

PREPARING TO INSTALL ADOBE® LIVECYCLE® ES3 (SINGLE SERVER)

Legal notices

For legal notices, see http://help.adobe.com/en_US/legalnotices/index.html.

Contents

Chapter 1: About This Document

1.1 Conventions used in this document	1
1.2 Additional information	3

Chapter 2: System Requirements

2.1 Prepare your server environment	4
2.2 System requirements	4
2.3 LDAP configuration	17
2.4 Global document storage directory	17
2.5 Installation considerations	18
2.6 LiveCycle IPv6 support	19
2.7 Central Migration Bridge service	22

Chapter 3: LiveCycle Credentials and Certificates

3.1 Obtaining the Reader Extensions Rights credential	24
3.2 Obtaining digital certificates for use with Digital Signatures	25

Chapter 4: Creating the LiveCycle Database

4.1 Database configuration requirements	26
4.2 Minimum database user permissions	27
4.3 Creating an Oracle database	27
4.4 Creating a SQL Server database	28
4.5 Creating a DB2 database	32
4.6 Creating a MySQL database	34

Chapter 5: Configuring a JBoss Application Server

5.1 Configuration options	37
5.2 Installing JDK for JBoss	38
5.3 Starting and stopping JBoss	39
5.4 Database connectivity for Adobe pre-configured JBoss 5.1	40
5.5 Next steps	47

Chapter 6: Configuring a WebSphere Application Server

6.1 Installing WebSphere	48
6.2 Preparing WebSphere Application Server	50
6.3 Configuring the LiveCycle database connectivity	52

Chapter 7: Configuring a WebLogic Server

7.1 LiveCycle requirements for WebLogic	53
7.2 Installing WebLogic Server	53
7.3 Installing JDK	53
7.4 Creating the WebLogic Server domain	55
7.5 WebLogic Managed Server	56
7.6 Stopping and restarting WebLogic	57

7.7 Creating JMX policies for database initialization 59

7.8 Configuring the LiveCycle database connectivity 60

Chapter 8: Preconfigured Environment Checklist

Chapter 9: Appendix - Manually Configuring JBoss

9.1 Installing the JDK for JBoss 63

9.2 Manually installing JBoss 64

9.3 Starting and stopping JBoss 64

9.4 Modifying the JBoss configuration 65

9.5 Copying jar files 68

9.6 LiveCycle database connectivity for manually installed JBoss 68

9.7 Next steps 76

Chapter 1: About This Document

Adobe® LiveCycle® Enterprise Suite 3 (ES3) is an enterprise server platform that helps you automate and streamline business processes. LiveCycle comprises the following components:

- J2EE-based Adobe® LiveCycle® Foundation 10 provides server capabilities and runtime environment
- Tools to design, develop, and test LiveCycle applications
- Modules and Services are deployed on LiveCycle server and provide functional services

For more information about the LiveCycle architecture and capabilities, see [LiveCycle Overview](#).

This document is part of a larger documentation set available at [LiveCycle Documentation page](#). It is advised that you start with the preparing guide and then move on to installation and configuration guide depending on whether you are performing a fresh installation (single server or cluster setup) or upgrading your existing LiveCycle deployment. For Turnkey deployment, which is only for evaluation purposes, see [Installing and Deploying LiveCycle using JBoss Turnkey](#).

1.1 Conventions used in this document

The following naming conventions are used for common file paths.

Name	Description	Default value
[LiveCycle root]	The installation directory that is used for all LiveCycle modules. The installation directory contains subdirectories for Adobe® LiveCycle® Configuration Manager, the LiveCycle SDK, and each LiveCycle module installed (along with the product documentation). This directory also includes directories relating to third-party technologies.	Windows: C:\Adobe\Adobe LiveCycle ES3\ AIX, Linux, and Solaris: opt/adobe/adobe_lifecycle_es3/
[appserver root]	The home directory of the application server that runs the services that are part of LiveCycle.	JBoss 5.1 on Windows: C:\jboss JBoss 5.1 on Linux and Solaris: /opt/jboss JBoss Enterprise Application Platform 5.1 on Windows: C:\jboss-eap-5.1\jboss-as JBoss Enterprise Application Platform 5.1 on Linux and Solaris: /opt/jboss-eap-5.1/jboss-as WebSphere 7.0.0.15 on Windows: C:\Program Files\IBM\WebSphere\AppServer WebSphere 7.0.0.15 on Linux and Solaris: /opt/IBM/WebSphere/AppServer WebSphere on AIX: /usr/IBM/WebSphere/AppServer, or, /opt/IBM/WebSphere/AppServer WebLogic 11g on Windows: C:\Oracle\Middleware\wlserver_10.3 WebLogic 11g on Linux and Solaris: /opt/Oracle/Middleware/wlserver_10.3
[WL_Home]	The install directory for WebLogic as specified for the WL_HOME environment variable.	WebLogic 11g on Windows: C:\Oracle\Middleware\ WebLogic 11g on Linux and Solaris: /opt/Oracle/Middleware
[appserverdomain]	The domain that you configured on WebLogic. The default domain is called <i>base_domain</i> .	WebLogic 11g on Windows: C:\Oracle\Middleware\user_projects\domains\base_domain WebLogic 11g on Linux and Solaris: /opt/Oracle/Middleware/user_projects/domains/base_domain
[WebSphere ND root]	The install directory for WebSphere Application Server, Network Deployment	WebSphere on Windows: C:\Program Files\IBM\WebSphere\AppServer WebSphere on Linux and Solaris: /opt/IBM/WebSphere/AppServer WebSphere on AIX: /usr/IBM/WebSphere/AppServer

About This Document

Name	Description	Default value
[server name]	The name of the server configured on your WebLogic or WebSphere server	WebLogic: server1 WebSphere: server1
[profile_name]	The profile name for the JBoss application server.	Adobe pre-configured JBoss: lc_<db-name>_cl Manual or Downloaded Jboss: standard
[dbserver root]	The location where the database server is installed	Depends on the database type and your specification during installation

Most of the information about directory locations in this document is cross-platform (all filenames and paths are case-sensitive on AIX, Linux, and Solaris). Any platform-specific information is indicated as required.

1.2 Additional information

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

For information about	See
LiveCycle and the modules in general	LiveCycle Overview
Other services and products that integrate with LiveCycle	LiveCycle Developer Center
All documentation that is available for LiveCycle	LiveCycle documentation
LiveCycle release information and last-minute changes that occur to the product	LiveCycle Release Notes
Patch updates, technical notes, and additional information about this product version	Adobe® Enterprise Support

Chapter 2: System Requirements

2.1 Prepare your server environment

Do the following tasks to prepare your server environment:

- 1 Verify that the installation media is not damaged. It is recommended that you copy the installer files to the hard disk of the computer where you intend to install LiveCycle, instead of installing directly from the installation media.
Also verify the MD5 checksum of the installation media matches the checksum posted at [Adobe Website](#).
- 2 Read “[2.2 System requirements](#)” on page 4, and ensure that your software infrastructure is compliant.
- 3 Read System requirements, and ensure that your hardware infrastructure is in place, as per your requirements.
- 4 Install the operating system and update with all necessary patches and service packs.
- 5 Install and configure the database server.
- 6 Install and configure the application server.
- 7 Ensure that your target computer has at least the required amount of RAM as described in “[2.2.1 Minimum hardware requirements](#)” on page 4.

2.2 System requirements

2.2.1 Minimum hardware requirements

This table provides the minimum hardware requirements that LiveCycle supports.

Operating system	Minimum hardware requirement
Microsoft Windows Server® 2008 R1/R2 (64-bit architecture)	Intel® Pentium® 4, 2.8 GHz processor or equivalent VMWare ESX 3.0 or later RAM: 4 GB (64-bit OS with 64-bit JVM) Free disk space: 5.4 GB of temporary space plus 3.4 GB for LiveCycle
Microsoft Windows Server® 2003 Enterprise Edition or Standard Edition SP2 and R2 SP2 (64-bit architecture)	Intel® Pentium® 4, 2.8 GHz processor or equivalent VMWare ESX 3.0 or later RAM: 4 GB (64-bit OS with 64-bit JVM) Free disk space: 5.4 GB of temporary space plus 3.4 GB for LiveCycle
Sun Solaris 10 (64-bit architecture)	UltraSPARC® IIIi, 1.5 GHz processor Solaris Containers (Zones) partitioning RAM: 4 GB (64-bit OS with 64-bit JVM) Free disk space: 5.4 GB of temporary space plus 3.4 GB for LiveCycle

System Requirements

Operating system	Minimum hardware requirement
IBM AIX 6.1	P5 pSeries 520 (Model 52A) 9131-52A, 1.8 GHz processor LPAR partitioning RAM: 4 GB (64-bit OS with 64-bit JVM) Free disk space: 5.4 GB of temporary space plus 3.4 GB for LiveCycle
SUSE Linux Enterprise Server 10.0 and 11.0 (64-bit edition only)	Dual core, 1 GHz processor VMWare ESX 3.0 or later RAM: 4 GB (64-bit OS with 64-bit JVM) Free disk space: 5.4 GB of temporary space plus 3.4 GB for LiveCycle
Red Hat Enterprise Linux AP or ES 5.5 (64-bit edition only)	Dual core, 1 GHz processor VMWare ESX 3.0 or later RAM: 4 GB (64-bit OS with 64-bit JVM) Free disk space: 5.4 GB of temporary space plus 3.4 GB for LiveCycle

2.2.1.1 Intel x86 compatibility

On supported Windows and UNIX -like environments, LiveCycle supports Intel and AMD64 compatible chipsets running either 32-bit or 64-bit supported operating systems.

***Note:** 32-bit platforms are supported only for development and evaluation purposes.*

2.2.1.2 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 V or later.

IBM AIX environments: Power4 or later

Memory requirements: 4 GB of RAM.

2.2.1.3 Recommended hardware requirements for client-side computers**Adobe® LiveCycle® Workbench 10**

- Disk space for installation:

1.5 GB for Workbench only

1.7 GB on a single drive for a full installation of Workbench, Adobe® LiveCycle® Designer 10, and the samples assembly

400 MB for temporary install directories - 200 MB in the user temp directory and 200 MB in the Windows temporary directory

***Note:** If all of these locations reside on a single drive, there must be 1.5 GB of space available during installation. The files copied to the temporary directories are deleted when installation is complete.*

- Memory for running Workbench: 2 GB of RAM
- Hardware requirement: Intel® Pentium® 4 or AMD equivalent, 1 GHz processor
- Minimum 1024 X 768 pixels or greater monitor resolution with 16-bit color or higher

System Requirements

- TCP/IPv4 or TCP/IPv6 network connection to the LiveCycle server

Note: You must have Administrative privileges to install Workbench on Windows. If you are installing using a non-administrator account, the installer will prompt you for the credentials for an appropriate account.

Designer

- Adobe® Reader® X and Adobe® Acrobat® X Pro (recommended) is required to benefit from all the new features in Designer.
- Adobe® Flash® Player 10.2 or later.
- (Optional) Adobe® Flash® Builder® 4.5 or later.
- (Optional) Adobe® Flex® SDK 4.1 (required for customizing form guide components that are shipped with Designer).

Note: Use the Flex SDK 4.1 included with the Designer installer only. Do not use any other version of Flex SDK obtained from the Adobe web site.

Note: For more information, see [Installing Your Development Environment](#).

End-user hardware requirements:

- Adobe® LiveCycle® Workspace 10: 1 GB of RAM (includes requirements for Adobe Flash and Adobe Reader)
- Adobe Flash Player 9 or later: 512 MB of RAM (1 GB recommended)
- Adobe Reader 8.x or later: 128 MB of RAM (256 MB recommended)

Note: For web browser requirements, see [LiveCycle Supported Platform Combinations](#).

2.2.2 Supported Platforms

For detailed information about the supported combinations of operating systems, application server, databases, JDKs, and so on, see [LiveCycle Supported Platform Combinations](#).

2.2.3 Additional requirements for AIX, Linux, and Solaris

Note: On AIX, Linux and Solaris machines, the LiveCycle installer uses the JDK installed on the machine. Therefore, you must ensure to install the supported JDK version. On other operating systems, the installer uses the JVM bundled with the installer.

2.2.3.1 Installing and configuring UTF-8

When installing LiveCycle on AIX, Linux, or Solaris operating systems, you must install and configure the US English version of UTF-8 locale if it is not already installed. You will need the install media (CDs or DVDs) for the operating system to perform this task.

Note: On Linux platforms, this locale is installed by default and is called `en_US.utf8`. It can be verified by using the `locale -a` command.

Installing UTF-8 on AIX

- 1 Verify the US English UTF-8 locale is not installed by typing `locale -a` in a command prompt. Verify that the command output does not contain the entry `EN_US.UTF-8`.
- 2 Access the AIX SMIT utility (in text mode) by typing `smitty mle_add_lang` at the root in the command prompt.

System Requirements

- 3 On the screen that appears, select **UTF-8 US English (United States) [EN_US]** from both the **CULTURAL CONVENTION** and **LANGUAGE TRANSLATION** drop-down lists.

***Note:** Keep the INPUT DEVICE/DIRECTORY as the default /dev/cd0 setting.*

- 4 Press **Enter** to proceed. A message such as the following will appear:

```
installp: Device /dev/cd0 not ready for operation.
Please insert media and press Enter to continue.
```

- 5 Insert the appropriate AIX install disk in the disk drive.
- 6 When the command is complete, exit the SMIT utility and type `locale -a` to verify that EN_US.UTF-8 is set as the locale.

Installing UTF-8 on Solaris

- 1 Verify the US English UTF-8 locale is not installed by typing `locale -a` in a command prompt. Verify that the command output does not contain the entry EN_US.UTF-8.

- 2 Insert the Solaris install CD #1 in the disk drive and mount it to an appropriate location, for example:

```
/cdrom/sol_10_807_sparc/s0
```

- 3 Type the following command as root: `localeadm -a nam -d /cdrom/sol_10_807_sparc/s0`

***Note:** This command installs all locales in the North America (nam) region even if you specify only the en_US.UTF-8 locale.*

- 4 When the command is complete, type `locale -a` to verify that EN_US.UTF-8 is set as the locale.

***Note:** See this link for [FAQs on Solaris Locales](#).*

2.2.3.2 Solaris

***Note:** Ensure that X Window libraries are installed on your operating system. This is required for Adobe® LiveCycle® PDF Generator 10 and Adobe® LiveCycle® Forms 10. See the documentation for your operating system for more information.*

***Important:** Do not use the Solaris `tar` command to extract files or errors (such as missing files) will occur. Download the GNU tar tool and use it to extract all files on a Solaris environment.*

2.2.3.3 Linux

On Linux operating systems, ensure the following:

- **All Linux distributions:** Ensure that X Window libraries are installed on your operating system. This is required for PDF Generator and Forms. See documentation for your operating system for more information.
- **SUSE Linux:** You must install the glibc-locale-32bit library that ships with SUSE Linux Enterprise Server; otherwise, LiveCycle will not generate PDF files. This library file is not installed by default, you must use YaST to install it. (See the [SUSE Linux Enterprise Server documentation](#) for details.)

If you plan to install LiveCycle on SUSE Linux 11, you must also install the libstdc++-libc6.2-2.so.3 libraries. SUSE Linux 11 does not include these libraries by default. For more information, see this [Novell Web](#) page. These libraries are required for running Adobe Central Pro Output Server

2.2.3.4 Configuring the file limit values on Solaris and Linux

To avoid StuckThread issues on a Solaris or Linux environment, add or increase the rlim values in the /etc/system file.

- 1 **(Linux)** Locate and open the /etc/security/limits.conf file.
(Solaris) Locate and open the /etc/system file.

System Requirements

- 2 **(Linux)** Add the following lines to the `/etc/security/limits.conf` file:

```
<app_group> soft nofile 8192
<app_group> hard nofile 8192
```

Replace `<app_group>` with the user group who will run the application server. You may also replace `<app_group>` with an asterisk (*) to match all users and user groups.

(Solaris) Locate and modify the `rlim` values in the `/etc/system` file as follows:

`set rlim_fd_cur`: The initial (soft) maximum number of file descriptors per process. Set this value to 8192 or more.

`set rlim_fd_max`: The hard maximum number of file descriptors per process. Set this value to 8192 or more. (This modification is required only if the default value is lower than 8192). You must have super user privileges to change this value.

Note: The `rlim_fd_max` value must be equal to or greater than the `rlim_fd_cur` value.

- 3 Save and close the file.

- 4 Restart your computer.

Verify the updated settings

- 1 Launch a new shell.
- 2 Type `ulimit -n` and press **Enter**.
- 3 Verify the value returned matches the `rlim` values you have set.

2.2.4 Privileges required to install on Windows

When installing on Windows, you must use an account that has administrator privileges. If you run the installer using a non-administrator account, enter the credentials of an account that has administrator privileges.

2.2.5 Configuring Windows installation

On 64-bit Windows Server 2008 R1, Windows Server 2008 R2 or Vista operating systems, modify the Admin Approval Mode security option as follows:

- 1 (On Windows 2008 R1 and Vista) Go to **Start > Control Panel > Administrative Tools > Local Security Policy > Local Policies > Security Options**.
(On Windows 2008 R2 and Windows 7) Go to **Start > Administrative Tools > Local Security Policy > Local Policies > Security Options**.
- 2 Locate **User Account Control: Behavior of the elevation prompt for administrators in Admin Approval Mode** and set it to **Elevate without prompting**.
- 3 Restart your computer.

2.2.5.1 Disable the Windows UAC on Vista and Windows Server 2008 R1

- 1 To access the System Configuration Utility, go to **Start > Run** and then enter `MSCONFIG`.
- 2 Click the **Tools** tab and scroll down and select **Disable UAC**.
- 3 Click **Launch** to run the command in a new window.
- 4 When finished, close the command window and close the System Configuration window.

- 5 Restart your computer.

To enable the UAC again, repeat the steps above and select **Enable UAC** before clicking Launch.

