



Installing and Configuring LiveCycle for JBoss

September 2007

Adobe® LiveCycle™
Version 7.2

© 2007 Adobe Systems Incorporated. All rights reserved.

Adobe® LiveCycle™ 7.2 Installing and Configuring LiveCycle for JBoss® for Microsoft® Windows® and Linux®
Edition 1.3, September 2007

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 and company logos in sample material or in the sample forms included in this software are for demonstration purposes only and are not intended to refer to any actual organization.

Adobe, the Adobe logo, Acrobat, Kozuka Gothic, Kozuka Mincho, LiveCycle, Minion, Myriad, PhotoShop, PostScript, and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

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

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

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

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

MacIntosh is a trademark of Apple Computer, Inc., registered in the United States and other countries.

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

Novell and SUSE are registered trademarks of Novell, Inc. in the United States and other countries.

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

Red Hat and JBoss are trademarks or registered trademarks of Red Hat, Inc. in the United States and other countries.

Sun, Java, JavaScript, and Solaris 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.

All other trademarks are the property of their respective owners.

This software is based in part on the work of the Independent JPEG Group.

Portions copyright 1992, 1993 Simmule Turner and Rich Salz. All rights reserved.

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

Portions Copyright (C) 1991, 1999 Free Software Foundation, Inc. The JBOSS, OmniORB, and JacORB libraries are licensed under the GNU Library General Public License, a copy of which is included with this software.

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

This product includes software developed by Brian M. Clapper bmc@clapper.org.

Portions of this code are licensed from Apple Computer, Inc. under the terms of the Apple Public Source License, Version 2. The source code version of these portions and the license are available at <http://www.opensource.apple.com/apsl/>.

Portions based in part on the work of the FreeType team.

Powered by Celequest. Contains technology distributed under license from Celequest Corporation. Copyright 2005 Celequest Corporation. All rights reserved.

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

This Program was written with MacApp®: ©1985-1988 Apple Computer, Inc. APPLE COMPUTER, INC. MAKES NO WARRANTIES WHATSOEVER, EITHER EXPRESS OR IMPLIED, REGARDING THIS PRODUCT, INCLUDING WARRANTIES WITH RESPECT TO ITS MERCHANTABILITY OR ITS FITNESS FOR ANY PARTICULAR PURPOSE. The MacApp software is proprietary to Apple Computer, Inc. and is licensed to Adobe for distribution only for use in combination with Adobe software.

Portions licensed under the Mozilla Public License Version 1.1, available at www.mozilla.org. Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

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

Preface	8
What's in this guide?	8
Who should read this guide?	8
Conventions used in this guide.....	8
Related documentation	9
Updated LiveCycle product information.....	10
1 Before You Install	11
About the installation, configuration, and deployment process.....	11
Methods for installing, configuring, and deploying LiveCycle products.....	12
Upgrading LiveCycle products.....	12
About Watched Folder installation.....	12
System requirements	13
Supported software	13
Platform and software combinations	14
Minimum hardware requirements	15
Additional requirements for LiveCycle PDF Generator Elements and LiveCycle PDF Generator Professional	15
Additional requirements for LiveCycle PDF Generator for PostScript	16
Installation, configuration, and deployment checklists	16
Turnkey installation and deployment checklist.....	16
Manual installation and deployment checklist.....	17
Updated LiveCycle product information.....	17
 <i>Part I: Turnkey Installation</i>	
2 Installing LiveCycle Products Using the Turnkey Installation	19
Running the turnkey installation	19
Modifying LiveCycle Services	24
Next Step.....	24
3 Post-deployment	25
Accessing Administrator.....	25
Accessing User Management	26
LiveCycle Assembler post-deployment tasks	26
Verifying the LiveCycle Assembler deployment.....	26
LiveCycle Forms post-deployment tasks	27
Verifying the LiveCycle Forms deployment	27
LiveCycle Print post-deployment tasks	27
Verifying deployment.....	27
Deploying and running the PrintIVS web application.....	28
Running the Print Submitter application	28
Running the Form Server Module API Print application.....	29
LiveCycle Form Manager post-deployment tasks	30
Accessing the LiveCycle Form Manager end-user pages	30
LiveCycle PDF Generator post-deployment tasks	30
Setting up job sources	30

Setting Adobe PDF Printer as the default printer	30
Installing fonts.....	31
Setting the LiveCycle PDF Generator conversion time-out.....	31
Next Step.....	32

Part II: Manual Installation, Configuration, and Deployment

4 Installing LiveCycle Products.....	34
Installing LiveCycle PDF Generator.....	34
Installing LiveCycle products	36
Installing LiveCycle Print.....	38
Installing Watched Folder	39
Viewing the error log	40
Next steps.....	40
5 Preparing Your Environment.....	41
Creating the database	41
Creating a MySQL database	41
Creating an Oracle database.....	42
Creating a DB2 database	43
Configuring a DB2 database for concurrent usage	44
Creating a SQL Server database.....	45
Creating a JMS database on SQL Server	45
Preparing the application server	47
Installing JBoss Application Server	47
Next step.....	47
6 Configuring LiveCycle Products for Deployment	48
Next step.....	51
7 Manually Configuring JBoss	52
Starting and stopping JBoss.....	52
Modifying the JBoss configuration files	53
Configuring JBoss for Watched Folder	57
Connecting JBoss for LiveCycle products to the LiveCycle database	57
Configuring the MySQL data source	57
Configuring the Oracle data source	58
Configuring the DB2 data source.....	59
Configuring the SQL Server data source	60
Next step.....	61
8 Manually Deploying to JBoss.....	62
About deploying LiveCycle products.....	62
JBoss directory name.....	62
Summary of deployable components.....	62
Deploying to JBoss.....	63
Applying the connection pool patch for Watched Folder	64
Viewing log files.....	64
Next step.....	64
9 Initializing the Database	65
Next step.....	66
10 Configuring LiveCycle Products to Access LDAP	67

Configuring LiveCycle products with LDAP67
Configuring LiveCycle products with LDAPS68

Part III: Post-Deployment Configuration

11 Configuring SSL on JBoss..... 70
 Creating an SSL Credential70
 Configuring SSL on JBoss72
 Next step.....74

12 Manually Configuring JBoss for BAM Server..... 75
 Installing JBoss Application Server75
 Creating the BAM metadata database75
 User accounts75
 Creating a BAM metadata database in Derby76
 Configuring JBoss for BAM Server.....77
 Installing database drivers on JBoss for BAM Server.....77
 Connecting JBoss to the BAM metadata database.....77
 Connecting JBoss for BAM Server to the LiveCycle database.....81
 Configuring required JBoss JVM options83
 Optional JBoss JVM options84
 Modifying the JBoss thread configuration85
 Configuring JBoss logging.....85
 Customizing port numbers86
 Deploying BAM Server88
 Next step.....88

Part VI: Additional LiveCycle Workflow Configuration

13 Getting Started with BAM Server 90
 Configuring LiveCycle Workflow Server for BAM Server.....90
 Accessing BAM Workbench and BAM Dashboard91
 Configuring BAM Server91
 Configuring the SMTP settings91
 Importing LiveCycle Workflow metadata definitions.....92
 Starting the JDBC agent93
 Configuring LDAP settings for BAM Server93
 Next steps.....97

14 Installing LiveCycle Workflow Designer 98
 Installing LiveCycle Workflow Designer98
 Connecting to application servers using non-default ports.....99
 Uninstalling LiveCycle Workflow Designer99
 Next steps.....100

A Supported Platform and Software Combinations 101

B Fonts Installed with the Font Manager Module 104

C Invoking LiveCycle Assembler Using LiveCycle Workflow and Watched Folder 105
 Summary of tasks105
 Deploying QPACs and creating a workflow process106
 Using dynamic or static DDX files106
 Configuring an Assembler QPAC in a workflow process.....107

Creating and configuring a watched folder	109
Creating a JobConfig.xml file	110
Preparing PDF and DDX files	112
Submitting the LiveCycle Assembler job for processing	113
D Developing Forms for LiveCycle	114
Publishing files from client software	114
Designing forms for LiveCycle products	114
Using dynamic forms with LiveCycle products	115
Embedding fonts in PDF/A-compliant forms	115
E Uninstalling LiveCycle Products	116
Removing the product files from a turnkey installation	116
Removing the product files from a manual installation	117
F Upgrading LiveCycle Products to Version 7.2 or 7.2.1	118
Upgrade guidelines	118
Updating your application server	118
Using automatic or turnkey installations for upgrading	119
Configuring using Configuration Manager during the upgrade process	119
Summary of manual upgrade process	119
LiveCycle Forms and LiveCycle Print	120
Upgrading from Adobe Form Server 6.0 to LiveCycle Forms 7.2	120
LiveCycle Forms and LiveCycle Print 7.x to LiveCycle Forms and LiveCycle Print 7.2	121
LiveCycle Form Manager	124
LiveCycle Assembler, LiveCycle Workflow, and Watched Folder	127
LiveCycle Workflow Designer	129
BAM Server for LiveCycle Workflow	130
About the BAM Server upgrade utility	130
LiveCycle PDF Generator	132
LiveCycle Document Security	134
LiveCycle Reader Extensions	137
LiveCycle Policy Server	140
G Enhancing Server Performance	141
Optimizing inline documents and impact on JVM memory	141
Cleaning up temporary files in the Global storage directory	142
DB2 configuration settings	142
Improving Windows Server Performance with LDAP	143
Index	144

Preface

This guide is one of several resources available to help you learn about the Adobe® LiveCycle™ suite of products. LiveCycle products help automate and accelerate the flow of business-critical information to and from customers, partners, constituents, and employees.

What's in this guide?

This guide provides information about how to install and configure the following LiveCycle products on Microsoft® Windows® and Linux® and how to deploy the products to a JBoss® Application Server:

- Adobe LiveCycle Assembler 7.2.1
- Adobe LiveCycle Forms 7.2
- Adobe LiveCycle Form Manager 7.2
- Adobe LiveCycle PDF Generator 7.2 Professional, LiveCycle PDF Generator 7.2 Elements, and LiveCycle PDF Generator 7.2 for PostScript®
- Adobe LiveCycle Print 7.2
- Adobe LiveCycle Workflow 7.2.1

This guide also provides information about how to install and configure Watched Folder 1.1.

Who should read this guide?

This guide provides information for administrators or developers responsible for installing, configuring, administering, or deploying LiveCycle products. The information provided is based on the assumption that anyone reading this guide is familiar with application servers, Linux and Windows operating systems, MySQL, Oracle®, DB2®, or SQL Server database servers, and web environments.

Conventions used in this guide

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

Name	Default value	Description
<i>[LiveCycle root]</i>	Windows: C:\Adobe\LiveCycle\ Linux: /opt/adobe/livecycle/	The installation directory that is used for all LiveCycle products. The installation directory contains subdirectories for Adobe Configuration Manager, product SDKs, and each LiveCycle product installed (along with the product documentation).

Name	Default value	Description
[product root]	Windows: C:\Adobe\LiveCycle\Assembler C:\Adobe\LiveCycle\pdfgenerator C:\Adobe\LiveCycle\Workflow C:\Adobe\LiveCycle\Forms C:\Adobe\LiveCycle\Print C:\Adobe\LiveCycle\Formmanager Linux: /opt/adobe/livecycle/Assembler /opt/adobe/livecycle/pdfgenerator /opt/adobe/livecycle/workflow /opt/adobe/livecycle/forms /opt/adobe/livecycle/print /opt/adobe/livecycle/formmanager	The directories where product-specific directories and files (such as documentation, uninstall files, samples, and license information) are located.
[appserver root]	Windows (Turnkey Installation): C:\Adobe\LiveCycle\jboss\ Windows (Manual Installation): C:\jboss Linux: /opt/jboss	The home directory of the application server that runs the LiveCycle products.
[jboss bam root]	Directory where you have installed JBoss.	The home directory of the instance of JBoss that runs Business Activity Monitor (BAM) Server.

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

Related documentation

This guide contains instructions for deploying LiveCycle products to JBoss. The *Installing and Configuring LiveCycle* guides for other supported application servers can be accessed at: www.adobe.com/support/documentation/en/livecycle/

The resources in this table can help you learn about and get started using LiveCycle products.

For information about	See
General information about a product and how it integrates with other Adobe products	<i>Overview</i> guides for each product
The product architecture, how to use the APIs, and how to develop custom applications for use with the products	The developer guides for each product. For example, <i>Developing Custom Application for LiveCycle Workflow</i> or <i>Developing Applications for LiveCycle Assembler</i> .

For information about	See
The EJB API, including descriptions and explanations of its classes and methods.	The API Reference for each product. Most API References are installed as JavaDocs with each product. However, some API references are provided as PDF documents with the product.
The syntax for the Document Description XML (DDX) grammar and related XML grammars supported by LiveCycle Assembler	<i>Document Description XML Reference</i>
Setting up and administering Watched Folders	<i>Watched Folder Administration Help</i>
Managing access to the Adobe Administrator user interface	<i>User Management Administration Help</i>
How to use LiveCycle Workflow Designer	<i>Creating Workflows or LiveCycle Workflow Designer Help</i>
Using the LiveCycle Workflow SDK to create Quick Process Action Components (QPACs) that invoke methods in the LiveCycle product EJB APIs	The QPAC guides, available with the LiveCycle Workflow SDK. For example <i>Using LiveCycle Assembler QPACs</i> or <i>Using LiveCycle Forms QPACs</i> .
Other services and products that integrate with LiveCycle products	www.adobe.com
Patch updates, technical notes, and additional information on this product version	www.adobe.com/support/products/enterprise/index.html

For information about additional documentation available for each of the LiveCycle products, see the doc_map.pdf files located in each [product root]/documentation folder.

Note: When copying and pasting strings directly from this guide, you may copy the tag <CR> if the instructions span a line in the guide.

Updated LiveCycle product information

Adobe Systems has posted a Knowledge Center article to communicate any updated LiveCycle product information with customers. You can access the article at:

www.adobe.com/support/products/enterprise/knowledgecenter/c4811.pdf.

This chapter describes how to prepare your system for installing LiveCycle products:

- [“About the installation, configuration, and deployment process” on page 11](#)
- [“Methods for installing, configuring, and deploying LiveCycle products” on page 12](#)
- [“Installing and deploying multiple LiveCycle products” on page 12](#)
- [“System requirements” on page 13](#)
- [“Installation, configuration, and deployment checklists” on page 16](#)
- [“Updated LiveCycle product information” on page 17](#)

Before you begin installing LiveCycle products on your application server, visit the Adobe LiveCycle product download page at the following location to make certain you have the latest version of the software:

www.adobe.com/support/products/enterprise/support_downloads.html

About the installation, configuration, and deployment process

Installing, configuring, and deploying LiveCycle products involves the following processes:

Installing: Installing the products places all of the required files onto your computer, within one installation directory structure. You install the products by running the installation program. The default installation directory is C:\Adobe\LiveCycle (Windows) or /opt/adobe/livecycle (Linux); however, you can install the files to a different directory. In this guide, the default installation directory is referred to as *[LiveCycle root]*. In order for multiple LiveCycle products to work with one another, you must install all of the products in the same *[LiveCycle root]* location. This enables you to assemble the multiple LiveCycle products into one EAR file. (See [“Installing LiveCycle Products” on page 34.](#))

Configuring and assembling: Configuring the products modifies a variety of settings that determine how the products work. Assembling the products packages all of the installed components that the products need into a deployable EAR file, according to your configuration instructions. You configure and assemble the products for deployment by running Configuration Manager. (See [“Configuring LiveCycle Products for Deployment” on page 48.](#)) You can configure and assemble multiple LiveCycle products at the same time.

Deploying: Deploying the products involves deploying the assembled EAR file and a few other configured files to the JBoss application server on which you plan to run your LiveCycle solution. If you have configured and assembled multiple products, most of the deployable components for the multiple products are packaged within the single deployable LiveCycle.ear file.

Initializing the LiveCycle database: Initializing the LiveCycle database creates tables for use with Adobe User Management and certain LiveCycle products. Deploying any LiveCycle product that connects to the LiveCycle database requires you to initialize the LiveCycle database after the deployment process. (See [“Initializing the Database” on page 65.](#))

Methods for installing, configuring, and deploying LiveCycle products

You can use one of the following methods for installing, configuring, and deploying LiveCycle products as well as initializing the database:

Turnkey: (Windows/JBoss/MySQL) The turnkey method lets you install the files, and then run Configuration Manager to configure the EAR file and other components and automatically performs the following tasks:

- Install and configure the JBoss Application Server and MySQL database.
- Assemble and deploy the products to JBoss.
- Initialize the MySQL database.

The turnkey method does not configure SSL. (See [“Turnkey installation and deployment checklist” on page 16](#) or [“Installing LiveCycle Products Using the Turnkey Installation” on page 19](#).)

Manual: The manual method lets you install the files, and then run Configuration Manager to configure the application server, the EAR file and other components, and deploy the EAR file. You can also choose to configure your application server and deploy your EAR file manually. However, you must manually install and, once configured, start your application server as well as create and configure the database before running Configuration Manager and deploying to the application server. You must also run Configuration Manager a second time (after deployment) to initialize the database. (See [“Manual installation and deployment checklist” on page 17](#).)

Installing and deploying multiple LiveCycle products

To deploy more than one of the LiveCycle products discussed in this guide so that they interoperate, you need to install the products in the same location, assemble them in a single EAR file, and then deploy the EAR file.

Upgrading LiveCycle products

For information on upgrading LiveCycle products, refer to [“Upgrading LiveCycle Products to Version 7.2 or 7.2.1” on page 118](#) in this guide.

About Watched Folder installation

Both LiveCycle Assembler and LiveCycle PDF Generator can use watched folder functionality as a method for initiating jobs for processing. However, LiveCycle Assembler and LiveCycle PDF Generator use different watched folder technologies.

LiveCycle Assembler uses a service called *Watched Folder*, which you install separately and configure with LiveCycle Workflow using Configuration Manager. This service is included in the deployable LiveCycle.ear file.

LiveCycle PDF Generator uses an embedded watched folder functionality, which does not require a separate installation. The configuration tasks required for LiveCycle PDF Generator watched folders is included in the [“LiveCycle PDF Generator post-deployment tasks” on page 30](#).

All other installation and configuration references to Watched Folder in this guide pertain to the service that is used by LiveCycle Workflow and LiveCycle Assembler.

System requirements

This section includes details about the software and hardware that is required for running LiveCycle products.

Note: If you are running the turnkey installation and configuration, the only prerequisite software requirement is the Java™ Development Kit (JDK) and a supported web browser. The turnkey method installs and configures JBoss (which includes the Apache web server) and a MySQL database.

The JDK is not a prerequisite for LiveCycle PDF Generator. If the JDK is not already installed, the turnkey installer will install it.

Supported software

This table provides a summary of the application servers, web browsers, and JDK versions that LiveCycle products support. For a complete list, see [“Supported Platform and Software Combinations” on page 101](#).

Required software	Supported version
Operating system	<ul style="list-style-type: none"> ● Microsoft Windows Server™ 2003 Enterprise Edition or Standard Edition ● (LiveCycle Workflow Designer) Microsoft Windows XP Service Pack 2 ● SUSE™ Linux Enterprise Server 9.0 i386 (32-bit) ● Red Hat® Linux Advanced Server 2.1 or 3.0 <p>Note: LiveCycle PDF Generator Elements and LiveCycle PDF Generator Professional are only supported on Microsoft Windows Server 2003.</p>
Application server	<ul style="list-style-type: none"> ● JBoss Application Server 3.2.5 ● JBoss Application Server 3.2.6 (Applicable only for BAM Server, which is a component of LiveCycle Workflow Server) <p>Note: Your application server must have an active Internet connection to access LiveCycle Administrator.</p>
Web browser	<ul style="list-style-type: none"> ● Microsoft Internet Explorer 6.0 for Windows ● Netscape 7.1 or higher for Windows ● Netscape 7.2 or higher for Linux ● Mozilla 1.8 or higher for Windows and Linux ● Safari 1.2.3, Safari 1.3, Safari 2.0 (end-user support for Macintosh only) ● (LiveCycle Forms) Firefox 1.0 <p>Note: LiveCycle PDF Generator only supports Microsoft Internet Explorer 6.0.</p>
JDK	<ul style="list-style-type: none"> ● J2SDK version 1.4.2_04 or later version of the 1.4.2 release (version 1.4.2_10 is not supported) <p>You must create or set the <code>JAVA_HOME</code> environment variable to point to the location where Java is installed. Ensure that the Java 2 Standard Edition (J2SE) installation <code>\bin</code> directory is in the <code>PATH</code> environment variable. (This is not required for LiveCycle PDF Generator and LiveCycle Assembler.)</p>

Required software	Supported version
Database	<ul style="list-style-type: none"> ● Microsoft SQL Server 2000 SP 3 ● MySQL 4.1 ● (Turnkey) HSQLDB 1.7.1 used as the JBoss JMS database (HSQLDB 1.7.1 is packaged with JBoss 3.2.5) ● Oracle 9i ● Oracle 10g
Database driver	<ul style="list-style-type: none"> ● IBM DB2 - db2jcc.jar and db2jcc_license_cu.jar ● Microsoft SQL Server 2000 - msbase.jar, mssqlserver.jar, and msutil.jar ● MySQL - mysql-connector-java-3.0.15-ga-bin.jar ● Microsoft SQL Server 2000 for BAM - msbase.jar, mssqlserver.jar, and msutil.jar ● Oracle - ojdbc14.jar, version 10.2.0.1
LDAP server	<ul style="list-style-type: none"> ● Sun ONE 5.1, 5.2 ● Microsoft Active Directory 2000 ● Microsoft Active Directory 2003 ● Novell® eDirectory 8.7

Note: If you run the turnkey installation, you must not have instances of JBoss Application Server or the MySQL database server running on the target computer.

Platform and software combinations

This table summarizes the software combination and database combination supported for JBoss 3.2.5. For a complete list of supported software on each operating system, see [“Supported Platform and Software Combinations” on page 101](#).

Note: LiveCycle PDF Generator Elements and LiveCycle PDF Generator Professional only run on the Windows Server 2003 Enterprise Edition platform.

Operating system	Database
Red Hat Linux Advanced Server 3.0	Microsoft SQL Server 2000 SP 3
Microsoft Windows Server 2003 Enterprise Edition or Standard Edition	MySQL 4.1

Minimum hardware requirements

The table in this section lists the supported operating systems and corresponding hardware. For any installation, these settings are recommended as a minimum:

- Disk space for installation: 3 GB per product, except LiveCycle PDF Generator, which requires 3.5 GB
- System temp space during installation: 3 GB
- Memory for running the products: 1 GB per product for each CPU, except LiveCycle PDF Generator, which requires 1.5 GB per CPU

Operating system	Minimum hardware requirement
Windows Server 2003 Enterprise Edition or Standard Edition	Intel® Pentium® 3 or x86 equivalent, 1GHz processor
SUSE Linux Enterprise Server 9.0 i386 (32-bit)	Pentium 3 or x86 equivalent, 1GHz processor
Red Hat Linux Advanced Server 2.1 or 3.0	Pentium 3 or x86 equivalent, 1GHz processor

Additional requirements for LiveCycle PDF Generator Elements and LiveCycle PDF Generator Professional

LiveCycle PDF Generator Elements and LiveCycle PDF Generator Professional must be installed on Microsoft Windows Server 2003 with the en_US locale. The Windows user who installs the product must have administrative rights, and must be the same user who installed Microsoft Office on that computer.

Before installing LiveCycle PDF Generator Elements or LiveCycle PDF Generator Professional, you must ensure that Adobe Acrobat® Professional or Acrobat Standard is not installed. If Acrobat is installed, you must uninstall it and restart the computer. However, Adobe Reader® can be installed.

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

- Microsoft Office 2000, XP, or 2003
- (LiveCycle PDF Generator Professional) Microsoft Office Visio 2003
- (LiveCycle PDF Generator Professional) Microsoft Project 2003
- (LiveCycle PDF Generator Professional) AutoCAD 2005
- Corel WordPerfect 12
- Adobe Photoshop® CS2

Before completing a turnkey installation of LiveCycle PDF Generator on Windows, ensure that the Service Control Manager command line tool, sc.exe, is installed in the Windows environment variable. Some Windows servers do not have this software preinstalled. By default, the sc.exe file is installed in the \Windows\system32 directory.

LiveCycle PDF Generator Professional and LiveCycle PDF Generator Elements must be installed on the same server where it needs to be configured and deployed.

Additional requirements for LiveCycle PDF Generator for PostScript

LiveCycle PDF Generator for PostScript is supported on all of the operating systems listed in [“Supported software” on page 13](#).

To install LiveCycle PDF Generator for PostScript on Linux, ensure that the bc tool (an arbitrary precision calculator language) is installed on the computer. You can download the GNU bc tool from www.gnu.org/software/bc/bc.html.

Installation, configuration, and deployment checklists

This section includes checklists that you can use to step through the installation and configuration process. A checklist is provided for installing and configuring when using either the turnkey method or the manual method.

Before starting the installation, ensure that the JAR files are not associated with WinZip or any other application other than the Java application launcher.

Turnkey installation and deployment checklist

The following table includes the steps required for installing LiveCycle products using the turnkey method.

Perform this type of installation and configuration if you do not yet have an application server installed and configured and you want to use JBoss, or if you want to quickly and easily get the products installed and configured for testing or demonstration purposes.

Do not use the turnkey configuration method if you plan to enable SSL. If you require SSL, perform a manual configuration. (See [“Configuring LiveCycle Products for Deployment” on page 48](#).)

Task	Topic
<input type="checkbox"/> Ensure that you have the required software installed in the target environment.	“System requirements” on page 13
<input type="checkbox"/> Run the installation program with the turnkey option enabled for JBoss. The product, JBoss, and MySQL are installed and configured.	“Installing LiveCycle Products Using the Turnkey Installation” on page 19
<input type="checkbox"/> Access Administrator and User Management.	“Accessing Administrator” on page 25
<input type="checkbox"/> Check the log file.	“Viewing log files” on page 64

Manual installation and deployment checklist

The following table includes the steps required for installing LiveCycle products using the manual method. Your application server must be installed before you perform the installation.

Perform this type of installation if you are installing the product in a production environment.

Note: If you are installing multiple products, ensure that they are all installed before running Configuration Manager to configure and deploy them.

Task	Topic
<input type="checkbox"/> Ensure that you have the required software installed in the target environment.	"System requirements" on page 13
<input type="checkbox"/> Create the database to use with User Management and install and prepare the application server.	"Preparing Your Environment" on page 41
<input type="checkbox"/> Run the installation program.	"Installing LiveCycle Products" on page 34
<input type="checkbox"/> Run Configuration Manager and select the Custom Configuration Wizard. This will configure and assemble the products.	"Configuring LiveCycle Products for Deployment" on page 48
<input type="checkbox"/> Configure JBoss settings. A variety of settings must be configured.	"Manually Configuring JBoss" on page 52
<input type="checkbox"/> Deploy the product deployment files to the application server.	"Manually Deploying to JBoss" on page 62
<input type="checkbox"/> Run Configuration Manager to initialize the database.	"Initializing the Database" on page 65
<input type="checkbox"/> Access Administrator and User Management.	"Accessing Administrator" on page 25
<input type="checkbox"/> Configure LDAP access.	"Configuring LiveCycle Products to Access LDAP" on page 67
<input type="checkbox"/> Configure SSL on the application server, if required.	"Configuring SSL on JBoss" on page 72
<input type="checkbox"/> (LiveCycle Workflow) Create the BAM metadata database and configure the application server for BAM Server.	"Manually Configuring JBoss for BAM Server" on page 75
<input type="checkbox"/> Check the log file.	"Viewing log files" on page 64

Updated LiveCycle product information

Adobe Systems has posted a Knowledge Center article to communicate any updated LiveCycle product information with customers. You can access the Knowledge Center article at: www.adobe.com/support/products/enterprise/knowledgecenter/c4811.pdf

Part I: Turnkey Installation

This section of the guide describes how to complete a turnkey installation of LiveCycle products

For information on the manual configuration and deployment of the products, see [“Manual Installation, Configuration, and Deployment” on page 33](#).

2

Installing LiveCycle Products Using the Turnkey Installation

This chapter describes how to install LiveCycle products using the turnkey method. The turnkey installation automatically performs all of the tasks required to install and configure LiveCycle products on a JBoss Application Server running on Windows.

The turnkey installation performs the following tasks:

- Installs the product files
- Installs a preconfigured version of JBoss 3.2.5 (with Apache web server embedded)
- Installs a preconfigured version of MySQL 4.1
- (LiveCycle PDF Generator) Installs a JDK if not present on the target system
- Starts Configuration Manager
- Configures and assembles the LiveCycle product components
- Deploys all of the required components to JBoss
- Initializes MySQL when installing LiveCycle products that require User Management

Running the turnkey installation

Before running the turnkey installation, ensure that the environment where you are installing and deploying LiveCycle products meets the system requirements. (See the note under [“System requirements” on page 13.](#)) This includes manually installing the required version of the JDK. During the turnkey installation, you will be prompted for the JDK root directory. The turnkey installation then installs JBoss and MySQL automatically.

The turnkey installation specifies “localhost” as the host name, “8080” as the port for use by JBoss, and “3306” as the port for MySQL. If JBoss and MySQL are already installed, ensure that they are not using these ports. You cannot configure an alternative host name or port for JBoss or MySQL during the turnkey installation and configuration process.

By default, the turnkey installation places the LiveCycle product and all of the related components and software in the \Adobe\LiveCycle directory (referred to as *[LiveCycle root]*).

Note: To successfully install, you need read and write permissions on the installation directory. The default installation directory is C:\Adobe\LiveCycle\, although you can specify a different directory as required.

JBoss and MySQL are installed and run from the *[LiveCycle root]/jboss* and *[LiveCycle root]/mysql* directories, but you can specify another directory by either typing in the path or browsing to it.

The turnkey installation creates the following Windows services:

- JBoss for Adobe LiveCycle
- MySQL for Adobe LiveCycle
- (LiveCycle Workflow) JBoss for Adobe LiveCycle Workflow BAM

These services are used by Configuration Manager during the turnkey installation. The services can be stopped and started using the Services window in the Administrative Tools area of the Windows Control Panel.

Note: If JBoss and MySQL are already installed, you must stop their services before running the turnkey installation.

For the turnkey installation, it is recommended that you accept the default configuration options. If you prefer to set all of the configuration options, run the manual installation, configuration, and deployment process. For information about the configuration options, see [“Configuring LiveCycle Products for Deployment” on page 48](#).

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

Use the following procedure to install all products except LiveCycle PDF Generator. (See [“To run the turnkey installation for LiveCycle PDF Generator:” on page 22](#))

➤ **To run the turnkey installation:**

1. At the root level of the installation DVD, double-click the .exe file.
2. If prompted, select a language for the installation program and click **OK**.
3. Click **Next** on the Welcome screen.
4. On the Before You Install screen, click **Next**.
5. Type the serial number in the text box and click **Next**.
6. Read the Product License Agreement, select **I accept the terms of the license agreement**, and then click **Next**.

Note: Depending on the product you are installing, the next steps may appear in a different order.

7. Select **JBoss**, select **Configure and deploy the product automatically**, and then click **Next**.
8. Read the license information associated with installing JBoss, select **I accept the terms of the license agreement**, and then click **Next**.
9. Read the license information associated with installing MySQL, select **I accept the terms of the license agreement**, and then click **Next**.
10. If required, type the path to your JDK or click **Browse** to navigate to its root directory, and then click **Next**.
11. Accept the default Adobe directory location as listed or click **Browse** and navigate to the directory where you want to install the product, and then click **Next**.

Note: If you type in the name of a directory that does not exist, it will be created for you.

12. Review the installation details, and then click **Install**. A summary screen appears when the product installation is completed.

13. On the summary screen, you have the following options:
 - If you are installing only one LiveCycle product or this is the last LiveCycle product you are installing, select **Launch the Configuration Manager**, and then click **Finish**. Proceed to step [14](#).
 - If you plan to install additional LiveCycle products, click **Finish** to exit the installation program and run the installation programs for the additional products.
14. If prompted, select a language for Configuration Manager and click **OK**.
15. On the Welcome screen, click **Next**.
16. On the Configuration Mode screen, select **Typical Configuration Wizard** and click **Next**.
17. If prompted, on the Configuration Preferences screen, select either **Use Previously Entered Values** or **Revert to Default Values**, and then click **Next**.
Note: This screen only appears if you have previously run Configuration Manager.
18. If required, on the Product Selection screen, select **JBoss** as the type of application server that you are using.
19. If required, select **Foundation** and the products that you want to configure and deploy, and then click **Next**.
20. (LiveCycle Forms) On the Adobe User Management Selection screen, select either **LiveCycle Forms with User Management and Administration** or **LiveCycle Forms without User Management and Administration**, and then click **Next**.
21. On the Configuration and Assemble Products screen, click **Next**.
22. Review the Application Configuration and Assembly Summary details, and then click **Next**. Configuration Manager configures the LiveCycle product properties using the default values and assembles the products into the LiveCycle.ear file.
23. When the configuration and assembly is complete, click **Next**.
24. On the Deploy LiveCycle screen, click **Next**.
25. On the Deploy products summary screen, click **Next**.
26. In the **Application Server Base URL** box, type the URL that Configuration Manager can use to connect to the application server, and then click **Next**. The URL must be in the following format:

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

where *[host name]* is the name or IP address of the computer that hosts the application server, and *[port]* is the HTTP service port that the application server uses (for example, *http://localhost:8080*).
27. On the Prepare database for LiveCycle screen, click **Next**.
28. On the Deployment Instructions screen, click **Next**.
29. On the Database Initialization screen, click **Initialize Database Now** to connect your LiveCycle products to the database, and then click **Next**.

