# Configuring LiveCycle® ES Application Server Clusters Using WebLogic®

**Adobe® LiveCycle® ES**

# Contents

# Preface

This document is one of several resources available to help you learn about Adobe® LiveCycle® ES (Enterprise Suite). LiveCycle ES is a flexible, extensible platform that helps automate and accelerate the flow of business-critical information to and from customers, partners, constituents, and employees.

## What's in this document?

This document provides information about how to install and configure the following solution components in a clustered environment on Microsoft® Windows®, Linux®, and Sun™ Solaris™, and how to deploy the solution components to BEA WebLogic Server®:

- Adobe LiveCycle Barcoded Forms ES
- Adobe LiveCycle Business Activity Monitoring ES
- Adobe LiveCycle Digital Signatures ES
- Adobe LiveCycle ES Connector for EMC Documentum
- Adobe LiveCycle ES Connector for IBM FileNet
- Adobe LiveCycle Forms ES
- Adobe LiveCycle Foundation
- Adobe LiveCycle Output ES
- Adobe LiveCycle PDF Generator ES
- Adobe LiveCycle Process Management ES
- Adobe LiveCycle Reader Extensions ES
- Adobe LiveCycle Rights Management ES

## Who should read this document?

This document provides information for administrators or developers responsible for installing, configuring, administering, or deploying LiveCycle ES components in a clustered environment. The information provided is based on the assumption that anyone reading this guide is familiar with Java 2 Platform, Enterprise Edition (J2EE) application servers, Linux, Windows, or Solaris operating systems, 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 |
|------|---------------|-------------|
| *[LiveCycleES root]* | Windows: C:\Adobe\LiveCycle8\ <br><br> Linux and UNIX: /opt/adobe/livecycle8/ | The installation directory that is used for all LiveCycle ES solution components. The installation directory contains subdirectories for LiveCycle Configuration Manager and the LiveCycle ES SDK. |
| *[appserver root]* | WebLogic Server on Windows: C:\bea\weblogic92\ <br><br> WebLogic Server on UNIX /usr/bea/weblogic92 | The home directory of the application server that runs the LiveCycle ES services. |
| *[server name]* | server1 for Weblogic | The name of the server configured on your application server. |
| *[BEA_HOME]* | WebLogic Server on Windows: C:\bea <br><br> WebLogic Server on Linux and UNIX: /usr/local/bea | The install directory for WebLogic Server as specified for the BEA_HOME environment variable. |
| *[appserverdomain]* | WebLogic Server on Windows: C:\bea\user_projects\domains\ *[domain name]* <br><br> WebLogic Server on Linux and UNIX: *[BEA_HOME]*/user_projects/ domains/*[domain name]* | The domain that you configured on WebLogic Server. |
| *[APACHE_HOME]* | The location where Apache is installed. | The directory to which the Apache HTTP Server is installed. This is applicable only if you intend to use Apache as a load balancer. |
| *[dbserver root]* | The location where the LiveCycle ES database server is installed. | Depends on the database type and your specification during installation. |

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

## Additional information

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

| For information about | See |
| --- | --- |
| Preparing to install LiveCycle ES | *Preparing to Install LiveCycle ES* |
| Upgrading from LiveCycle 7.x to LiveCycle ES | *Preparing to Upgrade to LiveCycle ES* |
| Performing administrative tasks for LiveCycle ES | *Administering LiveCycle ES* |
| Installing LiveCycle Workbench ES | *Installing Your Development Environment* |
| Other services and products that integrate with LiveCycle ES | www.adobe.com |
| Patch updates, technical notes, and additional information on this product version | www.adobe.com/support |

# 1 | Introduction

This section describes clustered application servers and the benefits and issues associated with setting up clusters.

## About clustering application servers

A *cluster* is a group of application server instances running simultaneously, which act like a single system, enabling high availability and load balancing. Within a cluster, multiple server instances can run on the same computer (known as a *vertical cluster*) or can be located on different computers (known as a *horizontal cluster*), or they can form a combination of both horizontal and vertical clusters. With clustering, client work can be distributed across several nodes instead of being handled by a single application server.
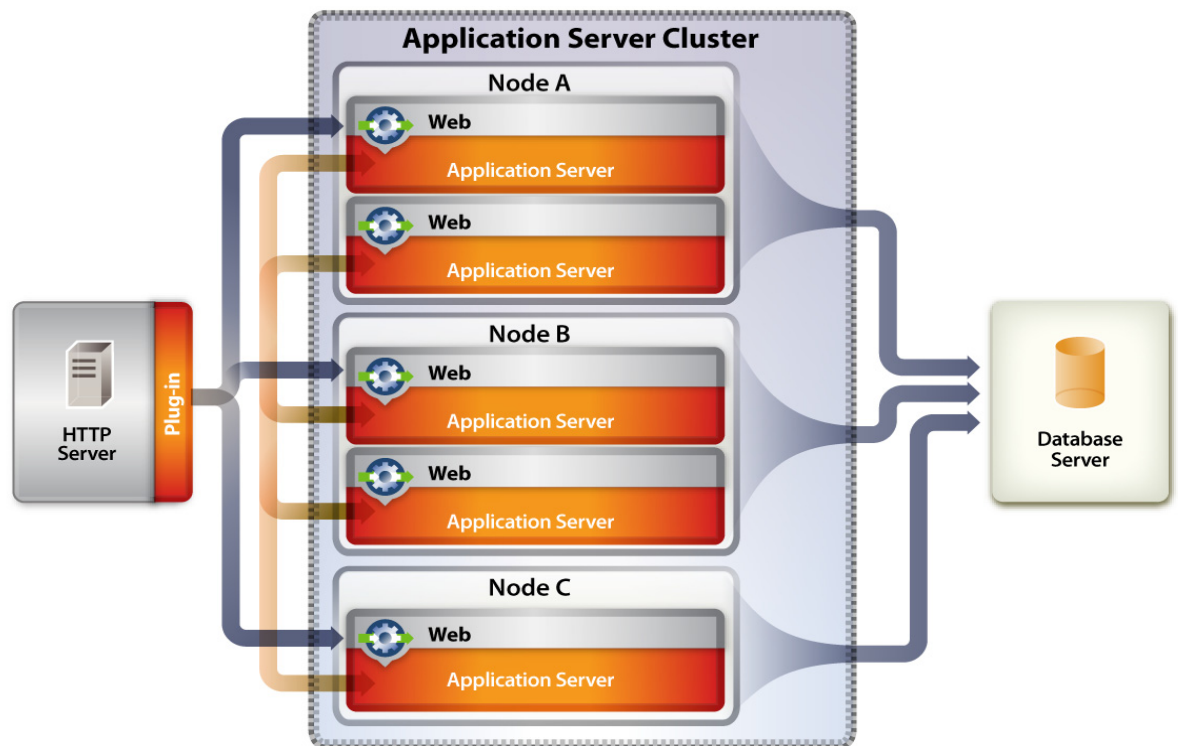


In a clustered configuration, application server instances are server members of the cluster, all of which must have identical application components deployed on them. However, other than the configured applications, cluster members do not have to share any other configuration parameters. For example, you can cluster multiple server instances on one computer, with a single instance on another computer, provided they are all running BEA WebLogic Server® 9.2.

By clustering, you can achieve one or more of the following benefits. How you implement clustering determines which benefits are achieved:

- Failover
- Load balancing
- Scalability

# Failover

*Failover* allows one or more application server instances to act as backup to a failed application server instance and resume processing the task, thereby enabling another application server to carry on processing. However, if an application server fails during a transaction, the backup application server does not recover the state of the failed instance. If a server fails when a user enters data into a form, for example, the data may have to be reentered.

