



Preparing to Install LiveCycle® ES (Single Server)

February 2009

Adobe® LiveCycle® ES
Update 1 (8.2)

© 2009 Adobe Systems Incorporated. All rights reserved.

Adobe® LiveCycle® ES Update 2 (8.2) Preparing to Install LiveCycle® ES (Single Server) for Microsoft® Windows®, Linux®, and UNIX®
Edition 2.3, February 2009

If this guide is distributed with software that includes an end user agreement, this guide, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by any such license, no part of this guide may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Adobe Systems Incorporated. Please note that the content in this guide is protected under copyright law even if it is not distributed with software that includes an end user license agreement.

The content of this guide is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

Please remember that existing artwork or images that you may want to include in your project may be protected under copyright law. The unauthorized incorporation of such material into your new work could be a violation of the rights of the copyright owner. Please be sure to obtain any permission required from the copyright owner.

Any references to company names, company logos and user names in sample material or sample forms included in this documentation and/or software are for demonstration purposes only and are not intended to refer to any actual organization or persons.

Adobe, the Adobe logo, Acrobat, Flash, Flex, Flex Builder, FrameMaker, LiveCycle, PageMaker, Photoshop, PostScript, and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

AIX, DB2, IBM, and WebSphere are trademarks of International Business Machines Corporation in the United States, other countries, or both.

BEA WebLogic Server is a registered trademark of BEA Systems, Inc.

Intel and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the U.S. and other countries.

Java, Solaris, and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

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 Sun Microsystems, Inc.

JBoss and Red Hat are trademarks or registered trademarks 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, Windows, Windows Server, and Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Oracle is a trademark of Oracle Corporation and may be registered in certain jurisdictions.

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.

This product contains either BSAFE and/or TIPEM software by RSA Security, Inc.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

This product includes software developed by the IronSmith Project (<http://www.ironsmith.org/>).

This product includes software developed by the OpenSymphony Group (<http://www.opensymphony.com/>).

This product includes software developed by the Indiana University Extreme! Lab (<http://www.extreme.indiana.edu/>).

This product includes copyrighted software developed by E. Wray Johnson for use and distribution by the Object Data Management Group (<http://www.odmg.org/>).

Portions © Eastman Kodak Company, 199- and used under license. All rights reserved. Kodak is a registered trademark and Photo CD is a trademark of Eastman Kodak Company.

Powered by Celequest. Copyright 2005-2008 Adobe Systems Incorporated. All rights reserved. Contains technology distributed under license from Celequest Corporation. Copyright 2005 Celequest Corporation. All rights reserved.

Single sign-on, extending Active Directory to Adobe LiveCycle ES provided by Quest Software “www.quest.com/identity-management” in a subsequent minor release that is not a bug fix (i.e., version 1.1 to 1.2 but not 1.1.1 to 1.1.2) of the Licensee Product that incorporates the Licensed Product.

The Spelling portion of this product is based on Proximity Linguistic Technology.

©Copyright 1989, 2004 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 1990 Merriam-Webster Inc. © Copyright 1990 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 2003 Franklin Electronic Publishers Inc. © Copyright 2003 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 2004 Franklin Electronic Publishers, Inc. © Copyright 2004 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 1991 Dr.Lluis de Yzaguirre I Maura © Copyright 1991 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 1990 Munksgaard International Publishers Ltd. © Copyright 1990 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 1995 Van Dale Lexicografie bv © Copyright 1996 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 1990 IDE a.s. © Copyright 1990 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 2004 Franklin Electronics Publishers, Inc. © Copyright 2004 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 1992 Hachette/Franklin Electronic Publishers, Inc. © Copyright 2004 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 2004 Bertelsmann Lexikon Verlag © Copyright 2004 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 2004 MorphoLogic Inc. © Copyright 2004 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 1990 Williams Collins Sons & Co. Ltd. © Copyright 1990 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA. © Copyright 1993-95 Russicon Company Ltd.

© Copyright 1995 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 2004 IDE a.s. © Copyright 2004 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

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

Notice to U.S. Government End Users. The Software and Documentation are "Commercial Items," as that term is defined at 48 C.F.R. §2.101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation," as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Unpublished-rights reserved under the copyright laws of the United States. Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704, USA. For U.S. Government End Users, Adobe agrees to comply with all applicable equal opportunity laws including, if appropriate, the provisions of Executive Order 11246, as amended, Section 402 of the Vietnam Era Veterans Readjustment Assistance Act of 1974 (38 USC 4212), and Section 503 of the Rehabilitation Act of 1973, as amended, and the regulations at 41 CFR Parts 60-1 through 60-60, 60-250, and 60-741. The affirmative action clause and regulations contained in the preceding sentence shall be incorporated by reference.

Contents

About This Document.....	7
Before you start.....	7
What's in this document?.....	7
Who should read this document?.....	7
1 System Requirements for Installing LiveCycle ES	10
LiveCycle ES patch updates.....	10
Third-party infrastructure support.....	10
Microsoft Windows Server.....	10
Red Hat Enterprise Linux and SUSE Linux.....	11
IBM AIX.....	12
Sun Solaris	13
System requirements.....	14
Minimum hardware requirements	14
Intel x86 compatibility	15
Recommended hardware requirements	15
Additional hardware requirement for LiveCycle Content Services ES.....	15
Recommended hardware requirements for client-side machines	15
Supported software	16
Minimum database user permissions.....	20
Web browser support.....	21
Additional requirements for Linux or UNIX operating systems.....	22
Additional requirements for PDF Generator ES and PDF Generator 3D ES	23
Additional requirements for Connector for IBM Content Manager	26
LDAP configuration	28
Global document storage directory	28
Planning and creating the global document storage directory	28
Location of the global document storage directory	28
Sizing factors for the global document storage directory	29
Securing the global document storage directory.....	29
Backing up the global document storage directory	29
LiveCycle ES Business Activity Monitoring requirements.....	30
Basic requirements.....	30
Creating the Business Activity Monitoring metadata database	31
JBoss requirements	31
WebLogic requirements	32
WebSphere requirements.....	33
Installation considerations.....	33
Installing from network drives	33
Using LiveCycle ES with a Luna HSM cluster.....	33
Manual use of Acrobat restricted	34
Restricted use of unlicensed Process Management ES functionality.....	34
LiveCycle ES IPv6 support	34
Processes with document form variables and digital signatures.....	35

2	LiveCycle ES Credentials and Certificates.....	36
	Obtaining the Reader Extensions ES Rights credential	36
	Obtaining digital certificates for use with Digital Signatures ES.....	37
3	Creating the LiveCycle ES Database.....	38
	Database configuration requirements	38
	Creating an Oracle database	38
	Creating a SQL Server database.....	39
	Installing the SQL Server database driver	40
	Setting up SQL Server for LiveCycle ES	40
	Sizing your SQL Server database.....	41
	Creating the LiveCycle ES database user, schema, and login	42
	Creating a DB2 database	44
	DB2 user account	44
	Additional DB2 requirements for LiveCycle Content Services ES.....	46
	Creating a MySQL database	46
4	Configuring a JBoss Application Server	49
	Installing the J2SE SDK	49
	UNIX logon scripts.....	50
	Adobe-preconfigured JBoss.....	50
	Starting and stopping JBoss.....	51
	Notes about configuring the JBoss Application Server.....	51
	JBoss 4.2.0 start up configuration for 64-bit support.....	52
	Configuring the LiveCycle ES database connectivity.....	52
	Configuring MySQL for Adobe-preconfigured JBoss.....	53
	Configuring Oracle for Adobe-preconfigured JBoss 4.0.3 SP1 and 4.2.0.....	55
	Configuring SQL Server for Adobe-preconfigured JBoss	59
	Configuring DB2 for Adobe-preconfigured JBoss	66
	Next steps.....	71
5	Configuring a WebLogic Server.....	72
	Overview.....	72
	LiveCycle ES requirements for WebLogic	72
	Installing WebLogic Server	73
	Configuring WebLogic 10.1 Node Manager (64-bit Java)	73
	Installing the J2SE SDK	73
	Creating the WebLogic Server domain	75
	Troubleshooting the WebLogic 9.2 (64 bit).....	76
	Starting Node Manager.....	76
	Creating a new WebLogic Managed Server	77
	Configuring WebLogic Server.....	78
	Configuring anonymous admin look-up.....	78
	Configuring WebLogic for Web Services.....	79
	Stopping and restarting WebLogic.....	80
	Configuring the LiveCycle ES database connectivity.....	81
	Next steps.....	81

6	Configuring a WebSphere Application Server	82
	Installing WebSphere Application Server.....	82
	Installing a Fix Pack to WebSphere 6.1	82
	Setting the JAVA_HOME and PATH environment variables.....	82
	Accessing WebSphere Administrative Console	84
	Setting directory permissions.....	84
	Preparing WebSphere Application Server	85
	Increasing the MaxPermSize and heap size (UNIX only).....	86
	Configuring the LiveCycle ES database connectivity.....	86
	Next steps.....	86
7	Preconfigured Environment Checklist.....	87
A	Appendix - Manually Configuring JBoss.....	89
	Installing the J2SE SDK.....	89
	Manually installing JBoss.....	90
	Starting and stopping JBoss.....	90
	Notes about configuring the JBoss Application Server	91
	Modifying the JBoss configuration files	91
	Configuring the LiveCycle ES database connectivity.....	99
	Configuring MySQL for manually installed JBoss.....	99
	Configuring Oracle for manually installed JBoss.....	102
	Configuring SQL Server.....	105
	Configuring the SQL Server data source	105
	Configuring DB2 for manually installed JBoss.....	109
	Next steps.....	112
	Index	113

About This Document

Before you start

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

What's in this document?

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

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

For installations of Adobe LiveCycle Data Services ES, see the documentation that is provided with the Data Services ES solution component.

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

After you complete the tasks in this document, proceed to the *Installing and Deploying LiveCycle ES* document for your application server. See ["Additional information" on page 8](#) 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 ES. The information provided is based on the assumption that anyone reading this document is familiar with 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>[appserver root]</i>	The home directory of the application server that runs the LiveCycle ES services.	JBoss on Windows: C:\jboss JBoss on Linux, Solaris: /opt/jboss 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 on Windows: C:\bea\weblogic92\ C:\bea\wlserver_10.1 WebLogic on Linux and UNIX: /opt/bea/weblogic92 /opt/bea/wlserver_10.1
<i>BEA_HOME</i>	The install directory for WebLogic as specified for the <i>BEA_HOME</i> environment variable.	WebLogic on Windows C:\bea WebLogic on Linux and UNIX: /opt/bea
<i>[appserverdomain]</i>	The domain that you configured on WebLogic. The default domain is called <i>base_domain</i> .	WebLogic on Windows: C:\bea\user_projects\domains\base_domain WebLogic on Linux and UNIX: /opt/bea/user_projects/domains/base_domain
<i>[dbserver root]</i>	The location where the LiveCycle ES database server is installed.	Depends on the database type and your specification during installation.

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

Additional information

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

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

2

System Requirements for Installing LiveCycle ES

This section provides the information that you need before you begin installing LiveCycle ES, including system requirements, deployment topology information, and a list of the information about the target installation environment that you need to have during installation.

The following steps are required for preparing your server environment:

1. Verify that all system component versions comply with system requirements (this document).
2. Prepare the hardware as per your requirements (this document).
3. Install the operating system and update with all necessary patches and service packs (this document).
4. Install and configure the database server (this document).
5. Install and configure the application server (this document).
6. Install and deploy LiveCycle ES solution components (*Installing and Deploying LiveCycle ES* for your application server).

LiveCycle ES patch updates

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

Third-party infrastructure support

This section summarizes the referenced platform and database combinations for JBoss® Application Server 4.0.3 SP1 and 4.2.0, BEA WebLogic Server® 9.2 and 10.1, and IBM WebSphere® Application Server 6.1.0.7. To facilitate the use of WebLogic for joint Adobe and BEA customers, BEA has provided the [BEA Download](#) site from which you can download the exact version of WebLogic that LiveCycle ES requires.

Microsoft Windows Server

LiveCycle ES supports the following Microsoft Windows operating system:

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

Note: Microsoft Windows XP is supported for nonproduction environments.

This table lists the supported application server, Java™ Development Kit (JDK), and database configurations for this operating system.

Application server	JDK	OS/JVM architecture	Database
Red Hat JBoss Application Server 4.0.3 SP1	Sun JDK 5.0 Update 11 or later updates to 5.0	32-bit OS and 32-bit JVM™	<ul style="list-style-type: none"> MySQL 5.0 IBM DB2 8.2, 9.1 Oracle 9i, 10g Microsoft SQL Server 2005 SP2
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 IBM DB2 8.2, 9.1 Oracle 9i, 10g Microsoft SQL Server 2005 SP2
BEA WebLogic 9.2 (Advantage & Premium)	BEA JRockit® JDK supplied with application server	32-bit OS and 32-bit JVM	<ul style="list-style-type: none"> IBM DB2 8.2, 8.1 FixPack 7a Oracle 9i, 10g Microsoft SQL Server 2005 SP2
BEA WebLogic 10.1 MP1 (Advantage & Premium)	BEA JRockit JDK	32-bit OS and 32-bit JVM	<ul style="list-style-type: none"> MySQL 5.0
BEA WebLogic 10.1 MP1 (Advantage & Premium)	BEA JRockit JDK	64-bit OS and 64-bit JVM all others.	<ul style="list-style-type: none"> IBM DB2 8.2, 9.1 Oracle 9i, 10g Microsoft SQL Server 2005 SP2
IBM WebSphere 6.1.0.7 (Base & Network Deployment Edition) Note: It is recommended that WebSphere installations use the latest Fixpack and SR.	JDK 1.5 SR4 available for download from IBM	32-bit OS and 32-bit JVM or 64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> IBM DB2 8.2, 9.1 Oracle 9i, 10g Microsoft SQL Server 2005 SP2

Note: For a detailed list of the supported database editions, see [“Supported software” on page 24](#).

Red Hat Enterprise Linux and SUSE Linux

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

- Red Hat Enterprise Linux AS or ES 4.0 (x86/EDT/AMD64 32-bit architectures)
- Red Hat Enterprise Linux AS or ES 5.0 (EDT/AMD64 64-bit architectures)
- SUSE Linux Enterprise Server 9.0 (x86/EDT/AMD64 32-bit architectures)
- SUSE Linux Enterprise Server 10.0 (EDT/AMD64 64-bit architectures)

This table lists the supported application server, JDK, and database configurations for the Red Hat Enterprise Linux AS or ES 4.0 and SUSE Linux Enterprise Server 9.0 operating systems.

Application server	JDK	OS/JVM architecture	Database
Red Hat JBoss Application Server 4.0.3 SP1	(Red Hat Linux) Sun JDK 5.0 Update 11 or later updates to 5.0.	32-bit OS and 32-bit JVM	<ul style="list-style-type: none"> MySQL 5.0 IBM DB2 8.2, 9.1 Oracle 9i, 10g
BEA WebLogic 9.2 (Advantage & Premium)	BEA JRockit JDK	32-bit OS and 32-bit JVM	<ul style="list-style-type: none"> Oracle 9i, 10g
IBM WebSphere 6.1.0.7 (Base & Network Deployment Edition 6)	JDK 1.5 SR4 available for download from IBM	32-bit OS and 32-bit JVM	<ul style="list-style-type: none"> IBM DB2 8.2, 9.1 Oracle 9i, 10g
<p>Note: It is recommended that WebSphere installations use the latest Fixpack and SR.</p>			

Note: For a detailed list of the supported database editions, see [“Supported software” on page 24](#).

This table lists the supported application server, JDK, and database configurations for the Red Hat Enterprise Linux AS or ES 5.0 and SUSE Linux Enterprise Server 10.0 operating systems.

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 IBM DB2 8.2, 9.1 (Red Hat only) Oracle 9i, 10g
BEA WebLogic 10.1 MP1 (Advantage & Premium)	BEA JRockit JDK	64 bit OS and 64-bit JVM	<ul style="list-style-type: none"> Oracle 9i, 10g
IBM WebSphere 6.1.0.7 (Base & Network Deployment Edition)	JDK 1.5 SR4 available for download from IBM	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> IBM DB2 8.2, 9.1 Oracle 9i, 10g
<p>Note: It is recommended that WebSphere installations use the latest Fixpack and SR.</p>			

IBM AIX

LiveCycle ES supports the following IBM AIX operating system:

- AIX 5L 5.3 (64-bit and 32-bit architectures)

Note: It is recommended using the 64-bit architecture for AIX even when using the 32 bit JDK.

This table lists the supported application server, JDK, and database configurations for this operating system.