30. On the Database Initialization screen, click **Initialize Database Now** and, when the task is complete, click **Next**.

31. On the Tasks Completed screen, click **Finish**.

The product is now deployed to JBoss, and JBoss should be running. If JBoss is not running, you can start the JBoss for Adobe LiveCycle service by using the Windows Service utility.

By default, the JBoss and MySQL services are installed with the "Manual" startup type. You can modify this by selecting Start > Administrative Tools > Services, locating these services and changing them to "Automatic". Otherwise, when you restart your machine, you will need to start the JBoss and MySQL services manually.

The JBoss log file (server.log) is located in the *[LiveCycle root]/jboss/server/all/log* directory. This file contains information on JBoss activity and progress.

► **To run the turnkey installation for LiveCycle PDF Generator:**

1. At the root level of the installation DVD, double-click the .exe file.
2. If prompted, select a language for the installation program and click **OK**.
3. Click **Next** on the Welcome screen.
4. Type the serial number in the text box and click **Next**.
5. (LiveCycle PDF Generator Professional, LiveCycle PDF Generator Elements) On the Adobe LiveCycle PDF Generator 7.2 pre-install requirements screen, review the preinstallation requirements and click **Next**.
6. If you are prompted to disable on-access virus scanning software, click **Next**. It is recommended that you disable any virus scanning software during the installation process to improve the speed of the installation.
7. Read the Product License Agreement, select **I accept the terms of the license agreement**, and then click **Next**.
8. Accept the default Adobe directory location as listed or click **Browse** and navigate to the directory where you want to install the product, and then click **Next**.

Note: If you type in the name of a directory that does not exist, it will be created for you.

Note: If you are installing multiple LiveCycle products, they must all be installed in the same *[LiveCycle root]* directory.

9. Select **JBoss**, select **Configure and deploy the product automatically**, and then click **Next**.
10. If required, type the path to your JDK or click **Browse** to navigate to its root directory, and then click **Next**.
11. Read the license information associated with installing JBoss, select **I accept the terms of the license agreement**, and then click **Next**.
12. Read the license information associated with installing MySQL, select **I accept the terms of the license agreement**, and then click **Next**.

13. (LiveCycle PDF Generator Professional, LiveCycle PDF Generator Elements) Enter the Microsoft Office administrator user name and password (if installed), and then click **Next**.
14. (LiveCycle PDF Generator Professional, LiveCycle PDF Generator Elements) On the Corel WordPerfect screen, browse to the location of the Corel WordPerfect executable file (if installed) and click **Next** or, if Corel WordPerfect is not installed, click **Next**.
15. (LiveCycle PDF Generator Professional, LiveCycle PDF Generator Elements) On the Adobe Photoshop screen, browse to the location of the Adobe Photoshop executable file (if not installed in the default directory) and click **Next** or, if Photoshop is not installed, click **Next**.
16. Review the installation details, and then click **Install**. A summary screen appears when the product installation is completed.
17. On the summary screen you have the following options:
 - If you are installing only one LiveCycle product or this is the last LiveCycle product you are installing, select **Launch the Configuration Manager**, and then click **Finish**. Proceed to step [18](#).
 - If you plan to install additional LiveCycle products, click **Finish** to exit the installation program and run the installation programs for the additional products.
18. If prompted, select a language for Configuration Manager and click **OK**.
19. On the Welcome screen, click **Next**.
20. Select **Typical Configuration Wizard** and click **Next**.
21. If prompted, on the Configuration Preferences screen, select either **Use last entered values** or **Revert to the default values**, and then click **Next**.

Note: This screen only appears if you have previously run Configuration Manager.
22. If required, select **JBoss** as the type of application server that you are using.
23. Select **Foundation** and the products that you want to configure and deploy, and then click **Next**.
24. On the Configure and Assemble Summary screen, click **Next**.
25. Review the configuration summary details, and then click **Next**. Configuration Manager configures the LiveCycle product properties using the default values and assembles the products into the LiveCycle.ear file and the additional pdfg-all.ear or pdfg-ps-all.ear file.
26. On the Configure and Assemble Products screen, click **Next**.
27. In the **Application Server Base URL** box, type the URL that Configuration Manager can use to connect to the application server. The URL must be in the following format:

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

where *[host name]* is the name or IP address of the computer that hosts the application server, and *[port]* is the HTTP service port that the application server uses (for example, `http://localhost:8080`).
28. On the Deploy Products screen, click **Next**.
29. On the Deployment Instructions screen, click **Next**.

30. On the Database Initialization screen, click **Initialize Database Now** to connect your LiveCycle products to the database, and then click **Next**.
31. On the Tasks Completed screen, click **Finish**.
The product is now deployed to JBoss, and JBoss should be running. If JBoss is not running, you can start the JBoss for Adobe LiveCycle service by using the Windows Service utility.
The JBoss log file (server.log) is located in the `[LiveCycle root]/jboss/server/all/log` directory. This file contains information on JBoss activity and progress.

Modifying LiveCycle Services

By default, the turnkey installation configures the JBoss for Adobe LiveCycle and MySQL for Adobe LiveCycle services to require manual startup. If you want to set these services to start up automatically, perform the following procedure.

► **To modify LiveCycle services:**

1. Select **Start > All Programs > Administrative Tools > Services**.
2. Right-click the JBoss for Adobe LiveCycle service and click **Properties**.
3. From the Startup Type menu, select **Automatic**, and then click **OK**.
4. Right-click the MySQL for Adobe LiveCycle service and click **Properties**.
5. From the Startup Type menu, select **Automatic**, and then click **OK**.
6. (LiveCycle Workflow Business Activity Monitor) Right-click the JBoss for Adobe LiveCycle Workflow BAM and click **Properties**.
7. From the Startup Type menu, select **Automatic**, and then click **OK**.

Next Step

For LiveCycle Forms and LiveCycle Assembler, you can now deploy a sample application to test the installation and deployment. (See [“Verifying the LiveCycle Forms deployment” on page 27](#) and [“Verifying the LiveCycle Assembler deployment” on page 26](#).)

You can also now access Administrator and User Management and begin developing applications for your LiveCycle product. (See [“Post-deployment” on page 25](#) and the developer guides for the products you have installed.)

3

Post-deployment

This chapter describes how to get started using your LiveCycle products after they have been installed, configured, and deployed to your application server:

- [“Accessing Administrator” on page 25](#)
- [“Accessing User Management” on page 26](#)
- [“LiveCycle Assembler post-deployment tasks” on page 26](#)
- [“LiveCycle Forms post-deployment tasks” on page 27](#)
- [“LiveCycle Print post-deployment tasks” on page 27](#)
- [“LiveCycle Form Manager post-deployment tasks” on page 30](#)
- [“LiveCycle PDF Generator post-deployment tasks” on page 30](#)

Accessing Administrator

Administrator is the web-based portal for accessing a variety of configuration pages that let you set run-time properties that control the way LiveCycle products operate. When you log into Administrator, you can access User Management, Adobe JMX Monitor, Watched Folder configuration (installed with Watched Folder), Process Manager and server settings (installed with LiveCycle Workflow), and administrative configuration options for other LiveCycle products.

The default user name and password for logging into Administrator is *administrator* and *password*. After you log in the first time, you can access User Management and change the password.

Before you access Administrator, the LiveCycle products must be deployed and running on your application server.

For information on using Administrator, see *Adobe Administration Help* (available from the Help menu of the Administrator window).

► To access Administrator:

1. Type the following URL in a web browser:

```
http://[host name]:[port]/adminui
```

The default port number for JBoss is 8080.

2. In the **User Name** field, type `administrator` and, in the **Password** field, type `password`.
3. After logging in, you can click **Services** to access the LiveCycle product Administrator pages, or **Settings** to access the User Management pages.

Accessing User Management

User Management allows administrators to maintain a database for all users and groups that are synchronized with one or more third-party user directories. User Management provides authentication, authorization, and user management for LiveCycle products, including LiveCycle PDF Generator, LiveCycle Assembler, LiveCycle Form Manager, LiveCycle Forms, and LiveCycle Workflow.

User Management allows you to enable single sign-on (SSO) between LiveCycle products and Netegrity SiteMinder protected applications using Security Assertion Markup Language (SAML). When SSO is implemented, the LiveCycle user login pages are not required and do not appear if the user has already been authenticated via their company portal.

For information about using User Management, see *Adobe User Management Administration Help* (available from the Help menu of the User Management window).

► To access User Management:

1. From the home page of Administrator, click **Settings**.
2. On the Settings page, click **User Management**.

Note: For information about configuring users with User Management, click **User Management Help** in the upper right corner of the User Management page.

► (LiveCycle Forms) To access Administrator without User Management:

- Type the following URL in a web browser:

```
http://[host name]:[port]/FormServerAdmin/settings.html
```

LiveCycle Assembler post-deployment tasks

This section describes the post-deployment tasks specific to LiveCycle Assembler.

Verifying the LiveCycle Assembler deployment

You can verify that LiveCycle Assembler is deployed and running properly by deploying the prebuilt sample. (See the *Samples Guide for LiveCycle Assembler*, located in the `[LiveCycle root]/Assembler/samples/docs` directory.)

► To deploy the LiveCycle Assembler sample:

1. Navigate to the `[LiveCycle root]/Assembler/samples/build/prebuilt` directory.
2. Deploy the `jboss.sample-assembler7.ear` file by copying it to the `[appserver root]/server/all/deploy` directory.
3. Type the following URL in a web browser:
`http://localhost:8080/adobe/livecycle/samples/assembler7/`
4. Click **About Assembler** and confirm that the LiveCycle Assembler information is returned to the Result window.

You can now create applications for LiveCycle Assembler. (See the *Developing Applications for LiveCycle Assembler* guide.)

LiveCycle Forms post-deployment tasks

This section describes the post-deployment tasks specific to LiveCycle Forms.

You can now create custom applications for LiveCycle Forms. (See the *Developing Custom Applications* guide.) For details on the Form Server Module, see the *Form Server Module API Reference*. For details on the XML Form Module API, see the *XML Form Module API Reference*.

You can optionally install LiveCycle Print to enable LiveCycle Forms to output forms as Adobe PostScript, Printer Control Language (PCL), and Zebra Programming Language (ZPL) print streams, in addition to PDF and HTML forms. Without LiveCycle Print, LiveCycle Forms cannot output forms as print streams. Instead, LiveCycle Forms sends forms across a network and renders them to client devices, such as web browsers, as PDF or HTML forms.

Verifying the LiveCycle Forms deployment

You can verify that LiveCycle Forms is deployed and running properly by accessing the Installation Verification Sample application that is installed with the product.

► **To access the Installation Verification Sample:**

1. Copy the FormsIVS.ear file from the `[LiveCycle root]/components/forms/samples/common/` directory to the `[appserver root]/server/all/deploy` directory.
2. Restart the application server.
3. Type the following URL in a web browser:

```
http:// [host name] : [port] /FormsIVS
```

where `[host name]` is the name of the server on which Forms has been deployed. If you are invoking FormsIVS on the server itself, provide the server name instead of "localhost".

Note: For information about using the sample application, click the Help link in the Samples page in the web browser.

LiveCycle Print post-deployment tasks

This section describes the post-deployment tasks specific to LiveCycle Print.

Verifying deployment

You can verify that LiveCycle Print is deployed and running properly using any of the following methods:

- Deploying and running the Print Installation and Verification Sample (PrintIVS) web application.
- Running the Print Submitter console application (you do not have to deploy this application).
Both of these applications invoke LiveCycle Print by using the LiveCycle Print API (this API consumes the LiveCycle Print web service).
- Running the Forms Installation and Verification Sample (FormsIVS) application. If LiveCycle Print is deployed, the Output format list contains three additional options: PostScript, ZPL, and PCL. For information about the Installation Verification Sample application, see ["Verifying the LiveCycle Forms deployment" on page 27](#).

In addition to running the two sample print applications, you can also invoke LiveCycle Print by using the Form Server Module API Print application. This sample Print application is a console application that invokes LiveCycle Print by using the Form Server Module API. (See [“Running the Form Server Module API Print application” on page 29.](#))

Deploying and running the PrintIVS web application

The PrintIVS web application is packaged in the `adobe-printSubmitter.ear` file, which is located in the `[LiveCycle root]/components/print/common/ear` directory. After you run Configuration Manager, the PrintIVS is deployed by default and is located in the `[LiveCycle root]/configurationManager/export` directory.

To run this application, you must understand the print specification because this application requires the same values as the print specification. (See “About the print specification” in the LiveCycle Print *Getting Started* guide.)

► To deploy and run the PrintIVS web application:

1. Copy the `adobe-printSubmitter.ear` file from the `[LiveCycle root]/components/print/common/ear` directory to the `[appserver root]/server/all/deploy` directory.
2. To run the sample application, type `http://[host name]:[port]/PrintIVS` in the URL line of a web browser.
3. Enter the required values and click **Print**.

Running the Print Submitter application

The Print Submitter application consists of the `lcprint.bat` (Windows) or `lcprint.sh` (Linux) file, which is located in the `[LiveCycle root]/components/print/samples/common/lcprint` directory.

You can run the Print Submitter application from a command prompt to invoke LiveCycle Print.

► To run the Print Submitter application:

1. From a command prompt, navigate to the `[LiveCycle root]/components/print/samples/common/lcprint` directory.

If you have not installed to the default directory, create a new environment variable named `LiveCycleRoot` and set the value to the installation directory used on your system.

2. Enter the following command:

```
lcprint <-d,--dataFile filename> <-f,--specFile filename> <-s,--server  
server:port>
```

The first argument, which is optional, is a fully qualified path to the data file. A series of data files accompanies LiveCycle Print and are located in the `[LiveCycle root]/components/print/samples/common/lcprint /data` directory.

Most of these files contain multiple records. For example, `PurchaseOrder_batch_25.xml` contains 25 records. You can create multiple print streams by using a data file that contains multiple records. (See “Working with multiple data records” in the LiveCycle Print *Getting Started* guide.)

The second argument is a fully qualified path to the print specification that is sent to LiveCycle Print. You can amend the `PrintSpecTemplate.xml` file and use this file. The `PrintSpecTemplate.xml` file is

located in the same directory as the `lcprint.bat` file. (See “About the print specification” in the LiveCycle Print *Getting Started* guide.)

One value that you must specify within the print specification is the location of the form design on which a print stream is based. Sample form designs are located in the `[LiveCycle root]/components/print/samples/common/lcprint/forms` directory.

The third argument, which is required, specifies the server name and port.

You can also use the Print Submitter console application to retrieve information about a previous request. To perform this task, use the following command:

```
lcprint <-r,--result requestId>
```

where `requestId` is a string value, such as `59572cd82`, that specifies a request identifier. This command returns result data that specifies the status of a request. (See “About result data” in the LiveCycle Print *Getting Started* guide.)

Running the Form Server Module API Print application

The Form Server Module API Print application uses the Form Server Module API to invoke LiveCycle Print from a command prompt. (See “Working with the Form Server Module” in the LiveCycle Print *Getting Started* guide.)

The Form Server Module API Print application consists of a Java file named `FsPrint.java`, which is located in the `[LiveCycle root]/components/print/samples/common/fsprint` directory. If you are using Windows, you must run the `fsprint.bat` file located in this directory. If you are using Linux, you must run the `fsprint.sh` file.

Note: The Form Server Module API Print application creates a `SOAPClient` object that specifies a valid SOAP endpoint. The default SOAP endpoint in the `FsPrint.java` file is configured for the default SOAP endpoint that JBoss uses. If LiveCycle Forms is deployed on a different J2EE application server, you must edit the `FsPrint.java` file by commenting and uncommenting the appropriate SOAP endpoint. For more information about using a `SOAPClient` object, see the LiveCycle Forms *Developing Custom Applications* guide.

► To run the Form Server Module API Print application:

1. From a command prompt, navigate to the application `[LiveCycle root]/components/print/samples/common/fsprint` directory.

If you have not installed to the default directory, create a new environment variable named `LiveCycleRoot` and set the value to the installation directory used on your system.

2. Enter the following command:

```
fsprint <-argument 1> <-argument2>
```

The first argument is a fully qualified path to the form design (an XDP file). You can use a sample form design that is located in the `[LiveCycle root]/components/print/samples/common/lcprint/forms` directory.

The second argument, which is optional, defines the print stream destination. For example, you can define a file URI to a PostScript file. If this value is not specified, output is written to the standard output. This value is similar to the `PrintURI` element that is located in the print specification. (See “About the print specification” in the LiveCycle Print *Getting Started* guide.)

Tip: The source code for the Form Server Module API Print application is located in the `FsPrint.java` file. You can open this file to see the application logic that invokes LiveCycle Print by using the Form Server Module API.

LiveCycle Form Manager post-deployment tasks

This section describes the post-deployment tasks specific to LiveCycle Form Manager.

Accessing the LiveCycle Form Manager end-user pages

After you configure User Management to connect to your organization's LDAP server, users can access the LiveCycle Form Manager end-user pages by going to the following URL:

```
http://[host name]:[port]/fm
```

The default port number for JBoss is 8080.

You can also access the end-user pages using the default administration user name (Administrator) and password (password).

For information about using the LiveCycle Form Manager end-user pages, see *LiveCycle Form Manager Help* (available from the Help menu of the LiveCycle Form Manager window).

LiveCycle PDF Generator post-deployment tasks

This section describes the post-deployment tasks specific to LiveCycle PDF Generator.

Setting up job sources

You can configure LiveCycle PDF Generator to accept jobs from watched folders, email, and the web UI (sample code). (See *LiveCycle PDF Generator Administration Help*.)

Additionally, LiveCycle PDF Generator can accept jobs submitted with the LiveCycle PDF Generator web services interface. To enable this, the sample web services client application installed with LiveCycle PDF Generator requires developer modifications before it can be used for job submission. For more information on configuring these job sources, see the LiveCycle PDF Generator *Developing Custom Applications* guide.

Setting Adobe PDF Printer as the default printer

This section applies to LiveCycle PDF Generator Professional and Elements only.

You must set up the Adobe PDF Printer as the default printer on the server.

► **To set up Adobe PDF Printer as the default printer:**

1. Navigate to the printer panel in your operating system.
2. Set **Adobe PDF** as the default printer.

Installing fonts

You can install the fonts that are used in documents converted by LiveCycle PDF Generator. Fonts may be referenced by PostScript or EPS documents, or they may be referenced by native file formats, such as those used by Microsoft Word or Microsoft Excel.

To make additional fonts available to any application, install the fonts using one of these methods:

- Install the fonts in the appropriate folder:
 - (Windows) C:\Windows\Fonts
 - (Linux) /usr/X11R6/lib/X11/fonts
- Using Configuration Manager, on the Font Manager Module Configuration screen, browse to the directory that contains the fonts and then redeploy the Font Manager Module. For information about deploying the Font Manager Module, see [“Configuring LiveCycle Products for Deployment” on page 48](#).

For a list of fonts installed with the Font Manager Module, see [“Fonts Installed with the Font Manager Module” on page 104](#).

Note: Fonts in the Font Manager Module are not available to the native applications that LiveCycle PDF Generator uses to convert native file formats. Such native applications include Microsoft Word and Acrobat. To make a font available to native applications, you must install it in the system font folder. If your LiveCycle PDF Generator server is clustered, the font must be installed on each computer used to host native applications.

Satisfying font references

This section applies to LiveCycle PDF Generator Elements and LiveCycle PDF Generator Professional only.

For font references in a native file, LiveCycle PDF Generator Elements and LiveCycle PDF Generator Professional use the native application with the Adobe PDF printer to convert a native file format to PDF. The native application satisfies font references from the C:\Windows\Fonts directory.

For font references in a PDF document, these products attempt to satisfy the font references from the following sources in the following order:

- Fonts embedded in the PDF document
- Font Manager Module
- C:\WINDOWS\Fonts

Setting the LiveCycle PDF Generator conversion time-out

You can configure LiveCycle PDF Generator conversion time-out settings at the application server level or job source level. If the time-out setting is configured at more than one level, the lowest level is applicable.

For details about configuring the setting at the application server level, see [“Manually Configuring JBoss” on page 52](#).

► To set the conversion time-out settings at the job source level:

- In Administrator, select **Create PDF** and, in the **Specify a timeout** box, type a value in seconds. By default, this value is read from the server.aes.properties file.

► **To set the conversion time-out settings for watched folders and email sources:**

1. In Administrator, select **Services > Adobe LiveCycle PDF Generator > Configuration Files > Export Configuration**.
2. Select **Download entire configuration** and click **Download**. By default, this value is read from the server.aes.properties file.
3. Edit the native2pdfconfig.xml file in a text editor and set the <timeout> level.
4. Upload your changes by selecting **Import Configuration**, browsing to the location of the native2pdfconfig.xml file, and then clicking **Import**.

Note: The time-out level for the web services APIs is read from the server.aes.properties file.

Next Step

If you have installed and configured LiveCycle Workflow manually, you must now configure the JBoss application server for BAM Server. (See [“Manually Configuring JBoss for BAM Server” on page 75.](#))

If you have installed and configured LiveCycle Workflow using the turnkey method, you can proceed to the instructions for configuring BAM Server. (See [“Getting Started with BAM Server” on page 90.](#))

Part II: Manual Installation, Configuration, and Deployment

This section of the guide describes how to manually install, configure and deploy your LiveCycle products.

For information on the automatic installation, configuration and deployment of the products, see [“Installing LiveCycle Products Using the Turnkey Installation” on page 19](#).

4

Installing LiveCycle Products

This chapter describes how to use the installation program to install LiveCycle products on a Windows or Linux operating system.

If you are installing LiveCycle PDF Generator (all versions), see [Installing LiveCycle PDF Generator](#).

Note: If you are installing multiple LiveCycle products we recommend you install LiveCycle PDF Generator first. If you install LiveCycle PDF Generator after installing other LiveCycle products you must complete the steps in [“To install LiveCycle PDF Generator with previously installed LiveCycle products:” on page 35](#).

If you are installing any of the following LiveCycle products, see [“Installing LiveCycle products” on page 36](#):

- LiveCycle Assembler
- LiveCycle Forms
- LiveCycle Form Manager
- LiveCycle Workflow

If you are installing LiveCycle Print (for use with LiveCycle Forms), see [“Installing LiveCycle Print” on page 38](#).

Note: You must install LiveCycle Forms prior to installing LiveCycle Print.

If you are installing Watched Folder (for use with LiveCycle Workflow and LiveCycle Assembler), see [“Installing Watched Folder” on page 39](#).

Before you install the products, you must ensure that your environment includes the software and hardware required to run LiveCycle products. You should also understand the installation options and have prepared the environment as required. (See [“Before You Install” on page 11](#).)

If you are installing to a location where a LiveCycle product is already installed, install the new product, run Configuration Manager to reassemble the products, undeploy the deployed product from the application server, and then redeploy the files to the application server.

Installing LiveCycle PDF Generator

The procedure in this section applies to LiveCycle PDF Generator Professional, LiveCycle PDF Generator Elements, and LiveCycle PDF Generator for PostScript.

When you run the installation program, you need the following information:

- The serial number for the product you are installing.
- The type of installation and configuration you are performing. (See [“Methods for installing, configuring, and deploying LiveCycle products” on page 12](#).)

Note: To successfully install, you need read and write permissions on the installation directory. The following are the default installation directories, although you can specify a different directory as required:

- Windows: C:\Adobe\LiveCycle\
- Linux: /opt/adobe/lifecycle/

You must configure the JBoss service to run with the same login properties as the Microsoft Office administrator. Configure the JBoss service login properties with the Services window in the Administrative Tools area of the Windows Control Panel.

If you are installing LiveCycle PDF Generator Professional or LiveCycle PDF Generator Elements, ensure that Acrobat is not installed on the computer.

When installing on Linux, the installation program uses the logged-in user's home directory as a temporary directory for storing files. As a result, messages such as the following may appear in the console:

```
WARNING: could not delete temporary file /home/<username>/ismmp001/1556006
```

When you complete the installation, you must manually delete the temporary files.

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

► **To install LiveCycle PDF Generator with previously installed LiveCycle products:**

If you installed other LiveCycle products previously you must complete these steps before installing LiveCycle PDF Generator.

1. From the Windows services panel, select **JBoss for Adobe LiveCycle**.
2. Stop the service if it is running.
3. Right-click the service and open the properties.
4. Select the **Log On** tab.
5. Select **This account**.
6. Enter the user name and password used when installing the Microsoft Office and Adobe Acrobat applications
7. Start the **JBoss for Adobe LiveCycle** service.

► **To install LiveCycle PDF Generator for manual deployment:**

1. At the root level of the installation media, start the installation program:
 - (Windows) Double-click the .exe file.
 - (Linux) From a command prompt, type: `file_name.bin`

Note: You may have to change the permissions on the install program under Linux. To do so, type:
`chmod +x filename.bin`

2. If prompted, select a language for the installation program and click **OK**.
3. On the Welcome screen, click **Next**.
4. Type the serial number in the text box and click **Next**.
5. (LiveCycle PDF Generator Professional, LiveCycle PDF Generator Elements) On the Adobe LiveCycle PDF Generator 7.2 preinstallation requirements screen, review the preinstallation requirements and click **Next**.

6. If you are prompted to disable on-access virus scanning software, click **Next**. It is recommended that you disable any virus scanning software during the installation process to improve the speed of the installation.
7. Read the Product License Agreement, select **I accept the terms of the license agreement**, and then click **Next**.
8. Accept the default directory as listed or click **Browse** and navigate to the directory where you want to install the product, and then click **Next**.
Note: If you type in the name of a directory that does not exist, it will be created for you.
9. If prompted, select the application server you are deploying the product to and click **Next**.
Note: Do not select the option to automatically configure and deploy.
10. (LiveCycle PDF Generator Professional, LiveCycle PDF Generator Elements) On the Corel WordPerfect screen, browse to the Corel WordPerfect executable file (if installed) and click **Next** or, if Corel WordPerfect is not installed, click **Next**.
11. (LiveCycle PDF Generator Professional, LiveCycle PDF Generator Elements) On the Adobe Photoshop screen, browse to the Adobe Photoshop executable file (if installed) and click **Next** or, if Photoshop is not installed, click **Next**.
12. Review the installation details, and then click **Install**. The installation program displays the progress of the installation. A summary screen appears when the product installation is completed.
13. Ensure that the **Start Configuration Manager** option is not selected, and then click **Finish**.
Caution: Do not run Configuration Manager until you have installed all of the products you require and have prepared your environment. (See [“Preparing Your Environment” on page 41.](#)) After you have completed installing all of the products and preparing your environment, you only need to run Configuration Manager once to configure and assemble the deployable components. If you add another LiveCycle product later, you can run Configuration Manager again to configure, assemble and deploy the product modules. (See [“Configuring LiveCycle Products for Deployment” on page 48.](#))

Installing LiveCycle products

This section applies only to LiveCycle Assembler, LiveCycle Forms, LiveCycle Form Manager, and LiveCycle WorkFlow.

When you run an installation program, you need the following information:

- The serial number for the product you are installing.
- The type of installation and configuration you are performing. (See [“Methods for installing, configuring, and deploying LiveCycle products” on page 12.](#))

Note: To successfully install, you need read and write permissions on the installation directory. The following are the default installation directories, although you can specify a different directory as required:

- Windows: C:\Adobe\LiveCycle\
- Linux: /opt/adobe/lifecycle/

Note: If you select Japanese when installing LiveCycle Forms on a Japanese localized computer, the character set is not updated. You can change this setting post-deployment by using Administrator to access the Internationalization settings. This ensures that Japanese characters render properly. (See [“Accessing Administrator” on page 25.](#))

When installing on Linux, the installation program uses the logged-in user's home directory as a temporary directory for storing files. As a result, messages such as the following may appear in the console:

```
WARNING: could not delete temporary file /home/<username>/ismp001/1556006
```

When you complete the installation, you must manually delete the temporary files.

Caution: Ensure that the temporary directory for your operating system that you are using has a minimum of 3 GB of free space available for the installation program:

- (Windows) TMP or TEMP path as set in the Environment Variables
- (Linux) Logged-in user's home directory

► **To install LiveCycle products for manual deployment:**

1. Navigate to the root directory of the installation media. For Workflow navigate to the `AdobeLiveCycleWorkflow7.2` directory.
2. Start the installation program:
 - (Windows) Double-click the `.exe` file.
 - (Linux) From a command prompt, type: `file_name.bin`

Caution: When you are installing the product on a Linux operating system, you must be logged in as the root user to successfully install the product to the default location, `/opt/adobe/livecycle/`. If you are logged in as a non-root user, you must change the installation directory to one for which you have permissions (for example, `$HOME/adobe/livecycle`).

3. If prompted, select a language for the installation program and click **OK**.
4. On the Welcome screen, click **Next**.
5. On the Read Before Install screen, if you are upgrading from a previous version, ensure that you have complied with the upgrade instructions, and then click **Next**. If you are not upgrading, click **Next**.
6. Type the serial number in the text box and click **Next**.
7. If you are prompted to disable on-access virus scanning, click **Next**. It is recommended that you disable any virus scanning software during the installation process to improve the speed of the installation.
8. Read the Product License Agreement, select **I accept the terms of the license agreement**, and then click **Next**.

Note: Depending on the product you are installing, the next two steps may appear in reverse order.

9. Select the application server you are deploying the product to and click **Next**.

Note: Do not select the option to automatically configure and deploy.

10. Accept the default directory as listed or click **Browse** and navigate to the directory where you want to install the product, and then click **Next**.

Note: If you type in the name of a directory that does not exist, it will be created for you.

Caution: When you install the product, you can specify a different installation location. If you are installing on Linux, the directory you specify should not contain any spaces; otherwise, the installation program does not install the product.

11. Review the installation details, and then click **Install**. The installation program displays the progress of the installation. A summary screen appears when the product installation is completed.
12. Ensure that the **Start Configuration Manager** option is not selected, and then click **Finish**.

Caution: Do not run Configuration Manager until you have installed all of the products you require and have prepared your environment. (See [“Preparing Your Environment” on page 41.](#)) After you have completed installing all of the products and preparing your environment, you only need to run Configuration Manager once to configure and assemble the deployable components. If you add another LiveCycle product later, you can run Configuration Manager again to configure, assemble and deploy the product modules. (See [“Configuring LiveCycle Products for Deployment” on page 48.](#))

Note: If you installed LiveCycle Workflow, you can now install Watched Folder. (See [“Installing Watched Folder” on page 39.](#))

Note: If you installed LiveCycle Forms, you can now install LiveCycle Print. (See [“Installing LiveCycle Print” on page 38.](#))

Installing LiveCycle Print

Before installing LiveCycle Print you must first install LiveCycle Forms. (See [“Installing LiveCycle Products” on page 34.](#))

► To install LiveCycle Print:

1. At the root level of the installation media, start the installation program:
 - (Windows) Double-click the `adobe_livecycle_print_7_2.exe` file.
 - (Linux) From a command prompt, type: `adobe_livecycle_print_7_2.bin`

Note: You may have to change the permissions on the install program under Linux. To do so, type:
`chmod +x filename.bin`
2. If prompted, select a language for the install program and click **OK**.
3. On the Welcome screen, click **Next**.
4. On the Read Before Install screen, if you are upgrading from a previous version, ensure that you have complied with the upgrade instructions, and then click **Next**. If you are not upgrading, click **Next**.
5. Type the serial number in the text box and click **Next**.
6. Read the Product License Agreement, select **I accept the terms of the license agreement**, and then click **Next**.
7. Accept the default directory as listed or click **Browse** and navigate to the `[LiveCycle root]` directory where you installed LiveCycle Forms.

Note: You must install LiveCycle Print in the same `[LiveCycle root]` directory as LiveCycle Forms.
8. Review the installation details, and then click **Install**. The installation program displays the progress of the installation. A summary screen appears when the product installation is completed.

9. Ensure that the **Start Configuration Manager** option is not selected, and then click **Finish**.

Caution: Do not run Configuration Manager until you have installed all of the products you require and have prepared your environment. (See [“Preparing Your Environment” on page 41.](#)) After you have completed installing all of the products and preparing your environment, you only need to run Configuration Manager once to configure and assemble the deployable components. If you add another LiveCycle product later, you can run Configuration Manager again to configure, assemble and deploy the product modules. (See [“Configuring LiveCycle Products for Deployment” on page 48.](#))

Note: In order to test your installation with the LiveCycle Print sample applications, you need to deploy the adobe-printSubmitter.ear file. (See [“Deploying and running the PrintIVS web application” on page 28.](#))

Installing Watched Folder

Installing Watched Folder places the files required for configuration and deployment onto your computer. Ensure that you have already installed LiveCycle Workflow and LiveCycle Assembler before performing this procedure.

► To install Watched Folder:

1. From the Watched Folder directory on the Adobe LiveCycle Assembler installation DVD, start the installation program:
 - (Windows) Double-click the setupwin32.exe file.
 - (Linux) From a command prompt, type: `setupLinux.bin`
2. If prompted, select a language for the installation and click **OK**.
3. Click **Next** on the Welcome screen.
4. If you want to increase the speed of the installation, disable any virus scanning software, and then click **Next**.
5. Read the Product License Agreement, select **I accept the terms of the license agreement**, and then click **Next**.
6. Accept the default location, or click **Browse** and navigate to the location where you want to install Watched Folder, and then click **Next**.
7. Review the installation details, and then click **Install**. A summary screen appears when the product installation is completed.
8. Ensure that the **Start Configuration Manager** option is not selected, and then click **Finish**.

