



Preparing to Install LiveCycle® ES2

December 07, 2011

Adobe® LiveCycle® ES2

Version 9

© 2011 Adobe Systems Incorporated and its licensors. All rights reserved.

Preparing to Install Adobe® LiveCycle® ES2 (Single Server)

December 07, 2011

This guide is licensed for use under the terms of the Creative Commons Attribution Non-Commercial 3.0 License. This License allows users to copy, distribute, and transmit the guide for noncommercial purposes only so long as (1) proper attribution to Adobe is given as the owner of the guide; and (2) any reuse or distribution of the guide contains a notice that use of the guide is governed by these terms. The best way to provide notice is to include the following link. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>.

Adobe, the Adobe logo, Adobe Reader, Acrobat, Flash, Flex, Flex Builder, FrameMaker, LiveCycle, PageMaker, Photoshop, and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. AIX and IBM are trademarks of International Business Machines Corporation in the United States, other countries, or both. Intel and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the U.S. and other countries. Oracle, Java, Solaris, and Sun are trademarks or registered trademarks of Oracle and/or its affiliates. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. Products bearing SPARC trademarks are based upon an architecture developed by Oracle Corporation. Red Hat is a trademark or registered trademark of Red Hat, Inc. in the United States and other countries. Linux is the registered trademark of Linus Torvalds in the U.S. and other countries. Microsoft, SharePoint, Windows, Windows Server, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Novell and SUSE are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. UNIX is a trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd. All other trademarks are the property of their respective owners.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA.

Contents

About This Document.....	6
Before you start.....	6
What's in this document?.....	6
Who should read this document?.....	6
1 System Requirements	10
LiveCycle ES2 patch updates	10
Third-party infrastructure support.....	10
Third-party patch support	10
Microsoft Windows operating systems.....	10
Linux operating systems	13
IBM AIX.....	14
Sun Solaris	15
System requirements.....	16
Minimum hardware requirements	16
Intel x86 compatibility	18
Recommended hardware requirements	18
Additional requirements for AIX, Linux, and Solaris.....	18
Additional hardware requirement for LiveCycle Content Services ES2	20
Recommended hardware requirements for client-side computers.....	20
Supported software	22
Installation user account for Windows.....	26
Configuration for 64-bit Windows installations.....	26
Minimum database user permissions.....	27
Web browser support.....	28
Additional requirements for PDF Generator ES2 and PDF Generator 3D ES2	30
Additional requirements for Connector for IBM Content Manager	36
Additional requirements for LiveCycle ES2 Connector for IBM FileNet	37
LDAP configuration	38
Global document storage directory	38
Planning and creating the global document storage directory	39
Location of the global document storage directory	39
Sizing factors for the global document storage directory	39
Securing the global document storage directory	40
Backing up the global document storage directory	40
LiveCycle Business Activity Monitoring ES2 requirements	40
Basic requirements.....	40
Creating the Business Activity Monitoring ES2 databases.....	41
JBoss requirements for BAM	42
WebLogic requirements for BAM.....	42
WebSphere requirements for BAM.....	43
Installation considerations.....	43
Disabling Virus scans (Windows only)	43
Installing from network drives	43
Using LiveCycle ES2 with a Luna HSM cluster	43
Manual use of Acrobat restricted.....	44

Temporary directory	44
LiveCycle ES2 IPv6 support.....	44
Server configuration for enabling CIFS	48
Creating a virtual interface (AIX, Linux, and Solaris only).....	48
Configuring Windows Server for CIFS	48
Processes with document form variables and digital signatures	49
2 LiveCycle ES2 Credentials and Certificates.....	53
Obtaining the Reader Extensions ES2 Rights credential.....	53
Obtaining digital certificates for use with Digital Signatures ES2	54
3 Creating the LiveCycle ES2 Database.....	55
Database configuration requirements	55
Creating an Oracle database	55
User account and rights.....	56
Creating a SQL Server database.....	56
Installing the SQL Server database driver	57
Setting up SQL Server for LiveCycle ES2.....	57
Sizing your SQL Server database.....	58
Creating the LiveCycle ES2 database user, schema, and login	59
Creating a DB2 database	61
DB2 user account	61
Additional DB2 requirements for LiveCycle Content Services ES2	65
Creating a MySQL database	66
Limitation on document usage in processes for MySQL databases.....	66
MySQL user account	66
MySQL initial server configuration requirements	67
4 Configuring a JBoss Application Server	68
Adobe-preconfigured JBoss.....	68
Installing the Sun Java JDK for JBoss.....	69
Logon scripts for Linux and Solaris.....	70
Starting and stopping JBoss.....	70
Notes about configuring the JBoss Application Server.....	71
Configuring the LiveCycle ES2 database connectivity for Adobe pre-configured JBoss 4.2.1	71
Configuring MySQL for Adobe-preconfigured JBoss 4.2.1	72
Configuring Oracle for Adobe-preconfigured JBoss 4.2.1	74
Configuring SQL Server for Adobe-preconfigured JBoss 4.2.1.....	76
Next steps.....	80
5 Configuring a WebLogic Server.....	81
Overview.....	81
LiveCycle ES2 requirements for WebLogic	81
Installing WebLogic Server	82
Configuring WebLogic 10g R3 on Solaris (64-bit Java)	82
Installing the JRockit Java JDK.....	82
Creating the WebLogic Server domain	84
Starting Node Manager.....	86
Creating a new WebLogic Managed Server	86
Configuring WebLogic Server.....	88
Configuring WebLogic for Web Services.....	88
Stopping and restarting WebLogic.....	89
Configuring the LiveCycle ES2 database connectivity	90

Next steps.....	90
6 Configuring a WebSphere Application Server	91
Installing WebSphere Application Server.....	91
Installing a Fix Pack to WebSphere.....	91
Setting the JAVA_HOME and PATH environment variables.....	91
Starting WebSphere Base and accessing WebSphere Administrative Console	92
Starting WebSphere ND and accessing WebSphere Administrative Console.....	93
Setting directory permissions.....	93
Preparing WebSphere Application Server	94
Required application server instances	94
Increasing the SOAP request time out	94
Increasing the Deployer heap size.....	95
Configuring the LiveCycle ES2 database connectivity	96
Next steps.....	96
7 Preconfigured Environment Checklist.....	97
A Appendix - Manually Configuring JBoss.....	99
Installing the JDK for JBoss	99
Set the JAVA_HOME environment variable (Windows)	99
Set the PATH environment variable (Windows)	100
Set the JAVA_HOME environment variable (Linux and Solaris).....	100
Set the PATH environment variable (Linux and Solaris)	100
Verify JAVA_HOME environment variable setting (Windows, Linux, or Solaris)	100
Manually installing JBoss.....	100
Download JBoss	100
Starting and stopping JBoss.....	101
Start JBoss	101
Stop JBoss	101
Modifying the JBoss configuration	101
Remove JMS and clustering configuration files.....	102
Disable cluster configuration.....	102
Modify the JBoss configuration	102
Changing the hibernate option (Content Services ES2 only)	108
LiveCycle ES2 database connectivity	108
Configuring MySQL for manually installed JBoss.....	108
Configuring Oracle for manually installed JBoss.....	111
Configuring SQL Server	114
Next steps.....	117

About This Document

This document provides the information that is required to prepare your server environment before you perform an automatic or manual installation and deployment of Adobe LiveCycle ES2 (Enterprise Suite) 9.0 on Microsoft® Windows®, Red Hat® Linux®, SUSE™ Linux, IBM® AIX®, and Sun™ Solaris™.

Before you start

Ensure that you have the most current version of the [LiveCycle ES2 documentation](#).

What's in this document?

This document contains the following types of information:

- All hardware and software requirements and configurations that *must* be already in place to ensure a successful LiveCycle ES2 installation process
- All tasks that *can* be performed without LiveCycle ES2 being installed and deployed but are needed for installing or deploying

For installations of Adobe LiveCycle Data Services ES2, see the documentation that is provided with the Data Services ES2 module.

After you complete the tasks in this document, proceed to the *Installing and Deploying LiveCycle ES2* document for your application server. See [“Additional information” on page 9](#) for a list of the available documentation.

Who should read this document?

This document provides information for administrators or developers who are responsible for preparing the application and database servers for development, staging, and production environments before installing, configuring, administering, and deploying LiveCycle ES2. The information provided is based on the assumption that anyone reading this document is familiar with JBoss, WebSphere, and Webogic application servers, Red® Hat® Linux®, SUSE™ Linux, Microsoft® Windows®, IBM AIX®, or Sun™ Solaris™ operating systems, MySQL, Oracle®, IBM DB2®, or SQL Server database servers, and web environments.

Conventions used in this document

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

Name	Description	Default value
<i>[LiveCycleES2 root]</i>	The installation directory that is used for all LiveCycle ES2 modules. The installation directory contains subdirectories for Adobe LiveCycle Configuration Manager, the LiveCycle ES2 SDK, and each LiveCycle ES2 module installed (along with the product documentation). This directory also includes directories relating to third-party technologies.	Windows: C:\Adobe\Adobe LiveCycle ES2\ AIX, Linux, and Solaris: /opt/adobe/adobe_lifecycle_es2/
<i>[appserver root]</i>	The home directory of the application server that runs the LiveCycle ES2 services.	JBoss 4.2.0 or 4.2.1 on Windows: C:\jboss JBoss 4.2.0 or 4.2.1 on Linux and Solaris: /opt/jboss JBoss Enterprise Application Platform 4.3 on Windows: C:\jboss-eap-4.3\jboss-as JBoss Enterprise Application Platform 4.3 on Linux and Solaris: /opt/jboss-eap-4.3/jboss-as 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, or, /opt/IBM/WebSphere/AppServer WebLogic 10g on Windows: C:\bea\wlserver_10.3 WebLogic 11g on Windows: C:\Oracle\Middleware\wlserver_10.3 WebLogic 10g on Linux and Solaris: /opt/bea/wlserver_10.3 WebLogic 11g on Linux and Solaris: /opt/Oracle/Middleware/wlserver_10.3

Name	Description	Default value
<i>WL_HOME</i>	The install directory for WebLogic as specified for the <i>WL_HOME</i> environment variable.	WebLogic 10g on Windows: C:\bea\ WebLogic 11g on Windows: C:\Oracle\Middleware\ WebLogic 10g on Linux and Solaris: /opt/bea/ WebLogic 11g on Linux and Solaris: /opt/Oracle/
<i>[appserverdomain]</i>	The domain that you configured on WebLogic. The default domain is called <i>base_domain</i> .	WebLogic 10g on Windows: C:\bea\user_projects\domains\base_domain WebLogic 11g on Windows: C:\Oracle\Middleware\user_projects\domains\ base_domain WebLogic 11g on Linux and Solaris: /opt/Oracle/user_projects/domains/base_dom ain
<i>[dbserver root]</i>	The location where the LiveCycle ES2 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 file names and paths are case-sensitive on AIX, Linux, and Solaris). Any platform-specific information is indicated as required.

Additional information

The resources in this table can help you learn about LiveCycle ES2.

For information about	See
LiveCycle ES2 and the modules in general	LiveCycle ES2 Overview
What's new in this LiveCycle ES2 release	What's New for LiveCycle ES2
LiveCycle ES2 terminology	LiveCycle ES2 Glossary
Other services and products that integrate with LiveCycle ES2	http://adobe.com/products/livecycle
Other LiveCycle ES2 modules	http://partners.adobe.com/public/developer/main.html
Installing LiveCycle ES2	Installing and Deploying LiveCycle ES2 for JBoss Installing and Deploying LiveCycle ES2 for WebSphere Installing and Deploying LiveCycle ES2 for WebLogic
Installing LiveCycle ES2 using the turnkey method	Installing and Deploying LiveCycle ES2 Using Turnkey
All documentation that is available for LiveCycle ES2	Adobe LiveCycle ES2 documentation
LiveCycle ES2 release information and last-minute changes that occur to the product	LiveCycle ES2 Release Notes
Patch updates, technical notes, and additional information about this product version	LiveCycle Technical Support

1

System Requirements

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 ES2, instead of installing from the installation media.

Also verify the MD5 checksum of the installation media matches the checksum posted at [Adobe website](#).

2. Verify that all system component versions comply with system requirements.
3. Prepare the hardware 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.

1.1 LiveCycle ES2 patch updates

Before you install LiveCycle ES2, ensure that you download any required patch updates, which are located at [LiveCycle Technical Support](#).

1.2 Third-party infrastructure support

1.2.1 Third-party patch support

The third-party reference platforms described in this document represent a specific patch level of third-party infrastructure that was current during the development and release of this version of Adobe LiveCycle ES2.

To review Adobe's policy regarding third-party patch support and software compatibility, see the knowledgebase article [Adobe LiveCycle Third-party Patch Support Policy](#).

1.2.2 Microsoft Windows operating systems

LiveCycle ES2 supports the following Microsoft Windows operating systems:

- Windows Server 2008 Standard and Enterprise Edition R2 (64-bit)
- Windows Server 2008 Standard and Enterprise Edition SP1 and later (64-bit) and VMWare ESX/GSX architectures
- Windows Server 2003 Standard and Enterprise Edition SP2 and later (32-bit and 64-bit)
- Windows Server 2003 R2 Standard SP2 and later (32-bit and 64-bit)

- Windows Server 2003 R2 Enterprise Edition SP2 and later running on 32-bit, 64-bit, and VMWare ESX/GSX architectures

Note: LiveCycle ES2 Server is supported in non-production environment on Microsoft Windows XP (SP2 or SP3), Windows Vista (SP1, all flavors, 32-bit and 64-bit), and Windows 7 (32-bit and 64-bit). However, these operating systems are supported for LiveCycle clients and for installing LiveCycle developer tools.

Note: LiveCycle Business Activity Monitoring ES2 is not supported on 32-bit systems; deploy Business Activity Monitoring ES2 on a 64-bit system.

1.2.2.1 Windows Server 2008

Application server	JDK	OS/JVM architecture	Database
Red Hat JBoss Application Server 4.2.1 and JBoss Enterprise Application Platform 4.3	Sun JDK 6.0 update 14 or later updates to 6.0	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> • MySQL 5.0.18 or 5.1.30 (recommended) • Microsoft SQL Server 2005 SP3, 2008 • Oracle 10g, 11g R2
Oracle WebLogic 10g R3 (Standard & Enterprise editions)	Oracle JRockit® Real Time 3.1.2-1.6.0.14	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> • MySQL 5.0.18 or 5.1.30 (recommended) • IBM DB2 9.1 or later versions of 9.x • Oracle 10g, 11g R2 • Microsoft SQL Server 2005 SP3, 2008
New for 9.0.0.2 Oracle WebLogic 11g R1 - 10.3.3	Oracle JRockit® Real Time 3.1.2-1.6.0.14	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> • MySQL 5.0.18 or 5.1.30 (recommended) • IBM DB2 9.1 or later versions of 9.x • Microsoft SQL Server 2005 SP3, 2008 • Oracle 10g, 11g R2
New for 9.0.0.2 IBM WebSphere 7.0.0.9 (Base and Network Deployment Edition)	WebSphere Java SDK 1.6 SR7	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> • IBM DB2 9.1 or later versions of 9.x • Oracle 10g, 11g R2 • MS SQL Server 2005 SP3, 2008

Note: When using PDF Generator ES2 with a 64-bit application server, an additional 32-bit JVM must also be installed.

1.2.2.2 Windows Server 2003

Application server	JDK	OS/JVM architecture	Database
Red Hat JBoss Application Server 4.2.0	Sun JDK 5.0 update 11 or later updates to 5.0	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> MySQL 5.0.18 or 5.1.30 (recommended) Microsoft SQL Server 2005 SP3, 2008 Oracle 10g, 11g R2
Red Hat JBoss Application Server 4.2.1 and JBoss Enterprise Application Platform 4.3	Sun JDK 6.0 update 14 or later updates to 6.0	32-bit OS and 32-bit JVM	<ul style="list-style-type: none"> MySQL 5.0.18 or 5.1.30 (recommended)
Red Hat JBoss Application Server 4.2.1 and JBoss Enterprise Application Platform 4.3	Sun JDK 6.0 update 14 or later updates to 6.0	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> Microsoft SQL Server 2005 SP3, 2008 Oracle 10g, 11g R2 MySQL 5.0.18 and 5.1.30
Oracle WebLogic 10g R3 (Standard & Enterprise editions)	Oracle JRockit® Real Time 3.1.2-1.6.0.14	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> MySQL 5.0.18 or 5.1.30 (recommended) IBM DB2 9.1 or later versions of 9.x Oracle 10g, 11g R2 Microsoft SQL Server 2005 SP3, 2008
New for 9.0.0.2 Oracle WebLogic 11g R1 - 10.3.3	Oracle JRockit® Real Time 3.1.2-1.6.0.14	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> MySQL 5.0.18 or 5.1.30 (recommended)
IBM WebSphere 6.1.0.21 (Base & Network Deployment Edition)	WebSphere Java SDK 1.5 SR8	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> IBM DB2 9.1 or later versions of 9.x Oracle 10g, 11g R2 Microsoft SQL Server 2005 SP3, 2008
New for 9.0.0.2 IBM WebSphere 7.0.0.9 (Base and Network Deployment Edition)	WebSphere Java SDK 1.6 SR7	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> IBM DB2 9.1 or later versions of 9.x Oracle 10g, 11g R2 Microsoft SQL Server 2005 SP3, 2008

Note: When using PDF Generator ES2 with a 64-bit application server, an additional 32-bit JVM must also be installed. For a detailed list of supported software, see [“Supported software” on page 22](#).