# Load balancing

*Load balancing* is a technique used to distribute work across a number of systems so that no single device is overwhelmed. If one server starts to get congested or overloaded, requests are forwarded to another server that has more capacity.

## Application server load balancing

*Application server load balancing* is useful for managing the load between application servers. Application servers can be configured to use a weighted round-robin routing policy that ensures a balanced routing distribution based on the set of server weights that have been assigned to the members of a cluster. Configuring all servers in the cluster to have the same weight produces a load distribution in which all servers receive approximately the same number of requests. Weighting some servers more heavily sends more requests to these servers than to those that are weighted less heavily.

Preferred routing configurations can also be configured to ensure, for example, that only cluster members on that node are selected (using the round-robin weight method) and cluster members on remote nodes are selected only if a local server is not available. Application server load balancing is best used when balancing is needed between tiers.

## Web server load balancing

*Web server load balancing* is useful for queuing and throttling requests. For the Apache HTTP Server, the most commonly used method for load balancing is Round-Robin DNS.

*Round-Robin DNS* is a relatively simple method of load balancing, where a domain name system (DNS) server provides a name to address resolution and is always involved when a host name is included in a URL. A Round-Robin DNS server can resolve one single host name into multiple IP addresses, such that requests for a single URL (containing a host name) actually reference different web servers. The client requests a name resolution for the host name but, in fact, receives different IP addresses, therefore spreading the load among the web servers. In a simple configuration, the Round-Robin DNS server cycles through the list of available servers.

# Scalability and availability

*Scalability* in a cluster means that an administrator can increase the capacity of the application dynamically to meet the demand without interrupting or negatively impacting service. WebLogic clusters allow administrators to remove nodes from a cluster in order to upgrade components, such as memory, or to add nodes to the cluster without bringing down the cluster itself.

# Terminology

WebLogic uses specific terminology, which is defined here to avoid confusion:

**server:** Represents an instance of a Java™ virtual machine (JVM).

**machine:** Represents a physical system running one or more instances of WebLogic Server.

**cluster:** Represents a logical grouping of multiple application servers within a domain for administration, application deployment, load balancing, and failover purposes.

# Clustering LiveCycle products

If you install a LiveCycle ES product on an application server cluster, here are some things you must know:

- LiveCycle ES must be clustered by using a homogeneous topology (all nodes in the cluster must be configured identically) on each application server it is deployed to. You can ensure that all modules are configured identically by configuring run-time properties in the single-installation staging area.

- The configuration is deployed using the single entity approach; all nodes in a cluster are deployed as if deploying to a single node.

Setting up a clustered environment for LiveCycle ES involves these tasks:

1. Preparing the hardware as per your requirements

2. Installing the operating system and updating with all necessary patches and service packs

3. Installing and configuring the database server

4. Installing and configuring the application server

5. Configuring the JMS resources

6. Installing LiveCycle ES solution components and configuring LiveCycle ES for deployment

7. Deploying LiveCycle ES to WebLogic

# Supported topologies

The following sections discuss various topologies, both clustered and non-clustered, that you can employ. For additional information about configuring WebLogic in a cluster, see edocs.bea.com/platform/suppconfigs/configs92/92_over/overview.html.

## Combined web, application, and database servers

This topology consists of a web server, an application server, and a database server on the same node. This topology is the simplest one and must be used for development only.

## Combined web and application servers with separate database server

This topology can be considered for production in case the load on the user interface (including the web tier) is minimal, with a small number of users.

Combining the web and application servers means that all Enterprise JavaBeans™ (EJB) look-ups are local, and therefore reduces the overhead of doing a remote look-up. Also, this topology reduces the network overhead of a round trip between the web tier and the application tier.

However, with both servers on the same node, if the web tier is compromised, both tiers are compromised. If the web tier experiences a heavy load, the application server processing is affected and vice versa. User response time is usually affected in situations when users need to wait a significant amount of time to get a page back due to all server resources (that is, CPU and/or memory) being consumed by the application server. If the web tier has a large session size, the application could be deprived of the memory required to process messages off the Java Message Service (JMS) layer.

## Single web server with combined application and database server

The simplest topology that should be considered for a production environment is a web server and combined application server with a database server. Use this topology only if you are sure that your database load will be minimal. In this scenario, the web server is providing a redirection to the application server. The advantages of this topology are low cost, low complexity, and no need for load balancing. The disadvantages of this topology are little redundancy, low scalability, inability to perform updates and upgrades, and possible low performance due to too many CPU processes.

## Separate web, application, and database servers

This topology is the most common in production systems because it allows allocation of separate resources to each of the tiers. In this case, the web server acts as a proxy to the web tier on the application server that hosts the web components. This level of indirection provides additional security by securing the application server even if the web server is compromised.

## Adding additional web servers

You can add additional web servers for scalability and failover. When using multiple web servers, the WebLogic HTTP plug-in configuration file must be applied to each web server. Failure to do so after introducing a new application will likely cause a 404 File Not Found error to occur when a user tries to access the web application.

## Adding additional application servers

This topology is used in most large-scale production systems where the application servers are clustered to provide high availability and, based on the topology, failover and load balancing.

Clustering application servers has these benefits:

- Allows you to use cheaper hardware configurations and still achieve higher performance
- Allows you to upgrade software on servers without downtime
- Provides higher availability (that is, if one server fails, the other nodes in the cluster pick up the processing)
- Provides the ability to leverage load balancing algorithms on the web server (by using load balancers) as well as on the EJB tier for processing requests

LiveCycle ES products are typically CPU-bound and, as a result, performance gains are better achieved by adding more application servers than by adding more memory or disk space to an existing server.

## Multiple JVMs

Vertical scaling of multiple JVMs offers the following advantages:

**Increased processing power efficiency:** An instance of an application server runs in a single JVM process. However, the inherent concurrency limitations of a JVM process prevent it from fully utilizing the memory and processing power of multi-CPU systems.  Creating additional JVM processes provides multiple thread pools, each corresponding to the JVM process associated with each application server process. This correspondence avoids concurrency limitations and lets the application server use the full processing power of the machine.

**Load balancing:** Vertical scaling topologies can use the WebLogic Server  workload management facility.

**Process failover:** A vertical scaling topology also provides failover support among application server cluster members. If one application server instance goes offline, the other instances on the machine continue to process client requests.

## JMS messaging

You can configure embedded JMS in various topologies:

- Single instance of WebLogic Server, with a single JMS server

- Multiple instances of WebLogic Server, with a single JMS server

- Multiple instances of WebLogic Server, each with their own JMS servers

A single JMS server in a single or multiserver environment is simple to configure and does not require JMS Server Management. However, it can only handle low volumes. Because the data store (where messages are stored) is not available for tuning and/or management, the single JMS server cannot be tuned as it is started, nor can it be managed by WebLogic Server.

# Unsupported topologies

The following topologies are not supported for LiveCycle ES.

### Splitting the web container/EJB container

Splitting LiveCycle ES servers into presentation/business logic tiers and running them on distributed computers is not supported.

### Geographically distributed configuration

Many applications locate their systems geographically to help distribute the load and provide an added level of redundancy. LiveCycle ES does not support this configuration because LiveCycle ES components cannot be pulled apart to run on different hosts; LiveCycle ES is deployed as a monolithic application.

