to both Adobe preconfigured JBoss and manually installed JBoss.

Setting SQL Server as the JMS data source

If you are running LiveCycle ES with a SQL Server database, you must set SQL Server to be the default data source for JBoss. (It is configured by default to use Hypersonic for the data source.)

This procedure assumes that the SQL Server JDBC driver is installed in the *[appserver root]*/server/all/lib directory.

➤ **To set SQL Server to be the JMS data source:**

1. Navigate to the *[appserver root]*/server/all/deploy directory and delete the hsqldb-ds.xml file. If you are using the Adobe preconfigured JBoss from the DVD, delete the mysql-ds.xml file.

2. Install the JBossMQ patch file, jbossmq.jar. You must replace the existing jbossmq.jar file in the *[appserver root]*/server/all/lib directory. You can obtain the updated jbossmq.jar file from the thirdparty\jboss\patches directory in the installation media.

3. Navigate to the [appserver root]/docs/examples/jca directory and copy the mssql-ds.xml file to the *[appserver root]*/server/all/deploy directory.

4. Open the mssql-ds.xml file in a text editor and modify the `<local-tx-datasource>` element with your SQL Server connection settings:

    ```
    <jndi-name>DefaultDS</jndi-name>
    <connection-url>jdbc:sqlserver:// your-hostname:
       your-port;DatabaseName=yourdatabase-name</connection-url>
    <driver-class>com.microsoft.sqlserver.jdbc.SQLServerDriver
       </driver-class>
    <user-name>yourUserName</user-name>
    <password>yourPassword</password>
    ```

5. Save and close the file.

6. Open the *[appserver root]*/server/all/conf/standardjaws.xml file and change the <type-mapping> element to SQL Server:

    ```
    <type-mapping>MS SQLSERVER2000</type-mapping>
    ```

7. Open the *[appserver root]*/server/all/conf/standardjbosscmp-jdbc.xml file and change the following elements (not necessarily contiguous):

    ```
    <datasource>java:/DefaultDS</datasource>
    <datasource-mapping>MS SQLSERVER2000</datasource-mapping>
    <fk-constraint>true</fk-constraint>
    ```

8. (Manually installed JBoss only) Remove the comment tags from the `<datasource-mapping>` tag to uncomment it. For example, remove "`<!—optional since 4.0 ....`" and the corresponding --> tag.

9. Save and close the file.

10. Open the *[appserver root]*/server/all/conf/login-config.xml file in a text editor and add the following lines within the <policy> element:

    ```
    <application-policy name = "MSSQLDbRealm">
       <authentication>
       <login-module code =
       "org.jboss.resource.security.ConfiguredIdentityLoginModule"
          flag = "required">
       <module-option name = "principal">database_name</module-option>
       <module-option name = "userName">database_username</module-option>
       <module-option name ="password">password</module-option>
       <module-option name =
       "managedConnectionFactoryName">jboss.jca:service=LocalTxCM,name=
          DefaultDS
       </module-option>
       </login-module>
       </authentication>
    </application-policy>
    ```

11. Replace the bold text with values that are specific to your database:

    database_name, **database_username**, and **password**: The user name and password that the application server uses to access the database.

12. Save and close the file.

13. Delete the *[appserver root]*/server/all/deploy-hasingleton/jms/hssql-jdbc2-service.xml file. Or, If you are using the Adobe preconfigured JBoss from the DVD, delete the mysql-jdbc2-service.xml. Copy the *[appserver root]*/docs/examples/jms/mssql-jdbc2-service.xml file to *[appserver root]*/server/all/deploy-hasingleton/jms.

14. Open the *[appserver root]*/server/all/deployhasingleton/jms/mssql-jdbc2-service.xml file and change the mbean code to reference MSSQLJDBCStateManager and DefaultDS:

```
<mbean code="org.jboss.mq.pm.jdbc2.MSSQLPersistenceManager"
name="jboss.mq:service=PersistenceManager">
<depends optional-attributename="
ConnectionManager">jboss.jca:service=DataSourceBinding,name=DefaultDS
</depends>
<attribute name="SqlProperties">
...
```

15. Rename the *[appserver root]*/server/all/deploy-hasingleton/jms/hsqldb-jdbc-state-service.xml file to mssql-jdbc-state-service.xml. Or, if you are using the Adobe preconfigured JBoss from the DVD, rename the *[appserver root]*/server/all/deploy-hasingleton/jms/mysql-jdbc-state-service.xml file to mssql-jdbc-state-service.xml.

16. Open the *[appserver root]*/server/all/deploy-hasingleton/jms/mssql-jdbc-stateservice.xml file and change the mbean code to reference MSSQLJDBCStateManager:

```
<mbean code="org.jboss.mq.sm.jdbc.MSSQLJDBCStateManager"
name="jboss.mq:service=StateManager">
```

17. Restart JBoss.

Configuring DB2 for Adobe preconfigured JBoss

To enable JBoss to connect to the DB2 database that stores LiveCycle ES data, you must complete the following tasks if you are manually deploying LiveCycle ES:

● Obtain and copy the DB2 JDBC driver to the instance of JBoss where you will deploy LiveCycle ES.

● Create a data source file and deploy it to the instance of JBoss where you will deploy LiveCycle ES.

● Configure JBoss to use DB2 for JMS.

Configuring the DB2 data source

Before configuring the DB2 data source, you must have already created the database on DB2. (See "Creating a DB2 database" on page 23.)

➤ **To install the DB2 database driver:**

● Copy the db2jcc.jar and db2jcc_license_cu.jar files from the adobe/livecycle/lib/db/db2 directory to the *[appserver root]*/server/all/lib directory.