1.2.3 Linux operating systems

LiveCycle ES2 supports the following:

- Red Hat Enterprise Linux AP or ES 5 (Intel/AMD 64-bit architectures)
- SUSE Linux Enterprise Server 10.0 and 11.0 (Intel/AMD 64-bit architectures)

Note: PDF Generator 3D ES2 is not supported on non-Windows platforms.

Note: When using PDF Generator ES2 with a 64-bit application server, an additional 32-bit JVM must also be installed.

Note: On Linux operating systems, you must ensure that X Window libraries are installed. This is required for PDF Generator ES2 and Forms ES2. See documentation for your operating system for more information.

1.2.3.1 Red Hat Enterprise Linux

Caution: Ensure that the cURL, Xorg-x11-apps, and Compat-libstdc++ packages and all dependencies are installed on your Red Hat Linux computer to avoid problems when rendering forms.

Application server	JDK	OS/JVM architecture	Database
Red Hat JBoss Application Server 4.2.0	Sun JDK 5.0 update 11 or later updates to 5.0	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> ● MySQL 5.0.18 or 5.1.30 (recommended) ● Oracle 10g, 11g R2
Red Hat JBoss Application Server 4.2.1 and JBoss Enterprise Application Platform 4.3	Sun JDK 6.0 update 14 or later updates to 6.0	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> ● MySQL 5.0.18 or 5.1.30 (recommended) ● Oracle 10g, 11g R2
Oracle WebLogic 10g R3 (Standard & Enterprise editions)	Oracle JRockit® Real Time 3.1.2-1.6.0.14	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> ● Oracle 10g, 11g R2
New for 9.0.0.2 Oracle WebLogic 11g R1 - 10.3.3	Oracle JRockit® Real Time 3.1.2-1.6.0.14	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> ● Oracle 10g, 11g R2
IBM WebSphere 6.1.0.21 (Base & Network Deployment Edition)	WebSphere Java SDK 1.5 SR8	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> ● IBM DB2 9.1 or later versions of 9.x ● Oracle 10g, 11g R2
New for 9.0.0.2 IBM WebSphere 7.0.0.9 (Base and Network Deployment Edition)	WebSphere Java SDK 1.6 SR7	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> ● IBM DB2 9.1 or later versions of 9.x ● Oracle 10g, 11g R2

1.2.3.2 SUSE Linux

Note: Ensure that X Window libraries are installed on your operating system. This is required for PDF Generator ES2 and Forms ES2. See documentation for your operating system for more information.

Caution: You must install the glibc-locale-32bit library that ships with SUSE Linux Enterprise Server or LiveCycle ES2 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 10.0 documentation.)

Note: If you plan to install LiveCycle ES2 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

Application server	JDK	OS/JVM architecture	Database
Red Hat JBoss Application Server 4.2.1 and JBoss Enterprise Application Platform 4.3	Sun JDK 6.0 update 14 or later updates to 6.0	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> MySQL 5.0.18 or 5.1.30 (recommended) Oracle 10g, 11g R2
New for 9.0.0.2 IBM WebSphere 7.0.0.9 (Base and Network Deployment Edition)	WebSphere Java SDK 1.6 SR7	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> IBM DB2 9.1 or later versions of 9.x Oracle 10g, 11g R2

1.2.4 IBM AIX

LiveCycle ES2 supports AIX 5L 5.3 and AIX 6.1 (both 64-bit architecture).

Note: PDF Generator 3D ES2 is not supported on non-Windows platforms.

Note: Ensure that X Window libraries are installed on your operating system. This is required for PDF Generator ES2 and Forms ES2. See the documentation for your operating system for more information.

Note: For a detailed list of the supported platforms, see [“Supported software” on page 22](#).

1.2.4.1 AIX 5.3

Application server	JDK	OS/JVM architecture	Database
IBM WebSphere 6.1.0.21 (Base and Network Deployment Edition)	WebSphere Java SDK 1.5 SR8	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> IBM DB2 9.1 or later versions of 9.x Oracle 10g, 11g R2
<p>*New for 9.0.0.2*</p> IBM WebSphere 7.0.0.9 (Base and Network Deployment Edition)	WebSphere Java SDK 1.6 SR7	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> IBM DB2 9.1 or later versions of 9.x Oracle 10g, 11g R2

1.2.4.2 AIX 6.1

Application server	JDK	OS/JVM architecture	Database
<p>*New for 9.0.0.2*</p> IBM WebSphere 7.0.0.9 (Base and Network Deployment Edition)	WebSphere Java SDK 1.6 SR7	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> IBM DB2 9.1 or later versions of 9.x Oracle 10g, 11g R2

Note: For a detailed list of the supported database editions, see [“Supported software” on page 22](#).

1.2.5 Sun Solaris

LiveCycle ES2 supports the Sun Solaris 10 (SPARC® architectures) operating system. See also [“Additional requirements for AIX, Linux, and Solaris” on page 18](#).

Note: PDF Generator 3D ES2 is not supported on non-Windows platforms.

Note: Ensure that X Window libraries are installed on your operating system. This is required for PDF Generator ES2 and Forms ES2. See the documentation for your operating system for more information.

Caution: 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.

Application server	JDK	OS/JVM architecture	Database
Red Hat JBoss Application Server 4.2.0	Sun JDK 5.0 update 11 or later updates to 5.0	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> Oracle 10g, 11g R2
Red Hat JBoss Application Server 4.2.1 and JBoss Enterprise Application Platform 4.3	Sun JDK 6.0 update 14 or later updates to 6.0	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> MySQL 5.0.18 or 5.1.30 (recommended) Oracle 10g, 11g R2
Oracle WebLogic 10g R3 (Standard & Enterprise editions)	Sun JDK 6.0 update 7	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> Oracle 10g, 11g R2
New for 9.0.0.2 Oracle WebLogic 11g R1 - 10.3.3	Sun JDK 6.0 update 14 or later updates to 6.0	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> Oracle 10g, 11g R2
IBM WebSphere 6.1.0.21 (Base & Network Deployment Edition)	WebSphere Java SDK 1.5 SR8	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> IBM DB2 9.1 or later versions of 9.x Oracle 10g, 11g R2
New for 9.0.0.2 IBM WebSphere 7.0.0.9 (Base and Network Deployment Edition)	WebSphere Java SDK 1.6 SR7	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> IBM DB2 9.1 or later versions of 9.x Oracle 10g, 11g R2

Note: When using PDF Generator ES2 with a 64-bit application server, an additional 32-bit JVM must also be installed. For a detailed list of supported software, see [“Supported software” on page 22](#).

1.3 System requirements

1.3.1 Minimum hardware requirements

This table provides the minimum hardware requirements that LiveCycle ES2 supports.

Operating system	Minimum hardware requirement
Microsoft Windows Server® 2008 R1/R2	Intel® Pentium® 4, 2.8 GHz processor or equivalent VMWare ESX 3.0 or later RAM: 4 GB Free disk space: 5.4 GB of temporary space plus 3.4 GB for LiveCycle ES2
Microsoft Windows Server® 2003 Enterprise Edition or Standard Edition SP2 and R2 (32- and 64-bit architectures)	Intel® Pentium® 4, 2.8 GHz processor or equivalent VMWare ESX 3.0 or later RAM: 3 GB (32-bit) or 4 GB (64-bit) Free disk space: 5.4 GB of temporary space plus 3.4 GB for LiveCycle ES2 RAM requirements if Business Activity Monitoring ES2 is deployed - 8 GB of RAM for heavy development; 16 GB of RAM for production environment; 2GB of RAM for evaluation
Sun Solaris 10	UltraSPARC® IIIi, 1.5 GHz processor Solaris Containers (Zones) partitioning RAM: 3GB (64-bit OS with 64-bit JVM) Free disk space: 5.4 GB of temporary space plus 3.4 GB for LiveCycle ES2
IBM AIX 5L 5.3	P4 pSeries 615 (Model 6C3) 7029-6C3, 1.2 GHz processor LPAR partitioning RAM: 3GB (64-bit OS with 64-bit JVM) Free disk space: 5.4 GB of temporary space plus 3.4 GB for LiveCycle ES2
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: 3GB (64-bit OS with 64-bit JVM) Free disk space: 5.4 GB of temporary space plus 3.4 GB for LiveCycle ES2
Red Hat Enterprise Linux AP or ES 5 (64-bit edition only)	Dual core, 1 GHz processor VMWare ESX 3.0 or later RAM: 3GB (64-bit OS with 64-bit JVM) Free disk space: 5.4 GB of temporary space plus 3.4 GB for LiveCycle ES2

Note: LiveCycle Business Activity Monitoring ES2 is not supported on 32-bit systems; deploy Business Activity Monitoring ES2 on a 64-bit system. Even though the LiveCycle ES2 installer can install the Business Activity Monitoring ES2 file set on a 32-bit system, it fails during configuration.

1.3.2 Intel x86 compatibility

On supported Windows and UNIX -like environments, LiveCycle ES2 supports Intel and AMD64 compatible chipsets running either 32-bit or 64-bit supported operating systems.

1.3.3 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: 3.2 GB of RAM.

1.3.4 Additional requirements for AIX, Linux, and Solaris

Caution: On AIX, Linux, and Solaris operating systems, use binary mode when downloading the installer from Adobe web site.

1.3.4.1 Installing and configuring UTF-8

When installing LiveCycle ES2 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.

► To install 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.
3. (AIX 5.3 and 6.1) 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.

► **To install UTF-8 on Solaris 10:**

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](#).

1.3.4.2 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.

► **Modify the `rlim` values:**

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

If any of the values fail to match the updated settings, ensure you have performed the steps as described and restart your computer.

1.3.5 Additional hardware requirement for LiveCycle Content Services ES2

If you are installing LiveCycle Content Services ES2 for use with a DB2 database, you must have a minimum of 2 GB of RAM on the database computer.

1.3.6 Recommended hardware requirements for client-side computers

Workbench ES2

- Disk space for installation:
 - 1.5 GB for Workbench ES2 only
 - 1.7 GB on a single drive for a full installation of Workbench ES2, Designer ES2, 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 ES2: 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
- TCP/IPv4 or TCP/IPv6 network connection to the LiveCycle ES2 server

Note: You must have Administrative privileges to install Workbench ES2 on Windows. If you are installing using a non-administrator account, the installer will prompt you for the credentials for an appropriate account.

Designer ES2

- A minimum of Adobe Reader 9.3. Acrobat 9.3 Pro Extended (recommended) is required to benefit from all the new features in Designer ES2, including the 3D features.
- Adobe Flash Player 9.0 or later.
- (Optional) Flex Builder™ 3.0 or later.
- (Optional) Flex SDK 3.4 (required for customizing form guide components that are shipped with Designer ES2).

Note: Use the Flex SDK 3.4 included with the Designer ES2 installer only. Do not use any version of Flex SDK obtained from the Adobe web site.

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

End-user hardware requirements:

- LiveCycle Workspace ES2: 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 [“End-user user interface” on page 28](#).

1.3.7 Supported software

Required software	Supported version
Operating system	<ul style="list-style-type: none"> ● Microsoft Windows <ul style="list-style-type: none"> ● Windows Server 2008 Enterprise Edition or Standard Edition SP1 or later (64-bit) ● Windows Server 2008 R2 Standard Edition or Enterprise Edition ● Windows Server 2003 Enterprise Edition or Standard Edition SP2 and later (32-bit and 64-bit) ● Windows Server 2003 R2 Standard Edition SP2 and later (32-bit and 64-bit) ● Windows Server 2003 R2 Enterprise Edition SP2 and later running on 32-bit, 64-bit, and VMWare ESX/GSX architectures ● (PDF Generator 3D ES2) Microsoft Windows <ul style="list-style-type: none"> ● Windows Server 2008 Enterprise Edition or Standard Edition SP1 or later (64-bit) ● Windows Server 2008 R2 Standard Edition or Enterprise Edition ● Windows Server 2003 Enterprise Edition or Standard Edition SP2 and later (32-bit and 64-bit) ● Windows Server 2003 R2 Standard Edition SP2 and later (32-bit and 64-bit) ● Windows Server 2003 R2 Enterprise Edition SP2 and later running on 32-bit, 64-bit, and VMWare ESX/GSX architectures ● (Workbench ES2) Microsoft Windows <ul style="list-style-type: none"> ● Windows 7 32-bit and 64-bit (Home Basic, Home Premium, Professional, and Enterprise Editions) ● Windows Vista 32-bit and 64-bit (Business, Home Basic, Home Premium, and Ultimate editions) ● Windows XP Professional SP2 and above ● Windows Server 2003 Enterprise Edition or Standard Edition SP2 and later (32-bit and 64-bit) ● Windows Server 2003 R2 Standard Edition SP2 and later (32-bit and 64-bit) ● Windows Server 2008 Enterprise Edition or Standard Edition SP1 or later ● Windows Server 2008 R2 Standard or Enterprise Edition ● Sun Solaris 10 (64-bit edition only) ● IBM AIX 5L 5.3 and 6.1 (64-bit edition only) ● SUSE Linux Enterprise Server 10.0 (64-bit edition only) ● SUSE Linux Enterprise Server 11.0 (64-bit edition only) ● Red Hat Enterprise Linux AP or ES 5 (64-bit edition only)

Required software	Supported version
Application server	<ul style="list-style-type: none"> ● JBoss Application Server 4.2.0 and 4.2.1 ● JBoss Enterprise Application Platform 4.3 ● IBM WebSphere 6.1.0.21 (Base and Network Deployment Edition) ● IBM WebSphere 7.0.0.9 (Base & Network Deployment Edition) ● Oracle WebLogic 10g R3 (Standard and Enterprise editions) ● Oracle WebLogic 11g R3 - 10.3.3
Web browser	For a list of web browsers, see “Web browser support” on page 28 .
JDK	<ul style="list-style-type: none"> ● JBoss 4.2.1 and JBoss EAP 4.3 on all platforms: Sun Java SE Development Kit (JDK) 6.0 update 14 or later updates to 6.0 or later available at Sun Developer Network. ● WebLogic on Windows and Linux: Oracle JRockit® Real Time 3.1.2-1.6.0.14 available from Oracle. ● WebLogic on Solaris: Sun Java SE Development Kit (JDK) 6.0 update 7 or later available from Sun Developer Network. Requires the Daylight Savings Time (TZ) update, available at Sun Updater Tool, to be run in North America. ● WebSphere 6.1 on all platforms: WebSphere Java SDK 1.5 SR8. ● WebSphere 7.0 on all platforms: WebSphere Java SDK 1.6 SR7. <p>To use AES 256 encryption with LiveCycle Rights Management ES2, obtain and install the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy files from the Java SE Downloads.</p> <p>Note: These requirements are optional and required only if you need to use Advanced Encryption Standard (AES) 256.</p>
32-bit JDK (required for 64-bit application server only)	<ul style="list-style-type: none"> ● Sun JDK 6 update 14 or later updates to 6.0 available at Sun Developer Network.
Database	<ul style="list-style-type: none"> ● IBM DB2 9.1 or later versions of 9.x(Enterprise Edition) ● Microsoft SQL Server 2005 SP3 or 2008 (Standard and Enterprise Edition) ● MySQL 5.0.18 or 5.1.30 (recommended) InnoDB Engine ● Oracle 10g or 11g R2 (Standard and Enterprise Editions) <p>Note: LiveCycle ES2 is compliant with the database vendors' compatibility statements. (See the vendors' websites.)</p>

Required software	Supported version
Database driver	<ul style="list-style-type: none"> ● IBM DB2 9.x driver is required for all supported versions of DB2: db2jcc.jar (version 3.50.152) ● SQL Server JDBC 1.2: sqljdbc.jar for both Microsoft SQL Server 2005 SP3 and Microsoft SQL Server 2008. ● MySQL: Connector/J 5.1.6: mysql-connector-java-5.1.6-bin.jar ● Oracle 10g and 11g R2: ojdbc5.jar (release 11.1.0.6) for JDK 1.5 or ojdbc6.jar (release 11.1.0.6) for JDK 1.6 <p>Note: The LiveCycle installation media includes these database drivers in the <i>[DVD_root]/third_party/db/database</i> directory.</p> <p>Note: All LiveCycle Business Activity Monitoring ES2 metadata databases use the drivers listed above.</p>
LDAP server	<ul style="list-style-type: none"> ● Sun ONE 5.1, 5.2, 6.x ● Microsoft Active Directory 2003 and 2008 ● Novell® eDirectory 8.7.3 ● IBM Tivoli Directory Server 6.0 ● IBM Domino Enterprise Server 8.0 and later updates of 8.0
Email servers	<ul style="list-style-type: none"> ● Microsoft Exchange 2000, 2003, 2007 ● Lotus Notes/Domino 6/7 ● SendMail (included with Red Hat 5) ● Novell GroupWise 6/7 ●
LiveCycle Data Services ES2	<ul style="list-style-type: none"> ● Adobe Flex® Builder™ 2.0.1 Hot Fix 2 or later ● Adobe Flash Builder 4 or later <p>or</p> <ul style="list-style-type: none"> ● Flex SDK 2.0.1 Hot Fix 2 or later ● Flex is required for these tasks: <ul style="list-style-type: none"> ● Using LiveCycle Data Services ES2 ● Customizing form guides in LiveCycle Designer ES2 ● Customizing LiveCycle Workspace ES2 ● Creating Flex applications for LiveCycle Workspace ES2 ● Calling LiveCycle ES2 APIs using Flex