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

# Third-party infrastructure support

This section summarizes the referenced platform and database combinations for BEA WebLogic Server 9.2.

| Operating system | Application server | JDK | Database [1] |
|---|---|---|---|
| Microsoft Windows Server® 2003, Enterprise Edition SP1, Standard Edition SP1, R2 <br><br> (32-bit architectures and VMWare ESX/GSX) | BEA WebLogic® 9.2, (Advantage & Premium) | BEA JRockit® JDK | - IBM DB2 8.2, 8.1 FixPack 7a <br> - Oracle 9i, 10g <br> - Microsoft SQL Server 2005 SP1 |
| Red Hat® Enterprise Linux AS or ES 4.0 <br><br> (32-bit architectures) <br><br> SUSE™ Linux Enterprise Server 9.0 <br><br> (32-bit architectures) | BEA WebLogic 9.2 (Advantage & Premium) | BEA JRockit JDK | - IBM DB2 V8.2, V8.1 FixPack 7a <br> - Oracle 9i, 10g <br> - Microsoft SQL Server 2005 SP1 |
| Solaris 9, 10 <br><br> (SPARC® Architectures) | BEA WebLogic 9.2 (Advantage & Premium) | Sun JDK supplied with application server | - Oracle 9i, 10g |

[1] For a detailed list of the supported database editions, see .

# System requirements

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

## Minimum hardware requirements

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

| Operating system | Minimum hardware requirement |
|---|---|
| Microsoft Windows Server 2003 Enterprise Edition or Standard Edition SP1 and R2 (32-bit edition only) | Intel® Pentium® 3 or x86 equivalent, 1 GHz processor<br><br>RAM - 2 GB<br><br>Free disk space - 2.7 GB of temporary space plus 1.7 GB for LiveCycle ES |
| Sun Solaris 9 and 10 | Sun UltraSPARC® IIe, 650 MHz processor<br><br>RAM - 4 GB<br><br>Free disk space - 2.7 GB of temporary space plus 1.7 GB for LiveCycle ES |
| SUSE Linux Enterprise Server 9.0 i386 (32-bit edition only) | Pentium 3 or x86 equivalent, 1 GHz processor<br><br>RAM - 2.5 GB<br><br>Free disk space - 2.7 GB of temporary space plus 1.7 GB for LiveCycle ES |
| Red Hat Enterprise Linux AS or ES 4.0 (32-bit edition only) | Pentium 3 or x86 equivalent, 1GHz processor<br><br>RAM - 2.5 GB<br><br>Free disk space - 2.7 GB of temporary space plus 1.7 GB for LiveCycle ES |

## Recommended hardware requirements

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

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

**Sun SPARC environments**: UltraSPARC III or later

**Memory requirements**: 4 GB of RAM

## Supported software

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

| Required software | Supported version |
|---|---|
| Operating system | • Microsoft Windows Server 2003 Enterprise Edition or Standard Edition SP1 and R2 (32-bit edition only) <br><br> • (Workbench ES) Windows XP Service Pack 2 <br><br> • Solaris 9 and 10 <br><br> • IBM AIX 5L 5.3 <br><br> • SUSE Linux Enterprise Server 9.0 (32-bit edition only) <br><br> • Red Hat Enterprise Linux AS or ES 4.0 (32-bit edition only) |
| Application server | • BEA WebLogic 9.2 (Advantage and Premium editions) <br><br> • |
| Web browser | For a complete list of web browsers, see "Web browser support" on page 17. |
| JDK | • WebLogic on all platforms except Solaris - BEA JRockit SDK 1.5.0_06 <br><br> • WebLogic on Solaris - Sun JDK 1.5.0_04. This version of the JDK requires the Daylight Savings Time (TZ) update to be run in North America and is located at http://java.sun.com/javase/tzupdater_README.html <br><br> For more information about JDKs, see "LDAP configuration" on page 21. |
| Database | • IBM DB2 8.2 or IBM DB2 8.1 Fix Pack 7 (Enterprise Edition) <br><br> • Microsoft SQL Server 2005 SP1 (Enterprise Edition) <br><br> • Oracle 9i or 10g (Standard and Enterprise Editions) |
| Database driver | • IBM DB2 - db2jcc.jar and db2jcc_license_cu.jar <br><br> • SQL Server 2005 JDBC 1.1 - sqljdbc.jar <br><br> • Oracle 9i/10g - ojdbc14.jar, Release 2 (10.2.0.2 Thin) <br><br> **Note:** All LiveCycle Business Activity Monitoring ES metadata databases use the drivers stated above. |
| LDAP server | • Sun ONE 5.1 <br><br> • Sun ONE 5.2 <br><br> • Microsoft Active Directory 2000 <br><br> • Microsoft Active Directory 2003 <br><br> • Novell® eDirectory 8.7 <br><br> • IBM Tivoli Directory Server 6.0 |
| Email servers | • Microsoft Exchange 2000 <br><br> • Microsoft Exchange 2003 <br><br> • Lotus Notes/Domino 6/7 <br><br> • SendMail (included with Red Hat 4 and SUSE 9) <br><br> • Novel GroupWise 6/7 |

| Required software | Supported version |
|---|---|
| Data Services ES | • Adobe Flex™ Builder™ 2.0.1 Hot Fix 2 or higher<br>or<br>• Flex SDK 2.0.1 Hot Fix 2 or higher |
| LiveCycle ES connector | • IBM FileNet P8 Content Engine 3.5.1<br>• EMC Documentum Content Server 5.3 SP4 |
| PDF client | • Adobe Acrobat® Professional and Acrobat Standard, versions 7.0 to 8.1 (for securing documents with policies and opening policy-protected documents)<br>• Adobe Reader®, versions 7.0 to 8.1 (for opening policy-protected documents) |

## Web browser support

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

### End-user interface

End-user components include these solution components:

- Adobe LiveCycle Workspace ES (Adobe Flash® required)
- Adobe LiveCycle Reader Extensions ES (Adobe Flash required)
- Adobe LiveCycle Rights Management ES (browser only)
- Adobe LiveCycle PDF Generator ES (browser only)

| Operating system | Flash Player | Supported browser |
|---|---|---|
| Microsoft Windows Vista™ | Flash Player 9* | Microsoft Internet Explorer 7 or later[1]<br>Firefox 2.0.0.1 or later[1] |
| Windows 2000 | Flash Player 9* | Internet Explorer 6 or later[1]<br>Firefox 2.0.0.1 or later[1] |
| Windows XP | Flash Player 9* | Internet Explorer 6 or later[1]<br>Firefox 2.0.0.1 or later[1] |
| Windows Server 2003 | Flash Player 9* | Internet Explorer 6 or later[1]<br>Firefox 2.0.0.1 or later[1] |
| OS X v 10.4.x (PowerPC) | Flash Player 9* | Firefox 2.0.0.1 or later[1]<br>Safari 2.x or later (not for Workspace ES)[2] |
| OS X v 10.4.x (Intel) | Flash Player 9* | Firefox 2.0.0.1 or later[1]<br>Safari 2.x or later (not for Workspace ES)[2] |

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

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

### Administrator user interface

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

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

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

## Additional requirements for Linux or UNIX operating systems

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

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

## Additional requirements for PDF Generator ES

**Note:** This section applies only if your configuration requires native file format conversion (for example, Microsoft Word to PDF conversion). These requirements apply to all nodes in the cluster.