Caution: Do not run Configuration Manager until you have installed all of the products you require and have prepared your environment. (See [“Preparing Your Environment” on page 41.](#)) After you have completed installing all of the products and preparing your environment, you only need to run Configuration Manager once to configure and assemble the deployable components. If you add another LiveCycle product later, you can run Configuration Manager again to configure, assemble and deploy the product modules. (See [“Configuring LiveCycle Products for Deployment” on page 48.](#))

Viewing the error log

If any errors occur during the installation, the installation program creates a log file called log.txt, which contains the error messages. The log file is located in the *[LiveCycle root]* directory.

Next steps

You must now prepare your database and application server for hosting LiveCycle products. (See [“Preparing Your Environment” on page 41.](#))

5

Preparing Your Environment

This chapter describes how to prepare your environment for hosting LiveCycle products. You must perform the tasks provided in this chapter before you configure LiveCycle products:

- [“Creating the database” on page 41](#)
- [“Preparing the application server” on page 47.](#)

You do not need to perform these tasks if you are performing a turnkey install.

Creating the database

This section describes how to set up the database that stores LiveCycle configuration information and run-time data. The procedures described in this section apply to all LiveCycle products that use User Management. If you previously configured the database for deploying other LiveCycle products, you do not need to perform the tasks again.

Note: User Management is an optional feature for LiveCycle Forms.

If this is the first installation of a LiveCycle product, you must create an empty database. All of the tables required to support LiveCycle products will be created by Configuration Manager when you initialize the database. (See [“Initializing the Database” on page 65.](#))

Before creating the database, you must ensure that you have read the preinstallation requirements and have the required software installed. (See [“Before You Install” on page 11.](#))

For information on creating the BAM metadata database for LiveCycle Workflow Server, see [“Creating the BAM metadata database” on page 75.](#)

Creating a MySQL database

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

Tip: If you want to set up a database schema and new users on MySQL using a graphical user interface (GUI), you must install the MySQL Administrator tool. For more information, see the MySQL user documentation

You need the database name and the user name and password of the MySQL user account when you configure the database connection on JBoss.

MySQL user account

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

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

MySQL database server configuration

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

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

► To customize the MySQL configuration:

1. Open the my.ini (Windows) or my.cnf (Linux) file in a text editor.
2. Add the following line to the [mysqld] section:

```
max_allowed_packet=25M
```
3. Save and close the file.

Creating an Oracle database

If you prefer not to use the default database that was created when you installed Oracle 9i or 10g, create a new database using the Database Configuration Assistant tool.

If any of the LDAP directories that the new database will synchronize with for authenticating LiveCycle users includes records with UTF-8 characters, you must create a database that uses the UTF-8 character set.

You must also create a new user on the database and assign it the DBA, CONNECT and RESOURCE roles, as well as the ACCESS_ANY_WORKSPACE, UNLIMITED TABLESPACE, and CREATE VIEW system privileges. For deployments on Linux, the user name must not exceed 8 characters and, on Windows, it must not exceed 12 characters.

The user name and password of the new user you create on the database is used again when you create the data source.

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

Creating a DB2 database

Create a DB2 database by running the script provided in this section. The script is tuned for a system that will use 1 GB of memory for the database. If your system has a different amount of memory dedicated for the database, see the DB2 documentation for details on configuring your system settings.

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

The user name and password of the new user you create on the database is used again when you create the data source.

After you create the database, you must configure it to enable concurrent usage. (See [“Configuring a DB2 database for concurrent usage” on page 44.](#))

► To create a DB2 database:

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

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

2. Make the following changes to the script:
 - Replace the instances of *dbname* and *DBNAME* with the name that you want for the LiveCycle 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 using the DB2 administration tools after initializing the database.

3. Save the text file in a location that DB2 Command Line Processor can access.
4. Open a command prompt.
5. (Windows) Enter the following command to open DB2 Command Line Processor:

```
db2cmd
```

6. Enter the following command to run the script:

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

Configuring a DB2 database for concurrent usage

If you are using a DB2 database, you must configure it for multiple-user scenarios.

► To configure the DB2 database for concurrent usage:

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

DB2 property	Required value
Workload type	Mixed
Average number of SQL transactions per unit of work	Fewer than 10
Database Administration Priority	Faster transaction performance
Populated Database	Set according to the current state of the database. The database is not populated if the database has not yet been initialized using Configuration Manager.
Isolation Level	Cursor Stability

Creating a SQL Server database

You can create a SQL Server database that LiveCycle products will use to store run-time and configuration data. For information on creating a SQL Server database, refer to the SQL Server documentation. LiveCycle products support SQL Server 2000 SP3a.

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

The SQL Server database can be configured with the Windows or SQL Server authentication types. To run the database with JBoss, the authentication type must be set to SQL Server.

Note: LiveCycle PDF Generator also requires a JMS database on SQL Server. (See [“Creating a JMS database on SQL Server” on page 45.](#))

Creating a JMS database on SQL Server

This section applies to LiveCycle PDF Generator only.

For LiveCycle PDF Generator deployed on JBoss, in addition to the LiveCycle database, you also need to create a Java Message Service (JMS) database on the SQL Server.

► To create a JMS database:

1. Select **Start > Programs > Microsoft SQL Server > Enterprise Manager**.
2. In the navigation tree, select **Console Root > Microsoft SQL Servers > SQL Server Group > [servername] > Databases**.
3. Right-click **Databases** and click **New Database**.
4. In the **Name** box, type a name for the database and click **OK**.
5. In the navigation tree, select **Console Root > Microsoft SQL Servers > SQL Server Group > [servername] > Security > Logins**.
6. Right-click **Logins** and select **New Login**.
7. On the SQL Server Properties screen, type a name for this login in the **Name** box, select **SQL Server Authentication**, and type a password in the **Password** box.
8. Click the **Database Access** tab, select the database that you created in step 4 and, in the **Permit in Database Role** list, select **public** and **db_owner**.
9. Click the **Server Roles** tab, select **Database Creators**, and then click **OK**.
10. On the Confirm Password screen, type the password you created in step 7 and click **OK**.
11. In the navigation tree, select and right-click **Logins**, select **New Login**, type `guest` in the **Name** box, and select **SQL Server Authentication**.

12. Ensure that the default database is set to the database that you created in step 4 and that the default language is English.
13. Click the **Database Access** tab, select the database that you created in step 4 and, in the **Permit in Database Role** list, select **public**, and then click **OK**.
14. In the Confirm Password dialog box, select **Blank Password** and click **OK**.

Note: You may use any combination of user name and password to create a JMS database. If you choose to create a password, enter it in the required field instead of selecting Blank Password.

15. Select **Start > Programs > Microsoft SQL Server > Query Analyzer**, and then log in as the user you created in step 7.
16. Copy the following SQL script into the Query window:

```
USE YourDatabase;

CREATE TABLE JMS_MESSAGES (MESSAGEID INTEGER NOT NULL,
    DESTINATION VARCHAR(150) NOT NULL, TXID INTEGER, TXOP CHAR(1),
    MESSAGEBLOB IMAGE, PRIMARY KEY (MESSAGEID, DESTINATION));
CREATE TABLE JMS_TRANSACTIONS (TXID INTEGER);

CREATE TABLE JMS_USERS (USERID VARCHAR(32) NOT NULL,
    PASSWD VARCHAR(32) NOT NULL, CLIENTID VARCHAR(128), PRIMARY KEY(USERID));
CREATE TABLE JMS_ROLES (ROLEID VARCHAR(32) NOT NULL,
    USERID VARCHAR(32) NOT NULL, PRIMARY KEY(USERID, ROLEID));
CREATE TABLE JMS_SUBSCRIPTIONS (CLIENTID VARCHAR(128) NOT NULL,
    SUBNAME VARCHAR(128) NOT NULL, TOPIC VARCHAR(255) NOT NULL,
    SELECTOR VARCHAR(255), PRIMARY KEY(CLIENTID, SUBNAME));

INSERT INTO JMS_USERS (USERID, PASSWD) VALUES ('guest', 'guest');
INSERT INTO JMS_USERS (USERID, PASSWD) VALUES ('j2ee', 'j2ee');
INSERT INTO JMS_USERS (USERID, PASSWD, CLIENTID) VALUES ('john', 'needle',
    'DurableSubscriberExample');
INSERT INTO JMS_USERS (USERID, PASSWD) VALUES ('nobody', 'nobody');
INSERT INTO JMS_USERS (USERID, PASSWD) VALUES ('dynsub', 'dynsub');
INSERT INTO JMS_ROLES (ROLEID, USERID) VALUES ('guest', 'guest');
INSERT INTO JMS_ROLES (ROLEID, USERID) VALUES ('j2ee', 'guest');
INSERT INTO JMS_ROLES (ROLEID, USERID) VALUES ('john', 'guest');
INSERT INTO JMS_ROLES (ROLEID, USERID) VALUES ('subscriber', 'john');
INSERT INTO JMS_ROLES (ROLEID, USERID) VALUES ('publisher', 'john');
INSERT INTO JMS_ROLES (ROLEID, USERID) VALUES ('publisher', 'dynsub');
INSERT INTO JMS_ROLES (ROLEID, USERID) VALUES ('durpublisher', 'john');
INSERT INTO JMS_ROLES (ROLEID, USERID) VALUES ('durpublisher', 'dynsub');
INSERT INTO JMS_ROLES (ROLEID, USERID) VALUES ('noacc', 'nobody');
```

17. In the first line of the script, replace *YourDatabase* with the name of the database that you created in step 4.
18. Run the script by clicking the green arrow on the taskbar. The SQL script should run without error.

Preparing the application server

You must install the application server that you will use to run LiveCycle products. You also need to perform some preliminary configuration on the application server.

Installing JBoss Application Server

If you are manually configuring JBoss for running LiveCycle products, you must download and install JBoss Application Server 3.2.5. You can obtain JBoss at this location:

<http://labs.jboss.com/portal/jbossas/download>

If you are performing a turnkey install, you do not need to install JBoss because Configuration Manager installs it automatically.

Next step

You must now configure the LiveCycle products for deployment. (See [“Configuring LiveCycle Products for Deployment” on page 48.](#))

6

Configuring LiveCycle Products for Deployment

This chapter describes how to configure LiveCycle products for deployment.

This chapter assumes that you have installed LiveCycle products. If you have not installed the products, see [“Installing LiveCycle Products” on page 34](#).

The instance of Configuration Manager that is installed with one LiveCycle product can be used with all LiveCycle products. When you run Configuration Manager, you can specify the LiveCycle products that you are configuring as well as the type of application server that you are leveraging in the solution. You can also set run-time properties for LiveCycle products and enable or disable security.

If you are already running one or more LiveCycle products, you must configure and assemble those existing products with the new one that you want to configure. To do this, you must undeploy the previously deployed products and select all of the products that you want to assemble during configuration. The previously deployed products are then included in the LiveCycle.ear file and deployed with the new product when you deploy the file to the application server.

Note: To assemble multiple LiveCycle products, each product must be installed in the same *[LiveCycle root]* directory.

When Configuration Manager completes the configuration of the products, it places the files to be deployed to the application server (LiveCycle.ear, adobe-FontManager.ear, and the product-related ear files) in the following directory:

- (Windows) *[LiveCycle root]*\configurationManager\export
- (Linux) *[LiveCycle root]*/configurationManager/export

► To configure the products for deployment:

1. Navigate to the *[LiveCycle root]*/configurationManager directory and start Configuration Manager:
 - (Windows) Double-click **ConfigurationManager.exe**.
 - (Linux) From a command prompt, type: `ConfigurationManager.bin`
2. If prompted, select a language for Configuration Manager and click **OK**.
3. On the Configuration Manager Welcome screen, click **Next**.
4. Select **Custom Configuration Wizard** and click **Next**.
5. If prompted, on the Configuration Preferences screen, select either **Use Previously Entered Values** or **Revert to Default Values**, and then click **Next**.
6. Select the application server you have installed, select **Foundation** and the products that you want to configure, and then click **Next**.
7. (LiveCycle Forms) On the Adobe User Management Selection screen, select either **LiveCycle Forms with User Management and Administration** or **LiveCycle Forms without User Management and Administration**, and then click **Next**.

8. Ensure that only **Configure and assemble LiveCycle products** is selected, and then click **Next**.
9. On the Configure and Assemble products screen, click **Next**.
10. (Optional) On the Data Manager Module Configuration screen, if you are using SSL security on your application server, select **Enable SSL** and type the SSL credential password.
If you have not yet set up your SSL credential, you can type a password here and use it when you create an SSL credential. (See [“Configuring SSL on JBoss” on page 72.](#))
11. Enter a directory to use for **Adobe LiveCycle products temp file**, and then click **Next**.

Note: (Linux) If you are logged in as a non-root user, specify a directory under your home directory.

For more information on the Adobe LiveCycle products temp file, see [“Optimizing inline documents and impact on JVM memory” on page 141.](#)

12. On the Data Manager Module Configuration continued screen, accept the default values for the following properties or specify new values, and then click **Next**:
 - **Local storage sweep interval (in seconds):** The amount of time between attempts to delete any files that are no longer needed and were used to pass the document data between LiveCycle services running on the same computer.
 - **Global storage sweep interval (in seconds):** The amount of time between attempts to delete any obsolete files that were used to pass the document data between LiveCycle services running on the different computers. Specify this property only when deploying LiveCycle products in a clustered environment.
 - **Default maximum inline size (in bytes):** The maximum number of bytes kept in memory when passing documents between different LiveCycle components. Documents that exceed this maximum are stored on the hard drive. Use this property for performance tuning. (See [“Optimizing inline documents and impact on JVM memory” on page 141.](#))
 - **Default disposal time-out (in seconds):** The maximum amount of time during which a document being passed between different LiveCycle components is considered active. After this time has passed, any files used to store this document are subject to removal. Use this property to control the usage of disk space.
 - **Use NFS protocol (Windows only):** Select this option when deploying LiveCycle products in a clustered environment. Additional NFS software should be installed on your computer running Windows before enabling this option. This option does not affect deployments on Linux.
 - **Global storage directory:** A path to a shared directory used to store long-lived documents that are passed between LiveCycle products. LiveCycle Workflow uses this directory to share process-related files among cluster nodes. LiveCycle Form Manager uses this directory to store index files used for full-text searches. Using an NFS shared directory can help to improve performance.

(LiveCycle PDF Generator) When LiveCycle PDF Generator is running as a service, to specify a network folder for the Global storage directory property, you must use the network address of the folder, and not the path of a mapped drive. For example, \\computer_name\temp is the network address of the temp folder on the computer named computer_name.

For more information on the Global storage directory property, see [“Optimizing inline documents and impact on JVM memory” on page 141.](#)

13. (LiveCycle Forms, LiveCycle Form Manager, LiveCycle Workflow, LiveCycle Assembler) (Optional) On the Font Manager Module Configuration screen, select fonts for LiveCycle Forms to use in addition to the fonts that are included with the product. In the **Fonts directory** box, type the path or browse to the directory that contains the fonts to add, and then click **Next**.

Note: Your right to use fonts provided by parties other than Adobe is governed by the license agreements provided to you by such parties in connection with those fonts, and is not covered under your license to use Adobe software. Adobe recommends that you review and ensure you are in compliance with all applicable non-Adobe license agreements before using non-Adobe fonts with Adobe software, particularly with respect to use of fonts in a server environment.

14. (LiveCycle Assembler) (Optional) On the Assembler Security Group configuration screen, specify the type of security you want to implement controlling permission to execute Assembler DDX jobs, and then click **Next**. Two types of user authentication are available:
 - Type an asterisk (*) to allow any user with a valid entry in the LDAP system to log in.
 - Leave the field blank to allow any user to log in.
15. (LiveCycle Forms, LiveCycle Form Manager, LiveCycle Workflow) (Optional) On the Form Server Module Configuration screen, accept the default values for the following properties or specify new values, and then click **Next**:
 - **Validation UI:** The type of UI to use on HTML forms if a validation error occurs. If you select **List**, all validation errors on the form are displayed as links. If you select **MessageBox**, each error message is displayed in a dialog box controlled by clicking next and previous buttons. The location of the list or buttons depends on the option selected for Validation Reporting.
 - **Validation Reporting:** The location in which the validation messages are displayed. Select one of the following options:
 - **Frame Left** - Displays validation messages within a frame on the left side of the web browser.
 - **Frame Right** - Displays validation messages within a frame on the right side of the web browser.
 - **Frame Top** - Displays validation messages within a frame at the top of the web browser.
 - **Frame Bottom** - Displays validation messages within a frame at the bottom of the web browser.
 - **No Frame Left** - Displays validation messages in the same window on the left side of the web browser.
 - **No Frame Right** - Displays validation messages in the same window on the right side of the web browser.
 - **No Frame Top** - Displays validation messages in the same window at the top of the web browser.
 - **No Frame Bottom** - Displays validation messages in the same window at the bottom of the web browser.
 - **None** - Does not display validation messages.
 - **No UI** - Returns the validation messages through the API (with data). The validation messages are not displayed on-screen.
 - **No UI With Form** - Returns the validation messages through the API (with the form). The validation messages are not displayed on-screen.
 - **Validation Border:** The frame border size (in pixels) when Validation Reporting is set to Frame Left, Frame Right, Frame Top, or Frame Bottom. The frame border size must be equal to or greater than 0.

- **Output Type:** The type of HTML output returned to the web browser. Select **Full HTML** to render the form within full HTML tags (a complete HTML page) or select **Form Body** to render the form within div tags (not a complete HTML page).
 - **Locale:** The language to be used for validation messages sent to client devices, such as web browsers, as part of HTML transformations. The default value is English (United States). For information on applicable values, see “Language and Locale Combinations” in the *Developing Custom Applications* guide that is installed with LiveCycle Forms.
 - **Charset:** The character set used to encode the output byte stream. This property is dependent on the `sFormPreference` parameter specified through the API. For HTML transformations, the Form Server Module supports character encoding values defined by the `java.nio.charset` package. To view a list of supported values, go to <http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html>.
 - **Cache Enabled:** Select this option to optimize performance. This option works with the form design’s caching property. For information on configuring this value in the form design, see *Adobe LiveCycle Designer Help*.
16. (LiveCycle Forms, LiveCycle Form Manager, LiveCycle Workflow) (Optional) On the XMLForm Module Configuration screen, accept the default values for the following properties or specify new values, and then click **Next**:
- **Pool Max:** The maximum number of processes that can be running. The recommended value for PoolMax is the total number of CPUs * 2 + constant (1 or 2). When the PoolMax limit is reached, service instances are created from PoolSize.
 - **Trace Level:** The trace level. The following options are available:
 - **None** - Tracing is not enabled.
 - **Enter/Exit** - Messages are logged each time a method is entered and exited.
 - **Parameters** - Messages are logged each time a method is entered and exited, and the value of all parameters passed into each method is logged.
 - **Details** - Messages are logged each time a method is entered and exited, the value of all parameters passed into each method and any extra messages are logged.
 - **Report timing information:** Select this option to monitor the start and end time of calls. You can use this information for performance tuning.
17. On the Application Configuration and Assembly Summary screen, review the configuration details, and then click **Next**.
18. On the Configure and Assemble Products Summary screen, click **Next**.
19. On the Tasks Completed screen, click **Next** (if you selected another task for Configuration Manager to perform) or click **Finish** to close Configuration Manager.

Next step

You must now configure your application server. (See [“Manually Configuring JBoss” on page 52.](#))

This chapter describes how to manually configure the JBoss Application Server to prepare for the manual deployment of LiveCycle products. You can also automatically configure your application server, as part of the turnkey installation. (See [“Installing LiveCycle Products Using the Turnkey Installation” on page 19.](#))

You need to perform the following tasks:

- Modify various JBoss configuration files on the instance of JBoss that hosts LiveCycle products. (See [“Modifying the JBoss configuration files” on page 53.](#))
- Copy the appropriate Watched Folder file to JBoss. (See [“Configuring JBoss for Watched Folder” on page 57.](#))
- Create the data source so that LiveCycle products can connect to the database. (See [“Connecting JBoss for LiveCycle products to the LiveCycle database” on page 57.](#))

Note: If you have installed LiveCycle PDF Generator in addition to other previously installed LiveCycle products, you need to change the credentials of the login user account to those of the account used to install Microsoft Office on the machine where LiveCycle is installed.

The procedures in this chapter are for configuring the application server for all LiveCycle products. If a product requires a different process, the steps are highlighted. If you previously configured your application server for deploying other LiveCycle products, you do not need to perform the procedures in this chapter again.

This chapter assumes that you have configured your database and created a database user account. If you have not done this, see [“Preparing Your Environment” on page 41.](#)

Starting and stopping JBoss

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

► To start JBoss:

1. From a command prompt, navigate to `[appserver root]/bin`.
2. Start the server by entering 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 server by entering the following command:
 - (Windows) `shutdown.bat -S`
 - (Linux) `./shutdown.sh -S`

Modifying the JBoss configuration files

You need to modify the following areas on the instance of JBoss that hosts LiveCycle products:

- endorsed directory
- run.bat file (Windows) or run.sh file (Linux)
- JMS
- jacob.properties file
- EAR file class-loading isolation
- maximum threads settings
- dom4j.jar file
- cluster-service.xml

Note: If you are running JBoss on Red Hat Enterprise Linux Advanced Server 3.0, you must set the LD_ASSUME_KERNEL environment variable to 2.4.19 using the export LD_ASSUME_KERNEL=2.4.19 command. You must then run JBoss from the same shell in which you set the environment variable.

► To create an endorsed directory:

1. Navigate to the `[appserver root]/server/all/lib` directory and create a new directory called `endorsed`.
2. Copy the following files from the `[LiveCycle root]/components/um/endorsed` directory to the endorsed directory you just created:
 - dom3-xercesImpl-2.4.0.jar
 - dom3-xml-apis-2.4.0.jar
 - xalan-2.4.1.jar

► (LiveCycle Form Manager) To modify the run.bat or run.sh file:

1. Navigate to the `[appserver root]/bin` directory and open the following file in a text editor:
 - (Windows) run.bat
 - (Linux) run.sh

Tip: Turn off word wrapping while editing to ensure that there are no hard returns in these command lines.

2. (Windows) In the run.bat file, locate the following line:

```
"%JAVA%" %JAVA_OPTS% -classpath "%JBOSS_CLASSPATH%" org.jboss.Main %*
```

Replace the line with the following text:

```
"%JAVA%" %JAVA_OPTS% -classpath "%JBOSS_CLASSPATH%" -Xms128m -Xmx512m  
-Dorg.apache.lucene.writeLockTimeout=10000  
-Djava.endorsed.dirs=%JBOSS_HOME%/server/all/lib/endorsed org.  
jboss.Main -c all %*
```

3. (Linux) In the run.sh file, locate the following text:

```
# Execute the JVM
"$JAVA" $JAVA_OPTS \
  -classpath "$JBOSS_CLASSPATH" \
  org.jboss.Main "$@"
```

Replace the text with the following text:

```
# Execute the JVM
"$JAVA" $JAVA_OPTS -Xms128m -Xmx1024m \
  -classpath "$JBOSS_CLASSPATH" \
  -Dorg.apache.lucene.writeLockTimeout=10000\
  -Djava.endorsed.dirs=$JBOSS_HOME/server/all/lib/endorsed \
  org.jboss.Main -c all "$@"
```

4. (SQL Server) Add the following argument to the text you modified in step [2](#) or [3](#):

```
-Dadobe.workflow.scheduler.jdbc.delegate=org.quartz.impl.jdbcjobstore
.MSSQLDelegate \
```

Note: This command should immediately follow the appropriate -Djava command above. For Windows, it must precede "org.jboss.Main".

5. Save the file.

► **To modify JMS configuration:**

1. (LiveCycle Form Manager and LiveCycle Workflow) Navigate to the `[appserver root]/server/all/deploy/jms` directory and open the `hajndi-jms-ds.xml` file in a text editor.
2. (LiveCycle Form Manager and LiveCycle Workflow) Add the following text after the `Jms XA Resource adapter` section:

```
<tx-connection-factory>
<track-connection-by-tx>>true</track-connection-by-tx>
<jndi-name>adobe_JmsQueueXA</jndi-name>
<xa-transaction/>
<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:/DefaultJMSProvider
</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/>
<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:/DefaultJMSProvider
</config-property>
</tx-connection-factory>
```

3. (LiveCycle PDF Generator, LiveCycle Assembler) Navigate to the `[appserver root]/server/all/deploy/jms` directory and unpack the `jms-ra.rar` file.
4. (LiveCycle PDF Generator, LiveCycle Assembler) Navigate to the `[appserver root]/server/all/deploy/jms/META-INF` directory and open the `ra.xml` file in a text editor.
5. (LiveCycle PDF Generator, LiveCycle Assembler) In the `ra.xml` file, locate the following lines:

```
<config-property-name>Strict</config-property-name>
<config-property-type>java.lang.Boolean</config-property-type>
<config-property-value>true</config-property-value>
```

Change the value for `<config-property-value>` from `true` to `false` and save your changes to the file.

6. Repackage the `jms-ra.rar` file.

► **To modify the `jacorb.properties` file:**

1. Navigate to the `[appserver root]/server/all/conf` directory and open the `jacorb.properties` file in a text editor.
2. Locate the `jacorb.poa.thread_pool_max` property and change the property value to 16.

► **To modify the maximum threads settings:**

1. Navigate to the `[appserver root]/server/all/deploy/jbossweb-tomcat50.sar/` directory and open the `server.xml` file in a text editor.
2. In the `<connector>` element for port 8080, set `maxThreads` to 75, `minSpareThreads` to 25, and `maxSpareThreads` to 50, as shown in the following example:

```
<!-- A HTTP/1.1 Connector on port 8080 -->
<Connector port="8080" address="{jboss.bind.address}"
maxThreads="75" minSpareThreads="25" maxSpareThreads="50"
enableLookups="false" redirectPort="8443" acceptCount="250"
connectionTimeout="20000" disableUploadTimeout="true"/>
```

3. Save and close the file.

► **To set EAR file class-loading to isolation mode:**

1. Navigate to the `[appserver root]/server/all/deploy/jbossweb-tomcat50.sar/META-INF` directory and open the `jboss-service.xml` file in a text editor.
2. Locate and set the following attribute:

```
<attribute name="UseJBossWebLoader">false</attribute>
```

3. Save and close the file.
4. Navigate to the `[appserver root]/server/all/conf` directory and open the `jboss-service.xml` file in a text editor.

5. (LiveCycle PDF Generator) Set `<attribute name="Transaction Timeout">600</attribute>` to an appropriate value for your system. To be applicable, this value must be higher than the value that will be configured at the Job Source (using Administrator after deployment). (See *LiveCycle PDF Generator Administrator Help*.)

6. In the Deployers section of the file, locate the following line:
`<mbean code="org.jboss.deployment.EARDeployer"
name="jboss.j2ee:service=EARDeployer">`

7. Add the following text after the line.

```
<!-- Isolate all ears in their own classloader space -->  
<attribute name="Isolated">true</attribute>  
<!-- Enforce call by value to all remote interfaces -->  
<attribute name="CallByValue">true</attribute>
```

8. Save and close the file.

► **To update the dom4j.jar file:**

1. Go to <http://dom4j.org/> and download a version of the dom4j.jar file later than Implementation-Version 1.5.
2. Rename the file that you downloaded to dom4j.jar.
3. Replace the dom4j.jar file in the `[appserver root]/lib` directory with the updated file that you downloaded.

Note: You must replace the dom4j.jar file because the version of the file included in JBoss 3.2.5 contains `org.jaxen.*` classes. These classes cause conflicts with some components of LiveCycle products.

► **To modify the cluster-service.xml file:**

1. Navigate to the `[appserver root]/server/all/deploy` directory and open the cluster-service.xml file in a text editor.
2. Locate the UDP element and set the following attribute values:

- `mcast_port=<random_port_number>`
- `ip_ttl="1"`
- `ip_mcast="false"`

The `mcast_port` value can be changed to any random port number between 30000 and 60000. Each JBoss instance must use a unique port number; otherwise, JBoss attempts to cluster the servers, which will impact the performance.

3. Save and close the file.

Configuring JBoss for Watched Folder

If you are deploying Watched Folder for use with LiveCycle Workflow and LiveCycle Assembler, you must perform the following procedure on the application server where LiveCycle Workflow and Watched Folder will be deployed.

► **To configure JBoss for Watched Folder:**

- Copy the `jboss-watchedfolder-service.xml` file from the `[LiveCycle root]/components/watchedFolder/jboss/config` directory to the `[appserver root]/server/all/deploy-hasingleton/jms` directory.

Connecting JBoss for LiveCycle products to the LiveCycle database

You must configure the data source to connect to the LiveCycle database. For JBoss, you can configure a MySQL, Oracle 9i, Oracle 10g, DB2, or SQL Server data source. (See [“Preparing Your Environment” on page 41.](#))

For information about using a type of database other than the default type used for the JBoss JMS database, see the JBoss documentation.

Configuring the MySQL data source

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

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

Before configuring the MySQL data source, you must have already created the LiveCycle database on MySQL. (See [“Creating a MySQL database” on page 41.](#))

When you perform a turnkey installation, the application server and products are automatically configured to interact with the MySQL database, which is automatically installed during the turnkey installation.

► **To obtain and copy the JDBC driver for MySQL:**

1. Obtain the `mysql-connector-java-3.0.15-ga-bin.jar` file from either the `[LiveCycle root]/configurationManager/lib` directory or the MySQL website at www.mysql.com/products/connector/j/.
2. Copy the file to the `[appserver root]/server/all/lib` directory.

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

1. Open a text editor and create a new text file using the following code:

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
  <local-tx-datasource>
    <jndi-name>IDP_DS</jndi-name>
    <connection-url>jdbc:mysql://host_name:port/database_name
    </connection-url>
    <driver-class>com.mysql.jdbc.Driver</driver-class>
    <user-name>database_username</user-name>
    <password>password</password>
    <min-pool-size>1</min-pool-size>
    <max-pool-size>100</max-pool-size>
    <blocking-timeout-millis>20000</blocking-timeout-millis>
    <idle-timeout-minutes>10</idle-timeout-minutes>
    <prepared-statement-cache-size>50</prepared-statement-cache-size>
    <transaction-isolation>TRANSACTION_READ_COMMITTED
    </transaction-isolation>
  </local-tx-datasource>
</datasources>
```

2. Replace the bold text with values that are specific to your LiveCycle database:
 - **host_name**: The name, IP address, or fully-qualified path of the computer that hosts the LiveCycle database.
 - **port**: The port used to access the LiveCycle database. The default port is 3306.
 - **database_name**: The name of the database that stores the LiveCycle data.
 - **database_username** and **password**: The user name and password that the application server uses to access the LiveCycle database.
3. Save the file as adobe-ds.xml in the `[appserver root]/server/all/deploy` directory.
4. Restart JBoss.

Configuring the Oracle data source

To enable JBoss to connect to the Oracle 9i or 10g tablespace that stores LiveCycle data, you must complete the following tasks:

- Obtain and copy the Oracle 9i JDBC driver to the instance of JBoss where you will deploy LiveCycle products.
 - Note:** For Oracle 10g, JDBC driver classes are available with Oracle Client or Database installation, and do not need be downloaded separately. The JDBC driver is available in the `[dbserver root]/jdbc/lib` directory.
- Create a data source file and deploy it to the instance of JBoss where you will deploy LiveCycle products.

Before configuring the Oracle data source, you must have already created the LiveCycle database on Oracle. (See [“Creating an Oracle database” on page 42.](#))

► **To obtain and copy the JDBC driver for Oracle 9i or 10g:**

1. Obtain the ojdbc14.jar file appropriate for your database from the Oracle website at www.oracle.com/technology/software/index.html.
2. Copy the ojdbc14.jar file to the `[appserver root]/server/all/lib` directory.

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

1. Open a text editor and create a new text file using the following code:

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
  <local-tx-datasource>
    <jndi-name>IDP_DS</jndi-name>
    <connection-url>jdbc:oracle:thin:@host_name:port:service_name
    </connection-url>
    <driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
    <user-name>database_username</user-name>
    <password>password</password>
    <!-- Checks the Oracle error codes and messages for fatal errors -->
    <exception-sorter-class-name>
      org.jboss.resource.adapter.jdbc.vendor.OracleExceptionSorter
    </exception-sorter-class-name>
  </local-tx-datasource>
</datasources>
```

For information about the Oracle connection URL, see

www.oracle.com/technology/tech/java/sqlj_jdbc/htdocs/jdbc_faq.htm#05_04.

2. Replace the bold text with values that are specific to your LiveCycle database:
 - **host_name**: The name, IP address, or fully-qualified path of the computer that hosts the LiveCycle database.
 - **port**: The port used to access the LiveCycle database. The default port is 1521.
 - **service_name**: The name of the Oracle database service.
 - **database_username** and **password**: The user name and password that the application server uses to access the LiveCycle database.
3. Save the file as `adobe-ds.xml` in the `[appserver root]/server/all/deploy` directory.
4. Restart JBoss.

Configuring the DB2 data source

To enable JBoss to connect to the DB2 database that stores LiveCycle data, you must complete the following tasks:

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

Before configuring the DB2 data source, you must have already created the LiveCycle database on DB2. (See [“Creating a DB2 database” on page 43](#).)