2.2.5.2 Disable the Windows UAC on Windows Server 2008 R2 and Windows 7

- 1 To access the System Configuration Utility, go to **Start > Run** and then enter **MSCONFIG**.
- 2 Click the **Tools** tab and scroll down and select **Change UAC Settings**.
- 3 Click **Launch** to run the command in a new window.
- 4 Adjust the slider to the **Never notify** level.
- 5 When finished, close the command window and close the System Configuration window.
- 6 Restart your computer.

To enable the UAC again, repeat the steps above and adjust the slider to a desired level before restarting your computer.

Important: The Windows User Account Control (UAC) must remain disabled for PDF Generator to work properly. You can run the installation and configuration process by turning on the UAC with the Elevate without prompting option enabled. However, disable UAC to run PDF Generator.

2.2.6 Additional requirements for PDF Generator

Note: You cannot use the Shared Printer Protocol for the `SendToPrinter` API on Windows 2008 machines that have PDF Generator deployed on them. Use alternate protocols like CIFS or Direct IP.

2.2.6.1 User account for Windows

You must use a user account with administrator privileges for the following tasks:

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

2.2.6.2 Using 64-bit application servers with PDF Generator

If you are using a 64-bit application server on a system with PDF Generator, ensure that a 32-bit Java 6 JDK is installed in addition to the 64-bit one the application server uses. Set the environment variable `JAVA_HOME_32`. This variable is required to point to a 32-bit JDK on systems where a 64-bit application server is in use. The specific path varies based on the installation directory you specified and the operating system you are installing on.

Note: You need to install the 32-bit Sun JDK and configure `JAVA_HOME_32` to point to the directory where it resides. Review *Sun Java 6 Release Notes > Supported System Configurations* and download the 32-bit version for your operating system. However, if you are using AIX, do not set `JAVA_HOME_32`.

Important: Ensure that `JAVA_HOME_32` is set only as an environment variable and is not included in the `PATH`. If `JAVA_HOME_32` is included in the `PATH`, Java core dumps may appear during EAR deployment or when you restart the server.

2.2.6.2.1 Set the Windows `JAVA_HOME_32` variable

- 1 Select **Start > Control Panel > System**.
- 2 Click the **Advanced** tab.

System Requirements

- 3 Click **Environment Variables** and, under System Variables, click **New**.
- 4 Enter the environment variable `JAVA_HOME_32`. This value is the directory that contains the JDK. For example, type the following:

```
D:\Program Files (x86)\Java\jdk1.6.0_26
```

2.2.6.2.2 Set the JAVA_HOME_32 variable on Linux or Solaris

Set the `JAVA_HOME_32` variable for the supported JDK for Bourne and Bash shells as shown in this example:

```
JAVA_HOME_32=/opt/jdk1.6.0_26
export JAVA_HOME_32
```

2.2.6.3 Native file conversion software installation

Before you install PDF Generator, install the software that supports the file formats for which PDF conversion support is required and manually activate the licenses for the software using the user account that is used for running the application server process.

Refer to the individual licensing agreement for each native application that your LiveCycle deployment will support, and ensure that your LiveCycle deployment meets the licensing requirements specified. Typically, each LiveCycle user who will use native application support must also have an activated license on their own computer for the native application.

PDF Generator can be extended to convert these additional file types to PDF files by using the following applications:

Note: PDF Generator supports only English, French, German, and Japanese versions of the supported applications.

- Microsoft Office 2007, 2010 (DOC, DOCX, XLS, XLSX, PPT, PPTX, RTF, TXT, Microsoft Office open XML Formats)
- Microsoft Office Visio 2007, 2010 (VSD)
- Microsoft Publisher 2007, 2010 (PUB)
- Microsoft Project 2007, 2010 (MPP)
- AutoCAD 2008 and 2009 (DWG, DWF)

Note: LiveCycle supports only 32-bit editions of all the above mentioned software.

Note: Native file conversions using AutoCAD for DWG, DXF, and DWF files are supported only on 32-bit environments. Conversions for these file types using Acrobat are supported on both 32-bit and 64-bit platforms.

Note: LiveCycle PDF Generator does not support DWF and DWG files created with AutoCAD 2013.

- Corel WordPerfect 12, X4 (WPD)
- Adobe Photoshop® CS3
- Adobe FrameMaker® 7.2, 8.0 (FM)
- Adobe PageMaker® 7.0 (PMD, PM6, P65, PM)
- OpenOffice 3.3 (ODT, ODP, ODS, ODG, ODF, SXW, SXI, SXC, SXD, SXM)

Note: OpenOffice 3.3 or later must be installed on the server to convert the documents created in version 3.3.

Note: On Linux platform, OpenOffice must be installed under `/root` user. If OpenOffice is installed for specific users, PDFG might not be able to convert OpenOffice documents.

You do not need to install a native software application to convert the following native file formats:

- Print files (PS, PRN, EPS)
- Web files (HTML)

System Requirements

- Image files (JPEG, GIF, BMP, TIFF, PNG)

2.2.6.4 Installing Acrobat for PDF Generator

Install Acrobat X before running the LiveCycle installer. Ensure that you launch Acrobat at least once after installing it to avoid PDF Generator configuration issues. Dismiss all modal dialog boxes that appear on launching Acrobat.

***Note:** Ensure that Acrobat is installed using the same user account that you will use to install LiveCycle.*

However, if LiveCycle is installed and Acrobat X is not installed, install Acrobat X and then run the Acrobat_for_PDFG_Configuration.bat script, located in the folder *[LiveCycle root]\pdfg_config*. Otherwise, PDF conversions may fail.

The Configuration Manager sets the Acrobat_PATH (case-sensitive) environment variable automatically. You can also choose to set it manually, see “[2.2.6.7 Setting environment variables](#)” on page 11. Restart your application server after setting the environment variable.

***Note:** Depending on the JDK that you use, download the required JAR files from either Sun or IBM website.*

2.2.6.5 Configure Acrobat to use SHX fonts (Windows only)

***Note:** Perform these steps to configure Acrobat if you want PDF Generator to use SHX fonts to convert AutoCAD DWG files without installing AutoCAD. Also, these steps need to be performed for all user accounts configured in Administration Console.*

- 1 Open Acrobat.
- 2 Select **Edit > Configurations**.
- 3 Select **Convert to PDF > Autodesk AutoCAD**.
- 4 Click **Edit Settings**.
- 5 Click **Configuration Preferences**.
- 6 Click **Browse** next to the SHX Font File Search Path and specify the path to the SHX font file.
- 7 Click **OK** on each opened dialog.

2.2.6.6 QuickTime 7

PDF Generator requires that QuickTime 7 (Player or Pro) be installed if you want to convert video embedded in files, such as PowerPoint presentations to PDF multimedia files. This application is available from the Apple Downloads site.

2.2.6.7 Setting environment variables

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

The names of these environment variables are listed here:

- FrameMaker_PATH
- Notepad_PATH (This should be left blank)
- OpenOffice_PATH
- PageMaker_PATH
- Photoshop_PATH
- WordPerfect_PATH
- Acrobat_PATH

System Requirements

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

For example, the variable `FrameMaker_PATH` may contain the value `C:\Program Files\Adobe\FrameMaker7.2\FrameMaker.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 (x86)\OpenOffice.org 3\`.

Paths for Microsoft Office applications such as Word, PowerPoint, Excel, Visio, and Project or for AutoCAD are not required. 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** and, under System Variables, click **New**.
- 4 Enter the environment variable name you need to set (for example, `FrameMaker_PATH`). This value is the directory that contains the executable file. For example, type the following:

```
C:\Program Files\Adobe\FrameMaker7.2\FrameMaker.exe
```

2.2.6.8 Service Control Manager command line tool

Before you complete an automatic installation of PDF Generator 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 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. To confirm that the tool is installed on your server, type `sc.exe` from a command prompt. The tools usage is returned.

2.2.6.9 Headless mode configuration

If you are running PDF Generator in a headless mode environment (that is, on a server without a monitor, keyboard, or mouse), the x11 libraries must be installed. Some flavors of Linux do not install these libraries by default; therefore, you must obtain the libraries and install them manually.

Note: *Activating x11 forwarding on a shell session causes the SOAP UI to create UI elements during SOAP requests, leading to request failures. To avoid request failures, you must add the `-Djava.awt.headless=true` JVM argument.*

2.2.6.10 Enabling multi-threaded file conversions and multi-user support for PDF Generator

By default, PDF Generator can convert only one OpenOffice, Microsoft Word, or PowerPoint document at a time. If you enable multi-threaded conversions, PDF Generator can convert more than one of the documents concurrently by launching multiple instances of OpenOffice or PDFMaker (which is used to perform the Word and PowerPoint conversions).

Note: *Multi-threaded file conversions are not supported with Microsoft Word 2003 and PowerPoint 2003. In addition, Microsoft Excel (2003, 2007 and 2010 versions) is not supported. To enable multi-threaded file conversions, upgrade to Microsoft Word 2007 or 2010 and PowerPoint 2007 or 2010.*

Note: *Microsoft Excel and Microsoft Visio files are not converted simultaneously. During conversion, `EXCEL.exe` and `VISIO.exe` are watched in the task manager. The Visio file is processed only after processing on the Excel file is complete.*

System Requirements

Each instance of OpenOffice or PDFMaker is launched using a separate user account. Each user account that you add must be a valid user with administrative privileges on the LiveCycle Server computer. On 64-bit Windows, disable Windows UAC. For more information, see “[2.2.5 Configuring Windows installation](#)” on page 8

After your LiveCycle Server is configured, add LiveCycle user accounts in Administration Console. See the User accounts for multi-threaded file conversions section in the LiveCycle installation guide for your application server. To enable multiuser support for native files and OpenOffice files on a Windows environment, add a minimum of three users with the following permissions.

When you add users for PDF Generator native conversions, grant the user running the application server with the Replace a process level token privilege. For more information, see “[2.2.6.12 Granting the Replace a process level token privilege \(Windows only\)](#)” on page 14

Dismiss initial dialogs and automatic updates for native applications

Converting native files from PDF Generator requires dismissing any initial registration, activation, and Improvement program dialogs with the option to not show them again. Automatic updates for these applications also needs to be disabled as these update dialogs can cause failures on a running server.

The dialogs and automatic update need to be disabled for the user running the server and all user accounts configured under PDFG Accounts for multi-user support. The dialogs need to be dismissed for the following applications if installed on the server:

- Microsoft Word (Windows)
- Microsoft Excel (Windows)
- Microsoft PowerPoint (Windows)
- Microsoft Project (Windows)
- Microsoft Publisher (Windows)
- Microsoft Visio (Windows)
- OpenOffice (Windows/ Solaris / Linux)
- Adobe Acrobat (Windows)
- Adobe Photoshop (Windows)
- Adobe FrameMaker (Windows)
- Adobe PageMaker (Windows)
- Autodesk AutoCAD (Windows)
- Corel WordPerfect (Windows)

Disable error reporting on Windows Server 2003 or 2008 (Optional but recommended)

While converting a document to PDF using PDF Generator on Windows Server 2003 or 2008, Windows may report that the executable has encountered a problem and needs to close. However, it does not impact the PDF conversion as it continues in the background.

To avoid receiving the error, you can disable the Windows error reporting. For more information on disabling error reporting, see the following articles:

- Windows Server 2003 - <http://technet.microsoft.com/en-us/query/cc773392>
- Windows Server 2008 - <http://technet.microsoft.com/en-us/library/cc754364.aspx>

System Requirements**Additional configuration required for OpenOffice on Linux or Solaris**

- 1 Add entries for additional users (other than the administrator who runs the LiveCycle Server) in the `/etc/sudoers` file. For example, if you are running LiveCycle as a user named `lcadm` and a server named `myhost`, and you want to impersonate `user1` and `user2`, add the following entries to `/etc/sudoers`:

```
lcadm myhost=(user1) NOPASSWD: ALL
```

```
lcadm myhost=(user2) NOPASSWD: ALL
```

This configuration enables `lcadm` to run any command on host `'myhost'` as `'user1'` or `'user2'` without prompting for password.

- 2 Allow all the LiveCycle users to make connections to the LiveCycle Server. For example, to allow a local user named `user1` the permission of making the connection to the LiveCycle Server, use the following command:

```
xhost +local:user1@
```

For more details, refer to `xhost` command documentation.

- 3 Restart the server.

2.2.6.11 Multi-user support for PDF Generator

To enable multi-user support for native files and OpenOffice files on a Windows environment, a minimum of three users with the following permissions must be added. On a AIX, Linux, or Solaris platform, create at least one user.

Platform	User permissions
Windows 2008 Server	Users with administrative privileges, UAC disabled and with replace a process level token privilege
Windows 2003 Server	Users with administrative privileges
AIX, Linux, and Solaris	Users with <code>sudo</code> privileges

When you add users for PDF Generator native conversions, you must grant the user running the application server with the *Replace a process level token* privilege. See [“2.2.6.12 Granting the Replace a process level token privilege \(Windows only\)”](#) on page 14.

2.2.6.12 Granting the Replace a process level token privilege (Windows only)

User accounts that are used to start the application server from a command prompt and not as a Windows service requires the *Replace a process level token* privilege. This setting is required for PDF Generator.

- 1 Click **Start > Run**, and then type `gpedit.msc`.
- 2 On the Group Policy dialog box, select **Computer Configuration > Windows Settings > Security Settings > Local Policies > User Rights Assignment**, and double click **Replace a process level token**.
- 3 Click **Add User or Group**, add the Windows user account that is used to open the command prompt from which the application server is started.
- 4 Restart Windows, and then start the application server.

2.2.7 Additional requirements for Connector for Documentum

If LiveCycle is connecting to Documentum, you must install Document Foundation Classes on machine hosting LiveCycle.

2.2.8 Additional requirements for Connector for IBM Content Manager

Adobe® LiveCycle®10 Connector for IBM® Content Manager requires the following software installed (both available from the IBM website):

- DB2 Universal Database Client
- IBM Information Integrator for Content (II4C)

See “Post-Deployment Activities” chapter in the Installing and Deploying LiveCycle document for your application server.

2.2.8.1 Configure the connection for a single IBM Content Manager datastore

- 1 Start the DB2 Configuration Assistant.
- 2 Click **Selected > Add Database Using Wizard**.
- 3 Select **Manually Configure a Connection to a Database** and click **Next**.
- 4 Select **TCP/IP** and click **Next**.
- 5 Specify the following TCP/IP communication options and then click **Next**:
 - In the **Host Name** box, type the host name of the server hosting DB2 Content Manager.
 - Leave the Service Name box empty.
 - In the **Port Number** box, type the port number. The default DB2 Content Manager port number is 50000.
- 6 In the **Database Name** box, type the IBM Content Manager datastore name and, in the **Database Alias** box, type the alias name for the datastore and then click **Next**.
- 7 Click **Next** to accept the default data source settings.
- 8 In the **Operating System** list, select the operating system you are using and then click **Next**.
- 9 Specify the following system options and then click **Next**:
 - In the **System Name** box, type the server name hosting DB2. If you click Discover, DB2 Content Manager searches for the system name you specified and, if the system is not found, all of the DB2 instances are listed.
 - In the **Host Name** box, type the name of the host, or click View Details to show the domain and IP address of the system you named in the previous step.
 - In the **Operating System** list, select the operating system (Windows, Linux, or AIX) on which you deployed DB2 Content Manager.
- 10 (Optional) To specify Security options, select **Use Authentication Value in Server's DBM Configuration** and click **Finish**.
- 11 In the Test Connection dialog box, test the connection as required.

2.2.8.2 Configure connections for multiple IBM Content Manager datastores

- 1 Configure the initial connection by following the steps in “[2.2.8.1 Configure the connection for a single IBM Content Manager datastore](#)” on page 15.
- 2 Add additional database connections by modifying the cmbicmsrvs.ini file (the file that stores the datastore information) as follows:
 - From a command prompt window, change the directory to [II4C home]/bin (for example, C:\Program Files\db2cmv8\ on Windows **or** /opt/IBM/db2cmv8 on AIX, Linux, or Solaris).
 - Run the cmbenv81.bat (Windows) or cmbenv81.sh (AIX, Linux, or Solaris) file to set the environment and the classpath for the Java Utilities of II4C.

System Requirements

- Change the directory to *[II4C working directory]/cmgmt/connectors* where *[II4C working directory]* is one of the following paths:

(Windows) C:/Program Files/db2cmv8

(Linux, AIX) /home/ibmcmadm

(Solaris) /export/home/ibmcmadm

- Run the command

```
java com.ibm.mm.sdk.util.cmbsrvsictm -a add -s <library server database name> -sm <database schema name>
```

where *<library server database name>* is the same as Database Alias configured in step 6 above.

Note: The following procedure allows users without DB2 rights to share the connection credentials through the *cmbicmenv.ini* file.

2.2.8.3 Configure a multiuser connection to the IBM Content Manager datastore

- 1 From a command prompt window, change the directory to *[II4C home]/bin* (for example, C:\Program Files\db2cmv8\ on Windows **or** /opt/IBM/db2cmv8 on AIX, Linux, or Solaris).
- 2 Run the *cmbenv81.bat* (Windows) or *cmbenv81.sh* (AIX, Linux, or Solaris) file to set the environment and the classpath for the Java Utilities of II4C.
- 3 Change the directory to *[II4C working directory]/cmgmt/connectors*, where *[II4C working directory]* is one of the following paths:

(Windows) C:/Program Files/db2cmv8

(Linux, AIX) /home/ibmcmadm

(Solaris) /export/home/ibmcmadm

- 4 Run the command

```
java com.ibm.mm.sdk.util.cmbenvicm -a add -s <library server database name> -u <database user ID> -p <database password>
```

where *<library server database name>* is the same as Database alias configured in step 6 above.

2.2.9 Additional requirements for Connector for IBM FileNet

These requirements are optional and required only if you are installing Adobe® LiveCycle® 10 Connector for IBM® FileNet.

IBM FileNet 4.0

If LiveCycle is connecting to IBM FileNet 4.0 Content Engine, you must install the Content Engine Java Client. Use the IBM FileNet 4.0 content engine client installer located by default in C:\Program Files\FileNet\CEClient. During installation, select only the Java client component on the component selection screen.