## User account

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

- Installing Microsoft Office

- Installing LiveCycle PDF Generator ES

- Installing Acrobat 8.1 for PDF Generator ES

- Running the application server process

## Acrobat

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

## Native file conversion software installation

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

- Microsoft Office 2000, XP, 2003, or 2007 (DOC, XLS, PPT)

- Microsoft Office Visio 2003 (VSD)

- Microsoft Project 2003 (MPP)

- AutoCAD 2005 (DWG)

- Corel WordPerfect 12 (WPD)

- Adobe Photoshop® CS2 (PSD)

- Adobe FrameMaker® 7.2 (FM)

- Adobe PageMaker® 7.0 (PMD, PM6, P65, PM)

- OpenOffice 2.1 (all associated formats)

**Note:** After installing any of these software on server; launch the software manually to dismiss any startup dialogs.

## Windows environment variables

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

The names of these environment variables are listed below:

- Acrobat_PATH

- Notepad_PATH

- Photoshop_PATH

- WordPerfect_PATH

- PageMaker_PATH

- FrameMaker_PATH

● OpenOffice_PATH

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

For example, the variable `Photoshop_PATH` might contain the value `D:\Program Files\Adobe\Adobe Photoshop CS2\Photoshop.exe`. However, `OpenOffice_PATH` is different from others. This variable must be set to the OpenOffice installation folder (instead of the path to the executable). A typical value of `OpenOffice_PATH` on Windows would be `C:\Program Files\OpenOffice.org 2.1`.

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

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

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

2. Click the **Advanced** tab and click **Environment Variables**.

3. Under System Variables are, click **New**.

4. Enter the environment variable name you need to set (for example, `Photoshop_PATH`). This directory is the one that contains the executable file. For example, type the following code:

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

## Service Control Manager command line tool

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

## Additional requirements for Business Activity Monitoring ES

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

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

# LDAP configuration

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

Install and configure your LDAP server and database according to the vendor's documentation. See "Supported software" on page 15 for a list of supported LDAP servers. Take note of the LDAP administrator name and password to use during the LiveCycle ES configuration process. You configure LiveCycle ES to connect with the LDAP database after you have installed and deployed your LiveCycle ES services. This configuration is done using User Manager. See the *Installing and Deploying LiveCycle ES* document for your application server, available at http://www.adobe.com/support/documentation/en/livecycle/es/.

# Installation considerations

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

## Installing from network drives

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

## Manual use of Acrobat restricted

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

# 2 LiveCycle ES Credentials and Certificates

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

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

## Obtaining the Reader Extensions ES Rights credential

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

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

The following types of Rights credentials are available:

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

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

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

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

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

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

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

# Obtaining digital certificates for use with Digital Signatures ES

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

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

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

**Note:** You can configure and manage certificates, credentials, and certificate revocation lists (CRLs) for use with LiveCycle ES using Trust Store Management, which is accessible through LiveCycle Administration Console. (For more information see *Trust Store Management Help* at http://www.adobe.com/go/learn_lc_adminTrustStore.)

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

The following file types are supported:

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

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

**CRLs**: Base64 and DER-encoded .crl files

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

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

# 3 | Creating the LiveCycle ES Database

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

The database will contain these elements:

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

Before creating the database, you must ensure that you have read the preinstallation requirements and that you have the required software installed. (See "System requirements" on page 14.)

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

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

## Creating an Oracle database

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

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

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

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

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

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

- SID (Service ID)

- User name and password of the Oracle user account

- Host name or IP address of database server

- Oracle port number (default port = 1521)

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

# Creating a DB2 database

Create a DB2 database by running the script provided in this section. The script is tuned for a system that will use 1 GB of memory for the database. If your system has a smaller amount of memory dedicated for the database, see the DB2 documentation for details on configuring your system settings. LiveCycle ES supports DB2 with WebLogic Server 9.2.

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

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

- Database name

- User name and password of the DB2 user account

- Host name or IP address of database server

- DB2 port number (default port = 50000)

➤ **To create a DB2 database:**

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

   **Note:** Use the appropriate path separator for your operating system.  The sample below shows a Windows path separator (\). Use the forward slash (/) on UNIX and Linux.