► **To obtain and copy the JDBC driver for DB2:**

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

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

1. Open a text editor and create a new text file using the following code:

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
  <local-tx-datasource>
    <jndi-name>IDP_DS</jndi-name>
    <connection-url>jdbc:db2://server_name:port/database_name
    </connection-url>
    <driver-class>com.ibm.db2.jcc.DB2Driver
    </driver-class>
    <user-name>database_username</user-name>
    <password>password</password>
    <SelectMethod>Cursor</SelectMethod>
    <min-pool-size>1</min-pool-size>
    <max-pool-size>100</max-pool-size>
    <blocking-timeout-millis>20000</blocking-timeout-millis>
    <idle-timeout-minutes>10</idle-timeout-minutes>
  </local-tx-datasource>
</datasources>
```

2. Replace the bold text with values that are specific to your LiveCycle database:
 - **server_name**: The name of the computer that runs DB2.
 - **port**: The port used to access DB2. The default port is 50000.
 - **database_name**: The name of the DB2 database that stores the LiveCycle data. (See [“Creating the database” on page 41.](#))
 - **database_username** and **password**: The user name and password that the application server uses to access the LiveCycle database.
3. Save the file as `adobe-ds.xml` in the `[appserver root]/server/all/deploy` directory.
4. Restart JBoss.

Configuring the SQL Server data source

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

- Obtain and copy the SQL Server JDBC driver files to the instance of JBoss where you will deploy LiveCycle products.
- Create a SQL Server data source file and deploy it to the instance of JBoss where you will deploy LiveCycle products.

Before configuring the SQL Server data source, you must have already created the LiveCycle database on SQL Server. (See [“Creating a SQL Server database” on page 45.](#))

► **To obtain and copy the JDBC driver for SQL Server:**

1. Obtain and install SQL Server 2000 Driver for JDBC SP3 from the Microsoft website at www.microsoft.com/sql/downloads.
2. Copy the msbase.jar, mssqlserver.jar, and msutil.jar files from the C:\Program Files\Microsoft SQL Server 2000 Driver for JDBC\lib directory to the `[appserver root]/server/all/lib` directory.

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

1. Open a text editor and create a new text file using the following code:

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
<local-tx-datasource>
  <jndi-name>IDP_DS</jndi-name>
  <connection-url>jdbc:microsoft:sqlserver://host_name:port;
  DatabaseName=database_name;SelectMethod=Cursor</connection-url>
  <driver-class>com.microsoft.jdbc.sqlserver.SQLServerDriver
  </driver-class>
  <user-name>database_username</user-name>
  <password>password</password>
  <SelectMethod>Cursor</SelectMethod>
  <min-pool-size>1</min-pool-size>
  <max-pool-size>100</max-pool-size>
  <blocking-timeout-millis>20000</blocking-timeout-millis>
  <idle-timeout-minutes>10</idle-timeout-minutes>
  <check-valid-connection-sql>SELECT 1</check-valid-connection-sql>
</local-tx-datasource>
</datasources>
```

2. Replace the bold text with values that are specific to your LiveCycle database:
 - **host_name**: The name, IP address, or fully-qualified path of the computer that hosts the LiveCycle database.
 - **port**: The port used to access the LiveCycle database. The default port is 1433.
 - **database_name**: The name of the database you created to use as the LiveCycle database.
 - **database_username** and **password**: The user name and password that the application server uses to access the LiveCycle database.
3. Save the file as `adobe-ds.xml` in the `[appserver root]/server/all/deploy` directory.
4. Restart JBoss.

Next step

If you are installing and configuring LiveCycle Workflow and you plan to install and configure BAM Server as well, you must now manually configure JBoss for use with BAM Server. (See [“Creating a BAM metadata database in Derby” on page 76.](#))

If you are not installing LiveCycle Workflow, you must now deploy the products to the application server database. (See [“Deploying to JBoss” on page 63.](#))

This chapter describes how to deploy your LiveCycle products to JBoss:

- [“About deploying LiveCycle products” on page 62](#)
- [“Deploying to JBoss” on page 63](#)
- [“Applying the connection pool patch for Watched Folder” on page 64](#)
- [“Viewing log files” on page 64](#)

About deploying LiveCycle products

Before you deploy LiveCycle products, ensure that you have met the following requirements:

- You have installed the required software and files and know the location of the directories you will be working with. (See [“System requirements” on page 13.](#))
- You have installed LiveCycle products. (See [“Installing LiveCycle Products” on page 34.](#))
- You have configured JBoss. (See [“Manually Configuring JBoss” on page 52.](#))
- You have run Configuration Manager to configure and assemble the products according to your system and application server requirements. If you need to make any further changes to the properties after deployment, you can run Configuration Manager to make the changes, and then redeploy the updated EAR file. (See [“Configuring LiveCycle Products for Deployment” on page 48.](#))

If you have a web server installed, see your web server documentation for information on the configuration required to allow access to the application server.

JBoss directory name

This chapter refers to the JBoss 3.2.5 home directory as *[appserver root]* and the location where LiveCycle products and components are installed as *[LiveCycle root]*. (See [“Conventions used in this guide” on page 8.](#))

Summary of deployable components

This table lists the deployable components for LiveCycle products.

Component	LiveCycle product
adobe-ds.xml	All
adobe-FontManager.ear	All
adobe-service.xml	All
DocumentServicesLibrary.jar	All
LCM.ear	All

Component	LiveCycle product
LiveCycle.ear	All
adobe-Assembler7.ear	LiveCycle Assembler
mysql-connector-java-3.0.15-ga-bin.jar	LiveCycle Assembler and LiveCycle PDF Generator
FormsIVS.ear	LiveCycle Forms
pdfg-all.ear	LiveCycle PDF Generator Professional and LiveCycle PDF Generator Elements
pdfg-ps-all.ear	LiveCycle PDF Generator for PostScript
adobe-printSubmitter.ear	LiveCycle Print

Deploying to JBoss

You deploy LiveCycle products to JBoss 3.2.5 by copying the deployable components to the deployment directory. JBoss can be running or stopped when you copy the files to the directory. After you copy the files, you must start or restart the server to ensure that the services start correctly.

► To deploy LiveCycle products to JBoss:

1. Copy the following files from the *[LiveCycle root]/configurationManager/export* directory to the *[appserver root]/server/all/deploy* directory:
 - adobe-FontManager.ear
 - LiveCycle.ear
2. Copy the LCM.ear file from the *[LiveCycle root]/configurationManager/deploy/jboss* directory to the *[appserver root]/server/all/deploy* directory.
3. Copy the adobe-service.xml file from the *[LiveCycle root]/configurationManager/deploy/jboss* directory to the *[appserver root]/server/all/deploy/jms* directory.
4. Copy the DocumentServicesLibrary.jar file from the *[LiveCycle root]/components/csa/jboss/lib/adobe* directory to the *[appserver root]/server/all/lib* directory.
5. (LiveCycle Assembler) Copy the adobe-Assembler7.ear file from the *[LiveCycle root]/configurationManager/export* directory to the *[appserver root]/server/all/deploy* directory.
6. (LiveCycle Forms) Copy the FormsIVS.ear file from the *[LiveCycle root]/components/forms/samples/common* directory to the *[appserver root]/server/all/deploy* directory.
7. (LiveCycle PDF Generator Professional, LiveCycle PDF Generator Elements) Copy the pdfg-all.ear file from the *[LiveCycle root]/configurationManager/export* directory to the *[appserver root]/server/all/deploy* directory.

8. (LiveCycle PDF Generator for PostScript) Copy the pdfg-ps-all.ear file from the `[LiveCycle root]/configurationManager/export` directory to the `[appserver root]/server/all/deploy` directory.
9. Restart JBoss to ensure that the applications start up.

Note: When you start JBoss, you may see several error messages in the JBoss server log file. These messages occur if you have not yet initialized the database.

Applying the connection pool patch for Watched Folder

When Watched Folder is under load for a certain amount of time (in hours), JBoss 3.2.5 throws an exception because all of the database connections in the connection pool are busy. This is a known issue in JBoss 3.2.5. The connection pool patch modifies the `jboss-jca.sar` file.

Apply this patch when using Watched Folder in a production environment or when testing Watched Folder under load in a test environment. (The patch is not required for using Watched Folder for basic demonstration or test purposes.)

► To apply the connection pool patch to JBoss 3.2.5:

1. Stop the JBoss service.
2. Replace the `jboss-jca.sar` file in the `[appserver root]\server\all\deploy` directory with the `jboss-jca.sar` file in the `[LiveCycle root]\configurationManager\deploy\jboss\` directory.
3. Restart the JBoss service.

Viewing log files

Events, such as run-time or startup errors, are recorded to the application server log files. If you have any problems deploying to the application server, you can use the log files to help you find the problem. You can open the log files using any text editor.

The following log files are located in the `[appserver root]/server/all/log` directory:

- `boot.log`
- `server.log.[yyyy-mm-dd]`
- `server.log`

Next step

You must now initialize the database using Configuration Manager. (See [“Initializing the Database” on page 65.](#))

This chapter describes how to manually initialize the LiveCycle database.

If you are using User Management with LiveCycle products, you must run Configuration Manager to initialize the database for integration with User Management. When you run Configuration Manager, the application server must be running. Initializing the database is a process that prepares a database for use with LiveCycle products by creating tables and optionally adding data to them. You only need to initialize the database the first time you deploy the product.

During database initialization, provide an SSL-enabled port only if the application server has trusted SSL certificates. If the application server does not have trusted SSL certificates, a non-SSL enabled port must be provided to complete database initialization. (See [“Configuring SSL on JBoss” on page 70.](#))

This chapter assumes that you have configured the LiveCycle database for integration with the product and have deployed the product to the application server. If you have not done this, see [“Preparing Your Environment” on page 41](#) and [“Manually Deploying to JBoss” on page 62.](#)

► **To initialize the database:**

1. Start the application server.
2. Start Configuration Manager by navigating to the *[LiveCycle root]/configurationManager* directory and entering the following command:
 - (Windows) `ConfigurationManager.exe`
 - (Linux) `ConfigurationManager.bin`
3. On the Welcome screen, click **Next**.
4. On the Configuration Mode screen, select **Custom Configuration Wizard** and click **Next**.
5. If prompted, on the Configuration Preferences screen, select either **Use Previously Entered Values** or **Revert to Default Values**, and then click **Next**.
6. On the Product Selection screen, select **Foundation**, select the products that you want to configure and deploy, and then click **Next**.
7. On the Task Selection screen, select **Bootstrap database** and click **Next**.

Note: You must only select the Bootstrap database; no other options are required.
8. On the Application Server URL screen, type the URL that Configuration Manager can use to connect to the application server in the **Application Server Base URL** box, and then click **Next**. On the Prepare database for LiveCycle screen, click **Next**.
9. On the Database Initialization screen, click **Initialize Database Now** and, when the task is complete, click **Next**.
10. On the Tasks Completed screen, click **Finish**.

Next step

You can now access User Management and Administrator. (See [“Post-deployment” on page 25.](#))

You can also configure LiveCycle products to access LDAP. (See [“Configuring LiveCycle Products to Access LDAP” on page 67.](#))

If you are using LiveCycle Workflow, you can now install LiveCycle Workflow Designer. (See [“Installing LiveCycle Workflow Designer” on page 98.](#))

This chapter describes how to configure LiveCycle products with LDAP or LDAP over SSL (LDAPS).

- [“Configuring LiveCycle products with LDAP” on page 67](#)
- [“Configuring LiveCycle products with LDAPS” on page 68](#)

Caution: If you are installing and deploying more than one LiveCycle product, you must consult the appropriate *Installing and Configuring* guide to obtain specific SSL and security settings for each of the LiveCycle products you have installed.

Configuring LiveCycle products with LDAP

Use the following procedure as a guideline when configuring User Management to support authentication using LDAP.

➤ **To configure User Management with LDAP:**

1. Open a web browser, navigate to `http:// [host name] : [port] /adminui`, and log in. (See [“Accessing Administrator” on page 25.](#))
2. Select **Settings > User Management > Domain Management** and click **New Domain**.
3. In the **ID** box, type a unique identifier for the domain.
4. In the **Name** box, type a descriptive name for the domain.
5. Click **Add Authentication** and, in the Authentication Provider list, select **LDAP**.
Note: JAAS has been deprecated for User Management in LiveCycle 7.2.
6. Click **OK**.
7. Click **Add Directory**.
8. Under **Populate Page With**, select a directory settings option such as **Default Sun ONE values**.
9. Specify values in the **Server, Port, SSL** and **Binding** boxes as required. For details on the settings, see Directory settings in the online help.
10. Configure the **User Settings** and **Group Settings** as required. For details on the settings, see Directory settings in the online help.
11. (Optional) Test your configuration:
 - Click **Test**.
 - On the Test Directory pane, in the **Find** box, enter an object name and, in the **using** box, select the object’s type, such as **Login ID**.
 - Click **Test**. If successful, your object’s details will be displayed, then click **Back**.
12. Click **OK** to exit the Add Directory page, and click **OK** again to exit.

Configuring LiveCycle products with LDAPS

Use the following procedure as a guideline when configuring User Management to support authentication using LDAPS.

► **To configure User Management with LDAPS:**

1. Enable SSL on the directory server. For details, see the documentation provided by your directory vendor.
2. Export a client certificate from the directory server.
3. Use the `keytool` program to import the client certificate file into the default JVM certificate store. The procedure for this varies depending on your JVM and client install paths. For example, if you are using JDK 1.4.2, from a command prompt, type the following code:

```
keytool -import -file client_certificate -alias myalias -keystore  
jdk142_08/jre/lib/security/cacerts
```

When prompted, enter the password (for Java, the default password is `changeit`). You will receive a message stating that the certificate was imported successfully.

4. Enable SSL in User Management. To access the User Management settings, select **Settings > User Management** in the Administrator pages. When configuring the directory settings, select **Yes** for the **SSL** property, and change the **Port** property accordingly. The default port number is 636.

Troubleshooting

If you experience problems when using SSL, use an LDAP browser to check whether it can access the LDAP system when using SSL. If the LDAP browser cannot access the LDAP system, the issue is related to the configuration of your certificate and application server. If the LDAP browser can access the LDAP system, User Management is not configured properly.

Part III: Post-Deployment Configuration

This section of the guide describes the additional configuration tasks that you need to perform after LiveCycle products are deployed to the application server and the LiveCycle database is initialized.

This chapter describes how to create SSL credentials and configure SSL on the application server to enhance the security of communication with your application server.

The information in this chapter applies to both turnkey and manual installations.

Note: It is recommended that you complete the installation, configuration and deployment of your LiveCycle products and ensure that the products are running correctly before configuring SSL on the application server.

It is important to ensure that the security settings configured on the application server and in the LiveCycle.ear file are consistent.

Caution: If you are installing and deploying more than one LiveCycle product, you must consult the appropriate *Installing and Configuring* guide to obtain specific SSL and security settings for each of the LiveCycle products you have installed.

If you have not already enabled SSL in the Data Manager Module (assembled as part of the LiveCycle.ear file), run Configuration Manager to reconfigure and reassemble the products with SSL enabled, and then redeploy the LiveCycle.ear file. The SSL password that you specify in Configuration Manager must match the password that you provide when configuring SSL on the application server. (See [“Configuring LiveCycle Products for Deployment” on page 48.](#))

To configure SSL on the application server, you must perform the following tasks:

- [“Creating an SSL Credential” on page 70](#)
- [“Configuring SSL on JBoss” on page 72](#)

Creating an SSL Credential

To configure SSL on JBoss, you need an SSL credential for authentication. You can use the IBM Key Management tool that is installed with Java keytool to perform the following tasks to create a credential:

- Create a public/private key pair.
- Wrap the public key in an X.509 v1 self-signed certificate that is stored as a single-element certificate chain.
- Store the certificate chain and the private key in a new keystore.

The keytool command is typically located in your Java jre/bin directory. You use the command `keytool -genkey` to start the keytool and generate the key pair. The keytool command must include several options and option values, which are listed in the following table. The keytool command is typically located in the Java jre/bin directory and must include several options and option values, which are listed in the following table.

Keytool option	Description	Option value
<code>-alias</code>	The alias of the keystore.	<code>ads-credentials</code>

Keytool option	Description	Option value
-keyalg	The algorithm to use to generate the key pair.	RSA You can use a different algorithm, depending on your company's policy.
-keystore	The location and name of the keystore file. The location can include the absolute path of the file, or can be relative to the current directory of the command prompt where the keytool command is entered.	[appserver_root]/server/all/svcnative/ads-ssl.jks
-validity	The number of days that the certificate is considered valid.	3650 You can use a different value, depending on your company's policy.
-storepass	The password that protects the contents of the keystore.	The keystore password must correspond with the SSL credential password that was specified for the Data Manager Module using Configuration Manager.
-keypass	The password that protects the private key of the key pair.	Use the same password that you used for the -storepass option. The key password must be at least 6 characters in length.
-dname	The distinguished name that identifies the person who owns the keystore.	"CN= [User name], OU= [Group Name], O= [Company Name], L= [City Name], S= [State or province], C= [Country Code] " <ul style="list-style-type: none"> • [User name] is the identification of the user who owns the keystore. • [Group Name] is the identification of the corporate group to which the keystore owner belongs. • [Company Name] is your organization's name. • [City Name] is the city in which your organization is located. • [State or province] is the state or province in which your organization is located. • [Country Code] is the two-letter code for the country in which your organization is located.

You can run keytool using a single command that includes all of the information needed to create the keystore. For more information about using the keytool command, see the keytool.html file that is part of your JDK documentation.

► **To create an SSL credential:**

1. From a command prompt, navigate to `[appserver_root]/server/all/svcnative`.
2. Enter the following command:

```
[JAVA_HOME]/bin/keytool -genkey -v -alias ads-credentials -keyalg RSA  
-keystore "ads-ssl.jks" -validity 3650 -storepass store_password  
-keypass key_password -dname "CN=User ID, OU=Group Name, O=Company Name,  
L=City Name, S=State, C=Country Code"
```

Note: You must replace `[JAVA_HOME]` with the directory where the JDK is installed, and replace the text in bold with values that correspond with your environment.

The keystore file is created in the `[appserver_root]/server/all/svcnative` directory.

Configuring SSL on JBoss

You can enable SSL on the application server by editing the files indicated in this section. When SSL is configured, you must manually start JBoss.

► **To enable SSL on JBoss:**

1. In a text editor, open the `jacorb.properties` file from the `[appserver_root]/server/all/conf` directory and edit the SSL configuration section so that it contains the following text:

```
#####  
### SSL Configuration ###  
#####  
# the qualified classname of the ssl socket factory class  
jacorb.ssl.socket_factory=org.jacorb.security.ssl.sun_jsse.SSLSocket  
Factory  
# the qualified classname of the ssl server socket factory class  
jacorb.ssl.server_socket_factory=org.jacorb.security.ssl.sun_jsse.  
SSLServerSocketFactory  
# The name and location of the keystore. This should be absolute  
# to the directory where this property file resides.  
#jacorb.security.keystore=c:/jboss-3.2.5/server/all/svcnative/  
ads-ssl.jks  
jacorb.security.keystore=<--replace with absolute path of ads-ssl.jks-->  
jacorb.security.keystore_password=yourSSLpassword  
# trusted ca certs are also in the same keystore  
jacorb.security.jsse.trustees_from_ks=on  
jacorb.security.support_ssl=on  
#client side ssl supported or enforced?  
jacorb.security.ssl.client.supported_options=60  
jacorb.security.ssl.client.required_options=0  
#server side ssl supported or enforced?  
jacorb.security.ssl.server.supported_options=60  
jacorb.security.ssl.server.required_options=60
```

2. Save the `jacorb.properties` file.

3. In a text editor, open the server.xml file from the `[appserver root]/server/all/deploy/jbossweb-tomcat50.sar` directory, and then uncomment the following section:

```
<!-- SSL/TLS Connector configuration using the SSL domain keystore
<Connector port="8443" address="{jboss.bind.address}"
    maxThreads="100" minSpareThreads="5" maxSpareThreads="15"
    scheme="https" secure="true" clientAuth="false"
    keystoreFile="{jboss.server.home.dir}/conf/chap8.keystore"
    keystorePass="rmi+ssl" sslProtocol = "TLS" />
-->
```

4. Change the value for the `keystoreFile` attribute to `"{jboss.server.home.dir}/svcnative/ads-ssl.jks"`, which is the location of your keystore file.
5. Change the value of the `keystorePass` attribute to the keystore password that you specified when you created the keystore.
6. Save the server.xml file.
7. In a text editor, open the `iiop-service.xml` file from the `[appserver_root]/server/all/deploy` directory, and then uncomment the following text:

```
<!-- (uncomment to use IIOP over SSL)
<mbean code="org.jboss.security.plugins.JaasSecurityDomain"
    name="jboss.security:service=JaasSecurityDomain, domain=IIOP+SSL">
    <constructor>
        <arg type="java.lang.String" value="IIOP+SSL"/>
    </constructor>
    <attribute name="KeyStoreURL">iiop-over-ssl.keystore</attribute>
    <attribute name="KeyStorePass">iiop-over-ssl</attribute>
</mbean>
-->
```

8. Set the values for the `KeyStoreURL` and `KeyStorePass` attributes to match your configuration:

```
<attribute name="KeyStoreURL">
    [appserver_root]/server/all/svcnative/ads-ssl.jks
</attribute>

<attribute name="KeyStorePass">
    keystore password
</attribute>
```

9. In the CORBA ORB service section, uncomment the following text:

```
<!-- (uncomment to use IIOP over SSL)
<attribute name="SecurityDomain">java:/jaas/IIOP+SSL</attribute>
<depends>jboss.security:service=JaasSecurityDomain, domain=IIOP+SSL
</depends>
</mbean>
-->
```

10. Save the iiop-service.xml file.
11. Restart the application server.
12. To access Administrator using SSL, type the following URL in a web browser:
`https:// [host name] : [port] /adminui`
The default port for JBoss is 8443.

Next step

If you are installing LiveCycle Workflow Business Activity Monitor, you can now configure an application server for hosting BAM Server. (See [“Manually Configuring JBoss for BAM Server” on page 75.](#)) You can also install LiveCycle Workflow Designer. (See [“Installing LiveCycle Workflow Designer” on page 98.](#))

This chapter applies to LiveCycle Workflow licenses that permit the use of Business Activity Monitor only.

This chapter describes how to create the LiveCycle Workflow Business Activity Monitor metadata database and configure the JBoss application server for BAM Server, which is a component of LiveCycle Workflow.

If you are installing the products using the turnkey method, JBoss and MySQL are installed and configured automatically and you do not need to perform the procedures in this chapter.

This chapter refers to the home directory of the instance of JBoss that you configured for BAM Server as `[jboss_bam_root]`. (See [“Conventions used in this guide” on page 8](#).)

This chapter assumes that you have configured the LiveCycle database. If you have not done this, see [“Preparing Your Environment” on page 41](#).

Installing JBoss Application Server

If you are manually configuring JBoss for running BAM Server, you must download and install JBoss Application Server 3.2.6. You can obtain JBoss at this location:

<http://labs.jboss.com/portal/jbossas/download>

Creating the BAM metadata database

You must create a LiveCycle Workflow Business Activity Monitor 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 Monitor metadata can grow quite large, you must allocate at least 50 MB for the BAM metadata database. For production deployments, allocate at least 200 MB.

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

User accounts

You must also create a user account that BAM Server can use to connect to the Business Activity Monitor metadata database. The user account must have create, modify, and update privileges on the database.

For Oracle, the database user account must be associated with a tablespace that you create specifically for Business Activity Monitor metadata.

Creating a BAM metadata database in Derby

If you are using MySQL as your LiveCycle database, you must create a BAM metadata database in Derby.

► **To create a BAM metadata database in Derby:**

1. Download the `derby.jar` and `derbytools.jar` files from the Apache Derby website (<http://db.apache.org/derby/index.html>).
2. Set your classpath to include the following JAR files:
 - **derby.jar:** Contains the Derby engine and the Derby Embedded JDBC driver
 - **derbytools.jar:** Provides the `ij` tool used to create the database.
3. Navigate to or create the directory where you want to create your Derby database for BAM Server metadata (for example, `Adobe/LiveCycle/ljboss_bam_home`).

4. From the command line, start the `ij` tool using the following command:

```
java org.apache.derby.tools.ij
```

The following output (where `<version#>` is the current version of Derby) confirms that `ij` started successfully:

```
ij version <version#>
ij>
```

The following error message indicates that the JAR files are not in your classpath.

```
java org.apache.derby.tools.ij Exception in thread "main"
java.lang.NoClassDefFoundError: org/apache/derby/tools/ij
```

Repeat step [2](#) to set the classpath properly.

5. Enter the following command to create a database, where `<db_name>` is the name you want to assign to the database:

```
ij>connect 'jdbc:derby:<db_name>;create=true';
```

For example:

```
ij>connect 'jdbc:derby:MyDerbyDB;create=true';
```

6. Enter the following command to quit the `ij` tool:

```
ij>exit;
```

7. To confirm that you created the database successfully, list the contents of the directory where you created your Derby database. The directory should contain these items:
 - A subdirectory with the name you specified in step [5](#) (for example, `MyDerbyDB`) with the files that make up the database you created
 - A log file (`derby.log`) describing the steps performed during successful creation of the Derby database or the errors that led to failure to create the database

Configuring JBoss for BAM Server

If you are installing LiveCycle Workflow and manually deploying to JBoss, you must manually install and configure JBoss for BAM Server to run correctly.

BAM Server must be deployed on a different instance of JBoss than the one where LiveCycle Workflow Server is deployed. You must configure the following areas on the instance of JBoss that runs BAM Server:

- Install the database drivers that JBoss requires to communicate with the Business Activity Monitor metadata database and the LiveCycle database. (See [“Installing database drivers on JBoss for BAM Server” on page 77.](#))
- Create a data source file so that BAM Server can connect to the Business Activity Monitor metadata database. (See [“Connecting JBoss to the BAM metadata database” on page 77.](#))
- Create a data source file so that BAM Server can connect to the LiveCycle database. (See [“Connecting JBoss for BAM Server to the LiveCycle database” on page 81.](#))
- Configure the Java virtual machine (JVM) that runs the JBoss application server (See [“Configuring required JBoss JVM options” on page 83.](#))
- Change the thread configuration on JBoss. (See [“Modifying the JBoss thread configuration” on page 85.](#))
- Configure JBoss so that Business Activity Monitor events are not duplicated in JBoss logs. (See [“Configuring JBoss logging” on page 85.](#))
- Change JBoss ports to avoid conflicts with other applications that are running on the same computer. (See [“Customizing port numbers” on page 86.](#))

Installing database drivers on JBoss for BAM Server

To enable BAM Server to connect to the Business Activity Monitor metadata database and the LiveCycle database, you need to install the drivers for the types of databases that you are using.

► To install the JDBC driver for SQL Server:

- Copy the SQL Server 2000 database driver files (msbase.jar, mssqlserver.jar, msutil.jar) to the `[jboss bam root]/server/default/lib` directory.

Note: LiveCycle and BAM Server run in completely different JBoss trees, which means drivers are needed in each tree, as a result you will have to reinstall these drivers for BAM Server. (See [“To obtain and copy the JDBC driver for SQL Server:” on page 61.](#))

Connecting JBoss to the BAM metadata database

You must configure a connection to the Business Activity Monitor metadata database on the instance of JBoss that runs BAM Server so that it can retrieve information about the process metrics that it monitors.

The procedure to configure the connection that you use depends on the type of database that you are using to store Business Activity Monitor metadata:

- [“Connecting to the BAM metadata database on SQL Server” on page 78](#)
- [“Connecting to the BAM metadata database on Derby” on page 79](#)

Connecting to the BAM metadata database on SQL Server

To enable BAM Server to connect to the SQL Server database that stores Business Activity Monitor metadata, you must create a data source file and deploy the file to the instance of JBoss that runs BAM Server. You must also modify the default data-type mappings on JBoss.

► To create a data source for the Business Activity Monitor metadata database on SQL Server:

1. Open a text editor and create a new text file using the following code:

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
<no-tx-datasource>
<jndi-name>com.celequest.metadata.metaDataSource</jndi-name>
<connection-url><!--Insert jdbcURL --></connection-url>
<driver-class>com.microsoft.jdbc.sqlserver.SQLServerDriver</driver-class>
<min-pool-size>5</min-pool-size>
<max-pool-size>20</max-pool-size>
<idle-timeout-minutes>5</idle-timeout-minutes>
<track-statements>false</track-statements>
<application-managed-security/>
</no-tx-datasource>
</datasources>
```

2. Replace `<!--Insert jdbcURL -->` inside the `connection-url` element with the following text:

```
jdbc:microsoft:sqlserver:// [host name] : [port] ;SelectMethod=cursor;
databaseName= [dbname] ;user= [user_name] ;password= [dbpassword]
```

- `[host name]` is the name of the computer where SQL Server 2000 is running.
- `[port]` is the port used to access the database. The default port for SQL Server 2000 is 1433.
- `[dbname]` is the name of the BAM metadata database.
- `[user_name]` is the name of the database user account that can access the BAM metadata database.
- `[dbpassword]` is the password for the user name specified for `[user_name]`.

3. Save the file as `celequest_metadata-ds.xml` in the `[jboss bam root]/server/default/deploy` directory.

► To modify the SQL Server data-type mappings on JBoss:

1. Navigate to the `[jboss bam root]/server/default/conf` directory and open the `standardjbosscmp-jdbc.xml` file in a text editor.
2. In the `<defaults>` section near the top of the file, change the following parameter as indicated by the bold text:

```
<datasource>java:/com.celequest.metadata.metaDataSource</datasource>
<datasource-mapping>MS SQLSERVER2000</datasource-mapping>
<create-table>false</create-table>
...
<list-cache-max>10000</list-cache-max>
```

3. Locate the type-mappings element that has a name child element that contains the text MS SQLSERVER2000, and insert the following code below the final function-mapping element:

```
<mapping>
  <java-type>com.celequest.metadata.ejb.BigString</java-type>
  <jdbc-type>VARBINARY</jdbc-type>
  <sql-type>IMAGE</sql-type>
</mapping>
<mapping>
  <java-type>com.celequest.jar.Jar</java-type>
  <jdbc-type>VARBINARY</jdbc-type>
  <sql-type>IMAGE</sql-type>
</mapping>
<mapping>
  <java-type>com.celequest.alert.AlertFiring</java-type>
  <jdbc-type>VARBINARY</jdbc-type>
  <sql-type>IMAGE</sql-type>
</mapping>
<mapping>
  <java-type>com.celequest.alert.AlertAck</java-type>
  <jdbc-type>VARBINARY</jdbc-type>
  <sql-type>IMAGE</sql-type>
</mapping>
<mapping>
  <java-type>com.celequest.alert.AlertInstance</java-type>
  <jdbc-type>VARBINARY</jdbc-type>
  <sql-type>IMAGE</sql-type>
</mapping>
<mapping>
  <java-type>com.celequest.util.misc.SerializableWrapper</java-type>
  <jdbc-type>VARBINARY</jdbc-type>
  <sql-type>IMAGE</sql-type>
</mapping>
```

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

Connecting to the BAM metadata database on Derby

To enable BAM Server to connect to the Derby database that stores Business Activity Monitor metadata, you must configure the data source for the instance of JBoss that runs BAM Server.

► To configure a data source on JBoss:

1. Navigate to the directory `[jboss bam root]/server/default/deploy` and open the file `celequest_metadata-ds.xml` in a text editor.
2. Set the following parameters:

Parameter	Required value
<code><jndi-name></code>	Specify the JNDI name <code>com.celequest.metadata.metaDataSource</code>

Parameter	Required value
<connection-url>	Specify the complete connection URL for the Derby database: <code>jdbc:derby:[directory_path]</code> where <code>[directory_path]</code> is the location of the database you created in step 5 of the procedure “Creating a BAM metadata database in Derby” on page 76 . The connection URL can be one of these: <ul style="list-style-type: none">• The absolute path on your file system• A path that is relative to the value of the system property <code>derby.system.home</code> (if it is set); otherwise, a path that is relative to the current working directory of the JVM For more information, see http://wiki.apache.org/db-derby/DatabaseNamesRelativeAbsolute .
<driver-class>	Specify the driver class: <code>org.apache.derby.jdbc.EmbeddedDriver</code>
<user-name>	Set a valid user name. Note: The specified user should have read privileges.
<password>	Enter the password for the user.
<min-poll-size>	Set to 5.
<max-poll-size>	Set to 200.
<track-statements>	Set to false .

For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
  <no-tx-datasource>
    <jndi-name>com.celequest.metadata.metaDataSource</jndi-name>
    <connection-url>jdbc:derby:C:\jboss-3.2.6\MyDerbyDB</connection-url>
    <driver-class>org.apache.derby.jdbc.EmbeddedDriver</driver-class>
    <!-- The login and password -->
    <user-name></user-name>
    <password></password>
    <min-pool-size>5</min-pool-size>
    <max-pool-size>200</max-pool-size>
    <idle-timeout-minutes>5</idle-timeout-minutes>
    <track-statements>false</track-statements>
    <application-managed-security/>
  </no-tx-datasource>
</datasources>
```

3. Save the edited `celequest_metadata-ds.xml` file.

Connecting JBoss for BAM Server to the LiveCycle database

You must configure a connection to the LiveCycle database on the instance of JBoss that runs BAM Server so that it can retrieve information about the process run-time data.

The procedure to configure the connection that you use depends on the type of database that you are using to store LiveCycle data:

- [“Connecting to the LiveCycle database on SQL Server” on page 81](#)
- [“Connecting to the LiveCycle database on MySQL” on page 82](#)

Connecting to the LiveCycle database on SQL Server

To enable BAM Server to connect to the LiveCycle database, you need to create a data source file and deploy it to the instance of JBoss that runs BAM Server.