➤ **To obtain the latest db2-jdbc-service.xml file:**

1. Go to labs.jboss.com/projects/download.

2. Click **JBoss Application Server**.

3. Click **Download** by 4.2.0 GA.

4. Select **jboss-4.2.0.GA.zip**.

5. After the file downloads, open jboss-4.2.0.GA.zip and extract the jboss-4.2.0.GA\docs\examples\jms\
    db2-jdbc2-service.xml file to the *[appserver root]*/docs/examples/jms directory.

➤ **To create the DB2 data source file:**

1. Copy the adobe-ds-jboss-db2.xml file from the *[DVD_root]*/third_party/datasources directory to the
    [appserver root]/server/all/deploy directory.

2. Open the adobe-ds-jboss-db2.xml file in a text editor and locate this line:

    ```
    <connection-url>jdbc:db2://localhost:50000/adobe</connection-url>
    ```

3. Replace the following text with values that are specific to your database:

    - `localhost`: The name, IP address, or fully-qualified path of the computer that hosts the database.
      The default is `localhost`.

    - `50000`: The port used to access the database. The default is `50000`.

    - `adobe`: The name of the database that stores the LiveCycle ES data. You will need to update the
      default value, `adobe`, with your database name.

    - `database_username` and `password`: The user name and password that the application server
      uses to access the database. You will need to update the default values, `adobe` and `adobe`, with the
      credentials for your database.

4. Complete step 3 for the `IDP_DS`, `EDC_DS` and `com.celequest.metadata.metaDatasource`
    elements.

    Note: The `com.celequest.metadata.metaDatasource` element is required only if you are using
    Business Activity Monitoring ES

5. Save the file as adobe-ds.xml in the *[appserver root]*/server/all/deploy directory.

6. Restart JBoss.

Setting DB2 as the JMS data source

If you are running LiveCycle ES with a DB2 database, you must set DB2 to be the default JMS data source
for JBoss. (JBoss is configured by default to use Hypersonic for the data source.)

This procedure assumes that the DB2 JDBC driver is installed in the *[appserver root]*/server/all/lib directory.

➤ **To set DB2 to be the JMS data source:**

1. Navigate to the *[appserver root]*/server/all/deploy directory and delete the mysqldb-ds.xml file.

2. Copy the db2-ds.xml file from the *[appserver root]*/docs/examples/jca directory to the *[appserver
    root]*/server/all/deploy directory.

3. Open the *[appserver root]*/server/all/deploy/db2-ds.xml file in a text editor and modify the
   `<local-tx-datasource>` element with your DB2 connection settings:

   ```
   <jndi-name>DB2DS</jndi-name>
   <connection-url>jdbc:db2:your_database</connectionurl>
   <driver-class>COM.ibm.db2.jdbc.app.DB2Driver</driver-class>
   <user-name>x</user-name>
   <password>y</password>
   ```

4. Replace the bold text with values that are specific to your database:

   - *DB2DS*: You must replace with `DefaultDS`.

   - *your_database*: The name of the DB2 database that stores the LiveCycle ES data.

   - *COM.ibm.db2.jdbc.app.DB2Driver*: Replace with `com.ibm.db2.jcc.DB2Driver`.

   - *x* and *y*: The user name and password that the application server uses to access the database.

5. Save and close the file.

6. Open the *[appserver root]*/server/all/conf/standardjaws.xml file and change the `<type-mapping>`
   element to `DB2`:

   ```
   <type-mapping>DB2</type-mapping>
   ```

7. Save and close the file.

8. Open the *[appserver root]*/server/all/conf/standardjbosscmp-jdbc.xml file and change the following
   elements (not necessarily contiguous):

   ```
   <datasource>java:/DefaultDS</datasource>
   <datasource-mapping>DB2</datasource-mapping>
   <fk-constraint>true</fk-constraint>
   ```

9. Open the *[appserver root]*/server/all/conf/login-config.xml file in a text editor and add the following
   text within the `<policy>` element:

   ```
   <application-policy name = "DB2DbRealm">
      <authentication>
        <login-module code =
        "org.jboss.resource.security.ConfiguredIdentityLoginModule" flag =
        "required">
        <module-option name = "principal">database_name </module-option>
        <module-option name = "userName">database_username</module-option>
        <module-option name = "password">password</module-option>
        <module-option name ="managedConnectionFactoryName">
        jboss.jca:service=LocalTxCM,name=DefaultDS
        </module-option>
        </login-module>
      </authentication>
   </application-policy>
   ```

10. Replace the bold text with values that are specific to your database:

    database_name, database_username and *password*: The user name and password that the
    application server uses to access the database.

11. Save and close the file.

12. Delete the mysql-jdbc.service.xml file from *[appserver root]*/server/all/deploy-hasingleton/jms.

13. Copy *[appserver root]*/docs/examples/jms/db2-jdbc2-service.xml file to the
    [appserver root]/server/all/deploy-hasingleton/jms directory.

14. Open the *[appserver root]*/server/all/deploy-hasingleton/jms/db2-jdbc2-service.xml file and change
    `DB2DS` to `DefaultDS`:

    ```
    <mbean code="org.jboss.mq.pm.jdbc2.PersistenceManager"
    name="jboss.mq:service=PersistenceManager">
    <depends optional-attributename="ConnectionManager">
    jboss.jca:service=DataSourceBinding,name=DefaultDS
    </depends>
    <attribute name="SqlProperties">
    ...
    ```

15. Save and close the file.

16. Rename the mysql-jdbc-state-service.xml file, located in the *[appserver root]*/server/all/
    deploy-hasingleton/jms directory, to db2-jdbc-state-service.xml.