Application server	JDK	OS/JVM architecture	Database
IBM WebSphere 6.1.0.7 (Base & Network Deployment Edition)	JDK 1.5 SR4 available for download from IBM	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> • IBM DB2 8.2, 9.1 • Oracle 9i, 10g
IBM WebSphere 6.1.0.7 (Base & Network Deployment Edition)	JDK 1.5 SR4 available for download from IBM	64-bit OS and 32-bit JVM	<ul style="list-style-type: none"> • IBM DB2 8.2, 9.1 • Oracle 9i, 10g
Note: It is recommended that WebSphere installations use the latest Fixpack and SR.			

Note: For a detailed list of the supported database editions, see [“Supported software” on page 24](#).

Sun Solaris

LiveCycle ES supports the Sun Solaris 9, 10 (SPARC® architectures) operating systems.

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.

This table lists the supported application server, JDK, and database configurations for these operating systems.

Application server	JDK	OS/JVM architecture	Database
(Solaris 10 only) 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 9i, 10g
BEA WebLogic 9.2 (Advantage & Premium)	Sun JDK 5.0 Update 11 or later updates to 5.0.	64-bit OS and 32-bit JVM	<ul style="list-style-type: none"> • Oracle 9i, 10g
(Solaris 10 only) BEA WebLogic 10.1 MP1 (Advantage & Premium)	Sun JDK supplied with application server.	64-bit OS and 64-bit JVM	<ul style="list-style-type: none"> • Oracle 9i, 10g
IBM WebSphere 6.1.0.7 (Base & Network Deployment Edition)	JDK 1.5 SR4 available for download from IBM.	64-bit OS and 64-bit JVM only on Solaris 10 64-bit OS and 32-bit JVM on Solaris 9 and 10	<ul style="list-style-type: none"> • IBM DB2 8.2, 9.1 • Oracle 9i, 10g
Note: It is recommended that WebSphere installations use the latest Fixpack and SR.			

Note: For a detailed list of the supported database editions, see [“Supported software” on page 24](#).

System requirements

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

Minimum hardware requirements

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

Operating system	Minimum hardware requirement
Microsoft Windows Server® 2003 Enterprise Edition or Standard Edition SP2 and R2 (32-bit and 64-bit architectures)	Intel® Pentium® 4, 2.8 GHz processor VMWare ESX and GSX RAM: <ul style="list-style-type: none">● 2 GB (32-bit OS)● 3 GB (64-bit OS) Free disk space: 5.4 GB of temporary space plus 3.4 GB for LiveCycle ES
Sun Solaris 9 and 10	UltraSPARC® IIe, 650 MHz processor RAM: <ul style="list-style-type: none">● 3GB (64-bit OS with 64 bit JVM)● 2GB (64-bit OS with 32 bit JVM) Free disk space: 5.4 GB of temporary space plus 3.4 GB for LiveCycle ES
IBM AIX 5L 5.3	P4 pSeries 615 (Model 6C3) 7029-6C3, 1.2 GHz processor RAM: <ul style="list-style-type: none">● 3GB (64-bit OS with 64 bit JVM)● 2GB (64-bit OS with 32 bit JVM) Free disk space: 5.4 GB of temporary space plus 3.4 GB for LiveCycle ES
SUSE Linux Enterprise Server 9.0 (32-bit edition only)	Pentium 4 or x86 equivalent, 1 GHz processor RAM: <ul style="list-style-type: none">● 2 GB (32-bit OS) Free disk space: 5.4 GB of temporary space plus 3.4 GB for LiveCycle ES

Operating system	Minimum hardware requirement
SUSE Linux Enterprise Server 10.0 (64-bit edition only)	Pentium 4 or x86 equivalent, 1 GHz processor RAM: <ul style="list-style-type: none">• 3GB (64-bit OS with 64 bit JVM) Free disk space: 5.4 GB of temporary space plus 3.4 GB for LiveCycle ES
Red Hat Enterprise Linux AS or ES 4.0 (32-bit edition only)	Pentium 4 or x86 equivalent, 1 GHz processor RAM: <ul style="list-style-type: none">• 2 GB (32-bit OS) Free disk space: 5.4 GB of temporary space plus 3.4 GB for LiveCycle ES
Red Hat Enterprise Linux AS or ES 5.0 (64-bit edition only)	Pentium 4 or x86 equivalent, 1 GHz processor RAM: <ul style="list-style-type: none">• 3GB (64-bit OS with 64 bit JVM) Free disk space: 5.4 GB of temporary space plus 3.4 GB for LiveCycle ES

Intel x86 compatibility

On supported Windows and Linux environments, LiveCycle ES supports Intel EDT64 and AMD64 compatible chipsets running either 32-bit or 64-bit supported operating systems.

Recommended hardware requirements

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

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

Sun SPARC environments: UltraSPARC V or later.

IBM AIX environments: Power4 or later

Memory requirements: 4 GB of RAM.

Additional hardware requirement for LiveCycle Content Services ES

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

Recommended hardware requirements for client-side machines

The following minimum RAM requirements are recommended for client-side computers that are used either for development or for end-user interactions. These computers will need adequate resources for running applications (such as LiveCycle Workbench ES, LiveCycle Workspace ES, Adobe Flash Player, Adobe Reader) according to the tasks they will perform.

Development environment hardware requirements:

- Workbench ES: 1 GB of RAM
- Adobe Flex® Builder 2.0.1 Hot Fix 2 or later, or Flex SDK 2.0.1 Hot Fix 2 or later: 1 GB of RAM (2 GB recommended)
- Adobe Flash Player 9 or later: 512 MB of RAM (1 GB recommended)
- Adobe Acrobat® Professional and Acrobat Pro Extended, versions 7.0 to 9.0: 1 GB of RAM

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

End-user hardware requirements:

- LiveCycle Workspace ES: 1 GB of RAM
- Adobe Flash Player 9 or later: 512 MB of RAM (1 GB recommended)
- Adobe Reader 7.0.9: 128 MB of RAM (256 MB recommended)

Note: For web browser requirements, see [“End-user user interface” on page 29](#).

Supported software

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

Required software	Supported version
Operating system	<ul style="list-style-type: none"> ● Microsoft Windows <ul style="list-style-type: none"> ● Standard Server 2003 SP2 and later ● Enterprise Edition SP2 and later ● Standard Server R2 SP2 and later ● Enterprise Edition R2 SP2 and later running on 32-bit, 64-bit, and VMWare ESX/GSX architectures ● (Workbench ES) Microsoft Windows <ul style="list-style-type: none"> ● Windows Vista 32 bit and 64 bit ● XP Professional SP2 and above Platform 32 bit ● XP Tablet PC SP2 and above 32 bit ● 2003 Server SP1 and above 32 bit and 64 bit ● Sun Solaris 9 and 10 ● IBM AIX 5L 5.3 ● SUSE Linux Enterprise Server 9.0 (32-bit edition only) ● Red Hat Enterprise Linux AS or ES 4.0 (32-bit edition only) or ES 5.0 (64-bit edition only)
Application server	<ul style="list-style-type: none"> ● JBoss 4.0.3 SP1 ● JBoss 4.2.0 ● IBM WebSphere 6.1.0.7 (Base and Network Deployment editions) ● BEA WebLogic 9.2 (Advantage and Premium editions) ● BEA WebLogic 10.1 MP1 (Advantage and Premium editions) <p>Note: To facilitate the use of WebLogic for joint Adobe and BEA customers, BEA provides the BEA Download site from which you can download the exact version of WebLogic that LiveCycle ES requires.</p> <p>Note: It is recommended that WebSphere installations use the latest FixPack and SR.</p>
Web browser	For a list of web browsers, see “Web browser support” on page 29 .

Required software	Supported version
JDK	<ul style="list-style-type: none"> ● JBoss on all platforms: Sun JDK 1.5.0_11 (or later version of the 1.5.0 release) available at Sun Developer Network. ● WebLogic on all platforms except Solaris: BEA JRockit SDK 1.5.0_06 or later updates of 1.5.0 ● WebLogic on Solaris: Sun JDK 1.5.0_04. Requires the Daylight Savings Time (TZ) update, available at Sun Updater Tool, to be run in North America. ● WebSphere on all platforms except Solaris: IBM JDK 1.5.0 SR4 (available for download from IBM). ● WebSphere on Solaris: Sun JDK 1.5.0_09 <p>For more information about JDKs, see “LDAP configuration” on page 36.</p> <p>To use AES 256 encryption, 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>
Database	<ul style="list-style-type: none"> ● IBM DB2 8.2 or IBM DB2 9.1 <p>Note: If you are using WebLogic 10 MP1 with DB2 9.1 you need to contact BEA to obtain a custom patch that fixes a DB2 9.1 driver issue; otherwise, the JDBC connection does not function properly.</p> <ul style="list-style-type: none"> ● Microsoft SQL Server 2005 SP2 (Standard and Enterprise Edition) ● MySQL 5.0.18 (InnoDB Engine) ● Oracle 9i or 10g (Standard and Enterprise Editions) <p>Note: LiveCycle ES is compliant with the database vendors' compatibility statements. (See the vendors' websites.)</p>
Database driver	<ul style="list-style-type: none"> ● IBM DB2 9.1 driver is required for all supported versions of DB2: db2jcc.jar and db2jcc_license_cu.jar ● SQL Server 2005 JDBC 1.1: sqljdbc.jar ● MySQL: mysql-connector-java-3.1.12-bin.jar (3.1.12) ● Oracle 9i and 10g: ojdbc14.jar, Release 2 (10.2.0.2 Thin or later update of 10.2) <p>Note: All LiveCycle ES Business Activity Monitoring metadata databases use the drivers listed above.</p>

Required software	Supported version
LDAP server	<ul style="list-style-type: none">● Sun ONE 5.1● Sun ONE 5.2● Microsoft Active Directory 2000● Microsoft Active Directory 2003● 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● Microsoft Exchange 2003● Microsoft Exchange 2007● Lotus Notes/Domino 6/7● SendMail (included with Red Hat 4/5 and SUSE 9)● Novel GroupWise 6/7
LiveCycle Data Services ES	<ul style="list-style-type: none">● Adobe Flex® Builder™ 2.0.1 Hot Fix 2 or later or <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 ES● Customizing form guides in LiveCycle Designer ES● Customizing LiveCycle Workspace ES● Creating Flex applications for LiveCycle Workspace ES● Calling LiveCycle ES APIs using Flex

Required software	Supported version
LiveCycle ES connector	<p>Connector for EMC® Documentum®:</p> <ul style="list-style-type: none"> ● EMC Documentum Content Server 5.3 (and later service packs) ● EMC Documentum Content Server 6.0 (and later service packs) <p>In addition, on your LiveCycle ES 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 3.5.x ● IBM FileNet P8 Content Engine 4.0.x ● IBM FileNet P8 Process Engine 3.5.x ● IBM FileNet P8 Process Engine 4.0.x <p>Note: The version of Process Engine should correspond to the version of your Content Engine (for example, Process Engine 3.5.x for Content Engine 3.5.x).</p> <p>In addition, on your LiveCycle ES server, install the version of IBM FileNet P8 Content Java API that corresponds to your version of Content Engine (for example, 3.5.x Content Java API or 4.0.x Content Engine Java API). 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 8.3 (and later fix packs) <p>In addition, on your LiveCycle ES 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 ES server) ● Information Integrator for Content (II4C) available from IBM <p>Note: IBM Content Manager running on Oracle database is not supported.</p>

Required software	Supported version
PDF client	<ul style="list-style-type: none"> ● Adobe Acrobat® Professional and Acrobat Pro Extended, versions 7.0 to 9.0 (for securing documents with policies and opening policy-protected documents) ● Adobe Reader, versions 7.0 to 9.0 (for opening policy-protected documents) ● Acrobat Reader 6.0: The following limitations apply: <ul style="list-style-type: none"> ● LiveCycle Reader Extensions ES supports only UB3, which works down to Adobe Reader 7.0. Adobe Reader 6.0 supports UB2; LiveCycle ES does not generate files from UB2. ● Files from previous versions of Reader Extensions ES work for submitting forms online or by email. ● Designer ES, version 8.2, can create PDF forms that support Adobe Reader 6.0.2 and later. ● Workspace ES requires Adobe Reader 7.0.5 (and later), 8.0 (and later), 8.1 (and later), and 9.0 (and later). ● Apple® QuickTime 7 Player or Pro (for converting embedded video to PDF multimedia)

Minimum database user permissions

This section outlines the minimum database permissions required by the user performing the LiveCycle ES database initialization and by runtime users.

Database	Initialization permissions	Runtime permissions
Oracle	CREATE SESSION CREATE CLUSTER CREATE TABLE CREATE VIEW CREATE SEQUENCE UNLIMITED TABLE SPACE	CREATE SESSION CREATE CLUSTER UNLIMITED TABLE SPACE
MySQL	CREATE DROP REFERENCES ALTER INDEX CREATE_NEW SELECT INSERT UPDATE DELETE	SELECT INSERT UPDATE DELETE

Database	Initialization permissions	Runtime permissions
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	Refer to “DB2 user account” on page 52 for a complete description.	Refer to “DB2 user account” on page 52 for a complete description.

Web browser support

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

End-user user interface

End-user components include these solution components:

- LiveCycle Workspace ES (Flash Player required)
 - Adobe Flash Player 9.0.115.0 or later is required for Workspace ES or for using form guides in Workspace ES.
- LiveCycle Reader Extensions ES (Flash Player required)
- LiveCycle Rights Management ES (browser only)
- LiveCycle PDF Generator ES and LiveCycle PDF Generator 3D ES (browser only)

Operating system	Flash Player	Supported browser
Microsoft Windows Vista™	Flash Player 9*	Microsoft Internet Explorer 7 or later ⁽¹⁾ Firefox 2.0.0.1 or later ⁽¹⁾
Windows 2000	Flash Player 9*	Internet Explorer 6 or later ⁽¹⁾ Firefox 2.0.0.1 or later ⁽¹⁾
Windows XP	Flash Player 9*	Internet Explorer 6 or later ⁽¹⁾ Firefox 2.0.0.1 or later ⁽¹⁾
Windows Server 2003	Flash Player 9*	Internet Explorer 6 or later ⁽¹⁾ Firefox 2.0.0.1 or later ⁽¹⁾

Operating system	Flash Player	Supported browser
OS X v 10.4.x (PowerPC)	Flash Player 9*	Firefox 2.0.0.1 or later ⁽¹⁾ Safari 2.x or later (Workspace ES requires version 3.0.3 or later)
OS X v 10.4.x (Linux)	Flash Player 9*	Firefox 2.0.0.1 or later ⁽¹⁾ Safari 2.x or later (Workspace ES requires version 3.0.3 or later)

⁽¹⁾“or later” includes major revisions. For example, Internet Explorer 6.0 or later also covers Internet Explorer 7.0.

Administrator user interface

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

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

⁽¹⁾“or later” includes major revisions. For example, Microsoft Internet Explorer 6.0 or later also covers Microsoft Internet Explorer 7.0

Additional requirements for Linux or UNIX operating systems

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

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

Additional requirements for PDF Generator ES and PDF Generator 3D ES

This section applies only if your LiveCycle ES installation includes LiveCycle PDF Generator ES or LiveCycle PDF Generator 3D ES. If you are not installing PDF Generator ES, proceed to [“Additional requirements for Connector for IBM Content Manager” on page 34.](#)

User account on Windows

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

- Installing Microsoft Office
- Installing PDF Generator ES or PDF Generator 3D ES
- Installing Acrobat 9.0 Professional Extended for PDF Generator ES or PDF Generator 3D ES
- Running the application server process

Using 64-bit Windows with PDF Generator ES and PDF Generator 3D ES

If you are using a 64-bit Windows operating system with PDF Generator ES or PDF Generator 3D ES, ensure that a 32-bit Java 5 JDK is installed in addition to the 64-bit one the application server uses. You must set the following environment variable.