Required software	Supported version
LiveCycle ES2 connector	<p>Connector for EMC® Documentum®:</p> <ul style="list-style-type: none"> ● EMC Documentum Content Server 6.0 ● EMC Documentum Content Server 6.5 <p>In addition, on your LiveCycle ES2 server, install the version of EMC Documentum Foundation Classes (DFC) that corresponds to your version of Content Server. (See <i>Documentum Foundation Classes Installation Guide</i> available from EMC Documentum.)</p> <p>Connector for IBM FileNet:</p> <ul style="list-style-type: none"> ● IBM FileNet P8 Content Engine 4.0.x, 4.5 and 5.0 ● IBM FileNet P8 Process Engine 4.0.x, 4.5 and 5.0 <p>Note: The version of Process Engine should correspond to the version of your Content Engine (for example, Process Engine 4.0.x for Content Engine 4.0.x).</p> <p>In addition, on your LiveCycle ES2 server, install the version of IBM FileNet P8 Content Java API that corresponds to your version of Content Engine (for example, 4.0.x Content Engine Java API or 4.5 Content Engine Java API). (See “Additional requirements for LiveCycle ES2 Connector for IBM FileNet” on page 37.) For a list of the required JAR files, see <i>Content Java API Developer's Guide</i> available from IBM FileNet.</p> <p>Connector for IBM Content Manager:</p> <ul style="list-style-type: none"> ● IBM Content Manager 8.4 (version 8.4 - 8.4.1.1 only) <p>In addition, on your LiveCycle ES2 server, install the version of IBM software that corresponds to your version of IBM Content Manager:</p> <ul style="list-style-type: none"> ● DB2 Universal Database Client (not required if IBM Content Manager is on same server as the LiveCycle ES2 server) ● Information Integrator for Content (II4C) available from IBM <p>Note: IBM Content Manager running on Oracle database is not supported.</p> <p>Connector for Microsoft SharePoint:</p> <ul style="list-style-type: none"> ● Microsoft SharePoint Server 2007 ● Microsoft SharePoint Server 2010 <p>In addition, the server running SharePoint Server must have Microsoft .NET Framework 3.5 installed.</p>
PDF client	<ul style="list-style-type: none"> ● Adobe Acrobat® Professional, Acrobat® Standard, and Acrobat Pro Extended, versions 8.0 to 9.3. ● Adobe Reader, versions 8.0 to 9.3. <p>Note: For policy protecting a document, you require Acrobat Professional, Acrobat Standard, or Acrobat Pro Extended, versions 8.0 to 9.3. Adobe Reader doesn't offer the capabilities to policy-protect documents.</p> <ul style="list-style-type: none"> ● Apple® QuickTime 7 Player or Pro (for converting embedded video to PDF multimedia)

Note: LiveCycle Business Activity Monitoring ES2 is not supported on 32-bit systems. You must deploy it on a 64-bit system. Even though the LiveCycle ES2 installer can install the Business Activity Monitoring ES2 file set on a 32-bit system, it fails during configuration.

1.3.8 Installation user account for Windows

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

1.3.9 Configuration for 64-bit Windows installations

On 64-bit Windows Server 2008, Windows Vista, or Windows 7, modify the Admin Approval Mode security option as follows:

1. Go to **Start > Control Panel > 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 prompt**.
3. Restart your computer.

Caution: The Windows UAC must remain disabled for PDF Generator ES2 or PDF Generator 3D ES2 to work properly. During the install and configuration process, turning on the UAC with the *Elevate without prompt* option enabled is sufficient, but the UAC must be completely disabled in order to run PDF Generator ES2 or PDF Generator 3D ES2.

If you are installing and configuring on an evaluation system, you can enable the UAC on the computer after you have deployed to your production computer or uninstalled PDF Generator ES2 or PDF Generator 3D ES2.

► **Disable the Windows UAC on Vista:**

1. To access the System Configuration Utility, go to **Start > Run** and in the **Open:** box 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.

► **Disable the Windows UAC on Server 2008 or Windows 7:**

1. Go to **Start > Control Panel > System and Security**, and under Action Center, select **Change User Account Control Settings**.
2. On the **Choose when to be notified about change to your computer** screen, move the slider to **Never Notify**.
3. Click OK.

To enable the UAC again, repeat the steps above and select the **Use User Account Control (UAC) to help protect your computer** option before restarting the computer.

1.3.10 Minimum database user permissions

Database	Initialization permissions	Runtime permissions
Oracle	CREATE SESSION CREATE CLUSTER 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 "DB2 user account" on page 61 for a complete description.	See "DB2 user account" on page 61 for a complete description.

1.3.11 Web browser support

1.3.11.1 End-user user interface

End-user components include these modules:

- LiveCycle Workspace ES2 (Flash Player required)

Note: Adobe Flash Player 9.0.115.0 or later is required for Workspace ES2 or for using form guides in Workspace ES2.

- LiveCycle Reader Extensions ES2 (Flash Player required)
- LiveCycle Rights Management ES2 (Flash Player required)
- LiveCycle PDF Generator ES2 and LiveCycle PDF Generator 3D ES2 (browser only)
- LiveCycle Content Services ES2 (browser only)

Operating system	Flash Player	Supported browser
Microsoft Windows 7	Flash Player 10	Microsoft Internet Explorer 8 Firefox 3.0 or later ⁽¹⁾
Microsoft Windows Vista™	Flash Player 9 or 10	Microsoft Internet Explorer 7 or later ⁽¹⁾ Firefox 3.0 or later ⁽¹⁾
Windows 2000	Flash Player 9 or 10	Internet Explorer 6 or later ⁽¹⁾ Firefox 3.0 or later ⁽¹⁾
Windows XP	Flash Player 9 or 10	Internet Explorer 6 or later ⁽¹⁾ Firefox 3.0 or later ⁽¹⁾
Windows Server 2008	Flash Player 10	Internet Explorer 8 Firefox 3.0 or later ⁽¹⁾
Windows Server 2003	Flash Player 9 or 10	Internet Explorer 6 or later ⁽¹⁾ Firefox 3.0 or later ⁽¹⁾
Mac OS X v 10.4.x or 10.5.x (PowerPC)	Flash Player 9 or 10	Firefox 3.0 or later (not for Workspace ES2) ⁽¹⁾⁽²⁾ Safari 3.x or 4.x (Workspace ES2 and Content Services ES2 require version 3.0.3 or later)
Mac OS X v 10.4.x, 10.5.x., or 10.6.x (Intel)	Flash Player 9 or 10	Firefox 3.0 or later (not for Workspace ES2) ⁽¹⁾⁽²⁾ Safari 3.x or 4.x (Workspace ES2 and Content Services ES2 require version 3.0.3 or later)

⁽¹⁾“or later” includes major revisions. For example, Internet Explorer 6 or later also covers Internet Explorer 7 and 8.

⁽²⁾Workspace ES2 supports Internet Explorer and Firefox on Windows but only Safari 3.0.3 or later on the Mac.

- LiveCycle Forms ES2

Operating system	Flash Player	Supported browser
Microsoft Windows 7	N/A	Microsoft Internet Explorer 8 Firefox 3.0 or later ⁽¹⁾ Netscape 8.x or later
Microsoft Windows Vista™	N/A	Microsoft Internet Explorer 6 or later ⁽¹⁾ Firefox 3.0 or later ⁽¹⁾ Netscape 8.x or later
Windows XP	N/A	Microsoft Internet Explorer 6 or later ⁽¹⁾ Firefox 3.0 or later ⁽¹⁾ Netscape 8.x or later
Windows Server 2008	N/A	Firefox 3.0 or later ⁽¹⁾
Windows Server 2003	N/A	Firefox 3.0 or later ⁽¹⁾
Linux (Intel)	N/A	Firefox 3.0 or later ⁽¹⁾ Netscape 8.x or later
Mac OS X v 10.4.x or 10.5.x (PowerPC)	N/A	Safari 3.x or 4.x
Mac OS X v 10.4.x, 10.5.x., or 10.6.x (Intel)	N/A	Safari 3.x or 4.x

⁽¹⁾“or later” includes major revisions. For example, Internet Explorer 6 or later also covers Internet Explorer 7 and 8.

- LiveCycle Business Activity Monitoring ES2

Operating system	Flash Player	Supported browser
Microsoft Windows Vista™	Flash Player 9 or 10	Microsoft Internet Explorer 7 or later ⁽¹⁾
Windows XP	Flash Player 9 or 10	Microsoft Internet Explorer 6 or later ⁽¹⁾
Windows Server 2000	Flash Player 9 or 10	Microsoft Internet Explorer 6 or later ⁽¹⁾

⁽¹⁾“or later” includes major revisions. For example, Internet Explorer 6 or later also covers Internet Explorer 7 and 8.

1.3.11.2 Administrator user interface

Operating system	Flash Player	Supported browser
Microsoft Windows 7	N/A	Internet Explorer 8 Firefox 3.0 or later ⁽¹⁾
Microsoft Windows Vista	N/A	Internet Explorer 7 or later ⁽¹⁾ Firefox 3.0 or later ⁽¹⁾
Windows 2000	N/A	Internet Explorer 6 or later ⁽¹⁾ Firefox 3.0 or later ⁽¹⁾
Windows XP	N/A	Internet Explorer 6 or later ⁽¹⁾ Firefox 3.0 or later ⁽¹⁾
Windows Server 2008	N/A	Internet Explorer 8 Firefox 3.0 or later ⁽¹⁾
Windows Server 2003	N/A	Internet Explorer 6 or later ⁽¹⁾ Firefox 3.0 or later ⁽¹⁾
Linux (Intel)	N/A	Firefox 3.0 or later ⁽¹⁾

⁽¹⁾“or later” includes major revisions. For example, Microsoft Internet Explorer 6 or later also covers Microsoft Internet Explorer 7 and 8.

1.3.11.3 Using Content Services ES2 with Firefox

Due to certain restrictions to how Firefox handles local links beginning with `file:///`, in order to use Content Services ES2 with Firefox you must update your Firefox installation by downloading and installing the plug-in from the following location: <http://forge.alfresco.com/projects/firefox-ext/>

1.3.12 Additional requirements for PDF Generator ES2 and PDF Generator 3D ES2

Note: PDF Generator 3D ES2 is supported on Windows environments only.

Note: Ensure that X Window libraries are installed on your operating system. This is required for PDF Generator ES2 and Forms ES2. See documentation for your operating system for more information.

Note: You cannot use the Shared Printer Protocol for the `SendToPrinter` API on Windows 2008 machines that have PDF Generator ES2 deployed on them. Use alternate protocols like CIFS or Direct IP.

1.3.12.1 User account for Windows

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

- Installing Microsoft Office

- Installing PDF Generator ES2 or PDF Generator 3D ES2
- Installing Acrobat 9.3 Professional Extended for PDF Generator ES2 or PDF Generator 3D ES2
- Running the application server process

1.3.12.2 Using 64-bit application servers with PDF Generator

If you are using a 64-bit application server on a Windows or UNIX-like system with PDF Generator ES2 or PDF Generator 3D ES2, 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 Windows and UNIX-like 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, except for AIX.

Caution: 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.

► **To set the Windows `JAVA_HOME_32` variable:**

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 `JAVA_HOME_32`. This directory is the directory that contains the JDK. For example, type the following code:

```
D:\Program Files (x86)\Java\jdk1.6.0_14
```