   ```
   CREATE DATABASE DBNAME USING CODESET UTF-8 TERRITORY DEFAULT;
   CONNECT TO DBNAME;
   CREATE BUFFERPOOL "DBNAME_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_HOME\DBNAME_TEMP') EXTENTSIZE 32 PREFETCHSIZE 16 BUFFERPOOL
   DBNAME_BP8K;
   CREATE REGULAR TABLESPACE DBNAME_DATA_8K IN DATABASE PARTITION GROUP
   IBMDEFAULTGROUP PAGESIZE 8192 MANAGED BY DATABASE USING
   (FILE'DB2_HOME\DBNAME_DATA'64000) EXTENTSIZE 16 PREFETCHSIZE 16
   BUFFERPOOL DBNAME_BP8K;
   DROP TABLESPACE USERSPACE1;
   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* with the name you want for the LiveCycle ES database.

   - Replace *DB2_HOME* with the path to the root directory where DB2 is installed. Ensure that the path separator is appropriate to your operating system. If DB2 is on a UNIX or Linux system, ensure that the database creator/user has write access to the DB2_HOME directory.

   - Ensure that commands do not include line breaks and that each command is terminated by a semicolon (;).

   - Change `64000` in the following line based on your database size:

     `(FILE'DB2_HOME\DBNAME_DATA'64000)`

     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. Ensure that the path separator is appropriate to your operating system.

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

4. Open a command prompt (or use the DB2 Command Editor).

5. Enter the following command to run the script (or copy and paste the contents into the DB2 Command Editor Commands window):

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

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

1. Log in to the IBM Control Center.

2. Expand your **db** > **User and Group Objects** > **DB Users** and select **Add User**.

3. Select the user you want to use.

4. Under **Authorities**, choose **Connect to database** and **Create tables**.

5. Click **OK**.

# Creating a SQL Server database

You can create a SQL Server database that LiveCycle ES will use to store run-time and configuration data. For information, see the SQL Server documentation. LiveCycle ES supports SQL Server 2005 SP1 with WebLogic Server 9.2.

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, see the SQL Server documentation.

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

- Database name
- User name and password of the SQL Server user account
- Host name or IP address of database server
- SQL Server port number (default port = 1443)

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

## Setting up SQL Server for LiveCycle ES

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

### Increasing memory

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

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

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

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

3. Select the **Memory** page and set **Minimum Server Memory (in MB)** to as large a size as you have free memory on the server.

### Setting processor priority

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

➤ **To set the processor priority:**

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

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

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

### Increasing the recovery interval

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

➤ **To increase the recovery interval:**

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

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

3. Select the **Database Settings** page and set **Recovery Interval (Minutes)** to 5 minutes.

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

## Sizing your SQL Server database

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

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

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

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

## Creating the LiveCycle ES database

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

**Note:** In the following procedure, *database_name* represents the name you specified when you created your database.

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

1. Expand the **Server** connection, right-click the **Databases** folder and select **New Database**.

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

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

4. Set **Database Data Initial Size MB** to the appropriate value:

   ● For small development or small production system: 200 MB.

- For larger systems: See "Sizing your SQL Server database" on page 28.

5. Set **Database Data Autogrowth** to 50%.

6. Set **Database Log Initial Size** to the appropriate value:

   - For small development or small production system: 20 MB

   - For larger systems: See "Sizing your SQL Server database" on page 28.

7. Set **Database Log Autogrowth** to 50%.

## Creating the LiveCycle ES user in SQL Server

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

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

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

2. Expand the **Server** connection and the **Security** folder, and right-click **Logins** and select **New Login**.

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

4. Select the option for **SQL Server Authentication** and specify a new password.

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

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

7. Expand the **Server** connection and the **Databases** folder, expand *database_name*, expand the **Security** folder, and right-click **Schemas** and select **New Schema**.

8. Enter the **Schema Name** *database_username* and click **OK**.

9. Expand the **Server** connection and the **Databases** folder, expand *database_name*, expand the **Security** folder, and right-click **Users** and select **New User**.

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

11. Set the default schema to `database_username`.

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

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

## Associating the LiveCycle ES user with the database

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

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

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

1. Expand the **Security** folder and the **Logins** subfolder, right-click ***database_username*** and select **Properties**.

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

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

4. Ensure that ***database_name*** is selected in the Users Mapped To This Login table.

5. In the Database Role Membership For *database_name* table, ensure that **db_owner** and **public** are selected.

## Setting the isolation level for the LiveCycle ES database

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

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

➤ **To set the isolation level:**

1. Expand the **Databases** folder, and right-click ***database_name*** and select **New Query**.

2. In the Query pane, enter this text:

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

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

# 4 | Creating a WebLogic Server Cluster

You can install WebLogic Server on each computer in the cluster or you can install the application server on a single computer. The Administration Server should be installed on one of the servers but not necessarily on one of the servers in the cluster. For details about installing WebLogic Server, see http://e-docs.bea.com/common/docs92/install/index.html.

You must perform the following tasks to install and configure WebLogic Server in a cluster environment:

- Synchronize the clock times on the cluster computers. (See "Installing WebLogic Server" on page 32.)
- Install the WebLogic Server software. (See "Installing WebLogic Server" on page 32.)
- Create the WebLogic domain. (See "Creating a WebLogic domain" on page 32.)
- Configure the WebLogic Server cluster. (See "Configuring the WebLogic Server cluster" on page 33.)

## Preparing to install

Before installing WebLogic Server, ensure that your system meets the following configuration requirements:

**Disk space:** Ensure that the partition that will hold the application server has a minimum of 10 GB of free disk space. In addition to the space required to install the product, your environment variable `TEMP` or `TMP` must point to a valid temporary directory with at least 500 MB of free disk space. The downloadable executable requires approximately 500 MB, plus an additional 1.0 GB to unpack the images.

**IP address settings:** All of the computers must have a fixed IP address managed through a single DNS.

**IP multicast:** All of the computers must fully support IP multicast packet propagation, which means that all routers and other tunneling technologies must be configured to propagate multicast messages to clustered server instances. The network latency must be low enough to ensure that most multicast messages reach their final destination within 200 to 300 milliseconds. Finally, the multicast time-to-live (TTL) value for the cluster must be high enough to ensure that routers do not discard multicast packets before they reach their final destination.

**Versions:** All of the computers in the cluster must have the same version and same service pack of WebLogic Server software.

**Horizontal clustering:** If your configuration is horizontally clustered (that is, instances of WebLogic Server are installed on separate computers), ensure that all computers are on the same network subnet and that the computer clocks are synchronized.

**Account privileges:** (Windows) You must install and run WebLogic Server under a user account with administrator privileges.

# Synchronizing clock times

You must ensure that all machines in a horizontal cluster synchronize their clock times regularly. Your LiveCycle ES installation may encounter problems if the node times differ by more than a few seconds.

Apply the standard time synchronization practices employed by your network to all machines of the LiveCycle ES cluster.