For IBM FileNet 4.0 Process Engine settings, copy the *pe.jar* file from the Process Engine directory to the computer that will host LiveCycle. Create the directory C:\FileNetPE\files and copy the *pe.jar* file there. The Process Engine client install directory is now C:\FileNetPE.

IBM FileNet 4.5

If LiveCycle is connecting to IBM FileNet 4.5 Content Engine, you must install the Content Engine Java Client. Use the IBM FileNet 4.5 content engine client installer located by default in C:\Program Files\FileNet\CEClient. During installation, select at least one of the components from Application Engine or Process Engine on the component selection screen.

For IBM FileNet 4.5 Process Engine, you must install the IBM FileNet 4.5 Process Engine Client located by default in C:\Program Files\FileNet\BPMClient. During installation, select the **Other** option on the component selection screen.

IBM FileNet 5.0

If LiveCycle is connecting to IBM FileNet 5.0 Content Engine, you must install the Content Engine Java Client. Use the IBM FileNet 5.0 content engine client installer located by default in C:\Program Files\FileNet\CEClient. During installation, select at least one of the components from Application Engine or Process Engine on the component selection screen.

For IBM FileNet 5.0 Process Engine, you must install the IBM FileNet 5.0 Process Engine Client located by default in C:\Program Files\FileNet\BPMClient. During installation, select the Other option on the component selection screen.

2.3 LDAP configuration

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

If you do not have an existing LDAP server and database, install and configure your LDAP server and database according to the vendor's documentation. Make note of the LDAP administrator name and password to use during the LiveCycle configuration process. Configure LiveCycle to connect with the LDAP database after you install and deploy your services that are part of LiveCycle. This configuration is done by using the User Manager service.

See the [Installing and Deploying LiveCycle](#) document for your application server.

2.4 Global document storage directory

The global document storage (GDS) directory is used to store long-lived files that are used within a process as well as critical LiveCycle product components. The lifetime of long-lived files is intended to span multiple restarts of a LiveCycle system, and can span days and even years. These files may include PDF files, policies, or form templates.

Long-lived files are a critical part of the overall state of many LiveCycle deployments. Input documents for asynchronous job invocation are also stored in the GDS directory and must be available in order to process requests.

You must create the GDS directory before you initialize the LiveCycle database. See “[2.4.1 Location of the global document storage directory](#)” on page 17 for information.

2.4.1 Location of the global document storage directory

You configure the location of your GDS directory with Configuration Manager after you install LiveCycle. The GDS directory you specify should be highly available and should have low access time to enhance performance. If the GDS directory is on a shared network drive, it is recommended that you specify the location as \\computer_name\GDS.

If you must change the GDS directory location after completing the installation (see [LiveCycle Administration Help](#)), you should plan an appropriate location for the GDS directory.

Important: Module deployment will fail on Windows if the GDS directory is at the drive root (for example, D:\). For GDS, you must make sure that the directory is not located at the root of the drive but is located in a subdirectory. For example, the directory should be D:\GDS and not simply D:\.

2.4.2 Sizing factors for the global document storage directory

The size of the global document storage directory depends on expected LiveCycle usage factors for the deployment. You should allocate a minimum of 10 GB of disk space for the GDS directory.

The following factors also affect the sizing:

- The typical volume of documents that LiveCycle processes. Processing high volumes of documents requires a larger GDS directory.
- The typical size of documents that LiveCycle processes. Processing large documents requires a larger shared GDS directory.
- The complexity of documents that LiveCycle processes. Processing complex documents, such as documents that are processed by multiple services that are part of LiveCycle, require a larger GDS directory.

2.4.3 Securing the global document storage directory

Access to the GDS directory must be secure. The long-lived documents in this directory may contain sensitive user information, such as information that requires special credentials when accessed by using the LiveCycle SDK or user interfaces.

Use a security method that is appropriate to your operating system. It is recommended that only the operating system account that is used to run the application server has read and write access to this directory.

Note: Deleting files or directories from the GDS directory can render the LiveCycle server inoperative.

2.4.4 Backing up the global document storage directory

The global document storage directory should be backed up to allow administrators to restore LiveCycle in case of failure.

If the global document storage directory becomes unavailable or is lost due to failure, LiveCycle will not run until the GDS directory and database are restored by a consistent back up or LiveCycle is reinitialized with a new installation.

2.5 Installation considerations

2.5.1 Disabling Virus scans (Windows only)

To improve the speed of the installation, you may choose to disable any on-access virus scanning software for the duration of the installation.

2.5.2 Installing from network drives

It is recommended that you install LiveCycle only from the installation media or a local disk drive. Attempting to install the software over a network results in considerable delays in starting and installing. It is also likely that installing from a network drive will add to the length of the directory path, which will cause the LiveCycle installer to prevent the installation from proceeding.

2.5.3 Manual use of Acrobat restricted

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

2.5.4 Temporary directory

LiveCycle requires a temporary directory to store documents that are larger than the maximum inline size set for document objects. For detailed information on how you can improve the performance of your LiveCycle Server by specifying an appropriate value for this setting, see this [blog on the Document Max Inline Size performance knob](#).

(AIX, Linux, and Solaris only) If a non-root user is running the application server, the user must have read, write, and execute permissions on the specified temporary directory.

Important: (For cluster deployments only) Future upgrades might fail if you specify a shared network directory as the temporary directory.

2.6 LiveCycle IPv6 support

LiveCycle includes IPv6 support. The default configurations defined in the installation documentation for LiveCycle set IPv4 as the default IP protocol because this protocol has the most compatibility with third-party infrastructure.

Do not enable IPv6 unless your deployment must use it. The number of supported platform configurations is reduced when enabling IPv6 support with LiveCycle. You should verify that all third-party software, hardware, and networks that you plan to use have IPv6 support before you attempt to enable IPv6.

Note: If you are enabling CIFS in an IPv6 environment, you must explicitly enable IPv6 configuration after you configure your LiveCycle installation using Configuration Manager. See “Enabling CIFS in IPv6 mode” in the guide for your application server.

2.6.1 Supported IPv6 configurations

Not all infrastructure components support IPv6. For example, Oracle database does not support IPv6. You can use these databases by configuring the connection between the application server and the databases with IPv4, and the remaining communications over IPv6.

- Databases that support IPv6: Microsoft SQL Server 2005 and 2008, DB2 9.1 or later versions of 9.x
- Application servers that support IPv6: JBoss 5.1.0, Weblogic 11g, WebSphere 7.0.0.15.

2.6.2 IPv6 implementation guidelines

When you use IPv6 implementation either partially or fully, keep the following points in mind:

- After installing LiveCycle, do not use the option to start the Configuration Manager directly from the LiveCycle installer. Instead, navigate to the `[LiveCycle root]\configurationManager\bin\IPv6` directory, and run the IPv6-specific script (`ConfigurationManager_IPv6.bat` or `ConfigurationManager_IPv6.sh`) to launch the Configuration Manager.
- If you have chosen to validate the application server configuration using the Configuration Manager, the validation will fail after you enable IPv6 for the application server. You can ignore this error message during the process. After you restart the application server in the IPv6 mode, the application server can connect to the database.

System Requirements

- *(WebLogic only)* You need to enable IPv6 for the managed servers only. The Admin Server can continue to run on IPv4, and can be accessed with its IPv4 address. However, the managed server that you have started in IPv6 environment can only be accessed through its IPv6 address or a hostname resolved through DNS.
- *(WebLogic only)* Even if you are running Configuration Manager on the same computer that hosts the application server, you must provide the listen address of the managed server for bootstrapping and deploying LiveCycle modules. This listen address must be the DNS name that resolves to IPv6 address of the computer.
- To have a pure IPv6 communication with the database server, modify both EDC_DS and IDP_DS connection settings to use the hostname of the database which resolves to a numeric IPv6 address.
- Many software components such as database drivers do not completely support numeric IPv6 addresses. So, it is recommended that you use a DNS-resolved hostname instead of numeric IPv6 addresses.
- Ensure that name used for mapping IPv6 is added to the CSRF filter section. If the name is not added, see Preventing CSRF attacks in [Administration Help](#).

Note: Name used for mapping IPv6 must not contain square brackets ([]).

- In an IPv6 environment, if you are using Microsoft SQL Server, you should specify the database server IP address in the following format. Note that in this string, `serverName` is a keyword, and so must not be replaced with the actual server name.

```
jdbc:sqlserver://;serverName=<IPv6 address>; portNumber=<port>;databaseName=<db_name>
```

Here, instead of the numeric IPv6 address, you can specify the hostname of the SQL Server database.

2.6.3 Configuring IPv6 for JBoss

- 1 You can download and install JBoss from <http://www.jboss.org/jbossas/downloads/> or obtain the jboss.zip file from the third-party directory on the installation media and extract the bundled JBoss.
- 2 Modify adobe-ds.xml and the database-specific data source configuration file to connect to the LiveCycle database.
- 3 Modify the login-config.xml file to connect to the LiveCycle database.
- 4 Modify the following files to enable IPv6:
 - **(JBoss 5.1 on Windows)** `[appserver root]\bin\run.conf.bat`
 - **(JBoss 5.1 on other platforms)** `[appserver root]\bin\run.conf`
 - **(JBoss 4.2.1 on Windows)** `[appserver root]\bin\run.bat`
 - **(JBoss 4.2.1 on other platforms)** `[appserver root]\bin\run.sh`

Do the following in this file:

 - Change `-Djava.net.preferIPv4Stack=true` to `-Djava.net.preferIPv6Stack=true`
 - Add the `-Djava.net.preferIPv6Addresses=true` argument.
 - `[appserver root]\bin\service.bat`: Replace `-b 0.0.0.0` with `-b <hostname mapped to IPv6 address>`
- 5 Launch Configuration Manager by invoking the `[LiveCycle root]\configurationManager\bin\IPv6\ ConfigurationManager_IPv6.bat` or `ConfigurationManager_IPv6.sh` script.
- 6 In the Configuration Manager, select the steps to configure EAR files, bootstrap and deploy LiveCycle modules.
- 7 After the Configuration Manager process is completed, copy these EAR files to the `[appserver root]\server\<profile_name>\deploy` directory.
- 8 Start JBoss from a command line.

- 9 Provide Configuration Manager hostname of the computer that is mapped to its IPv6 address and then bootstrap the application server to deploy the LiveCycle modules.

2.6.4 Configuring IPv6 for WebLogic

- 1 Install LiveCycle using the installer.
- 2 Do not launch the Configuration Manager when the installer finishes. Launch Configuration Manager by invoking the `[LiveCycle root]\configurationManager\bin\IPv6\ConfigurationManager_IPv6.bat` or `ConfigurationManager_IPv6.sh` script.
- 3 Select the options to configure LiveCycle EARs, WebLogic application server, and verify the application server settings using Configuration Manager.

Note: You will get an error message that the Configuration Manager validation of data source has failed. This is because the application server is not yet started in the IPv6 mode, and the data source is now configured in the IPv6 mode. You can ignore this warning at this stage.

- 1 From the WebLogic Server Administration Console, change the application **Server Start** arguments of the managed server to enable IPv6.
 - Change `-Djava.net.preferIPv4Stack=true` to `-Djava.net.preferIPv6Stack=true`.
 - Add the `-Djava.net.preferIPv6Addresses=true` argument.
- 2 Change the listen address of the managed server to enable it using IPv6 address.
 - In the WebLogic Server Administration Console, select **Environment** > **Servers** > **[managed server name]Configuration** tab.
 - In the Listen Address field, enter the hostname of the computer. Ensure that this hostname resolves to the IPv6 address of this computer.
- 3 Save the changes, and then restart the managed server.
- 4 Launch Configuration Manager by invoking the `[LiveCycle root]\configurationManager\bin\IPv6\ConfigurationManager_IPv6.bat` or `ConfigurationManager_IPv6.sh` script.
- 5 In the Configuration Manager, select the steps to deploy EAR files, bootstrap, deploy LiveCycle modules.
- 6 Enter the same hostname as provided in the listen address field of the managed server.

Note: Even if you are running Configuration Manager on the same computer, you must provide the listen address of the managed server for bootstrapping and deploying LiveCycle modules.

2.6.5 Configuring IPv6 for WebSphere

- 1 Install LiveCycle using the installer script. After the installation is complete, do not start the Configuration Manager when the installer prompts.
- 2 Navigate to the `[LiveCycle root]\configurationManager\bin\IPv6` directory, and run the IPv6-specific script (`ConfigurationManager_IPv6.bat` or `ConfigurationManager_IPv6.sh`) to launch LCM.
- 3 Use the Configuration Manager options to configure EAR and the application server.
- 4 Follow the steps in the Configuration Manager to configure the application server. While configuring database, provide hostname of database that is mapped to IPv6 address.
- 5 Validate the application server configuration by Configuration Manager. Ignore the warning if data source validation fails. You can validate data sources from WebSphere Administrative Console.

System Requirements

- 6 On the WebSphere Administrative Console, select **Servers > Server Types > WebSphere application servers > [server name] > Java and Process Management > Process definition > Java Virtual Machine**. In **Generic JVM arguments**, add the `-Djava.net.preferIPv6Addresses=true` argument.
- 7 Select **Servers > Server Types > WebSphere application servers > [server name] > Java and Process Management > Process definition > Java Virtual Machine**, and click **Custom Properties**. Change `java.net.preferIPv4Stack=true` to `java.net.preferIPv6Stack=true`.
- 8 Manually deploy the EAR files to WebSphere Application Server by using WebSphere Administrative Console. Configured EAR files are available at `[LiveCycle root]/configurationManager/export` folder.
- 9 Restart WebSphere Application Server.
- 10 Navigate to the `[LiveCycle root]/configurationManager/bin/IPv6` directory, and run `ConfigurationManager_IPv6.bat` or `ConfigurationManager_IPv6.sh` to launch Configuration Manager.
- 11 In the Configuration Manager, select the options to bootstrap and deploy LiveCycle modules. Provide the application server hostname that is mapped to IPv6 address.

Note: After you start the application server in an IPv6 environment, (with the flag `-Djava.net.preferIPv6Stack=true`), you can access it only through its IPv6 address or hostname mapped to IPv6 address.

2.7 Central Migration Bridge service

The Central Migration Bridge service helps you to migrate existing applications from your Adobe Central Pro Output Server or Adobe Web Output Pak products to work with the Output service. The Central Migration Bridge service allows you to use your current IFD/MDF templates, data transformation scripts, and DAT files in a LiveCycle environment as part of an overall migration strategy.

Note: The Central Migration Bridge is useful only if you have existing Central Pro applications to migrate.

Authorized use of Central Migration Bridge

To use the Central Migration Bridge service, you must have a valid license for Central Pro Output Server 5.7 or an executed Central Pro Output Server 5.7 migration agreement. To install Central Pro Output Server 5.7, use your existing media and existing product authorization code (PAC). Your PAC is for a specific operating system platform; if this differs from the operating system platform on which you are installing LiveCycle, you will need to acquire a PAC for that operating system. Contact your Adobe representative for more information regarding migration, or acquiring Central Pro Output Server 5.7 media or PAC.

Installation considerations

The Central Migration Bridge service interacts directly with the Central Pro (version 5.7) executable files. You must have Central Pro installed on the same server as LiveCycle, but it is not a prerequisite to installing LiveCycle (that is, it can be installed before or after LiveCycle). Consult the Central Pro documentation set for installation instructions.

Important: Do not start Central Pro or change its properties to run automatically.

On Windows, the Central Pro service *Adobe Central Output Server* is installed as a manual service. Do not run the service or change its properties to run automatically.

On AIX, Linux, or Solaris, do not start the Central Pro daemon *jfdaemon*. If you edited the start-up script of your computer to launch *jfdaemon* when restarting the computer, change the script to prevent the daemon from being launched automatically. (See the Central Pro installation documentation.) Do not start Central from the command line by launching the *jfserver* process.

System Requirements

Note: *The LiveCycle User that invokes the Central Migration Bridge service must have access rights to the Central Pro install directory as well as execute permissions on the Central Pro executable files.*

LiveCycle turnkey installations

If you are installing and configuring a LiveCycle turnkey environment using Express mode, the Central Migration Bridge service will be installed and configured by default - you will not be prompted for any input.

Note: *Ensure that the Adobe Central Pro product is installed in the default directory.*

LiveCycle custom installations

If you are installing and configuring LiveCycle using the Custom mode (Partial turnkey or Manual), Configuration Manager will prompt you to include Central Migration Bridge in the deployment.

By default, the service will use the default Central Pro installation path. If Central Pro is installed in a different location, update the [Central Install Dir] configuration for the Central Migration Bridge Service by going to Administration Console.

When you have finished installing LiveCycle, if Central Pro is not installed in the default location, do the following steps to point LiveCycle to the correct directory:

- 1 Log in to Administration Console.
- 2 Click **Services > Applications and Services > Service Management**.
- 3 Click the **Central Migration Bridge:1.0** service.
- 4 Type the correct path to the Central Pro install directory.
- 5 Click **Save**.

Note: *This configuration can also be made in Workbench. See [Creating and managing processes](#).*

Chapter 3: LiveCycle Credentials and Certificates

This chapter describes how to do the following tasks:

- Obtain the Adobe® LiveCycle® Reader® Extensions 10 Rights credential
- Obtain digital certificates for use with Adobe® LiveCycle® Digital Signatures 10.

3.1 Obtaining the Reader Extensions Rights credential

The Reader Extensions Rights credential is a digital certificate that is specific to Reader Extensions that enables Adobe Reader usage rights to be activated in PDF documents. If the credential is not installed, Reader Extensions 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 that Reader Extensions processes. It is a critical 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 short validity period that is provided to customers who want to evaluate Reader Extensions. Usage rights applied to documents using this credential expire when the credential expires. This type of credential is valid only for two to three months.

Production: A credential with a long validity period that is provided to customers who purchased the full product. Production credentials 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, you receive an evaluation Rights credential from the sales representative you ordered the product from or from the website where you downloaded the evaluation product.