► To create a data source for the LiveCycle database on SQL Server 2000:

1. Open a text editor and create a new text file using the following code:

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
<no-tx-datasource>
<jndi-name>com.celequest.adobe</jndi-name>
<connection-url><!--Insert jdbcURL --></connection-url>
<driver-class>com.microsoft.jdbc.sqlserver.SQLServerDriver
</driver-class>
<!-- The login and password -->
<user-name><!--Insert dbUsername --></user-name>
<password><!--Insert dbPassword --></password>
<min-pool-size>5</min-pool-size>
<max-pool-size>20</max-pool-size>
<idle-timeout-minutes>5</idle-timeout-minutes>
<track-statements>>false</track-statements>
<application-managed-security/>
</no-tx-datasource>
</datasources>
```

2. Replace `<!--Insert jdbcURL -->` in the `connection-url` element with the following text:

```
jdbc:microsoft:sqlserver://[host name]:[port];SelectMethod=cursor;
databaseName=[dbname];user=[dbuser];password=[dbpassword]
```

- `[host name]` is the name of the computer where SQL Server 2000 is running.
 - `[port]` is the port used to access the LiveCycle database. The default port for SQL Server 2000 is 1433.
 - `[dbname]` is the name of the LiveCycle database.
 - `[dbuser]` is the user name of the database user account that can access the LiveCycle database.
 - `[dbpassword]` is the password for the user name specified for the `[dbuser]` parameter.
3. Replace `<!--Insert dbUsername -->` in the `user-name` element with the user name of the database user account that can access the LiveCycle database.

4. Replace `<!--Insert dbPassword -->` in the `<password>` element with the password for the user name specified for the `<user-name>` element.
5. Save the file as `adobe-ds.xml` in the `[jboss bam root]/server/default/deploy/` directory.

Connecting to the LiveCycle database on MySQL

To enable BAM Server to connect to the LiveCycle database, you need to create a data source file and deploy it to the instance of JBoss that runs BAM Server.

► To install the JDBC driver for MySQL

1. Obtain the **mysql-connector-java-3.0.15-ga-bin-jar** file from the MySQL website: <http://www.mysql.com/products/connector/j/>
2. Copy the file to the `[jboss_bam_root]/server/default/lib` directory.

► To create a data source for the LiveCycle database on MySQL:

1. Open a text editor and create a new text file using the following code:

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
<no-tx-datasource>
<jndi-name>com.celequest.adobe</jndi-name>
<connection-url><![CDATA[<!--Insert jdbcURL -->]]>
</connection-url>
<driver-class>com.mysql.jdbc.Driver</driver-class>
<!-- The login and password -->
<user-name><!--Insert dbUsername --></user-name>
<password><!--Insert dbPassword --></password>
<min-pool-size>5</min-pool-size>
<max-pool-size>20</max-pool-size>
<idle-timeout-minutes>5</idle-timeout-minutes>
<track-statements>false</track-statements>
<application-managed-security/>
</no-tx-datasource>
</datasources>
```

2. Replace `<!--Insert jdbcURL -->` in the `connection-url` element with the following text:

```
jdbc:mysql://[host_name]/[db_name]?user=[dbUsername]&password=
[dbpassword]&autoReconnect=true&autoReconnectForPools=true
```

- `[host_name]` is the name of the computer on which MySQL is running.
 - `[db_name]` is the name of the LiveCycle database.
 - `[dbUsername]` is the user name of the database user account that can access the LiveCycle database.
 - `[dbpassword]` is the password for the user name specified for the `[dbUsername]` parameter.
3. Replace `<!--Insert dbUsername -->` in the `user-name` element with the user name of the database user account that can access the LiveCycle database.
 4. Replace `<!--Insert dbPassword -->` in the `password` element with the password for the user name specified for the `<user-name>` element.
 5. Save the file as `adobe-ds.xml` in the `[jboss bam root]/server/default/deploy` directory.

Configuring required JBoss JVM options

You must configure the Java virtual machine (JVM) to support BAM Server.

► **To configure the JBoss JVM options:**

1. Navigate to the `[jboss bam root]/bin` directory and open the startup script in a text editor:
 - (Windows) `run.bat`
 - (Linux) `run.sh`
2. Change the `JAVA_OPTS` memory settings to 128M minimum and 768M maximum, and include `-server` to improve the performance:
 - (Windows) `set JAVA_OPTS=%JAVA_OPTS% -server -Xms128m -Xmx768m`
 - (Linux) `JAVA_OPTS="$JAVA_OPTS -server -Xms128m -Xmx768m"`

For large data loads, consider increasing the maximum memory to `-Xmax1536m` (1.5 GB).
3. Add a `JAVA_OPTS` setting to disable the duplicate logging of system output. This is a workaround to JBoss bug 877974:
 - Windows:

```
set JAVA_OPTS=%JAVA_OPTS%
-Dorg.jboss.logging.Log4jService.catchSystemOut=false
-Dorg.jboss.logging.Log4jService.catchSystemErr=false
```
 - Linux:

```
JAVA_OPTS="$JAVA_OPTS
-Dorg.jboss.logging.Log4jService.catchSystemOut=false
-Dorg.jboss.logging.Log4jService.catchSystemErr=false"
```
4. Add a `JAVA_OPTS` setting to enable UTF-8 character support:
 - (Windows) `set JAVA_OPTS=%JAVA_OPTS% -Dfile.encoding=utf8`
 - (Linux) `JAVA_OPTS="$JAVA_OPTS -Dfile.encoding=utf8"`
5. Add a `JAVA_OPTS` setting to increase the thread stack size to 300:
 - (Windows) `set JAVA_OPTS=%JAVA_OPTS% -XX:ThreadStackSize=300`
 - (Linux) `JAVA_OPTS="$JAVA_OPTS -XX:ThreadStackSize=300"`
6. (Optional) Add a `JAVA_OPTS` setting to include optional JVM options:
 - (Windows) `set JAVA_OPTS=%JAVA_OPTS% [JVM options]`
 - (Linux) `JAVA_OPTS="$JAVA_OPTS [JVM options]`

`[JVM options]` are the options that you want to include, in the format `-Dcom.celequest.property.[PARAMETER]=[VALUE]`

For information on these options, see ["Optional JBoss JVM options" on page 84](#).
7. Save and close the startup script file.

Optional JBoss JVM options

You can set several optional JVM parameters that are related to the Business Activity Monitor metadata database, including language, country, and collation strength. These JVM parameters are expressed using commands in the following format:

```
-Dcom.celequest.property.[PARAMETER]=[VALUE]
```

This table provides valid values for `[PARAMETER]` and `[VALUE]`.

Parameter	Description
LANGUAGE	<p>The ISO language code for the server.</p> <p>For example, the following command sets the LANGUAGE parameter to Portuguese:</p> <pre>-Dcom.celequest.property.LANGUAGE=PT</pre>
COUNTRY	<p>The ISO country code for the server.</p>
LOCALESORT	<p>This option specifies whether locale-sensitive string comparisons are performed. Valid values are <code>true</code> and <code>false</code>:</p> <ul style="list-style-type: none">Specify <code>true</code> if you want locale-sensitive string comparisons performed.Specify <code>false</code> if you do not want locale-sensitive string comparisons performed. <p>The default value is <code>false</code>.</p>
STRENGTH	<p>The level of collation strength, which is the extent to which non-English characters are compared and collated.</p> <p>Valid values (in order of least discriminating to most discriminating) are <code>primary</code>, <code>secondary</code>, <code>tertiary</code>, and <code>identical</code>. The default value is <code>tertiary</code>.</p> <p>For more information about collation parameters, see the Sun Java documentation at http://java.sun.com/j2se/1.4.2/docs/api/java/util/Locale.html.</p> <p>Note: This setting is functional only if <code>LOCALESORT</code> is set to <code>true</code>.</p>
DECOMPOSITION	<p>The mode of collation decomposition. Valid values are <code>none</code>, <code>canonical</code>, and <code>full</code>. The default value is <code>canonical</code>.</p> <p>For more information about collation parameters, see the Sun Java documentation at http://java.sun.com/j2se/1.4.2/docs/api/java/util/Locale.html</p> <p>Note: This setting is functional only if <code>LOCALESORT</code> is set to <code>true</code>.</p>

To set multiple properties, separate each command with a space. For example, the following commands set the language to Portuguese and the locale to Brazil:

```
-Dcom.celequest.property.LANGUAGE=PT -Dcom.celequest.property.LOCALE=BR
```

Modifying the JBoss thread configuration

Before deploying BAM Server to JBoss, you must modify the thread configuration in the JBoss server.xml file.

► **To modify the thread configuration:**

1. Navigate to the `[jboss bam root]/server/default/deploy/jbossweb-tomcat50.sar` directory and open the server.xml file.
2. Locate the `<!-- A HTTP/1.1 Connector on port 8080 -->` heading.

Note: The port value could be 9080 if are running the turnkey, and are modifying the JBoss thread.

3. Change the values in the `Connector` element for the HTTP port to match the bold text in the following XML code:

```
<!-- A HTTP/1.1 Connector on port 9080 -->
<Connector port="9080" address="{jboss.bind.address}"
maxThreads="400" minSpareThreads="100" maxSpareThreads="75"
enableLookups="false" redirectPort="8443" acceptCount="100"
connectionTimeout="20000" disableUploadTimeout="true"/>
```

4. (Optional) Add the `compression` and `socketBuffer` attributes to the same `Connector` element to enhance the overall performance of Business Activity Monitor, as shown by the bold text in the following example:

```
<!-- A HTTP/1.1 Connector on port 9080 -->
<Connector port="9080" address="{jboss.bind.address}"
maxThreads="400" minSpareThreads="100" maxSpareThreads="75"
enableLookups="false" redirectPort="8443" acceptCount="100"
connectionTimeout="20000" disableUploadTimeout="true"
compression="on" socketBuffer="1045576" />
```

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

Configuring JBoss logging

Messages sent to Business Activity Monitor logs may also be recorded in the JBoss server.log file. By default, JBoss logs all of the messages published by BAM Server, in addition to its own messages and those of any other applications that it hosts. You can modify the JBoss log4j.xml file to prevent the logging of Business Activity Monitor messages to the JBoss log file.

Caution: Do not perform this procedure when JBoss is running, or JBoss will duplicate messages to both the console and the log file.

Note: By default, Business Activity Monitor servers publish messages of Info severity or greater. You can change which messages they publish with the Business Activity Monitor (BAM) Workbench Administration Console.

► **To restrict JBoss from recording Business Activity Monitor messages:**

1. Stop JBoss.
2. In a text editor, open the log4j.xml file from the `[jboss bam root]/default/conf/` directory.
3. Locate the following XML code:

```
<root>
  <appender-ref ref="CONSOLE"/>
  <appender-ref ref="FILE"/>
</root>
```

4. Delete the `appender-ref` element with the `ref` attribute of value `FILE`. The `root` element should now appear like the following XML code:

```
<root>
  <appender-ref ref="CONSOLE"/>
</root>
```

5. Locate the `Categories` section of the file and add the following `category` element so that JBoss sends all of the messages from the "org" categories to the console, such as those published by JBoss, and does not log messages from the "com" categories (such as `com.celequest`) to the log file:

```
<category name="org">
  <appender-ref ref="FILE"/>
</category>
```

6. Save the log4j.xml file and restart JBoss.

Customizing port numbers

You must modify the ports that JBoss uses so they do not conflict with ports in use by other applications on the host computer, such as the instance of JBoss that runs LiveCycle Workflow Server.

► **To modify the ports:**

1. Navigate to the `[jboss bam root]/server/default/deploy/jbossweb-tomcat50.sar` directory and open the `server.xml` file in a text editor.
2. Modify the HTTP port and redirector port settings, which are located in the `HTTP/1.1 Connector` block:

```
<!-- A HTTP/1.1 Connector on port 9080 -->
<Connector port="9080" address="{jboss.bind.address}"
  maxThreads="400" minSpareThreads="100" maxSpareThreads="75"
  enableLookups="false" redirectPort="18443" acceptCount="100"
  connectionTimeout="20000" disableUploadTimeout="true"/>
```

3. Modify the AJP port and redirector port settings:

```
<!-- A AJP 1.3 Connector on port 18009 -->
<Connector port="18009" address="{jboss.bind.address}"
  enableLookups="false" redirectPort="18443" debug="0"
  protocol="AJP/1.3"/>
```

4. Modify the SSL/TLS port setting:

```
<!-- SSL/TLS Connector configuration using the admin dev1 guide keystore
<Connector port="18443" address="{jboss.bind.address}"
maxThreads="100" minSpareThreads="5" maxSpareThreads="15"
scheme="https" secure="true" clientAuth="false"
keystoreFile="{jboss.server.home.dir}/conf/chap8.keystore"
keystorePass="rmi+ssl" sslProtocol = "TLS" />
-->
```

5. Save and close the server.xml file.

6. Navigate to the *[jboss bam root]/server/default/deploy/http-invoker.sar/META-INF* directory and open the *jboss-service.xml* file in a text editor.

7. Update all occurrences of the *InvokerURLSuffix* attribute:

```
<attribute name="InvokerURLSuffix">:9080/invoker/...
```

8. Save and close the *jboss-service.xml* file.

9. Navigate to the *[jboss bam root]/server/default/conf* directory and open the *jboss-service.xml* file.

10. In the *Class Loading* section, in the element `<mbean code="org.jboss.web.WebService" name="jboss:service=WebService">`, change the port to 18083:

```
<attribute name="Port">18083</attribute>
```

11. In the *JNDI* section, in the element `<mbean code="org.jboss.naming.NamingService" name="jboss:service=Naming" xmbeandd="resource:xmdesc/NamingService-xmbean.xml">`, change the port to 11099:

```
<attribute name="Port">11099</attribute>
```

12. Change the *rmi* port to 11098:

```
<attribute name="RmiPort">11098</attribute>
```

13. In the *Invokers to the JMX node* section, in the element `<mbean code="org.jboss.invocation.pooled.server.PooledInvoker" name="jboss:service=invoker, type=pooled">`, change the *RMIObjectPort* to 14444:

```
<attribute name="RMIObjectPort">14444</attribute>
```

14. Change the *ServerBindPort* to 14445:

```
<attribute name="ServerBindPort">14445</attribute>
```

15. Save and close the *jboss-service.xml* file.

16. Navigate to the *[jboss bam root]/server/default/deploy/jms* directory and open the *uil2-service.xml* file in a text editor.

17. In the element `<mbean code="org.jboss.mq.il.uil2.UILServerILService" name="jboss.mq:service=InvocationLayer, type=UIL2">`, change the *ServerBindPort* to 18093:

```
<attribute name="ServerBindPort">18093</attribute>
```

18. Save and close the *uil2-service.xml* file.

Deploying BAM Server

You deploy BAM Server to JBoss by copying the CAS_Adobe.ear file to the instance of JBoss that you configured for BAM Server and the CQIntegration.ear file to the instance of JBoss that you configured or LiveCycle Workflow Server. JBoss can be running or stopped when you copy the files to the directories. After you copy the files, you must start or restart the server to ensure that the services start correctly.

► **To deploy BAM Server to JBoss:**

1. Copy the CAS_Adobe.ear file from the `[LiveCycle root]/configurationManager/working/bam/common` directory to the `[jboss bam root]/server/default/deploy` directory.
2. Restart the instance of JBoss that you configured for BAM Server.
3. Copy the CQIntegration.ear file from the `[LiveCycle root]/configurationManager/working/bam/common` directory to the `[appserver root]/server/all/` deploy directory.
4. Restart the instance of JBoss that you configured for LiveCycle Workflow Server.

Next step

You must now configure Business Activity Monitor-related properties on LiveCycle Workflow Server and BAM Server. (See [“Getting Started with BAM Server” on page 90.](#))

Part VI: Additional LiveCycle Workflow Configuration

This section of the guide describes the additional configuration required for LiveCycle Workflow installations.

This chapter describes how to perform the initial configuration of LiveCycle Workflow Server to enable BAM Server to monitor process activity. You must configure settings on both LiveCycle Workflow Server and BAM Server:

- [“Configuring LiveCycle Workflow Server for BAM Server” on page 90](#)
- [“Configuring BAM Server” on page 91](#)

Configuring LiveCycle Workflow Server for BAM Server

You can configure LiveCycle Workflow Server to connect to BAM Server using Administrator. You must specify the server on which BAM Server is running and the user account information with which to access BAM Server.

► **To configure LiveCycle Workflow Server for BAM Server:**

1. Log into Administrator. (See [“Accessing Administrator” on page 25.](#))
2. Click **Services**, and then click **Adobe LiveCycle Workflow**.
3. Click **Server Settings**, and then click **BAM Configuration Settings**.
4. Specify values for the following properties:
 - **Host:** The host name or IP address of the server on which BAM Server is running. The default value is `localhost`.
 - **Port:** The service port of the application server on which BAM Server is running. The default value is the default port of the application server for which LiveCycle Workflow was installed.
 - **User Name:** (Optional) The user name of the administrator user account that LiveCycle Workflow Server uses to access BAM Server. The default user name is `system`.

Note: If you specify the user name for a different user account, you must ensure that the user has complete administrative privileges for BAM Server. For information about administering BAM user accounts, see *BAM Workbench Help* or the *Using LiveCycle Workflow Workbench* guide.

 - **Password:** (Optional) A valid password for the user name specified above. The default password is `manager`.
5. Click **Save** and restart the LiveCycle.ear.

Accessing BAM Workbench and BAM Dashboard

After BAM Server is deployed and running on the application server, you can access BAM Workbench and Business Activity Monitor (BAM) Dashboard through a web browser. BAM Workbench includes tools for administering BAM Server and for setting up reports that users view in BAM Dashboard.

A user name and password are required to log into the Business Activity Monitor pages. When you deploy BAM Server, a default administration account is created with the user name *system* and password *manager*.

Tip: You can create a different user for administration or modify the existing one using BAM Workbench. If you modify the existing one, you must change the corresponding properties on LiveCycle Workflow Server. (See [“Configuring LiveCycle Workflow Server for BAM Server” on page 90.](#))

For information on using the tools provided with the administration framework, log into BAM Workbench and see the Help.

► To access BAM Workbench:

1. Start BAM Workbench by typing `http://[host name]:[port]/celequest/workbench` in the URL line of a web browser.
2. In the **User Name** box, type a valid user name, and in the **Password** box, type the associated password. You can use the default user name *system*, and the default password *manager*.

► To access BAM Dashboard:

1. Start BAM Dashboard by typing `http://[host name]:[port]/celequest/dashboard` in the URL line of a web browser.
2. In the **User Name** box, type a valid user name, and in the **Password** box, type the associated password. You can use the default user name *system*, and the default password *manager*.

Configuring BAM Server

You can configure BAM Server to monitor LiveCycle Workflow processes using BAM Workbench. (See [“Accessing BAM Workbench and BAM Dashboard” on page 91.](#))

You need to use BAM Workbench to perform the following tasks:

- Configure the SMTP server settings for sending email messages.
- Import the metadata definitions used for monitoring LiveCycle Workflow processes.
- Start the JDBC agent.
- Configure the LDAP settings for BAM Server.

Configuring the SMTP settings

BAM Server uses an SMTP server to send email messages to Business Activity Monitor users. You must specify the server on which the SMTP server is running and the user account information with which the BAM Server can access the SMTP server.

► **To configure the SMTP settings:**

1. Ensure that the LiveCycle Workflow Server and the BAM Server applications are started.
2. Ensure that your BAM metadata database is started.
3. Start BAM Workbench by typing `http:// [host name] : [port] /celequest/workbench` in the URL line of a web browser.
4. Log in using the user name and password that is configured for Business Activity Monitor administration. You can use the default user name `system`, and the default password `manager`.
5. If this is your first time logging into BAM Workbench, the First Time Setup dialog box appears. If the dialog box appears, click **Open System Settings**, and then click the **SMTP Configuration** tab.
6. Specify values for the following properties:
 - **SMTP Host:** The network name or IP address of the SMTP server that BAM Server uses to send email messages.
 - **SMTP From Address:** The email address that BAM Server uses to receive email messages.
 - **SMTP User:** The user name for the email account that BAM Server uses.
 - **SMTP Password:** The password that corresponds with the SMTP user.
7. Select the **Checkpoint Configuration** tab, and in the **Recovery Log Directory** box, type the location of the recovery log directory.
8. Select the **Logging** tab, and in the **Logging Directory** box, type the location of the logging directory.
9. Click **OK**.

Importing LiveCycle Workflow metadata definitions

You must import LiveCycle Workflow metadata definitions to configure the process metrics that BAM Server monitors. Importing the LiveCycle Workflow metadata file creates the Adobe JDBC agent.

► **To import LiveCycle Workflow metadata definitions:**

1. Ensure that the LiveCycle Workflow Server and the BAM Server applications are running.
2. Ensure that the BAM metadata database is started.
3. Start BAM Workbench by typing `http:// [host name] : [port] /celequest/workbench` in the URL line of a web browser.
4. Click the **Administration Console** tab, and then click **Import/Export**.
5. Select **Import Metadata from a JAR file (upload)** and type the full path to the location of the LiveCycle Workflow BAM metadata template that is appropriate for the type of database that you are using for the LiveCycle database:
 - (SQL Server) `adobeimport_SQLServer.jar`
 - (MySQL) `adobeimport_MySQL.jar`These files are installed in the `[LiveCycle root]/Workflow/bam/BAMAppTemplate` directory.
6. Click **OK**.

Starting the JDBC agent

You must start the JDBC agent named *adobe*.

► **To start the JDBC agent:**

1. Ensure that the LiveCycle Workflow Server and BAM Server applications are started.
2. Open BAM Workbench in a web browser. (See [“Accessing BAM Workbench and BAM Dashboard” on page 91.](#))
3. Click the **Administration Console** tab and, in the left panel, select **Agents**.
4. In the Agents table, click the icon in the Status column to enable the agent called **adobe**.
5. When prompted, click **Enable All** to enable all dependent objects. The status should change to enabled.

Configuring LDAP settings for BAM Server

In addition to manually creating users and user permissions, Business Activity Monitor lets you to import user information from supported LDAP providers. You can schedule automatic synchronizations or perform manual synchronizations with the LDAP server to automatically update the existing users and roles.

When synchronizing with the LDAP server, the user base DN, login identification and password, full name, description, and email address properties are cached in the BAM metadata database.

When BAM Server imports users from the LDAP server, LDAP groups are converted to Business Activity Monitor roles. Users are assigned roles according to the group they belong to in LDAP. For more information, see [“Limitations of BAM Server LDAP connectivity” on page 93.](#)

Note: BAM Server integrates with any LDAP provider that supports LDAP version 3 protocol.

Limitations of BAM Server LDAP connectivity

The following limitations apply to the BAM Server connectivity with the LDAP server:

- You can configure a connection to only one LDAP server.
- BAM Server creates roles based on groups that are defined on the LDAP server. When BAM Server encounters a group for which a role is not yet created, it creates the role and assigns it a set of zero permissions. You can later modify the permissions as required. (See [“Configuring LDAP role mapping” on page 96.](#))
- If BAM Server imports a user and the user does not belong to a group to which a Business Activity Monitor role corresponds, the user is created but remains unassigned to any roles.
- You cannot change the role that a user is assigned to if the user is imported from the LDAP server. Role assignments for imported users can be accomplished by making changes to the LDAP server. However, you can assign manually-created users to roles that are created based on LDAP groups.

Best practices for BAM Server LDAP connectivity

When setting up the BAM Server connectivity with the LDAP server, it is strongly recommended you adhere to the following best practices:

- If the connection to the LDAP server is not secure, you should use SSL.
- For authentication, Simple Authentication and Security Layer (SASL) is the recommended method and is well supported by LDAP.
- For security reasons, the access permissions of the LDAP synchronization user should be limited to querying the LDAP server. For more information on the synchronization user, see [“Configuring automatic LDAP synchronization” on page 94](#).

Caution: The password for this user is stored in the BAM Server metadata using reversible symmetric encryption. Therefore, anyone with access to the metadata can obtain this password.

Configuring automatic LDAP synchronization

You can schedule the automatic synchronization of BAM Server with the LDAP server. Synchronization ensures that the user accounts and role definitions that BAM Server caches in the BAM metadata database are up to date with the content of the LDAP server.

During synchronization, BAM Server creates new roles based on any new groups in the LDAP server and removes existing roles based on any groups that have been removed from the LDAP server since the previous synchronization.

Note: Roles are removed even if users that were created manually are assigned to the roles. For these users, if the roles are removed due to synchronization, the user accounts still exist but are no longer assigned to the roles.

► To configure automatic LDAP synchronization:

1. Start BAM Workbench by typing `http://[host name]:[port]/cequest/workbench` in the URL line of a web browser.
2. Click the **Administration Console** tab, and then click **System Settings**.
3. Click the **LDAP Synchronization** tab.
4. Select the **LDAP Enabled** option, if it is not already selected.
5. Specify values for the following properties:
 - **Initial Context Factory:** The JNDI name through which BAM Server connects to the LDAP server. The default value is `com.sun.jndi.ldap.LdapCtxFactory`. You probably do not need to change this value.
 - **LDAP Server:** The DNS name or IP address of the LDAP server.
 - **LDAP Port:** The port on which the LDAP server is running. The default port is typically 389. However, if you select the SSL option, the default port is typically 636. You must confirm with your LDAP administrator which port to specify.
 - **LDAP SSL:** Select this option if the LDAP server is configured to use SSL. Selecting this option may affect the LDAP Port setting.

- **LDAP Authentication:** The authentication method used by the LDAP server. Select one of the following options:
 - Simple
 - SASL (Simple Authentication and Security Layer). Select this option for Sun ONE.
 - Compare Encrypted Password
 - **LDAP Principal DN Prefix:** For the simple authentication method, the text you specify will be inserted before the user's login name:
 - For LDAP servers that require DN login, set this to the appropriate property value followed by an equal sign (for example, `cn=` or `uid=`).
 - For ActiveDirectory, leave this value blank.
 - **LDAP Principal DN Suffix:** For the simple authentication method, the text you specify will be inserted after the user's login name:
 - For LDAP servers that require DN login, set this to the appropriate chain of values. The first character of the suffix must be a comma (","). For example:
`,ou=Users,dc=domain,dc=name`
 - For ActiveDirectory, which requires a simple login with an email address, set this to an at symbol (@) followed by the domain name that is set for ActiveDirectory.
 - **LDAP Synchronization User:** The user that binds to the server and reads the lists of users and roles. For security purposes, you must specify a user account that can only read the LDAP directory.
 - **LDAP Synchronization Password:** The password associated with the user specified for the LDAP Synchronization User option.
6. Click **Test Connection**. The connection and the user mapping and role mapping configuration are tested. If the connection settings are correct, a message will indicate that the connection was successful. If you have not yet configured LDAP User Mapping and LDAP Role Mapping, the message returns an error.
 7. To set the LDAP Synchronization Schedule, use the **Add Schedule**, **Edit Schedule**, and **Remove Schedule** buttons to create the desired schedule.

Note: You should set synchronization for a time when the fewest number of users are likely to be logged in.
 8. Click **OK**.

Configuring LDAP user mapping

You can configure the user mapping parameters that determine which users are imported and synchronized from the LDAP server. The parameters you specify depend on the LDAP server provider that you are using.

► To configure the LDAP user mapping parameters:

1. Start BAM Workbench by typing `http:// [host name] : [port] /celequest/workbench` in the URL line of a web browser.
2. Click the **Administration Console** tab, and then click **System Settings**.
3. Click the **LDAP User Mapping** tab.

4. Specify values for the following parameters:
 - **LDAP User Base DN:** The root of the tree that will be searched for users. For example:
 - (Sun ONE) `OU=people,DC=your_domain,DC=com`
 - (Active Directory) `CN=Users,DC=yourdomain,DC=com`
 - **LDAP User Search Filter:** The format that is appropriate for the type of LDAP server you are using. For example, your LDAP server could have a special group for Business Activity Monitor users. This filter could then ensure that only users with this group membership are imported.
 - **LDAP User LoginID:** The login ID of the indicated provider. This value will become the user's login ID in Business Activity Monitor.
 - **LDAP User Full Name:** Enter `cn` if you are using either Sun ONE or Active Directory.
 - **LDAP User Description:** Enter `description` if you are using either Sun ONE or Active Directory.
 - **LDAP User PrimaryEmail:** Enter `userPrincipalName`. This refers to the user's email address in the LDAP directory.
 - **LDAP User EncryptedPassword:** Enter the password associated with the specified user if you are using either Sun ONE or Tivolie. Leave blank for Active Directory.
5. Click the **LDAP Synchronization** tab and click **Test Connection** to see if the users are imported successfully.
6. Click **OK**.

Configuring LDAP role mapping

You can configure the role mapping parameters that determine which groups or roles are imported or synchronized, or both. The parameters you specify depend on the LDAP server provider that you are using.

► To configure the LDAP role mapping parameters:

1. Start BAM Workbench by typing `http://[host_name]:[port]/celequest/workbench` in the URL line of a web browser.
2. Click the **Administration Console** tab, and then click **System Settings**.
3. Click the **LDAP Role Mapping** tab.
4. Specify values for the following parameters:
 - **LDAP Role Base DN:** The format that is appropriate for the type of LDAP server you are using:
 - (Sun ONE) `OU=Groups,DC=your_domain,DC=com`
 - (Active Directory) `CN=Users,DC=yourdomain,DC=com`
 - **LDAP Role Search Filter:** The format that is appropriate for the type of LDAP server you are using:
 - (Sun ONE) `(&(objectclass=groupOfUniqueNames))`
 - (Active Directory) `(&(objectclass=group))`
 - **LDAP Role LoginID:** Enter `cn` if you are using either Sun ONE or Active Directory.
 - **LDAP Role Full Name:** Enter `displayname` if you are using either Sun ONE or Active Directory.
 - **LDAP Role Description:** Enter `description` if you are using either Sun ONE or Active Directory.

- **LDAP Role Member:** The name of the multivalued property that contains role members:
 - (Sun ONE) `uniqueMember`
 - (Active Directory) `member`
 - **LDAP Role Member is:** Use one of the following options:
 - `Distinguished Name`, if the role member properties identify users by distinguished names, such as `cn=jadmin,ou=people,dc=your domain,dc=com`.
 - `Login identification`, if the role member properties identify users by the value of the property used as the Business Activity Monitor login ID.
5. Click the **LDAP Synchronization** tab and click **Test Connection** to see if the users are imported successfully.
 6. Click **OK**.

Manually synchronizing with the LDAP server

You can manually synchronize BAM Server with the LDAP server at any time. Synchronization requests are queued to prevent concurrent synchronizations.

► To manually synchronize with the LDAP server:

1. Start BAM Workbench by typing `http://[host name]:[port]/celequest/workbench` in the URL line of a web browser.
2. Click the **Administration Console** tab, and then click **System Settings**.
3. Click the **LDAP Synchronization** tab.
4. Click **Synchronize Now**. The time required to synchronize depends on your environment. A message appears when the synchronization is complete.
5. Click **OK**.

Next steps

You can now install LiveCycle Workflow Designer. (See [“Installing LiveCycle Workflow Designer” on page 98](#).)

14 Installing LiveCycle Workflow Designer

This chapter describes how to install LiveCycle Workflow Designer.

Installing LiveCycle Workflow Designer

You must run an installation program to install LiveCycle Workflow Designer and to configure a connection to LiveCycle Workflow Server. When you run the installation program, you need to provide the name of the server where LiveCycle Workflow Server is deployed.

Before you install LiveCycle Workflow Designer, you must have J2SDK version 1.4.2_04, or a later release of 1.4.2 installed (version 1.4.2_10 is not supported).

LiveCycle Workflow Designer is installed in the following location by default:

- (Windows) C:\Adobe\LiveCycle\WorkflowDesigner
- (Linux) /opt/adobe/livecycle/workflow_designer

► To install LiveCycle Workflow Designer:

1. At the root level of the installation media, run the installation program by entering the following command:
 - (Windows) adobe_lc_workflow_721_des.exe
 - (Linux) adobe_lc_workflow_721_des.bin
2. On the Welcome screen, click **Next**.
3. Read the license agreement, select **I accept the terms of the license agreement**, and then click **Next**.
4. Accept the default installation directory or click **Browse** and navigate to the directory where you want to install the product, and then click **Next**.

Note: If you type in the name of a directory that does not exist, it will be created for you.

5. Select JBoss as the type of application server that hosts LiveCycle Workflow Server and click **Next**.
6. In the text box, type the name of the computer running LiveCycle Workflow Server, and then click **Next**.
7. Review the installation details, and then click **Back** to change any settings or click **Install** to continue.
8. Click **Finish**.
9. If you are not installing LiveCycle Workflow Designer on the same computer as where LiveCycle Workflow Server is deployed, ensure that the system clock is set to the correct time.

If the system clock is not within two hours of the correct time for the current time zone, LiveCycle Workflow Designer will not be able to connect to the application server. The system clock of the application server must also be set to the correct time.

Connecting to application servers using non-default ports

The default configuration of LiveCycle Workflow Designer uses default JNDI and HTTP ports to connect to the application server. If the application server uses non-default port numbers, you must manually modify the LiveCycle Workflow Designer configuration file.

► **To change the port used for JNDI and HTTP connections:**

1. Open the `qlc-config.xml` file in a text editor from the following location:

- (Windows) `[LiveCycle root]/WorkflowDesigner/qlc/`
- (Linux) `[LiveCycle root]/workflow_designer/qlc/`

2. Locate the `property` element with the `name` attribute of `java.naming.provider.url`, and change the URL in the `value` attribute so that it includes the port that you use for JNDI connections.

For example, the following `property` element configures the port to use for JNDI connections to 1099: `<property name="java.naming.provider.url" value="jnp://localhost:1099/" />`

3. Locate the `application-services` element and change the value of the `port` attribute to the port that your application server uses for HTTP connections.