# Installing WebLogic Server

The following procedure details how to install WebLogic Server. It is assumed that you have downloaded and unzipped the installation file to an installation directory, and have opened a system terminal and navigated to that directory.

You must perform the following tasks to install WebLogic Server:

- Install the WebLogic Server software (See "To install WebLogic Server software:" on page 32.)

- Create the WebLogic domain (See "To create a WebLogic domain:" on page 32.)

- Create a boot.properties file to enable WebLogic Server. This allows the start of the server without manually entering the user name and password at the command line. (See "To create a boot.properties file:" on page 33.)

➤ **To install WebLogic Server software:**

1. Log in to the computer on which you will install WebLogic Server as a user with administrator privileges.

2. Run the installation program appropriate for your operating system:
   - (Windows) `server920_win32.exe`
   - (Linux) `server920_linux.bin`
   - (Solaris) `server920_solaris.bin`

3. Complete the steps of the installation wizard, accepting the default options presented.

4. In the Installation Complete window, deselect **Run Quickstart** and click **Done**.

5. Repeat steps 1 to 4 on each computer in the cluster.

# Creating a WebLogic domain

You must create your WebLogic domain and create a boot.properties file.

➤ **To create a WebLogic domain:**

1. Log on to the computer on which you installed WebLogic Server as a user with administrator privileges.

2. From a command prompt, navigate to the directory *[appserver root]*/common/bin and start the WebLogic Configuration Wizard by entering the following command:
   - (Windows) `config.cmd`
   - (Linux, UNIX) `./config.sh`

3. On the Create or Extend a Configuration screen, select **Create a new WebLogic domain** and click **Next**.

4. Select **Generate a domain configured automatically to support the following BEA products** and click **Next**.

5. Enter a user name and password, confirm the password by retyping it, and then click **Next**.

6. In the left pane, select **Production Mode**.

7. In the right pane, select **BEA Supplied SDKs** and the appropriate platform, and then click **Next**:

   - (Solaris) **Sun SDK**.

   - (All other platforms) **JRockit JVM**.

8. In Customize Environment and Services Settings, select **No** and click **Next**.

9. Enter the domain name and click **Next**.

   **Note:** All computers in the cluster must use the same domain name. For this document this defined domain name appears as *[domain name]*.

10. On the Creating Configuration screen, when the configuration creation is 100% complete, do one for these tasks:

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

    - (Linux, UNIX) Click **Done**.

11. Repeat steps 1 to 10 for each additional WebLogic Server instance of the cluster.

➤ **To create a boot.properties file:**

1. Open a text editor and enter the following lines:

   ```
   username=[username from weblogic install]
   password=[password from weblogic install]
   ```

2. Save the text file as *[BEA_HOME]*/user_projects/domains/*[domain name]*/boot.properties.

3. Restart the Admin server.

**Note:** For more information, see http://edocs.bea.com/wls/docs92/server_start/overview.html.

## Configuring the WebLogic Server cluster

You must perform the following tasks to configure your WebLogic Server cluster:

- Create the members of the cluster. (See "To create the members of the cluster:" on page 34.)

- Create the cluster and add the members to the cluster. (See "To create the cluster:" on page 35.)

- Configure time-out settings for the cluster. (See "Changing the LiveCycle ES time-out settings" on page 35.)

- Configure the node manager for the cluster (See "Configuring the node manager for the cluster" on page 36.)

- Start the node manager and the nodes of the cluster (See "Starting the node manager and managed servers" on page 37.)

➤ **To create the members of the cluster:**

1. In a command window, navigate to *[BEA_HOME]*/user_projects/domains/*<domain name>*/bin and enter the following command to start WebLogic Admin Server:

   - (Windows) `startWebLogic.cmd`

   - (Linux, UNIX) `nohup ./startWebLogic.sh&`

2. Start WebLogic Server Administration Console by typing `http://<Computer1>:7001/console` in the URL line of a web browser.

3. Log in by entering the user name and password created when you installed WebLogic Server.

4. Under Domain Structure, click **Environments** > **Machines**.

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

6. Click **New** and enter the computer name in the **Name** box, select the appropriate operating system from the **Machine OS** list, and then click **OK**. The Summary of Machines screen appears. Repeat this step for each computer you want to add to the cluster.

7. Click the machine name you just created.

8. Click the **Configuration** tab, click **Servers**, and then click **Add**.

9. Select **Create a new server and associate it with this machine** and click **Next**.

10. In the **Server Name** box, enter your server name (for example, `Server1`).

11. In the **Server Listen Address** box, enter one of these locations (where the server will listen for incoming connections):

    - The static IP address of the computer

    - The DNS name of the computer

    - If a local computer is being used, "localhost" or 127.0.0.1

12. In the **Server Listen Port**, enter a unique port number for the server.

    **Note:** Do not use the default value 7001. This port is required by the administration server.

13. Click **Finish**.

14. Repeat steps 7 to 13 for each computer on which you are creating servers.

15. In the left pane, click **Activate Changes**.

   **Note:** Ensure that you use appropriate values for the server name and a unique port number.

   **Note:** For any remote computer added to the cluster, Node Manager Listen Address needs to be changed. By default, it is localhost. Change the Listen Address from localhost to the host or IP address of the computer where the Node Manager resides.

➤ **To create the cluster:**

1. Return to the WebLogic Server Administration Console home screen.

2. Under Domain Structure, click **Environment** > **Clusters**.

3. In the **Clusters** table, click **New**.

4. In the **Name** box, enter your cluster name (for example, type `mycluster`).

5. In the **Multicast Address** box, enter an IP multicast address within the range 224.0.0.0 to 239.255.255.255 (for example, type `239.192.0.1`).

   **Note:** The multicast address should be unique to the cluster within the same network.

6. In the **Multicast Port** box, enter a port number. Valid values are 1 to 65535.

7. Click **OK**.

➤ **To assign servers to the cluster:**

1. Return to the WebLogic Server Administration Console home screen.

2. Under Domain Structure, click **Environment** > **Clusters**.

3. In the **Cluster** table, select the cluster created in "To create the cluster:" on page 35.

4. In the **Configuration** tab, click **Servers**, and then click **Add**.

5. From the **Select a server** list, select the server name to add, and then click **Finish**.

   **Note:** Do not add AdminServer to the cluster.

6. In the left pane, click **Activate Changes**.

## Changing the LiveCycle ES time-out settings

Depending on your deployment, LiveCycle ES EAR files can get large. To avoid EAR file deployment time-outs, you must increase the WebLogic Server time-out settings.

➤ **To configure transaction time-out settings for the cluster:**

1. In the WebLogic Server Administration Console, click **Lock & Edit**.

2. Under Domain Structure, click your domain name as defined in "To create a WebLogic domain:" on page 32.

3. Click the **Configuration** tab and then click **JTA**.

4. In the **Timeout Seconds** box, enter 600 and, in the **Abandon Timeout Seconds** box, enter 600.

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

**Note:** Repeat these steps for each server in the cluster.

➤ **To configure stuck thread time-out settings for the cluster:**

1. Under Domain Structure, click **Environment** > **Servers**.

2. In the **Servers** table, click the name of the server.

3. Click the **Configuration** tab and then click **Tuning**.

4. In the **Stuck Thread Max Time** box, enter `1200`.

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

**Note:** Repeat these steps for each server in the cluster.

## Configuring the node manager for the cluster

You must configure the node manager for the cluster. This allows you to use the administration server to start, stop, monitor, and perform other common tasks on the nodes of the cluster from the administration console.

➤ **To configure the node manager for the cluster:**

1. Under Domain Structure in the WebLogic Server Administration Console, click your domain name as defined in "To create a WebLogic domain:" on page 32.

2. Click the **Security** tab, click **General**, and then click **Advanced** to expand the advanced details.

3. Click **Lock & Edit**.

4. In the **NodeManager Usernam**e box, change the user name to the value established when creating the domain.

5. In the **NodeManager Password** box, change the password to the value established when creating the domain.

6. Click **Activate Changes**.

7. (Horizontally clustered) On the computer(s) that are not running AdminServer, do one of these tasks:

    ● (Windows) Navigate to *[appserver root]*\common\bin and enter the following command:

      ```
      wlst.cmd
      ```

    ● (Linux, UNIX) Navigate to *[appserver root]*/common/bin and enter the following command:

      ```
      ./wlst.sh
      ```

8. (Horizontally clustered) At the `wlst` command prompt, enter the following command to connect to AdminServer:

    ```
    connect('<adminusername>','<adminpassword>','<adminserverURL>')
    ```

    where

    ● `<adminusername>` is the user name of the administration server.

    ● `<adminpassword>` is the password for the administration server user.

    ● `<adminserverURL>` is the URL to the administration server.

    **Note:** For help about this command, type `help('connect')` at the `wlst` command prompt.

9. (Horizontally clustered) When connected to the administration server, enter the following command to enroll the secondary computer in the cluster:

```
nmEnroll('[appserver domain]')
```

10. (Horizontally clustered) Complete the task by entering the following command:

```
exit()
```

➤ **To allow mutual access between servers of the cluster:**

Complete this procedure for each computer in a horizontal cluster. This procedure does not apply to vertically clustered instances of the cluster.

1. Open a text editor and enter the IP address or host name of each computer in the cluster as individual lines, as shown in this example:

```
localhost
127.0.0.1
11.11.11.11
22.22.22.22
```

2. Save the file to each computer in the cluster as nodemanager.hosts, in one of these locations:

   - (Windows) *[appserver root]*\common\nodemanager
   - (Linux, UNIX) *[appserver root]*/common/nodemanager

3. In a text editor, open the existing hosts file of any computer in the cluster from one of the following locations:

   - (Windows) C:\WINDOWS\system32\drivers\etc
   - (Linux, UNIX) /etc

4. Add the IP address and host name of all computers in the cluster.

5. Save the edited file to each computer in the cluster in one of the following locations:

   - (Windows) C:\WINDOWS\system32\drivers\etc
   - (Linux, UNIX) /etc

## Starting the node manager and managed servers

Use the following procedures to start the node manager and the managed servers of the cluster.

➤ **To start the node manager:**

1. On one computer of the cluster, open a command window and navigate to the directory:

   - (Windows) `[appserver root]\server\bin` and then enter this command:

     ```
     startNodeManager.cmd
     ```

   - (Linux, UNIX) `[appserver root]/server/bin` and then enter this command:

     ```
     nohup ./startNodeManager.sh&
     ```

   **Note:** `startNodeManager.cmd` should be installed as Windows Services to run silently in the background. For details, see
   http://e-docs.bea.com/wls/docs92/server_start/nodemgr.html#wp1083125.

2.  Repeat the above step for each additional computer in the cluster.

➤ **To start the managed servers:**

1.  Under Domain Structure in the LiveCycle ES Administration Console, click **Environment** > **Clusters**.

2.  Click the name of the cluster you want to start.

3.  Click the **Control** tab, select the check box for each server to start, and then click **Start**.

# 5 | Configuring the Database

All members in the cluster share the JDBC data sources. You must create and configure the shared JDBC data source:

- Create a data source. (See "Creating and configuring the IDP_DS data source" on page 39.)
- Configure the transaction settings for the data source. (See "To configure transaction settings for the data source:" on page 40.)
- Configure the database properties of the data source. (See "To configure database settings:" on page 40.)
- Assign the data source to the cluster. (See "Assigning the data source to the cluster" on page 40.)
- Configure the maximum pool capacity of the data source. (See "Configuring the maximum pool capacity of the data source" on page 41.)

## Location of JDBC drivers

During configuration, you are required to provide the location of the JDBC drivers for your database. The following sections provide the location of these drivers. The Oracle, DB2, and MySQL drivers are in the *[LiveCycleES root]*/lib/db/*[database]* directory. You install the SQL Server database driver using the following procedure.

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

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

2. (Windows) Run the file. Extract the files to a temporary directory (*[SQL_root]*).

   (Linux) Extract the *.tar.gz files to a temporary directory (*[SQL_root]*). The JDBC driver file is *[SQL_root]*/sqljdbc_1.1/enu/sqljdbc.jar.

## Creating and configuring the IDP_DS data source

You must create the data source for your cluster.

➤ **To create a data source:**

1. In the WebLogic Server Administration Console, under Domain Structure, click **Services** > **JDBC** and then click **Data Sources**.

2. Click **New**.

3. (Optional) In the **Name** box, enter the name for the data source (for example, type IDP_DS).

4. In the **JNDI Name** box, enter IDP_DS  as the name of the data source.

5. In the **Database** list, select the database type that matches your database.

6.  In the **Database Driver** list, select the appropriate database driver for your database.

7.  Click **Next**.

➤ **To configure transaction settings for the data source:**

1.  Select **Supports Global Transactions** and then select **Emulate Two-Phase Commit**.

2.  Click **Next**.

➤ **To configure database settings:**

1.  In the **Database Nam**e box, enter the name of the database.

2.  In the **Host Name** box, enter the IP address of the computer hosting the database.

3.  In the **Port** box, enter the port number of the computer hosting the database.

4.  In the **Database User Name** box, enter the user name for the database.

5.  In the **Password** box, enter the password for the database and then enter it again in the **Confirm Password** box.

6.  Click **Next**.

7.  Click **Test Configuration**. A confirmation response is displayed, confirming that the database configuration is correct.

8.  When the test succeeds, click **Finish**.

9.  Under Change Center, click **Activate Changes**.

# Assigning the data source to the cluster

You must assign the data source to the cluster.

➤ **To assign a data source to the cluster:**

1.  In WebLogic Server Administration Console, under Domain Structure, click **Services** > **JDBC** and then click **Data Sources**.

2.  Click the name of the data source to assign to the cluster.

3.  Click the **Targets** tab.

4.  In the Clusters area, select the cluster name and then select **All servers in the cluster**.

5.  Click **Save**

6.  Under Change Center, click **Activate Changes**.

# Configuring the maximum pool capacity of the data source

You must configure a maximum pool capacity for the data source.

➤ **To set the maximum pool capacity:**

1. In the WebLogic Server Administration Console, under Domain Structure, click **Services** > **JDBC** and then click **Data Sources**.

2. Click the name of the data source to configure.

3. Click the **Configuration** tab and then click **Connection Pool**.

4. In the **Maximum Capacity** box, enter `100`.

5. Click **Save.**

6. Under Change Center, click **Activate Changes**.

   **Note:** When creating a JDBC data source for Oracle, in the **Initial Capacity** box, enter `15`.

# Creating and configuring a data source for Rights Management

For Rights Management you must create a separate data source for your cluster.

➤ **To create a data source:**

1. In the WebLogic Server Administration Console, under Domain Structure, click **Services** > **JDBC** and then click **Data Sources**.

2. Click **New**.

3. In the **Name** box, enter `RM_DS` as the name for the data source.

4. In the **JNDI Name** box, enter `EDS_DS` as the name of the data source.

5. In the **Database** list, select the database type that matches your database.

6. In the **Database Driver** list, select the appropriate database driver for your database.

7. Click **Next**.

➤ **To configure transaction settings for the data source:**

1. Select **Supports Global Transactions** and then select **One-Phase Commit**.

2. Click **Next**.

➤ **To configure database settings:**

1. In the **Database Nam**e box, enter the name of the database.

2. In the **Host Name** box, enter the IP address of the computer hosting the database.

3. In the **Port** box, enter the port number of the computer hosting the database.

4.  In the **Database User Name** box, enter the user name for the database.

5.  In the **Password** box, enter the password for the database and then enter it again in the **Confirm Password** box.

6.  Click **Next**.

7.  Click **Test Configuration**. A confirmation response is displayed, confirming the database configuration is correct.

8.  When the test succeeds, click **Finish**.

9.  Under Change Center, click **Activate Changes**.

# Configuring the maximum pool capacity of the data source (Rights Management)

You must configure a maximum pool capacity for the Rights Management data source.

➤ **To set the maximum pool capacity:**

1.  In the WebLogic Server Administration Console, under Domain Structure, click **Services** > **JDBC** and then click **Data Sources**.

2.  Click the name of the data source to configure.

3.  Click the **Configuration** tab and then click **Connection Pool**.

4.  In the **Initial Capacity** box, enter 10.

5.  In the **Maximum Capacity** box, enter  30.

6.  In the **Capacity Increment** box, enter 5.

7.  In the **Statement Cache Size** box, enter 100.

8.  Click **Save.**

9.  Under Change Center, click **Activate Changes**.

# Assigning the data source to the cluster (Rights Management)

You must assign the data source to the cluster.

➤ **To assign a data source to the cluster:**

1.  In the WebLogic Server Administration Console, under Domain Structure, click **Services** > **JDBC** and then click **Data Sources**.

2.  Click the name of the data source to assign to the cluster.

3.  Click the **Targets** tab.

4.  In the Clusters area, select the cluster name and then select **All servers in the cluster**.

5.  Click **Save**

6.  Under Change Center, click **Activate Changes**.

# 6    Configuring JMS Resources

You must perform the following tasks to create and configure JMS resources for your cluster:

- Configure a JDBC store. (See "Configuring a persistent store" on page 44.)
- Create a JMS server. (See "Creating a JMS server" on page 45.)
- Create JMS modules. (See "Creating JMS modules" on page 45.)
- Add resources to the JMS modules. (See "Adding resources to a JMS module" on page 45.)