`JAVA_HOME_32` (required to point to a 32-bit JDK on Windows 64-bit systems)

► 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\Java\jdk1.5.0_14
```

Native file conversion software installation

Before you install PDF Generator ES or PDF Generator 3D ES, install the software that supports the file formats for which PDF conversion support is required and manually activate the licenses for the software. You must activate one license on your LiveCycle ES server for each native application that PDF Generator ES or PDF Generator 3D ES supports. Refer to the individual licensing agreement for each native application that your LiveCycle ES deployment will support, and ensure that your LiveCycle ES deployment meets the licensing requirements specified. Typically, each LiveCycle ES user who will use native application support must also have an activated license on their own computer for the native application.

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

- Microsoft Office 2000, XP, or 2003 (DOC, XLS, PPT, RTF, TXT)
- Microsoft Office 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)
- Corel WordPerfect 12 (WPD)
- Adobe Photoshop® CS2
- Adobe FrameMaker® 7.2, 8 (FM)
- Adobe PageMaker® 7.0 (PMD, PM6, P65, PM)
- OpenOffice 2.4 (ODT, ODP, ODS, ODG, ODF, SXW, SXI, SXC, SXD)

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)

Acrobat

PDF Generator ES or PDF Generator 3D ES requires that Acrobat 9 Pro Extended is installed. You must install Acrobat before you run the LiveCycle ES installer. Acrobat Pro Extended is provided with the LiveCycle ES media or as an option for ESD downloads.

You also need to update the `ACROBAT_PATH` environment variable and restart the application server.

To use AES 256 encryption, you must obtain and install the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy files from [Java SE Downloads](#).

Note: These requirements are optional and required only if you need to use Advanced Encryption Standard (AES) 256 with Acrobat 9 Pro Extended.

QuickTime 7

PDF Generator ES or PDF Generator 3D ES 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.

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, WordPerfect, and Acrobat.

The names of these environment variables are listed here:

- `FrameMaker_PATH`
- `Notepad_PATH`
- `OpenOffice_PATH`
- `PageMaker_PATH`
- `Photoshop_PATH`
- `WordPerfect_PATH`

These environment variables are optional and need to be set only if you plan to use the corresponding application to convert PDF files through PDF Generator ES or PDF Generator 3D ES. 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 `Photoshop_PATH` may contain the value `D:\Program Files\Adobe\Adobe Photoshop CS3\Photoshop.exe`. However, `OpenOffice_PATH` is different from others. This variable must be set to the OpenOffice installation folder (instead of the path to the executable). A typical value of `OpenOffice_PATH` on Windows would be `C:\Program Files\OpenOffice.org 2.0\`.

Paths for Microsoft Office applications such as Word, PowerPoint, Excel, Visio, and Project or for AutoCAD are not required. The Generate PDF service starts these applications automatically if they are installed on the server.

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

1. Select **Start > Control Panel > System**.
2. Click the **Advanced** tab.
3. Click **Environment Variables** and, under System Variables, click **New**.
4. Enter the environment variable name you need to set (for example, `Photoshop_PATH`). This directory is the directory that contains the executable file. For example, type the following code:

```
D:\Program Files\Adobe\Adobe Photoshop CS3\Photoshop.exe
```

IPP Client installation

PDF Generator ES includes an Internet Printing Protocol (IPP) client installer for installation of the PDF Generator ES Internet printer. After the installation is completed, a PDF Generator ES 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 IPP client, see *Installing and Deploying LiveCycle ES* document for your application server.

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

Service Control Manager command line tool

Before you complete an automatic installation of PDF Generator ES on Windows, ensure that the Service Control Manager command line tool, `sc.exe`, is installed in the Windows environment. Some Windows servers do not have this software preinstalled. By default, the `sc.exe` file is installed in the `C:\Windows\system32` directory. Most OS installations 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.

Headless mode configuration

If you are running PDF Generator ES 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.

Additional requirements for Connector for IBM Content Manager

These requirements are optional and required only if you are installing LiveCycle ES Connector for IBM Content Manager.

LiveCycle ES 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)

The following sections describe how to configure these applications. Specific application server configurations are described in the “Post-Deployment Activities” chapter in the *Installing and Deploying LiveCycle ES* 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 34.](#)
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 Linux or UNIX).
 - Run the cmbenv81.bat (Windows) or cmbenv81.sh (UNIX/Linux) 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 Linux or UNIX).
2. Run the cmbenv81.bat (Windows) or cmbenv81.sh (UNIX/Linux) 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.

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 8.0, LDAP configuration settings are automatically migrated.

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 24](#). Make note of the LDAP administrator name and password to use during the LiveCycle ES configuration process. Configure LiveCycle ES to connect with the LDAP database after you install and deploy your LiveCycle ES services. This configuration is done by using the User Manager service. See the *Installing and Deploying LiveCycle ES* document for your application server.

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 ES product components. The lifetime of long-lived files is intended to span multiple restarts of a LiveCycle ES system, and can span for 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 ES deployments. If some or all long-lived documents are lost or corrupted, the LiveCycle ES 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. Therefore, it is important that you consider the reliability of the file system that hosts the GDS directory.

Planning and creating the global document storage directory

You should plan the location, size, and security aspects of your GDS directory in advance (see ["Location of the global document storage directory" on page 36](#), ["Sizing factors for the global document storage directory" on page 37](#), and ["Securing the global document storage directory" on page 37](#)). You should also plan the backup strategy and practices that you will implement (see ["Backing up the global document storage directory" on page 37](#)).

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

Location of the global document storage directory

You configure the location of your GDS directory with LiveCycle Configuration Manager after you install LiveCycle ES. The GDS directory you specify should be highly available and should have low access time to enhance performance.

If you leave the location setting empty during installation, the location defaults to a directory under the application server installation:

- (JBoss) `[appserver root]/server/<server>/all/svcnative/DocumentStorage`
- (WebLogic) `[appserverdomain]/<domain>/adobe/<server>/DocumentStorage`
- (WebSphere) `[appserver root]/installedApps/adobe/<server>/DocumentStorage`

You can change the GDS directory location after completing the installation (see [Administering LiveCycle ES](#)), but this is not a trivial procedure. Plan an appropriate location for the GDS directory that will be long-lived.

Caution: Component 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:\.

Sizing factors for the global document storage directory

The size of the shared directory depends on expected LiveCycle ES 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 ES processes. Processing high volumes of documents requires a larger GDS directory.
- The typical size of documents that LiveCycle ES processes. Processing large documents requires a larger shared GDS directory.
- The complexity of documents that LiveCycle ES processes. Processing complex documents (that is, documents that are processed by multiple LiveCycle ES services) requires a larger GDS directory.

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 ES 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 ES installation inoperative.

Backing up the global document storage directory

The GDS directory should be backed up to allow administrators to restore LiveCycle ES in case of failure. (See [Administering LiveCycle ES.](#))

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

LiveCycle ES Business Activity Monitoring requirements

The following recommendations are required only if you are installing the LiveCycle ES Business Activity Monitoring solution component.

Basic requirements

This section describes the basic requirements for installing, deploying, and running Business Activity Monitoring. It covers the following requirements:

- Dedicated JVM
- Client memory
- Web browser
- Mail server

Dedicated JVM

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

Client memory

The client computer that accesses Business Activity Monitoring should have a minimum of 512 MB RAM (1.0 GB is recommended).

Web browser

Business Activity Monitoring is tested for Microsoft Internet Explorer 6.0 (with patch 828750) or later browser for accessing BAM Workbench.

Note: You must also install Adobe Flash, version 8.0 or later.

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

Mail Server

Business Activity Monitoring requires a running Simple Mail Transfer Protocol (SMTP) email server for delivering email notifications. The server is external to Business Activity Monitoring and is managed by your email system administrator. Contact that administrator to set up an account specifically for Business Activity Monitoring. 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.

Creating the Business Activity Monitoring metadata database

You must create a Business Activity Monitoring metadata database to store the definitions of the process metrics that BAM Server monitors, as well as the details of any alerts and object run-time data that need to be persisted to disk.

Because Business Activity Monitoring metadata can grow quite large, you must 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.

The following three tables provide information about which databases are supported for BAM Server metadata databases for JBoss, Weblogic, and WebSphere.

You need to create a user account that BAM Server can use to connect to the BAM Server metadata database. For database permissions, see [“Minimum database user permissions” on page 28](#).

JBoss requirements

The following tables describe the database and system requirements for installing and deploying Business Activity Monitoring on JBoss Application Server.

Disk space: 115 MB

Metadata database:

Database	JDBC driver
DB2 8.2	<ul style="list-style-type: none">• db2jcc.jar• db2jcc_license_cu.jar
Oracle 9.2 or later	<ul style="list-style-type: none">• Oracle thin driver (ojdbc14.jar)• Oracle thick driver (OCI) <p>Note: Use the Oracle 10G JDBC driver for both configurations.</p>
MS SQL Server 2000	<ul style="list-style-type: none">• sqljdbc.jar <p>OR</p> <ul style="list-style-type: none">• mssqlserver.jar• msbase.jar• /msutil.jar <p>Note: The sqljdbc.jar driver is backward-compatible to version 2000.</p>

Database	JDBC driver
MS SQL Server 2005	<ul style="list-style-type: none">• sqljdbc.jar <p>Note: Ensure that the version 2000 drivers are not in the classpath.</p>
MySQL 5.0 or later	<ul style="list-style-type: none">• mysql-connector-java-[version]-bin.jar

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

Note: The sqljdbc.jar driver is backward-compatible to version 2000. However, if you use the 2005 driver (sqljdbc.jar), ensure that the version 2000 drivers are not in the classpath.

WebLogic requirements

The following table describes the database and system requirements for installing and deploying Business Activity Monitoring on WebLogic Server.

Disk space: 750 MB

Metadata database:

Database	JDBC driver
IBM DB2	db2jcc.jar db2jcc_license_cu.jar
SQL Server 2000 plus SP3 or later	Embedded BEA drivers for MS SQL Server
SQL Server 2005	Embedded BEA drivers for MS SQL Server
MySQL 5.0 or later	mysql-connector-java-[version]-bin.jar
Oracle 9.2 or later	Embedded BEA drivers for Oracle thin or thick drivers

WebSphere requirements

The following table describes the database and system requirements for installing and deploying Business Activity Monitoring on WebSphere Application Server.

Metadata database:

Database	JDBC driver
IBM DB2	<ul style="list-style-type: none">• db2jcc.jar• db2jcc_license_cu.jar
MS SQL Server 2000	<ul style="list-style-type: none">• mssqlserver.jar• msbase.jar• msutil.jar
MS SQLServer 2005 (WebSphere 6.1.x only)	Use the native WebSphere drivers.
MySQL 5.0 or later	mysql-connector-java-[version]-bin.jar
Oracle	ojdbc14.jar (from version 10g) Note: You can use either the OCI (thick) or thin driver. Use of the thick driver requires installation of the Oracle Client.

Installation considerations

This section includes considerations for administrators when installing LiveCycle ES.

Installing from network drives

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

Using LiveCycle ES 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
```

Manual use of Acrobat restricted

If you installed the PDF Generator ES or PDF Generator 3D ES 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.

Restricted use of unlicensed Process Management ES functionality

When you install LiveCycle ES in a production environment without licensing Process Management ES, some core Process Management ES functionality is available but not the following notable exceptions:

- Run a long-lived process
- Use a “User Task” in a process
- Access Workspace ES

If you require any of these features, you must purchase the appropriate license.

LiveCycle ES IPv6 support

LiveCycle ES 8.2 introduces IPv6 support. The default configurations defined in the installation documentation for LiveCycle ES set IPv4 as the default IP protocol because this protocol has the most compatibility with third-party infrastructure. It is recommended that IPv6 is not enabled unless there is a specific requirement for it in your deployment. The number of supported platform configurations is reduced when enabling IPv6 support with LiveCycle ES. It is recommended that you verify that all third-party software, hardware, and networks that you plan to use have IPv6 support before you attempt to enable IPv6.

Caution: When installing on a pure IPv6 WebSphere Application Server environment (that is, no IPv4 addresses at all), you must configure and deploy the LiveCycle ES EAR files manually. The tasks required are as follows:

- Run LiveCycle Configuration Manager to configure LiveCycle ES
- Manually configure WebSphere
- Manually deploy the LiveCycle ES EAR files

Once these tasks are done, run the following tasks on LiveCycle Configuration Manager:

- Initialize the LiveCycle ES database
- Deploy the LiveCycle ES components
- Validate the LiveCycle ES component deployment

Processes with document form variables and digital signatures

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

Here are possible solutions for avoiding this problem; choose the method that is most appropriate for your LiveCycle ES environment:

Solution 1: Complete all processes that use a form document variable before you upgrade or move to the remote server. You might choose 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.

3

LiveCycle ES Credentials and Certificates

This chapter describes how to do the following tasks:

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

Obtaining the Reader Extensions ES Rights credential

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

The Rights credential extends the usage rights of each PDF file that Reader Extensions ES 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 ES. 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 (currently until 2028) 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 ES, 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 ES, the production Rights credential is delivered by Electronic Software Download (ESD). A production Rights credential is unique to your organization and can enable the specific usage rights that you require.

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

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

Obtaining digital certificates for use with Digital Signatures ES

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

Digital certificates are obtained from a Certificate Authority (CA) and sent to you by email or over the web as a certificate file. This certificate file contains the public keys (also called *certificates*) and references to private keys (also called *credentials*) 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 (UNIX) 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 ES by using Trust Store Management, which is accessible through the web-based LiveCycle Administration Console. (See [LiveCycle ES Trust Store Management Help](#).)

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

The following file types are supported:

Certificates: DER-encoded X509v3 and base64-encoded certificate (.cer) files. Certificates 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 ES 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.

4

Creating the LiveCycle ES Database

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

The database will contain these elements:

- LiveCycle ES solution component services
- LiveCycle ES run-time configurations
- LiveCycle ES process data
- Customer process definitions and templates
- Application server managed data, such as JMS tables

Before you create the database, you must ensure that you read the pre-installation requirements and have the required software installed. (See [“System Requirements for Installing LiveCycle ES” on page 17.](#))

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

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

Database configuration requirements

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

Creating an Oracle database

If you prefer not to use the default database that was created when you installed Oracle 9i or Oracle 10g, create a new database by using the Database Configuration Assistant tool. LiveCycle ES supports Oracle 9i and Oracle 10g with WebLogic 9.2, WebLogic 10.1 MP1, WebSphere 6.1.0.7, JBoss 4.0.3 SP1, and JBoss 4.2.0.

These requirements are for creating your Oracle database:

- The initial database size must be a minimum of 1175 MB (You must increase this initial size if you are deploying LiveCycle ES Content Services).
- The database must support UTF-8 characters.
- The Database Character Set is Unicode (AL32UTF8), and the National Character Set is AL16UTF16 (Unicode UTF-16 universal character set).
- The database setting NLS_LENGTH_SEMANTICS is set to BYTE. 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.

You must also 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

For deployments on Linux, 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 port number (default is 1521)

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

Creating a SQL Server database

You can create a SQL Server database that LiveCycle ES will use to store run-time and configuration data. For information about creating a SQL Server database, refer to the SQL Server documentation. LiveCycle ES supports SQL Server 2005 SP1 (English and Japanese) with JBoss 4.0.3 SP1 and JBoss 4.2.0, BEA WebLogic 9.2, BEA WebLogic 10.1 MP1, and IBM WebSphere 6.1.0.7.

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

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

Installing the SQL Server database driver

If you have not done so already, download the SQL Server 2005 JDBC Driver 1.1 from the [Microsoft Download Center](#) location on the Microsoft website.

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.

Setting up SQL Server for LiveCycle ES

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

Increasing memory

The default SQL Server settings do not aggressively allocate memory. This situation significantly 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 ES database.
2. Right-click the database connection and select **Properties**.
3. Select the **Memory** page and enter a size in the **Minimum Server Memory (in MB)** box that is equal to the size of the free memory on the server.
4. Restart the SQL Server database.

Setting processor priority

On dedicated database servers, which are recommended for production installations of LiveCycle ES, 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 ES database.
2. Right-click the database connection and select **Properties**.
3. Select the **Processors** page and select **Boost SQL Server Priority**.
4. Restart the SQL Server database.

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 ES database.
2. Right-click the database connection and select **Properties**.
3. Select the **Database Settings** page and type 5 in the **Recovery Interval (Minutes)** box.
4. Restart the SQL Server database.

Integrated security

If you are using SQL Server 2005 integrated security, you must set your SQL Server 2005 database to Mixed Mode for WebSphere or WebLogic. For JBoss (Windows), you can set your SQL Server 2005 database to either Mixed Mode or Windows Authentication Mode. If you set it to Windows Authentication Mode for JBoss (Windows), follow the steps in ["To configure integrated security on Windows:" on page 61](#).

Note: This is an optional configuration.

Sizing your SQL Server database

The default database sizes that SQL Server provides are too small for LiveCycle ES. Even if the database is set to auto-grow, unintended 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 ES database to ensure that it does not lose its restricted space or the space available on the disks where the database resides.

Creating the LiveCycle ES database user, schema, and login

You are now ready to create the LiveCycle ES 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 ES. Any other collation may cause your database initialization to fail. The collation for your LiveCycle ES database instance can be different from the collation used when creating the SQL Server database.

► To create the LiveCycle ES database:

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

2. 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. 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 49](#).

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 49](#).

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

Creating the LiveCycle ES user in SQL Server

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

► To create the LiveCycle ES user:

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

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

3. 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**.

Associating the LiveCycle ES user with the database

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

► To associate the LiveCycle ES user with the database:

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

Setting the isolation level for the LiveCycle ES database

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

► To set the isolation level:

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

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

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

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

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

Creating a DB2 database

Create a DB2 database by running the script provided in this section. The script is tuned for a system that will use 1 GB of memory for the database. If your system has less memory dedicated for the database, see the DB2 documentation for details about configuring your system settings. LiveCycle ES supports DB2 with WebSphere 6.1.0.7, JBoss 4.0.3 SP1, JBoss 4.2.0, WebLogic 9.2, and WebLogic 10.1.