For example, the following `application-services` element configures the port to use for HTTP connections to 8080: `<application-services host="localhost" port="8080" context="adobe-services" />`

4. Save and close the file.

Uninstalling LiveCycle Workflow Designer

LiveCycle Workflow Designer includes an uninstall program that is created during installation. The uninstall program removes the product files.

► **To remove LiveCycle Workflow Designer:**

1. Start the uninstall program:

- (Windows) Navigate to `[LiveCycle root]/WorkflowDesigner/_uninst/` and double-click **uninstall.exe**.
- (Linux) Navigate to `[LiveCycle root]/workflow_designer/_uninst/` and run `uninstall.bin`.

2. If prompted, select a language for the uninstall program and click **OK**.

3. On the Welcome screen, click **Next**.

4. Review the summary information and click **Uninstall**.

5. If a Remove Existing File dialog box appears, select **Yes To All**.

6. Click **Finish**.

Next steps

You can now deploy QPACs. (See the *Creating Workflows* guide or *LiveCycle Workflow Designer Help*.)

If you have installed Watched Folder and LiveCycle Assembler, you can create workflows that involve Watched Folder using the LiveCycle Assembler QPACs. (See [“Invoking LiveCycle Assembler Using LiveCycle Workflow and Watched Folder” on page 105.](#))

A

Supported Platform and Software Combinations

This appendix provides the supported platforms and software combinations for LiveCycle products. For a summary of the platforms combinations, see [“Before You Install” on page 11](#).

Note: LiveCycle PDF Generator Elements and LiveCycle PDF Generator Professional run only on the Windows Server 2003 Enterprise Edition platform.

Platform	Application server	JDK	Database
Windows Server 2003, Standard Edition	JBoss 3.2.5	J2SDK version 1.4.2_04	MS SQL Server 2000 SP3
	JBoss 3.2.5	J2SDK version 1.4.2_04	MySQL 4.1
Windows Server 2003, Enterprise Edition	JBoss 3.2.5	J2SDK version 1.4.2_04	IBM DB2 8.2 (Version 8.1 Fix Pack 7)
	JBoss 3.2.5	J2SDK version 1.4.2_04	Oracle 9i
	JBoss 3.2.5	J2SDK version 1.4.2_04	Oracle 10g
	IBM WebSphere® 5.1.1.5	IBM JDK installed with WebSphere	Oracle 9i
	BEA WebLogic Server® 8.1 SP5	J2SDK version 1.4.2_08	Oracle 9i
	BEA WebLogic Server 8.1 SP5	J2SDK version 1.4.2_08	Oracle 10g
	IBM WebSphere 5.1.1.5	IBM JDK installed with WebSphere	Oracle 10g
	IBM WebSphere 5.1.1.5	IBM JDK installed with WebSphere	IBM DB2 8.2 (Version 8.1 Fix Pack 7)
	IBM WebSphere 5.1.1.5	IBM JDK installed with WebSphere	MS SQL Server 2000 SP3

Platform	Application server	JDK	Database
Red Hat Linux Advanced Server 2.1 Update 3	JBoss 3.2.5	J2SDK version 1.4.2_04	MS SQL Server 2000 SP3
	JBoss 3.2.5	J2SDK version 1.4.2_04	MySQL 4.1
	JBoss 3.2.5	J2SDK version 1.4.2_04	Oracle 9i
	JBoss 3.2.5	J2SDK version 1.4.2_04	Oracle 10g
	JBoss 3.2.5	J2SDK version 1.4.2_04	IBM DB2 8.2 (Version 8.1 Fix Pack 7)
	BEA WebLogic Server 8.1 SP5	J2SDK version 1.4.2_08	Oracle 9i
	BEA WebLogic Server 8.1 SP5	J2SDK version 1.4.2_08	Oracle 10g
	IBM WebSphere 5.1.1.5	IBM JDK installed with WebSphere	IBM DB2 8.2 (Version 8.1 Fix Pack 7)
	IBM WebSphere 5.1.1.5	IBM JDK installed with WebSphere	Oracle 9i
	IBM WebSphere 5.1.1.5	IBM JDK installed with WebSphere	Oracle 10g
Red Hat Linux Advanced Server 3.0	JBoss 3.2.5	J2SDK version 1.4.2_04	MySQL 4.1
	JBoss 3.2.5	J2SDK version 1.4.2_04	Oracle 9i
	JBoss 3.2.5	J2SDK version 1.4.2_04	Oracle 10g
	JBoss 3.2.5	J2SDK version 1.4.2_04	IBM DB2 8.2 (Version 8.1 Fix Pack 7)
	BEA WebLogic Server 8.1 SP5	J2SDK version 1.4.2_08	Oracle 9i
	BEA WebLogic Server 8.1 SP5	J2SDK version 1.4.2_08	Oracle 10g
	IBM WebSphere 5.1.1.5	IBM JDK installed with WebSphere	IBM DB2 8.2 (Version 8.1 Fix Pack 7)
	IBM WebSphere 5.1.1.5	IBM JDK installed with WebSphere	Oracle 9i
	IBM WebSphere 5.1.1.5	IBM JDK installed with WebSphere	Oracle 10g
IBM AIX® 5.2	IBM WebSphere 5.1.1.5	IBM JDK installed with WebSphere	Oracle 9i
	IBM WebSphere 5.1.1.5	IBM JDK installed with WebSphere	Oracle 10g
	IBM WebSphere 5.1.1.5	IBM JDK installed with WebSphere	IBM DB2 8.2 (Version 8.1 Fix Pack 7)

Platform	Application server	JDK	Database
IBM AIX 5.3	IBM WebSphere 5.1.1.5	IBM JDK installed with WebSphere	Oracle 9i
	IBM WebSphere 5.1.1.5	IBM JDK installed with WebSphere	Oracle 10g
	IBM WebSphere 5.1.1.5	IBM JDK installed with WebSphere	IBM DB2 8.2 (Version 8.1 Fix Pack 7)
Sun™ Solaris™ 8	BEA WebLogic Server 8.1 SP5	J2SDK version 1.4.2_08	Oracle 9i
	BEA WebLogic Server 8.1 SP5	J2SDK version 1.4.2_08	Oracle 10g
	IBM WebSphere 5.1.1.5	IBM JDK installed with WebSphere	IBM DB2 8.2 (Version 8.1 Fix Pack 7)
	IBM WebSphere 5.1.1.5	IBM JDK installed with WebSphere	Oracle 9i
	IBM WebSphere 5.1.1.5	IBM JDK installed with WebSphere	Oracle 10g
Sun Solaris 9	BEA WebLogic Server 8.1 SP5	J2SDK version 1.4.2_08	Oracle 9i
	IBM WebSphere 5.1.1.5	IBM JDK installed with WebSphere	Oracle 9i
	IBM WebSphere 5.1.1.5	IBM JDK installed with WebSphere	Oracle 10g
SUSE Linux ES 9.0	JBoss 3.2.5	J2SDK version 1.4.2_04	MS SQL Server 2000 SP3
	JBoss 3.2.5	J2SDK version 1.4.2_04	Oracle 9i
	JBoss 3.2.5	J2SDK version 1.4.2_04	Oracle 10g
	JBoss 3.2.5	J2SDK version 1.4.2_04	MySQL 4.1
	BEA WebLogic Server 8.1 SP5	J2SDK version 1.4.2_08	Oracle 9i
	IBM WebSphere 5.1.1.5	IBM JDK installed with WebSphere	Oracle 9i
	IBM WebSphere 5.1.1.5	IBM JDK installed with WebSphere	Oracle 10g

B

Fonts Installed with the Font Manager Module

Some modules require access to fonts that are installed with the Font Manager Module. The Font Manager Module contains a number of bundled fonts that you can use in your custom applications.

The Font Manager Module is not used when converting native documents into PDF documents or when converting PDF documents into some other format.

You can add fonts to the Font Manager Module, which is subsequently packaged in the LiveCycle.ear file. For information about adding your own fonts, see [“Configuring LiveCycle Products for Deployment” on page 48.](#)

The following fonts are installed with the Font Manager Module:

- Adobe Serif MM
- Adobe Sans MM
- Adobe Ming Std Light
- Adobe Song Std Light
- Adobe Myungjo Std Medium
- Adobe Pi Std
- Bell Gothic Std Light
- Bell Gothic Std Bold
- Bell Gothic Std Black
- Courier Std
- Courier Std Bold
- Courier Std Bold Oblique
- Courier Std Oblique
- EuroSign (TTF)
- Kozuka Gothic® Std Extra Light
- Kozuka Gothic Std Light
- Kozuka Gothic Std Regular
- Kozuka Gothic Std Medium
- Kozuka Gothic Std Bold
- Kozuka Gothic Std Heavy
- Kozuka Gothic Pro Medium
- Kozuka Mincho® Std Extra Light
- Kozuka Mincho Std Light
- Kozuka Mincho Std Regular
- Kozuka Mincho Std Medium
- Kozuka Mincho Std Bold
- Kozuka Mincho Std Heavy
- Kozuka Mincho Pro-VI Regular
- Letter Gothic Std Medium
- Letter Gothic Std Slanted
- Letter Gothic Std Bold
- Letter Gothic Std Bold Slanted
- Minion® Pro Italic
- Minion Pro Semibold
- Minion Pro Semibold Italic
- Minion Pro Bold
- Minion Pro Bold Italic
- Minion Pro Regular
- Myriad® Pro Light
- Myriad Pro Light Italic
- Myriad Pro Regular
- Myriad Pro Italic
- Myriad Pro Bold
- Myriad Pro Semibold
- Myriad Pro Semibold Italic
- Myriad Pro Bold Italic
- Myriad Pro Black
- Myriad Pro Black Italic
- Symbol

C

Invoking LiveCycle Assembler Using LiveCycle Workflow and Watched Folder

This appendix describes how to invoke LiveCycle Assembler through LiveCycle Workflow, using the LiveCycle Assembler QPACs and Watched Folder.

This appendix assumes that you have installed, configured, and deployed LiveCycle Assembler, LiveCycle Workflow, and Watched Folder.

Note: Watched folders can be implemented for use with LiveCycle PDF Generator, but need to be configured separately. The contents of this chapter refer only to the Watched Folder service that integrates LiveCycle Workflow and LiveCycle Assembler. For information on setting up watched folders for LiveCycle PDF Generator, see *LiveCycle PDF Generator Administration Help*, available from the LiveCycle PDF Generator web interface.

Summary of tasks

Using either Watched Folder or a client application you developed, you can direct LiveCycle Workflow to initiate a workflow that activates a LiveCycle Assembler QPAC.

This section outlines the tasks required to set up a workflow that completes a LiveCycle Assembler process that is invoked by Watched Folder. The detailed steps required to complete the tasks are located in the section or document indicated. You must perform the tasks in the order that they are listed in the table.

Note: Do not run Configuration Manager until you have completed installing the files for each of the products.

Step	Task	See
1	Deploy LiveCycle Assembler QPACs to LiveCycle Workflow Designer.	<i>Using LiveCycle Assembler QPACs</i> (available on the LiveCycle Assembler installation DVD or on the LiveCycle Workflow SDK website)
2	Create and deploy a workflow that includes a LiveCycle Assembler QPAC.	"Deploying QPACs and creating a workflow process" on page 106 <i>LiveCycle Workflow Creating Workflows</i> <i>LiveCycle Workflow Designer Help</i>
3	Create a watched folder on the file system.	"Creating and configuring a watched folder" on page 109
4	Log into Administrator and create and configure a watched folder.	"Creating and configuring a watched folder" on page 109 <i>Watched Folder Administration Help</i>
5	Create a JobConfig.xml file.	"Creating a JobConfig.xml file" on page 110 <i>Watched Folder Administration Help</i>

Step	Task	See
6	Prepare the test or production collateral, which includes input PDF files and a DDX file.	“Preparing PDF and DDX files” on page 112 <i>Developing Applications for LiveCycle Assembler</i> <i>Document Description XML Reference</i>
7	Submit the LiveCycle Assembler job for processing.	“Submitting the LiveCycle Assembler job for processing” on page 113 <i>Developing Custom Applications for LiveCycle Workflow</i>

Deploying QPACs and creating a workflow process

There are two QPACs used to create workflows for LiveCycle Assembler: the Assembler QPAC and the CreateDocumentList QPAC. The Assembler QPAC is used for all LiveCycle Assembler workflows, whether you are using Watched Folder or another method of initiating the process (for example, client applications or web services).

This section describes how to create a workflow process that uses the Assembler QPAC, including how to create the required variables and action properties in LiveCycle Workflow Designer. For information on creating a workflow process that uses the CreateDocumentList QPAC, see the *Using LiveCycle Assembler QPACs* guide.

The steps for creating a workflow process that includes an Assembler QPAC are similar to those for creating any workflow process using LiveCycle Workflow Designer. The procedure here provides details specific to the Assembler QPAC. For more information on creating workflows, setting up variables, and setting action properties in a QPAC, see the *Creating Workflows* and *Using LiveCycle Assembler QPACs* guides.

Using dynamic or static DDX files

When you submit jobs to Watched Folder for processing by LiveCycle Workflow and LiveCycle Assembler, you can use a dynamic or a static DDX file. When you create or use an application to submit jobs to LiveCycle Assembler that uses only the QPAC workflow (without Watched Folder), you must use a static DDX file.

A workflow that uses a dynamic DDX file processes the DDX file that is included in the input folder in a watched folder. To use a dynamic DDX file in your workflow process, you must create an `<inputddx>` variable and specify it in the action properties of the QPAC.

A workflow that uses a static DDX file processes a particular DDX file that is loaded into the QPAC and used every time that the process is called by the JobConfig.xml file. To use a static DDX file in your workflow process, you must browse to the DDX file and load it into the QPAC when you set the action properties for the workflow. You can use a static DDX if you are using a watched folder, or if you are running the process using only the QPACs.

Configuring an Assembler QPAC in a workflow process

When you create a new workflow process in LiveCycle Workflow Designer, you must create variables that will be used by the workflow process. When you add the Assembler QPAC to the workflow, you must set the action properties for the QPAC. The action properties use the variables that you created.

The selections you make on the Action Properties dialog box depend on whether you are using a dynamic or static DDX file.

Note: The variable names used in this chapter, such as `<inputddx>` and `<inputdocmap>`, are user-defined in LiveCycle Workflow Designer; you can use any name when you create the variable.

Variables for the Assembler QPAC

Variable	Property values
<code><inputdocmap></code>	Type: map of document in: selected out: not selected required: not selected
<code><inputddx></code> (required for dynamic DDX file only)	Type: document in: selected out: not selected required: not selected
<code><outputmap></code>	Type: map and document in: not selected out: selected required: not selected
<code><joblog></code> (not required when Logging Level is set to Off)	Type: Document in: not selected out: selected required: not selected

Action properties for the Assembler QPAC

Type of workflow	Select on Input tab	Select on Output tab
Using Watched Folder with a dynamic DDX file	Use a Document Variable to load the DDX file: Click the ellipsis button and select the <code><inputddx></code> variable. Input Document Map Variable: Select the <code><inputdocmap></code> variable.	Output Document Map: Click the ellipsis button and select the <code><outputmap></code> variable.
Using Watched Folder with a static DDX file	Use the Browse button to locate the DDX file: Browse to the DDX file that you want to use. Input Document Map Variable: Select the <code><inputdocmap></code> variable.	Output Document Map: Click the ellipsis button and select the <code><outputmap></code> variable.

Type of workflow	Select on Input tab	Select on Output tab
Using QPAC only (with a static DDX file)	Input Document Map: Do not select Use the Browse button to locate the DDX file: Browse to the DDX file that you want to use.	Output Document Map: Click the ellipsis button and select the <outputmap> variable.

Note: For information about selecting options on the Configuration tab, see the next section, [Error logging and fail modes in the Assembler QPAC workflows.](#)

Error logging and fail modes in the Assembler QPAC workflows

When you set up a workflow using the Assembler QPAC, you can select the error logging level and the fail mode that LiveCycle Workflow uses when exceptions occur.

You can set the error logging level to Off (no logging), Normal (brief messages), or Debug (detailed descriptions of errors). You can associate a process variable (for example, <joblog>) of type Document to the Output Log Document field in order to store the log messages.

You can set the fail mode to Stall Process or Terminate Process to direct the Assembler QPAC to stall or terminate a process when an exception occurs in LiveCycle Assembler.

When a job fails with Stall Process selected, the workflow process stalls, but is not terminated. If logging is set to Normal or Debug, information about the exception appears on the LiveCycle Workflow Stalled Action page in Administrator. Component administrators should search for the process and terminate it, and then return to Administrator and delete the stalled action if desired. The logged messages are displayed on the stalled action page, but not on the process page.

When a job fails with Terminate Process selected, the workflow process terminates when an exception is thrown. If logging is set to Normal or Debug, information about the exception is written to the log file for the component for which the exception was thrown. For Watched Folder, the log file is the activity.log file that is placed in the /failure subdirectory of the configured watched folder.

If you are configuring an Assembler QPAC for use with Watched Folder, you should set the fail mode to Terminate Process. Watched Folder users may not have access to Administrator, so if processes stall but are not terminated, users do not receive messages about the exception until the process is terminated. Setting the fail mode to Terminate Process ensures that error messages are logged to the activity.log file, which is available in the [watchedfolder]/failure directory (when the logging level is set to Normal or Debug).

For information on setting error logging and fail modes, see *Using LiveCycle Assembler QPACs*.

► To create a workflow that includes an Assembler QPAC:

1. Deploy the Assembler QPAC to LiveCycle Workflow Designer. (See *Using LiveCycle Assembler QPACs*.)
2. In LiveCycle Workflow Designer, select **File > New > Process Category**.
3. Select the new process category and select **File > New > Process Type**.
4. Right-click the new process type and select **New Workflow**.
5. Name the workflow and use the default values for the remaining options (synchronous is selected and the other options are unchecked).

6. Create the variables for the workflow, as described in the [“Variables for the Assembler QPAC” on page 107](#) table, by clicking the green plus (+) button on the Variables palette. For more information on creating variables, see *LiveCycle Workflow Designer Help*.
7. Create a new component category. (See *Creating Workflows*.)
8. Add the Assembler QPAC (deploy the component) to the new component category, and set the action properties for the workflow, as described in the [“Action properties for the Assembler QPAC” on page 107](#) table.
9. Save and deploy the workflow process.

Creating and configuring a watched folder

You must create a directory on your file system that you can configure as a watched folder. After you have configured the watched folder using the Watched Folder configuration interface, the watched folder contains the following subdirectories:

```
/[watchedfolder]
  /input
  /stage
  /failure
  /result
  /preserve
```

For information on the subdirectories, see *Watched Folder Administration Help*.

To submit files for processing, you place them in the `/[watched folder]/input` directory.

When multiple files need to be included in the LiveCycle Assembler process, do not transfer the individual files to the input directory; instead, create a separate collection directory, place all of the required files in it (that is, the input PDF files and, if required, the DDX file), and then place the collection directory in the input directory of the watched folder.

► To create and configure a watched folder:

1. Create a directory to be a watched folder on your file system on your server.
2. Log into Administrator.
3. Click **Services**, click **Watched Folder**, and then click **Watched Folder Management and Status**.
4. Click **New Watched Folder**.
5. In the **Watched Folder Name** box, type a name for the watched folder. (This does not have to match the name of the directory you created on the file system in step [1](#), but it is recommended for tracking purposes.)
6. In the **Watched Folder Path** box, type the full path to the directory you created on the file system.
7. In the **Include File Pattern(s)** box, type the name of the directory you created to contain the input PDF files (that you will place in the input directory of the watched folder). You can specify more than one file or name, if you intend to place multiple files and folders in the watched folder.

8. Set the other options according to your requirements. For information on the options available, see *Watched Folder Administration Help*.
9. Click **OK**.

Tip: If you are transferring a large number of source files to the watched folder, it may take several minutes or longer. It is possible that Watched Folder might scan the watched folder, recognize a new collection directory, and initiate a workflow process before all of the files in the directory have been copied over. To avoid this, you must name the collection directory something other than the name that Watched Folder is scanning for while you copy the directory to the input directory. When all of the input files have been copied over, you must rename the collection directory to the name specified in the Watched Folder configuration (the **Include File Pattern(s)** value).

Creating a JobConfig.xml file

The JobConfig.xml file can be placed at the root of the input directory or inside the collection directory together with the other collateral files.

If you plan to place multiple jobs (collection directories) in the watched folder for different workflows that use dynamic DDX files, the JobConfig.xml file must be placed in the individual collection directories that are submitted to the input directory in the watched folder.

If you plan to submit multiple jobs for the same workflow that uses a static DDX file, you can place the JobConfig.xml at the root of the input directory. Subsequent incoming jobs use this same JobConfig.xml file.

To create a JobConfig.xml file, create an XML file similar to the following examples.

Example C.1 A JobConfig.xml file for a dynamic DDX file

```
<?xml version="1.0" encoding="UTF-8"?>
<process-config xmlns="http://adobe.com/watchedfolder"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://adobe.com/watchedfolder
C:\WatchedFolder\schemas\JobConfiguration.xsd">
  <process-type>myworkflow</process-type>
  <process-variables>
    <process-variable>
      <filter-pattern>*.pdf</filter-pattern>
      <process-variable-type>map</process-variable-type>
      <process-variable-name>inputdocmap</process-variable-name>
      <process-variable-datatype>document</process-variable-datatype>
      <process-input>>true</process-input>
    </process-variable>
    <process-variable>
      <filter-pattern>*.ddx</filter-pattern>
      <process-variable-type>single</process-variable-type>
      <process-variable-name>inputddx</process-variable-name>
      <process-variable-datatype>document</process-variable-datatype>
      <process-input>>true</process-input>
    </process-variable>
  </process-variables>
</process-config>
```

Example C.2 A JobConfig.xml file for a static DDX file

Use the same JobConfig.xml described previously, but omit the <inputddx> process variable.

```
<?xml version="1.0" encoding="UTF-8"?>
<process-config xmlns="http://adobe.com/watchedfolder"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://adobe.com/watchedfolder
C:\WatchedFolder\schemas\JobConfiguration.xsd">
  <process-type>myworkflow</process-type>
  <process-variables>
    <process-variable>
      <filter-pattern>*.pdf</filter-pattern>
      <process-variable-type>map</process-variable-type>
      <process-variable-name>inputdocmap</process-variable-name>
      <process-variable-datatype>document</process-variable-datatype>
      <process-input>>true</process-input>
    </process-variable>
  </process-variables>
</process-config>
```

The content of the JobConfig.xml file maps to the variables and actions defined in the workflow process described in [“Configuring an Assembler QPAC in a workflow process” on page 107](#).

Node	Description
process-config	The root of the job configuration file.
process-type	The name of the workflow process type. This value must match the value specified in the workflow process in LiveCycle Workflow Designer.
process-variables	Contains multiple process-variable nodes.
process-variable	Each process-variable node maps to a variable included in the workflow process. Its subelements specify property values for each process variable.
filter-pattern	The pattern used to filter all of the files from the input set that will be set for the process variable. This pattern matches the type of files to be processed in relation to the variable.
process-variable-type	The type of input variable: single, list, or map. In the example above, <code>list</code> is used for the <inputdocmap> variable, because the variable specified a series of documents (all of the files that will be placed in the watched folder). For the <inputddx> variable, <code>single</code> is specified, because only one file will be included.

Node	Description
<code>process-variable-name</code>	The name of the workflow variable as defined in the workflow. In the example above, the variable names match those created in “Configuring an Assembler QPAC in a workflow process” on page 107 : <code>inputdocmap</code> and <code>inputddx</code>
<code>process-variable-datatype</code>	The type of workflow variable. Two types of variables are supported: <code>document</code> and <code>xml</code> . This matches the <code>containedtype</code> attribute of the variable (not the <code>collectiontype</code> attribute).
<code>process-input</code>	Watched Folder does not currently use this setting. Leave the value set to <code>true</code> .

For more information on creating the `JobConfig.xml` file, see *Watched Folder Administration Help*.

Preparing PDF and DDX files

The PDF files included in the input directory or defined by the Assembler QPAC must correspond to the file pattern value defined for the `source` element in the DDX file.

When using Watched Folder with LiveCycle Assembler, you must specify the Input Document Map variable (as described in [“Configuring an Assembler QPAC in a workflow process” on page 107](#)). You must use the `sourceMatch` attribute for the `source` element to define the file pattern. For information on setting the regular expression pattern for the `sourceMatch` attribute, see the *Document Description XML Reference*.

For example, in the DDX file, the PDF source can be specified in the following ways:

- Pattern matching for specific file names:
`<PDF source="mysource" sourceMatch="appendix[\d]+[.]pdf" select="1-last"/>`
- Pattern matching with wildcard value:
`<PDF source="mysource" sourceMatch=".[.]pdf" select="1-last"/>`

In the examples above, pattern-matching matches any input PDF files that begin with the characters `appendix` followed by a digit, and then followed with the file name extension `.pdf`. The wildcard value matches at least one input PDF file with any characters that include digits and end with `.pdf`.

For more information on setting the `source` element in the DDX file, see “Specifying multiple input streams” in the *Developing Applications for LiveCycle Assembler* guide and the *Document Description XML Reference*.

Submitting the LiveCycle Assembler job for processing

You can submit a LiveCycle Assembler job to Watched Folder programmatically using a batch file that transfers the package containing the required files to the configured watched folder. (Instructions for creating a batch file or script to complete this action is beyond the scope of this guide.) You can also submit a job manually by dragging or copying the package of files to Watched Folder.

If you have created a LiveCycle Assembler workflow in LiveCycle Workflow Designer that does not use Watched Folder, you can submit the job for processing using a client application that you create using the LiveCycle Workflow Java API. For information on developing custom client applications for LiveCycle Workflow, see the *Developing Custom Applications for LiveCycle Workflow* guide, available with the LiveCycle Workflow SDK.

You can also submit jobs for processing using a variety of other mechanisms available with LiveCycle Workflow, such as web services, email, and messaging queues, as well by using online forms. For more information, see the *Developing Custom Applications for LiveCycle Workflow* and *Developing Custom QPACs* guides.

Note: When NFS sharing is set up for Watched Folder, temporary files and directories related to Watched Folder jobs are placed in the Global storage directory, but are not automatically deleted when the job processing is complete. To prevent performance issues, you should regularly remove files with name Session *[NNNN]*, where *NNNN* is the workflow process ID from the Global storage directory.

The default location of the Global storage directory is *{TempDir}/AdobeDocumentStorage/global*. If *[TempDir]* is not specified by the user in Configuration Manager, the default location is `java.io.tmpdir`.

D

Developing Forms for LiveCycle

This appendix provides information that is useful for form developers, including how to use WebDAV with LiveCycle Form Manager and which items LiveCycle products require for different types of forms.

Publishing files from client software

Form developers and form owners can use WebDAV-enabled client software to publish and access their files in the LiveCycle products repository used by LiveCycle Form Manager.

To use WebDAV, your computer must be running file management software (such as Microsoft Windows Explorer) that can connect to WebDAV servers.

To create a connection to a WebDAV directory, you must specify the URL of the directory. The following URL is the default URL of the LiveCycle Form Manager repository:

```
http:// [host name] : [port] /appstore/Forms
```

where *[host name]* is the name of the computer that is running LiveCycle products and *[port]* is the port being used for LiveCycle products.

For example, the form developers in your company use LiveCycle Designer to create forms. The form developers want to use the Publish menu command in LiveCycle Designer to copy forms and their supporting files to the LiveCycle products repository. However, they first use Windows Explorer to add a network place, which creates a connection to the URL of the LiveCycle products repository. The connection appears as a folder in My Network Places. After adding the network place, the form developers can use the folder to publish files to the repository.

Designing forms for LiveCycle products

To use forms with LiveCycle products, there may be requirements for the design of the form, the configuration of LiveCycle products, or both. The following types of forms have special requirements for use with LiveCycle products:

- Dynamic forms
- PDF/A-compliant forms
- Forms that include subform objects
- PDF forms created with Acrobat

For information on designing forms for use with LiveCycle Workflow, see the *Creating Workflows* guide, which is installed with LiveCycle Workflow.

Using dynamic forms with LiveCycle products

Dynamic forms can expand or shrink to accommodate the amount of incoming data. Dynamic forms typically include buttons that request form data from LiveCycle Forms. LiveCycle Forms renders the form with the data and returns it to the client. The client's browser is refreshed with the form, which has been expanded or contracted, depending on the amount of new data.

For example, some forms enable users to query a database by specifying search criteria. The returned data appears on the same form (a subform) in a table. The returned data determines the number of rows in the table. To submit the query, the user clicks a button, which sends the request to LiveCycle Forms.

LiveCycle products support the use of the following types of buttons for dynamic forms:

- One that updates the data on a dynamic form
- One that returns a PDF document to a client, who can then save a local copy of the document or print the document

LiveCycle products use specific button names to recognize these buttons. Dynamic forms must use these button names. You can use the default names, or you can go to the Administrator pages and configure LiveCycle products to use different names.

Property	Default	Description and options
Re-render Button name	RENDERAGAIN	The button name that indicates to LiveCycle products that this button sends a request to LiveCycle Forms to render the current form again with updated data.
Print Button Name	PRINTFORM	The button name that indicates to LiveCycle products that this button creates a PDF document from the current form and returns it to the end user.

Embedding fonts in PDF/A-compliant forms

When creating a form design that is intended for the creation of a PDF/A Level 1B-compliant form when rendered by LiveCycle Forms, all of the fonts used must be embedded in the form design. If the fonts are not embedded, the rendered form is not PDF/A Level 1B-compliant, although it is a valid PDF form.

Only fonts licensed for a minimum of View and Print usage can be embedded in the form design. The fonts included in the Font Manager Module meet this criteria.

This chapter describes how to uninstall the LiveCycle product files that were installed during a manual or turnkey installation.

Removing the product files from a turnkey installation

Each LiveCycle product that installed using the turnkey installation includes an uninstall program that you can use to remove the product from your computer. The uninstall program only removes the specified LiveCycle product. It does not remove any files that you deployed to the application server or modules that are shared with other LiveCycle products.

Removing the product components does not remove JBoss and MySQL. You must manually remove JBoss and MySQL. When you uninstall the services, you can still start JBoss and MySQL from the command line.

Caution: Before removing MySQL, back up any data that you do not want to lose.

► To remove the product files:

1. Navigate to the `[product root]/_uninst/` directory and double-click the `[product name]_uninstall.exe` file. Alternatively, you can use the Add or Remove Programs function in the Windows Control Panel to start the uninstall program.
2. If prompted, select a language for the uninstall program and click **OK**.
3. Follow the on-screen instructions, and then click **Finish**.

► To remove the JBoss and MySQL services:

1. Ensure that the JBoss and MySQL services are stopped.

To stop the JBoss for Adobe LiveCycle service, from a command prompt, type:

```
net stop "JBoss for Adobe LiveCycle"
```

To stop the MySQL for Adobe LiveCycle service, from a command prompt, type:

```
net stop "MySQL for Adobe LiveCycle"
```

2. Manually uninstall the JBoss for Adobe LiveCycle service by navigating to the `[LiveCycle root]/jboss/` directory and, from a command prompt, typing:

```
JBossService.exe -uninstall "JBoss for Adobe LiveCycle"
```

3. (LiveCycle Workflow Business Activity Monitor) Manually uninstall the JBoss for Adobe LiveCycle Workflow BAM service by navigating to the `[LiveCycle root]/jboss/` directory and, from a command prompt, typing:

```
JBossService.exe -uninstall "JBoss for Adobe LiveCycle Workflow BAM"
```

4. Manually uninstall the MySQL for Adobe LiveCycle service by navigating to the `[LiveCycle root]/jboss/` directory and, from a command prompt, typing:

```
JBossService.exe -uninstall "MySQL for Adobe LiveCycle"
```

Note: If you are uninstalling multiple LiveCycle products and you installed either LiveCycle Assembler or LiveCycle PDF Generator last, there are two service executables: `JavaService.exe` and `JBossService.exe`.

Removing the product files from a manual installation

The uninstall program located in the `[product root]` directory does not remove any files that you deployed to your application server or modules that are shared with other LiveCycle products.

Caution: By running the uninstall program, all of the contents within the product installation directory are subject to removal without further warning. Before proceeding, back up any data you do not want to lose.

► To remove the files from your computer:

1. Navigate to the `[product root]/_uninst/` directory and start the uninstall program:
 - (Windows) Double-click the `.exe` file. Alternatively, you can use the Add or Remove Programs function.
 - (Linux) From a command prompt, type: `file_name.bin`
2. If prompted, select a language for the uninstall program and click **OK**.
3. Follow the on-screen instructions, and then click **Finish**.

Upgrading LiveCycle Products to Version 7.2 or 7.2.1

This chapter describes the tasks required to upgrade your current LiveCycle products to version 7.2, or to version 7.2.1 for LiveCycle Assembler and LiveCycle Workflow.

This document should be used in conjunction with the *Installing and Configuring LiveCycle* guide or the *Installing and Configuring LiveCycle Security Products* guide for your application server. Throughout this document, specific sections in these installing and configuring guides are listed when more detailed information is available.

For a complete list of the supported platforms and system requirements for LiveCycle 7.2 products, see the “Before You Install” chapter in this guide.

This chapter uses the following naming conventions for common file paths:

[LiveCycle root] Refers to the location where LiveCycle products and components earlier than versions 7.2 or 7.2.1 are installed.

[LiveCycle72 root] Refers to the location where LiveCycle products and components with versions 7.2 and 7.2.1 are installed.