17. Restart JBoss.

Configuring DB2 for manually installed JBoss

To enable JBoss to connect to the DB2 database that stores LiveCycle ES data, you must complete the
following tasks if you are manually deploying LiveCycle ES:

● Obtain and copy the DB2 JDBC driver to the instance of JBoss where you will deploy LiveCycle ES.

● Create a data source file and deploy it to the instance of JBoss where you will deploy LiveCycle ES.

● Configure JBoss to use DB2 for JMS.

Configuring the DB2 data source

Before configuring the DB2 data source, you must have already created the database on DB2. (See
"Creating a DB2 database" on page 23.)

➤ **To install the DB2 database driver:**

1. Obtain the db2jcc.jar and db2jcc_license_cu.jar files from the DB2 installation directories on your DB2
   server.

2. Copy the files from the adobe/livecycle/lib/db/db2 directory to the *[appserver root]*/server/all/lib
   directory.

➤ **To obtain the latest db2-jdbc-service.xml file:**

1. Go to labs.jboss.com/projects/download.

2. Click **JBoss Application Server**.

3. Click **Download** by 4.2.0 GA.

4. Select **jboss-4.2.0.GA.zip**.

5. After the file downloads, open jboss-4.2.0.GA.zip and extract the db2-jdbc2-service.xml file to the
   [appserver root]/docs/examples/jms directory.

➤ **To create the DB2 data source file:**

1. Copy the adobe-ds-jboss-db2.xml file from the *[DVD_root]*/third_party/datasources directory to the *[appserver root]*/server/all/deploy directory.

2. Open the adobe-ds-jboss-db2.xml file in a text editor and locate this line:

   ```
   <connection-url>jdbc:db2://localhost:50000/adobe</connection-url>
   ```

3. Replace the following text with values that are specific to your database:

   - **`localhost`**: The name, IP address, or fully-qualified path of the computer that hosts the database. The default is `localhost`.

   - **`50000`**: The port used to access the database. The default is `50000`.

   - **`adobe`**: The name of the database that stores the LiveCycle ES data. You will need to update the default value, `adobe`, with your database name.

   - **`database_username`** and **`password`**: The user name and password that the application server uses to access the database. You will need to update the default values, `adobe` and `adobe`, with the credentials for your database.

4. Complete step <u>3</u> for the `IDP_DS`, `EDC_DS` and `com.celequest.metadata.metaDatasource` elements.

   Note: The `com.celequest.metadata.metaDatasource` element is required only if you are using Business Activity Monitoring ES

5. Save the file as adobe-ds.xml in the *[appserver root]*/server/all/deploy directory.

6. Restart JBoss.

Setting DB2 as the JMS data source

If you are running LiveCycle ES with a DB2 database, you must set DB2 to be the default JMS data source for JBoss. (JBoss is configured by default to use Hypersonic for the data source.)

This procedure assumes that the DB2 JDBC driver is installed in the *[appserver root]*/server/all/lib directory.

➤ **To set DB2 as the JMS data source:**

1. Navigate to the *[appserver root]*/server/all/deploy directory and delete the hsqldb-ds.xml file.

2. Copy the db2-ds.xml file from the *[appserver root]*/docs/examples/jca directory to the *[appserver root]*/server/all/deploy directory.

3. Open the *[appserver root]*/server/all/deploy/the db2-ds.xml file in a text editor and modify the `<local-tx-datasource>` element with your DB2 connection settings:

   ```
   <jndi-name>DB2DS</jndi-name>

   <connection-url>jdbc:db2:database_hostname:50000/your_database</connectionur
   l>
           <driver-class>COM.ibm.db2.jdbc.app.DB2Driver</driver-class>
           <user-name>x</user-name>
           <password>y</password>
   ```

4. Replace the bold text with values that are specific to your database:

   - *DB2DS*: You must replace with `DefaultDS`.

   - *database_hostname*:The hostname of the server hosting the database.

   - *your_database*: The name of the DB2 database that stores the LiveCycle ES data.

   - *COM.ibm.db2.jdbc.app.DB2Driver*: Replace with `com.ibm.db2.jcc.DB2Driver`.

   - *x* and *y*: The user name and password that the application server uses to access the database.

5. Save and close the file.

6. Open the *[appserver root]*/server/all/conf/standardjaws.xml file and change the `<type-mapping>` element to `DB2`:

   ```
   <type-mapping>DB2</type-mapping>
   ```

7. Save and close the file.

8. Open the *[appserver root]*/server/all/conf/standardjbosscmp-jdbc.xml file and change the following elements (not necessarily contiguous):

   ```
   <datasource>java:/DefaultDS</datasource>
   <datasource-mapping>DB2</datasource-mapping>
   <fk-constraint>true</fk-constraint>
   ```

9. Remove the text: `<!—optional since 4.0` and `..` from the `<datasource-mapping>` tag to uncomment it.

10. Open the *[appserver root]*/server/all/conf/login-config.xml file in a text editor and add the following text within the `<policy>` element:

    ```
    <application-policy name = "DB2DbRealm">
       <authentication>
          <login-module code =
          "org.jboss.resource.security.ConfiguredIdentityLoginModule" flag =
          "required">
          <module-option name = "principal">database_name </module-option>
          <module-option name = "userName">database_username</module-option>
          <module-option name = "password">password</module-option>
          <module-option name ="managedConnectionFactoryName">
          jboss.jca:service=LocalTxCM,name=DefaultDS
          </module-option>
          </login-module>
       </authentication>
    </application-policy>
    ```

11. Replace the bold text with values that are specific to your database:

    database_name, database_username and *password*: The user name and password that the application server uses to access the database.