If your organization purchases a production version of Reader Extensions, 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 through a partner or software provider who integrated Reader Extensions 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).*

3.2 Obtaining digital certificates for use with Digital Signatures

Digital certificates are required for use with Digital Signatures. Although you can configure and manage digital certificates after you install and configure LiveCycle, obtaining them before you install ensures that you are ready to use LiveCycle 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*) that are 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 keys securely stored in an encrypted file or HSM.

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

Note: *You can configure and manage certificates, credentials, and Certification Revocation Lists (CRLs) for use with LiveCycle by using Trust Store Management, which is accessible through the web-based Administration Console. (See [Administration 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 that verify 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-encoded and DER-encoded CRLfiles.

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

Chapter 4: Creating the LiveCycle Database

4.1 Database configuration requirements

This section describes how to set up the database for use with LiveCycle. This section describes special tables, indexes, and other properties that are required in the LiveCycle database that are not configured by Configuration Manager.

The database will contain these elements:

- LiveCycle services
- LiveCycle run-time configurations
- LiveCycle process data
- Customer process definitions and templates
- Application server managed data

Before you create the database, you must ensure that you read the pre-installation requirements and have the required software installed.

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

4.2 Minimum database user permissions

Database	Initialization permissions	Runtime permissions
Oracle	CREATE SESSION CREATE TABLE CREATE VIEW CREATE SEQUENCE UNLIMITED TABLE SPACE	CREATE SESSION UNLIMITED TABLE SPACE (only needed if you do not configure user quotas) CREATE TABLE
MySQL	SELECT INSERT UPDATE DELETE CREATE DROP REFERENCES INDEX ALTER CREATE_TEMP_TABLE LOCK_TABLES	SELECT INSERT UPDATE DELETE
SQL Server - DB level	Create Table Create View Connect	Connect
SQL Server - Schema level	Alter Insert References Select Update Delete	Insert Select Update Delete
DB2	See “4.5.1 DB2 user account” on page 32 for a complete description.	See “4.5.1 DB2 user account” on page 32 for a complete description.

4.3 Creating an Oracle database

If you prefer not to use the default database that was created when you installed Oracle 11g, create a new database by using the Database Configuration Assistant tool. LiveCycle supports Oracle 11g on WebLogic, WebSphere, and JBoss application servers.

Note: You can use the *Transaction Processing* or *General Purpose* templates while configuring an Oracle database instance for LiveCycle. If you wish to use the *Custom Database* template for configuring a database instance, the minimum set of database components you must include are **Oracle JVM** and **Enterprise Manager Repository**.

Do the following when you create your Oracle database:

- Set the initial database size to a minimum of 500 MB.
- Create user quotas to allow the database to grow to accommodate persistent data from LiveCycle applications.
- Enable support for UTF-8 encoding.
- Set Database Character Set to Unicode (AL32UTF8), and the National Character Set is AL16UTF16 (Unicode UTF-16 universal character set).
- Set NLS_LENGTH_SEMANTICS to BYTE (if required). The database initialization fails if you set any other value.
- You must install Oracle using Transaction Processing and set the connection mode for the server to Dedicated Processing.

4.3.1 User account and rights

Create a new user account on the database and assign it the following system privileges:

- CREATE SEQUENCE
- CREATE VIEW
- UNLIMITED TABLESPACE
- CREATE TABLE
- CREATE CLUSTER
- CREATE SESSION

Important: (WebLogic only) When using WebLogic and an Oracle database, ensure that the database user name does not have a number as the first character, nor a hyphen (-) within the user name. Otherwise, the database does not bootstrap properly.

For deployments on AIX, Linux, or Solaris, the user name must not exceed 8 characters; 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 the database server
- Oracle LISTENER port number (default is 1521)

For information about using Oracle 11g, see the appropriate user documentation.

4.4 Creating a SQL Server database

You can create a SQL Server database that LiveCycle will use to store run-time and configuration data. For information about creating a SQL Server database, refer to the SQL Server documentation. LiveCycle supports SQL Server 2005 SP2 and 2008 (English and Japanese) on JBoss, WebLogic, and WebSphere.

Create a SQL Server database, and create a user account and assign it DB_OWNER privileges for use 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

4.4.1 Set up SQL Server for LiveCycle

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

4.4.1.1 Increase memory

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

Note: This section is recommended but optional.

- 1 Using Microsoft SQL Server Management Studio, connect to the database server where you will host the LiveCycle database.
- 2 Right-click the database server 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 on the server.
- 4 Restart the SQL Server database.

4.4.1.2 Set the processor priority

On dedicated database servers, which are recommended for production installations of LiveCycle, the SQL Server process is configured so that it does not consume too much of the system CPU resources.

Note: This section is recommended but optional.

- 1 Using Microsoft SQL Server Management Studio, connect to the database server where you will host the LiveCycle database.
- 2 Right-click the database server connection and select **Properties**.
- 3 Select the **Processors** page and select **Boost SQL Server Priority**.
- 4 Restart the SQL Server database.

4.4.1.3 Increase the recovery interval

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

Note: This section is recommended but optional.

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

- 4 Restart the SQL Server database.

4.4.1.4 Integrated security

Note: This is an optional configuration.

If you are using SQL Server integrated security, you can set your SQL Server database to Mixed Mode or Windows Authentication Mode. However, if you are using Windows Authentication Mode, you must configure integrated security on Windows to establish a trusted connection with SQL Server.

- For JBoss, see Configure Integrated Security on Windows.
- For WebLogic, see Configuring SQL Server database connectivity in [Installing and Deploying LiveCycle for WebLogic](#).
- For WebSphere, see Configuring SQL Server database connectivity in [Installing and Deploying LiveCycle for WebSphere](#).

4.4.2 Sizing your SQL Server database

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

Medium size deployments: Environments where the LDAP directory has approximately 100,000 users and 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 approximately 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 database to ensure that it does not lose its restricted space or the space available on the disks where the database resides.

4.4.3 Creating the LiveCycle database user, schema, and login

You are now ready to create the LiveCycle database user, schema, and login.

Important: Ensure that you use the `SQL_Latin1_General_CP1_CI_AS` collation (or the `Japanese_CI_AS` collation if your database will run in a Japanese environment) when you create the database instance for LiveCycle. Any other collation may cause your database initialization to fail. The collation for your LiveCycle database instance can be different from the collation used when creating the SQL Server database.

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

- 2 Enter the database name of your choice.

Note: The database name is very important, and the name chosen must be consistently used in the following procedures where a reference to `database_name` exists.

- 3 (SQL Server 2005 Only) 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 systems, specify 200 MB.
- For larger systems, see “[4.4.2 Sizing your SQL Server database](#)” on page 30.

- 5 In the **Database DataAutogrowth** box, enter 50%.

Creating the LiveCycle Database

- 6 In the **Database LogInitial Size** box, enter the appropriate value:
 - For small development or small production systems, specify 20 MB.
 - For larger systems, see “4.4.2 Sizing your SQL Server database” on page 30.
- 7 In the **Database LogAutogrowth** box, enter 50%.
- 8 Click **OK** to create the database.

4.4.3.1 Create the LiveCycle 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 must specify for the new user.

- 1 Using Microsoft SQL Server Management Studio, connect to the database server where you created the LiveCycle database.
- 2 Click **Server > Security**, and then right-click **Logins** and select **New Login**.
- 3 Enter the login name *[database_username]*, and then select **SQL Server Authentication** and type a new password.
- 4 Ensure that **Enforce Password Expiration, User must change password on next login** is also deselected.
- 5 Leave the default database as **Master**, and click **OK**.
- 6 Click **Server > Databases > [database_name] > Security**, and then right-click **Schemas** and select **New Schema**.
- 7 In the **Schema Name** box, type *[database_username]*, and click **OK**.
- 8 Click **Server > Databases > [database_name] > Security**, and then right-click **Users** and select **New User**.
- 9 In the New User dialog box, type the login name and user name *[database_username]*.
- 10 Set the default schema to *[database_username]*.

Note: The schema name should be the same as the [database_username].
- 11 In the Schemas Owned By This User area, select the schema *[database_username]*.
- 12 In the Database Role Membership area, select **db_owner**.

4.4.3.2 Associate the LiveCycle user with the database

After you create the LiveCycle user, associate it with the LiveCycle database.

- 1 Click **Security > Logins**, and then right-click *[database_username]* and select **Properties**.
- 2 In Login Properties, on the General page, set the user's default database to *[database_name]*.
- 3 Select the **User Mapping** page and, in the Users Mapped To This Login section, verify that *[database_name]* is selected, **User** is set to *[database_username]*, and **Default Schema** is set to *[database_username]*.
- 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 and then click **OK**.

4.4.3.3 Set the isolation level for the LiveCycle database

LiveCycle requires a specific isolation level to manage deadlocking when long-running transactions occur at the same time as numerous shorter reads.

Important: You must set the isolation level for MS SQL Server to avoid deadlocking issues.

- 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.

4.5 Creating a DB2 database

Create a DB2 database by running one of the scripts 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 less memory dedicated for the database, see the appropriate DB2 documentation for details about configuring your system settings. LiveCycle supports DB2 with WebSphere 7.0, and WebLogic 11g.

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 the database server
- DB2 port number

4.5.1 DB2 user account

LiveCycle requires a dedicated system database account. If the account is the schema owner no other privileges are required. The schema owner can alter, create and drop any object in the schema. If the account is not the schema owner then the ALTERIN, CREATEIN AND DROPIN schema privileges are required. The user will also have to belong to the DB2USERS group.

The DB2 user account needs to have schema owner rights to create tables both at product installation time and system runtime. The application server may add further requirements to the database user and schema due to its internal use of the database.

For information about creating a user account, see the DB2 documentation.

For deployments on AIX, Linux, or Solaris, the user name must not exceed 8 characters; on Windows, it must not exceed 12 characters.

4.5.2 Create the DB2 database

- ❖ On the computer that hosts DB2, create a new text file that includes the following DB2 script for your database version.

Note: The following text contains formatting characters for line breaks. When you copy this text to a location outside this document, remove the formatting characters.

Creating the LiveCycle Database

```
-- Create a database, using a custom USERSPACE1 definition
-- Configures the database for typical LC use. For large number of connections, add:
--     num_remote_apps
-- to the AUTOCONFIGURE section
-- To use: Globally replace (case sensitive) DB_NAME with the name of the database to be
created.
-- Modify the FILE locations based on system configuration
-- Note: The size of the file has to be specified in pages (4K pages in this case) when used
as part of the create database command
create database DB_NAME using codeset utf-8 territory default
USER TABLESPACE MANAGED BY DATABASE USING (FILE'C:\Db2\DB_NAME\DATA_4K_1.db2' 8192) AUTORESIZE
YES
AUTOCONFIGURE using workload_type mixed isolation CS admin_priority performance apply DB and
DBM;
-- Create required buffer pools. Let the DB manage the sizes based on load.
-- Creates 8K pool for LC
-- Let DB2 manage the sizes based on usage patterns
connect to DB_NAME;
CREATE BUFFERPOOL DB_NAME_BP8K IMMEDIATE SIZE AUTOMATIC PAGESIZE 8192;
ALTER BUFFERPOOL IBMDEFAULTBP IMMEDIATE SIZE AUTOMATIC;
connect reset;
-- Create required tablespaces for tables requiring 8K and 32K pools.
connect to DB_NAME;
-- 8K tablespace used by a number of tables.
-- Note that the majority of data is stored in the 4K tablespace (created as part of the database
create)
CREATE TEMPORARY TABLESPACE DB_NAME_TEMP_8K IN DATABASE PARTITION GROUP IBMTEMPGROUP PAGESIZE
8192 MANAGED BY SYSTEM
USING ('C:\Db2\DB_NAME\TEMP_8K') PREFETCHSIZE AUTOMATIC BUFFERPOOL DB_NAME_BP8K;
-- Start off large enough to allow a reasonable test to run without needing to extend the space
-- Should be modified to fit expected usage patterns
CREATE LARGE TABLESPACE DB_NAME_DATA_8K IN DATABASE PARTITION GROUP IBMDEFAULTGROUP PAGESIZE
8192 MANAGED BY
DATABASE USING (FILE'C:\Db2\DB_NAME\DATA_8K.db2'1G)
PREFETCHSIZE AUTOMATIC BUFFERPOOL DB_NAME_BP8K AUTORESIZE YES INCREASESIZE 1 G;
commit work;
connect reset;
deactivate database DB_NAME;
activate database DB_NAME;
```

1 Make the following changes to the script:

- Replace the instances of *dbname* and *DBNAME* with the name you want for the LiveCycle database.
- If you are creating multiple database instances on the same host, create an uniquely named buffer pool (for example, *BP8K_1*) for each database instance, using a buffer pool size such that all will not exceed 10% of total physical memory. For example, on a machine with 1GB of physical memory, the total buffer pool size should not exceed 100MB ("SIZE 100000")
- 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 you initialize the database.

- 2 Save the text file in a location that DB2 Command Line Processor can access.
- 3 Open a DB2 command prompt and type the following command to run the script:

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

4.5.3 Configure DB2 for concurrent usage

- 1 Open the DB2 Control Center:
 - (Windows) Select **Start > Programs > IBM DB2 > General Administration Tools > Control Center**.
 - (AIX, Linux, and Solaris) From a command prompt, enter the command `db2jcc`.
- 2 In the DB2 Control Center object tree, click **All Databases**.
- 3 Right-click the database created for LiveCycle products and click **Configuration Advisor**.
- 4 Follow the steps in the Configuration Advisor wizard and set the following properties:

DB2 Property	Required Value
Workload type	Mixed
Average number of SQL transactions per unit of work	Less than 10
Database Administration Priority	Faster transaction performance
Populated Database	Set according to the current state of the database instance. The instance is not populated if it has not yet been initialized by the Configuration Manager.
Isolation Level	Read Committed

- 5 Click **Finish**.

4.5.4 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**.
- 3 Select **Add User** and select the user.
- 4 Under Authorities, select **Connect to database**, **Create tables**, and **Database administration authority**, and then select **Apply**.
- 5 Click **OK**.

4.6 Creating a MySQL database

Note: The information contained in this document is meant for users who are installing MySQL manually and not for a turnkey installation. See *Installing and Deploying LiveCycle for JBoss Using Turnkey*.

Use the MySQL tools to create a MySQL 5 database for use with LiveCycle and a MySQL user account that the application server can use to connect to the database. You also must modify the MySQL database server configuration. For information about creating the database and user account, see the MySQL documentation. LiveCycle supports MySQL 5 with JBoss 5.1.

Note: MySQL does not support the use of special characters or spaces in the user name or password. Ensure that your user name and password adhere to this restriction. Also, to avoid Scheduler errors, do not use a dash (-) in the MySQL database name.

Note: On AIX, Linux, or Solaris, you must set the `lower_case_table_names` system variable to 1 to ensure that table names are case-insensitive. On Windows systems, this parameter does not have any effect. For more information about setting table names for case-sensitivity, see <http://dev.mysql.com/doc/refman/5.1/en/identifier-case-sensitivity.html>.

Note: To set up a database schema and new users on MySQL by 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 in the Variables section of the MYSQL UI.

You 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

4.6.1 Limitation on document usage in processes for MySQL databases

The MySQL database limits the amount of data that can be stored in a table row. Processes that involve multiple instances of documents can require more storage space than the MySQL row size limitation. Because LiveCycle stores data for a process instance in a single table row, processes that involve large amounts of data can exceed the MySQL limitation on row size, causing errors to occur.

If you run a process that involves many instances of documents and errors occur in the database, you can redesign the process so that the operations that handle documents execute in several subprocesses. For example, you can redesign your process so that its operations are executed in two processes. The first process includes the first half of the operations of the original process, and the second process includes the last half of the operations. The first process must include the invoke operation of the second process so that the second process runs as a subprocess.

4.6.2 MySQL user account

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

- SELECT
- INSERT
- UPDATE
- DELETE
- CREATE
- DROP
- REFERENCES
- INDEX

- ALTER
- CREATE_TMP_TABLE
- LOCK_TABLES

4.6.3 MySQL initial server configuration requirements

The following configuration setting is required in the `mysqld` section of the `my.ini` file (Windows) or `/etc/my.cnf` file (AIX, Linux, or Solaris):

- `max_allowed_packet=25M`

Note: Failure to configure this setting will result in “packet too large” errors that will cause module deployment to fail.

These configuration settings are recommended in the `mysqld` section of the `my.ini` file (Windows) or `/etc/my.cnf` file (AIX, Linux, or Solaris):

- `key_buffer_size=64M`
- `sort_buffer_size=1M`
- `default-storage-engine= INNODB`
- `innodb_buffer_pool_size=100M`
- `innodb_additional_mem_pool_size=5M`
- `innodb_log_file_size=170M`
- `innodb_log_buffer_size=8M`
- `innodb_flush_log_at_trx_commit=2`
- `query_cache_type=2`
- `default-character-set=utf8`
- `read_buffer_size=1M`
- `read_rnd_buffer_size=1M`

Chapter 5: Configuring a JBoss Application Server

5.1 Configuration options

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

- Turnkey with JBoss 5.1 (Windows only)
- Adobe preconfigured JBoss with JBoss 5.1
- Manual JBoss configuration with JBoss 5.1 (See “[Appendix - Manually Configuring JBoss](#)” on page 63.)

This section covers the use of the Adobe preconfigured JBoss Application Server when you are not using the turnkey installation option. You need to use the non-turnkey method for any of the following scenarios:

- You are using a Linux or Solaris operating systems.
- You are using Oracle or SQL Server, or if you don’t want to use the MySQL that install by default in a turnkey setup.
- You need advanced control of the JBoss configurations.

LiveCycle runs on JBoss on Windows Server 2008 R1 and R2 (Enterprise Edition or Standard Edition), Red Hat Enterprise Linux 5.5, Red Hat Enterprise Linux Advanced Platform 5.5, Solaris 10, and SUSE Enterprise Linux 11 platforms.

Note: If you are planning to use the LiveCycle 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 for JBoss Using Turnkey](#).