The *Installing and Configuring LiveCycle* guides apply to the following products:

- LiveCycle Assembler 7.2.1
- Adobe LiveCycle Forms 7.2
- Adobe LiveCycle Form Manager 7.2
- Adobe LiveCycle PDF Generator 7.2
- Adobe LiveCycle Print 7.2
- Adobe LiveCycle Workflow 7.2.1
- Adobe LiveCycle Workflow Designer 7.2.1
- Watched Folder 1.1

The *Installing and Configuring LiveCycle Security Products* guides apply to the following products:

- Adobe LiveCycle Document Security 7.2
- Adobe LiveCycle Reader Extensions 7.2
- Adobe LiveCycle Policy Server 7.2

Upgrade guidelines

This section describes guidelines you must follow when upgrading LiveCycle products to version 7.2 or version 7.2.1.

Updating your application server

Ensure that you apply the required patches and fix packs to the application server on which you are running LiveCycle products, and obtain the updated database drivers. Your environment must meet the system requirements described in the “Support software” chapter in this guide.

Using automatic or turnkey installations for upgrading

For installations for deployment to WebSphere or WebLogic, you can use Configuration Manager to configure and deploy the product, initialize the database, and verify the deployment.

Before using Configuration Manager to automatically configure LiveCycle products, the existing LiveCycle components must be undeployed and the application server restarted.

For installations for deployment to JBoss, you can configure the products for deployment and initialize the database using Configuration Manager, but you must deploy the product components manually. The turnkey installation option is only supported for upgrading LiveCycle Reader Extensions.

If you are running LiveCycle products that were originally installed using the turnkey installation option, you can upgrade to version 7.2 or 7.2.1 by following the manual or the auto-configuration upgrade instructions that are included in this guide. Instructions for upgrading LiveCycle Reader Extensions using the turnkey installation option are also included in this chapter.

Configuring using Configuration Manager during the upgrade process

It is recommended that you do not automatically configure the WebSphere or WebLogic application server using Configuration Manager because current configuration settings on your application server may be overwritten.

During the upgrade process, you will configure some product and application server run-time properties using Configuration Manager. For upgrades from 7.x versions, you should configure the upgraded product using the same property values. The “Configuring LiveCycle Products for Deployment” chapter in the *Installing and Configuring* guides provides information about the properties that you will configure when you upgrade the product.

Summary of manual upgrade process

This checklist describes the high-level tasks that you must perform to upgrade from a LiveCycle 7.x product to a LiveCycle 7.2 or 7.2.1 product. For detailed information, see the specific upgrade procedures for the product you are upgrading.

The sections referenced in the “See” column are in the *Installing and Configuring* guides for your application server, depending on the product you are installing.

Task	See
Back up the database that contains the current LiveCycle configuration and run-time data.	The database server documentation.
Back up copies of the currently deployed LiveCycle EAR and WAR files to a separate directory.	
(JBoss) Stop services, if applicable (JBoss for Adobe LiveCycle, MySQL for Adobe LiveCycle, JBoss for Workflow BAM, AdobeDocumentSecurity, or AdobeReaderExtensions).	

Task	See
Undeploy LiveCycle components from the application server.	Refer to the application server documentation.
Uninstall the previous version of LiveCycle products using the uninstaller program.	The “Uninstalling LiveCycle Products” chapter
Apply the required patches and fix packs to the application server and obtain the updated database drivers.	The “Supported software” section of the “Before You Install” chapter
Install the LiveCycle 7.2 or 7.2.1 product to a new (non-default) location on your file system. Do not install to the same directory where your previous LiveCycle products are installed.	The “Installing LiveCycle Products” chapter
Run Configuration Manager to configure the product. Apply the configuration data that you used in the original installation. For WebSphere and WebLogic, you can automatically deploy the product, initialize the database, and verify the deployment using Configuration Manager.	(JBoss) The “Configuring LiveCycle Products for Deployment” chapter (WebLogic, WebSphere) The “Automatically Configuring LiveCycle Products” or “Configuring LiveCycle Products for Deployment” chapter
(JBoss) Deploy the product components to the application server.	The “Manually deploying to JBoss” chapter
(JBoss) Run Configuration Manager to initialize the database.	The “Initializing the Database” chapter

LiveCycle Forms and LiveCycle Print

This section provides upgrading instructions for LiveCycle Forms 7.2 and LiveCycle Print 7.2.

Note: You must install LiveCycle Forms 7.2 before installing LiveCycle Print 7.2.

Upgrading from Adobe Form Server 6.0 to LiveCycle Forms 7.2

This procedure describes how to upgrade from Adobe Form Server 6.0 to LiveCycle Forms 7.2.

► **To upgrade from Form Server 6.0 to LiveCycle Forms 7.2:**

1. Undeploy the earlier product from the application server. (For information, see your application server documentation.)
2. Uninstall the earlier product using the uninstaller program. (For information, see the Form Server 6.0 *Installation and Configuration Guide*.)
3. Upgrade the application server and database to ensure that they meet the system and software requirements for LiveCycle 7.2 products.

4. Install and deploy LiveCycle Forms 7.2, and initialize the database using the instructions in this document for the manual installation and deployment option. For WebSphere and WebLogic, you can automatically deploy the product, initialize the database, and verify the deployment using Configuration Manager.

Note: If you did not include User Management with the installation and configuration of LiveCycle Forms, you do not need to connect to a database, and you do not need to follow steps to initialize the database.

LiveCycle Forms and LiveCycle Print 7.x to LiveCycle Forms and LiveCycle Print 7.2

This procedure describes how to upgrade from LiveCycle Forms 7.0 or 7.1 to LiveCycle Forms 7.2, and from LiveCycle Print 7.1 to LiveCycle Print 7.2.

It is recommended that you install LiveCycle Forms 7.2 to a new directory so that you do not overwrite the previously installed version.

► **To retrieve form configuration properties (for LiveCycle Forms 7.0 or 7.1 configured *without* User Management):**

1. Type the following URL in a web browser:

```
http:// [host_name] : [port] /FormServerAdmin/settings.html
```

The default port number for WebLogic is 7001. If you are running a Managed Server, you may have configured the application server to use a different port number, such as 8001. The default port for WebSphere is 9080; the default port for JBoss is 8080.

2. Record the settings that appear on this page.

► **To retrieve form configuration properties (for LiveCycle Forms 7.0 or 7.1 configured *with* User Management):**

1. Type the following URL in a web browser:

```
http:// [host_name] : [port] /adminui
```

The default port number for WebLogic is 7001. If you are running a Managed Server, you may have configured the application server to use a different port number, such as 8001. The default port for WebSphere is 9080; the default port for JBoss is 8080.

2. Log into Administrator.
3. Click **Services**, and then click **Adobe LiveCycle Forms**.
4. Record the settings that appear on this page.
5. These properties must be set for the new installation on the Form Server Module Configuration screen on Configuration Manager when you install and configure LiveCycle Forms 7.2.

► **To upgrade to LiveCycle Forms 7.2 and LiveCycle Print 7.2:**

1. Back up the database that currently contains the LiveCycle Forms 7.0 or 7.1 configuration and run-time data.
2. Ensure that you have a back-up copy of the currently deployed LiveCycle EAR and WAR files that are configured for the current production system that you are planning to upgrade.
3. Back up the XDC files from LiveCycle Print 7.1.
4. (JBoss) If your LiveCycle deployment is running on a JBoss Application Server with a MySQL database that you installed using the turnkey option, stop the JBoss for Adobe LiveCycle service and the MySQL for Adobe LiveCycle service.
5. Undeploy the following files from your application server:
 - LiveCycle.ear
 - FormsIVS.ear
 - adobe-FontManager.war
 - LCMBootstrapper.war
 - adobe-printSubmitter.ear (LiveCycle Print only)

For information about undeploying from WebLogic or WebSphere, see “Uninstalling EAR files” in the *Installing and Configuring LiveCycle* guide.

Note: When the LiveCycle.ear file is undeployed, LiveCycle Print 7.1 does not work.

6. Stop the WebLogic Server or WebSphere Application Server.
7. Uninstall the previous versions of LiveCycle products using the uninstaller program. (See “Uninstalling LiveCycle Products” in the related *Installing and Configuring LiveCycle* guide.)
8. Upgrade the application server and database to ensure that they meet the system and software requirements for LiveCycle Forms 7.2.
9. Install LiveCycle Forms 7.2 to a new (non-default) directory (for example, C:\Adobe\LiveCycle72\ or /opt/adobe/livecycle72/). Follow the instructions in “Installing LiveCycle Products” in the *Installing and Configuring LiveCycle* guide.

Note: If you are installing or upgrading multiple LiveCycle 7.2 products, be sure to install them to the same root *[LiveCycle72 root]* directory.
10. Copy the DocumentServicesLibrary.jar file, according to your application server:
 - (WebLogic) From the *[LiveCycle72 root]/components/csa/weblogic/lib/adobe* directory to the *[appserverdomain]/lib* directory.
 - (WebSphere) From the *[LiveCycle72 root]/components/csa/websphere/lib/adobe* directory to the *[appserver root]/optionalLibraries*.
 - (JBoss) From the *[LiveCycle72 root]/components/csa/jboss/lib/adobe* directory to the *[appserver root]/server/all/lib* directory.

Note: Modify the XDC files installed with LiveCycle Print 7.2 to match those that you are using with version 7.1, and use these modified files. For information about the XDC files included with LiveCycle Print, see the *Getting Started* guide and the *Editing XDC Files to Customize Printing Workflows* guide for LiveCycle Print.

11. Start the application server.
12. Configure LiveCycle Forms 7.2 using Configuration Manager. (See the “Configuring LiveCycle Products for Deployment” chapter in the *Installing and Configuring LiveCycle* guide.) As you proceed through the Configuration Manager screens, choose the following options:
 - **Configuration Mode:** Select **Custom Configuration Wizard**.
 - **Product Selection:** Select the application server you are using, as well as **Foundation**, **LiveCycle Forms**, and **LiveCycle Print** (if applicable).
 - **Adobe User Management Selection:** Select one of the following options:
 - **LiveCycle Forms with User Management and Administrator** if you used LiveCycle Forms with User Management in the previous deployment
 - **LiveCycle Forms without User Management and Administrator** if you did not previously use LiveCycle Forms with User Management
 - **Task Selection:** Select the following options:
 - **Configure and Assemble products**
 - **Bootstrap Database** (only if you are including User Management in the configuration)

For WebSphere and WebLogic, also select these options:

- **Deploy products**
- **Verify deployed products**

13. Follow the instructions on the remaining Configuration Manager screens. Accept the default values in Configuration Manager whether you are configuring with User Management and Administrator or without User Management and Administrator.

Note: When prompted to specify the Global Storage Directory location, specify the same location that you currently use for this directory.

If you are deploying to WebSphere or WebLogic, complete steps 14 and 15. If you are deploying to JBoss, complete steps 16 and 17.

14. (WebSphere and WebLogic) On the Confirm Products to Deploy screen, select these EAR files:
 - LiveCycle.ear
 - adobe-FontManager.ear.
 - adobe-printSubmitter.ear (LiveCycle Print only)
15. (WebSphere and WebLogic) Follow the instructions on the Configuration Manager screens to initialize the database and verify the deployed products.
16. (JBoss) Deploy LiveCycle Forms 7.2. (See “Manually Deploying to JBoss” in the *Installing and Configuring LiveCycle* guide.)
17. (JBoss) (User Management configuration) Run Configuration Manager to initialize the database. Select **Custom Configuration Wizard**, and then select **Bootstrap** database.

Note: Initializing the database is necessary to add new table columns to the database schema. Initializing the database does not alter existing data.

18. Deploy the FormsIVS.ear files according to the steps required for your application server. (See “LiveCycle Forms post-deployment tasks” in the *Installing and Configuring LiveCycle* guide.)

19. Verify the deployment by following the instructions in “LiveCycle Forms post-deployment tasks” in the *Installing and Configuring LiveCycle* guide.
20. Update your application’s class path with the location of these JAR files: `formserver-client.jar`, `adobe-common.jar`, `datamanager-client.jar`, and `AdobeCSAUtils.jar`. Add the `um-client.jar` file to the application class path if the application passes an `InvocationContext` object in the Form Server Module API. For details about these files, see *Developing Custom Applications* located in the `[LiveCycle72 root]/forms/documentation` directory.

► **To configure LiveCycle Forms (when not using User Management):**

1. Type the following URL in a web browser:

```
http://[host_name]:[port]/FormServerAdmin/settings.html
```

The default port number for WebLogic is 7001. If you are running a Managed Server, you may have configured the application server to use a different port number, such as 8001. The default port for WebSphere is 9080; the default port for JBoss is 8080.

2. Enter the settings that you recorded when you retrieved the LiveCycle Forms configuration settings from the previous LiveCycle.ear file. (See [“To retrieve form configuration properties \(for LiveCycle Forms 7.0 or 7.1 configured without User Management\):”](#) on page 121.)
3. Click **Save**.

Note: The settings modified here are not retained when the application server is restarted.

► **To configure LiveCycle Forms (when using User Management):**

1. Type the following URL in a web browser:

```
http://[host_name]:[port]/adminui
```

2. The default port number for WebLogic is 7001. If you are running a Managed Server, you may have configured the application server to use a different port number, such as 8001. The default port for WebSphere is 9080; the default port for JBoss is 8080.
3. Log into Administrator.
4. Click **Services**, and then click **Adobe LiveCycle Forms**.
5. Enter the settings you recorded when you retrieved the LiveCycle Forms configuration settings from the previous LiveCycle.ear file. (See [“To retrieve form configuration properties \(for LiveCycle Forms 7.0 or 7.1 configured with User Management\):”](#) on page 121.)
6. Click **Save**.

LiveCycle Form Manager

This section provides instructions for upgrading from LiveCycle Form Manager 7.0.1 to LiveCycle Form Manager 7.2. To perform this upgrade, you must update the server components that are deployed to the application server and initialize the database.

It is recommended that you install LiveCycle Form Manager 7.2 to a new directory so that you do not overwrite the previously installed version.

► **To upgrade LiveCycle Form Manager 7.0.1 to 7.2:**

1. Back up the database that currently contains the LiveCycle Form Manager 7.0.1 configuration and run-time data.
2. Ensure that you have a back-up copy of the currently deployed LiveCycle EAR and WAR files that are configured for the current production system that you are planning to upgrade.
3. (JBoss) If your LiveCycle deployment is running on a JBoss Application Server with a MySQL database that you installed using the turnkey option, stop the JBoss for Adobe LiveCycle service and the MySQL for Adobe LiveCycle service.
4. Undeploy the following LiveCycle Form Manager 7.0.1 components by following the instructions in the *Installing and Configuring* guide for LiveCycle Form Manager 7.0.1:
 - adobe-FontManager.war
 - LiveCycle.ear
 - LCMBostrapper.war

For information about undeploying from WebLogic or WebSphere, see “Uninstalling EAR files” in this *Installing and Configuring LiveCycle* guide.

5. Stop the WebLogic Server or WebSphere Application Server.
6. Uninstall the previous version of LiveCycle products using the uninstaller program. (See “Uninstalling LiveCycle Products” in the related *Installing and Configuring LiveCycle* guide.)
7. Upgrade the application server and database to ensure that they meet the system and software requirements for LiveCycle 7.2 products.
8. Install LiveCycle Form Manager 7.2 to a new (non-default) directory (for example, C:\Adobe\LiveCycle72\ or /opt/adobe/livecycle72/). Follow the instructions in “Installing LiveCycle Products” in the *Installing and Configuring LiveCycle* guide.

Note: If you are installing or upgrading multiple LiveCycle 7.2 products, be sure to install them to the same *[LiveCycle72 root]* directory.

9. Copy the DocumentServicesLibrary.jar file, according to your application server:
 - (WebLogic) From the *[LiveCycle72 root]/components/csa/weblogic/lib/adobe* directory to the *[appserverdomain root]/lib* directory
 - (WebSphere) From the *[LiveCycle72 root]/components/csa/websphere/lib/adobe* directory to the *[appserver root]/optionalLibraries* directory
 - (JBoss) From the *[LiveCycle72 root]/components/csa/jboss/lib/adobe* directory to the *[appserver root]/server/all/lib* directory
10. Start the application server.

11. Configure LiveCycle Form Manager 7.2 using Configuration Manager. (See “Configuring LiveCycle Products for Deployment” in the *Installing and Configuring LiveCycle* guide.) As you proceed through the Configuration Manager screens, choose the following options:
 - **Configuration Mode:** Select **Custom Configuration Wizard**.
 - **Product Selection:** Select the application server you are using, as well as **Foundation** and the product you are configuring.
 - **Task Selection:** Select the following options:
 - **Configure and Assemble products**
 - **Bootstrap Database**

For WebSphere and WebLogic, also select these options:

 - **Deploy products**
 - **Verify deployed products**
12. Follow the instructions on the Configuration Manager screens to configure the LiveCycle.ear file. Ensure that you configure your LiveCycle 7.2 EAR file with the same settings that you chose for version 7.0.1. If you are deploying to WebSphere or WebLogic, complete steps 13 and 14. If you are deploying to JBoss, complete steps 15 and 16.

Note: When prompted to specify the Global Storage Directory location, specify the same location that you currently use for this directory.

13. (WebSphere and WebLogic) On the Confirm Products to Deploy screen, select these EAR files:
 - LiveCycle.ear
 - adobe-FontManager.ear
14. (WebSphere and WebLogic) Follow the instructions on the Configuration Manager screens to initialize the database and verify the deployed products.
15. (JBoss) Deploy LiveCycle Form Manager 7.2 components. (See “Manually Deploying to JBoss” in the *Installing and Configuring LiveCycle* guide.)
16. (JBoss) Run Configuration Manager to initialize the database. Select **Custom Configuration Wizard**, and then select **Bootstrap database**. (See “Initializing the Database” in the *Installing and Configuring LiveCycle* guide.)

Note: Initializing the database is necessary to add new table columns to the database schema. Initializing the database does not alter existing data.

17. Verify the deployment by following the instructions in “LiveCycle Form Manager post-deployment tasks” the *Installing and Configuring LiveCycle* guide.

Note: You can also verify your installation and configuration by using the samples available at www.adobe.com/devnet/livecycle/samples.html.

LiveCycle Assembler, LiveCycle Workflow, and Watched Folder

This section describes how to upgrade any or all of the following products:

- LiveCycle Workflow 7.0.1 or 7.2 to LiveCycle Workflow 7.2.1
- LiveCycle Assembler 7.2 to LiveCycle Assembler 7.2.1
- Watched Folder 1.0 to Watched Folder 1.1

If you are upgrading LiveCycle Workflow as a stand-alone product, do not include the steps that pertain to Watched Folder or LiveCycle Assembler. Watched Folder is intended for use with LiveCycle Assembler and LiveCycle Workflow; you must install LiveCycle Workflow before installing Watched Folder.

To perform this upgrade, you must update the server components that are deployed to the application server and initialize the database. It is recommended that you install LiveCycle Assembler 7.2.1 to a new directory so that you do not overwrite the previously installed version.

When you configure LiveCycle Assembler 7.2.1 using Configuration Manager, ensure that you set the Security Groups option to the same value that you have set for LiveCycle Assembler 7.2. (The value can be set to an asterisk (*), which only allows users with a valid entry in the LDAP system to log in, or to no value, which allows any user to log in.)

► To upgrade to LiveCycle Assembler 7.2.1, LiveCycle Workflow 7.2.1, and Watched Folder 1.1:

1. Back up the database that currently contains the LiveCycle Workflow 7.0.1 or 7.2 configuration and run-time data.
2. Ensure that you have a back-up copy of the currently deployed LiveCycle EAR and WAR files that are configured for the current production system that you are planning to upgrade.
3. (JBoss) If your LiveCycle deployment is running on a JBoss Application Server with a MySQL database that you installed using the turnkey option, stop the JBoss for Adobe LiveCycle service and the MySQL for Adobe LiveCycle service.
4. Undeploy the following files from your application server:
 - LiveCycle.ear
 - adobe-FontManager.war
 - LCMBootstrapper.war
 - adobe-Assembler7.ear (LiveCycle Assembler only)

For information about undeploying from WebLogic or WebSphere, see “Uninstalling EAR files” in the *Installing and Configuring LiveCycle* guide.

5. (WebSphere and WebLogic) Stop the application server.
6. Uninstall the previous versions of LiveCycle products using the uninstaller program. (See “Uninstalling LiveCycle Products” in the *Installing and Configuring LiveCycle* guide.)
7. Install LiveCycle Assembler 7.2.1 to a new (non-default) directory (for example, C:\Adobe\LiveCycle72\ or /opt/adobe/livecycle72/). Follow the instructions in “Installing LiveCycle Products” in the *Installing and Configuring LiveCycle* guide.

8. Install LiveCycle Workflow 7.2.1 to the same directory where you installed LiveCycle Assembler 7.2.1 by following the instructions in “Installing LiveCycle Products” in the related *Installing and Configuring LiveCycle* guide.

Note: You can install LiveCycle Assembler and LiveCycle Workflow in any order.

9. Install Watched Folder to the same directory where you installed LiveCycle Workflow by following the instructions in “Installing Watched Folder” in the *Installing and Configuring LiveCycle* guide. LiveCycle Workflow must already be installed.

Note: If you are installing or upgrading multiple LiveCycle 7.2 products, be sure to install them to the same *[LiveCycle72 root]* directory.

10. Copy the DocumentServicesLibrary.jar file, according to your application server:
 - (WebLogic) From the *[LiveCycle72 root]/components/csa/weblogic/lib/adobe* directory to the *[appserverdomain]/lib* directory.
 - (WebSphere) From the *[LiveCycle72 root]/components/csa/websphere/lib/adobe* directory to the *[appserver root]/optionalLibraries*.
 - (JBoss) From the *[LiveCycle72 root]/components/csa/jboss /lib/adobe* directory to the *[appserver root]/server/all/lib* directory. Copy the adobe-service.xml file from the *[LiveCycle72_root]/configurationManager/deploy/jboss* directory to the *[appserver root]/server/all/deploy/jms* directory.
11. Start the application server.
12. Configure the LiveCycle products by using Configuration Manager. (See “Configuring LiveCycle Products for Deployment” in the *Installing and Configuring LiveCycle* guide.) As you proceed through the Configuration Manager screens, choose the following options:
 - **Configuration Mode:** Select **Custom Configuration Wizard**.
 - **Product Selection:** Select the application server you are using, and then select **Foundation**, as well as all of the products that you installed.
 - **Task Selection:** Select the following options:
 - **Configure and Assemble products**
 - **Bootstrap Database**For WebSphere and WebLogic, also select these options:
 - **Deploy products**
 - **Verify deployed products**
13. Follow the instructions on the remaining Configuration Manager screens. Ensure that you configure your 7.2.1 EAR file with the same settings you chose for version 7.0.1 or 7.2. If you are deploying to WebSphere or WebLogic, complete steps 14 and 15. If you are deploying to JBoss, complete steps 16 and 17.

Note: When prompted to specify the Global Storage Directory location, specify the same location that you currently use for this directory.
14. (WebSphere and WebLogic) On the **Confirm Products to Deploy** screen, select the archives that you are deploying:
 - adobe-FontManager.ear
 - LiveCycle.ear
 - adobe-Assembler7.ear (LiveCycle Assembler only)

15. (WebSphere and WebLogic) Follow the instructions on the Configuration Manager screens to initialize the database and verify the deployed products.
 16. (JBoss) Deploy LiveCycle Assembler, LiveCycle Workflow, and Watch Folder 7.2. (See “Manually Deploying to JBoss” in the *Installing and Configuring LiveCycle* guide.) These components consist of these EAR files:
 - LiveCycle.ear
 - adobe-FontManager.ear
 - LCM.ear
 - adobe-Assembler7.ear
 17. (JBoss) Run Configuration Manager to reinitialize the database. Select **Custom Configuration Wizard**, and then select **Bootstrap database**. (See “Initializing the Database” in the *Installing and Configuring LiveCycle* guide.)
 18. Upgrade the Assembler QPAC. (See *Upgrading LiveCycle Workflow Designer* or the *Creating Workflows* guide, available with the LiveCycle Workflow documentation set.)
- Note:** Initializing the database is necessary to add new table columns to the database schema. Initializing the database does not alter existing data.
19. Verify the deployment by following the instructions in the “Post Deployment” section in the *Installing and Configuring LiveCycle* guide.

Note: You can also verify your installation and configuration by using the samples available at www.adobe.com/devnet/livecycle/samples.html.

LiveCycle Workflow Designer

You need to use the version of LiveCycle Workflow Designer that LiveCycle Workflow 7.2.1 provides. If you want to use the new User QPAC with existing workflows, you need to migrate your workflows.

You must upgrade your LiveCycle QPACs from LiveCycle Workflow 7.0.1 or 7.2 to version 7.2.1. See “Updating components” in the *Creating Workflows* guide that is included with LiveCycle Workflow Designer.

► To upgrade LiveCycle Workflow Designer:

1. Uninstall LiveCycle Workflow 7.0.1 Designer.
2. Install LiveCycle Workflow 7.2.1 Designer. (See “Installing LiveCycle Workflow Designer” in the *Installing and Configuring LiveCycle* guide.)

Note: LiveCycle Workflow 7.2.1 provides an updated User QPAC that includes new features. If any of your workflows include User actions and you want to use the new features, you must migrate your workflows. For more information, see the topic “Migrating Workflows from Older Versions” in the *Creating Workflows* guide or in *LiveCycle Workflow Designer Help*.

BAM Server for LiveCycle Workflow

This section describes general best practices to follow when upgrading to a new version of BAM Server. You must use the BAM Server upgrade utility to upgrade metadata XML files that you export from LiveCycle Workflow Business Activity Monitor during the upgrade process.

Upgrading BAM Server involves performing the following tasks:

- Exporting the BAM metadata using BAM Workbench
- Upgrading the BAM metadata using the BAM Server upgrade utility
- Undeploying the BAM Server EAR file from the application server
- Removing recovery log files and dropping existing tables into the BAM metadata database
- Deploying the BAM Server EAR file that LiveCycle Workflow 7.2.1 provides
- Importing the upgraded BAM metadata using BAM Workbench

About the BAM Server upgrade utility

The BAM Server upgrade utility upgrades exported metadata files so that they include new features required for the current product version.

The BAM Server upgrade utility is packaged in the `cqupgrade.jar` file. When you install LiveCycle Workflow, this file is located in the `[LiveCycle root]/Workflow/bam/CQUpgrade` directory.

Note: The upgrade utility generates the message “Error parsing input file” when files cannot be upgraded. If this error is generated, the file specified in the error message may be damaged. Contact Adobe Systems for assistance.

Syntax

The following command upgrades metadata files that have been exported to a JAR file:

```
java -jar cqupgrade.jar -jar -i inJar.jar [-o outJar.jar]
```

Parameters

You can use the following parameters in the BAM Server upgrade utility command:

inJar.jar: The path to the JAR file that contains the exported metadata to upgrade.

outJar.jar (optional): The path to the upgraded JAR file. If you omit this parameter, the upgraded files are saved in the JAR file specified in *inJar.jar*.

The following examples show commands that you can use to upgrade metadata stored in JAR files using the BAM Server upgrade utility:

- The following command upgrades the metadata files that are stored in a JAR file named `toUpgrade.jar`, which is located in the same directory as `cqupgrade.jar`:

```
java -jar cqupgrade.jar -i toUpgrade.jar
```

- The following command upgrades the metadata files in the `toUpgrade.jar` file and saves the upgraded files to a different JAR file named `upgraded.jar`. The file is saved in the same directory as `cqupgrade.jar`:

```
java -jar cqupgrade.jar -i toUpgrade.jar -o upgraded.jar
```

► **To upgrade Business Activity Monitor:**

1. Log into BAM Workbench.
2. Click the **Administration** tab and then click **Import/Export**.
3. In the **Operations** menu, select **Export Metadata to a JAR File (download)**.
4. In the File Download dialog box, click **Save**.
5. Specify a location and file name for the exported JAR file, and click **Save**.
6. Click **Close** in the Download Complete dialog box.
7. Click **System Settings**, and then click the **Checkpoint Configuration** tab. Note the directory path in the **Recovery Log Directory** box. You will need to know where the recovery log directory is at a later step in this procedure.
8. Shut down Business Activity Monitor.
9. Use the BAM Server upgrade utility (cqupgrade.jar) to upgrade the metadata that you exported in a JAR file. (See [“About the BAM Server upgrade utility” on page 130.](#))
10. Undeploy the BAM Server EAR file by following the instructions for your application server:
 - (JBoss) Remove the following directories:
 - `[jboss bam root]/server/default/work`
 - `[jboss bam root]/server/default/tmp`
 - `[jboss bam root]/server/default/data`
 - `[jboss bam root]/server/conf/jboss.web`
 - (WebSphere) Undeploy the old BAM Server EAR file using WebSphere Administrative Console.
 - (WebLogic) Undeploy the old BAM Server EAR file using WebLogic Server Administration Console.
11. Delete all of the files from the recovery log directory that have names similar to the following patterns:
 - `filestore*.dat`
 - `DEFAULTRECOVERYLOGGER_*`
 - `chkpoint_.x`
12. Use your database management tools to drop the database tables that store the BAM metadata. Alternatively, you may want to create a new BAM metadata database. For more information about the BAM metadata database, see “Creating the BAM metadata database” in the *Installing and Configuring LiveCycle* guide for your application server.
13. Deploy the new version of BAM Server to the application server. (See “Deploying BAM Server” in the *Installing and Configuring LiveCycle* guide.)
14. Log into BAM Workbench.
15. Review the BAM Server configuration settings to determine if updates are required. (See “Getting Started with BAM Server” in the *Installing and Configuring LiveCycle* guide.)
16. Click the **Administration** tab and then click **Import/Export**.

17. In the **Operations** menu, select **Import Metadata from a JAR File (upload)**.
18. Click **Browse** to locate the JAR file that contains the upgraded BAM metadata, and then click **OK**.
19. Restart the BAM Server instance.

LiveCycle PDF Generator

To upgrade from LiveCycle PDF Generator 7.0.1 or 7.0.2 to LiveCycle PDF Generator 7.2, you must undeploy the product you are currently using before installing and deploying the new product.

The instructions in this section apply to LiveCycle PDF Generator for PostScript, LiveCycle PDF Generator Elements, and LiveCycle PDF Generator Professional.

► To upgrade LiveCycle PDF Generator 7.0.2 to LiveCycle PDF Generator 7.2:

1. Back up the database that currently contains the LiveCycle PDF Generator 7.0.1 or 7.0.2 configuration and run-time data.
2. Ensure that you have a back-up copy of the currently deployed LiveCycle EAR and WAR files that are configured for the current production system that you are planning to upgrade.
3. (JBoss) If your LiveCycle PDF Generator deployment is running on a JBoss Application Server with a MySQL database that you installed using the turnkey option, stop the JBoss for Adobe LiveCycle service.
4. If you are installing LiveCycle PDF Generator Professional or LiveCycle PDF Generator Elements, uninstall Adobe Acrobat 7.0.5 from the Add/Remove programs window in the Microsoft Windows Control Panel, and then reboot your system.
5. Undeploy the following LiveCycle PDF Generator components by following the instructions in the *Installing and Configuring* guide for LiveCycle PDF Generator 7.0.2:
 - pdfg-all.ear (or pdfg-ps-all.ear)
 - LiveCycle.ear
 - adobe-FontManager.war
 - LCMBootstrapper.war

For information about undeploying from WebLogic or WebSphere, see “Uninstalling EAR files” in the *Installing and Configuring LiveCycle* guide.

6. (WebSphere and WebLogic) Stop the application server.
7. Uninstall the previous versions of LiveCycle products by using the uninstaller program. (See “Uninstalling LiveCycle Products” in the *Installing and Configuring LiveCycle* guide.)
8. Install LiveCycle PDF Generator 7.2 to a new (non-default) directory (for example, C:\Adobe\LiveCycle72\ **or** /opt/adobe/livecycle72/). Follow the instructions in “Installing LiveCycle Products” in the *Installing and Configuring LiveCycle* guide.

Note: If you are installing or upgrading multiple LiveCycle 7.2 products, be sure to install them to the same *[LiveCycle72 root]* directory.

9. Copy the DocumentServicesLibrary.jar file, according to your application server:
 - (WebLogic) From the `[LiveCycle72 root]/components/csa/weblogic/lib/adobe` directory to the `[appserverdomain]/lib` directory
 - (WebSphere) From the `[LiveCycle72 root]/components/csa/websphere/lib/adobe` directory to the `[appserver root]/optionalLibraries`
 - (JBoss) From the `[LiveCycle72 root]/components/csa/jboss/lib/adobe` directory to the `[appserver root]/server/all/lib` directory. Copy the `adobe-service.xml` file from the `[LiveCycle72_root]/configurationManager/deploy/jboss` directory to the `[appserver root]/server/all/deploy/jms` directory.
10. Start the application server.
11. Configure the LiveCycle products by using Configuration Manager. (See “Configuring LiveCycle Products for Deployment” in the *Installing and Configuring LiveCycle* guide.) As you proceed through the Configuration Manager screens, choose the following options:
 - **Configuration Mode:** Select **Custom Configuration Wizard**.
 - **Product Selection:** Select the application server you are using, and then select **Foundation**, and **LiveCycle PDF Generator**.
 - **Task Selection:** Select all of the following options:
 - **Configure and Assemble products**
 - **Bootstrap Database**For WebSphere and WebLogic, also select these options:
 - **Deploy products**
 - **Verify deployed products**
12. Follow the instructions on the remaining Configuration Manager screens. Ensure that you configure your LiveCycle 7.2 EAR file with the same settings you chose for version 7.0.2. If you are deploying to WebSphere or WebLogic, complete steps 13 and 14. If you are deploying to JBoss, complete steps 15 and 16.