12. Save and close the file.

13. Delete the hssql-jdbc2-service.xml file from the *[appserver root]*/server/all/deploy-hasingleton/jms directory.

14. Copy the db2-jdbc2-service.xml file from the *[appserver root]*/docs/examples/jms directory to the *[appserver root]*/server/all/deploy-hasingleton/jms directory.

15. Open the *[appserver root]*/server/all/deploy-hasingleton/jms/db2-jdbc2-service.xml file and change
   `DB2DS` to `DefaultDS`:

   ```
   <mbean code="org.jboss.mq.pm.jdbc2.PersistenceManager"
   name="jboss.mq:service=PersistenceManager">
   <depends optional-attributename="ConnectionManager">
   jboss.jca:service=DataSourceBinding,name=DefaultDS
   </depends>
   <attribute name="SqlProperties">
   ...
   ```

16. Save and close the file.

17. Rename the hsqldb-jdbc-state-service.xml file, located in the *[appserver root]*/server/all/
   deploy-hasingleton/jms directory, to db2-jdbc-state-service.xml.

18. Restart JBoss.

Next steps

Install LiveCycle ES by following the instructions provided in *Installing and Deploying LiveCycle ES for JBoss*
at www.adobe.com/go/learn_lc_installJBoss.

5 | Configuring a WebLogic Server

This chapter describes how to install and configure the BEA WebLogic Server that will host your LiveCycle ES installation.

This section uses the following conventions:

Name	Default value
[appserver root]	(Windows) C:\bea\weblogic92
	(Linux and UNIX) /opt/bea/weblogic92
BEA_HOME	(Windows) C:\bea
	(Linux and UNIX) /opt/bea
[appserverdomain] By default the domain is called "base_domain".	(Windows) C:\bea\user_projects\domains\base_domain (Linux and UNIX) /opt/bea/user_projects/domains/base_domain

Overview

You will perform the following tasks to install and configure your BEA WebLogic Server 9.2 environment:

1. Obtain WebLogic Server 9.2 and install as per the manufacturer's documentation. (See "Installing WebLogic Server" on page 59).

2. Create and configure the WebLogic Server domain. (See "Configuring anonymous admin look-up" on page 64).

3. Access the WebLogic Server Administration Console.

4. Create a Managed Server. (See "Creating a new WebLogic Managed Server" on page 63).

5. Configure application server and data source connections. (See "Configuring WebLogic Server" on page 64).

LiveCycle ES requirements for WebLogic

There are two primary startup scenarios for WebLogic:

- Using Node Manager and configuring through the Administration Server
- Directly starting the Managed Servers with configurations in the StartManagedWebLogic scripts

To use the automatic configuration features of LiveCycle Configuration Manager (this is the recommended option), you must use the Node Manager and configure through the Administration Server.

If you are directly starting the Managed Servers, you can still use the automatic configuration features of LiveCycle Configuration Manager. However, you must manually enter the LiveCycle Configuration Manager changes into the StartManagedWebLogic scripts. See *Installing and Deploying LiveCycle ES for WebLogic* at www.adobe.com/go/learn_lc_installWebLogic for the appropriate steps.

Installing WebLogic Server

Obtain WebLogic Server 9.2 for your operating system and install it according to the manufacturer's documentation. See the BEA website for more information at http://bea.com/.

Note: If you have WebLogic Server 9.1 installed, uninstall it before installing WebLogic Server 9.2.

In addition to the space required to install LiveCycle ES, your environment variable `TEMP` or `TMP` must point to a valid temporary directory with at least 500 MB plus another 1 GB to unpack image files.

You must install and run WebLogic Server using a user account with computer administrator privileges.

Installing the J2SE SDK

You must upgrade the BEA JRockit® JDK to 1.5.0_06 on Windows and Linux platforms.

➤ **To install the JDK:**

1. You can download the JDK updates for Windows or Linux from this location:

   http://commerce.bea.com/products/weblogicjrockit/5.0/jr_50.jsp

2. Download the specific version for your platform from the following section:

   JRockit 5.0 R26.4 CR302700 (for use with WLS 9.2 MP1)
   Java Development Kit

As part of your WebLogic installation, a Java SDK was installed. The `JAVA_HOME` and `PATH` environment variables must point to the Java SDK where LiveCycle ES is to be deployed.

➤ **To set the JAVA_HOME environment (Windows):**

1. Select **Start** > **Control Panel** > **System**.

2. Click the **Advanced** tab**.**

3. Click **Environment Variables**.

4. In the System Variables area, click **New**.

5. Enter `JAVA_HOME` as the variable name and the directory where you installed Java SDK. This directory is where WebLogic installed the Java SDK containing the /bin subdirectory. For example, type this text:

   ```
   C:\bea\jrockit-jdk1.5.0_06
   ```

Note: To verify your `JAVA_HOME` environment variable, open a command prompt and run the following command:

   ```
   %JAVA_HOME%\bin\java -version
   ```

You should receive a response that begins with `java version "1.5.0"`.

➤ **To set the PATH environment variable (Windows):**

1. Select **Start** > **Control Panel** > **System**.

2. Click the **Advanced** tab.

3. Click **Environment Variables**.

4. In the System Variables area, select the `PATH` variable and click **Edit**. Append the following text to the beginning of the variable value:

   ```
   %JAVA_HOME%\bin;
   ```

➤ **To set the JAVA_HOME environment (UNIX and Linux):**

- It is recommended that you set the `JAVA_HOME` variable for Borne and Bash shells as shown in the following example:

  ```
  JAVA_HOME=/opt/java1.5
  export JAVA_HOME
  ```

Note: The specific path varies based on the installation directory you specified and the operating system you are installing on.