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

DB2 user account

LiveCycle ES 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 Linux and UNIX, the user name must not exceed 8 characters; on Windows, it must not exceed 12 characters.

► To create a DB2 database:

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

Note: The following text contains formatting characters for line breaks. If you copy this text to a location outside this document, remove the formatting characters when you paste it to the new location.

```
create database dbname using codeset utf-8 territory default;
connect to dbname;
CREATE BUFFERPOOL BP8K SIZE 50000 PAGESIZE 8192 NOT EXTENDED STORAGE;
connect reset;
connect to dbname;
CREATE TEMPORARY TABLESPACE DBNAME_TEMP_8K IN DATABASE PARTITION GROUP
IBMTEMPGROUP PAGESIZE 8192 MANAGED BY SYSTEM USING
('DB2_root\DBNAME_TEMP') EXTENTSIZE 32 PREFETCHSIZE 16 BUFFERPOOL BP8K;
CREATE REGULAR TABLESPACE DBNAME_DATA_8K IN DATABASE PARTITION GROUP
IBMDEFAULTGROUP PAGESIZE 8192 MANAGED BY DATABASE USING
(FILE'DB2_root\DBNAME_DATA'64000) EXTENTSIZE 16 PREFETCHSIZE 16
BUFFERPOOL BP8K;
commit work;
connect reset;
connect to dbname;
alter bufferpool ibmdefaultbp immediate size 96000;
```

```
alter bufferpool bp8k immediate size 32000;
commit work;
connect reset;
update db cfg for dbname using dbheap 4000;
update db cfg for dbname using logbufsz 2048;
update db cfg for dbname using locklist 2000;
update db cfg for dbname using chngpgs_thresh 40;
update db cfg for dbname using logfilsiz 4000;
deactivate database dbname;
activate database dbname;
```

2. (Content Services ES only) If you are using Content Services ES with any of the other solution components, use the following script:

Note: The following text contains formatting characters for line breaks. If you copy this text to a location outside this document, remove the formatting characters when you paste it to the new location.

```
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;
```

3. Make the following changes to the script:

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

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

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

► **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 **Grant**.
5. Click **OK**.

Additional DB2 requirements for LiveCycle Content Services ES

The script described in step 2 in “To create a DB2 database:” on page 52 creates an additional DB2 database page size and sets it to 32 KB. If LiveCycle Content Services ES is not being deployed, the default DB2 database page size of 8 KB is acceptable.

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 ES for JBoss Using Turnkey](#).

Use the MySQL tools to create a MySQL 5 database for use with LiveCycle ES and a MySQL user account that the application server can use to connect to the database. You also must modify the MySQL database server configuration. For information about creating the database and user account, see the MySQL documentation. LiveCycle ES supports MySQL 5 with JBoss 4.03 SP1, JBoss 4.2.0, and WebLogic 10.1.

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

Note: 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

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 ES 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.

MySQL user account

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

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

MySQL initial server configuration recommendations

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

- `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
- max_allowed_packet=25M
- 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 4.0.3SP1 (Windows only)
- Adobe preconfigured JBoss with JBoss 4.0.3SP1 or JBoss 4.2.0
- Manual JBoss configuration with JBoss 4.0.3SP1 or JBoss 4.2.0 (See [“Appendix - Manually Configuring JBoss” on page 89.](#))

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

- You are using a Linux or Solaris operating systems.
- You are using DB2, Oracle, SQL Server, or MySQL as your database.
- You need advanced control of the JBoss configurations.

LiveCycle ES runs on JBoss on Windows Server 2003 (Enterprise Edition or Standard Edition), Red Hat Linux ES/AS 4.0 or 5.0, Solaris 9/10, and SUSE Linux ES 9 platforms. (See [“Third-party infrastructure support” on page 17.](#))

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

This chapter uses the following conventions.

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

Installing the J2SE SDK

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

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

► **To set the `JAVA_HOME` environment variable (Windows):**

1. Select **Start > Control Panel > System**.
2. Click the **Advanced** tab 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 SDK as the value. This directory contains the `/bin` subdirectory. For example, type the following path:

```
C:\Program Files\Java\jdk1.5.0_11
```

5. (Optional) To verify your `JAVA_HOME` environment variable, open a command prompt and run the following command:

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

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

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

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

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

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

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

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

UNIX logon scripts

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

Adobe-preconfigured JBoss

The Adobe-preconfigured JBoss software is delivered as part of the LiveCycle ES 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 89](#).

Adobe downloads the standard JBoss 4.0.3 SP1 and JBoss 4.2.0 archives from the JBoss website and configures them with the options required to run LiveCycle ES in a stand-alone configuration. You can review these configurations in the next section.

The preconfigured JBoss software is located on the installation media or electronic software distribution (ESD) download in the `/third_party` directory. After you determine the version of JBoss you want to use, copy the JBoss directory and its contents to the location where you intend to install JBoss. This JBoss installation has all the configuration steps completed except for the steps detailed in [“Configuring MySQL for Adobe-preconfigured JBoss” on page 53](#), 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.

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.

► 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 all -b server_IP_Address`
 - (Linux) `./run.sh -c all -b server_IP_Address`
 - (Solaris) `./run.sh -c all -b server_IP_Address (`

Note: Adobe pre-configured JBoss must be bound to a specific IP address if it is to be accessed remotely.

► 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`
 - (Linux) `./shutdown.sh -S`
 - (Solaris) `./shutdown.sh -S`

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 probably crash. JBoss also has a few configuration files that are formatted as .property files. These files must be UNIX text files on Linux or Solaris. It is important to take care if they are edited on Windows environments at any time.

JBoss 4.2.0 start up configuration for 64-bit support

In order to support JBoss Application Server 4.2.0 running on a 64-bit Windows operating system, you must modify the run.bat file.

► To modify the run.bat file (JBoss 4.2.0 with Windows, 64-bit only)

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

2. Locate the following line:

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

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. Save and close the run.bat file.

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.

► To modify the run.conf file (JBoss 4.2.0 with Solaris 10, Redhat 5.0, 64-bit only)

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

2. Modify the line that immediately follows `Setup JBoss specific properties to configure encoding to UTF-8 and set security manager`:

```
JAVA_OPTS="$JAVA_OPTS -d64 -Dadobeidp.serverName=server1  
-Dfile.encoding=utf8  
-Dprogram.name=$PROGNAME -Djava.security.manager  
-Djava.security.policy=../server/all/conf/server.policy  
-Djava.security.debug=failure"
```

3. Save and close the run.conf file.

Configuring the LiveCycle ES database connectivity

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

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

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

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

Configuring MySQL for Adobe-preconfigured JBoss

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

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

If you are using a manually installed JBoss, see [“Configuring MySQL for manually installed JBoss” on page 99](#).

Configuring the MySQL data source

Before configuring the MySQL data source, you must have already created the database on MySQL. (See [“Creating a MySQL database” on page 54](#).)

► To create the MySQL data source file:

1. Delete the `adobe-ds.xml` file from the `[appserver root]/server/all/deploy` directory.
2. Copy the `adobe-ds-jboss-mysql.xml` file from the `[DVD_root]/third_party/datasources` directory on the installation DVD to the `[appserver root]/server/all/deploy` directory.
3. Open the `adobe-ds-jboss-mysql.xml` file in a text editor and locate this line:

```
<connection-url>jdbc:mysql://localhost:3306/adobe</connection-url>
```
4. Replace the following values 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`.
 - **3306**: The port used to access the database. The default port is `3306`.
 - **adobe**: The name of the database that stores the LiveCycle ES data. Replace the default value, `adobe`, with your database name.
5. In the lines that follow the `<connection-url>` settings, locate the `user-name` and `password` settings and replace the default values of `adobe` and `adobe` with the user name and password that the application server uses to access your 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>
```
 - For `com.celequest.metadata.metaDatasource` (Business Activity Monitoring only):

```
<min-pool-size>1</min-pool-size>  
<max-pool-size>20</max-pool-size>
```
7. Save the file as `adobe-ds.xml` in the `[appserver root]/server/all/deploy` directory.

Setting MySQL as the JMS data source

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

Caution: If the JMS data source is different from IDP_DS, such as a separate database or MQSeries, some JMS messages may be lost upon recovery from a system failure. This loss may include jobs that are waiting to be processed or processes that previously failed. To avoid this issue, configure the JMS data source to use the same database instance as IDP_DS.

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

► To set MySQL as the JMS data source:

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

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

- **yourHostname:** Replace this value with the name of the server hosting the database.
- **yourPort:** Replace this value with the port number for your database server.
- **yourDatabaseName:** Replace this value with the database that will connect with LiveCycle ES.
- **yourUserName:** Replace this value with the user name that will access the database.
- **yourPassword:** Replace this value with the database user's password.

2. Save and close the file.

3. Open the `[appserver root]/server/all/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">database_name</module-option>
      <module-option name="userName">database_username</module-option>
      <module-option name="password">database_password</module-option>
      <module-option
        name="managedConnectionFactoryName">jboss.jca:service=LocalTxCM,
        name=Default DS </module-option>
      </login-module>
    </authentication>
  </application-policy>
```

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

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

5. Save and close the file.
6. Restart JBoss.

Configuring Oracle for Adobe-preconfigured JBoss 4.0.3 SP1 and 4.2.0

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

- Obtain and copy the Oracle JDBC driver to the instance of JBoss where you will deploy LiveCycle ES.
- Create a data source file and deploy it to the instance of JBoss where you will deploy LiveCycle ES.
- Configure JBoss to use Oracle for JMS.

Configuring the Oracle data source

This section applies to both JBoss 4.0.3 SP1 and JBoss 4.2.0. After you configure the data source, you must complete one of these sections [“Setting Oracle as the JMS data source for JBoss 4.0.3 SP1” on page 56](#) or [“Setting Oracle as the JMS data source for JBoss 4.2.0” on page 58](#).

Before you configure the Oracle data source, you must have the database created on Oracle. (See [“Creating an Oracle database” on page 46](#).)

► To install the Oracle 10g database driver:

1. You can access the Oracle 10g driver file from the `[LiveCycleES root]/lib/db/oracle` directory. You can also download the Oracle 10g Release 2 (10.2.0.2 Thin) driver from [JDBC Driver Downloads](#).
2. Copy the `ojdbc14.jar` driver file to the `[appserver root]/server/all/lib` directory.

► To create the Oracle data source file:

1. Delete the `adobe-ds.xml` file from the `[appserver root]/server/all/deploy` directory.
2. Copy the `adobe-ds-jboss-oracle.xml` file from the `[DVD_root]/third_party/datasources` directory on the installation DVD to the `[appserver root]/server/all/deploy` directory.
3. Open the `adobe-ds-jboss-oracle.xml` file in a text editor and locate this line:

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

4. 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 name of the database that stores the LiveCycle ES data. Replace the default value, `adobe`, with your database name.
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>
```

- For com.celequest.metadata.metaDatasource (Business Activity Monitoring only):

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

6. Save the file as adobe-ds.xml in the `[appserver root]/server/all/deploy` directory.
7. Restart JBoss.

Setting Oracle as the JMS data source for JBoss 4.0.3 SP1

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

Caution: If the JMS data source is different from IDP_DS, such as a separate database or MQSeries, some JMS messages may be lost upon recovery from a system failure. This loss may include jobs that are waiting to be processed or processes that previously failed. To avoid this issue, configure the JMS data source to use the same database instance as IDP_DS.

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

► To set Oracle to be the JMS data source:

1. Go to the `[appserver root]/server/all/deploy` directory and delete the `mysql-ds.xml` file.
2. Go to the `[appserver root]/docs/examples/jca` directory and copy the `oracle-ds.xml` file 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:

```
<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 text in the lines above with values that are specific to your database:
 - **OracleDS**: Change this value to `DefaultDS`.
 - **youroraclehost**: Replace this value with the host name of your Oracle server.
 - **1521**: If Oracle is not using the default port, replace this value with the appropriate port number.
 - **yoursid**: Replace this value with your Oracle System Identifier.
 - Replace **x** with the user name to access the Oracle database and **y** with the password associated with the user. The application server uses these credentials to access the database.
5. Save and close the file.
6. Open the `[appserver root]/server/all/conf/standardjaws.xml` file and change the `<type-mapping>` element to Oracle 9i:

```
<type-mapping>Oracle9i</type-mapping>
```

7. Scroll down the file to the `<type-mapping-definition>` element. Change the `<name>` element to Oracle 9i:

```
<name>Oracle9i</name>
```

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

```
<datasource>java:/DefaultDS</datasource>  
<datasource-mapping>Oracle9i</datasource-mapping>  
<fk-constraint>true</fk-constraint>
```

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

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

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

database_name, **database_username** and **password** are the database values that the application server uses to access the database.

13. Save and close the file.
14. Delete the `mysql-jdbc2-service.xml` file from the `[appserver root]/server/all/deploy-hasingleton/jms` directory.
15. Copy the `oracle-jdbc2-service.xml` file from the `[appserver root]/docs/examples/jms` directory to the `[appserver root]/server/all/deploy-hasingleton/jms` directory.
16. Open the `[appserver root]/server/all/deploy-hasingleton/jms/oracle-jdbc2-service.xml` file in a text editor and change `OracleDS` to `DefaultDS`:
17. In the `oracle-jdbc2-service.xml` file in the `[appserver root]/server/all/deploy-hasingleton/jms` directory, change `OracleDS` to `DefaultDS`:

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

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

18. Save and close the file.
19. Rename the `[appserver root]/server/all/deploy-hasingleton/jms/mysql-db-jdbc-state-service.xml` file to `oracle-jdbc-state-service.xml`.
20. Restart JBoss.

Setting Oracle as the JMS data source for JBoss 4.2.0

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

Caution: If the JMS data source is different from IDP_DS, such as a separate database or MQSeries, some JMS messages may be lost upon recovery from a system failure. This loss may include jobs that are waiting to be processed or processes that previously failed. To avoid this issue, configure the JMS data source to use the same database instance as IDP_DS.

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

► To set Oracle to be the JMS data source:

1. Go to the `[appserver root]/server/all/deploy` directory and delete the `mysql-ds.xml` file.
2. Go to the `[appserver root]/docs/examples/jca` directory and copy the `oracle-ds.xml` to the `[appserver root]/server/all/deploy` directory.
3. Open the `[appserver root]/server/all/deploy/oracle-ds.xml` file in a text editor and modify the `<local-tx-datasource>` element.

```
<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 text in the above lines with values that are specific to your database:
 - **OracleDS**: Change this value to `DefaultDS`.
 - **youroraclehost**: Replace this value with the host name of your Oracle server.
 - **1521**: If Oracle is not using the default port, replace this value with the appropriate port number.
 - **yoursid**: Replace this value with your Oracle System Identifier.
 - Replace **x** with the user name to access the Oracle database and **y** with the password associated with the user. The application server uses these credentials to access the database.
5. Save and close the file.
6. Open the `[appserver root]/server/all/conf/standardjbosscmp-jdbc.xml` file and change the following elements (not necessarily on consecutive lines):

```
<datasource>java:/DefaultDS</datasource>
```

```
<datasource-mapping>Oracle9i</datasource-mapping>  
<fk-constraint>true</fk-constraint>
```

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

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

9. Replace the bold text in the above lines with values that are specific to your database:
database_username and **password** are the database values that the application server uses to access the database.

10. Save and close the file.
11. Delete the `mysql-jdbc2-service.xml` file from the `[appserver root]/server/all/deploy/jms` directory.
12. Copy the `oracle-jdbc2-service.xml` file from the `[appserver root]/docs/examples/jms` directory to the `[appserver root]/server/all/deploy/jms` directory.
13. In the `oracle-jdbc2-service.xml` file in the `[appserver root]/server/all/deploy/jms` directory, change OracleDS to DefaultDS:

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

14. Save and close the file.
15. Rename the `[appserver root]/server/all/deploy/jms/mysqlldb-jdbc-state-service.xml` file to `oracle-jdbc-state-service.xml`.
16. Restart JBoss.

Configuring SQL Server for Adobe-preconfigured JBoss

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

- Obtain and copy the SQL Server JDBC driver files to the instance of JBoss where you will deploy LiveCycle ES.
- Create a SQL Server data source file and deploy it to the instance of JBoss where you will deploy LiveCycle ES (for example, `[appserver_root]/server/all/deploy`).
- Configure JBoss to use SQL Server for JMS.

Configuring the SQL Server data source

Before you configure the SQL Server data source, you must have the LiveCycle ES database created on SQL Server. (See [“Creating a SQL Server database” on page 47.](#))

► To install the SQL database driver:

1. Obtain the SQL Server 2005 JDBC 1.1 database driver from the Microsoft website.
2. (Windows) Download the *.exe file and run 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_1.1/enu` directory to the `[appserver root]/server/all/lib` directory.

► To create the SQL Server data source file:

1. Delete the `adobe-ds.xml` file from the `[appserver root]/server/all/deploy` directory.
2. Copy the `adobe-ds-jboss-mssql.xml` file from the `[DVD_root]/third_party/datasources` directory to the `[appserver root]/server/all/deploy` directory.
3. Open the `adobe-ds-jboss-mssql.xml` file in a text editor and locate this line:

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

4. 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 ES data. You will need to update the default value, `adobe`, with your database name.
 - **database_username** and **password**: The user name and password that the application server uses to access the database. You will need to update the default values, `adobe` and `adobe`, with the credentials for your database.
5. Ensure that the `<driver-class>` is set as follows:

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

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>
```
- For com.celequest.metadata.metaDatasource (BAM only):

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

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

8. Restart JBoss.

This next procedure provides instructions on how to use integrated security to make a trusted connection with SQL Server.