► **To 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_14  
export JAVA_HOME_32
```

1.3.12.3 Native file conversion software installation

Before you install PDF Generator ES2 or PDF Generator 3D ES2, 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.

You must activate one license on LiveCycle ES2 server for each native application that PDF Generator ES2 or PDF Generator 3D ES2 supports. Refer to the individual licensing agreement for each native application that your LiveCycle ES2 deployment will support, and ensure that your LiveCycle ES2 deployment meets the licensing requirements specified. Typically, each LiveCycle ES2 user who will use native application support must also have an activated license on their own computer for the native application.

PDF Generator ES2 or PDF Generator 3D ES2 can be extended to convert these additional file types to PDF files by using the following applications:

- Microsoft Office 2003, 2007 (DOC, XLS, PPT, RTF, TXT, Microsoft Office open XML Formats)
- Microsoft Office Visio 2003, 2007 (VSD)
- Microsoft Publisher 2003, 2007 (PUB)
- Microsoft Project 2003, 2007 (MPP)
- AutoCAD 2005, 2006, 2007, 2008 (DWG, DXF, DWF)

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.

- Corel WordPerfect 12, X4 (WPD)
- Adobe Photoshop® CS2
- Adobe FrameMaker® 7.2, 8.0 (FM)
- Adobe PageMaker® 7.0 (PMD, PM6, P65, PM)
- OpenOffice 2.4.2, 3.1 (ODT, ODP, ODS, ODG, ODF, SXW, SXI, SXC, SXD, SXM).

Note: OpenOffice 3.1 or later must be installed on the server to convert the documents created in version 3.1. OpenOffice 2.4.2 cannot convert documents created in later versions of OpenOffice.

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

1.3.12.4 Acrobat

PDF Generator ES2 or PDF Generator 3D ES2 requires that Acrobat Pro Extended is installed. You must install Acrobat before you run the LiveCycle ES2 installer.

The LiveCycle ES2 installer sets the `Acrobat_PATH` environment variable automatically. If you need to set it manually, see “Setting environment variables” in the [Installing and Deploying LiveCycle ES2](#) guide for your application server. You will need to restart the application server.

Note: The environment variable `Acrobat_PATH` is case-sensitive.

To use AES 256 encryption in LiveCycle Encryption service to encrypt a PDF document, you must obtain and install the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy files. For example, for Sun JDK, download JCE files from [Java SE Downloads](#).

After downloading the policy file, replace the existing `local_policy.jar` and `US_export_policy.jar` files under `[JAVA_HOME]/jre/lib/security` folder with the downloaded JAR files.

Note: On a 64-bit Windows server using LiveCycle Turnkey installation, copy the downloaded JAR files under `[LiveCycleES2 root]/Java/jdk1.6.0_14/jre/lib/security`.

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

1.3.12.5 QuickTime 7

PDF Generator ES2 requires that QuickTime 7 (Player or Pro) is 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.

1.3.12.6 Setting Windows environment variables

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

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 ES2 or PDF Generator 3D ES2. 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\OpenOffice.org 3.0\`.

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.

Note: The `Acrobat_PATH` environment variable is case-sensitive.

► 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 directory is the directory that contains the executable file. For example, type the following code:

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

1.3.12.7 Network Printer Client installation

PDF Generator ES2 includes the Network Printer Client installer for installation of the PDF Generator ES2 Internet printer. After the installation is completed, a PDF Generator ES2 printer is added to the list of existing printers on the clients computer. This printer can then be used to send documents for conversion to PDF. For more information about installing the Network Printer Client, see [Installing and Deploying LiveCycle ES2](#) document for your application server.

Note: The PDF Generator ES2 Network Printer Client is only supported on the following 32-bit Windows platforms: Windows XP, Windows 2000, Windows Server 2003, Windows Vista.

1.3.12.8 Service Control Manager command line tool

Before you complete an automatic installation of PDF Generator ES2 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.

1.3.12.9 Headless mode configuration

If you are running PDF Generator ES2 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. For more information, see the Help system for your operating system.

1.3.12.10 Enabling multi-threaded file conversions

By default, PDF Generator ES2 can convert only one OpenOffice, Microsoft Word, or PowerPoint document at a time. If you enable multi-threaded conversions, PDF Generator ES2 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 (both 2003 and 2007 versions) are not supported. To enable multi-threaded file conversions, upgrade to Microsoft Word 2007 and PowerPoint 2007.

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 ES2 server computer.

On Windows platforms, you must add the right to replace the process-level token. (See [“Granting the Replace a process level token privilege \(Windows only\)” on page 35.](#))

On 64-bit Windows, the Windows UAC must be disabled. (See [“Configuration for 64-bit Windows installations” on page 26.](#))

On all platforms, you must also configure the user permissions. (See [“Multi-user support for PDF Generator ES2” on page 35.](#)) When adding users for OpenOffice, Microsoft Word, or Microsoft PowerPoint on Windows 2003 or 2008, or for OpenOffice on Linux or Solaris, dismiss the initial activation dialogs for all users.

After your LiveCycle ES2 server is configured, you must add LiveCycle ES2 user accounts in LiveCycle Administration Console and on AIX, Linux, or Solaris platforms, turn off password prompting. See the “Configuring user accounts for multi-threaded file conversions” in the [Installing and Deploying LiveCycle ES2](#) guide for your application server.

Additional configuration required for OpenOffice on Linux or Solaris

1. Add entries for additional users (other than the administrator who runs the LiveCycle ES2 server) in the `/etc/sudoers` file. For example, if you are running LiveCycle ES2 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 users that you added via Add a user account to make connections to the LiveCycle ES2 server. For example, to allow a local user named `user1` the permission of making the connection to the LiveCycle ES2 server, use the following command:

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

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

3. Restart the server.

1.3.12.11 Multi-user support for PDF Generator ES2

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 and UAC disabled
Windows 2003 Server	Users with administrative privileges
AIX, Linux, and Solaris	Users with <code>sudo</code> privileges

When you add users for PDF Generator ES2 native conversions, you must grant the user running the application server with the *Replace a process level token* privilege. See [“Granting the Replace a process level token privilege \(Windows only\)” on page 35](#).

1.3.12.12 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 ES2.

► **To grant the *Replace a process level token* privilege:**

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.

1.3.13 Additional requirements for Connector for IBM Content Manager

LiveCycle ES2 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 ES2](#) document for your application server.

► **To 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.

► **To configure connections for multiple IBM Content Manager datastores:**

1. Configure the initial connection by following the steps in ["To configure the connection for a single IBM Content Manager datastore:" on page 36](#).
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 *[I4C 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 I4C.
 - Change the directory to *[I4C working directory]/cmgmt/connectors* where *[I4C 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.cmbsrvsadm -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.

► **To configure a multiuser connection to the IBM Content Manager datastore:**

1. From a command prompt window, change the directory to *[I4C 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 I4C.
3. Change the directory to *[I4C working directory]/cmgmt/connectors*, where *[I4C 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.

1.3.14 Additional requirements for LiveCycle ES2 Connector for IBM FileNet

These requirements are optional and required only if you are installing Connector for IBM FileNet.

IBM FileNet 4.0

If LiveCycle ES2 is connecting to IBM FileNet 4.0 Content Engine, you must install the Content Engine Java Client. Use the IBM FileNet 4.0 content server installer located by default in C:\Program Files\FileNet\Content Engine. 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 ES2. 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 ES2 is connecting to IBM FileNet 4.5 Content Engine, you must install the Content Engine client located by default in C:\Program Files\FileNet. 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 ES2 is connecting to IBM FileNet 5.0 Content Engine, you must install the Content Engine client located by default in C:\Program Files\FileNet. 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.

1.4 LDAP configuration

This configuration is optional and required only if you are using an LDAP directory to authenticate users. When you upgrade LiveCycle Policy Server 7.x or Rights Management ES LDAP configuration settings are automatically migrated.

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. For a list of supported LDAP servers, see ["Supported software" on page 22](#). Make note of the LDAP administrator name and password to use during the LiveCycle ES2 configuration process. Configure LiveCycle ES2 to connect with the LDAP database after you install and deploy your LiveCycle ES2 services. This configuration is done by using the User Manager service. See the [Installing and Deploying LiveCycle ES2](#) document for your application server.

1.5 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 ES2 product components. The lifetime of long-lived files is intended to span multiple restarts of a LiveCycle ES2 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 ES2 deployments. If some or all long-lived documents are lost or corrupted, the LiveCycle ES2 server may

become unstable. Input documents for asynchronous job invocation are also stored in the GDS directory and must be available in order to process requests.

You can also use the document storage in LiveCycle ES2 database. See [LiveCycle ES2 Administration Help](#) for details.

1.5.1 Planning and creating the global document storage directory

You should plan the location, size, and security aspects of your GDS directory in advance. See the following sections:

- ["Location of the global document storage directory" on page 39](#)
- ["Sizing factors for the global document storage directory" on page 39](#)
- ["Securing the global document storage directory" on page 40](#).
- ["Backing up the global document storage directory" on page 40](#).

You must create the GDS directory before you initialize the LiveCycle ES2 database.

1.5.2 Location of the global document storage directory

You configure the location of your GDS directory with LiveCycle Configuration Manager after you install LiveCycle ES2. 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 leave the location setting empty during installation, the location defaults to a directory under the application server installation:

- (JBoss) `[appserver root]/server/lc_<dbname>/svcnative/DocumentStorage`
- (WebLogic) `[appserverdomain]/adobe/<server>/DocumentStorage`
- (WebSphere) `[appserver root]/installedApps/adobe/<server>/DocumentStorage` or `[appserver root]/profiles/<profileName>/installedApps/adobe/<server>/DocumentStorage`

If you must change the GDS directory location after completing the installation (see [LiveCycle ES2 Administration Help](#)), you should plan an appropriate location for the GDS directory.

Caution: 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:\.

1.5.3 Sizing factors for the global document storage directory

The size of the shared directory depends on expected LiveCycle ES2 usage factors for the deployment. You should allocate a minimum of 10 GB of disk space for the GDS directory, but the following factors also affect the sizing:

- The typical volume of documents that LiveCycle ES2 processes. Processing high volumes of documents requires a larger GDS directory.
- The typical size of documents that LiveCycle ES2 processes. Processing large documents requires a larger shared GDS directory.

- The complexity of documents that LiveCycle ES2 processes. Processing complex documents (such as documents that are processed by multiple LiveCycle ES2 services) requires a larger GDS directory.

1.5.4 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 ES2 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: Incorrectly deleting files or directories from the GDS directory can render the LiveCycle ES2 installation inoperative.

1.5.5 Backing up the global document storage directory

The global document storage directory should be backed up to allow administrators to restore LiveCycle ES2 in case of failure.

If the global document storage directory becomes unavailable or is lost due to failure, LiveCycle ES2 will not run until the GDS directory and database are restored by a consistent back up or LiveCycle ES2 is reinitialized with a new installation.

If you use the LiveCycle ES2 database for document storage, backup of GDS happens along with the database backup. See [LiveCycle ES2 Administration Help](#) for details.

1.6 LiveCycle Business Activity Monitoring ES2 requirements

Note: Business Activity Monitoring ES2 (BAM) is not supported on 32-bit systems. Even though you can install BAM fileset with LiveCycle ES2 on a 32-bit system, you must deploy BAM on a 64-bit system.

1.6.1 Basic requirements

1.6.1.1 Dedicated JVM

Business Activity Monitoring ES2 requires a dedicated JVM. If you are deploying LiveCycle ES2 and Business Activity Monitoring ES2 on the same computer, be aware that Business Activity Monitoring ES2 must run on a dedicated JVM. This requires that Business Activity Monitoring ES2 is deployed separately from LiveCycle ES2 as follows:

- If deployed on JBoss, the two applications must be deployed on two completely separate JBoss implementations.
- If deployed on WebLogic or WebSphere, the two applications must be deployed on completely separate server definitions.

1.6.1.2 Client memory

The client computer that accesses Business Activity Monitoring ES2 should have a minimum of 512 MB RAM (1 GB is recommended).

1.6.1.3 Web browser

Business Activity Monitoring ES2 is tested for Microsoft Internet Explorer 6.0 (with patch 828750) or later browser for accessing BAM Workbench and BAM Dashboard. Firefox is not supported.

Note: You must also install Adobe Flash, version 9.0.115.0 or later.

Note: If your implementation of Business Activity Monitoring ES2 is running in an Asian language, you must configure the browser for the appropriate language support. See the documentation for Internet Explorer or Windows.

1.6.1.4 Mail Server

Business Activity Monitoring ES2 requires a running Simple Mail Transfer Protocol (SMTP) email server for delivering email notifications. The server is external to Business Activity Monitoring ES2 and is managed by your email system administrator. Contact that administrator to set up an account specifically for Business Activity Monitoring ES2. You need an account and password for sending mail, an address to use in the From address field, and the name of the email server host.

1.6.2 Creating the Business Activity Monitoring ES2 databases

There are two databases required to use the Business Activity Monitoring ES2 functionality: a metadata database and a geography database.

Create a Business Activity Monitoring ES2 metadata database to store the definitions of the process metrics that BAM Server monitors. It also stores the details of any alerts and object run-time data that are persisted to disk.

Because Business Activity Monitoring ES2 metadata can grow large, allocate at least 50 MB for the BAM Server metadata database. For production deployments, allocate at least 200 MB.

BAM Server can require specific settings for some aspects of the BAM Server metadata database configuration. The settings depend on the type of application server that is hosting BAM Server and the type of database server that is used to store the BAM Server metadata.

Also, create a Business Activity Monitoring ES2 geography database to support Geography Maps, which is required for using Geography charts in the dashboard. The size of the geography database depends on the number of geography code entries being accessed. 100 MB - 150 MB is the recommendation.

You need to create a user account that BAM Server can use to connect to the BAM Server metadata and geography databases. For database permissions, see [“Minimum database user permissions” on page 27](#). For all other database settings, use the configurations described for a LiveCycle ES2 database. See [“Creating the LiveCycle ES2 Database” on page 55](#) for your database type.

1.6.3 JBoss requirements for BAM

Metadata and geography databases:

Database	JDBC driver
Oracle	<ul style="list-style-type: none">Oracle thin driver (ojdbc6.jar) release 11.1.0.6 <p>Note: Use the Oracle 10g JDBC driver for both configurations.</p>
MS SQL Server	<ul style="list-style-type: none">sqljdbc.jar (version 1.2) <p>Note: Use SQL Server JDBC Driver 1.2 for both Microsoft SQL Server 2005 SP3 and Microsoft SQL Server 2008.</p> <p>Note: Ensure that the version 2000 drivers are not in the classpath.</p>
MySQL	<ul style="list-style-type: none">mysql-connector-java-5.1.6-bin.jar

Note: A Microsoft issue exists with the sqljdbc.jar driver, which is addressed in the Microsoft Knowledge Base [Article 917054](#).

1.6.4 WebLogic requirements for BAM

Metadata and geography databases:

Database	JDBC driver
IBM DB2	<ul style="list-style-type: none">db2jcc.jar (version 3.50.152)
MS SQL Server	<ul style="list-style-type: none">sqljdbc.jar (version 1.2) <p>Note: Use SQL Server JDBC Driver 1.2 for both Microsoft SQL Server 2005 SP3 and Microsoft SQL Server 2008.</p> <p>Note: Ensure that the version 2000 drivers are not in the classpath.</p>
MySQL	<ul style="list-style-type: none">mysql-connector-java-5.1.6-bin.jar
Oracle	<ul style="list-style-type: none">Oracle thin driver (ojdbc6.jar) release 11.1.0.6 <p>Note: Use the Oracle 10g JDBC driver for both configurations.</p>

1.6.5 WebSphere requirements for BAM

Metadata and geography databases:

Database	JDBC driver
IBM DB2	<ul style="list-style-type: none">db2jcc.jar (version 3.50.152)
MS SQL Server	<ul style="list-style-type: none">sqljdbc.jar (version 1.2) <p>Note: Use SQL Server JDBC Driver 1.2 for both Microsoft SQL Server 2005 SP3 and Microsoft SQL Server 2008.</p>
Oracle	<ul style="list-style-type: none">Oracle thin driver (ojdbc6.jar for JDK 1.6 or ojdbc5.jar for JDK 1.5) release 11.1.0.6 <p>Note: Use the Oracle 10g JDBC driver for both configurations.</p>

1.7 Installation considerations

1.7.1 Disabling Virus scans (Windows only)

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

1.7.2 Installing from network drives

It is recommended that you install LiveCycle ES2 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 ES2 installer to prevent the installation from proceeding.

1.7.3 Using LiveCycle ES2 with a Luna HSM cluster

When using a SafeNet Luna ethernet-attached Hardware Security Module (HSM) cluster, you must ensure HAOnly mode is enabled on the device.

► To enable HAOnly mode on the Luna device

1. Use the vtl tool shipped with the Luna client to determine if HAOnly mode is enabled. Type:

```
vtl haAdmin -HAOnly -show
```

2. 1) If HAOnly mode is not enabled, type:

```
vtl haAdmin -HAOnly -enable
```

1.7.4 Manual use of Acrobat restricted

If you installed the PDF Generator ES2 or PDF Generator 3D ES2 for native document conversion, use of the bundled Acrobat installation is restricted to the Generate PDF and Generate 3D PDF services and is not licensed for any other use.

1.7.5 Temporary directory

LiveCycle ES2 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 ES2 server by specifying an appropriate value for this setting, see this [blog](#).

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

Note: If you do not create the temporary directory, the default system-configured location is used.

1.7.6 LiveCycle ES2 IPv6 support

LiveCycle ES2 includes IPv6 support. The default configurations defined in the installation documentation for LiveCycle ES2 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 ES2. 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 ES2 installation using LiveCycle Configuration Manager. See “Enabling CIFS in IPv6 mode” in the [Installing and Deploying LiveCycle ES2](#) guide for your application server.

1.7.6.1 Supported IPv6 configurations

Not all infrastructure components support IPv6. For example, MySQL and Oracle databases do not support IPv6. However, you can still 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
- Databases that do not support IPv6: Oracle 10g, 11g R2; MySQL 5.1
- Application servers that support IPv6: JBoss 4.2.0, 4.2.1, JBoss EAP 4.3; WebLogic 10g R3 and 11g R1; WebSphere 7.0.0.9.

1.7.6.2 IPv6 implementation guidelines

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

- After installing LiveCycle ES2, do not use the option to start the LiveCycle Configuration Manager directly from the LiveCycle ES2 installer. Instead, navigate to the `[LiveCycle ES2 root]\configurationManager\bin\IPv6` directory, and run the IPv6-specific script (`ConfigurationManager_IPv6.bat` or `ConfigurationManager_IPv6.sh`) to launch the LiveCycle Configuration Manager.

- If you have chosen to validate the application server configuration using the LiveCycle 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.
- *(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 LiveCycle 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 ES2 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.
- *(Cluster installation only)* If you are installing LiveCycle ES2 on to a server cluster, you must map the numeric IPv6 addresses of each cluster node to the computer's host name in DNS or in the `hosts` file on each cluster node. The `hosts` file is located at:
 - Solaris: `/etc/inet/ipnodes`
 - Windows: `C:\Windows\system32\drivers\etc\hosts`
 - Linux: `/etc/hosts`
- *(JBoss turnkey installation and JBoss for BAM)* Edit the following files:
 - `[LiveCycle ES2 root]\jboss\bin\run.bat`: Change `-Djava.net.preferIPv4Stack=true` to `-Djava.net.preferIPv6Stack=true`
 - `[LiveCycle ES2 root]\jboss\bin\service.bat`: Replace `-b 0.0.0.0` with `-b <hostname resolved to 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.
- 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.

1.7.6.3 Configuring IPv6 for JBoss

1. Unzip the bundled JBoss.
2. Modify `adobe-ds.xml` and the database-specific data source configuration file to connect to the LiveCycle ES2 database.
3. Modify the `login-config.xml` file to connect to the LiveCycle ES2 database.
4. Modify the following files to enable IPv6:

- `[LiveCycleES2 root]\jboss\bin\run.bat`: Do the following in this file:
 - Change `-Djava.net.preferIPv4Stack=true` to `-Djava.net.preferIPv6Stack=true`
 - Add the `-Djava.net.preferIPv6Addresses=true` argument.
 - `[LiveCycleES2 root]\jboss\bin\service.bat`: Replace `- b 0.0.0.0` with `-b <hostname mapped to IPv6 address>`
5. Launch LiveCycle Configuration Manager by invoking the `[LiveCycleES2 root]\configurationManager\bin\IPv6\ ConfigurationManager_IPv6.bat` or `ConfigurationManager_IPv6.sh` script.
 6. In the LiveCycle Configuration Manager, select the steps to configure EAR files, bootstrap and deploy LiveCycle ES2 modules.
 7. After the LiveCycle Configuration Manager process is completed, copy these EAR files to the `[LiveCycleES2 root]\jboss\server\all\deploy` directory.
 8. Start JBoss from a command line.
- Note:** If you are running JBoss on Linux, you must edit the `run.sh` file to use IPv6.
9. Provide LiveCycle Configuration Manager hostname of the computer that is mapped to its IPv6 address and then bootstrap the application server to deploy the LiveCycle ES2 modules.

1.7.6.4 Configuring IPv6 for WebLogic

1. Install LiveCycle ES2 using the installer.
 2. Do not launch the LiveCycle Configuration Manager when the installer finishes. Launch LiveCycle Configuration Manager by invoking the `[LiveCycleES2 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 LiveCycle Configuration Manager.
- Note:** You will get an error message that the LiveCycle 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.
4. 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.
 5. 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.
 6. Save the changes, and then restart the managed server.

7. Launch LiveCycle Configuration Manager by invoking the `[LiveCycleES2 root]\configurationManager\bin\IPv6\ConfigurationManager_IPv6.bat` or `ConfigurationManager_IPv6.sh` script.
8. In the LiveCycle Configuration Manager, select the steps to deploy EAR files, bootstrap, deploy LiveCycle ES2 modules.
9. Enter the same hostname as provided in the listen address field of the managed server.

Note: Even if you are running LiveCycle Configuration Manager on the same computer, you must provide the listen address of the managed server for bootstrapping and deploying LiveCycle ES2 modules.

1.7.6.5 Configuring IPv6 for WebSphere

1. Install LiveCycle ES2 using the installer script. After the installation is complete, do not start the LiveCycle Configuration Manager when the installer prompts.
2. Navigate to the `[LiveCycleES2 root]\configurationManager\bin\IPv6` directory, and run the IPv6-specific script (`ConfigurationManager_IPv6.bat` or `ConfigurationManager_IPv6.sh`) to launch LCM.
3. Use the LiveCycle Configuration Manager options to configure EAR and the application server.
4. Follow the steps in the LiveCycle 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 LiveCycle Configuration Manager. Ignore the warning if data source validation fails. You can validate data sources from WebSphere Administrative Console.
6. On the WebSphere Administrative Console, navigate to the page where `JAVA_OPTIONS` are specified and do the following tasks:
 - Change `-Djava.net.preferIPv4Stack=true` to `-Djava.net.preferIPv6Stack=true`.
 - Add the `-Djava.net.preferIPv6Addresses=true` argument.
7. Manually deploy the EAR files to WebSphere Application Server by using WebSphere Administrative Console. Configured EAR files are available at `[LiveCycleES2 root]/configurationManager/export` folder.
8. Restart WebSphere Application Server.
9. Navigate to the `[LiveCycleES2 root]\configurationManager\bin\IPv6` directory, and run `ConfigurationManager_IPv6.bat` or `ConfigurationManager_IPv6.sh` to launch LiveCycle Configuration Manager.
10. In the LiveCycle Configuration Manager, select the options to bootstrap and deploy LiveCycle ES2 modules. Provide the appserver give 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.

1.8 Server configuration for enabling CIFS

To enable file access to Content Services ES2 as a network folder for Windows clients of LiveCycle ES2, you need to enable Common Internet File System (CIFS) in Content Services ES2. Perform the following steps before you can enable CIFS:

- [“Creating a virtual interface \(AIX, Linux, and Solaris only\)” on page 48](#)
- [“Configuring Windows Server for CIFS” on page 48](#)

Note: If you are enabling CIFS in an IPv6 environment, you must explicitly enable IPv6 configuration after you configure your LiveCycle ES2 installation using LiveCycle Configuration Manager. See “Enabling CIFS in IPv6 mode” in the [Installing and Deploying LiveCycle ES2](#) guide for your application server.

1.8.1 Creating a virtual interface (AIX, Linux, and Solaris only)

If you want to enable CIFS on LiveCycle ES2 deployed on AIX, Linux, or Solaris platform, you must create a virtual interface and assign it an IP address on the server that runs LiveCycle ES2. This is required because the CIFS protocol uses the same ports that are used by Samba service on AIX, Linux, or Solaris computers. You specify this virtual IP address as the alternate IP address when you enable CIFS using the LiveCycle Configuration Manager.

If your LiveCycle ES2 implementation is on IPv6, you should create virtual IPv6 interfaces on the CIFS server. The interfaces that you create should be within the same network as the CIFS clients. See the documentation related to your operating system for more information.

Note: You should add the virtual interface that you create to the `/etc/hosts` file to make them persist across reboots.

1.8.2 Configuring Windows Server for CIFS

You will need to manually configure the Windows Server 2003 and 2008 computers that host LiveCycle ES2 to enable CIFS on them. When CIFS is enabled, users can access the Content Services ES2 repository as a network folder and perform various file operations on their local file system. In LiveCycle Content Services ES2, CIFS is supported for enterprise domain users with ActiveDirectory as their directory provider.

Note: Ensure that the CIFS server has a static IP address for Java-based CIFS implementation. For Windows native implementation, a static IP address is not required.

On Windows computers, you need to do the following:

- [“Enable NetBIOS over TCP/IP” on page 48](#)
- [“Add additional IP address” on page 49](#)
- [“Disable SMB over NetBIOS registry \(Windows 2003 only\)” on page 49](#)
- [“Disable File and Printer Sharing \(Windows 2008 only\)” on page 49](#)

1.8.2.1 Enable NetBIOS over TCP/IP

You need to enable NetBIOS over TCP/IP so that clients connecting to the LiveCycle ES2 server can have their requests resolved for the server host name.

1. In the **Local Area Connection Properties** dialog box, on the **General** tab, select **Internet Protocol**, and then click **Properties**.
2. In the **General** tab of the **Internet Protocol (TCP/IP) Properties** dialog box, ensure that the server has a static IP address. Click **Advanced**.
3. In the **Advanced TCP/IP Settings** dialog box, select the **WINS** tab and select **Enable NetBIOS over TCP/IP**.

1.8.2.2 Add additional IP address

Note: This step is required only for Java-based CIFS implementation.

1. In the **Local Area Connection Properties** dialog box, on the **General** tab, select **Internet Protocol**, and then click **Properties**.
2. In the **General** tab of the **Internet Protocol (TCP/IP) Properties** dialog box, ensure that the server has a static IP address. Click **Advanced**.
3. In the **Advanced TCP/IP Settings** dialog box, select the **IP Settings** tab and click **Add**.
4. Specify a static IP address and click **Add**.

1.8.2.3 Disable SMB over NetBIOS registry (Windows 2003 only)

You must disable SMB over NetBIOS by editing the Windows registry.

1. In the Windows Registry Editor, navigate to **HKEY_LOCAL_MACHINE > SYSTEM > CurrentControlSet > Services > NetBT > Parameters**.
2. Set the DWORD **SMBDeviceEnabled** to 0. If it is not present, add a new DWORD value with name `SMBDeviceEnabled` and set it to 0.

1.8.2.4 Disable File and Printer Sharing (Windows 2008 only)

Caution: After you disable file and printer sharing with this step, you will not be able to access the server directly using Windows file sharing protocol. This computer will not display in the Windows network view.

- Go to Network Settings and deselect **File and Printer Sharing for Microsoft Clients** and click **Apply**.

1.9 Processes with document form variables and digital signatures

If you are upgrading from a previous version of LiveCycle and changing your LiveCycle ES2 server, you may disrupt any processes that use the document form variable or digital signatures. This is because these forms are rendered only once, setting the submit URL. Changing the server breaks the certificate.

Choose the solution that is most appropriate for your LiveCycle ES2 environment from the following solutions:

Solution 1: Complete all processes that use a form document variable before you upgrade or move to the remote server. Use this method if you maintain legacy LiveCycle servers after the upgrade. This

approach also eliminates the need for *throw-away* work to be done to manage the redirection of the form submissions. This method is not practical if you have many outstanding processes.

Solution 2: If the server being upgraded is not being decommissioned, a reverse proxy approach is preferable. With this method, you maintain the reverse proxy on the old system until all the migrated processes are completed.

Solution 3: You can use the Apache `mod_rewrite` module to modify the embedded URLs in each form as they are delivered to the client.

Note: If your LiveCycle implementation is on IPv6, clients using EJB invocation for PDF creation report exceptions. This is a known issue attributed to Sun JDK 6. See http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6230761 for details.

1.10 LiveCycle ES2 Central Migration Bridge

The LiveCycle ES2 Central Migration Bridge service helps you to migrate existing applications from your Adobe Central Pro Output Server or Web Output Pak products to work with the LiveCycle ES2 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 ES2 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 Adobe Central Pro Output Server 5.7 or an executed Adobe 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 ES2, you will need to acquire a PAC for that operating system. Contact your Adobe representative for more information regarding migration, executing a migration agreement, 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 ES2, but it is not a prerequisite to installing LiveCycle ES2 (that is, it can be installed before or after LiveCycle ES2). Consult the Central Pro documentation set for installation instructions.

Caution: 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.

Note: The LiveCycle ES2 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 ES2 turnkey installations

If you are installing and configuring a LiveCycle ES2 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 ES2 custom installations

If you are installing and configuring LiveCycle ES2 using the Custom mode (Partial turnkey or Manual), LiveCycle 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 LiveCycle Administration Console.

When you have finished installing LiveCycle ES2, if Central Pro is not installed in the default location, do the following steps to point LiveCycle ES2 to the correct directory:

1. Log in to LiveCycle 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 ES2. (See [Creating LiveCycle ES2 Processes](#) or [Creating LiveCycle ES2.5 Processes](#), as applicable.)

This chapter describes how to do the following tasks:

- Obtain the Adobe LiveCycle Reader Extensions ES2 Rights credential
- Obtain digital certificates for use with Adobe LiveCycle Digital Signatures ES2

2.1 Obtaining the Reader Extensions ES2 Rights credential

The LiveCycle Reader Extensions ES2 Rights credential is a digital certificate that is specific to Reader Extensions ES2 that enables Adobe Reader usage rights to be activated in PDF documents. If the credential is not installed, Reader Extensions ES2 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 ES2 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 ES2. 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 ES2, 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 ES2, 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 ES2 through a partner or software provider who integrated Reader Extensions ES2 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).

2.2 Obtaining digital certificates for use with Digital Signatures ES2

Digital certificates are required for use with Digital Signatures ES2. Although you can configure and manage digital certificates after you install and configure LiveCycle ES2, obtaining them before you install ensures that you are ready to use LiveCycle ES2 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 ES2 by using Trust Store Management, which is accessible through the web-based LiveCycle Administration Console. (See [LiveCycle Administration Console 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 ES2 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.

3

Creating the LiveCycle ES2 Database

This section describes how to set up the database for use with LiveCycle ES2.

The database will contain these elements:

- LiveCycle ES2 services
- LiveCycle ES2 run-time configurations
- LiveCycle ES2 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. (See [“System Requirements” on page 10.](#))

If this is the first installation of LiveCycle ES2, you must create an empty database. All the tables required to support LiveCycle ES2 are created when you initialize the LiveCycle ES2 database using LiveCycle Configuration Manager. (See the [Installing and Deploying LiveCycle ES2](#) document for your application server.)

If you install Business Activity Monitoring, you must create metadata database and geography databases. For information about creating the LiveCycle ES2 Business Activity Monitoring ES2 (BAM) metadata and geography databases for LiveCycle Process Management ES2, see the [Installing and Deploying LiveCycle ES2](#) document for your application server.

3.1 Database configuration requirements

This section describes special tables, indexes, and other properties that are required in the LiveCycle ES2 database that are not configured by LiveCycle Configuration Manager.

3.2 Creating an Oracle database

If you prefer not to use the default database that was created when you installed Oracle 10g or Oracle 11g R2, create a new database by using the Database Configuration Assistant tool. LiveCycle ES2 supports Oracle 10g and Oracle 11g R2 on WebLogic, WebSphere, and JBoss application servers.

If you install Business Activity Monitoring ES2, you must first create the metadata and geography databases. See [“Creating the Business Activity Monitoring ES2 databases” on page 41.](#)

Note: You can use the Transaction Processing or General Purpose templates while configuring an Oracle database instance for LiveCycle ES2. 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. Increase this initial size if you are deploying LiveCycle Content Services ES2.

- Create user quotas to allow the database to grow to accommodate persistent data from LiveCycle ES2 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.
- If you are deploying Content Services ES2, set the block size of your Oracle database to 16k. Modify the `db_block_size` value in the `initSID.ora` file located in your Oracle install directory.

Note: This value depends on your operating system. For Windows platforms, the range is 2k-16k.

3.2.1 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

Caution: (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 10g or Oracle 11g R2, see the appropriate user documentation.

3.3 Creating a SQL Server database

You can create a SQL Server database that LiveCycle ES2 will use to store run-time and configuration data. For information about creating a SQL Server database, refer to the SQL Server documentation. LiveCycle ES2 supports SQL Server 2005 SP3 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

3.3.1 Installing the SQL Server database driver

Note: If you have not done so already, download the SQL Server JDBC Driver 1.2 (for all platforms) from the [Microsoft Download Center](#) location on the Microsoft website. Ensure that you install the SQL Server database driver on the server where you plan to install LiveCycle ES2.

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 1.2 for both Microsoft SQL Server 2005 SP3 and Microsoft SQL Server 2008.

3.3.2 Setting up SQL Server for LiveCycle ES2

Before you create the LiveCycle ES2 database, optimize SQL Server by changing these settings.

3.3.2.1 Increasing 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.

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

1. Using Microsoft SQL Server Management Studio, connect to the database server where you will host the LiveCycle ES2 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.

3.3.2.2 Setting processor priority

On dedicated database servers, which are recommended for production installations of LiveCycle ES2, 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.

► **To set the processor priority:**

1. Using Microsoft SQL Server Management Studio, connect to the database server where you will host the LiveCycle ES2 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.

3.3.2.3 Increasing the recovery interval

This setting specifies the amount of time the deployment waits for recovery after a crash. The SQL Server default setting is 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.

► **To increase the recovery interval:**

1. Using Microsoft SQL Server Management Studio, connect to the database server where you will host the LiveCycle ES2 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.

3.3.2.4 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” on page 117](#).
- For WebLogic, see the *Configuring SQL Server database connectivity* section in [Installing and Deploying LiveCycle ES2 for WebLogic](#).
- For WebSphere, see the *Configuring the SQL Server data source* section in [Installing and Deploying LiveCycle ES2 for WebSphere](#).

Note: Windows Authentication Mode is not supported for partial turnkey installation. LiveCycle Configuration Manager fails to validate the database connection. Therefore, for partial turnkey installation, the authentication type must be set to SQL Server.

3.3.3 Sizing your SQL Server database

The default database sizes that SQL Server provides are too small for LiveCycle ES2. 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 ES2 database to ensure that it does not lose its restricted space or the space available on the disks where the database resides.

3.3.4 Creating the LiveCycle ES2 database user, schema, and login

You are now ready to create the LiveCycle ES2 database user, schema, and login.

Caution: 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 ES2. Any other collation may cause your database initialization to fail. The collation for your LiveCycle ES2 database instance can be different from the collation used when creating the SQL Server database.

► To create the LiveCycle ES2 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 ["Sizing your SQL Server database" on page 58](#).
5. In the **Database Data Autogrowth** box, enter 50%.
6. In the **Database Log Initial Size** box, enter the appropriate value:
 - For small development or small production systems, specify 20 MB.
 - For larger systems, see ["Sizing your SQL Server database" on page 58](#).
7. In the **Database Log Autogrowth** box, enter 50%.
8. Click **OK** to create the database.

3.3.4.1 Creating the LiveCycle ES2 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.

► **To create the LiveCycle ES2 user:**

1. Using Microsoft SQL Server Management Studio, connect to the database server where you created the LiveCycle ES2 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**.

3.3.4.2 Associating the LiveCycle ES2 user with the database

After you create the LiveCycle ES2 user, associate it with the LiveCycle ES2 database.

► **To associate the LiveCycle ES2 user with the 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**.

3.3.4.3 Setting the isolation level for the LiveCycle ES2 database

LiveCycle ES2 requires a specific isolation level to manage deadlocking when long-running transactions occur at the same time as numerous shorter reads.

► **To set the isolation level:**

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

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

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

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

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

3.4 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 ES2 supports DB2 with WebSphere 6.1, WebSphere 7.0, and WebLogic.

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

3.4.1 DB2 user account

LiveCycle ES2 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.

► **To create a DB2 database:**

1. **(Content Services ES2 not included)** 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.

DB2 9.1

```
create database dbname using codeset utf-8 territory default;
connect to dbname;
CREATE BUFFERPOOL BP8K SIZE 50000 PAGESIZE 8192 NOT EXTENDED STORAGE;
connect reset;
connect to dbname;
CREATE TEMPORARY TABLESPACE DBNAME_TEMP_8K IN DATABASE PARTITION GROUP
IBMTEMPGROUP PAGESIZE 8192 MANAGED BY SYSTEM USING
('DB2_root\DBNAME_TEMP') EXTENTSIZE 32 PREFETCHSIZE 16 BUFFERPOOL BP8K;
CREATE REGULAR TABLESPACE DBNAME_DATA_8K IN DATABASE PARTITION GROUP
IBMDEFAULTGROUP PAGESIZE 8192 MANAGED BY DATABASE USING
(FILE'DB2_root\DBNAME_DATA'9000) EXTENTSIZE 16 PREFETCHSIZE 16
BUFFERPOOL BP8K;
commit work;
connect reset;
connect to dbname;
alter bufferpool ibmdefaultbp immediate size 96000;
alter bufferpool bp8k immediate size 32000;
commit work;
connect reset;
update db cfg for dbname using dbheap 4000;
update db cfg for dbname using logbufsz 2048;
update db cfg for dbname using locklist 2000;
update db cfg for dbname using chngpgs_thresh 40;
update db cfg for dbname using logfilsiz 4000;
deactivate database dbname;
activate database dbname;
```

DB2 9.5

```
create database dbname using codeset utf-8 territory default;
connect to dbname;
CREATE BUFFERPOOL "BP8K" SIZE 50000 PAGESIZE 8192;
connect reset;
connect to dbname;
CREATE TEMPORARY TABLESPACE DBNAME_TEMP_8K IN DATABASE PARTITION GROUP
IBMTEMPGROUP PAGESIZE 8192 MANAGED BY SYSTEM USING
('DB2_root\DBNAME_TEMP') EXTENTSIZE 32 PREFETCHSIZE 16 BUFFERPOOL BP8K;
CREATE REGULAR TABLESPACE DBNAME_DATA_8K IN DATABASE PARTITION GROUP
IBMDEFAULTGROUP PAGESIZE 8192 MANAGED BY DATABASE USING
(FILE'DB2_root\DBNAME_DATA'9000) EXTENTSIZE 16 PREFETCHSIZE 16 BUFFERPOOL
BP8K;
commit work;
connect reset;
connect to dbname;
alter bufferpool ibmdefaultbp immediate size 96000;
alter bufferpool bp8k immediate size 32000;
commit work;
connect reset;
update db cfg for dbname using dbheap 4000;
update db cfg for dbname using logbufsz 2048;
update db cfg for dbname using locklist 2000;
update db cfg for dbname using chngpgs_thresh 40;
update db cfg for dbname using logfilsiz 4000;
deactivate database dbname;
```