5.1.1 Notes about configuring JBoss

The JBoss Application Server is configured using various XML configuration files. JBoss must be shut down before editing any of these configuration files. If JBoss is running and these files are changed, JBoss will likely crash. On Linux or Solaris, JBoss also has several `.property` configuration files that must be UNIX text files. It is important to take care if they are edited on Windows environments at any time.

5.1.2 Adobe-preconfigured JBoss

The Adobe-preconfigured JBoss software is delivered as part of the LiveCycle media. This is the simplest option for installing JBoss because advanced knowledge of JBoss is not required.

If you intend to download a supported version of JBoss from the internet, you must follow the configuration steps in “[Appendix - Manually Configuring JBoss](#)” on page 63.

Adobe downloads EAP 5.1 archive from the Red Hat website and configures it with the options required to run LiveCycle in a stand-alone configuration. You can review these configurations in the next section.

Note: JBoss 5.1 deployment creates distinct directories for the data sources for each database type under the `[appserver root]/server/` folder. Each database is prefixed with “lc_”. Thus, data source files for MySQL, Oracle, and MS SQL databases are placed in `lc_mysql`, `lc_oracle`, and `lc_sqlserver` directories, respectively. In the following sections, `lc_<db-name>` is used to designate these distinct directories.

If you download the Electronic Software Distribution (ESD) from the Adobe web site, extract the lces_server_10_0_2_jboss_all_win.zip (Windows) or lces_server_10_0_2_jboss_all_unix.tar.gz (Linux or Solaris) file to your file system. After the main archive file is extracted, extract the following directories: merge-modules, server, and third_party.

Note: Be sure to keep the directory hierarchy unchanged from the original ESD file.

The preconfigured JBoss software is located on the installation media or electronic software distribution (ESD) in the /third_party directory. The jboss.zip file contains the Adobe pre-configured JBoss 5.1.

Depending on your installation requirements, copy the JBoss directories and their contents to the location where you intend to install JBoss. The JBoss installation has all the configuration steps completed except for the steps detailed in “5.4 Database connectivity for Adobe pre-configured JBoss 5.1” on page 40, which you must complete if you plan to use MySQL. If you do not intend to use a MySQL database, you must complete the appropriate configuration steps for your database detailed later in this chapter.

Note: The preconfigured JBoss includes a number of JSafe JAR files, which provide cryptography and Public Key Infrastructure (PKI) related FIPS-certified implementations in LiveCycle. When you install the product files, these files are deployed in the [appserver root]/server/lc_<db-name>/lib directory. Depending on the class loading, these JSafe JARs are used, even if your applications have their own version of JSafe JAR files.

5.2 Installing JDK for JBoss

You must download and install Oracle JDK 6.0 update 26 or later updates to 6.0 versions from <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Create or set the JAVA_HOME environment variable to point to the location where the Java JDK is installed.

5.2.1 Set the JAVA_HOME environment variable (Windows)

- 1 Select **Start > Control Panel > System**.
- 2 Click the **Advanced** tab and click **Environment Variables**.
- 3 In the **New System Variable** area, click **New**.
- 4 In the **New System Variable** box, type `JAVA_HOME` as the variable name and enter the directory where you installed the Java JDK as the value. This directory contains the /bin subdirectory. For example, type the following path:

```
C:\Program Files\Java\jdk1.6.0_26
```

5.2.2 Set the PATH environment variable (Windows)

- 1 Select **Start > Control Panel > System**.
- 2 Click the **Advanced** tab and click **Environment Variables**.
- 3 In the System Variables area, select **Path** and click **Edit**.
- 4 Add the following text at the beginning of the value:

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

5.2.3 Set the JAVA_HOME environment (Linux and Solaris)

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

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

5.2.4 Set the PATH environment variable (Linux or Solaris)

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

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

5.2.5 (Windows, Linux, or Solaris) Verify your JAVA_HOME environment variable setting

(Optional) Open a command prompt and run the following command:

```
java -version
```

You should receive a response that begins with the Java version you have installed (for example, 1.6.0_26).

5.2.6 (Solaris) Modify the run.conf file

For JVM to run in 64-bit mode, add the **-D64** JVM argument to the following line in the `[appserver root]/bin/run.conf` file.

```
JAVA_OPTS="$JAVA_OPTS -Xms1024m -Xmx2048m -XX:MaxPermSize=512m -
Dorg.jboss.resolver.warning=true -Dsun.rmi.dgc.client.gcInterval=3600000 -
Dsun.rmi.dgc.server.gcInterval=3600000 -Dsun.lang.ClassLoader.allowArraySyntax=true"
```

5.3 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 JBoss software.

All JBoss start configurations are located in the `[appserver root]/server` directory. For Adobe-preconfigured JBoss, the start configuration to call depends on the database you have installed, those being `lc_mysql`, `lc_oracle`, and `lc_sqlserver`.

5.3.1 Start JBoss

1 From a command prompt, go to `[appserver root]/bin`.

2 Start the application server by typing the following command:

- (Windows) `run.bat -c [profile_name] -b [server_IP_Address]`
- (Linux and Solaris) `./run.sh -c [profile_name] -b [server_IP_Address]`

where `[profile_name]` is the configuration required for your database and `[server_IP_Address]` is the IP address of the server.

Note: Adobe pre-configured JBoss must be bound to a specific IP address or all interfaces using `-b 0.0.0.0` if it is to be accessed remotely. If a hostname or IP address is used, connecting to JBoss with `localhost` in the URL will not work.

For example, to start your preconfigured JBoss instance for SQL Server database running on Windows, type:

```
run.bat -c lc_sqlserver -b [server_IP_Address]
```

5.3.2 Stop JBoss

- 1 From a command prompt, go to `[appserver root]/bin`.
- 2 Stop the application server by typing the following command:
 - (Windows) `shutdown.bat -s [server name]:<jndi -port>`
 - (Linux and Solaris) `./shutdown.sh -s [server name]:<jndi -port>`

5.3.3 Logon scripts for Linux and Solaris

To prepare the environment for production usage, it is recommended for production use cases that the environment be set up automatically during user login. This requires that the `JAVA_HOME` and `PATH` environment variables are set in the users login scripts for the user that the JBoss process will run as.

5.4 Database connectivity for Adobe pre-configured JBoss 5.1

To configure a connection to the LiveCycle database, you must complete the following tasks:

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

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

Note: Before proceeding with the following tasks, ensure that Jboss is not running.

5.4.1 Configuring MySQL for Adobe pre-configured JBoss 5.1

To enable Adobe pre-configured JBoss to connect to the MySQL database that stores LiveCycle data, you must create a data source file and deploy it to the instance of JBoss where you will deploy LiveCycle.

Note: Encrypt the password in the data source files (`adobe-ds.xml` and `mysql-ds.xml`) and the `login-config.xml` file using one of the methods described at <http://community.jboss.org/wiki/EncryptingDataSourcePasswords>. You can also use the instructions available on http://blogs.adobe.com/livecycle/2009/10/livecycle_-_encrypting_clearite.html.

5.4.1.1 Edit `adobe-ds.xml` file

Before configuring the MySQL data source, you must have already created the database on MySQL as described in “[4.6 Creating a MySQL database](#)” on page 34.

- 1 Open the `[appserver root]/server/lc_mysql/deploy/adobe-ds.xml` file in a text editor and locate this line for both IDP_DS and EDC_DS:

```
<connection-url>jdbc:mysql://localhost:3306/adobe</connection-url>
<driver-class>com.mysql.jdbc.Driver</driver-class>
<security-domain>MySQLDbRealm</security-domain>
```
- 2 Replace the following text in the file 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 data. Replace the default value, `adobe`, with your database name.
- 3 In the lines that follow the `<connection-url>` settings, locate the `user-name` and `password` settings and replace the default values with the user name and password that the application server uses to access your database.
 - 4 Ensure that the minimum and maximum values for the data source connections are set as follows:
 - For `IDP_DS`:

```
<min-pool-size>1</min-pool-size>
<max-pool-size>30</max-pool-size>
```
 - For `EDC_DS`:

```
<min-pool-size>1</min-pool-size>
<max-pool-size>20</max-pool-size>
```
- Note:** If your LiveCycle server handles heavy load, increase the maximum number of JDBC connections to ensure that all jobs are processed. In such cases, increase `<max-pool-size>` to 50 or more for both `IDP_DS` and `EDC_DS`.
- 5 Save and close the file.

5.4.1.2 Edit `mysql-ds.xml` file

If you are running LiveCycle with a MySQL database, you must set MySQL as the default data source for JBoss. This procedure assumes that the MySQL JDBC driver is installed in the `[appserver root]/server/lc_mysql/lib` directory.

- 1 Open the `[appserver root]/server/lc_mysql/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://localhost:3306/adobe</connection-url>
<driver-class>com.mysql.jdbc.Driver</driver-class>
<security-domain>MySQLDbRealm</security-domain>
```
- 2 Replace the following text in the file with values that are specific to your database:
 - **localhost:** Replace this value with the name of the server hosting the database.
 - **3306:** Replace this value with the port number for your database server.
 - **adobe:** Replace this value with the database that will connect with LiveCycle.
- 3 In the lines that follow the `<connection-url>` settings, locate the `user-name` and `password` settings and replace the default values with the user name and password that the application server uses to access your database.
- 4 Save and close the file.

5.4.1.3 Edit `login-config.xml` file

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

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

- 2 Replace the bold text with values that are specific to your database so that the application server can access your database.
- 3 Save and close the file.
- 4 Start JBoss.

5.4.2 Configuring Oracle for Adobe-preconfigured JBoss 5.1

To enable JBoss to connect to the Oracle database that stores LiveCycle data, you must create a data source file and deploy it to the instance of JBoss where you will deploy LiveCycle.

***Note:** Encrypt the password in the data source files (*adobe-ds.xml* and *oracle-ds.xml*) and the *login-config.xml* file using one of the methods described at <http://community.jboss.org/wiki/EncryptingDataSourcePasswords>. You can also use the instructions available on http://blogs.adobe.com/livecycle/2009/10/livecycle_-_encrypting_cleararte.html.*

5.4.2.1 Edit *adobe-ds.xml* file

- 1 Open the *[appserver root]/server/lc_oracle/deploy/adobe-ds.xml* file in a text editor and locate this line:

```
<connection-url>jdbc:oracle:thin@localhost:1521:adobe</connection-url>
<driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
<security-domain>OracleDbRealm</security-domain>
```

- 2 Replace the following text from the above line 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 SID of the database that stores the LiveCycle data. Replace the default value, *adobe*, with your database SID.
- 3 In the lines that follow the *<connection-url>* settings, locate the *user-name* and *password* settings and replace the default values with the user name and password that the application server uses to access your database.
- 4 (Only for Oracle RAC) Replace the connection URL mentioned in step 1 with the following connection URL:

```
jdbc:oracle:thin:@(DESCRIPTION=(ENABLE=broken) (ADDRESS_LIST=(ADDRESS= (PROTOCOL=TCP)
(HOST=yourhost1) (PORT=1521)) (ADDRESS=(PROTOCOL=TCP) (HOST=yourhost2) (PORT=1521))
(LOAD_BALANCE=on) (FAILOVER=on)) (CONNECT_DATA=(SERVER=dedicated)
(SERVICE_NAME=service.yourcompany.com) (FAILOVER_MODE=(TYPE=session) (METHOD=basic)
(RETRIES=10) (DELAY=3))))
```

***Note:** Ensure that this entry appears as a single line in the *adobe-ds.xml* file.*

5 (Only for Oracle RAC) Replace the following text from the connection URL in step 4 with values that are specific to your database:

- **yourhost1:** The name, IP address, or fully-qualified domain name of the first node in the cluster that hosts the database.
- **yourhost2:** The name, IP address, or fully-qualified domain name of the second node in the cluster that hosts the database.

***Note:** The cluster hosting the database could have *n* nodes. **yourhost1** and **yourhost2** are examples in the case of a two-node cluster.*

- **service.yourcompany.com:** The service name for the Oracle RAC database.

6 Ensure that the minimum and maximum values for the data source connections are set as follows:

- For IDP_DS:

```
<min-pool-size>1</min-pool-size>  
<max-pool-size>30</max-pool-size>
```

- For EDC_DS:

```
<min-pool-size>1</min-pool-size>  
<max-pool-size>20</max-pool-size>
```

***Note:** If your LiveCycle server handles heavy load, increase the number of maximum JDBC connections to ensure that all jobs are processed. In such cases, increase `<max-pool-size>` to 50 or more for both IDP_DS and EDC_DS.*

7 Save and close the file.

5.4.2.2 Edit oracle-ds.xml file

If you are running LiveCycle with an Oracle database, you must set Oracle as the default data source for JBoss. This procedure assumes that the Oracle JDBC driver is installed in the `[appserver root]/server/lc_oracle/lib` directory.

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

```
<jndi-name>DefaultDS</jndi-name>  
<connection-url>jdbc:oracle:thin@localhost:1521:adobe</connection-url>  
<driver-class>oracle.jdbc.driver.OracleDriver</driver-class>  
<security-domain>OracleDbRealm</security-domain>
```

2 Replace the bold text in the above lines with values that are specific to your database:

localhost: 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.

adobe: Replace this value with your database SID.

3 In the lines that follow the `<connection-url>` settings, locate the `user-name` and `password` settings and replace the default values with the user name and password that the application server uses to access your database.

4 (Only for Oracle RAC) Replace the connection settings mentioned in the first step with the following connection URL:

```
jdbc:oracle:thin:@(DESCRIPTION=(ENABLE=broken) (ADDRESS_LIST=(ADDRESS= (PROTOCOL=TCP)  
(HOST=yourhost1) (PORT=1521)) (ADDRESS=(PROTOCOL=TCP) (HOST=yourhost2) (PORT=1521))  
(LOAD_BALANCE=on) (FAILOVER=on)) (CONNECT_DATA=(SERVER=dedicated)  
(SERVICE_NAME=service.yourcompany.com) (FAILOVER_MODE=(TYPE=session) (METHOD=basic)  
(RETRIES=10) (DELAY=3))))
```

Note: Ensure that this entry appears as a single line in the `oracle-ds.xml` file.

- 5 (Only for Oracle RAC) Replace the following text from the connection URL in the previous step with values that are specific to your database:

- **yourhost1:** The name, IP address, or fully-qualified domain name of the first node in the cluster that hosts the database.
- **yourhost2:** The name, IP address, or fully-qualified domain name of the second node in the cluster that hosts the database.

Note: The cluster hosting the database could have *n* nodes. **yourhost1** and **yourhost2** are examples in the case of a two-node cluster.

- **service.yourcompany.com:** The service name for the Oracle RAC database.

- 6 Save and close the file.

5.4.2.3 Edit login-config.xml file

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

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

- 2 Replace the bold text with values that are specific to your database so that the application server can access your database.

Note: For Oracle RAC, replace **adobe** with the service name specific to your database.

- 3 Save and close the file.
- 4 Start JBoss.

5.4.3 Configuring SQL Server for Adobe-preconfigured JBoss 5.1

To enable JBoss to connect to the SQL Server database that stores LiveCycle data, you must create a SQL Server data source file and deploy it to the instance of JBoss where you will deploy LiveCycle (for example, `[appserver root]/server/lc_sqlserver/deploy`).

Note: Encrypt the password in the data source files (`adobe-ds.xml` and `mssql-ds.xml`) and the `login-config.xml` file using one of the methods described at <http://community.jboss.org/wiki/EncryptingDataSourcePasswords>. You can also use the instructions available on http://blogs.adobe.com/livecycle/2009/10/livecycle_-_encrypting_clearte.html.

5.4.3.1 Edit adobe-ds.xml file

- 1 Open the `[appserver root]/server/lc_sqlserver/deploy/adobe-ds.xml` file in a text editor and locate this line:


```
<connection-url>jdbc:sqlserver://localhost:1433;DatabaseName=adobe</connection-url>  
<driver-class>com.microsoft.sqlserver.jdbc.SQLServerDriver</driver-class>  
<security-domain>MSSQLDbRealm</security-domain>
```

2 Replace the following text from the above line 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 data. You will need to update the default value, adobe, with your database name.

3 In the lines that follow the <connection-url> settings, locate the user-name and password settings and replace the default values with the user name and password that the application server uses to access your database.

4 Ensure that the minimum and maximum values for the data source connections are set as follows:

- For IDP_DS:

```
<min-pool-size>1</min-pool-size>  
<max-pool-size>30</max-pool-size>
```

- For EDC_DS:

```
<min-pool-size>1</min-pool-size>  
<max-pool-size>20</max-pool-size>
```

Note: If your LiveCycle server handles heavy load, increase the number of maximum JDBC connections to ensure that all jobs are processed. In such cases, increase <max-pool-size> to 50 or more for both IDP_DS and EDC_DS.

5 Save and close the file.

5.4.3.2 Edit mssql-ds.xml file

If you are running LiveCycle with a MS SQL Server database, you must set SQL Server as the default data source for JBoss. This procedure assumes that the SQL Server JDBC driver is installed in the [appserver root]/server/lc_sqlserver/lib directory.

1 Open the [appserver root]/server/lc_sqlserver/deploy/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://localhost:1433;DatabaseName=adobe</connection-url>  
<driver-class>com.microsoft.sqlserver.jdbc.SQLServerDriver</driver-class>  
<security-domain>MSSQLDbRealm</security-domain>
```

2 Replace the bold text in the above lines with values that are specific to your database:

localhost: Replace this value with the host name of your SQL Server.

1433: If SQL Server is not using the default port, replace this value with the appropriate port number.

adobe: Replace this value with your SQL Server database name.

3 In the lines that follow the <connection-url> settings, locate the user-name and password settings and replace the default values with the user name and password that the application server uses to access your database.

Note: The JDBC driver for SQL Server does not parse numeric IPv6 addresses. If you are employing the SQL Server in an IPv6 environment, the server name must be specified in the following format:

```
<connection-url>jdbc:sqlserver://;serverName=<Numeric IPv6 address of the database  
host>:1433;databaseName=<db-name></connection-url>
```

Note: When you specify the IPv6 address of the server in this format, you must note that the string `;serverName` is a keyword; it must not be replaced with the actual server name.

- 4 Save and close the file.