Note: You must already have a user account created and associated with your Windows domain.

► **To map the Windows login to the LiveCycle ES database user:**

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

► **To configure integrated security on Windows:**

Note: You must run the JBoss service as the Windows user configured in ["To map the Windows login to the LiveCycle ES database user:" on page 61](#). 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\all\deploy`, to add `integratedSecurity=true` to the connection URL as shown in this example:

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

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

3. 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 the user account that you mapped in ["To map the Windows login to the LiveCycle ES database user:" on page 61](#). If you are running JBoss from the command line, you must be logged in as this user.

Configuring JMS with Microsoft SQL Server for JBoss 4.0.3 SP1

This section applies to Adobe-preconfigured JBoss 4.0.3 SP1.

Setting SQL Server as the JMS data source

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

Caution: If the JMS data source is different from IDP_DS (for example, a separate database or MQSeries), some JMS messages may be lost upon recovery from a system failure. This loss may include jobs that are waiting to be processed or processes that previously failed. To avoid this issue, configure the JMS data source to use the same database instance as IDP_DS.

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

► To set SQL Server to be the JMS data source:

1. Navigate to the `[appserver root]/server/all/deploy` directory and delete the `hsqldb-ds.xml` file. If you are using the Adobe-preconfigured JBoss from the DVD, delete the `mysql-ds.xml` file.
2. Install the JBossMQ patch file called `jbossmq.jar`. Replace the existing `jbossmq.jar` file in the `[appserver root]/server/all/lib` directory. You can get the updated `jbossmq.jar` file from the `thirdparty\jboss\patches` directory in the installation media.
3. Navigate to the `[appserver root]/docs/examples/jca` directory and copy the `mssql-ds.xml` file to the `[appserver root]/server/all/deploy` directory.
4. Open the `mssql-ds.xml` file in a text editor and change the `<jndi-name>` value from `MSSQLDS` to `DefaultDS`.
5. Modify the `<local-tx-datasource>` element to use your SQL Server connection settings:

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

- **your-hostname:** Replace this value with the name of the server hosting the SQL Server database
 - **yourdatabase-name:** Replace this value with the SQL Server database that will connect with LiveCycle ES.
 - **yourUserName:** Replace this value with the user name that will access the database.
 - **yourPassword:** Replace this value with the database user's password.
6. Save and close the file.

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

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

Note: The `<datasource-mapping>` element should always be `MS SQLSERVER2000`, even if you are using SQL Server 2005.

8. Save and close the file.

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

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

Note: The `<datasource-mapping>` element should always be `MS SQLSERVER2000`, even if you are using SQL Server 2005.

10. Save and close the file.

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

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

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

database_name, **database_username**, and **password** are the database values that the application server uses to access the database.

13. Save and close the file.

14. Delete the `[appserver root]/server/all/deploy-hasingleton/jms/hssql-jdbc2-service.xml` file or, If you are using the Adobe-preconfigured JBoss from the DVD, delete the `mysql-jdbc2-service.xml`.

15. Copy the `[appserver root]/docs/examples/jms/mssql-jdbc2-service.xml` file to `[appserver root]/server/all/deploy-hasingleton/jms`.

16. Open the `[appserver root]/server/all/deploy-hasingleton/jms/mssql-jdbc2-service.xml` file and change `mbean code` to reference `MSSQLJDBCStateManager` and `DefaultDS`:

```
<mbean code="org.jboss.mq.pm.jdbc2.MSSQLPersistenceManager"
name="jboss.mq:service=PersistenceManager">
```

```
<depends optional-attribute-name="
ConnectionManager">jboss.jca:service=DataSourceBinding,name=DefaultDS
</depends>
<attribute name="SqlProperties">
...

```

17. Rename the `[appserver root]/server/all/deploy-hasingleton/jms/hsqldb-jdbc-state-service.xml` file to `mssql-jdbc-state-service.xml` or, if you are using the Adobe-preconfigured JBoss from the DVD, rename the `[appserver root]/server/all/deploy-hasingleton/jms/mysql-jdbc-state-service.xml` file to `mssql-jdbc-state-service.xml`.
18. Open the `[appserver root]/server/all/deploy-hasingleton/jms/mssql-jdbc-state-service.xml` file and change mbean code to reference `MSSQLJDBCStateManager`:

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

```

19. Restart JBoss.

Configuring JMS with Microsoft SQL Server for JBoss 4.2.0

This section applies to Adobe-preconfigured JBoss 4.2.0.

Setting SQL Server as the JMS data source

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

Caution: If the JMS data source is different from `IDP_DS` (for example, a separate database or `MQSeries`), some JMS messages may be lost upon recovery from a system failure. This loss may include jobs that are waiting to be processed or processes that previously failed. To avoid this issue, configure the JMS data source to use the same database instance as `IDP_DS`.

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

► To set SQL Server to be the JMS data source:

1. Navigate to the `[appserver root]/server/all/deploy` directory and delete the `hsqldb-ds.xml` file or, if you are using the Adobe-preconfigured JBoss from the DVD, delete the `mysql-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 `<jndi-name>` value from `MSSQLDS` to `DefaultDS`.
4. Modify the `<local-tx-datasource>` element to use your SQL Server connection settings:

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

```

- ***your-hostname***: Replace this value with the name of the server hosting the SQL Server database
- ***yourdatabase-name***: Replace this value with the SQL Server database that will connect with LiveCycle ES.
- ***yourUserName***: Replace this value with the user name that will access the database.
- ***yourPassword***: Replace this value with the database user's password.

5. Save and close the file.

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

```
<datasource>java:/DefaultDS</datasource>
<datasource-mapping>MS_SQLSERVER2005</datasource-mapping>
<fk-constraint>true</fk-constraint>
```

Note: The `<datasource-mapping>` should always be `MS_SQLSERVER2000`, even if you are using SQL Server 2005.

7. Save and close the file.

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

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

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

database_name, ***database_username***, and ***password*** are the database values that the application server uses to access the database.

10. Save and close the file.

11. Delete the `[appserver root]/server/all/deploy/jms/mysql-jdbc2-service.xml` file.

12. Copy the `[appserver root]/docs/examples/jms/mssql-jdbc2-service.xml` file to `[appserver root]/server/all/deploy/jms`.

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

```
<mbean code="org.jboss.mq.pm.jdbc2.MSSQLPersistenceManager"
name="jboss.mq:service=PersistenceManager">
<depends optional-attribute-name="
```

```
    ConnectionManager">jboss.jca:service=DataSourceBinding,name=DefaultDS
  </depends>
  <attribute name="SqlProperties">
  ...
```

14. Rename the `[appserver root]/server/all/deploy/jms/mysql-jdbc-state-service.xml` file to `mssql-jdbc-state-service.xml`.
15. Open the `[appserver root]/server/all/deploy/jms/mssql-jdbc-state-service.xml` file and change `mbean` code to reference `MSSQLJDBCStateManager`:

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

16. Restart JBoss.

Configuring DB2 for Adobe-preconfigured JBoss

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

- Obtain and copy the DB2 JDBC driver to the instance of JBoss where you will deploy LiveCycle ES.
- Create a data source file and deploy it to the instance of JBoss where you will deploy LiveCycle ES.
- Configure JBoss to use DB2 for JMS.

Configuring the DB2 data source for JBoss 4.0.3 SP1

Before you configure the DB2 data source, you must have the database created on DB2. (See [“Creating a DB2 database” on page 52.](#))

► To install the DB2 database driver:

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

► To create the DB2 data source file:

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

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

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

4. 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`.
 - **50000:** The port used to access the database. The default is `50000`.
 - **adobe:** The name of the database that stores the LiveCycle ES data. You will need to update the default value, `adobe`, with your database name.

- **database_username** and **password**: The user name and password that the application server uses to access the database. You will need to update the default values, `adobe` and `adobe`, with the credentials for your database.
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>
```
 - For `com.celequest.metadata.metaDatasource` (BAM only):

```
<min-pool-size>1</min-pool-size>
<max-pool-size>20</max-pool-size>
```
 6. Save the file as `adobe-ds.xml` in the `[appserver root]/server/all/deploy` directory.
 7. Restart JBoss.

Configuring the DB2 data source for JBoss 4.2.0

Before configuring the DB2 data source, you must have the database created on DB2. (See [“Creating a DB2 database” on page 52.](#))

► To install the DB2 database driver:

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

► To obtain the latest `db2-jdbc2-service.xml` file:

1. Go to the [JBoss Download](#) site.
2. Under Servers/Runtime, click **JBoss Application Server**.
3. On the next page, click **Download** beside **4.2.0 GA**.
4. On the next page, click the **jboss-4.2.0.GA.zip** file.
5. After the file is downloaded, open `jboss-4.2.0.GA.zip` and extract the `db2-jdbc2-service.xml` file to the `[appserver root]/docs/examples/jms` directory.

► To create the DB2 data source file:

1. Delete the `adobe-ds.xml` file from the `[appserver root]/server/all/deploy` directory.
2. Copy the `adobe-ds-jboss-db2.xml` file from the `[DVD_root]/third_party/datasources` directory to the `[appserver root]/server/all/deploy` directory.
3. Open the `adobe-ds-jboss-db2.xml` file in a text editor and locate this line:

```
<connection-url>jdbc:db2://localhost:50000/adobe</connection-url>
```
4. 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`.
 - **50000**: The port used to access the database. The default is `50000`.
 - **adobe**: The name of the database that stores the LiveCycle ES data. You will need to update the default value, `adobe`, with your database name.
 - **database_username** and **password**: The user name and password that the application server uses to access the database. You will need to update the default values, `adobe` and `adobe`, with the credentials for your database.
5. Ensure 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>
```
 - For **com.celequest.metadata.metaDatasource** (BAM only):

```
<min-pool-size>1</min-pool-size>  
<max-pool-size>20</max-pool-size>
```
6. Save the file as `adobe-ds.xml` in the `[appserver root]/server/all/deploy` directory.
7. Restart JBoss.

Setting DB2 as the JMS data source for JBoss 4.03 SP1

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

Caution: If the JMS data source is different from `IDP_DS` (for example, a separate database or MQSeries), some JMS messages may be lost upon recovery from a system failure. This loss may include jobs that are waiting to be processed or processes that previously failed. To avoid this issue, configure the JMS data source to use the same database instance as `IDP_DS`.

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

► To set DB2 to be the JMS data source:

1. Navigate to the `[appserver root]/server/all/deploy` directory and delete the `mysql-ds.xml` file.
2. Copy the `db2-ds.xml` file from the `[appserver root]/docs/examples/jca` directory to the `[appserver root]/server/all/deploy` directory.
3. Open the `[appserver root]/server/all/deploy/db2-ds.xml` file in a text editor and modify the `<local-tx-datasource>` element with your DB2 connection settings:

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

4. Replace the bold text with values that are specific to your database:
 - **DB2DS**: Replace with `DefaultDS`.
 - **your_database**: The name of the DB2 database that stores the LiveCycle ES data.
 - **COM.ibm.db2.jdbc.app.DB2Driver**: Replace with `com.ibm.db2.jcc.DB2Driver`.
 - **x** and **y**: The user name and password that the application server uses to access the database.
5. Save and close the file.
6. Open the `[appserver root]/server/all/conf/standardjaws.xml` file and change the `<type-mapping>` element to DB2:

```
<type-mapping>DB2</type-mapping>
```
7. Save and close the file.
8. Open the `[appserver root]/server/all/conf/standardjbosscomp-jdbc.xml` file and change the following elements (not necessarily on consecutive lines):

```
<datasource>java:/DefaultDS</datasource>
<datasource-mapping>DB2</datasource-mapping>
<fk-constraint>true</fk-constraint>
```
9. Open the `[appserver root]/server/all/conf/login-config.xml` file in a text editor and add the following text within the `<policy>` element:

```
<application-policy name = "DB2DbRealm">
  <authentication>
    <login-module code =
      "org.jboss.resource.security.ConfiguredIdentityLoginModule" flag =
      "required">
      <module-option name = "principal">database_name </module-option>
      <module-option name = "userName">database_username</module-option>
      <module-option name = "password">password</module-option>
      <module-option name = "managedConnectionFactoryName">
        jboss.jca:service=LocalTxCM,name=DefaultDS
      </module-option>
    </login-module>
  </authentication>
</application-policy>
```
10. Replace the bold text with values that are specific to your database:
 - **database_name**, **database_username**, and **password**: The user name and password that the application server uses to access the database.
11. Save and close the file.
12. Delete the `mysql-jdbc.service.xml` file from `[appserver root]/server/all/deploy-hasingleton/jms`.
13. Copy `[appserver root]/docs/examples/jms/db2-jdbc2-service.xml` file to the `[appserver root]/server/all/deploy-hasingleton/jms` directory.
14. Open the `[appserver root]/server/all/deploy-hasingleton/jms/db2-jdbc2-service.xml` file and change DB2DS to `DefaultDS`:

```
<mbean code="org.jboss.mq.pm.jdbc2.PersistenceManager"
  name="jboss.mq:service=PersistenceManager">
```

```
<depends optional-attribute-name="ConnectionManager">  
  jboss.jca:service=DataSourceBinding,name=DefaultDS  
</depends>  
<attribute name="SqlProperties">  
  ...
```

15. Save and close the file.
16. Rename the `mysql-jdbc-state-service.xml` file, located in the `[appserver root]/server/all/deploy-hasingleton/jms` directory, to `db2-jdbc-state-service.xml`.
17. Restart JBoss.

Setting DB2 as the JMS data source for JBoss 4.2.0

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

Caution: If the JMS data source is different from IDP_DS (for example, a separate database or MQSeries), some JMS messages may be lost upon recovery from a system failure. This loss may include jobs that are waiting to be processed or processes that previously failed. To avoid this issue, configure the JMS data source to use the same database instance as IDP_DS.

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

► To set DB2 to be the JMS data source:

1. Navigate to the `[appserver root]/server/all/deploy` directory and delete the `mysql-ds.xml` file.
2. Copy the `db2-ds.xml` file from the `[appserver root]/docs/examples/jca` directory to the `[appserver root]/server/all/deploy` directory.
3. Open the `[appserver root]/server/all/deploy/db2-ds.xml` file in a text editor and modify the `<local-tx-datasource>` element with your DB2 connection settings:

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

4. Replace the bold text with values that are specific to your database:
 - **DB2DS**: Replace with `DefaultDS`.
 - **your_database**: The name of the DB2 database that stores the LiveCycle ES data.
 - **COM.ibm.db2.jdbc.app.DB2Driver**: Replace with `com.ibm.db2.jcc.DB2Driver`.
 - **x** and **y**: The user name and password that the application server uses to access the database.
5. Save and close the file.
6. Open the `[appserver root]/server/all/conf/standardjbosscomp-jdbc.xml` file and change the following elements (not necessarily on consecutive lines):

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

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

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

8. Replace the bold text with values that are specific to your database:
 - **database_name**, **database_username** and **password**: The user name and password that the application server uses to access the database.
9. Save and close the file.
10. Delete the `mysql-jdbc.service.xml` file from `[appserver root]/server/all/deploy/jms`.
11. Copy `[appserver root]/docs/examples/jms/db2-jdbc2-service.xml` file to the `[appserver root]/server/all/deploy/jms` directory.
12. Open the `[appserver root]/server/all/deploy/jms/db2-jdbc2-service.xml` file and change `DB2DS` to `DefaultDS`:

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

13. Save and close the file.
14. Rename the `mysql-jdbc-state-service.xml` file, located in the `[appserver root]/server/all/deploy/jms` directory, to `db2-jdbc-state-service.xml`.
15. Restart JBoss.