➤ **To set the PATH environment variable (UNIX and Linux):**

- It is recommended that you set the `PATH` variable for Borne and Bash shells as shown in the following example:

  ```
  PATH=$JAVA_HOME/bin:$PATH
  export PATH
  ```

Creating the WebLogic Server domain

To deploy applications on WebLogic Server, you must create a WebLogic Server domain. A domain is the basic administrative unit for WebLogic Server. Here are the two basic types of WebLogic Server domains:

Standalone Server Domain: This type of domain can be used for development or test environments in which a single server instance acts as both Administration Server and Managed Server. This option is easier to manage and allows you to use the *hot deploy* feature in WebLogic on a development environment.

Domain with Managed Servers: A production environment typically consists of an Administration Server with one or more Managed Servers. The Administration Server is used to perform management operations; the applications and resources are deployed to individual Managed Servers.

Note: In production environments, BEA recommends that you deploy applications only on Managed Servers in the domain; the Administration Server should be reserved for management tasks.

➤ **To create a WebLogic domain with Managed Servers:**

1. From a command prompt, start the WebLogic Configuration Wizard by navigating to the *[appserver root]*/common/bin directory and typing the appropriate command:

   ● (Windows) `config.cmd`

   ● (Linux, UNIX) `./config.sh`

2. On the Create Or Extend A Configuration screen, select **Create a new WebLogic domain** and click **Next**.

3. On the Select Domain Source screen, select **Generate a domain configured automatically to support the following BEA products:** and click **Next**.

   Note: By default, **WebLogic Server (Required)** is selected. Ensure that **Workshop for WebLogic Platform** is deselected.

4. Type your WebLogic user name and password, confirm the password by retyping it, and then click **Next**. Record this information because LiveCycle Configuration Manager will prompt you for the WebLogic user name and password.

5. In the WebLogic Domain Startup Mode panel, select **Production Mode**.

   Caution: Adobe does not recommend using Development Mode.

6. In the right panel, select the BEA-supplied JDK according to your operating system and then click **Next**.

   ● (Windows, Linux) **JRockit SDK 1.5._06@ *BEA_HOME*/jrockit90_150_06.** You must select **Other JDK** in the UI and specify the location of the *jrockit-jdk1.5.0_06*.

   ● (Solaris) **Sun SDK 1.5.0_04 @ *BEA_HOME*/jdk150_04**

7. On the Customize Environment And Services Settings screen, make sure that **No** is selected and then click **Next**.

8. On the Create WebLogic Domain screen, either accept the default values or enter the domain name and location as required, and click **Create**.

9. On the Creating Domain screen, when the configuration creation is 100% complete, do the following tasks:

   ● (Windows) Select **Start Admin Server** and click **Done**.

   ● (Linux, UNIX) Click **Done**. Start the server by navigating from a command prompt to the *[appserverdomain]* and typing `./startWebLogic.sh`

10. When prompted, enter the WebLogic user name and password you entered in step 4.

WebLogic Administration Server tasks

This section outlines the steps to correctly configure the Administration Server to deploy LiveCycle ES.

➤ **To set the memory size for managed servers:**

1. To access the WebLogic Administration Console, type `http://`*[host name]*`:`*[Port]*`/console` in the URL line of a web browser, where *[Port]* is the non-secure listening port. By default, this port value is 7001.

2. On the login screen, type your WebLogic user name and password and click **Log In**.

3. Under Change Center, click **Lock & Edit**.

4. Under Domain Structure, expand **Environment** and click **Servers**.

5. In the right pane, under Servers, click the name of the Administration Server.

6. On the **Configuration** tab, click the **Server Start** tab.

7. In the **Arguments** field, type this text:

    (Solaris) `-Xms256m –Xmx1024m –XX:MaxPermSize=256m`

    (All other operating systems) `-Xms256m –Xmx1024m`

8. Click **Save** and then click **Activate Changes**.

➤ **To set the memory size for the managed server:**

1. Move to the following directory: *BEA_HOME\user_projects\domains\[appserverdomain]\bin*

2. Edit the following file in a text editor:

    - (Windows) setDomainEnv.cmd
    - (UNIX) setDomainEnv.sh

3. Search for the line `set MEM_ARGS=-Xms256m -Xmx512m` and change it to `MEM_ARGS=-Xms256m -Xmx1024m`.

4. Search for the following line: `-XX:MaxPermSize=128m` and change it to `-XX:MaxPermSize=256m`.

    Note: There are two locations of -XX:MaxPermSize=256:

    - if "%JAVA_VENDOR%"=="Sun" (...
    - If "%JAVA_VENDOR%"=="HP" (...

Starting Node Manager

You must start the Node Manager before continuing the rest of the sections in this chapter.

➤ **To start Node Manager on Windows:**

1. If the WebLogic Node Manager was installed as a Windows service, open the Windows Services control panel and ensure that the WebLogic Node Manager service is running. For example, the WebLogic Node Manager service will appear similar to the value **BEA Products NodeManager (C_bea_weblogic92)**.

2. If the WebLogic Node Manager was not installed as a Windows service, do these tasks:

    - From a new command prompt, navigate to the *[appserver root]*\server\bin directory.
    - Type the following command:

        ```
        startNodeManager.cmd
        ```

➤ **To start Node Manager on UNIX or Linux:**

1. From a new command prompt, navigate to the *[appserver root]*/server/bin directory.

2. Type the following command:

    ```
    ./startNodeManager.sh
    ```

Creating a new WebLogic Managed Server

You must deploy LiveCycle ES applications on Managed Servers in a domain; the Administration Server should be reserved for management tasks. For information about creating a server domain and about Administration and Managed Servers, see the WebLogic product documentation.