5.4.3.3 Edit login-config.xml file

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

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

- 2 Replace the bold text with values that are specific to your database so that the application server can access your database.
- 3 Save and close the file.
- 4 Start JBoss.

5.4.3.4 Map the Windows login to the LiveCycle database user

- 1 Using Microsoft SQL Server Management Studio, connect to the database server that hosts the LiveCycle database.
- 2 Set the Authentication mode to **Windows authentication**.
- 3 Under **Security > Logins**, create a new account for the Windows domain user and select **Windows authentication**.
- 4 Click **User Mapping** on the Login - New screen and set the database and default schema for the new user.
- 5 Select **db_owner** as the Database role and click **OK**.

To verify that you created the user, expand the LiveCycle database in the tree and open **Security > Users**. The new user appears in the list of users.

5.4.3.5 Install the SQL Server database driver

Note: If you have not done so already, download the SQL Server JDBC Driver 3.0 (for all platforms) from the Microsoft Download Center or copy the driver from `[DVD root]/third_party/` to the `[appserver root]/server/<profile_name>/lib` directory.

If you are installing from Microsoft Download Center, follow the instructions on the website for downloading and installing the driver. Make a note of the directory location where you install the driver on your system.

Note: Use SQL Server JDBC Driver 3.0 for both Microsoft SQL Server 2005 SP2 and Microsoft SQL Server 2008.

5.4.3.6 Configure integrated security on Windows

Note: You must run the JBoss service as the Windows user configured in “[5.4.3.4 Map the Windows login to the LiveCycle database user](#)” on page 46. If JBoss is not running as a service, you must log in to Windows as that particular user before you start JBoss from a command prompt.

- 1 Modify the adobe-ds.xml and mssql-ds.xml files, located in `[appserver root]\server\lc_sqlserver\deploy`, to add `integratedSecurity=true` to the connection URL as shown in this example:

```
jdbc:sqlserver://<hostname>:<port>;databaseName=<db-name>;integratedSecurity=true.
```
- 2 Replace the values shown in bold in the above example with values appropriate to your database server.
- 3 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 within the Microsoft SQL JDBC 3.0 driver installation. The default location is `[SQL_root]/sqljdbc_3.0/enu/auth/x86` for 32-bit operating systems and `[SQL_root]/sqljdbc_3.0/enu/auth/x64` for 64-bit operating systems.

Note: Use SQL Server JDBC Driver 3.0 for Microsoft SQL Server 2008.

- 4 Open the properties for the JBoss for Adobe LiveCycle ES3 service or the JBoss service that you configured, and click the **Log On** tab.
- 5 Select **ThisAccount** and type the value of the user account that you mapped in “[5.4.3.4 Map the Windows login to the LiveCycle database user](#)” on page 46. If you are running JBoss from the command line, you must be logged in as this user.

5.5 Next steps

Install LiveCycle by following the instructions provided in [Installing and Deploying LiveCycle for JBoss](#).

Chapter 6: Configuring a WebSphere Application Server

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

6.1 Installing WebSphere

You must install WebSphere Application Server for running LiveCycle products. For use with LiveCycle, WebSphere can be installed either as WebSphere Base or as WebSphere ND with one base profile. See the WebSphere documentation to determine which method is best for your system.

6.1.1 Installing a Fix Pack to WebSphere

After you install WebSphere 7.0, you must update to WebSphere 7.0.0.15 and SR 9 before you deploy LiveCycle.

See the [WebSphere support website](#) for Fix Pack installation instructions and to access the WebSphere updates.

6.1.2 Installing WebSphere Feature Pack for JPA 2.0

After you install WebSphere 7.0, you must install WebSphere Feature Pack for JPA 2.0. See, [WebSphere Support website](#) and search *Installing and uninstalling the feature pack* to download and install Feature Pack.

6.1.3 JAVA_HOME and PATH environment variables

As part of your WebSphere installation, a Java SDK (JDK) was installed. The JAVA_HOME and PATH environment variables can point to the JDK where LiveCycle will be deployed.

6.1.3.1 Set the JAVA_HOME environment variable (Windows)

- 1 Select **Start > Control Panel > System**.
- 2 Click the **Advanced** tab and click **Environment Variables**.
- 3 In the System Variables area, click **New**.
- 4 Type `JAVA_HOME` as the variable name and, as its value, specify the directory where the JDK bundled with WebSphere is installed. For example, type the following path:

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

6.1.3.2 Set the PATH environment variable (Windows)

- 1 Select **Start > Control Panel > System**.
- 2 Click the **Advanced** tab and click **Environment Variables**.
- 3 In the System Variables area, select **Path**, click **Edit**, and then append the following text to the beginning of the variable value:

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

6.1.3.3 Set the DISPLAY setting (AIX)

- If you are using a Windows machine to access the AIX console, type the following text at the command prompt:

```
export DISPLAY= <IP Address of the Windows machine telnet, or cygwin to Linux or  
Solaris machine>:0.0
```

*Note: If the DISPLAY setting is not set correctly, the following error occurs when you start Configuration Manager:
"No JRE is found..."*

6.1.3.4 Set the JAVA_HOME environment variable (AIX, Linux, and Solaris)

- Set the JAVA_HOME variable for Bourne and Bash shells as shown in the following example:

```
JAVA_HOME=/opt/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.

6.1.3.5 Set the PATH environment variable (AIX, Linux, and Solaris)

- Set the PATH variable for Bourne and Bash shells as shown in the following example:

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

6.1.3.6 (Windows, AIX, Linux, or Solaris) Verify JAVA_HOME environment variables

(Optional) Open a command prompt and run the following command:

```
java -version
```

You should receive a response that displays the Java version installed.

6.1.4 Starting WebSphere Base and accessing WebSphere Administrative Console

6.1.4.1 Start WebSphere Base

- ❖ If the server is not already running, start the WebSphere Administrative Console, from `[appserver root]/profiles/<profile_name>/bin`, type the appropriate command, replacing **server1** with the name of your server instance:

- (Windows) `startServer.bat server1`
- (AIX, Linux, and Solaris) `./startServer.sh server1`

6.1.4.2 Access the WebSphere Administrative Console

1 Start the WebSphere Application Server.

2 In the address bar of a web browser, type the appropriate URL:

```
http://[host name]:[port]/ibm/console
```

If you are logged into the server that you are currently using, you can replace `[hostname]` with `localhost`. The port value depends on the application server and whether Administrative Security is enabled. The default port value for WebSphere is 9060. If Administrative Security is enabled, the default SSL port value is 9043.

1 If WebSphere Administrative Security is enabled, type the WebSphere user ID and password in the boxes provided.

- 2 Click **Log In**.

6.1.5 Starting WebSphere ND and accessing WebSphere Administrative Console

- 1 Open a command prompt and navigate to `[WebSphere ND root]/profiles/Dmgr01/bin`.
- 2 **(Windows)** Run `startManager.bat`.
(AIX, Linux, and Solaris) Run `startManager.sh`.
- 3 Navigate to `[WebSphere ND root]/profiles/[profile name]/bin` and run `startNode.bat`.
- 4 Once the services have started, connect to the server by typing the URL `http://[hostname]:[port]/ibm/console` in the address bar of a web browser.
If you are logged into the server that you are currently using, you can replace `[hostname]` with `localhost`. The port value depends on the application server and whether Administrative Security is enabled. The default port value for WebSphere is 9060. If Administrative Security is enabled, the default SSL port value is 9043.
- 5 If WebSphere Administrative Security is enabled, type the WebSphere user ID and password in the boxes provided and click **Log In**.
- 6 In the navigation tree of the WebSphere Administrative Console, click **Servers > Server Types > WebSphere application servers**.
- 7 Select the listed server and click **Start**. Notice that the server status changes as the server is started.

6.1.6 Directory permissions

The LiveCycle application will extract files to the `[appserver root]/installedApps` directory. 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.

***Note:** It is recommended that you modify the location of the extracted files to `[appserver root]/profiles/<profile_name>/installedApps`.*

6.1.6.1 Modify the location for the extracted files

- 1 Log in to the WebSphere Administrative Console.
- 2 Click **Servers > Server Types > WebSphere Application servers** and click your server name, such as **server1**.
- 3 Under Server Infrastructure, click **Java and Process Management > Process Definition**.
- 4 Under Additional Properties, click **Java Virtual Machine** and then click **Custom Properties**.
- 5 Click **New** and create a custom property named `adobeidp.RootDirectory`.
- 6 Set the value of `adobeidp.RootDirectory` to the path where Adobe native files should be extracted, such as `[appserver root]/profiles/<profile_name>/installedApps`.
- 7 Click **OK** or **Apply**.
- 8 In the Messages box, click **Save directly to master configuration**, and then restart the application server.

6.2 Preparing WebSphere Application Server

This section describes how to prepare and configure an application server instance for your LiveCycle deployment.

6.2.1 Increase the SOAP request time out

- 1 Go to the `[appserver root]` directory and search for all files named `soap.client.props`. Multiple files may have this name. For example, on an AIX, Linux, or Solaris server, the following files exist:
 - `[appserver root]/profileTemplates/default/documents/properties/soap.client.props`
 - `[appserver root]/profileTemplates/management/documents/properties/soap.client.props`
 - `[appserver root]/profiles/<profile_name>/properties/soap.client.props`
 - `[appserver root]/profiles/<profile_name>/temp/soap.client.props`
- 2 Open each `soap.client.props` file in a text editor, find the `com.ibm.SOAP.requestTimeout` property, and change the value from 180 to 1800.
- 3 Save and close each `soap.client.props` file.
- 4 In the navigation tree of the WebSphere Administrative Console, click **Servers > Server Types > WebSphere Application servers** and, in the right pane, click the server name.
- 5 Under Server Infrastructure, click **Administration > Administration Services**.
- 6 Under Additional Properties, click **JMX Connectors**, and then click **SOAPConnector**.
- 7 On the next screen, click **Custom properties**, and then click **requestTimeout**.
- 8 If necessary, change 600 to 1800 in the **Value** box on the next screen. Click **OK** or **Apply**.
- 9 In the **Messages** box, click **Save directly to master configuration**.

6.2.2 Increase the Deployer heap size

You must increase the heap size in the `ejbdeploy.bat/sh` script to avoid time-out errors.

6.2.2.1 AIX, Linux, or Solaris

- 1 Go to the `[appserver root]/deploytool/itp/` directory and open `ejbdeploy.sh` for editing.
- 2 **(Solaris only)** In the `SunOS` section, find the `EJBDEPLOY_JVM_OPTIONS` attribute and change the value of the `-XX:PermSize` option to 256m, and ensure that the value of the `-Xverify` option is `none`.
- 3 Change the heap size in the `$JAVA_CMD\` section to the following value:
`-Xms256m -Xmx512m`
- 4 Save and close the file.

6.2.2.2 Windows

- 1 Go to `[appserver root]\deploytool\itp` and open the `ejbdeploy.bat` file in a text editor.
 - 2 Find the line beginning with `%JAVA_HOME%` and then find the argument `-Xmx`.
 - 3 Change the argument to `-Xmx512M`.
- ❖ Save and close the file.

6.2.3 Increase MaxPermSize (WebSphere on Solaris)

- 1 Log in to the WebSphere Administrative Console.
- 2 In the navigation tree of the WebSphere Administrative Console, click **Servers > Server Types > WebSphere Application servers** and, in the right pane, click the server name.

- 3 Under Server Infrastructure, click **Java and Process Management > Process Definition**.
- 4 Under Additional Properties, click **Java Virtual Machine**.
- 5 In the **Generic JVM Arguments**, enter the MaxPermSize parameter as `-XX:MaxPermSize=512m`.
- 6 Click **OK** or **Apply**.
- 7 In the Messages box, click **Save directly to master configuration**, and then restart the application server.

6.2.4 Configure inbound and outbound communication

Note: Perform these steps only if you have Global Security enabled on your WebSphere application server.

- 1 In WebSphere administration console, navigate to **Security > Global Security**.
- 2 In the Authentication section, click **RMI/IIOP security**.
- 3 Click **CSIV2 inbound communication** and set **Transport** to **SSL-supported**. Click **OK**.
- 4 In the Messages box, click **Save directly to master configuration**.
- 5 Click **CSIV2 outbound communication** and set **Transport** to **SSL-supported**.
- 6 Click **OK**.
- 7 In the Messages box, click **Save directly to master configuration**. Click **OK**.

6.3 Configuring the LiveCycle database connectivity

Next, configure the database connectivity and install LiveCycle by following the instructions provided in [Installing and Deploying LiveCycle for WebSphere](#).

Chapter 7: Configuring a WebLogic Server

7.1 LiveCycle requirements for WebLogic

Two primary startup scenarios are available for WebLogic:

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

To use the automatic configuration features of Configuration Manager (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 Configuration Manager. However, you must manually enter the Configuration Manager changes into the StartManagedWebLogic scripts. For the appropriate steps, see *Installing and Deploying LiveCycle for WebLogic*.

7.2 Installing WebLogic Server

Obtain Oracle WebLogic Server for your operating system and install it according to the manufacturer's documentation (see http://download.oracle.com/docs/cd/E14571_01/wls.htm).

Note: You must install and run WebLogic Server by using a user account that has computer administrator privileges.

7.2.1 Configuring WebLogic 11g R1 on Solaris (64-bit Java)

Additional steps need to be performed while installing WebLogic to use 64-bit Java on Solaris.

- 1 Run the WebLogic installer using the `-D64` flag.
- 2 From a new command prompt, navigate to `[appserver_root]/server/bin` directory
- 3 Type `export JAVA_OPTIONS=-D64`.
- 4 Start Node Manager by using the script; type `startNodeManager.sh`.

Note: You must start the WebLogic Administration Server using 64-bit Java. Before starting the WebLogic Administration server using `startWeblogic.sh` script, ensure that `JAVA_OPTIONS=-D64` is set in the environment.

7.3 Installing JDK

7.3.1 Install JDK

Depending on your operating system, obtain and install the Oracle JRockit® or Oracle JAVA for your operating system. For information on the supported version, see [Supported Platform Combinations](#).

The `JAVA_HOME` and `PATH` environment variables must point to the JDK on the server where LiveCycle will be deployed.

7.3.2 Set the JAVA_HOME environment (Windows)

- 1 Select **Start > Control Panel > System**.
- 2 Click the **Advanced** tab and click **Environment Variables**.
- 3 In the System Variables area, click **New**.
- 4 Enter `JAVA_HOME` as the variable name and the directory where you installed the Java SDK. This directory is where WebLogic installed the Java SDK that contains the `/bin` subdirectory. For example, type this text:

```
C:\Program Files\Java\jrockit-jdk1.6.0_24-R28.1.3-4.0.1
```

Note: If you are using PDF Generator to convert OpenOffice.org files, you must set the `JAVA_HOME_32` environment variable to point to the directory where the Sun JDK is installed. Download the Sun JDK from <http://www.java.sun.com>.

7.3.3 Set the PATH environment variable (Windows)

- 1 Select **Start > Control Panel > System**.
- 2 Click the **Advanced** tab and click **Environment Variables**.
- 3 In the System Variables area, select the `PATH` variable and then click **Edit**.
- 4 Append the following text to the beginning of the variable value:

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

Note: Verify that the last entry of the `PATH` environment variable has no trailing slash (\) otherwise the WebLogic server instance will fail on start-up. Remove the slash if it exists and save your modification.

7.3.4 Set the JAVA_HOME environment (Linux and Solaris)

Set the `JAVA_HOME` variable for Bourne and Bash shells as shown in this example:

Linux:

```
JAVA_HOME=/opt/jrockit 1.6.0.28
export JAVA_HOME
```

Solaris:

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

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

Note: You must set the `JAVA_HOME_32` environment variable if you are using PDF Generator to convert Open Office files on a WebLogic Server 11g environment.

7.3.5 Set the PATH environment variable (Linux and Solaris)

Set the `PATH` variable for Bourne and Bash shells as shown in this example:

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

7.3.6 Verify JAVA_HOME environment variable

(Optional) Open a command prompt and run the following command:

```
java -version
```

The command returns the Java version installed on your system.

7.4 Creating the WebLogic Server domain

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

Domain with Managed Servers (recommended): 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.

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 (not recommended).

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

7.4.1 Create a WebLogic 11g domain with Managed Server

- 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 and Solaris) `./config.sh`
- 2 On the Welcome 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 products** and click **Next**.
- 4 On the Create WebLogic Domain screen, either accept the default values or enter the domain name and location as required, and then click **Create**.
- 5 In the Configure Administrator Username and Password screen, type your WebLogic user name and password, confirm the password by retyping it, and then click **Next**. Record this user name and password because Configuration Manager will prompt you for them.
- 6 In the WebLogic Domain Startup Mode panel, select **Production Mode**.
Important: Using Development Mode is not recommended.
- 7 In the right pane, navigate to the location of the JDK according to your operating system and then click **Next**:
 - (Windows, Linux) Location of **JRockit SDK 1.6** or later. For example, `C:\Program Files\JRockit Real Time\jrrt-4.0.0-1.6.0`(Windows) or `/opt/JRockit Real Time/jrrt-4.0.0-1.6.0`(Linux)
 - (Solaris) Location of **Sun JDK 1.6** or later.
- 8 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 and Solaris) Click **Done**, and start the server by navigating from a command prompt to the `[appserverdomain]` and typing `./startWebLogic.sh`
- 9 (Windows, Linux, or Solaris) When prompted, enter the WebLogic user name and password that you entered in step 5.

Note: The Admin Server starts in the command prompt. Enter the user name and password in the command prompt.

7.5 WebLogic Managed Server

You must deploy LiveCycle 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: You must start the WebLogic Administration Server using 64-bit Java. Before starting the Weblogic Administration server using `startWeblogic.sh` script, ensure that `JAVA_OPTIONS=-D64` is set in the environment.

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

7.5.1 Start Node Manager

You must start the Node Manager before you continue the remaining sections in this chapter.

- From a new command prompt, navigate to the `[appserver root]\server\bin` directory and type the appropriate command:

(Windows) `startNodeManager.cmd`

(UNIX) `./startNodeManager.sh`

Note: On Windows, if you have installed Node Manager as a Windows service, you must start the Node Manager service. If not, use the command line option to start the Node Manager.

7.5.2 Create a new WebLogic Managed Server

- 1 If WebLogic Administration Server is not already running, from a command prompt, navigate to the `[appserverdomain]` directory and type the appropriate command:
 - (Windows) `startWebLogic.cmd`
 - (Linux and Solaris) `./startWebLogic.sh`
- 2 To access the WebLogic Server 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, click **Environment >Servers** and, in the right pane, click **New**.
- 6 In the Create New Server page, in the **Server Name** box, type a name (such as `server1`) for your Managed Server.
- 7 In the **Server Listen Address** box, type the computer name or the IP address.

Note: Make note of this value as you must use the same one when you configure the application server and initialize the database when running Configuration Manager, which defaults to `localhost`.

- 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 Under Domain Structure, click **Environment > Machines** and, in the right pane, click **New**.
- 12 In the Create New Machine page, type a name for the machine in the **Name** box, select the operating system you are using from the **Machine OS** list, and then click **OK**.
- 13 Click **Environment > Servers** and click the name of the Managed Server you created.
- 14 In the **Machine** list, select the machine you just created.
- 15 Verify that the listen port is the same as the port number you entered in step 8.
- 16 Click **Save**.

7.5.3 Memory settings (Solaris only)

On Solaris platform, Configuration Manager does not set the Maximum Heap Size settings for WebLogic, therefore you must configure LiveCycle manually using the WebLogic Server Administration Console.

- 1 On the Configuration tab, click the **Server Start** tab.
- 2 In the **Arguments** box, type the appropriate text to set the memory size for the Managed Server:
 - `-XX:MaxPermSize=512m`
- 3 Click **Save** and then click **Activate Changes**.

7.5.4 Complete the memory settings for the Managed Server on WebLogic 11g

- 1 Move to the following directory: `[WL_HOME]\user_projects\domains\[appserverdomain]\bin`.
- 2 Edit the following file in a text editor:
 - (Windows) `setDomainEnv.cmd`
 - (Linux and Solaris) `setDomainEnv.sh`
- 3 Search for the line `WLS_MEM_ARGS_64BIT=-Xms256m -Xmx512m` and change it to `WLS_MEM_ARGS_64BIT=-Xms256m -Xmx1024m`.
- 4 Search for the line `-XX:MaxPermSize=256m` and change it to `XX:MaxPermSize=512m`.
***Note:** You need to set the `MaxPermSize` parameter only for 64-bit JVM for Solaris.*
- 5 Save the changes and close the file.

7.6 Stopping and restarting WebLogic

After you make all your configuration changes, restart WebLogic for the changes to take effect. The WebLogic Managed Server, Node Manager and the WebLogic Administration Server also need to be restarted.

You must start the WebLogic Administration Server using 64-bit Java. Before starting the Weblogic Administration server using `startWeblogic.sh` script, ensure that `JAVA_OPTIONS=-D64` is set in the environment.

7.6.1 Stop WebLogic Managed Server

- 1 In the WebLogic Server Administration Console, under Domain Structure, click the domain name.
- 2 Click the **Control** tab and select the check box beside the server you want to stop.

- 3 Click **Shutdown** and select one of these 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 that is currently in progress.

Force Shutdown Now: Initiates a forced shutdown, which causes 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 shut down by viewing the table at the bottom of the Control tab. The table displays a list of all the servers and indicates their current state.

7.6.2 Stop WebLogic Administration Server

- 1 From a command prompt, navigate to *[appserverdomain]\bin*.
- 2 Type the following command:
 - (Windows) `stopWebLogic.cmd`
 - (Linux, UNIX) `./stopWebLogic.sh`
- 3 Enter the WebLogic user name and password (if you enabled security when installing WebLogic).

7.6.3 Start WebLogic Administration Server

- 1 From a command prompt, navigate to *[appserverdomain]*.
- 2 Type the following command:
 - (Windows) `startWebLogic.cmd`
 - (Linux, UNIX) `./startWebLogic.sh`
- 3 Enter the WebLogic user name and password (if you enabled security when installing WebLogic).

7.6.4 Start WebLogic Node Manager

- From a command prompt, navigate to the *[appserver root]/server/bin* directory and type the appropriate command:
(Windows) `startNodeManager.cmd`
(Linux, UNIX) `./startNodeManager.sh`

7.6.5 Start WebLogic Managed Server

- 1 When the WebLogic Administration Server has started, log in to the WebLogic Server Administration Console.
- 2 Under Domain Structure, click **Environment** > **Servers** and, in the right pane, click the managed server.
- 3 On the next screen, click the **Control** tab and select the check box beside the managed server you want to start.
- 4 Click **Start** and then click **Yes**.

7.7 Creating JMX policies for database initialization

You must create JMX policies to ensure that the database for core LiveCycle components initializes correctly.

Complete the following procedures.

7.7.1 Delegating MBean authorization to the realm

Before creating JMX policies, ensure that the security realm is set up to control access to MBeans. For more information, refer to the WebLogic Administration Console documentation.

- 1 In the WebLogic Administration Console, click **Domain Structure > Security Realms**.
- 2 Click **myrealm** from the Realms list on the **Summary of Security Realms** page.
- 3 On the Configuration > General page, ensure that **Use Authorization Providers to Protect JMX Access** is selected. If this option is not selected, perform the following steps:
 - Click **Lock & Edit** in the Change Center.
 - Select **Use Authorization Providers to Protect JMX Access**.
 - Click **Save**.
 - In the Change Center, click **Activate Changes**.
 - Restart the admin server and the managed server.

7.7.2 Create JMX policies

- 1 In the WebLogic Administration Console, click **Domain Structure > Security Realms**.
- 2 On the **Summary of Security Realms** page, click the name of the realm for which you want to modify JMX policies.
- 3 On the Settings page, click the **Roles and Policies** tab and then click the **Realm Policies** sub tab.
- 4 In the Name column of the **Policies** table, click **JMX Policy Editor**.
- 5 On the JMX Policy Editor page, ensure that the **GLOBAL SCOPE** option is selected. Click **Next**.
- 6 Ensure that the **ALL MBEANS TYPES** option is selected on the next page. Click **Next**.
- 7 Select the **Attributes: Permission to Write** option and click **Create Policy**.
- 8 On the Edit JMX Policies page, click **Add Conditions**.
- 9 Select **Role** from the **Predicate List** drop-down menu and click **Next**.
- 10 In the **Role Argument Name** box, enter **Anonymous** and click **Add**.

Note: The Anonymous role is a default WebLogic role for all runtime process users (for example, users required for bootstrapping an application).
- 11 Click **Finish**.
- 12 On the Edit JMX Policies page, click **Save**.
- 13 Repeat steps 1 to 6.
- 14 On the **JMX Policy Editor - Attributes and Operations** page, select the **Unregister instances of this MBean using MBean server** option and click **Create Policy**.

15 Repeat steps 8 to 12.

7.8 Configuring the LiveCycle database connectivity

Next, configure the database connectivity and install LiveCycle by following the instructions provided in [Installing and Deploying LiveCycle for WebLogic](#).

Chapter 8: Preconfigured Environment Checklist

Before you advance to the Application Server Installation guides, ensure you have recorded the following information on your system configuration:

Server setup

Preconfiguration	Record data here	Required?
System Requirements met	<enter text here>	Yes
Temp directory		
JDK Install location	<enter text here>	Yes
Global Storage Directory created	<enter text here>	Only if you do not want to use the default locations.

Application Server Configuration

Preconfiguration	Record data here	Required?
Application Server hostname or IP Address	<enter text here>	Yes
Application Server credentials	<enter text here>	Yes
Application Server port number	<enter text here>	Yes
Preconfiguration completed as described in “Configuring a JBoss Application Server” on page 37 or “Configuring a WebLogic Server” on page 53 or “Configuring a WebSphere Application Server” on page 48?	<enter text here>	Yes

Database Configuration

Preconfiguration	Record data here	Required?
Database hostname or IP Address	<enter text here>	Yes
Database account credentials	<enter text here>	Yes
Database port number	<enter text here>	Yes
Preconfiguration completed as described in “Creating the LiveCycle Database” on page 26?	<enter text here>	Yes

LDAP Server

Preconfiguration	Record data here	Required?
LDAP Server hostname or IP Address	<enter text here>	This is only required if you are using LDAP to authenticate users.
LDAP account credentials	<enter text here>	This is only required if you are using LDAP to authenticate users.

PDF Generator configuration (Windows)

Preconfiguration	Record data here	Required?
Install Microsoft Office	<enter text here>	Yes. Note that you must use a user account with administrator privileges to install Acrobat and Microsoft Office.
Install Acrobat	<enter text here>	Yes. Note that you must use a user account with administrator privileges to install Acrobat and Microsoft Office.
Other native application software installed	<enter text here>	Yes.
Configure the Windows environment variables	<enter text here>	Yes.
32-bit JDK installed		

Chapter 9: Appendix - Manually Configuring JBoss

This appendix describes the configuration that is required for JBoss 5.1 EAP that you can download from Red Hat. This option should be considered for advanced installations only. Advanced knowledge of JBoss is typically required.

LiveCycle ES3 runs on JBoss Windows Server 2008 R1/R2 (Enterprise or Standard Edition), Red Hat Linux ES/AS 5.5, SUSE Linux ES 11 platforms, and Solaris 10.

***Note:** If you plan to use the LiveCycle ES3 turnkey installation for JBoss, you do not need to review this document. All the instructions that are required for preparing your environment for a turnkey installation are available in [Installing and Deploying LiveCycle Using JBoss Turnkey](#).*

9.1 Installing the JDK for JBoss

You must download and install Oracle JDK 6.0 update 26 or later updates to 6.0 versions from <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Create or set the `JAVA_HOME` environment variable to point to the location where the JDK is installed.

9.1.1 Set the `JAVA_HOME` environment variable (Windows)

- 1 Select **Start > Control Panel > System**.
- 2 Click the **Advanced** tab.
- 3 Click **Environment Variables** and, under System Variables, click **New**.
- 4 In the **New System Variable** box, type `JAVA_HOME` as the variable name and enter the directory where you installed the JDK. This directory is the directory that contains the `/bin` subdirectory. For example, type the following path:

```
C:\Program Files\Java\jdk1.6.0_26
```

9.1.2 Set the `PATH` environment variable (Windows)

- 1 Select **Start > Control Panel > System**.
- 2 Click the **Advanced** tab and click **Environment Variables**.
- 3 In the System Variables area, select the `PATH` variable and then click **Edit**.
- 4 Append the following text to the beginning of the variable value:

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

9.1.3 Set the `JAVA_HOME` environment (Linux and Solaris)

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

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

9.1.4 Set the PATH environment variable (Linux and Solaris)

- Set the PATH variable for Bourne and Bash as shown in the following example:

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

9.1.5 Verify JAVA_HOME environment variable setting (Windows, Linux, or Solaris)

(Optional) Open a command prompt and run the following command:

```
java -version
```

The command returns the Java version installed on your system.

9.2 Manually installing JBoss

You can download and install JBoss 5.1 EAP from <http://www.jboss.org/jbossas/downloads/>.

9.3 Starting and stopping JBoss

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

All JBoss start configurations are located in the *[appserver root]/server* directory. For JBoss obtained directly from Red Hat, either use one of the JBoss provided start configurations (all, default or minimal) or a custom configuration.

9.3.1 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 [profile_name] -b [server_IP_Address]`
- (Linux and Solaris) `./run.sh -c [profile_name] -b [server_IP_Address]`

where *[profile_name]* is the configuration required for your database and *[server_IP_Address]* is the IP address of the server.

9.3.2 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 [server_IP_Address]:<jndi -port>`
- (Linux and Solaris) `./shutdown.sh -s [server_IP_Address]:<jndi -port>`

9.4 Modifying the JBoss configuration

The JBoss Application Server is configured using various XML configuration files. JBoss must be shut down before editing any of these configuration files. If JBoss is running and these files are changed, JBoss will probably crash. JBoss also has a few configuration files that are formatted as .property files. You must ensure that the .property files are saved as UNIX text files on Linux or Solaris if you edit these files on Windows environments at any time.

For single-server installations, you may use jboss profile located at `[appserver root]\server\standard\` as a template. For cluster installations, use jboss profile located at `[appserver root]\server\all\` as a template.

It is recommended that you make a copy of the profile (all or standard) and make changes to the copied profile.

9.4.1 Remove JMS and configuration files (optional)

Because LiveCycle does not use JMS configuration, you can delete the following files and directories that are part of JBoss. Ensure that you delete the entire directory and its contents listed below.

Note: Skip this step if you need to enable LiveCycle Foundation JMS services or deploy other services or applications that depend on the JMS service.

- `[appserver root]\server\<profile_name>\deploy\messaging`
- `[appserver root]\server\<profile_name>\deploy\jms-ra.rar`

9.4.2 Modify the JBoss configuration

Perform the following steps to modify the JBoss configuration to customize JBoss for LiveCycle.

- Update the `jacorb.properties` file
- Update the URI Encoding in the JBoss `server.xml` file
- Modify EAR file class-loading isolation
- Modify the `run.conf.bat` file (Windows)
- Modify `run.conf` (Linux and Solaris)
- Modify `log4j.xml`
- Modify the `jbosssts-properties.xml` file

9.4.2.1 Update the `jacorb.properties` file

- 1 Open the `[appserver root]/server/<profile_name>/conf/jacorb.properties` file in a text editor.
- 2 Locate the `jacorb.poa.thread_pool_max` setting and change the value to 16.
- 3 Save and close the file.

9.4.2.2 Update the URI Encoding in the JBoss `server.xml` file

- 1 Open the `[appserver root]/server/<profile_name>/deploy/jbossweb.sar/server.xml` file in a text editor.
- 2 Locate the following lines:

```
<Connector protocol="HTTP/1.1" port="8080" address="{jboss.bind.address}"
connectionTimeout="20000" redirectPort="8443" />
```
- 3 Append `URIEncoding="UTF-8"`:

```
<Connector protocol="HTTP/1.1" port="8080" address="{jboss.bind.address}"
connectionTimeout="20000" redirectPort="8443" URIEncoding="UTF-8" />
```

- 4 Save and close the file.

9.4.2.3 Modify EAR file class-loading isolation

- 1 Open the `[appserver root]/server/<profile_name>/deployers/ejb-deployer-jboss-beans.xml` file in an editor.
- 2 Locate `<property name="CallByValue">` and change the value to `true`.
- 3 Save and close the file.
- 4 Open the `[appserver root]/server/<profile_name>/deployers/ear-deployer-jboss-beans.xml` file in an editor.
- 5 Locate `<property name="isolated">` and change the value to `true`.
- 6 Locate `<property name="CallByValue">false</property>` and change the value to `true`.
- 7 Save and close the file.
- 8 Open the `[appserver root]/server/<profile_name>/conf/jboss-service.xml` file in an editor.
- 9 Locate `<attribute name="CallByValue">false</attribute>` and change the value to `true`.
- 10 Save and close the file.

9.4.2.4 Modify the run.conf.bat file (Windows only)

- 1 Open the `[appserver root]/bin/run.conf.bat` file in an editor.
- 2 Add the following lines

Memory argument for 32-bit JVM:

```
set "JAVA_HEAP_ARGS=-Xms1024m -Xmx1024m -XX:PermSize=128m -XX:MaxPermSize=192m"
```

Memory argument f for 64-bit JVM:

```
set "JAVA_HEAP_ARGS=-Xms1024m -Xmx2048m -XX:PermSize=256m -XX:MaxPermSize=512m -
XX:+UseCompressedOops"
```

Other argument f for 32-bit and 64-bit JVM:

```
set "JAVA_OPTS=%JAVA_OPTS% -Dadobeidp.serverName=server1 -Dfile.encoding=utf8 -
Djava.net.preferIPv4Stack=true"
set "JAVA_OPTS=%JAVA_OPTS% -DentityExpansionLimit=10000"
set "JAVA_OPTS=%JAVA_OPTS% -XX:+HeapDumpOnOutOfMemoryError"
```

- 3 (Optional) Modify JBoss Application Server to run in IPv6 mode as follows:

- Locate and modify `-Djava.net.preferIPv4Stack=false`
- Insert the string `-Djava.net.preferIPv6Stack=true`

Note: If the application server log contains the following error on startup, remove the value for the IPv6 stack and set the IPV4 value back to `true`:

```
"13:37:44,488 WARN [HANamingService] Failed to start AutomaticDiscovery java.net.SocketException: bad
argument for IP_MULTICAST_IF: address not bound to any interface at
java.net.PlainDatagramSocketImpl.socketSetOption(Native Method)at
java.net.PlainDatagramSocketImpl.setOption(PlainDatagramSocketImpl.java:260)"
```

Save and close the file.

9.4.2.5 Modify the run.conf file (JBoss with Solaris 10, Red Hat 5.5, 64-bit only)

Solaris JDKs from Sun require an additional argument to use 64-bit features. Without this configuration change, the Sun JDK defaults to 32-bit support only.

***Note:** If you're running JBoss as a non-root user, use `-Djava.io.tmpdir="location"` to set the location of the temporary directory to a directory to which you have access.*

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

2 Locate the section starting with:

```
if [ "x$JAVA_OPTS" = "x" ]; then
```

Modify the section to look like:

```
if [ "x$JAVA_OPTS" = "x" ]; then
    #JAVA_OPTS="-Xms1303m -Xmx1303m -XX:MaxPermSize=256m -Dorg.jboss.resolver.warning=true -
    Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000 -
    Dsun.lang.ClassLoader.allowArraySyntax=true"
    JAVA_OPTS="$JAVA_OPTS -Xms1024m -Xmx2048m -XX:MaxPermSize=512m -
    Dorg.jboss.resolver.warning=true -Dsun.rmi.dgc.client.gcInterval=3600000 -
    Dsun.rmi.dgc.server.gcInterval=3600000 -Dsun.lang.ClassLoader.allowArraySyntax=true"
    JAVA_OPTS="$JAVA_OPTS -Dadobeidp.serverName=server1 -Dfile.encoding=utf8 -
    Djava.net.preferIPv4Stack=true"
    JAVA_OPTS="$JAVA_OPTS -DentityExpansionLimit=10000"
    JAVA_OPTS="$JAVA_OPTS -XX:+UseCompressedOops -XX:+HeapDumpOnOutOfMemoryError"
```

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

3 (optional) Modify JBoss Application Server to run in IPv6 mode as follows:

- Locate and modify `-Djava.net.preferIPv4Stack=false`
- Add `-Djava.net.preferIPv6Stack=true`

4 Save and close the file.

9.4.2.6 Modify the log4j.xml file

1 Open the `[appserver root]/server/<profile_name>/conf/jboss-log4j.xml` file in an editor.

2 Locate the following text in the FILE appender section and add the line/change the value that appears in bold:

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

3 Locate the Limit categories section and add the following:

```
<category name="jacorb.config">
  <priority value="ERROR"/>
</category>
<category name="com.adobe">
  <priority value="INFO"/>
</category>
<category name="org.springframework">
  <priority value="WARN"/>
</category>
<!-- Limit the verbose MC4J EMS (lib used by admin-console) categories -->
<category name="org.mc4j.ems">
  <priority value="WARN"/>
</category>
```

- 4 Save and close the file.

9.4.2.7 Modify the jbossts-properties.xml file:

- 1 Ensure that transaction management works as expected by locating the jbossts-properties.xml file in the [appserver root]/server/<profile_name>/conf directory and opening the file in an editor.
- 2 If you are not using messaging, locate and comment out following lines

```
<property
name="com.arjuna.ats.jta.recovery.XAResourceRecovery.JBMESSAGING1" value="org.jboss.jms.server.recovery.MessagingXAResourceRecovery;java:/DefaultJMSProvider"/>
```

- 3 Save and close the file.

Note: For more information, see the JBoss article 11443 at <http://www.jboss.org/community/docs/DOC-11443>.

9.5 Copying jar files

Copy all the JAR files, except the JDBC JAR file from [DVD root]/third_party/jboss.zip/server/lc_<db-name>/lib to the [appserver root]/server/<profile_name>/lib directory of your downloaded JBoss.

9.6 LiveCycle database connectivity for manually installed JBoss

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

- Configure the LiveCycle 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 Configuration Manager and the application server to connect to the LiveCycle database. 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 an MySQL, Oracle, or SQL Server data source.

Note: Before proceeding with the following tasks, ensure that JBoss is not running.

9.6.1 Configuring MySQL for manually installed JBoss

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

- Obtain and copy the MySQL JDBC driver to the instance of JBoss where you will deploy LiveCycle.
- Create a data source file and deploy it to the instance of JBoss where you will deploy LiveCycle.
- Encrypt the password in the data source files (`adobe-ds.xml` and `mysql-ds.xml`) and the `login-config.xml` file using one of the methods described at <http://community.jboss.org/wiki/EncryptingDataSourcePasswords>. You can also use the instructions available on http://blogs.adobe.com/livecycle/2009/10/livecycle_-_encrypting_clearite.html.

9.6.1.1 Configuring the MySQL data source

Before you configure the MySQL data source, you must have the database created on MySQL. (See Creating a MySQL database.)

9.6.1.2 Install the MySQL database driver:

- Copy the `mysql-connector-java-5.1.6-bin.jar` driver file from the `[DVD_root]/third_party/db/mysql` directory on the installation DVD to the `[appserver root]/server/<profile_name>/lib` directory.

9.6.1.3 Edit `adobe-ds.xml` file

- 1 Copy the `adobe-ds.xml` file from the `[DVD_root]/third_party/additional/datasources/lc_mysql/deploy` directory on the installation DVD to the `[appserver root]/server/<profile_name>/deploy` directory.

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

```
<connection-url>jdbc:mysql://localhost:3306/adobe</connection-url>
<driver-class>com.mysql.jdbc.Driver</driver-class>
<user-name>adobe</user-name>
<password>adobe</password>
```

- 3 Replace the following values 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 data. You will need to update the default value, `adobe`, with your database name.

- 4 In the lines that follow the `<connection-url>` settings, locate the `user-name` and `password` settings and replace the default values with the user name and password that the application server uses to access your database.

- 5 Modify the minimum and maximum values for the data source connections:

- IDP_DS:

```
<min-pool-size>1</min-pool-size>
<max-pool-size>30</max-pool-size>
```

- EDC_DS:

```
<min-pool-size>1</min-pool-size>
<max-pool-size>20</max-pool-size>
```

Note: If your LiveCycle Server handles heavy load, increase the number of maximum JDBC connections to ensure that all jobs are processed. In such cases, increase `<max-pool-size>` to 50 or more for both IDP_DS and EDC_DS.

- 6 Save and close the file.

9.6.1.4 Set MySQL as the data source

- 1 Navigate to the `[appserver root]/server/<profile_name>/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/<profile_name>/deploy` directory.
- 3 Open the `[appserver root]/server/<profile_name>/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://mysql-hostname:3306/jbossdb</connection-url>
<driver-class>com.mysql.jdbc.Driver</driver-class>
<user-name>x</user-name>
<password>y</password>
```

- 4 Replace the bold values with values that are specific to your database:
 - **MySQLDS**: Change to `DefaultDS`.
 - **mysql-hostname**, **3306**, **jbossdb**, **x**, and **y**: The database values that the application server uses to access the database.
- 5 Add the following line to the `<local-tx-datasource>` section, if it does not already exist.

```
<transaction-isolation>TRANSACTION_READ_COMMITTED</transaction-isolation>
```
- 6 Save and close the file.

9.6.1.5 Edit the login-config.xml file

- 1 Open the `[appserver root]/server/<profile_name>/conf/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">adobe</module-option>
      <module-option name="userName">adobe</module-option>
      <module-option name="password">adobe</module-option>
      <module-option
        name="managedConnectionFactoryName">jboss.jca:service=LocalTxCM,
        name=DefaultDS </module-option>
      </login-module>
    </authentication>
  </application-policy>
```

- 2 Replace the bold values with values that are specific to your database.
- 3 Save and close the file.
- 4 Start JBoss.

9.6.2 Configuring Oracle for manually installed JBoss

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

- Obtain and copy the Oracle JDBC driver to the instance of JBoss where you will deploy LiveCycle.
- Create a data source file and deploy it to the instance of JBoss where you will deploy LiveCycle.
- Encrypt the password in the data source files (`adobe-ds.xml` and `oracle-ds.xml`) and the `login-config.xml` file using one of the methods described at <http://community.jboss.org/wiki/EncryptingDataSourcePasswords>. You can also use the instructions available on http://blogs.adobe.com/livecycle/2009/10/livecycle_-_encrypting_clearite.html.

9.6.2.1 Install the Oracle 11g database driver

Copy the `ojdbc6.jar` for JDK 1.6 driver file from the `[LiveCycle root]/lib/db/oracle` directory to the `[appserver root]/server/<profile_name>/lib` directory. You can also download the Oracle 11g driver from the JDBC Driver Downloads site, see [Supported Platform Combinations](#) for supported versions Oracle 11g driver.

9.6.2.2 Edit `adobe-ds.xml` file

- 1 Copy the `adobe-ds.xml` file from the `[DVD_root]/third_party/additional/datasources/lc_oracle/deploy` directory to the `[appserver root]/server/<profile_name>/deploy` directory.

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

```
<connection-url>jdbc:oracle:thin:@localhost:1521:adobe</connection-url>
<driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
<user-name>adobe</user-name>
<password>adobe</password>
```

- 3 Replace the following values 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:** Change the default value, `adobe`, with your database SID.

- 4 In the lines that follow the `<connection-url>` settings, locate the `user-name` and `password` settings and replace the default values with the user name and password that the application server uses to access your database.

- 5 (Only for Oracle RAC) Replace the connection URL mentioned in step 2 with the following connection URL:

```
jdbc:oracle:thin:@(DESCRIPTION=(ENABLE=broken) (ADDRESS_LIST=(ADDRESS= (PROTOCOL=TCP)
(HOST=yourhost1) (PORT=1521)) (ADDRESS= (PROTOCOL=TCP) (HOST=yourhost2) (PORT=1521))
(LOAD_BALANCE=on) (FAILOVER=on)) (CONNECT_DATA=(SERVER=dedicated)
(SERVICE_NAME=service.yourcompany.com) (FAILOVER_MODE=(TYPE=session) (METHOD=basic)
(RETRIES=10) (DELAY=3))))
```

Note: Ensure that this entry appears as a single line in the `adobe-ds.xml` file.

- 6 (Only for Oracle RAC) Replace the following text from the connection URL in step 5 with values that are specific to your database:

- **yourhost1:** The name, IP address, or fully-qualified domain name of the first node in the cluster that hosts the database.
- **yourhost2:** The name, IP address, or fully-qualified domain name of the second node in the cluster that hosts the database.

*Note: The cluster hosting the database could have n nodes. **yourhost1** and **yourhost2** are examples in the case of a two-node cluster.*

- **service.yourcompany.com:** The service name for the Oracle RAC database.

7 Modify the minimum and maximum values for the data source connections:

- IDP_DS:

```
<min-pool-size>1</min-pool-size>
<max-pool-size>30</max-pool-size>
```

- EDC_DS:

```
<min-pool-size>1</min-pool-size>
<max-pool-size>20</max-pool-size>
```

Note: If your LiveCycle Server handles heavy load, increase the number of maximum JDBC connections to ensure that all jobs are processed. In such cases, increase `<max-pool-size>` to 50 or more for both IDP_DS and EDC_DS.

8 Save and close the file.

9.6.2.3 Set Oracle as the data source

If you are running LiveCycle with a Oracle database, you must set Oracle to be the default data source for JBoss. This procedure assumes that the Oracle JDBC driver is installed in the `[appserver root]/server/<profile_name>/lib` directory.

- 1 Navigate to the `[appserver root]/server/<profile_name>/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/<profile_name>/deploy` directory.
- 3 Open the `[appserver root]/server/<profile_name>/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</connection-url>
<driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
<user-name>x</user-name>
<password>y</password>
```

- 4 Replace the bold values 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.

- 5 In the lines that follow the `<connection-url>` settings, locate the `user-name` and `password` settings and replace the default values with the user name and password that the application server uses to access your database.

- 6 (Only for Oracle RAC) Replace the connection settings mentioned in step 3 with the following connection URL:

```
jdbc:oracle:thin:@(DESCRIPTION=(ENABLE=broken) (ADDRESS_LIST=(ADDRESS= (PROTOCOL=TCP)
(HOST=yourhost1) (PORT=1521)) (ADDRESS=(PROTOCOL=TCP) (HOST=yourhost2) (PORT=1521))
(LOAD_BALANCE=on) (FAILOVER=on)) (CONNECT_DATA=(SERVER=dedicated)
(SERVICE_NAME=service.yourcompany.com) (FAILOVER_MODE=(TYPE=session) (METHOD=basic)
(RETRIES=10) (DELAY=3))))
```

Note: Ensure that this entry appears as a single line in the `oracle-ds.xml` file.

7 (Only for Oracle RAC) Replace the following text from the connection URL in step 6 with values that are specific to your database:

- **yourhost1:** The name, IP address, or fully-qualified domain name of the first node in the cluster that hosts the database.
- **yourhost2:** The name, IP address, or fully-qualified domain name of the second node in the cluster that hosts the database.

*Note: The cluster hosting the database could have n nodes. **yourhost1** and **yourhost2** are examples in the case of a two-node cluster.*

- **service.yourcompany.com:** The service name for the Oracle RAC database.

8 Save and close the file.

9.6.2.4 Edit the login-config.xml file

1 Open the `[appserver root]/server/<profile_name>/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">adobe</module-option>
      <module-option name = "userName">adobe</module-option>
      <module-option name = "password">adobe</module-option>
      <module-option name = "managedConnectionFactoryName">
        jboss.jca:service=LocalTxCM,name=DefaultDS</module-option>
    </login-module>
  </authentication>
</application-policy>
```

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

3 Save and close the file.

4 Start JBoss.

9.6.3 Configuring SQL Server for manually installed JBoss

To enable JBoss to connect to the SQL Server database that stores LiveCycle 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.
- Create a SQL Server data source file and deploy it to the instance of JBoss where you will deploy LiveCycle, such as `[appserver_root]/server/<profile_name>/deploy`.
- Encrypt the password in the data source files (`adobe-ds.xml` and `mssql-ds.xml`) and the `login-config.xml` file using one of the methods described at <http://community.jboss.org/wiki/EncryptingDataSourcePasswords>. You can also use the instructions available on http://blogs.adobe.com/livecycle/2009/10/livecycle_-_encrypting_clearte.html.

9.6.3.1 Configuring the SQL Server database connectivity

Before you configure the SQL Server data source, you must have the LiveCycle database created on SQL Server. (See Creating a SQL Server database.)

9.6.3.2 Install the SQL database driver for JBoss

- 1 Obtain the SQL Server JDBC 3.0 database driver from the Microsoft website.

Note: Use SQL Server JDBC Driver 3.0 for both Microsoft SQL Server 2005 SP2 and Microsoft SQL Server 2008.

- 2 (Windows) Download the *.exe file and run it, 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 sqjdbc.jar file from the `[SQL_root]/sqjdbc_3.0/enu` directory to the `[appserver root]/server/<profile_name>/lib` directory.
- 5 Delete the mysql-connector-java-3.1.12-bin.jar file located in the `[appserver root]/server/<profile_name>/lib` directory.

9.6.3.3 Edit adobe-ds.xml file

- 1 Copy the adobe-ds.xml file from the `[DVD_root]/third_party/additional/datasources/lc_sqlserver/deploy` directory to the `[appserver root]/server/<profile_name>/deploy` directory.
- 2 Open the adobe-ds.xml file in a text editor and modify the `<local-tx-datasource>` element with your SQL Server connection settings:

```
<connection-url>jdbc:sqlserver://localhost:1433;DatabaseName=adobe</connection-url>
<driver-class>com.microsoft.sqlserver.jdbc.SQLServerDriver</driver-class>
<user-name>adobe</user-name>
<password>adobe</password>
```

- 3 Replace the following values 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.
 - `adobe`: The name of the database that stores the LiveCycle data. You will need to update the default value, `adobe`, with your database name.
 - 4 Change the `<driver-class>` element as follows:
- ```
<driver-class>com.microsoft.sqlserver.jdbc.SQLServerDriver</driver-class>
```
- 5 In the lines that follow the `<driver-class>` settings, locate the `user-name` and `password` settings and replace the default values with the user name and password that the application server uses to access your database. Modify the minimum and maximum values for the data source connections:

- IDP\_DS:

```
<min-pool-size>1</min-pool-size>
<max-pool-size>30</max-pool-size>
```
- EDC\_DS:

```
<min-pool-size>1</min-pool-size>
<max-pool-size>20</max-pool-size>
```

**Note:** If your LiveCycle Server handles heavy load, increase the number of maximum JDBC connections to ensure that all jobs are processed. In such cases, increase `<max-pool-size>` to 50 or more for both IDP\_DS and EDC\_DS.

- 6 Save and close the file.

#### 9.6.3.4 Edit the mssql-ds.xml file

- 1 Navigate to the `[appserver root]/server/<profile_name>/deploy` directory and delete the `hsqldb-ds.xml` file.
- 2 Navigate to the `[appserver root]/docs/examples/jca` directory and copy the `mssql-ds.xml` file to the `[appserver root]/server/<profile_name>/deploy` directory.
- 3 Open the `mssql-ds.xml` file in a text editor and change the `<local-tx-datasource>` element with your SQL Server connection settings (not necessarily on consecutive lines):

```
<jndi-name>MSSQLDS</jndi-name>
<datasource-mapping>MS_SQLSERVER2000</datasource-mapping>
<connection-url>jdbc:sqlserver://localhost:1433; DatabaseName=MyDatabase</connection-url>
<driver-class>com.microsoft.sqlserver.jdbc.SQLServerDriver </driver-class>
<user-name>x</user-name>
<password>y</password>
```

- The `<jndi-name>` element to `DefaultDS`
- The `<datasource-mapping>` element to `MS_SQLSERVER2000`
- The Database name `MyDatabase` to your database name
- The `<user-name>`**x**element to your user name
- The `<password>`**y** element to your password

***Note:** The `<datasource-mapping>` element should point to `MS_SQLSERVER2000`, even if you are using MS SQL Server 2005.*

- 4 Save and close the file.

#### 9.6.3.5 Edit the login-config.xml file

- 1 Open the `[appserver root]/server/<profile_name>/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">adobe</module-option>
 <module-option name = "userName">adobe</module-option>
 <module-option name = "password">adobe</module-option>
 <module-option name =
 "managedConnectionFactoryName">jboss.jca:service=LocalTxCM,name= DefaultDS </module-
option>
 </login-module>
 </authentication>
</application-policy>
```

- 2 Replace the bold values with values that are specific to your database.
- 3 Save and close the file.
- 4 Start JBoss.

### 9.6.3.6 Configure Integrated Security on Windows

- 1 Modify the adobe-ds.xml and mssql-ds.xml files, located in *[appserver root]\server\<profile\_name>\deploy*, to add `integratedSecurity=true` to the connection URL, as shown in this example:

```
<connection-url>jdbc:sqlserver://<serverhost>:<port>;
databaseName=<dbname>;integratedSecurity=<true></connection-url>
```

- 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 within the Microsoft SQL JDBC 3.0 driver installation. The default location is *[SQL\_root]/sqljdbc\_3.0/enu/auth/x86* for 32-bit operating systems and *[SQL\_root]/sqljdbc\_3.0/enu/auth/x64* for 64-bit operating systems.
- 3 Open the properties for the JBoss for Adobe LiveCycle ES3 service or the JBoss service that you configured, and click the **Log On** tab.
- 4 Select **ThisAccount** and type the value of a valid user account. This change is not required if you are running JBoss from the command line.
- 5 Change SQL Server Security from Mixed mode to Windows Authentication only.

## 9.7 Next steps

Install LiveCycle by following the instructions provided in [Installing and Deploying LiveCycle server](#).