Next steps

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

5

Configuring a WebLogic Server

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

Name	Default value
<i>[appserver root]</i>	(Windows) C:\bea\weblogic92 or C:\bea\wlserver_10.1 (Linux and UNIX) /opt/bea/weblogic92 or /opt/bea/wlserver_10.1
<i>BEA_HOME</i>	(Windows) C:\bea (Linux and UNIX) /opt/bea
<i>[appserverdomain]</i>	(Windows) C:\bea\user_projects\domains\base_domain By default the domain is called <i>base_domain</i> . (Linux and UNIX) /opt/bea/user_projects/domains/base_domain

Overview

You will perform the following tasks to install and configure your BEA WebLogic Server 9.2/10.x environment:

1. Obtain WebLogic Server 9.2 or 10.x and install as per the manufacturer's documentation. (See ["Installing WebLogic Server" on page 73.](#))
2. Create and configure the WebLogic Server domain. (See ["Configuring anonymous admin look-up" on page 78.](#))
3. Access the WebLogic Server Administration Console.
4. Create a Managed Server. (See ["Creating a new WebLogic Managed Server" on page 77.](#))
5. Configure application server and data source connections. (See ["Configuring WebLogic Server" on page 78.](#))

LiveCycle ES 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 ES for WebLogic](#).

Installing WebLogic Server

Obtain WebLogic Server 9.2 or 10.x for your operating system and install it according to the manufacturer's documentation (see <http://bea.com/>). To facilitate the use of WebLogic for joint Adobe and BEA customers, BEA has provided the [BEA Downloads](#) site from which you can download the exact version of WebLogic that LiveCycle ES requires.

You must also obtain and install WebLogic Server patch 8ART from [BEA eSupport](#), who will provide you with installation instructions and a password for the patch.

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

Configuring WebLogic 10.1 Node Manager (64-bit Java)

Additional steps need to be done when installing WebLogic to use Node Manager with 64-bit Java. Here is an example of the required steps for UNIX:

1. Run the WebLogic installer using the `-d64` flag. (See your WebLogic documentation for more information.)
2. In a text editor, open `wlserver_10.1/common/bin/commEnv`.
3. Replace all instances of **sparc** with **sparc64**.
4. Move to the `wlserver_10.1/server/bin` directory.
5. Type `export JAVA_OPTIONS=-d64`.
6. Start Node Manager by using the script; type `startNodeManager.sh`.

Installing the J2SE SDK

This section describes how to install the JDK for Weblogic 9.2. WebLogic 10.1 is installed with the JDK that LiveCycle ES requires.

► (WebLogic 9.2 on Windows or Linux only) To install the JDK:

- Upgrade the BEA JRockit® JDK to JRockit 5.0 R26.4 CR302700 (and all later 5.0 releases of JRockit) on Windows and Linux operating systems.

As part of your WebLogic installation, a Java SDK was installed. The `JAVA_HOME` and `PATH` environment variables must point to the Java SDK on the server where LiveCycle ES 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:\bea\jrockit-jdk1.5.0_06
```

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

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

You should receive a response that begins with Java version `1.5.0_06` (or later version of 1.5.0).

Note: If you are using PDF Generator ES to convert Open Office files on a WebLogic Server 10.1 MP1 environment, you must set the `JAVA_HOME_32` environment variable.

► **To set the `JAVA_HOME_32` 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_32` as the variable name and the directory where you installed the supported Java SDK. For example, type this text:

```
C:\Program Files\java\jdk1.5.0_11
```

► **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;
```

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

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

```
JAVA_HOME=/opt/jrockit 1.5.0.6
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 ES to convert Open Office files on a WebLogic Server 10.1 MP1 environment.

► **To set the JAVA_HOME_32 environment (UNIX and Linux):**

- Set the JAVA_HOME_32 variable for the supported Java SDK for Borne and Bash shells as shown in this example:

```
JAVA_HOME_32=/opt/jdk1.5.0_11
export JAVA_HOME_32
```

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

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

- Set the PATH variable for Borne and Bash shells as shown in this example:

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

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

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

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

- (Windows) `config.cmd`
- (Linux, UNIX) `./config.sh`

2. On the 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 BEA products** and click **Next**.

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

4. Type your WebLogic user name and password, confirm the password by retyping it, and then click **Next**. Record this 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, select the BEA-supplied JDK according to your operating system and then click **Next**:

- (Windows, Linux) **JRockit SDK 1.5.0_06@BEA_HOME/jrockit90_150_06**. You must select **Other JDK** in the UI and specify the location of the jrockit-jdk1.5.0_06.
- (Solaris) **Sun SDK 1.5.0_04 @ BEA_HOME/jdk150_04**

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

8. On the Create WebLogic Domain screen, either accept the default values or enter the domain name and location as required, and 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, UNIX) Click **Done**, and start the server by navigating from a command prompt to the `[appserverdomain]` and typing `./startWebLogic.sh`

10. (Windows only) When prompted, enter the WebLogic user name and password that you entered in step 4.

Troubleshooting the WebLogic 9.2 (64 bit)

When using WebLogic 9.2 (64bit), refer to the BEA website for detailed instructions on addressing the issue if you see either of these errors:

- (When starting Administration Server) `javax.xml.namespace.QName; local class incompatible`
- (For Managed Server) `javax.xml.namespace.QName incompatible`

Starting Node Manager

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

► To start Node Manager:

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

(Windows) `startNodeManager.cmd`

(Linux, UNIX) `./startNodeManager.sh`

Creating a new WebLogic Managed Server

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

Note: The following procedure highlights the changes 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 `BEA_HOME\user_projects\domains\[domainname]` directory and type the appropriate command:
 - (Windows) `startWebLogic.cmd`
 - (Linux, UNIX) `./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 right pane, 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.
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 right pane, 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 select 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. On the Configuration tab, click the **Server Start** tab.
17. In the **Arguments** box, type the appropriate text to set the memory size for the Managed Server:
 - (Solaris) `-Xms256m -Xmx1024m -XX:MaxPermSize=256m`
 - (All other operating systems) `-Xms256m -Xmx1024m`

- (64-bit only) `-Xms256m -Xmx1792m -XX:MaxPermSize=512m`

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

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

► **To complete the memory settings for the Managed Server:**

1. Move to the following directory: `BEA_HOME\user_projects\domains\[appserverdomain]\bin`.
2. Edit the following file in a text editor:
 - (Windows) `setDomainEnv.cmd`
 - (UNIX) `setDomainEnv.sh`
3. Search for the line `set MEM_ARGS="-Xms256m -Xmx512m"` and change it to `MEM_ARGS="-Xms256m -Xmx1024m"`.
4. Search for the line `-XX:MaxPermSize=128m` and change it to `-XX:MaxPermSize=256m` for the 32-bit version and `-XX:MaxPermSize=512m` for the 64-bit version.

Note: `-XX:MaxPermSize=128m` has two locations:

- Linux, UNIX:
 - `if ["${JAVA_VENDOR}" = "Sun"] (...`
 - `if ["${JAVA_VENDOR}" = "HP"] (...`
 - Windows:
 - `if "%JAVA_VENDOR%"=="Sun" (...`
 - `If "%JAVA_VENDOR%"=="HP" (...`
5. Save the changes and close the file.

Configuring WebLogic Server

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

Configuring anonymous admin look-up

Enable anonymous admin look-up on the application server. This setting allows read-only access to the WebLogic Server MBeans.

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

1. With WebLogic Server running, type `http://[host name]:[port]/console` in the URL line of a web browser to access the WebLogic Server Administration Console.
2. Type the user name and password that was used in creating this WebLogic configuration, and then click **Log In**.

3. Under Change Center, click **Lock & Edit**.
4. Under Domain Structure, click the name of your domain.
5. In the right pane, click the **Security** tab and select **Anonymous Admin Lookup Enabled**.
6. Click **Save** and then click **Activate Changes**.

Note: Do not restart your server at this point. After all the WebLogic configuration changes are made, the server will be restarted when you get to the section [“Stopping and restarting WebLogic” on page 80](#).

Configuring WebLogic for Web Services

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

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

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

► 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 `[BEA_HOME]\weblogic92\server\bin\` folder, type `setWLSEnv.cmd`
 - (Linux, UNIX) From the `[BEA_HOME]/weblogic92/server/bin/` directory, type `setWLSEnv.sh`
3. Start the WebLogic scripting tool by typing the following command:

```
java weblogic.WLST
```

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

- From a command prompt, set the environment by using `<BEA Home>/weblogic92/server/bin/setWLSEnv.cmd`
- From a command prompt, copy the classpath that is returned from the `setWLSEnv.cmd` or `setWLSEnv.sh` command and append the exported `CLASSPATH`, as shown in this example:

```
CLASSPATH=/usr/local/boa92/patch_weblogic920/profiles/default/sys_manifest_classpath/weblogic_patch.jar:/usr/local/boa92/jdk150_04/lib/tools.jar:/usr/local/boa92/weblogic92/server/lib/weblogic_sp.jar:/usr/local/boa92/weblogic92/server/lib/weblogic.jar:/usr/local/boa92/weblogic92/server/lib/webservices.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=/usr/local/boa92/weblogic92/server/bin:/usr/local/boa92/jdk150_04/jre/bin:/usr/local/boa92/jdk150_04/bin:/usr/local/boa92/weblogic92/server/bin:/usr/local/boa92/jdk150_04/jre/bin:/usr/local/boa92/jdk150_04/bin:/usr/sbin:/usr/bin export PATH
```

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

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

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

Stopping and restarting WebLogic

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

► To 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.

► To stop WebLogic Administration Server:

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

► To restart WebLogic Administration Server:

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

2. Type the following command:
 - (Windows) `startWebLogic.cmd`
 - (Linux, UNIX) `./startWebLogic.sh`
 3. Enter the WebLogic user name and password (if you enabled security when installing WebLogic).
- **To restart WebLogic Managed Server:**
1. When the WebLogic Administration Server has started, log in to the WebLogic Server Administration Console.
 2. Under Change Center, click **Lock & Edit**.
 3. Under Domain Structure, click **Environment > Servers** and, in the right pane, click the managed server.
 4. On the next screen, click the **Control** tab and select the check box beside the managed server you want to start.
 5. Click **Start** and then click **Yes**.

Configuring the LiveCycle ES database connectivity

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

Next steps

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

6

Configuring a WebSphere Application Server

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

This section uses the following conventions.

Name	Default value
<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

Installing WebSphere Application Server

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

When you run the LiveCycle ES installer, you should run it as the same user that installed WebSphere Application Server.

Installing a Fix Pack to WebSphere 6.1

After you install WebSphere 6.1, you must update to Fix Pack 7 or later before you deploy LiveCycle ES.

After you install the WebSphere Fix Pack, the SR for the JDK that the Fix Pack requires must be installed. For Fix Pack 7, the SR for the JDK is SR4.

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

Setting the JAVA_HOME and PATH environment variables

As part of your WebSphere installation, a Java SDK (JDK) was installed. The JAVA_HOME and PATH environment variables must point to the JDK where LiveCycle ES 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 enter the directory where you installed the JDK as the value. Ensure that you installed the correct JDK (32-bit versus 64-bit version). (See ["Third-party](#)

[infrastructure support" on page 10.](#)) This directory is where WebSphere installed the Java SDK that contains the /bin sub directory. For example, type the following path:

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

5. (Optional) Verify your JAVA_HOME environment variable, open a command prompt and run the following command:

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

You should receive a response such as this one:

```
C:\Program Files\IBM\WebSphere\AppServer\java\bin>java -version
java version "1.5.0"
Java(TM) 2 Runtime Environment, Standard Edition (build pwi32dev-20060511
(SR2))
```

```
IBM J9 VM (build 2.3, J2RE 1.5.0 IBM J9 2.3 Windows Server 2003 x86-32
j9vmwi322
3-20060504 (JIT enabled)
J9VM - 20060501_06428_1HdSMR
JIT - 20060428_1800_r8
GC - 20060501_AA)
JCL - 20060511a
```

► **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 (UNIX and Linux):**

- From a command prompt, type the following text:

```
Export DISPLAY= <IP Address of the Windows machine telnet, or cygwin to
Unix-Linux 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 (UNIX and Linux):**

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

```
JAVA_HOME=/usr/IBM/WebSphere/AppServer/java
export JAVA_HOME
```

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

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

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

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

Accessing WebSphere Administrative Console

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

► **To start WebSphere Base:**

1. To start the 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`
- (UNIX, Linux) `./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 6.1.0.7 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**.

Setting directory permissions

The LiveCycle ES application must extract files to the *[appserver root]/installedApps* directory, such as `/usr/IBM/WebSphere/AppServer`. Therefore, it is important that writable permissions be given to that directory. If writable permissions cannot be given, the section below describes how to modify the location for the extracted files.

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

1. Log in to the WebSphere Administrative Console.
2. Click **Servers > Application servers** and click your server name, 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.

Preparing WebSphere Application Server

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

Required application server instances

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

Increasing the SOAP request time out

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

► To increase the SOAP request time-out value:

1. Go to the `[appserver root]` directory and search for all files named `soap.client.props`. Multiple files may have this name. For example, on a UNIX server, the following files exist:
 - `[appserver root]/profileTemplates/default/documents/properties/soap.client.props`
 - `[appserver root]/profileTemplates/cell/default/documents/properties/soap.client.props`
 - `[appserver root]/cip/profileTemplates/minimal/documents/properties/soap.client.props`
 - `[appserver root]/profiles/AppSrv01/properties/soap.client.props`
 - `[appserver root]/profiles/AppSrv01/temp/soap.client.props`
2. Open each `soap.client.props` file in a text editor, find the `com.ibm.SOAP.requestTimeout` property, and change the value from 180 to 1800.
3. Save each `soap.client.props` file.
4. In the navigation tree of the WebSphere Administrative Console, click **Servers > 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. On the next screen, in the **Value** box, change 600 to 1800 and then click **OK** or **Apply**.
9. In the Messages box, click **Save directly to master configuration**.

Increasing the MaxPermSize and heap size (UNIX only)

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

► **To increase the MaxPermSize and heap size:**

1. Go to the `[appserver root]/deploytool/itp/` directory and open `ejbdeploy.sh` for editing.
2. 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:
 - (32-bit OS) `-Xms256m -Xmx1024m`.
 - (64-bit OS) `-Xms256m -Xmx1792m`.

Configuring the LiveCycle ES database connectivity

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

Next steps

Install LiveCycle ES by following the instructions provided in [Installing and Deploying LiveCycle ES 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:

Preconfiguration	Record data here	Required?
<i>Server Setup</i>		
System Requirements met		Yes
JDK Install location (for 32 bit or 64 bit OS)		Yes
Global Storage Directory created		Only if you do not want to use the default locations.
<i>Application Server Configuration</i>		
Application Server hostname or IP Address		Yes
Application Server credentials		Yes
Application Server port number		Yes
Preconfiguration completed as described in "Configuring a JBoss Application Server" on page 49 or "Configuring a WebLogic Server" on page 72 or "Configuring a WebSphere Application Server" on page 82 .		Yes
<i>Database Configuration</i>		
Database hostname or IP Address		Yes
Database account credentials		Yes
Database port number		Yes
Preconfiguration completed as described in "Creating the LiveCycle ES Database" on page 46 .		Yes
<i>LDAP Server</i>		
LDAP Server hostname or IP Address		This is only required if you are using LDAP to authenticate users.

Preconfiguration	Record data here	Required?
LDAP account credentials		This is only required if you are using LDAP to authenticate users.
<i>PDF Generator ES/PDF 3D Generator ES configuration (Windows)</i>		
Install Microsoft Office		Yes. Note you must use the same user account to install Acrobat and Microsoft Office.
Install Acrobat		Yes. Note you must use the same user account to install Acrobat and Microsoft Office.
Other native application software installed		Yes.
Configure the Windows environment variables		Yes.

A

Appendix - Manually Configuring JBoss

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

If you intend to use the Adobe-preconfigured JBoss, see [“Configuring a JBoss Application Server” on page 49](#).

LiveCycle ES runs on JBoss on Windows Server 2003 (Enterprise Edition or Standard Edition), Red Hat Linux ES/AS 4.0 or 5.0, and SUSE Linux ES 9 platforms. (See [“Third-party infrastructure support” on page 10](#).)

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

This chapter uses the following conventions.

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

Installing the J2SE SDK

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

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

► To set the `JAVA_HOME` environment variable (Windows):

1. Select **Start > Control Panel > System**.
2. Click the **Advanced** tab.
3. Click **Environment Variables** 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 Java SDK. This directory is the directory that contains the `/bin` subdirectory. For example, type the following path:

```
C:\Program Files\Java\jdk1.5.0_11
```

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

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

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

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

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

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

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

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

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

Manually installing JBoss

If you are manually configuring JBoss for running LiveCycle ES, download and unpackage JBoss Application Server 4.0.3 SP1 or 4.2.0.