Note: The following procedure highlights the changes required to the default properties. For properties that are not provided, accept the existing settings. For more information about these screens, see *WebLogic Help.*

➤ **To create a new WebLogic Managed Server:**

1. If WebLogic Administration Server is not already running, from a command prompt, navigate to the *BEA_HOME*\user_projects\domains*[domainname]* directory and type the appropriate command:

   - (Windows) startWebLogic.cmd
   - (Linux, UNIX) ./startWebLogic.sh

2. To access the WebLogic Administration Console, type http://*[host name]*:7001/console in the URL line of a web browser.

3. Type the user name and password that were used to create this WebLogic configuration, and then click **Log In**.

4. Under Change Center, click **Lock & Edit**.

5. Under Domain Structure, expand **Environment** and click **Servers**.

6. In the right pane, under Servers, click **New**.

7. In the right pane, under Create a New Server, type a name for your managed server, such as server1, in the **Server Name** box.

8. In the **Server Listen Port** box, type a port number that is not currently in use, such as 8001. The Administration Server is already using port 7001.

9. Click **Finish** and then click **Activate Changes**.

10. Under Change Center, click **Lock & Edit**.

11. Click **Environment** > **Machines**.

12. In the right pane, under **Machines**, click **New**.

13. In Create a New Machine -> Enter Machine Name, select the operating system you are using.

14. Click **OK**.

15. Click **Environment** > **Servers** and select the Managed Server you created.

16. In the **Machine** list, select the computer you just created.

17. Make sure that the listen port is the same as the port number you entered in step 8.

18. Click **Save** and then click **Activate Changes**.

Note: To start a Managed Server from the Administration Console, the Node Manager must also be running. It is recommended that you always start and stop the Managed Server from the Administration Console.

Note: Do not restart your server at this point. After all the WebLogic configuration changes are made, the server will be restarted when you reach the procedure "Restarting WebLogic" on page 66.

Configuring WebLogic Server

You must make the following configuration changes to optimize your application server's performance for your LiveCycle ES installation.

Configuring anonymous admin look-up

You must enable anonymous admin lookup on the application server. This setting allows read-only access to the WebLogic Server MBeans.

➤ **To enable anonymous admin look-up:**

1. With WebLogic Server running, type `http://[host name]:[port]/console` in the URL line of a web browser to access the WebLogic Server Administration Console.

2. Type the user name and password that was used in creating this WebLogic configuration, and then click **Log In**.

3. Click **Lock & Edit**.

4. Under Domain Structure, click the name of your domain.

5. In the right pane, click the **Security** tab.

6. Select **Anonymous Admin Lookup Enabled** and then click **Save**.

7. Click **Activate Changes**.

Note: Do not restart your server at this point. After all the WebLogic configuration changes are made, the server will be restarted when you reach the procedure "Restarting WebLogic" on page 66.

Configuring WebLogic for Web Services

For LiveCycle ES to accept requests using web services, you must complete the following procedure to modify authentication of the servlet container.

Note: This step is required for the following WebLogic known issue: CASE_ID_NUM: 690940: CASE TITLE: WebLogic 9.2 Servlet Container always tries to authenticate username/password.

"An application will prompt for user name and password even though the application is not configured to use any security constraints. Upon adding this flag and setting it to `false`, it will disable the popup that normally surfaces during the request being forwarded as we see in our case."

➤ **To modify authentication for the servlet container:**

1. Start the WebLogic Administration Server.

2. Open a command prompt and set the environment using the setWLSEnv script, as shown in the following example:

   - (Windows): From the *[BEA_HOME]*`\weblogic92\server\bin\ folder, type:` `setWLSEnv.cmd`

   - (Linux/UNIX): From the *[BEA_HOME]*`/weblogic92/server/bin/ directory, type:` `setWLSEnv.sh`

3. Start the WebLogic scripting tool by typing the following command:

   ```
   java weblogic.WLST
   ```

   If you see "Exception in thread "main" java.lang.NoClassDefFoundError: weblogic/WLST" complete the following:

   - From a command prompt, set the environment using `<BEA Home>/weblogic92/server/bin/setWLSEnv.cmd`

   - From a command prompt, copy the CLASSPATH returned from the setWLSEnv.[cmd | sh] and append the export CLASSPATH, for example:

   ```
   CLASSPATH=/usr/local/bea92/patch_weblogic920/profiles/default/sys_manife
   st_classpath/weblogic_patch.jar:/usr/local/bea92/jdk150_04/lib/tools.jar
   :/usr/local/bea92/weblogic92/server/lib/weblogic_sp.jar:/usr/local/bea92
   /weblogic92/server/lib/weblogic.jar:/usr/local/bea92/weblogic92/server/l
   ib/webservices.jar export CLASSPATH
   ```

   - From a command prompt, copy the PATH returned from the setWLSEnv.[cmd | sh] and append export PATH, for example:

   ```
   PATH=/usr/local/bea92/weblogic92/server/bin:/usr/local/bea92/jdk150_04/j
   re/bin:/usr/local/bea92/jdk150_04/bin:/usr/local/bea92/weblogic92/server
   /bin:/usr/local/bea92/jdk150_04/jre/bin:/usr/local/bea92/jdk150_04/bin:/
   usr/sbin:/usr/bin export PATH
   ```

4. Type the following commands in WLST to update servlet container authentication:

   ```
   connect('<WebLogic username>','<WebLogic password>','<WebLogic URL>')
   edit()
   startEdit()
   cd('SecurityConfiguration')
   cd('<domain name>')
   set('EnforceValidBasicAuthCredentials','false')
   activate()
   exit()
   ```

   Note: The WebLogic URL will be in the format `t3://hostname:[port]`, where the port is probably 7001.

Restarting WebLogic

After you make all your configuration changes, you need to restart WebLogic for the changes to take affect. The WebLogic Managed Server and the WebLogic Administration Server need to be restarted. The Node Manager does not need to be restarted.