```
activate database dbname;
```

2. **(Content Services ES2 included)** If you are using Content Services ES2 with any of the other LiveCycle ES2 modules, use 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.

DB2 9.1

```
create database dbname using codeset utf-8 territory default;
connect to dbname;
CREATE BUFFERPOOL BP8K SIZE 50000 PAGESIZE 8192 NOT EXTENDED STORAGE;
CREATE BUFFERPOOL BP32K SIZE 4000 PAGESIZE 32768 NOT EXTENDED STORAGE;
connect reset;
connect to dbname;
CREATE TEMPORARY TABLESPACE DBNAME_TEMP_8K IN DATABASE PARTITION GROUP
IBMTEMPGROUP PAGESIZE 8192 MANAGED BY SYSTEM USING
('DB2_root\DBNAME_TEMP')
EXTENTSIZE 32 PREFETCHSIZE 16 BUFFERPOOL BP8K;
CREATE REGULAR TABLESPACE DBNAME_DATA_8K IN DATABASE PARTITION GROUP
IBMDEFAULTGROUP PAGESIZE 8192 MANAGED BY DATABASE USING
(FILE'DB2_root\DBNAME_DATA'64000) EXTENTSIZE 16 PREFETCHSIZE 16
BUFFERPOOL BP8K;
commit work;
connect reset;
connect to dbname;
CREATE TEMPORARY TABLESPACE DBNAME_TEMP_32K IN DATABASE PARTITION GROUP
IBMTEMPGROUP PAGESIZE 32768 MANAGED BY SYSTEM USING
('DB2_root\DBNAME32k_TEMP') EXTENTSIZE 32 PREFETCHSIZE 16 BUFFERPOOL
BP32K;
CREATE REGULAR TABLESPACE DBNAME_DATA_32K IN DATABASE PARTITION GROUP
IBMDEFAULTGROUP PAGESIZE 32768 MANAGED BY DATABASE USING
(FILE'DB2_root\DBNAME32k_DATA'9000) EXTENTSIZE 16
PREFETCHSIZE 16 BUFFERPOOL BP32K;
alter bufferpool ibmdefaultbp immediate size 96000;
alter bufferpool bp8k immediate size 32000;
commit work;
connect reset;
update db cfg for dbname using dbheap 4000;
update db cfg for dbname using logbufsz 2048;
update db cfg for dbname using locklist 2000;
update db cfg for dbname using chngpgs_thresh 40;
update db cfg for dbname using logfilsiz 4000;
deactivate database dbname;
activate database dbname;
```

DB2 9.5

```
create database dbname using codeset utf-8 territory default;
connect to dbname;
CREATE BUFFERPOOL BP8K SIZE 50000 PAGESIZE 8192;
CREATE BUFFERPOOL BP32K SIZE 500 PAGESIZE 32768;
CREATE TEMPORARY TABLESPACE dbname_TEMP_8K IN DATABASE PARTITION GROUP
IBMTEMPGROUP PAGESIZE 8192 MANAGED BY SYSTEM USING
('DB2_root\dbname8_TEMP') EXTENTSIZE 32 PREFETCHSIZE 16 BUFFERPOOL BP8K;
```

```
CREATE REGULAR TABLESPACE dbname_DATA_8K IN DATABASE PARTITION GROUP
IBMDEFAULTGROUP PAGESIZE 8192 MANAGED BY DATABASE USING
(FILE'DB2_root\dbname8_DATA'9000) EXTENTSIZE 16 PREFETCHSIZE 16
BUFFERPOOL BP8K;
CREATE TEMPORARY TABLESPACE dbname_TEMP_32K IN DATABASE PARTITION GROUP
IBMTMPGROUP PAGESIZE 32768 MANAGED BY SYSTEM USING
('DB2_root\dbname32_TEMP') EXTENTSIZE 32 PREFETCHSIZE 16 BUFFERPOOL
BP32K;
CREATE REGULAR TABLESPACE dbname_DATA_32K IN DATABASE PARTITION GROUP
IBMDEFAULTGROUP PAGESIZE 32768 MANAGED BY DATABASE USING
(FILE'DB2_root\dbname32_DATA'9000) EXTENTSIZE 16 PREFETCHSIZE 16
BUFFERPOOL BP32K;
alter bufferpool ibmdefaultbp immediate size 96000;
alter bufferpool BP8K immediate size 32000;
commit work;
connect reset;
update db cfg for dbname using dbheap 4000;
update db cfg for dbname using logbufsz 2048;
update db cfg for dbname using locklist 2000;
update db cfg for dbname using chngpgs_thresh 40;
update db cfg for dbname using logfilsiz 4000;
deactivate database dbname;
activate database dbname;
```

3. Make the following changes to the script:

- Replace the instances of *dbname* and *DBNAME* with the name you want for the LiveCycle ES2 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.

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

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

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

The DB2 database must be configured for concurrent use.

► **To 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 ES2 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 LiveCycle Configuration Manager.
Isolation Level	Cursor Stability

5. Click **Finish**.

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

1. Log in to the IBM Control Center.
2. Click *[database_name]* > **User and Group Objects** > **DB Users**.
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**.

3.4.2 Additional DB2 requirements for LiveCycle Content Services ES2

The script described in step 2 in [“To create a DB2 database:” on page 61](#) creates an additional DB2 database page size and sets it to 32 KB. If LiveCycle Content Services ES2 is not being deployed, the default DB2 database page size of 8 KB is acceptable.

3.4.2.1 Configuring multiple DB2 schemas

When configuring Content Services ES2 to run using multiple schemas on a single DB2 database instance, Content Services ES2 deployment will succeed on the first node but fail on all subsequent nodes. DB2 is case-sensitive and expects to receive values in uppercase letters. To avoid this problem, you must add the following JVM argument to your application server:

```
-Dhibernate.default_schema=<schema_name>
```

Note: <schema_name> must be replaced by the schema name in uppercase letters.

3.5 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 ES2 for JBoss Using Turnkey](#).

Use the MySQL tools to create a MySQL 5 database for use with LiveCycle ES2 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 ES2 supports MySQL 5 with JBoss 4.2.0, JBoss 4.2.1, WebLogic 10g R3 and WebLogic 11g R1.

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

3.5.1 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 ES2 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.

3.5.2 MySQL user account

The MySQL user account that you create must have these privileges to access the tables in the LiveCycle ES2 database:

- SELECT
- INSERT

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

3.5.3 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`

4

Configuring a JBoss Application Server

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

- Turnkey with JBoss 4.2.1 (Windows only)
- Adobe preconfigured JBoss with JBoss 4.2.1
- Manual JBoss configuration with JBoss 4.2.0 or JBoss 4.2.1 (See [“Appendix - Manually Configuring JBoss” on page 99.](#))

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 ES2 runs on JBoss on Windows Server 2003 SP2 and R2 SP2 (Enterprise Edition or Standard Edition), Windows Server 2008, Red Hat Enterprise Linux 5.0, Red Hat Enterprise Linux Advanced Platform 5.0, Solaris 10, and SUSE Enterprise Linux 10 platforms. (See [“Third-party infrastructure support” on page 10.](#))

Note: If you are planning to use the LiveCycle ES2 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.](#)

This chapter uses the following conventions.

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

4.1 Adobe-preconfigured JBoss

The Adobe-preconfigured JBoss software is delivered as part of the LiveCycle ES2 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 99.](#)

Adobe downloads the standard JBoss 4.2.1 archive from the JBoss website and configures it with the options required to run LiveCycle ES2 in a stand-alone configuration. You can review these configurations in the next section.

Note: JBoss 4.2.1 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 `JBoss_DVD.zip` (Windows) or `JBoss_DVD._unix.tar.gz` (Linux or Solaris) file to your file system. After the main archive file is extracted, extract the following archived directories: `\additional`, `\livecycle_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) download in the `/third_party` directory. This folder contains two JBoss installers:

- The `jboss.zip` file contains the Adobe pre-configured JBoss 4.2.1. Use this JBoss version for a LiveCycle ES2 server installation.
- The `jboss_4.2.1_bam.zip` file contains the Adobe pre-configured JBoss 4.2.1 for Business Activity Monitoring ES2.

Depending on your installation requirements, copy the JBoss directories and their contents to the location where you intend to install JBoss. Both the JBoss installations have all the configuration steps completed except for the steps detailed in [“Configuring the LiveCycle ES2 database connectivity for Adobe pre-configured JBoss 4.2.1” on page 71](#), 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: If you prefer to use JBoss 4.2.0, you must download the JBoss 4.2.0 installer from JBoss download site, and manually configure it. Adobe supplies only JBoss 4.2.1 preconfigured with the LiveCycle ES2 installation media.

4.2 Installing the Sun Java JDK for JBoss

For JBoss 4.2.0 and JBoss 4.2.1, you must download and install the following Sun JDK versions from www.java.sun.com:

- (JBoss 4.2.0) Sun JDK 5.0 update 11 or later updates to 5.0
- (JBoss 4.2.1 on all platforms) Sun JDK 6.0 update 14 or later updates to 6.0

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

► To 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_14
```

➤ **To 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;
```

➤ **To set the JAVA_HOME environment variable (Linux or Solaris):**

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

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

➤ **To 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
```

➤ **(Windows, Linux, or Solaris) To 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_14).

4.3 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 user's login scripts for the user that the JBoss process will run as.

4.4 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`. For JBoss obtained directly from Red Hat, either use one of the JBoss provided start configurations (`all`, `default` or `minimal`) or a custom configuration.

✦ To 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 [config] -b [server_IP_Address]`
- (Linux and Solaris) `./run.sh -c [config] -b [server_IP_Address]`

where `[config]` is the configuration required for your database and `[server_IP_Address]` is the actual 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]
```

➤ To 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>`

4.5 Notes about configuring the JBoss Application Server

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.

4.6 Configuring the LiveCycle ES2 database connectivity for Adobe pre-configured JBoss 4.2.1

To configure a connection to the LiveCycle ES2 database, you must complete the following tasks:

- Configure the LiveCycle ES2 data source.
- Configure JBoss to use your database as the default data source.

You must install database drivers to the installation directories of the application server. Drivers are required to enable LiveCycle Configuration Manager and the application server to connect to the LiveCycle ES2 database. You must install the drivers for the type of database that you use.

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.

4.6.1 Configuring MySQL for Adobe-preconfigured JBoss 4.2.1

To enable Adobe-preconfigured JBoss to connect to the MySQL database that stores LiveCycle ES2 data, you must create a data source file and deploy it to the instance of JBoss where you will deploy LiveCycle ES2.

If you are using a manually installed JBoss, see [“Configuring MySQL for manually installed JBoss” on page 108](#).

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_clear.html.

4.6.1.1 Edit `adobe-ds.xml` file

Before configuring the MySQL data source, you must have already created the database on MySQL. (See [“Creating a MySQL database” on page 66](#).)

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>
<user-name>adobe</user-name>
<password>adobe</password>
```

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 LiveCycle ES2 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 ES2 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 the file.

4.6.1.2 Edit mysql-ds.xml file

If you are running LiveCycle ES2 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>
<user-name>adobe</user-name>
<password>adobe</password>
```

- **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 ES2.
2. 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.
 3. Save and close the file.

4.6.1.3 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.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=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.

4.6.2 Configuring Oracle for Adobe-preconfigured JBoss 4.2.1

To enable JBoss to connect to the Oracle database that stores LiveCycle ES2 data, you must create a data source file and deploy it to the instance of JBoss where you will deploy LiveCycle ES2.

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_clear.html.

4.6.2.1 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>  
<user-name>adobe</user-name>  
<password>adobe</password>
```

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 ES2 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: You must also remove the commenting around the `<no-tx-datasource>` tag and provide the values for the BAM database hostname, username, and password.

Note: If your LiveCycle ES2 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 the file.

4.6.2.2 Edit oracle-ds.xml file

If you are running LiveCycle ES2 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>  
<user-name>adobe</user-name>  
<password>adobe</password>
```

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.

Replace the default user name and password with the user name and password that the application server uses these credentials to access the database.

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 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 oracle-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. Save and close the file.

4.6.2.3 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.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 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.

4.6.3 Configuring SQL Server for Adobe-preconfigured JBoss 4.2.1

To enable JBoss to connect to the SQL Server database that stores LiveCycle ES2 data, you must create a SQL Server data source file and deploy it to the instance of JBoss where you will deploy LiveCycle ES2 (for example, `[appserver root]/server/lc_sqlserver/deploy`).

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

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_clearart.html.

4.6.3.1 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>
<user-name>adobe</user-name>
<password>adobe</password>
```

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 ES2 data. You will need to update the default value, `adobe`, with your database name.
 - **username and password:** The user name and password that the application server uses to access the database. You will need to update the default values, `adobe` and `adobe`, with the credentials for your database.

3. Ensure that the `<driver-class>` is set as follows:

```
<driver-class>com.microsoft.sqlserver.jdbc.SQLServerDriver</driver-class>
```

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. 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 ES2 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 the file as `adobe-ds.xml` in the `[appserver root]/server/lc_sqlserver/deploy` directory.

4.6.3.2 Edit mssql-ds.xml file

If you are running LiveCycle ES2 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>
<user-name>adobe</user-name>
<password>adobe</password>
```

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=<LiveCycleES2 database></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.

4.6.3.3 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.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 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.

4.6.3.4 Map the Windows login to the LiveCycle ES2 database user

1. Using Microsoft SQL Server Management Studio, connect to the database server that hosts the LiveCycle ES2 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 ES2 database in the tree and open Security > Users. The new user appears in the list of users.

4.6.3.5 Configure integrated security on Windows

Note: You must run the JBoss service as the Windows user configured in [“Map the Windows login to the LiveCycle ES2 database user” on page 79](#). 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 file, located in `[appserver root]\server\lc_sqlserver\deploy`, to add `integratedSecurity=true` to the connection URL as shown in this example:

```
jdbc:sqlserver://<serverhost>:<port>;databaseName=<dbname>;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 with the Microsoft SQL JDBC 1.2 driver installation. The default location is `<InstallDir>/sqljdbc_1.2/enu/auth/x86` for 32-bit operating systems and `<InstallDir>/sqljdbc_1.2/enu/auth/x64` for 64-bit operating systems).

Note: Use SQL Server JDBC Driver 1.2 for both Microsoft SQL Server 2005 SP3 and Microsoft SQL Server 2008.

4. Open the properties for the JBoss for Adobe LiveCycle service and click the **Log On** tab.

5. Select **This Account** and type the value of the user account that you mapped in [“Map the Windows login to the LiveCycle ES2 database user” on page 79](#). If you are running JBoss from the command line, you must be logged in as this user.

4.7 Next steps

Install LiveCycle ES2 by following the instructions provided in [Installing and Deploying LiveCycle ES2 for JBoss](#).

5

Configuring a WebLogic Server

This chapter describes how to install and configure the Oracle WebLogic Server that will host your LiveCycle ES2 installation.

This section uses the following conventions.

Name	Default values for WebLogic 10g	Default values for WebLogic 11g
<i>[appserver root]</i>	(Windows) C:\bea\wlserver_10.3 (Linux and Solaris) /opt/bea/wlserver_10.3	(Windows) C:\Oracle\Middleware\wlserver_10.3 (Linux and Solaris) /opt/Oracle/Middleware/wlserver_10.3
<i>WL_HOME</i>	(Windows) C:\bea (Linux and Solaris) /opt/bea	(Windows) C:\Oracle\Middleware (Linux and Solaris) /opt/Oracle/Middleware
<i>[appserverdomain]</i> By default the domain is called <i>base_domain</i> .	(Windows) C:\bea\user_projects\domains\base_domain (Linux and Solaris) /opt/bea/user_projects/domains/base_domain	(Windows) C:\Oracle\Middleware\user_projects\domains\base_domain (Linux and Solaris) /opt/Oracle/Middleware/user_projects/domains/base_domain

5.1 Overview

You will perform the following tasks to install and configure your Oracle WebLogic Server 10g environment:

1. Obtain WebLogic Server and install as per the manufacturer's documentation. (See ["Installing WebLogic Server" on page 82.](#))
2. Access the WebLogic Server Administration Console.
3. Create a Managed Server. (See ["Creating a new WebLogic Managed Server" on page 86.](#))
4. Configure application server (See ["Configuring WebLogic Server" on page 88.](#))

5.2 LiveCycle ES2 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 LiveCycle 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 LiveCycle Configuration Manager. However, you must manually enter the LiveCycle Configuration Manager changes into the StartManagedWebLogic scripts. For the appropriate steps, see [Installing and Deploying LiveCycle ES2 for WebLogic](#).

5.3 Installing WebLogic Server

Obtain Oracle WebLogic Server for your operating system and install it according to the manufacturer's documentation (see <http://edocs.bea.com/wls/docs103/getstart/overview.html>).

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

5.3.1 Configuring WebLogic 10g R3 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. (See your WebLogic documentation for more information.)
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.

5.4 Installing the JRockit Java JDK

This section describes how to install the JRockit Java 6 JDK for WebLogic 10g R3 and 11g R1.