► **To download and unpackage JBoss:**

1. Go to the [JBoss Application Server Download](#) site.
2. Locate the **4.0.3 SP1** or **4.2.0** 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-4.0.3SP1.zip or jboss-4.2.0.GA.zip
 - (Linux) jboss-4.0.3SP1.tar.gz or jboss-4.2.0.GA-src.tar.gz
4. After the download is complete, extract the file to the directory of your choice on your server.

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.

► **To start JBoss:**

1. From a command prompt, navigate to *[appserver root]/bin*.
2. Start the application server by typing the following command:
 - (Windows) `run.bat -c all`
 - (Linux) `./run.sh -c all`

► **To stop JBoss:**

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

2. Stop the application server by typing the following command:

- (Windows) `shutdown.bat -S`
- (Linux) `./shutdown.sh -S`

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 probably crash. JBoss also has a few configuration files that are formatted as .property files. These files must be UNIX text files on Linux or Solaris. It is important to take care if they are edited on Windows environments at any time.

Modifying the JBoss configuration files

This section is required only if you are not using the Adobe-preconfigured JBoss software provided on the installation media.

Modify the following areas on the instance of JBoss that hosts LiveCycle ES:

- Update the xalan version (JBoss 4.0.3 SP1 only).
- Update the `jacorb.properties` file.
- Update the `quartz.jar` and `commons-collections.jar` libraries.
- Modify the EAR and WAR file class-loading isolation.
- Remove the JavaServer™ Faces (JSF) libraries (JBoss 4.0.3 SP1 only).
- Disable the JavaServer™ Faces (JSF) libraries (JBoss 4.2.0 only)
- Disable cluster service (for single node installations only).
- Enable Watched Folder invocations.
- Remove the tests directory (JBoss 4.0.3 SP1 only).
- Create the security policy file.
- Modify the `run.bat` file (Windows only).
- Modify the `run.sh` file (Linux only).
- Modify `run.conf`.
- Modify `log4j.xml`.
- Remove the JMX Console and Web Console.
- Create the Java Message Service (JMS) topics and queues.
- Modify the `jmx-invoker-service.xml` file so that authenticated users are not required.

► To update the xalan version (JBoss version 4.0.3 SP1 only):

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

<http://archive.apache.org/dist/xml/xalan-j/>

2. Download the required file:

- (Windows) `xalan-j_2_7_0-bin.zip`

- (Linux) xalan-j_2_7_0-bin.tar.gz
 - 3. Extract the xalan.jar file and the serializer.jar file from the downloaded archive file.
 - 4. From the *[appserver root]/lib/endorsed* directory, delete the old xalan.jar file.
 - 5. Copy the new xalan.jar file and the serializer.jar files to the *[appserver root]/lib/endorsed* directory.
- **To update the jacorb.properties file:**
1. Open the *[appserver root]/server/all/conf/jacorb.properties* file in a text editor.
 2. Locate the `jacorb.poa.thread_pool_max` setting and change the value to 16.
- **To update the quartz.jar and commons-collections.jar libraries (JBoss version 4.0.3 SP1 only):**
1. Delete the existing commons-collections.jar from the *[appserver root]/server/all/lib* directory.
 2. On the LiveCycle ES DVD or ESD, navigate to the */third_party/jboss_4.2.0/patches* directory.
 3. Copy the quartz.jar and commons-collections.jar files from this directory to the *[appserver root]/server/all/lib* directory.
- **To update the quartz.jar library (JBoss version 4.2.0 only):**
1. On the LiveCycle ES DVD or ESD, navigate to the */third_party/jboss_4.2.0/patches* directory.
 2. Copy the quartz.jar file from this directory to the *[appserver root]/server/all/lib* directory.
- **To modify WAR file class-loading isolation:**
1. Open the *[appserver root]/server/all/deploy/jboss-web.deployer/META-INF/jboss-service.xml* file in an editor.
 2. Locate `<attribute name="Java2ClassLoadingCompliance">` and ensure that the value is `false`.
 3. Locate `<attribute name="UseJBossWebLoader">` and ensure that the value is `false`.
 4. Save and close the file.
- **To modify EAR file class-loading isolation:**
1. Open the *[appserver root]/server/all/conf/jboss-service.xml* file in an editor.
 2. Locate `<attribute name="CallByValue">` and change the value to `true`.
 3. Save and close the file.
 4. Open the *[appserver root]/server/all/deploy/ear-deployer.xml* file in an editor.
 5. Locate `<attribute name="Isolated">` and change the value to `true`.
 6. Locate `<attribute name="CallByValue">` and change the value to `true`.
 7. Save and close the file.

► **To remove the JSF libraries (JBoss version 4.0.3 SP1 only):**

- For JBoss 4.0.3 navigate to the `[appserver root]/server/all/deploy/jbossweb.deploy` directory and delete the `jsf-libs` subdirectory and all its files.

► **To disable the JSF libraries (JBoss version 4.2.0 only)**

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/<config-name>/deploy/jboss-web.deployer/conf/web.xml`:

```
<!-- 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>
  -->
```

► **To disable the cluster service:**

1. Open the `[appserver root]/server/all/deploy/cluster-service.xml` file in an editor, and locate and modify the following attribute values:
 - `mcast_port="34374"` (for JBoss version 4.0.3 SP1)
 - `mcast_port="34845"` (for JBoss version 4.2.0)
 - `ip_ttl="1"`

- `ip_mcast` and change it to `false`
2. Save and close the file.
 3. Open the `[appserver root]/server/all/deploy/jms/hajndi-jms-ds.xml` file in an editor.
 4. Locate `jnp.disableDiscovery` and change the value to `true`.
 5. Save and close the file.

► **To enable Watched Folder invocations:**

1. Open the `[appserver root]/server/all/deploy/jms/hajndi-jms-ds.xml` file in an editor and add the following connection factories:

Note: The following text contains formatting characters for line breaks. If you copy and paste this text, you must remove the formatting characters.

```
<tx-connection-factory>
  <track-connection-by-tx>true</track-connection-by-tx>
  <jndi-name>adobe_JmsQueueXA</jndi-name>
  <xa-transaction/>
  <rar-name>jms-ra.rar</rar-name>
  <connection-definition>org.jboss.resource.
adapter.jms.JmsConnectionFactory</connection-definition>
  <adapter-display-name>JMS Adapter</adapter-display-name>
  <config-property type="java.lang.String" name="SessionDefaultType">
javax.jms.Topic</config-property>
  <security-domain-and-application>JmsXARealm
</security-domain-and-application>
  <config-property type="java.lang.String"
name="JmsProviderAdapterJNDI">
  java:/DefaultJMSPProvider</config-property>
</tx-connection-factory>
```

```
<tx-connection-factory>
  <track-connection-by-tx>true</track-connection-by-tx>
  <jndi-name>adobe_JmsTopicXA</jndi-name>
  <xa-transaction/>
  <rar-name>jms-ra.rar</rar-name>
  <connection-definition>org.jboss.resource.adapter.jms.
JmsConnectionFactory</connection-definition>
  <adapter-display-name>JMS Adapter</adapter-display-name>
  <config-property type="java.lang.String"
name="SessionDefaultType">javax.jms.Topic</config-property>
  <security-domain-and-application>JmsXARealm<
/security-domain-and-application>
  <config-property type="java.lang.String"
name="JmsProviderAdapterJNDI">java:/DefaultJMSPProvider</
config-property>
</tx-connection-factory>
```

2. Save and close the file.

➤ **To remove the tests directory (JBoss version 4.0.3 SP1 only):**

- Delete the `[appserver root]/docs/tests` directory.

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

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

```
grant {
    permission java.security.AllPermission;
};
```

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

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

1. Open the `[appserver root]/bin/run.bat` file in an editor.
2. 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. Locate `rem Sun JVM memory allocation pool parameters` and insert the following text immediately before that string:

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

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

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

5. (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)"
```

6. Save and close the file.

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

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

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

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

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

3. (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)"
```

4. Save and close the file.

► **To modify the `run.conf` file:**

1. Open the `[appserver root]/bin/run.conf` file in an editor.
2. Locate `JAVA_OPTS="-server -Xms128m -Xmx128m"` and modify it to read as follows:

- For JBoss version 4.0.3 SP1

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

- For JBoss version 4.2.0

```
JAVA_OPTS="-server -XX:MaxPermSize=256m -Xms256m -Xmx1024m  
-Dsun.rmi.dgc.client.gcInterval=3600000  
-Dsun.rmi.dgc.server.gcInterval=3600000"
```

3. If you are using Linux, you must also add the `-Djava.net.preferIPv4Stack=true` flag to remove the warning "[HANamingService] Failed to start AutomaticDiscovery".
4. Save and close the file.

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

1. Open the `[appserver root]/server/all/conf/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"/>  
</appender>
```

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 a com.adobe category, as shown in the following lines:

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

5. Add the following categories under the com.adobe 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>
```

6. Save and close the file.

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

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

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

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

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

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

```
<?xml version="1.0" encoding="UTF-8"?>
<server>
  <mbean code="org.jboss.mq.server.jmx.Queue"
name="jboss.mq.destination:service=Queue,name=adobe_PECCommandQueue">
  <depends optional-attribute-name="
DestinationManager">jboss.mq:service=DestinationManager</depends>
  </mbean>
  <mbean code="org.jboss.mq.server.jmx.Queue"
name="jboss.mq.destination:service=Queue,name=adobe_PEDCommandQueue">
  <depends optional-attribute-name="
DestinationManager">jboss.mq:service=DestinationManager</depends>
  </mbean>
  <mbean code="org.jboss.mq.server.jmx.Queue"
name="jboss.mq.destination:service=Queue,name=adobe_PEInteractionQueue">
  <depends optional-attribute-name="
DestinationManager">jboss.mq:service=DestinationManager</depends>
  </mbean>
  <mbean code="org.jboss.mq.server.jmx.Queue"
name="jboss.mq.destination:service=Queue,name=adobe_JobManagerQueue">
  <depends optional-attribute-name="
DestinationManager">jboss.mq:service=DestinationManager</depends>
  </mbean>
  <mbean code="org.jboss.mq.server.jmx.Topic"
name="jboss.mq.destination:service=Queue,name=adobe_TaskEventTopic">
  <depends optional-attribute-name="
DestinationManager">jboss.mq:service=DestinationManager</depends>
  </mbean>
</server>
```

3. Save and close the file.

► To 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.jms.connector.invoker.AuthenticationInterceptor"
securityDomain="java:/jaas/jmx-console"/>
```

3. Save and close the file.

Configuring the LiveCycle ES database connectivity

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

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

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

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

Configuring MySQL for manually installed JBoss

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

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

If you are using Adobe-preconfigured JBoss, see [“Configuring MySQL for Adobe-preconfigured JBoss” on page 53](#).

Caution: If the JMS data source is different from IDP_DS (for example, a separate database or MQSeries), some JMS messages may be lost upon recovery from a system failure. These messages may include jobs that are waiting to be processed or processes that previously failed. To avoid this issue, configure the JMS data source to use the same database instance as IDP_DS.

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 46](#).)

► To install the MySQL database driver:

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

► To create the MySQL data source file:

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

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

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

3. After the `max-pool-size` element, add the following text:

```
<valid-connection-checker-class-name>com.mysql.jdbc.integration.jboss.MySQLValidConnectionChecker</valid-connection-checker-class-name>
```

```
<exception-sorter-class-name>com.mysql.jdbc.integration.jboss.ExtendedMySQLExceptionSorter</exception-sorter-class-name>
  <new-connection-sql>SELECT count(*) from DUAL</new-connection-sql>
  <check-valid-connection-sql>SELECT count(*) from
    DUAL</check-valid-connection-sql>
```

4. 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 ES data. You will need to update the default value, adobe, with your database name.
 - **database_username** and **password:** The user name and password that the application server uses to access the database. You will need to update the default values, adobe and adobe, with the credentials for your database.
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>
```
 - com.celequest.metadata.metaDatasource (BAM only):

```
<min-pool-size>1</min-pool-size>
<max-pool-size>20</max-pool-size>
```
6. Save the file as adobe-ds.xml in the `[appserver root]/server/all/deploy` directory.

Setting MySQL as the JMS data source

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

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

► To set MySQL as the JMS data source:

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

```
<jndi-name>MySQLDS</jndi-name>
<connection-url>jdbc:mysql://yourHostname:yourPort/yourDatabaseName</connection-url>
<driver-class>com.mysql.jdbc.Driver</driver-class>
```

```
<user-name>yourUserName</user-name>  
<password>yourPassword</password>
```

4. Replace the bold values with values that are specific to your database:
 - **MySQLDS:** Change to DefaultDS.
 - **yourHostname, yourPort, yourDatabaseName, youruserName, and yourPassword:** The database values that the application server uses to access the database.
5. Save and close the file.
6. (**JBoss 4.0.3 SP1 only**) Open the `[appserver root]/server/all/conf/standardjaws.xml` file and change the `<type-mapping>` element to MySQL:

```
<type-mapping>mysql</type-mapping>
```

7. (**JBoss 4.0.3 SP1 only**) Save and close the file.
8. Open the `[appserver root]\server\all\conf\standardjbosscomp-jdbc.xml` file and change the following elements (not necessarily contiguous):

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

9. Remove the text `<!--optional since 4.0 and .."` from the `<datasource-mapping>` tag to uncomment it.
10. Save and close the file.
11. Open the `[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">database_name</module-option>  
      <module-option name="userName">database_username</module-option>  
      <module-option name="password">database_password</module-option>  
      <module-option  
        name="managedConnectionFactoryName">jboss.jca:service=LocalTxCM,  
        name=DefaultDS </module-option>  
      </login-module>  
    </authentication>  
  </application-policy>
```

12. Replace the bold values with values that are specific to your database:
 - **database_name, database_username, and database_password:** The database values that the application server uses to access the database.
13. Save and close the file.
14. Delete the `hsqldb2-service.xml` file in the `[appserver root]/server/all/deploy-hasingleton/jms` directory, and copy the `mysql-jdbc2-service.xml` file from the `[appserver root]/docs/examples/jms` directory to the `[appserver root]/server/all/deploy-hasingleton/jms` directory.

15. Open the `[appserver root]/server/all/deploy-hasingleton/jms/mysql-jdbc2-service.xml` file.

16. Change `MySqlDS` to `DefaultDS`:

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

```

17. (**JBoss 4.0.3 SP1 only**) In the same file, `mysql-jdbc2-service.xml`, change the `SELECT_MAX_TX` line to the following text:

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

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

19. Restart JBoss.

Configuring Oracle for manually installed JBoss

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

- Obtain and copy the Oracle JDBC driver to the instance of JBoss where you will deploy LiveCycle ES.
- Create a data source file and deploy it to the instance of JBoss where you will deploy LiveCycle ES.
- Configure JBoss to use Oracle for JMS.

Caution: If the JMS data source is different from `IDP_DS` (for example, a separate database or `MQSeries`), some JMS messages may be lost upon recovery from a system failure. These messages may include jobs that are waiting to be processed or processes that previously failed. To avoid this issue, configure the JMS data source to use the same database instance as `IDP_DS`.

Configuring the Oracle data source

Before you configure the Oracle data source, you must have the database created on Oracle. (See [“Creating an Oracle database” on page 38.](#))

► To install the Oracle 10g database driver:

- Copy the `ojdbc14.jar` driver file from the `[LiveCycleES root]/lib/db/oracle` directory to the `[appserver root]/server/all/lib` directory. You can also download the Oracle 10g Release 2 (10.2.0.2 Thin) driver from the [JDBC Driver Downloads](#) site.