➤ **To stop WebLogic Managed Server:**

1. From the WebLogic Server Administration Console, under Domain Structure, click the domain name.

2. Click the **Control** tab.

3. Select the check box beside the server you want to stop, click **Shutdown**, and select one of the following options:

    When work completes: Initiates a graceful shutdown of the selected server, causing the Managed Server to notify its subsystems to complete all in-work requests. A graceful shutdown gives the WebLogic Server subsystems time to complete certain application processing currently in progress.

    Force Shutdown Now: Initiates a forced shutdown, causing the Managed Server to instruct subsystems to immediately drop in-work requests.

4. At the WebLogic Server Administration Console prompt, click **Yes** to confirm the command.

You can verify that the Managed Server has been shut down by viewing the table at the bottom of the Control tab. The table displays a list of all of the servers and indicates their current state.

➤ **To stop WebLogic Administration Server:**

1. From a command prompt, navigate to *BEA_HOME\user_projects\domains\[appserverdomain]\bin*.

2. Type the following command:
    - (Windows) `stopWebLogic.cmd`
    - (Linux, UNIX) `./stopWebLogic.sh`

3. Enter the WebLogic user name and password.

➤ **To restart WebLogic Administration Server:**

1. From a command prompt, navigate to *BEA_HOME*/user_projects/domains/*[appserverdomain]*.

2. Type the following command:
    - (Windows) `startWebLogic.cmd`
    - (Linux, UNIX) `./startWebLogic.sh`

3. Enter the WebLogic user name and password.

➤ **To restart WebLogic Managed Server (if it did not start automatically):**

1. When the WebLogic Administration Server has started, log in to the WebLogic Server Administration Console.

2. Click **Lock & Edit**.

3. Under Domain Structure, click **Environment** > **Servers** and, in the right pane, click the managed server.

4. On the next screen, click the **Control** tab.

5. Select the check box beside the managed server you want to start, and then click **Start**.

6. Click **Yes**.

Configuring the LiveCycle ES database connectivity

All configurations required for database connectivity are discussed in *Installing and Deploying LiveCycle ES for WebLogic* at www.adobe.com/go/learn_lc_installWebLogic.

Next steps

Install LiveCycle ES by following the instructions provided in *Installing and Deploying LiveCycle ES for WebLogic*.

6 | **Configuring a WebSphere Application Server**

This chapter describes how to install and configure the WebSphere Application Server that will host your LiveCycle ES installation.

This section uses the following conventions:

Name	Default value
[appserver root]	(Windows) C:\Program Files\IBM\WebSphere\AppServer
	(Linux and UNIX) /opt/IBM/WebSphere/AppServer
	(AIX) /usr/IBM/WebSphere/AppServer

Installing WebSphere Application Server

You must install WebSphere Application Server 6.1 for running LiveCycle ES products.

When you run the LiveCycle ES installer, you should run it as the same user.

Installing a Fix Pack to WebSphere 6.1

After you have installed WebSphere 6.1, you must update to Fix Pack 5 or later before you deploy LiveCycle ES.

Refer to the WebSphere support website for access to the WebSphere updates.

Setting the JAVA_HOME and PATH environment variables

As part of your WebSphere installation, a Java SDK was installed. The `JAVA_HOME` and `PATH` environment variables must point to the Java SDK where LiveCycle ES is to be deployed.

➤ **To set the JAVA_HOME environment variable (Windows):**

1. Select **Start** > **Control Panel** > **System**.

2. Click the **Advanced** tab**.**

3. Click **Environment Variables**.

4. In the System Variables area, click **New**.

5. Enter `JAVA_HOME` as the variable name and enter the directory where you installed Java SDK. This directory is where WebSphere installed the Java SDK containing the /bin sub directory. For example, type the following:

    ```
    C:\Program Files\IBM\WebSphere\AppServer\java
    ```

Note: To verify your `JAVA_HOME` environment variable, open a command prompt and run the following command:

```
%JAVA_HOME%\bin\java -version
```

You should receive a response that begins with `java version "1.5.0"`.

➤ **To set the PATH environment variable (Windows):**

1. Select **Start** > **Control Panel** > **System**.

2. Click the **Advanced** tab**.**

3. Click **Environment Variables**.

4. In the System Variables area, select the PATH variable and click **Edit**. Append the following text to the beginning of the variable value:

   ```
   %JAVA_HOME%\bin;
   ```

➤ **To set the DISPLAY setting (UNIX and Linux):**

- From a command prompt, type the following text:

  ```
  DISPLAY=<IP Address of windows machine telnet, or cygwin to Unix-Linux
  machine>:0.0
  EXPORT DISPLAY.
  ```

Note: If the Display setting is not set correctly, the following error occurs when you start LiveCycle Configuration Manager as: "No JRE is found...".

➤ **To set the JAVA_HOME environment (UNIX and Linux):**

- It is recommended that you set the `JAVA_HOME` variable for Borne and Bash shells as shown in the following example:

  ```
  JAVA_HOME=/usr/IBM/WebSphere/AppServer/java
  export JAVA_HOME
  ```

Note: The specific path varies based on the installation directory you specified and the operating system you are installing on.

➤ **To set the PATH environment variable (UNIX and Linux):**

- It is recommended that you set the `PATH` variable for Borne and Bash shells as shown in the following example:

  ```
  PATH=$JAVA_HOME/bin:$PATH
  export PATH
  ```

Accessing WebSphere Administrative Console

Several procedures in this section require you to access the Administrative Console of the WebSphere Application Server.