► **(Windows or Linux only) To install the JDK:**

- Obtain the Oracle JRockit® Real Time 3.1.2-1.6.0.14 for your operating system and install it according to the JRockit instructions.

The `JAVA_HOME` and `PATH` environment variables must point to the JDK on the server where LiveCycle ES2 will be deployed.

► **To 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\JRockit Real Time\jrirt-3.1.2-1.6.0
```

Note: If you are using PDF Generator ES2 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>.

► **To 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.

► **To set the JAVA_HOME environment (Linux and Solaris):**

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

```
JAVA_HOME=/opt/jrockit 1.6.0.14  
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 LiveCycle PDF Generator ES2 to convert Open Office files on a WebLogic Server 10g environment.

► **To 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
```

► **(Windows, Linux, or Solaris) To 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 Java version `1.6.0_14` (or later version of 1.6.0).

5.5 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.

► To create a WebLogic 10g domain with Managed Servers:

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

- (Windows) `config.cmd`
- (Linux 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**.

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

4. 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 LiveCycle Configuration Manager will prompt you for them.

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

Caution: Using Development Mode is not recommended.

6. 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-3.1.2-1.6.0` (Windows) or `/opt/JRockit Real Time/jrrt-3.1.2-1.6.0` (Linux)
- (Solaris) Location of **Sun JDK 1.6** or later. For example, `/opt/jdk1.6.0_14`.

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

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

9. On the Creating Domain screen, when the configuration creation is 100% complete, do the following tasks:
 - (Windows) Select **Start Admin Server** and click **Done**.
 - (Linux and Solaris) Click **Done**, and start the server by navigating from a command prompt to the `[appserverdomain]` and typing `./startWebLogic.sh`
10. (Windows, Linux, or Solaris) When prompted, enter the WebLogic user name and password that you entered in step 4.

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

➤ **To 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 LiveCycle Configuration Manager will prompt you for them.
6. In the WebLogic Domain Startup Mode panel, select **Production Mode**.

Caution: 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\jrirt-3.1.2-1.6.0` (Windows) or `/opt/JRockit Real Time/jrirt-3.1.2-1.6.0` (Linux)
 - (Solaris) Location of **Sun JDK 1.6** or later. For example, `/opt/jdk1.6.0_14`.
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 4.

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

5.6 Starting Node Manager

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

► To start Node Manager:

- 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.

5.7 Creating a new WebLogic Managed Server

You must deploy LiveCycle ES2 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.

► To create a new WebLogic Managed Server:

1. If WebLogic Administration Server is not already running, from a command prompt, navigate to the `[WL_HOME]\user_projects\domains\[domainname]` 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 LiveCycle 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 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**.

LiveCycle Configuration Manager does not set the minimum and maximum memory settings for WebLogic, therefore you must configure LiveCycle ES2 manually using the WebLogic Server Administration Console.

17. On the Configuration tab, click the **Server Start** tab.
18. In the **Arguments** box, type the appropriate text to set the memory size for the Managed Server:
 - `-Xms256m -Xmx1792m -XX:MaxPermSize=512m`
19. Click **Save** and then click **Activate Changes**.

► **To complete the memory settings for the Managed Server on WebLogic 10g:**

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 `MEM_ARGS=-Xms256m -Xmx512m` and change it to `MEM_ARGS=-Xms256m -Xmx1024m`
4. Search for the line `-XX:MaxPermSize=128m` 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.

► **To 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=128m` 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.

5.8 Configuring WebLogic Server

You must make the following configuration changes to optimize your application server's performance for your LiveCycle ES2 installation.

5.8.1 Configuring WebLogic for Web Services

For LiveCycle ES2 to accept requests using web services, you must complete the following procedure to modify authentication of the servlet container.

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

1. Start the WebLogic Administration Server.
2. Open a command prompt and set the environment by using `setWLSEnv`, as shown in this example:
 - (Windows) From the `[WL_HOME]\wlserver_10.3\server\bin\` folder, type `setWLSEnv.cmd`
 - (Linux and Solaris) From the `[WL_HOME]/wlserver_10.3/server/bin/` directory, type `setWLSEnv.sh`
3. Start the WebLogic scripting tool by typing the following command:

```
java weblogic.WLST
```

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

- From a command prompt, set the environment by using `<WL_Home>/wlserver_10.3/server/bin/setWLSEnv.cmd`
- From a command prompt, copy the classpath that is returned from the `setWLSEnv.cmd` or `./setWLSEnv.sh` command and append `export CLASSPATH`, as shown in this example:

```
CLASSPATH=C:\bea\patch_wls1030\profiles\default\sys_manifest_classpath\w  
eblogic_patch.jar;C:\bea\patch_cie660\profiles\default\sys_manifest_clas  
spath\weblogic_patch.jar;C:\bea\JROCKI~1\lib\tools.jar;C:\bea\WLSERV~1.3  
\server\lib\weblogic_sp.jar;C:\bea\WLSERV~1.3\server\lib\weblogic.jar;C:  
\bea\modules\features\weblogic.server.modules_10.3.0.0.jar;C:\bea\WLSERV  
~1.3\server\lib\webservices.jar;C:\bea\modules\ORGAPA~1.5/lib/ant-all.ja  
r;C:\bea\modules\NETSFA~1.0_1/lib/ant-contrib.jar export CLASSPATH
```

- From a command prompt, copy the path returned from `setWLSEnv.cmd` or `./setWLSEnv.sh` and append `export PATH`, as shown in this example:

```
PATH=C:\bea\patch_wls1030\profiles\default\native;C:\bea\patch_cie660\profiles\default\native;C:\bea\WLSERV~1.3\server\native\win\32;C:\bea\WLSERV~1.3\server\bin;C:\bea\modules\ORGAPA~1.5\bin;C:\bea\JROCKI~1\jre\bin;C:\bea\JROCKI~1\bin;C:\Program Files\Common Files\NetSarang;c:\Program Files\Java\jdk1.5.0_17\bin;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\Program Files\Perforce;C:\bea\WLSERV~1.3\server\native\win\32\oci920_8 export PATH
```

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

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

Note: The WebLogic URL will be in the format `t3://hostname:[port]`, where `[port]` is the administration server port (usually 7001).

5.9 Stopping and restarting WebLogic

Note: 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.

► 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.

► **Stop WebLogic Administration Server:**

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

► **Restart WebLogic Administration Server:**

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

► **Restart 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`

► **Restart 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**.

5.10 Configuring the LiveCycle ES2 database connectivity

All configurations required for database connectivity are discussed in [Installing and Deploying LiveCycle ES2 for WebLogic](#).

5.11 Next steps

Install LiveCycle ES2 by following the instructions provided in [Installing and Deploying LiveCycle ES2 for WebLogic](#).

6

Configuring a WebSphere Application Server

This chapter describes how to install and configure the WebSphere Application Server that will host your LiveCycle ES2 installation.

This section uses the following conventions.

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

6.1 Installing WebSphere Application Server

You must install WebSphere Application Server for running LiveCycle ES2 products. For use with LiveCycle ES2, 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

(WebSphere 6.1) After you install WebSphere 6.1, you must update to WebSphere 6.1.0.21 and SR 8 before you deploy LiveCycle ES2. See the [WebSphere support website](#) for access to the WebSphere updates.

(WebSphere 7.0) After you install WebSphere 7.0, you must update to WebSphere 7.0.0.9 and SR 7 before you deploy LiveCycle ES2. See the [WebSphere support website](#) for access to the WebSphere updates.

6.1.2 Setting the 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 ES2 will be deployed.

► **To 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. See ["Third-party infrastructure support" on page 10](#). For example, type the following path:

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

► **To 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;
```

► **To 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 LiveCycle Configuration Manager: "No JRE is found..."

► **To 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.

► **To 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
```

► **(Windows, AIX, Linux, or Solaris) To 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 Java version installed.

6.1.3 Starting WebSphere Base and accessing WebSphere Administrative Console

Several procedures in this section require you to access the administrative console of the WebSphere Application Server.

► **To start WebSphere Base:**

1. If the server is not already running, start the WebSphere Administrative Console, from *[application server home]/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`

► **To 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 administering the server that you are currently using, you can replace `[host name]` 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.

3. If WebSphere Administrative Security is enabled, type the WebSphere user ID and password in the boxes provided.
4. Click **Log In**.

6.1.4 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://[host name]:[port]/ibm/console` in the address bar of a web browser.

If you are administering the server that you are currently using, you can replace `[host name]` 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.

6.1.5 Setting directory permissions

The LiveCycle ES2 application must extract files to the `[appserver root]/installableApps` 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.

► **To modify the location for the extracted files:**

1. Log in to the WebSphere Administrative Console.
2. Do one of the following:
 - (WebSphere 6.1) Click **Servers > Application servers** and click your server name, such as **server1**.
 - (WebSphere 7.0) 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/[server]/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 ES2 deployment.

6.2.1 Required application server instances

LiveCycle ES2 requires one or more application server instances. LiveCycle ES2 is deployed to one application server instance and, if you are using LiveCycle ES2 Business Activity Monitoring, LiveCycle ES2 must be installed to another application server instance.

6.2.2 Increasing the SOAP request time out

You must modify the SOAP request time-out value for LiveCycle ES2.

► **To increase the SOAP request time-out value:**

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]/profiles/<LiveCycle Server Profile Name>/properties/soap.client.props`
 - `[appserver root]/profiles/<LiveCycle Server Profile Name>/temp/soap.client.props`
 - (WebSphere 6.1 only)
`[appserver root]/profileTemplates/cell/default/documents/properties/soap.client.props`
 - (WebSphere 6.1 only)
`[appserver root]/cip/profileTemplates/minimal/documents/properties/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 each `soap.client.props` file.
4. In the navigation tree of the WebSphere Administrative Console, do one of the following:
 - **(WebSphere 6.1)** Click **Servers > Application Servers** and, in the right pane, click the server name.
 - **(WebSphere 7.0)** 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**.

Note: For deploying Content Services ES2, the special setting `-Dhibernate.dialect=org.hibernate.dialect.DB2Dialect` is required for the WebSphere Application Server and IBM DB2 64-bit database combination. This setting is required only when there are multiple schema on the same DB2 instance.

6.2.3 Increasing the Deployer heap size

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

► 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.

► 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`.
4. Save and close the file.

► Increase MaxPermSize (WebSphere on Solaris):

1. Log in to the WebSphere Administrative Console.

2. In the navigation tree of the WebSphere Administrative Console, do one of the following:
 - **(WebSphere 6.1)** Click **Servers > Application Servers** and, in the right pane, click the server name.
 - **(WebSphere 7.0)** 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.

6.3 Configuring the LiveCycle ES2 database connectivity

All configurations that are required for database connectivity are discussed in [Installing and Deploying LiveCycle ES2 for WebSphere](#).

6.4 Next steps

Install LiveCycle ES2 by following the instructions provided in [Installing and Deploying LiveCycle ES2 for WebSphere](#).

7

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
JDK Install location (for 32 bit or 64 bit OS)	<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 68 or "Configuring a WebLogic Server" on page 81 or "Configuring a WebSphere Application Server" on page 91 ?	<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 ES2 Database" on page 55 ?	<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 ES/PDF 3D Generator ES 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.

A

Appendix - Manually Configuring JBoss

This appendix describes the configuration that is required for the default JBoss application server that you can download from the Internet. This option should be considered for advanced installations only. Advanced knowledge of JBoss is typically required.

Topics in this chapter relate to JBoss versions 4.2.0 and 4.2.1. Steps specific to versions are indicated accordingly.

LiveCycle ES2 runs on JBoss on Windows Server 2003 and 2008 (Enterprise Edition or Standard Edition), Red Hat Linux ES/AS 5.0, SUSE Linux ES 10 platforms, and Solaris 10. (See [“Third-party infrastructure support” on page 10.](#))

Note: If you plan to use the LiveCycle ES2 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 ES2 Using Turnkey.](#)

This chapter uses the following conventions.

Name	Default value
<i>[appserver root]</i>	(JBoss 4.2.0 or 4.2.1 on Windows): C:\jboss (JBoss 4.2.0 or 4.2.1 on Linux and Solaris): /opt/jboss (JBoss EAP 4.3 on Windows): C:\jboss-eap-4.3\jboss-as (JBoss EAP 4.3 on Linux and Solaris): /opt/jboss-eap-4.3/jboss-as

A.1 Installing the JDK for JBoss

Download and install one of the following Sun JDK versions from www.java.sun.com according to your JBoss version.

- (JBoss 4.2.0) Sun JDK 5.0 update 11 or later updates to 5.0
- (JBoss 4.2.1 or JBoss EAP 4.3 on all platforms) Sun JDK 6.0 update 14 or later updates to 6.0

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

A.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_14
```

A.1.2 Set the PATH environment variable (Windows)

5. Select **Start > Control Panel > System**.
6. Click the **Advanced** tab and click **Environment Variables**.
7. In the System Variables area, select the `PATH` variable and then click **Edit**.
8. 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.

A.1.3 Set the JAVA_HOME environment variable (Linux and Solaris)

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

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

A.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
```

A.1.5 Verify JAVA_HOME environment variable setting (Windows, Linux, or Solaris)

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

```
java -version
```

You should receive a response that begins with Java version `1.6.0_14` (or later version of 1.6.0).

A.2 Manually installing JBoss

If you are manually configuring JBoss for running LiveCycle ES2, download and unpackage JBoss Application Server.

A.2.1 Download JBoss

1. Go to the [JBoss Application Server Download](#) site.

2. Locate the **4.2.0** or **4.2.1** entry and click the **Download** link beside it.
3. On the download page, in the list of files, click the appropriate file for your operating system:
 - (Windows) jboss-4.2.0.GA.zip or jboss-4.2.1.GA.zip
 - (Linux) jboss-4.2.0.GA-src.tar.gz or jboss-4.2.1.GA-src.tar.gz
4. After the download is complete, extract the file to the directory of your choice on your server.

A.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.

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

A.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 [config] -b [server_IP_Address]`
 - (Linux and Solaris) `./run.sh -c [config] -b [server_IP_Address]`where `[config]` is the configuration required for your database and `[server_IP_Address]` is the actual IP address of the server. For example:
 - Using Adobe-preconfigured JBoss:
 - (Windows) `run.bat -c lc <database name> -b [server_IP_Address]`
 - (Linux and Solaris) `./run.sh -c lc <database name> -b [server_IP_Address]`
 - Using manually configured JBoss:
 - (Windows) `run.bat -c all -b [server_IP_Address]`
 - (Linux and Solaris) `./run.sh -c all -b [server_IP_Address]`

A.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 name>:<jndi -port>`
 - (Linux and Solaris) `./shutdown.sh -s <server name>:<jndi -port>`

A.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.

A.4.1 Remove JMS and clustering configuration files

Because LiveCycle ES2 does not use JMS configuration, you can delete the following files and directories that are part of the downloaded 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\all\deploy-hasingleton`
- (JBoss 4.2.0 and 4.2.1 only) `[appserver root]\server\all\deploy\jms`
- `[appserver root]\server\all\deploy\deploy-hasingleton-service.xml`

If you're using JBoss EAP 4.3, perform these additional steps:

- Remove the `[appserver root]\server\all\deploy\jboss-messaging.sar` folder
- Open `[appserver root]\server\all\conf\jbossjta-properties.xml` and comment out the following property:

```
com.arjuna.ats.jta.recovery.XAResourceRecovery.JBMESSAGING1
```

If you are installing JBoss on a single server, remove the following files and directories related to configuration of clusters:

- `[appserver root]\server\all\deploy\deploy.last` (JBoss 4.2.0 and JBoss EAP 4.3 only)
- `[appserver root]\server\all\deploy\jboss-web-cluster.sar`
- `[appserver root]\server\all\farm`
- `[appserver root]\server\all\deploy\cluster-service.xml`
- `[appserver root]\server\all\deploy\ejb3-clustered-sfsbcache-service.xml`
- `[appserver root]\server\all\deploy\ejb3-entity-cache-service.xml`

A.4.2 Disable cluster configuration

In addition to removing the clustering-specific configuration files, you must edit the `jboss-service.xml` to disabled clustering.

1. Open the `[appserver root]\server\all\deploy\jboss-web.deployer\META-INF\jboss-service.xml` file in an editor.

2. Locate the following lines and comment them out:

```
<depends>jboss.cache:service=TomcatClusteringCache</depends>  
<depends>jboss:service=TransactionManager</depends>
```

3. Save the file.

A.4.3 Modify the JBoss configuration

Perform the following steps to modify the JBoss configuration to customize JBoss for LiveCycle ES2.

- Update the `jacorb.properties` file
 - Update the `quartz.jar` and `commons-collections.jar` libraries
 - Modify the EAR and WAR file class-loading isolation
 - Disable the JavaServer Faces (JSF) libraries (JBoss 4.2.0 only)
 - Disable cluster service (for single node installations only)
 - Modify the `run.bat` file (for Windows)
 - Modify `run.conf`
 - Modify `run.sh` (for Linux and Solaris, 64-bit only)
 - Modify `log4j.xml`
 - Remove the JMX Console and Web Console
 - Modify the `jmx-invoker-service.xml` file so that authenticated users are not required
- **Update the `jacorb.properties` file**
1. Open the `[appserver root]/server/all/conf/jacorb.properties` file in a text editor.
 2. Locate the `jacorb.poa.thread_pool_max` setting and change the value to 16.
- **Update the `quartz.jar` library (JBoss 4.2.0 only)**
1. Navigate to the `[appserver root]/jboss_4.2.0/patches` directory.
 2. Copy the `quartz.jar` file from this directory to the `[appserver root]/server/all/lib` directory.
- **Modify EAR file class-loading isolation**
1. Open the `[appserver root]/server/all/conf/jboss-service.xml` file in an editor.
 2. Locate `<attribute name="CallByValue">` and change the value to `true`.
 3. Save and close the file.
 4. Open the `[appserver root]/server/all/deploy/ear-deployer.xml` file in an editor.
 5. Locate `<attribute name="Isolated">` and change the value to `true`.
 6. Locate `<attribute name="CallByValue">` and change the value to `true`.
 7. Save and close the file.
- **Disable the JSF libraries**
1. Navigate to the `[appserver root]/server/all/deploy/jboss-web.deployer` directory and delete the `jsf-libs` subdirectory and all its files.
 2. Comment out or remove the following entries from `[appserver root]/server/all/deploy/jboss-web.deployer/conf/web.xml` if they aren't already commented out:

```
<!-- Comment/Remove this -->
  <!-- Configures JSF for a web application if the
  javax.faces.webapp.FacesServlet is declared -->
```

```
<!-- in web.xml.
-->
<!--
<listener>

<listener-class>org.jboss.web.jsf.integration.config.JBossJSFConfigureListen
er</listener-class>
</listener>
-->
<!-- Comment/Remove this -->
<!-- Listens to all web app lifecycle events so that @PreDestroy can be
called on -->
<!-- JSF managed beans that go out of scope. You can comment this out if you
-->
<!-- don't use JSF or you don't use annotations on your managed beans.
-->
<!--
<listener>

<listener-class>com.sun.faces.application.WebappLifecycleListener</listener-
class>
</listener>
-->

<!--
<init-param>
  <description>JSF standard tlds</description>
  <param-name>tagLibJar0</param-name>
  <param-value>jsf-libs/jsf-impl.jar</param-value>
</init-param>
-->
```

► Modify the run.bat file (Windows only)

1. Open the `[appserver root]/bin/run.bat` file in an editor.
2. Modify the line immediately following `rem Setup JBoss specific properties to set encoding to UTF-8:`

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

Note: Ensure that no line breaks appear in this new line.

3. Modify the line immediately following `rem Sun JVM memory allocation pool parameters to read as follows:`
 - (32-bit Windows) `set JAVA_OPTS=%JAVA_OPTS% -XX:MaxPermSize=256m -Xms1024m -Xmx1024m -Djava.net.preferIPv4Stack=true`
 - (64-bit Windows) `set JAVA_OPTS=%JAVA_OPTS% -XX:MaxPermSize=512m -Xms1024m -Xmx1792m -Djava.net.preferIPv4Stack=true`
4. (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)"
```

5. Save and close the file.

► Modify the `run.sh` file (JBoss with Solaris 10, Red Hat 5.0, 64-bit only)

Solaris and Linux 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.sh` file in an editor.

2. Locate the following line:

```
JAVA_OPTS=$JAVA_OPTS -XX:PermSize=256m -XX:MaxPermSize=256m -Xms1024m
-Xmx1024m
```

Note: The `JAVA_OPTS` arguments may be present in the `run.conf` file instead of the `run.sh` file. Locate these arguments in both file, and then make the required changes.

3. Change the `-XX:MaxPermSize=256m` and the `-Xmx1024m` arguments as follows:

```
JAVA_OPTS=%JAVA_OPTS% -XX:PermSize=256m -XX:MaxPermSize=512m -Xms1024m
-Xmx1792m
```

4. Add the following statement just after the `JAVA_OPTS` assignment statement in the file to set encoding to UTF-8:

```
JAVA_OPTS="$JAVA_OPTS -d64 -Dadobeidp.serverName=server1
-Dfile.encoding=utf8 -Dprogram.name=$PROGNAME
```

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

5. (Optional) Modify JBoss Application Server to run in IPv6 mode as follows:

- Locate and modify `-Djava.net.preferIPv4Stack=false`
- Add `-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)"
```

6. Save and close the file.

► Modify the `run.conf` file

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

2. Locate `JAVA_OPTS="-server -Xms128m -Xmx128m"` and modify it to read as follows for 32-bit operating systems:

```
JAVA_OPTS="-server -XX:MaxPermSize=256m -Xms1024m -Xmx1024m"  
-Dsun.rmi.dgc.client.gcInterval=3600000  
-Dsun.rmi.dgc.server.gcInterval=3600000
```

For 64-bit operating systems, modify the line to read as follows:

```
JAVA_OPTS="-server -d64 -XX:MaxPermSize=512m -Xms1024m -Xmx1792m"  
-Dsun.rmi.dgc.client.gcInterval=3600000  
-Dsun.rmi.dgc.server.gcInterval=3600000
```

3. (*Linux only*) You must also add the `"-Djava.net.preferIPv4Stack=true"` flag to `JAVA_OPTS` to remove the warning "[HANamingService] Failed to start AutomaticDiscovery".
4. Save and close the file.

► Modify the `log4j.xml` file to increase the logging level from `DEBUG` to `INFO`

1. Open the `[appserver root]/server/all/conf/jboss-log4j.xml` file in an editor.
2. Locate the following text in the `FILE` appender section and add the line that appears in bold:

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

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

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

Modify the last line to read as follows:

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

4. Locate the `Limit categories` section and add the following:

```
<category name="com.adobe">  
  <priority value="INFO"/>  
</category>  
<category name="org.apache.xml.security.signature.Reference">  
  <priority value="WARN"/>  
</category>  
<category name="org.alfresco">  
  <priority value="WARN"/>  
</category>  
<category name="org.alfresco.repo.policy">  
  <priority value="WARN"/>  
</category>  
<category name="org.springframework">  
  <priority value="WARN"/>  
</category>  
<category name="org.hibernate">
```

```
<priority value="WARN"/>
</category>
<category name="org.hibernate.cache.ReadWriteCache">
  <priority value="ERROR"/>
</category>
<category name="org.hibernate.cache.EhCacheProvider">
  <priority value="ERROR"/>
</category>
<category name="org.hibernate.engine.
StatefulPersistenceContext.ProxyWarnLog">
  <priority value="ERROR"/>
</category>
<category name="org.jbpm.jpdl.xml.JpdlXmlReader">
  <priority value="ERROR"/>
</category>
```

5. Save and close the file.

► Remove the JMX Console and Web Console

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

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

► Modify the `jmx-invoker-service.xml` file

1. Navigate to the `[appserver root]/server/all/deploy` directory and open the `jmx-invoker-service.xml` file in a text editor.

2. Ensure that the following lines are commented out in the `invoke` section:

```
<interceptor code="org.jboss.jmx.connector.invoker.AuthenticationInterceptor"
securityDomain="java:/jaas/jmx-console"/>
```

3. Save and close the file.

► Modify the `jbossjta-properties.xml` file:

1. Ensure that transaction management works as expected by locating the `jbossjta-properties.xml` file in the `[appserver root]/server/all/conf` directory and opening the file in an editor.

2. Locate and modify the `properties` element as follows (modification in bold). Add the property in bold if it doesn't already exist in the `jbossjta-properties.xml` file.

```
<properties depends="arjuna" name="jta">
  <property name="com.arjuna.ats.jta.allowMultipleLastResources"
value="true"/>
  <!-- ... other properties ... -->
</properties>
```

3. Save and close the file.

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

A.4.4 Changing the hibernate option (Content Services ES2 only)

If you are installing and deploying Content Services ES2 on a manually-configured JBoss environment, you must change hibernate to use the `CGLIB` bytecode provider instead of `JAVASSIST` otherwise Content Services ES2 will fail to deploy.

► Modify the hibernate bytecode provider

1. Locate and open the `persistence.properties` file:
`[JBOSS_HOME]/server/all/deploy/ejb3.deployer/META-INF/`
2. Locate and modify the `hibernate.bytecode.provider` value to `cglib`.
3. Save and close the file.

A.5 LiveCycle ES2 database connectivity

To configure the LiveCycle ES2 database connectivity, you must complete the following tasks:

- Configure the LiveCycle ES2 data source.
- Configure JBoss to use your database as the default data source.

You must install database drivers to the installation directories of the application server. Drivers are required to enable LiveCycle Configuration Manager and the application server to connect to the LiveCycle ES2 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 a MySQL, Oracle, or SQL Server data source.

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

A.5.1 Configuring MySQL for manually installed JBoss

To enable JBoss to connect to the MySQL database that stores LiveCycle ES2 data, you must complete these tasks.

- Obtain and copy the MySQL JDBC driver to the instance of JBoss where you will deploy LiveCycle ES2.
- Create a data source file and deploy it to the instance of JBoss where you will deploy LiveCycle ES2.
- 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.

If you are using Adobe-preconfigured JBoss, see [“Configuring MySQL for Adobe-preconfigured JBoss 4.2.1” on page 72](#).

A.5.1.1 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” on page 66](#).)

A.5.1.2 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/all/lib` directory.

A.5.1.3 Edit `adobe-ds.xml` file

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

3. `<password>adobe</password>` 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 ES2 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 ES2 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 the file.

A.5.1.4 Set MySQL as the data source

1. Navigate to the `[appserver root]/server/all/deploy` directory and delete the `hsqldb-ds.xml` file.
2. Navigate to the `[appserver root]/docs/examples/jca` directory and copy the `mysql-ds.xml` file to the `[appserver root]/server/all/deploy` directory.
3. Open the `[appserver root]/server/all/deploy/mysql-ds.xml` file in a text editor and modify the `<local-tx-datasource>` element with your MySQL connection settings:

```
<jndi-name>MySQLDS</jndi-name>
```

```
<connection-url>jdbc:mysql://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. Save and close the file.

A.5.1.5 Edit the standardjbosscomp-jdbc.xml file

1. Open the `[appserver root]\server\all\conf\standardjbosscomp-jdbc.xml` file and change the following elements (not necessarily contiguous):

```
<!-- optional since 4.0 <datasource-mapping>Hypersonic SQL  
</datasource-mapping> -->  
  <fk-constraint>>false</fk-constraint>
```

- Remove the text `<!-- optional since 4.0` and the trailing `-->` comment tag from the `<datasource-mapping>` tag to uncomment it.
- Replace `Hypersonic SQL` with `MySQL` in the `<datasource-mapping>` element.

The final `<datasource-mapping>` tag will look like the following line:

```
  <datasource-mapping>MySQL</datasource-mapping>
```

- Replace `false` with `true` in the `<fk-constraint>` element.

2. Save and close the file.

A.5.1.6 Edit the login-config.xml file

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

```
<application-policy name="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.

A.5.2 Configuring Oracle for manually installed JBoss

To enable JBoss to connect to the Oracle database that stores LiveCycle ES2 data, you must complete the following tasks if you are manually deploying LiveCycle ES2:

- Obtain and copy the Oracle JDBC driver to the instance of JBoss where you will deploy LiveCycle ES2.
- Create a data source file and deploy it to the instance of JBoss where you will deploy LiveCycle ES2.
- 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_clear.html.

A.5.2.1 Install the Oracle 11g R2 database driver

Copy the `ojdbc5.jar` for JDK 1.5 and `ojdbc6.jar` for JDK 1.6 driver file from the `[LiveCycleES2 root]/lib/db/oracle` directory to the `[appserver root]/server/all/lib` directory. You can also download the Oracle 11g R2 Release 1(11.1.0.6 Thin) driver from the [JDBC Driver Downloads](#) site.

A.5.2.2 Edit `adobe-ds.xml` file

1. Copy the `adobe-ds.xml` file from the `[DVD_root]/third_party/datasources/lc_oracle/deploy` directory to the `[appserver root]/server/all/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 ES2 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 the file.

A.5.2.3 Set Oracle as the data source

If you are running LiveCycle ES2 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/all/lib` directory.

1. Navigate to the `[appserver root]/server/all/deploy` directory and delete the `hsqldb-ds.xml` file.
2. Copy the `oracle-ds.xml` file from the `[appserver root]/docs/examples/jca` directory to the `[appserver root]/server/all/deploy` directory.
3. Open the `[appserver root]/server/all/deploy/oracle-ds.xml` file in a text editor and modify the `<local-tx-datasource>` element with your Oracle connection settings:

```
<jndi-name>OracleDS</jndi-name>
<connection-url>jdbc:oracle:thin:@youroraclehost:1521:yoursid
</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.

A.5.2.4 Edit the standardjbosscomp-jdbc.xml file

1. Open the `[appserver root]/server/all/conf/standardjbosscomp-jdbc.xml` file and change the following elements (not necessarily on consecutive lines):

```
<!-- optional since 4.0 <datasource-mapping>Hypersonic SQL  
</datasource-mapping> -->  
  <fk-constraint>>false</fk-constraint>
```

- Remove the text `<!-- optional since 4.0` and the trailing `-->` comment tag from the `<datasource-mapping>` tag to uncomment it.
- Replace `Hypersonic SQL` with `Oracle10g` in the `<datasource-mapping>` element.

The final `<datasource-mapping>` tag will look like the following line:

```
<datasource-mapping>Oracle10g</datasource-mapping>
```

Note: The Oracle database version that you specify in `[appserver root]/server/all/deploy/oracle-ds.xml` and `[appserver root]/server/all/conf/standardjbosscomp-jdbc.xml` must match.

- Replace `false` in the `<fk-constraint>` with `true`.
2. Save and close the file.

A.5.2.5 Edit the login-config.xml file

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

```
<application-policy name = "OracleDbRealm">
  <authentication>
    <login-module code =
      "org.jboss.resource.security.ConfiguredIdentityLoginModule" flag =
      "required">
      <module-option name = "principal">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:

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

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

A.5.3 Configuring SQL Server

To enable JBoss to connect to the SQL Server database that stores LiveCycle ES2 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 ES2.
- Create a SQL Server data source file and deploy it to the instance of JBoss where you will deploy LiveCycle ES2, such as `[appserver_root]/server/all/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_clear.html.

A.5.3.1 Configuring the SQL Server data source

Before you configure the SQL Server data source, you must have the LiveCycle ES2 database created on SQL Server. (See ["Creating a SQL Server database" on page 56](#).)

A.5.3.2 Install the SQL database driver for JBoss

1. Obtain the SQL Server JDBC 1.2 database driver from the Microsoft website.

Note: Use SQL Server JDBC Driver 1.2 for both Microsoft SQL Server 2005 SP3 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 sqljdbc.jar file from the *[SQL_root]/sqljdbc_1.2/enu* directory to the *[appserver root]/server/all/lib* directory.
5. Delete the mysql-connector-java-3.1.12-bin.jar file located in the *[appserver root]/server/all/lib* directory.

A.5.3.3 Edit adobe-ds.xml file

1. Copy the adobe-ds.xml file from the *[DVD_root]/third_party/datasources/lc_sqlserver/deploy* directory to the *[appserver root]/server/all/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 ES2 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 ES2 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 the file.

A.5.3.4 Edit the mssql-ds.xml file

1. Navigate to the `[appserver root]/server/all/deploy` directory and delete the `hsqldb-ds.xml` file.
2. Navigate to the `[appserver root]/docs/examples/jca` directory and copy the `mssql-ds.xml` file to the `[appserver root]/server/all/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.

A.5.3.5 Edit the standardjbosscomp-jdbc.xml file

1. Open the `[appserver root]/server/all/conf/standardjbosscomp-jdbc.xml` file and change the following elements (not necessarily on consecutive lines):

```
<!-- optional since 4.0 <datasource-mapping>Hypersonic SQL
</datasource-mapping> -->
  <fk-constraint>>false</fk-constraint>
```

- Remove the text `<!-- optional since 4.0` and the trailing `-->` comment tag from the `<datasource-mapping>` tag to uncomment it.
- Replace `Hypersonic SQL` with `MS_SQLSERVER2000` in the `<datasource-mapping>` element.

The final `<datasource-mapping>` tag will look like the following line:

```
<datasource-mapping>MS_SQLSERVER2000</datasource-mapping>
```

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

- Replace `false` in the `<fk-constraint>` with `true`.

2. Save and close the file.

A.5.3.6 Edit the login-config.xml file

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

```
<application-policy name = "MSSQLDbRealm">
  <authentication>
    <login-module code =
      "org.jboss.resource.security.ConfiguredIdentityLoginModule"
      flag = "required">
      <module-option name = "principal">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.

A.5.3.7 Configure Integrated Security on Windows

1. Modify the `adobe-ds.xml` and `mssql-ds.xml` files, located in `[appserver root]\server\all\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 with the Microsoft SQL JDBC 1.2 driver installation (default is `[install dir]/sqljdbc_1.2/enu/auth/x86` for 32-bit operating systems and `[install dir]/sqljdbc_1.2/enu/auth/x64` for 64-bit operating systems).
3. Open the properties for the JBoss for Adobe LiveCycle service and click the **Log On** tab.
4. Select **This Account** 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.

A.6 Next steps

Install LiveCycle ES2 by following the instructions provided in [Installing and Deploying LiveCycle ES2 for JBoss](#).