► To create the Oracle data source file:

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

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

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

3. Replace the following 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:** The name of the database that stores the LiveCycle ES data. You will need to update the default value, `adobe`, with your database name.
 - **database_username** and **password:** The user name and password that the application server uses to access the database. You will need to update the default values, `adobe` and `adobe`, with the credentials for your database.
4. 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>
```
 - **com.celequest.metadata.metaDatasource (BAM only):**

```
<min-pool-size>1</min-pool-size>  
<max-pool-size>20</max-pool-size>
```
5. Save the file as `adobe-ds.xml` in the `[appserver root]/server/all/deploy` directory.
6. Restart JBoss.

Setting Oracle as the JMS data source

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

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

► To set Oracle to be the JMS data source:

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

```
<jndi-name>OracleDS</jndi-name>  
<connection-url>jdbc:oracle:thin:@youroraclehost:1521:yoursid  
</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.
 - Replace ***x*** with the user name to access the Oracle database and ***y*** with the password associated with the user. The application server uses these credentials to access the database.
5. Save and close the file.
 6. **(JBoss 4.0.3 SP1 only)** Open the `[appserver root]/server/all/conf/standardjaws.xml` file and change the `<type-mapping>` element to Oracle9i:

```
<type-mapping>oracle9i</type-mapping>
```
 7. **(JBoss 4.0.3 SP1 only)** Save and close the file.
 8. Open the `[appserver root]/server/all/conf/standardjbosscomp-jdbc.xml` file and change the following elements (not necessarily on consecutive lines):

```
<datasource>java:/DefaultDS</datasource>
<datasource-mapping>oracle9i</datasource-mapping>
<fk-constraint>>true</fk-constraint>
```
 9. Remove the text: `<!--optional since 4.0 and ..` from the `<datasource-mapping>` tag to uncomment it.
 10. Save and close the file.
 11. Open the `[appserver root]/server/all/conf/login-config.xml` file in a text editor and add the following text within the `<policy>` element:

```
<application-policy name = "OracleDbRealm">
  <authentication>
    <login-module code =
      "org.jboss.resource.security.ConfiguredIdentityLoginModule" flag =
      "required">
      <module-option name = "principal">database_name</module-option>
      <module-option name = "userName">database_username</module-option>
      <module-option name = "password">password</module-option>
      <module-option name = "managedConnectionFactoryName">
        jboss.jca:service=LocalTxCM,name=DefaultDS</module-option>
    </login-module>
  </authentication>
</application-policy>
```
 12. Replace the bold text with values that are specific to your database:
 - ***database_name***, ***database_username***, and ***password***: The database values that the application server uses to access the database.
 13. Save and close the file.
 14. Delete the `hsqldb-jdbc2-service.xml` file located in the `[appserver root]/server/all/deploy-hasingleton/jms` directory
 15. Copy the `oracle-jdbc2-service.xml` file from the `[appserver root]/docs/examples/jms` directory to the `[appserver root]/server/all/deploy-hasingleton/jms` directory.

16. Open the `oracle-jdbc2-service.xml` file and change `OracleDS` to `DefaultDS`. This file is located in the `[appserver root]/server/all/deploy-hasingleton/jms/` directory.

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

```

17. Save and close the file.
18. Rename the `hsqldb-jdbc-state-service.xml` file to `oracle-jdbc-state-service.xml`. This file is located in the `[appserver root]/server/all/deploy-hasingleton/jms` directory.
19. Restart JBoss.

Configuring SQL Server

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

- Obtain and copy the SQL Server JDBC driver files to the instance of JBoss where you will deploy LiveCycle ES.
- Create a SQL Server data source file and deploy it to the instance of JBoss where you will deploy LiveCycle ES, such as `[appserver_root]/server/all/deploy`.
- Configure JBoss to use SQL Server for JMS.

Caution: If the JMS data source is different from `IDP_DS` (for example, a separate database or `MQSeries`), some JMS messages may be lost upon recovery from a system failure. These messages may include jobs that are waiting to be processed or processes that previously failed. To avoid this issue, configure the JMS data source to use the same database instance as `IDP_DS`.

Configuring the SQL Server data source

Before you configure the SQL Server data source, you must have the LiveCycle ES database created on SQL Server. (See [“Creating a SQL Server database” on page 39](#).)

► To install the SQL database driver:

1. Obtain the SQL Server 2005 JDBC 1.1 database driver from the Microsoft website.
2. (Windows) Download the *.exe file and run 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_1.1/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.

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

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

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

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

3. Change the `<driver-class>` element as follows:

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

4. 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. The default port is `1433`.
- **adobe:** The name of the database that stores the LiveCycle ES data. You will need to update the default value, `adobe`, with your database name.
- **database_username** and **password:** The user name and password that the application server uses to access the database. You will need to update the default values, `adobe` and `adobe`, with the credentials for your database.

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>
```

- `com.celequest.metadata.metaDatasource` (BAM only):

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

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

7. Restart JBoss.

This next procedure provides instructions about how to use integrated security to make a trusted connection with SQL Server.

► **To configure Integrated Security on Windows:**

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

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

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

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.

Configuring JMS with Microsoft SQL Server

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

Setting SQL Server as the JMS data source

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

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

► To set SQL Server to be the JMS data source:

1. Navigate to the `[appserver root]/server/all/deploy` directory and delete the `hsqldb-ds.xml` file. If you are using the Adobe-preconfigured JBoss from the DVD, delete the `mysql-ds.xml` file.
2. (**JBoss 4.0.3 SP1 only**) Install the JBossMQ patch file called `jbossmq.jar`. You must replace the existing `jbossmq.jar` file in the `[appserver root]/server/all/lib` directory. The updated `jbossmq.jar` file is located in `thirdparty\jboss\patches` directory in the installation media.
3. Navigate to the `[appserver root]/docs/examples/jca` directory and copy the `mssql-ds.xml` file to the `[appserver root]/server/all/deploy` directory.
4. Open the `mssql-ds.xml` file in a text editor and change the `<jndi-name>` element to `DefaultDS` and the `<local-tx-datasource>` element with your SQL Server connection settings:

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

5. Save and close the file.
6. (**JBoss 4.0.3 SP1 only**) Open the `[appserver root]/server/all/conf/standardjaws.xml` file and change the `<type-mapping>` element to SQL Server:

```
<type-mapping>MS SQLSERVER2005</type-mapping>
```

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

```
<datasource>java:/DefaultDS</datasource>
<datasource-mapping>MS SQLSERVER2005</datasource-mapping>
<fk-constraint>>true</fk-constraint>
```

8. Remove the comment tags from the `<datasource-mapping>` tag to uncomment it. For example, remove "`<!--optional since 4.0 ...-->`" and the corresponding `-->` tag.
9. Save and close the file.
10. Open the `[appserver root]/server/all/conf/login-config.xml` file in a text editor and add the following lines within the `<policy>` element:

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

11. Replace the bold values with values that are specific to your database:
 - **database_name, database_username, and password:** The database values that the application server uses to access the database.
12. Save and close the file.
13. Delete the `[appserver root]/server/all/deploy-hasingleton/jms/hssql-jdbc2-service.xml` file, and copy the `[appserver root]/docs/examples/jms/mssql-jdbc2-service.xml` file to `[appserver root]/server/all/deploy-hasingleton/jms`.
14. Open the `mssql-jdbc2-service.xml` file and change `mbean code` to reference `MSSQLJDBCStateManager` and `DefaultDS`. The file is located in the `[appserver root]/server/all/deploy-hasingleton/jms/` directory.

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

```
15. Rename the `hsqldb-jdbc-state-service.xml` file to `mssql-jdbc-state-service.xml`. The file is located in the `[appserver root]/server/all/deploy-hasingleton/jms/` directory.
16. Open the `mssql-jdbc-state-service.xml` file and change `mbean code` to reference `MSSQLJDBCStateManager`. The file is located in the `[appserver root]/server/all/deploy-hasingleton/jms/` directory.

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

```
17. Restart JBoss.

Configuring DB2 for manually installed JBoss

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

- Obtain and copy the DB2 JDBC driver to the instance of JBoss where you will deploy LiveCycle ES.
- Create a data source file and deploy it to the instance of JBoss where you will deploy LiveCycle ES.
- Configure JBoss to use DB2 for JMS.

Caution: If the JMS data source is different from IDP_DS (for example, a separate database or MQSeries), some JMS messages may be lost upon recovery from a system failure. These messages may include jobs that are waiting to be processed or processes that previously failed. To avoid this issue, configure the JMS data source to use the same database instance as IDP_DS.

Configuring the DB2 data source

Before configuring the DB2 data source, you must have the database created on DB2. (See [“Creating a DB2 database” on page 44.](#))

► To install the DB2 database driver:

1. Obtain the `db2jcc.jar` and `db2jcc_license_cu.jar` files from the DB2 installation directories on your DB2 server.
2. Copy the files from the `adobe/livecycle/lib/db/db2` directory to the `[appserver root]/server/all/lib` directory.

► To obtain the latest `db2-jdbc2-service.xml` file:

1. Go to the [JBoss Download](#) site.
2. Under Servers/Runtime, click **JBoss Application Server**.
3. On the next page, click **Download** beside **4.2.0 GA**.
4. On the next page, click the **jboss-4.2.0.GA.zip** file.
5. After the file is downloaded, open `jboss-4.2.0.GA.zip` and extract the `db2-jdbc2-service.xml` file to the `[appserver root]/docs/examples/jms` directory.

► To create the DB2 data source file:

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

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

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

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

- **adobe:** The name of the database that stores the LiveCycle ES data. You will need to update the default value, `adobe`, with your database name.
 - **database_username** and **password:** The user name and password that the application server uses to access the database. You will need to update the default values, `adobe` and `adobe`, with the credentials for your database.
4. 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>
```
 - `com.celequest.metadata.metaDatasource` (BAM only):

```
<min-pool-size>1</min-pool-size>
<max-pool-size>20</max-pool-size>
```
 5. Save the file as `adobe-ds.xml` in the `[appserver root]/server/all/deploy` directory.
 6. Restart JBoss.

Setting DB2 as the JMS data source

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

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

► To set DB2 as the JMS data source:

1. Navigate to the `[appserver root]/server/all/deploy` directory and delete the `hsqldb-ds.xml` file.
2. Copy the `db2-ds.xml` file from the `[appserver root]/docs/examples/jca` directory to the `[appserver root]/server/all/deploy` directory.
3. Open the `[appserver root]/server/all/deploy/the db2-ds.xml` file in a text editor and modify the `<local-tx-datasource>` element with your DB2 connection settings:

```
<jndi-name>DB2DS</jndi-name>
<connection-url>jdbc:db2:database_hostname:50000/your_database
</connection-url>
<driver-class>COM.ibm.db2.jdbc.app.DB2Driver</driver-class>
<user-name>x</user-name>
<password>y</password>
```
4. Replace the bold values with values that are specific to your database:
 - **DB2DS:** Replace with `DefaultDS`.
 - **database_hostname:** The hostname of the server hosting the database.
 - **your_database:** The name of the DB2 database that stores the LiveCycle ES data.
 - **COM.ibm.db2.jdbc.app.DB2Driver:** Replace with `com.ibm.db2.jcc.DB2Driver`.

- **x** and **y**: The user name and password that the application server uses to access the database.
5. Save and close the file.
 6. (JBoss 4.0.3 SP1 only) Open the `[appserver root]/server/all/conf/standardjaws.xml` file and change the `<type-mapping>` element to DB2:

```
<type-mapping>DB2</type-mapping>
```
 7. Save and close the file.
 8. Open the `[appserver root]/server/all/conf/standardjbosscomp-jdbc.xml` file and change the following elements (not necessarily on consecutive lines):

```
<datasource>java:/DefaultDS</datasource>
<datasource-mapping>DB2</datasource-mapping>
<fk-constraint>true</fk-constraint>
```
 9. Remove the text: `<!--optional since 4.0 and ..` from the `<datasource-mapping>` tag to uncomment it.
 10. Open the `[appserver root]/server/all/conf/login-config.xml` file in a text editor and add the following text within the `<policy>` element:

```
<application-policy name = "DB2DbRealm">
  <authentication>
    <login-module code =
      "org.jboss.resource.security.ConfiguredIdentityLoginModule" flag =
      "required">
      <module-option name = "principal">database_name</module-option>
      <module-option name = "userName">database_username</module-option>
      <module-option name = "password">password</module-option>
      <module-option name = "managedConnectionFactoryName">
        jboss.jca:service=LocalTxCM, name=DefaultDS
      </module-option>
    </login-module>
  </authentication>
</application-policy>
```
 11. Replace the bold text with values that are specific to your database:
 - **database_name**, **database_username**, and **password**: The database values that the application server uses to access the database.
 12. Save and close the file.
 13. Delete the `hssql-jdbc2-service.xml` file located in the `[appserver root]/server/all/deploy-hasingleton/jms/` directory.
 14. Copy the `db2-jdbc2-service.xml` file from the `[appserver root]/docs/examples/jms` directory to the `[appserver root]/server/all/deploy-hasingleton/jms` directory.
 15. Open the `db2-jdbc2-service.xml` file and change DB2DS to DefaultDS. The file is located in the `[appserver root]/server/all/deploy-hasingleton/jms/` directory.

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

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

16. Save and close the file.
17. Rename the hsqldb-jdbc-state-service.xml file to db2-jdbc-state-service.xml. The file is located in the *[appserver root]/server/all/deploy-hasingleton/jms/* directory.
18. Restart JBoss.

Next steps

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

Index

A

- admin look-up, enabling anonymous for WebLogic 78
- Administrative Console, WebSphere, accessing 84
- Adobe Acrobat licensing restriction 34
- Adobe LiveCycle Data Services ES 7
- Adobe LiveCycle ES database
 - connectivity, configuring for JBoss 99
 - connectivity, configuring for WebLogic 81
 - connectivity, configuring for WebSphere 86
 - DB2, creating 44
 - MySQL, creating 46
 - Oracle, creating 38
 - SQL Server, creating 39
- Adobe LiveCycle PDF Generator ES, requirements for 23
- Adobe LiveCycle Reader Extensions ES Rights credentials 36
- anonymous admin look-up for WebLogic 78
- application servers
 - instance on WebSphere, creating 85
 - supported 17

B

- bc tool 22

C

- certificates, obtaining 37
- configuration files, JBoss 91
- configuring
 - database connectivity for JBoss 99
 - database connectivity for WebLogic Server 81
 - database connectivity for WebSphere 86
 - LDAP 28
 - WebLogic Server 78
- content management systems supported 19
- conventions, path name 8
- creating
 - application server instance on WebSphere 85
 - DB2 database 44
 - WebLogic Managed Server 77
 - WebLogic Server domain 75
- credentials, LiveCycle Reader Extensions ES 36
- CRLs, obtaining 37

D

- data services supported 18
- databases
 - combinations with platforms 10
 - connectivity, configuring for WebLogic Server 81
 - connectivity, configuring for WebSphere 86
 - DB2, creating 44
 - drivers supported 18
 - MySQL, creating 46
 - Oracle, creating 38

- SQL server, creating 39

- supported 17

- DB2 database
 - configuring for manually configured JBoss 109
 - creating 44
- digital certificates, obtaining 37
- domain, creating for WebLogic Server 75

E

- email servers supported 18
- environment variables
 - setting for JBoss 49, 89
 - setting for PDF Generator ES 24
 - setting for WebLogic 73
 - setting for WebSphere 82

F

- file formats, software support for 23
- Fix Pack, installing for WebSphere 6.1 82

G

- GNU bc tool 22

H

- hardware requirements 14
- Hardware Security Module 37

I

- installing
 - JBoss 50
 - native file conversion software 23
 - summary of tasks 10
 - WebLogic Server 73
 - WebSphere Application Server 82

J

- J2SE SDK, installing for JBoss 49, 89
- JBoss Application Server
 - configuring DB2 109
 - configuring MySQL 53, 59, 66, 99
 - configuring Oracle database 55, 102
 - configuring SQL Server 59, 105
 - installing 50
 - installing the J2SE SDK 49, 89
 - modifying configuration files 91
 - starting and stopping 51, 90
 - turnkey installation 7
- JDK support 17
- JMS, configuring for SQL Server for JBoss 62, 64, 107

L

- LDAP
 - configuring 28
 - server support 18

M

- Managed Server, creating for WebLogic 77
- manually installing JBoss 90
- MySQL
 - configuring for Adobe preconfigured JBoss 53, 59, 66
 - configuring for manually installed JBoss 99
 - database, creating 46
 - recommended server configuration 47
 - user account 47

N

- naming conventions, file path 8
- Node Manager, starting 76

O

- operating systems
 - additional requirements 22
 - combinations with databases 10
 - supported 16
- Oracle database
 - configuring for Adobe preconfigured JBoss 55
 - configuring for manually installed JBoss 102
 - creating 38

P

- PDF clients supported 20
- platform matrix 10
- platforms, combinations with databases 10
- private keys 37
- public keys 37

R

- Rights credentials, obtaining 36

S

- SDK

- installing J2SE for JBoss 49, 89
- installing J2SE for WebLogic 73
- security, required to run Digital Signatures ES 37
- Service Control Manager tool 25
- software support 16
- SQL Server
 - configuring data source for JBoss 60, 105
 - configuring for JBoss 59, 105
 - configuring JMS for JBoss 62, 64, 107
 - database, creating 39
- starting
 - JBoss 51, 90
 - Node Manager 76
 - WebLogic 80
- stopping
 - JBoss 51, 90
 - WebLogic 80
- system requirements 10, 14

T

- turnkey installation for JBoss 7

U

- user accounts
 - MySQL 47
 - PDF Generator ES 23
- user interfaces, supported web browsers for 21
- users, adding to DB2 database 44

W

- web browsers supported 21
- Web Services, configuring WebLogic Server for 79
- WebLogic Managed Server, creating 77
- WebLogic Server
 - configuring 78
 - configuring database connectivity for 81
 - domain, creating 75
 - installing 73
 - stopping and starting 80
- WebSphere Application Server
 - Administrative Console, accessing 84
 - configuring database connectivity for 86
 - installing 82