➤ **To start WebSphere Base:**

1. To start the Administrative Console, from *[application server home]*/bin, type the appropriate command:

   - (Windows) `startServer.bat` ***server1***

- (UNIX, Linux) `./startServer.sh` **server1**

Note: You must replace **server1** with the name of your server instance.

➤ **To access the WebSphere Administrative Console:**

1. Start the WebSphere Application Server.

2. In the URL line of a web browser, type the appropriate URL:

   `http://[host name]:[port]/ibm/console`

   If you are administering the server you are currently using, you can replace *[hostname]* with `localhost`. The port value will depend on the application server and whether Global Security is enabled. The default port value for WebSphere 6.1.0.5 is 9060. If Global Security is enabled, the default SSL port value is 9043.

3. If WebSphere Global Security is enabled, type the WebSphere user ID and password in the boxes provided.

4. Click **Log In**.

Directory permissions

The LiveCycle ES application needs to extract files to the *[appserver root]*/installedApps directory (for example: /usr/IBM/WebSphere/AppServer); therefore, it is important that writable permissions be given to that directory. If writable permissions cannot be given, the section below describes how to modify the location for the extracted files.

➤ **To modify the location for the extracted files:**

1. Log in to the WebSphere Administrative Console.

2. Click **Servers** > **Application Servers**, and click your server name (for example, click **server1**).

3. Under Server Infrastructure, click **Java and Process Management**.

4. Click **Process Definition**.

5. Under Additional Properties, click **Java Virtual Machine**.

6. Under Additional Properties, click **Custom Properties**.

7. Click **New** and create a custom property named *adobeidp.RootDirectory*.

8. Set the value of adobeidp.RootDirectory to the path where Adobe native files should be extracted (for example, *[appserver root]*/profiles/*[server]*/installedApps).

9. Save and restart the application server.

Preparing WebSphere Application Server

This section describes how to prepare and configure an application server instance for your LiveCycle ES deployment.

Required application server instances

LiveCycle ES requires one or more application server instances. LiveCycle ES is deployed into one application server instance and, if you are using Business Activity Monitoring ES, it must be installed into an application instance.

Increase SOAP request time-out

You must modify the SOAP request time-out value for LiveCycle ES.

➤ **To increase the SOAP request time-out value:**

1. Navigate to your *[appserver root]* directory and search for all files named *soap.client.props*. Multiple files may have this name. Perform steps 2 to 4 for each of them.

2. Open the file in a text editor, find the **com.ibm.SOAP.requestTimeout** property, and change the value from 180 to 1800.

3. Save the file(s).

4. In the navigation tree, click **Servers** > **Application Servers** and, in the right pane, click the server name.

5. Under Server Infrastructure, expand **Administration** > **Administration Services**.

6. Under Additional Properties, click **JMX connectors**, and then click **SOAPConnector** from the list.

7. On the next screen, click **Custom Properties**, and then click **requestTimeout** from the list.

8. Under General Properties, in the **Value** box, change 600 to 1800.

9. Click **OK** or **Apply**.

10. Click **Save directly to master configuration**.

Configuring the LiveCycle ES database connectivity

All configurations required for database connectivity are discussed in *Installing and Deploying LiveCycle ES for WebSphere* at www.adobe.com/go/learn_lc_installWebSphere.

Next steps

Install LiveCycle ES by following the instructions provided in *Installing and Deploying LiveCycle ES for WebSphere* at www.adobe.com/go/learn_lc_installWebSphere.

Index

A

admin look-up, enabling anonymous for WebLogic 64
Administration Server, WebLogic, configuring 61
Administrative Console, WebSphere, accessing 69
Adobe Acrobat licensing restriction 18
Adobe LiveCycle Business Activity Monitoring ES 17
Adobe LiveCycle Data Services ES 6
Adobe LiveCycle ES database
    connectivity, configuring for JBoss 37
    connectivity, configuring for WebLogic 67
    connectivity, configuring for WebSphere 71
    DB2, creating 23
    MySQL, creating 21
    Oracle, creating 23
    SQL Server, creating 25
Adobe LiveCycle PDF Generator ES, requirements for 16
Adobe LiveCycle Reader Extensions ES Rights credentials 19
anonymous admin look-up for WebLogic 64
application servers
    instance on WebSphere, creating 71
    supported 12

B

bc tool 15

C

certificates, obtaining 20
configuration files, JBoss 32
configuring
    database connectivity for JBoss 37
    database connectivity for WebLogic Server 67
    database connectivity for WebSphere 71
    LDAP 18
    WebLogic Administration Server 61
    WebLogic Server 64
content management systems supported 14
conventions, path name 7
creating
    application server instance on WebSphere 71
    DB2 database 23
    WebLogic Managed Server 63
    WebLogic Server domain 60
credentials, LiveCycle Reader Extensions ES 19
CRLs, obtaining 20

D

data services supported 13
databases
    combinations with platforms 9
    connectivity, configuring for WebLogic Server 67
    connectivity, configuring for WebSphere 71
    DB2, creating 23
    drivers supported 13
    MySQL, creating 21
    Oracle, creating 23
    SQL server, creating 25
    supported 13
DB2 database
    configuring for Adobe preconfigured JBoss 51
    configuring for manually configured JBoss 54
    creating 23
digital certificates, obtaining 20
domain, creating for WebLogic Server 60

E

email servers supported 13
environment variables
    setting for JBoss 30
    setting for PDF Generator ES 16
    setting for WebLogic 59
    setting for WebSphere 68

F

file formats, software support for 16
Fix Pack, installing for WebSphere 6.1 68

G

GNU bc tool 15

H

hardware requirements 11
Hardware Security Module 20

I

installing
    JBoss 31
    native file conversion software 16
    summary of tasks 9
    WebLogic Server 59
    WebSphere Application Server 68