Note: When prompted to specify the Global Storage Directory location, specify the same location that you currently use for this directory.
13. (WebSphere and WebLogic) On the Confirm Products to Deploy screen, select the archives that you are deploying:
 - pdfg-all.ear (or pdfg-ps-all.ear)
 - adobe-FontManager.ear
 - LiveCycle.ear
14. (WebSphere and WebLogic) Follow the instructions on the Configuration Manager screens to initialize the database and verify the deployed products.
15. (JBoss) Deploy the LiveCycle PDG Generator 7.2 components:
 - LiveCycle.ear
 - adobe-FontManager.ear
 - LCM.ear
 - pdfg-all.ear (or pdfg-ps-all.ear)(See “Manually Deploying to JBoss” in the *Installing and Configuring LiveCycle* guide.)

16. (JBoss) Run Configuration Manager to reinitialize the database. Select **Custom Configuration Wizard**, and then select **Bootstrap database**. (See “Initializing the Database” in the *Installing and Configuring LiveCycle* guide.)

Note: Initializing the database is necessary to add new table columns to the database schema. Initializing the database does not alter existing data.

17. Verify the deployment by following the instructions in “LiveCycle PDF Generator post-deployment tasks” in the *Installing and Configuring LiveCycle* guide.

Note: You can also verify your installation and configuration by using the samples available at www.adobe.com/devnet/livecycle/samples.html.

► **To configure JMS Messaging for WebSphere running on Windows:**

1. In the WebSphere Administrative Console, delete the JMS queues, topics, and listener ports for each server.
2. Configure WebSphere MQ by following the instructions in “Configuring JMS resources for WebSphere MQ” in the *Installing and Configuring LiveCycle* guide.
3. Run the scripts provided by Configuration Manager to create queues in the MQ installation by following the instructions in “Preparing WebSphere MQ” in the *Installing and Configuring LiveCycle* guide.
4. Run Configuration Manager again and select the **Configure the Application Server** task. Follow the instructions on the Configuration Manager screens to configure application server details. (See “Changing the application server settings” in the *Installing and Configuring LiveCycle* guide.) As you proceed through the Configuration Manager screens, choose the following options:
 - **JVM Settings:** Select **Do Not Apply**.
 - **JMS settings:** Specify values for the properties of the MQ service.
 - **Datasource Creation:** Select **Do Not Apply**.
5. On the Configure Application Server screen, click **Apply Settings Now** to configure your application server, and, when the task is complete, click **Next**.
6. Restart the application server and, on the Application Server/Cluster Restart screen, click **Next** if you selected another task for Configuration Manager to perform, or click **Finish** to close Configuration Manager.
7. Verify that the applications start correctly.

LiveCycle Document Security

This section provides instructions for upgrading from LiveCycle Document Security 7.0 or 7.1 to LiveCycle Document Security 7.2. To perform this upgrade, you must update the server components that are deployed to the application server.

It is recommended that you install LiveCycle Document Security 7.2 to a new directory so that you do not overwrite the previously installed version.

► **To upgrade to LiveCycle Document Security 7.2:**

1. Back up the following files and folders:
 - ../trust.xml
 - ../credentials/
 - ../certificates/
 - ../CRLs/
 - ../keystore
 - ../trust.sig
2. Ensure that you have a back-up copy of the currently deployed LiveCycle EAR and WAR files that are configured for the current production system that you are planning to upgrade.
3. Undeploy the following LiveCycle Document Security components by following the instructions in the *Installing and Configuring* guide for LiveCycle Document Security 7.0 or 7.1:

WebSphere and WebLogic	JBoss
adobe-FontManager.war	adobe-FontManager.bar
adobe-PDFManipulation.war	adobe-PDFManipulation.bar
adobe-TrustManager.war	adobe-TrustManager.bar
DataManagerService.war	AdobeServices.sar
ServicesNatives-2.war	ServicesNatives-2.war
adobe-APSPProxy.war	adobe-APSPProxy.bar

4. (WebSphere) Set up default users, roles, and login files by following in the instructions in the procedure "To map users to roles" in the *Installing and Configuring LiveCycle Security Products* guide. (This procedure must be completed each time the LiveCycle-security.ear file is redeployed.)
5. Restart the application server.
6. Uninstall the previous version of LiveCycle products by using the uninstaller program. (See "Uninstalling LiveCycle Products" in the related *Installing and Configuring LiveCycle* guide.)
7. Install LiveCycle Document Security 7.2 to a new (non-default) directory (for example, C:\Adobe\LiveCycle72\ or /opt/adobe/livecycle72/). Follow the instructions in "Installing LiveCycle Products" in the *Installing and Configuring LiveCycle Security Products* guide.

Note: If you are installing or upgrading multiple LiveCycle 7.2 products, be sure to install them to the same [LiveCycle72 root] directory.
8. Copy the DocumentServicesLibrary.jar file, according to your application server:
 - (JBoss) From the [LiveCycle72 root]/components/csa/jboss/lib/adobe directory to the [appserver root]/server/all/lib directory.
 - (WebLogic) From the [LiveCycle72 root]/components/csa/weblogic/lib/adobe directory to the [appserver domain]/lib directory.
 - (WebSphere) From the [LiveCycle72 root]/components/csa/websphere/lib/adobe directory to the [appserver root]/optionalLibraries directory.

9. Configure LiveCycle Document Security 7.2 using Configuration Manager. (See “Configuring LiveCycle Products”.) As you proceed through the Configuration Manager screens, choose the following options:
 - **Configuration Mode:** Select **Custom Configuration Wizard**.
 - **Product Selection:** Select the application server you are using, as well as **Foundation** and the product you are configuring.
 - **Task Selection:** Select the following options:
 - **Configure and Assemble products**
 - **Bootstrap Database**For WebSphere and WebLogic, also select these options:
 - **Deploy products**
 - **Verify deployed products**
 - **Trust Directory Selection:** Select **Create a new trust directory**.
 - **Keystore Selection:** Select **Create a new keystore and pair**.
10. Follow the instructions on the remaining Configuration Manager screens. Ensure that you configure your 7.2 EAR file with the same settings you chose for version 7.0 or 7.1. If you are deploying to WebSphere or WebLogic, complete steps 11 and 12. If you are deploying to JBoss, complete steps 13 and 14.

Note: When prompted to specify the Global Storage Directory location, specify the same location that you currently use for the “Directory for Adobe LiveCycle products” temp file.
11. (WebSphere and WebLogic) On the Confirm Products to Deploy screen, select the archives that you are deploying:
 - adobe-FontManager.ear
 - LiveCycle.ear
 - LiveCycle-security.ear
12. (WebSphere and WebLogic) Follow the instructions on the Configuration Manager screens to initialize the database and verify the deployed products.
13. (JBoss) Deploy LiveCycle Document Security 7.2. (See “Manually Deploying to JBoss” in the *Installing and Configuring LiveCycle Security Products* guide.)
14. (JBoss) Run Configuration Manager to reinitialize the database. Select **Custom Configuration Wizard**, and then select **Bootstrap database**. (See “Initializing the Database” in the *Installing and Configuring LiveCycle Security Products* guide.)
15. Restart the application server.
16. Verify your installation and configuration by using the samples available at www.adobe.com/devnet/livecycle/samples.html.

LiveCycle Reader Extensions

This section provides instructions for upgrading from LiveCycle Reader Extensions 7.0 or 7.0.2 to LiveCycle Reader Extensions 7.2. To perform this upgrade, you must update the server components that are deployed to the application server.

Note: (JBoss) If you previously installed LiveCycle Reader Extensions on Windows for deployment to JBoss using the turnkey method, see “To upgrade LiveCycle Reader Extensions using the turnkey method” in the *Installing and Configuring LiveCycle Security Products* guide for the turnkey upgrade instructions.

It is recommended that you install LiveCycle Reader Extensions 7.2 to a new directory so that you do not overwrite the previously installed version.

If you are using an existing credential, ensure the credential has not expired and is still valid. (See “LiveCycle Reader Extensions Rights credential” in the *Installing and Configuring LiveCycle Security Products* guide.)

Custom applications using existing APIs supported in Adobe Document Services 6.0 for Reader Extensions are not supported in LiveCycle Reader Extensions 7.2. Custom applications using existing APIs supported in Adobe Reader Extensions Server 6.1 are supported by and continue to work with LiveCycle Reader Extensions 7.2.

Digital certificates issued for use with Reader Extensions Server 6.1 are compatible for use with LiveCycle Reader Extensions 7.2. If you are upgrading from version 6.1 to 7.2, you will not receive a new Rights credential.

► To upgrade LiveCycle Reader Extensions to 7.2:

1. Back up the following files and folders in the *[LiveCycle root]/ReaderExtensions/trust* directory:
 - *../trust.xml*
 - *../credentials/*
 - *../certificates/*
 - *../CRLs/*
 - *../keystore*
 - *../trust.sig*
2. Ensure that you have a back-up copy of the currently deployed LiveCycle EAR and WAR files that are configured for the current production system that you are planning to upgrade.
3. Undeploy the following LiveCycle Reader Extensions components by following the instructions in the *Installing and Configuring* guide for LiveCycle Reader Extensions 7.0 or 7.0.2:

WebSphere and WebLogic	JBoss
adobe-FontManager.war	adobe-FontManager.bar
adobe-PDFManipulation.war	adobe-PDFManipulation.bar
adobe-TrustManager.war	adobe-TrustManager.bar
DataManagerService.war	AdobeServices.sar
ServicesNatives-2.war	ServicesNatives-2.war

WebSphere and WebLogic	JBoss
ares.ear	ares.ear
adobe-CredentialSecurityEJB.ear	adobe-CredentialSecurityEJB.ear
adobe-APSPProxy.war	adobe-APSPProxy.bar

- Restart the application server.
- Uninstall the previous version of LiveCycle products y using the uninstaller program. (See “Uninstalling LiveCycle Products” in the related *Installing and Configuring LiveCycle* guide.)
- Install LiveCycle Reader Extensions 7.2 to a new (non-default) directory (for example, C:\Adobe\LiveCycle72\ or /opt/adobe/livecycle72/. Follow the instructions in “Installing LiveCycle Products” in the *Installing and Configuring LiveCycle Security Products* guide.

Note: If you are installing or upgrading multiple LiveCycle 7.2 products, be sure to install them to the same *[LiveCycle72 root]* directory.

- Copy the DocumentServicesLibrary.jar file, according to you application server:
 - (WebLogic) From the *[LiveCycle72 root]/components/csa/weblogic/lib/adobe* directory to the *[appserver domain]/lib* directory.
 - (WebSphere) From the *[LiveCycle72 root]/components/csa/websphere/lib/adobe* directory to the *[appserver root]/optionalLibraries* directory.
 - (JBoss) From the *[LiveCycle72 root]/components/csa/jboss/lib/adobe* directory to the *[appserver root]/server/all/lib* directory.
- Configure LiveCycle Reader Extensions 7.2 by using Configuration Manager. (See “Configuring LiveCycle Products” in the *Installing and Configuring LiveCycle Security Products* guide.) As you proceed through the Configuration Manager screens, choose the following options:
 - Configuration Mode:** Select **Custom Configuration Wizard**.
 - Product Selection:** Select the application server you are using, as well as **Foundation** and the product you are configuring.
 - Task Selection:** Select the following options:
 - Configure and Assemble products**For WebSphere and WebLogic, also select these options:
 - Deploy products**
 - Verify deployed products**
 - Trust Directory Selection:** Select **Create a new trust directory**.
 - Keystore Selection:** Select **Create a new keystore and pair**.
- Follow the instructions on the remaining Configuration Manager screens. Ensure that you configure your 7.2 EAR file with the same settings you chose for version 7.0 or 7.1. If you are deploying to WebSphere or WebLogic, complete steps 10 and 11. If you are deploying to JBoss, complete step 12.

Note: When prompted to specify the Global Storage Directory location, specify the same location that you currently use for the “Directory for Adobe LiveCycle products” temp file.

10. (WebSphere and WebLogic) On the Confirm Products to Deploy screen, select the archives that you are deploying:
 - adobe-FontManager.ear
 - LiveCycle.ear
 - LiveCycle-security.ear
11. (WebSphere and WebLogic) Follow the instructions on the Configuration Manager screens to initialize the database and verify the deployed products.
12. (JBoss) Deploy LiveCycle Reader Extensions 7.2. (See “Manually Deploying to JBoss” in the *Installing and Configuring LiveCycle Security Products* guide.)
13. Restart the application server.
14. Verify the installation and configuration by going to the appropriate URL:
 - (WebLogic) `http://[host_name]:7001/ReaderExtensions` **or**
`http://[host_name]:8001/ReaderExtensions` (Managed Server)
 - (WebSphere) `http://[host_name]:9080/ReaderExtensions`
 - (JBoss) `http://[host_name]:8080/ReaderExtensions`
15. Type the user name and password you created when you configured the LiveCycle Reader Extensions user. (See “Setting up users, roles, and login files” for JBoss or WebLogic or “To map users to roles” for WebSphere in the *Installing and Configuring LiveCycle Security Products* guide.)

► **(JBoss) To upgrade LiveCycle Reader Extensions using the turnkey method:**

1. Stop the AdobeReaderExtensions Windows service from the Services window in the Administrative Tools area of the Windows Control Panel.
2. Verify that port 8080 is not being used.
3. The turnkey installation specifies “localhost” as the host and “8080” as the port for use by JBoss. If JBoss is already installed, ensure that it is not using port 8080. You cannot configure an alternative host or port for JBoss during the turnkey installation and configuration process.
4. Install LiveCycle Reader Extensions 7.2 using the turnkey method to a new (non-default) directory (for example, C:\Adobe\LiveCycle72\). Follow the instructions in “Installing LiveCycle Reader Extensions or LiveCycle Document Security” in the *Installing and Configuring LiveCycle Security Products* guide.
5. Verify the installation and configuration by going to the URL
`http://[host_name]:8080/ReaderExtensions`.

Note: For turnkey installations, the default user name is administrator and the default password is password. To edit these values, see “Setting up users, roles, and login files” in the *Installing and Configuring LiveCycle Security Products* guide.

Note: You can also verify your installation and configuration by using the samples available at www.adobe.com/devnet/livecycle/samples.html.

LiveCycle Policy Server

This section provides instructions for upgrading from LiveCycle Policy Server 7.0.2 to LiveCycle Policy Server 7.2. To perform this upgrade, you must update the server components that are deployed to the application server and initialize the database.

It is recommended that you install LiveCycle Policy Server 7.2 to a new directory so that you do not overwrite the previously installed version.

► To upgrade LiveCycle Policy Server to 7.2:

1. Back up the database that currently contains the LiveCycle Policy Server 7.0.2 configuration and run-time data.
2. Ensure that you have a back-up copy of the currently deployed LiveCycle EAR and WAR files that are configured for the current production system that you are planning to upgrade.
3. Undeploy the following LiveCycle Policy Server 7.0.2 components:
 - asn1.jar
 - jsafe.jar
 - jsafeJCE.jar
 - edc-server-spi.jar
 - edc-server.ear
4. Remove the dom*.jar files from the `[appserver root]/java/jre/lib/endorsed` directory.
5. Uninstall the previous version of LiveCycle products by using the uninstaller program. (See “Uninstalling LiveCycle Products” in the relevant *Installing and Configuring LiveCycle* guide.)
6. Install LiveCycle Policy Server 7.2 to a new (non-default) directory (for example, `C:\Adobe\LiveCycle72\` or `/opt/Adobe/livecycle72/`). Follow the instructions in “Installing LiveCycle Products” in the *Installing and Configuring LiveCycle Security Products* guide.

Note: If you are installing or upgrading multiple LiveCycle 7.2 products, be sure to install them to the same `[LiveCycle72 root]` directory.

7. Deploy LiveCycle Policy Server 7.2. (See the chapter about manually deploying to the application server in the *Installing and Configuring LiveCycle Security Products* guide for your application server.)
8. Run Configuration Manager to reinitialize the database. Select **Custom Configuration Wizard**, and then select **Bootstrap database**. (See “Initializing the Database” in the *Installing and Configuring LiveCycle Security Products* guide.)

Note: Initializing the database is necessary to add new table columns to the database schema. Initializing the database does not alter existing data.

9. Configure various run-time settings for LiveCycle Policy Server. (See the “Post-deployment” section of the *Installing and Configuring LiveCycle Security Products* guide.)

Note: You can verify your installation and configuration by using the samples available at www.adobe.com/devnet/livecycle/samples.html.

This appendix provides general tips that you can use to improve server performance when using LiveCycle products.

Optimizing inline documents and impact on JVM memory

If you are typically processing documents of a relatively small size, you can improve the performance associated with the document transfer speed and storage space by implementing the following LiveCycle product configurations:

- Increase the maximum inline size for LiveCycle products so that it is larger than the size of most documents.
- For processing larger files, specify storage directories that are located on a high-speed disk system or a RAM disk.

The default maximum inline size and the storage directories (the Adobe LiveCycle products temporary file directory and the Global storage directory) are properties of the Data Manager Module. You can configure the Data Manager Module using Configuration Manager. (See [“Configuring LiveCycle Products for Deployment” on page 48.](#))

Note: The default maximum inline size is 65536 bytes.

Document size and maximum inline size

When a document that is sent for processing by LiveCycle products is less than or equal to the maximum inline size, the document is stored on the server inline and the document is serialized as an Adobe Document object. Storing documents inline can have significant performance benefits. However, if you are using LiveCycle Workflow, the content may also be stored in the database for tracking purposes, and so increasing the maximum inline size might affect the database size.

A document that is larger than the maximum inline size is stored on the local file system (in the storage directories specified using Configuration Manager), and the Adobe Document object that is transferred to and from the server is only a pointer to that file.

When document content is inlined (that is, less than the maximum inline size), the content is stored in the database (as part of the Document's serialization payload). So, increasing the maximum inline size might affect the database size.

JVM maximum heap size

An increase in the maximum inline size requires more memory for storing the serialized documents and so generally also requires an increase in the JVM maximum heap size. The maximum JVM heap size should not exceed 2GB.

A heavily-loaded system that is processing a large number of documents can rapidly saturate the JVM heap memory. To avoid an `OutOfMemoryError`, the JVM maximum heap size must be increased by an amount corresponding to the size of the inline documents multiplied by the number of documents that are typically executed at any given time.

JVM maximum heap size increase = (inline documents size) x (average number of documents processed)

Example G.1 Calculating the JVM maximum heap size

In this example, the current JVM maximum heap is set to 512 MB and the maximum inline size is 64 KB. The server needs to be configured for the scenario where 10 jobs are run simultaneously, and each job has 9 input files and 1 result file (a total of 10 files per job, and 100 files processed simultaneously). All of the files are under 512 KB in size.

To store all of the files inline, the maximum inline size must be set to at least 512 KB.

The required increase in the JVM maximum heap size is calculated using the following equation:

$$(512 \text{ KB}) \times (100) = 51200 \text{ KB, or } 512 \text{ MB}$$

The JVM maximum heap size must be increased by 512 MB for a total of 1GB.

Considering heap fragmentation

Setting the size of inline documents to large values raises the risk of an `OutOfMemoryError` on systems that are prone to heap fragmentation. To store a document inline, there must be sufficient contiguous space in the JVM heap memory. Some operating systems, JVMs, and garbage collection algorithms are prone to heap fragmentation. Fragmentation decreases the amount of contiguous heap space, and can lead to an `OutOfMemoryError` even when sufficient total free space exists.

For example, previous operations on the application server have left the JVM heap in a fragmented state, and the garbage collector is unable to compact the heap sufficiently to regain large blocks of free space. An `OutOfMemoryError` can occur even though the JVM maximum heap size has been adjusted for an increase in maximum inline size.

To account for heap fragmentation, the inline document size must not be set higher than 0.1% of the total heap size. For example, a JVM maximum heap size of 512 MB can support a maximum inline size of $512 \text{ MB} \times 0.001 = 0.512 \text{ MB}$, or 512 KB.

Cleaning up temporary files in the Global storage directory

When a workflow process is executed, temporary files are placed in the Global storage directory, but are not deleted when the process is complete. These files are placed under subdirectories with the name, `Session/[NNNN]`, where `NNNN` is the workflow process ID. To prevent running out of disk space, you must regularly remove the Session directories associated with fully completed processes.

If the Global storage directory is not set explicitly in Configuration Manager, the default location of the Global storage directory is `[TempDir]/AdobeDocumentStorage/global`. If `[TempDir]` is also not specified by the user in Configuration Manager, the default location is `java.io.tmpdir`.

DB2 configuration settings

If you are running LiveCycle Form Manager with a DB2 database and the computer stops responding, check the server log files for deadlock-related messages. If you see such messages in the log files, change your DB2 configuration parameters. Set the `LOCKTIMEOUT` parameter to 15, and double the values for the `APPLHEAPSZ`, `STMHEAP`, and `SORTHEAP` parameters. You must then restart the database and application server.

Improving Windows Server Performance with LDAP

Using connection pooling on the search connection can decrease the number of ports need by as much as 50%, since that connection always uses the same credentials for a given domain, and the context and related objects are closed explicitly.

► **To configure your Windows Server for connection pooling:**

1. Start the registry editor by clicking **Start** > **Run** and in the **Open** box type `regedit` and click **OK**.
2. Navigate to the registry key:
`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters`
3. In the right pane of the registry editor, look for the **TcpTimedWaitDelay** value name. If the name does not appear, you can add it by selecting **Edit** > **New** > **DWORD Value** from the menu bar.
4. In the **name** box, type `TcpTimedWaitDelay`.
Note: If you do not see the a flashing cursor and `New Value #` inside the box, right-click inside the right panel, select **Rename** from the menu, then in the **name** box, type `TcpTimedWaitDelay`.
5. Repeat this step 4 for the following value names: `MaxUserPort`, `MaxHashTableSize`, and `MaxFreeTcbs`.
6. Double-click inside the right pane to set the **TcpTimedWaitDelay** value. Under **Base**, select **Decimal**, and in the **Value** box, type `30`.
7. Double-click inside the right pane to set the **MaxUserPort** value. Under **Base**, select **Decimal**, and in the **Value** box, type `65534`.
8. Double-click inside the right pane to set the **MaxHashTableSize** value. Under **Base**, select **Decimal**, and in the **Value** box, type `65536`.
9. Double-click inside the right pane to set the **MaxFreeTcbs** value. Under **Base**, select **Decimal**, and in the **Value** box, type `16000`.

Caution: Serious problems might occur if you modify the registry incorrectly by using Registry Editor or by using another method. These problems might require that you reinstall your operating system. Modify the registry at your own risk.

Index

A

- accessing
 - Administrator 25
 - BAM Workbench and BAM Dashboard 91
 - Installation Verification Sample application 27
 - LiveCycle database tables 42
 - LiveCycle Form Manager end-user pages 30
 - repository 114
 - User Management 26
- action properties, Assembler QPAC 107
- adding
 - fonts for LiveCycle Forms to use 50
- Adobe Administrator, accessing 25
- Adobe Form Server, upgrading to LiveCycle Forms 120
- Adobe LiveCycle Assembler
 - deployment, verifying 26
 - installing 36
 - invoking 105
 - jobs, submitting for processing 113
 - upgrading 127
- Adobe LiveCycle database
 - about initializing 11
 - connecting JBoss to 57
 - connecting to 81
 - creating 41
 - initializing 65
 - tables, accessing 42
- Adobe LiveCycle Document Security
 - upgrading 134
- Adobe LiveCycle Form Manager
 - accessing end-user pages 30
 - installing 36
 - upgrading 124
 - using WebDAV with 114
- Adobe LiveCycle Forms
 - deployment, verifying 27
 - installing 36
 - PDF/A compliancy 115
 - upgrading 120
- Adobe LiveCycle PDF Generator
 - additional requirements for 15
 - conversion time-out, setting 31
 - installing 34
 - JMS database, creating 45
 - post-deployment 30
 - running turnkey installation 22
 - upgrading 132
- Adobe LiveCycle Policy Server
 - upgrading 140
- Adobe LiveCycle Print
 - deployment, verifying 27
 - installing 38
 - upgrading 120
- Adobe LiveCycle products
 - developing forms for 114
 - documentation resources 9
 - information updates 10
 - manual installation 34
 - upgrading to 7.2 or 7.2.1 118
- Adobe LiveCycle Reader Extensions
 - upgrading 137
- Adobe LiveCycle services, modifying 24
- Adobe LiveCycle Workflow
 - configuring for BAM Server 90
 - installing 36
 - invoking LiveCycle Assembler through 105
 - metadata definitions, importing 92
 - upgrading 127
- Adobe LiveCycle Workflow Business Activity Monitor. *See* BAM entries
- Adobe LiveCycle Workflow Designer
 - configuration file, modifying 99
 - installing 98
 - JNDI and HTTP ports 99
 - uninstalling 99
 - upgrading 129
- Adobe PDF Printer, setting as default 30
- Adobe User Management
 - accessing 26
 - configuring with LDAP 67
- adobe-printSubmitter.ear file 28
- application servers
 - configuring 52
 - connecting to 99
 - deploying to JBoss 62
 - supported 13
 - updating 118
- Assembler QPAC
 - about 106
 - creating workflow process 107
- authentication
 - using LDAP 67, 94
 - using SSL credential 70
- automatic installation
 - using to upgrade LiveCycle products 119

B

- BAM Dashboard, accessing 91
- BAM metadata database
 - connecting JBoss to 77
 - connecting to on SQL Server 78
 - creating 75
- BAM Server
 - about upgrade utility 130
 - configuring 91
 - configuring JBoss for 77
 - configuring LiveCycle Workflow for 90

- connecting JBoss to database for 77
- deploying to JBoss 88
- installing database drivers for 77
- synchronizing with LDAP server 94, 97
- upgrading for LiveCycle Workflow 130
- user account 75

BAM Workbench, accessing 91

Business Activity Monitor. *See* BAM entries

C

checklists 16

cluster-service.xml file, modifying 56

configuration checklists 16

configuration files

- LiveCycle Workflow Designer, modifying 99

configuration files, JBoss, modifying 53

Configuration Manager

- and directory setting 142
- using during upgrade process 119
- using to configure LiveCycle products 48

configuration properties, retrieving for upgrading 121

configuring

- See also* installing
- about 11
- application server 52
- BAM Server 91
- DB2 data source for JBoss 59
- DB2 database for concurrent usage 44
- JBoss for BAM Server 77
- JBoss for Watched Folder 57
- JBoss JVM 83
- job sources 30
- LiveCycle products for deployment 48
- LiveCycle Workflow for BAM Server 90
- MySQL data source for JBoss 57
- Oracle data source for JBoss 58
- SQL Server data source for JBoss 60
- SSL on JBoss 70, 72
- User Management with LDAP 67
- watched folders 109
- Windows Server for connection pooling 143
- workflow process with Assembler QPAC 107

connecting

- JBoss to BAM metadata database 77
- JBoss to LiveCycle database 57
- to application servers 99
- to BAM metadata database on SQL Server 78
- to LiveCycle database 81

connection pool, applying patch for Watched Folder 64

conventions, path name 8

conversion time-out, LiveCycle PDF Generator, setting 31

CreateDocumentList QPAC 106

creating

- data source files 57, 77
- databases 41, 75
- JobConfig.xml file 110
- SSL credential 70
- watched folders 109

credentials, SSL, creating 70

D

data source files, creating 57, 77

databases

- connecting JBoss to 77
- connecting to 81
- creating 41, 75
- creating data source files for 57
- drivers supported 14
- drivers, installing for BAM Server 77
- initializing 65
- operating system supported 14, 101
- supported 14

DB2 database

- configuration settings 142
- configuring for concurrent usage 44
- creating 43
- creating data source for JBoss 59

DDX files

- pattern matching 112
- workflow use of 106

deploying

- about 11, 62
- BAM Server to JBoss 88
- LiveCycle Assembler sample 26
- multiple LiveCycle products 12
- PrintIVS web application 28
- to JBoss 63

deployment

- checklists 16
- configuring LiveCycle products for 48

directories

- Global storage 49, 142
- installed with turnkey 19
- JBoss naming convention 62

document transfer performance, increasing 141

documentation resources 9

dom4j.jar file, updating 56

dynamic DDX files 106

dynamic forms, button support for 115

E

EAR file, setting class-loading 55

endorsed directory, creating 53

error logging, Assembler QPAC workflow 108

error logs, viewing 40

F

fail modes, Assembler QPAC workflow 108

files

- accessing using WebDAV clients 114
- deployable 62

fonts

- bundled 104
- embedding in PDF/A-compliant forms 115
- installing for converted documents 31
- selecting for LiveCycle Forms to use 50

Form Server Module API Print application, running 29

forms, designing 114

G

- Global storage directory
 - about 49
 - temporary files 142

H

- hardware requirements 15
- HTTP ports, LiveCycle Workflow Designer 99

I

- importing LiveCycle Workflow metadata definitions 92
- initializing database 11, 65
- installation
 - checklists 16
 - directories 19
 - methods for 12
- installing
 - See also* configuring
 - about 11
 - applications to JBoss 63
 - database drivers for BAM Server 77
 - fonts used in converted documents 31
 - JBoss 47, 75
 - LiveCycle products for manual deployment 36
 - LiveCycle Workflow Designer 98
 - multiple LiveCycle products 12
 - using manual method 34
 - using turnkey method 19

J

- jacorb.properties file, modifying 55
- JAR files, updating dom4.jar 56

JBoss

- configuration files, modifying 53
 - configuring for BAM Server 77
 - configuring for Watched Folder 57
 - configuring SSL on 70, 72
 - connecting to database for BAM Server 77
 - connecting to LiveCycle database 57
 - deployable components 62
 - deploying BAM Server to 88
 - deploying to 63
 - installation directory 19
 - installing 47, 75
 - message-recording restrictions 85
 - naming for home directory 62
 - port numbers, customizing 86
 - starting and stopping 52
 - thread configuration, modifying 85
- JBoss service, uninstalling 116
- JDBC driver
- copying for DB2 60
 - copying for MySQL 57
 - copying for Oracle 59
 - copying for SQL Server 61
- JDK support 13
- JMS

- configuration, modifying 54
- database, creating 45
- JNDI ports, LiveCycle Workflow Designer 99
- job sources, configuring 30
- JobConfig.xml file, creating for watched folders 110
- JVM
 - configuring 83
 - heap size, maximum, increasing 141

L

- LDAP
- configuring for BAM Server 93
 - configuring User Management with 67
 - improving Windows Server performance with 143
 - server support 14
- LiveCycle. *See* Adobe LiveCycle
- log files, viewing 40, 64

M

- manual installation
- about 12
 - checklist 17
 - removing 117
 - running 34
- multiple LiveCycle products, installing and deploying 12
- MySQL
- configuring data source for JBoss 57
 - connecting to LiveCycle database on 82
 - database, creating 41
 - installation directory 19
 - service, uninstalling 116
 - user account 42

N

- naming conventions, file path 8

O

- operating systems
- database support 14, 101
 - supported 13
- Oracle database
- creating 42
 - creating data source for JBoss 58

P

- PDF files, preparing 112
- PDF/A compliant forms, embedding fonts in 115
- performance, Windows Server, improving 143
- port numbers, customizing for JBoss 86
- Print Submitter application, running 28
- PrintExample BAT and SH files 29
- PrintIVS web application, deploying and running 28
- processing LiveCycle Assembler jobs 113
- product files
 - deployable 62
 - removing 116, 117

Q

- QPACs
 - creating 10
 - deploying 106

R

- removing. *See* uninstalling
- repository, accessing using WebDAV clients 114
- role mapping, LDAP, configuring 96
- run files, modifying 53

S

- samples
 - data files 28
 - form designs 28
 - Form Server Module API Print, running 29
 - Installation Verification Sample, accessing 27
 - LiveCycle Assembler, deploying 26
 - Print Submitter, running 28
 - PrintIVS web application, running 28
- SMTP settings, configuring for BAM Server 91
- software
 - client, publishing files from 114
 - combinations for installation 14
 - requirements 13
- SQL Server
 - configuring data source for JBoss 60
 - connecting BAM Server connecting to 78
 - connecting to LiveCycle database on 81
 - creating JMS database on 45
 - database, creating 45
 - installing driver for 77
 - JDBC driver 61
- SSL
 - configuring on JBoss 70, 72
 - credential, creating 70
- starting
 - Administrator 25
 - installation program 36
 - JBoss 52
 - User Management 26
- static DDX files 106
- stopping JBoss 52
- submitter.bat file 28
- submitting LiveCycle Assembler jobs for processing 113
- synchronizing BAM Server with LDAP server 94, 97
- system requirements 13

T

- thread configuration, JBoss, modifying 85
- thread settings, maximum, modifying 55
- turnkey installation
 - about 12
 - checklist 16
 - running 19
 - uninstalling 116
 - using to upgrade LiveCycle products 119

U

- uninstalling
 - LiveCycle Workflow Designer 99
 - product files 116, 117
- upgrading LiveCycle products 12, 118
- user accounts
 - BAM Server 75
 - MySQL 42
- User Management. *See* Adobe User Management
- user mapping, LDAP, configuring 95

V

- variables, Assembler QPAC 107
- verifying
 - LiveCycle Assembler deployment 26
 - LiveCycle Forms deployment 27
 - LiveCycle Print deployment 27
- viewing
 - log files 40
- viewing log files 64

W

- Watched Folder
 - about installation of 12
 - applying connection pool patch 64
 - configuring JBoss for 57
 - installing 39
 - invoking LiveCycle Assembler with 105
 - upgrading 127
- watched folders
 - creating and configuring 109
 - creating JobConfig.xml file for 110
- web browser support 13
- WebDAV clients 114
- workflow processes, creating 106

X

- XML file, job configuration 110