## Configuring a persistent store

It is useful to create the persistent JDBC stores before creating the JMS assets. It is recommended that you create a persistent store on each server in the cluster. These persistent stores will be used by the JMS Servers created in a subsequent step. For more information, see http://e-docs.bea.com/wls/docs92/config_wls/store.html.

➤ **To configure a JDBC store:**

1. In the WebLogic Server Administration Console, under Domain Structure, click **Services** and click **Persistent Stores**.

2. Click **Lock and Edit**.

3. Click **New** and then select **Create JDBCStore** from the list.

4. In the **Name** box, enter a name for the JDBC store (for example, type `IDPStore<serverName>`).

5. In the **Target** list, select the target for the JDBC store.

   **Note:** It is recommended that you select a migratable server (identified as **<server name>** (migratable) in the **Target** list) to provide failover functionality.

6. In the **Data Source** list, select **IDP_DS**.

7. In the **Prefix Name** box, type the server name.

   **Note:** It is highly recommended that you configure the **Prefix** option to a unique value for each configured JDBC store table. See details at http://e-docs.bea.com/wls/docs92/config_wls/store.html.

8. Click **Finish**.

9. Repeat steps 1 to 8 for each server in the cluster.

10. Click on **Activate Changes**.

    **Note:** The JDBC store of multiple computers in a horizontal cluster may point to different data sources.

# Creating a JMS server

You must create a JMS server on each server in the cluster.

➤ **To create a JMS server:**

1.  In the WebLogic Server Administration Console, under Domain Structure **Services**, click **Messaging** and then **JMS Servers**.

2.  Click **New** in the **JMS Servers** table.

3.  In the **Name** box, enter a name for the JMS server (for example, type `IDP_JMSServer_<serverName>`).

4.  In the **Persistent Store** list, select a JDBC store to use with this server.

    **Note:** After selecting the JDBC store, ensure that the server name is still listed as IDP_JMSServer_<serverName>.

5.  Click **Next**.

6.  In the **Target** list, select the server name to deploy this JMS Server.

    **Note:** It is recommended that you select a migratable server (identified as **<server name>** (migratable) in the **Target** list) to provide failover functionality.

7.  Click **Finish**

8.  Repeat steps 1 to 7 for each server in the cluster.

9.  Click **Activate Changes**.

# Creating JMS modules

You must create a JMS system module before you can add resources to create a JMS destination (that is, a queue or topic).

➤ **To create JMS modules:**

1.  In WebLogic Server Administration Console, under Domain Structure, click **Services**, click **Messaging** and then **JMS Modules**.

2.  Click **New** and, in the **Name** box, enter a name for the JMS module (for example, type `IDPJMSModule`).

3.  Click **Next** and, to select your cluster, select the check box beside the cluster name.

4.  Click **Next** and then select **Would you like to add resources to this JMS system module**.

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

# Adding resources to a JMS module

You must add resources (JMS connection factories, distributed queues, and distributed topics) to the JMS module you created.

Perform the following tasks to add resources to a JMS module:

- Add a distributed queue and configure failover destinations for queues. (See "Adding distributed queues to a JMS module" on page 46.)

- Add a distributed topic. (See "Adding distributed topics to a JMS module" on page 47.)

- Add a connection factory and enable XA on the connection factory. (See "Creating JMS connection factories" on page 48.)

Later, you can group your resources into a subdeployment for easier management. (See "Creating subdeployments" on page 49.) You must define the first JMS resource before you can create subdeployments.

## Adding distributed queues to a JMS module

You can add distributed queues to your JMS module.

After you add distributed queues as JMS resources, you can configure a failover destination for some JMS queues (that is, set a final queue that a message is delivered to if message handling fails after a number of retries that you have configured for the original queue).

➤ **To add a distributed queue to a JMS module:**

1. In the WebLogic Server Administration Console, under Domain Structure, click **Services**, click **Messaging** and then **JMS Modules**.

2. Click the name of your JMS module (for example, IDP_JMSModule).

3. Click **New** and select **Distributed Queue**.

4. Click **Next** and, in the **Name** box, enter a queue name, (for example, type `adobe_PECommandQueue`).

5. In the **JNDI Name** box, enter a JNDI name for the queue (for example, type `adobe_PECommandQueue`).

6. Accept the default values for the **Load Balancing** list and **Allocate Members Uniformly** box, and then click **Next**.

7. (Optional) Click **Advanced Targeting** and select a subdeployment.

8. Click **Finish**.

   **Note:** You can select a subdeployment only if you previously added an initial resource and created a subdeployment. (See "Creating subdeployments" on page 49.)

9. Repeat steps 1 to 6 for the distributed queues listed in this table.

| JMS queue name | JNDI name |
| --- | --- |
| adobe_PECommandQueue* | adobe_PECommandQueue |
| adobe_PEDCommandQueue | adobe_PEDCommandQueue |
| adobe_JobManagerQueue | adobe_JobManagerQueue |

**Note:** * Destination is configured to have an error destination.

➤ **To configure a queue failover destination:**

Choose the JMS queue name for the configuration. In this case, adobe_PECommandQueue is being used as an example.

1. In the WebLogic Server Administration Console, under Domain Structure, click **Services**, click **Messaging** and then **JMS Modules**.

2. Click the chosen destination (for example, **adobe_PECommandQueue**) from the **Summary of Resources** table.

3. Click the **Configurations** tab and then click **Delivery Failure**.

4. In the **Redelivery Limit** box, enter 5.

5. Select **Redirect** from the **Expiration Policy** list.

6. Select **adobe_PEDCommandQueue** from the **Error Destination** list.

7. Click **Save**.

## Adding distributed topics to a JMS module

You can add distributed topics to your JMS module.

➤ **To add a distributed topic to a JMS module:**

1. In the WebLogic Server Administration Console, under Domain Structure, click **Services**, click **Messaging** and then **JMS Modules**.

2. Click the name of your JMS module (for example, **IDP_JMSModule**).

3. Click **New** and select **Distributed Topic**.

4. Click **Next** and, in the **Name** box, enter a topic name (for example, type `adobe_TaskEventTopic`).

5. In the **JNDI Name** box, enter a JNDI name for the queue (for example, type `adobe_TaskEventTopic`).

6. Accept the default values for the **Load Balancing** list and **Allocate Members Uniformly** and click **Next**.

7. (Optional) Click **Advanced Targeting** and select a subdeployment.

8. Click **Finish**.

9. Repeat steps 1 to 8 for the topics described in the following table:

| JMS queue name | JNDI name |
| --- | --- |
| adobe_TaskEventTopic | adobe_TaskEventTopic |

# Creating JMS connection factories

You can add JMS connection factories to your JMS module.

**Note:** You must enable XA on the connection factories you create.

➤ **To create a JMS connection factory:**

1. In the WebLogic Server Administration Console, under Domain Structure, click **Services**, click **Messaging** and then **JMS Modules**.

2. Click the name of your JMS module (for example, **IDP_JMSModule**).

3. Click **New** and select **Connection Factory**.

4. Click **Next** and in the **Name** box, enter the name the factory (for example, type `IDPQueueConnectionFactory`).

5. Click **Next**.

6. (Optional) Click **Advanced Targeting** and select an existing subdeployment (for example, **IDP_JMS_SubDeployment**).

7. Click **Finish**.

8. Repeat steps 1 to 7 for the connection factories described in this table.

| Connection Factory Name | JNDI Name |
| --- | --- |
| IDPTopicConnectionFactory | IDPTopicConnectionFactory |
| IDPQueueConnectionFactory | IDPQueueConnectionFactory |
| JobManagerQueueConnectionFactory | JobManagerQueueConnectionFactory |

➤ **To enable XA on connection factories:**

1. In the WebLogic Server Administration Console, under Domain Structure, click **Services**, click **Messaging** and then **JMS Modules**.

2. Click the name of your JMS module (for example, **IDP_JMSModule**).

3. Click the name of a connection factory (for example, **IDPQueueConnectionFactory**).

4. Click the **Configuration** tab and then click **Transactions**.

5. Select **XA Connection Factory Enabled** and then c lick **Save**.

6. Complete steps 1 to 5 for the following connection factories:
   - IDPQueueConnectionFactory
   - IDPTopicConnectionFactory
   - JobManagerQueueConnectionFactory

## Creating subdeployments

A *subdeployment* is a grouping mechanism. You can group JMS resources related to specific targets (for example, servers or clusters) together into subdeployments and then configure the subdeployment to specific targets.

Grouping JMS resources with connection factories collocates the queues and connections, reducing network traffic. Additionally, if the JMS servers are identified as migratable in a cluster, the server can move all resources that are grouped together more easily.

Each JMS resource requires a subdeployment, which can then be assigned to a server. One subdeployment should be adequate for all LiveCycle ES JMS resources.

**Note:** You must create at least one initial resource before you can create a subdeployment.

➤ **To create a subdeployment:**

1. In the WebLogic Server Administration Console, under Domain Structure, click **Services**, click **Messaging** and then **JMS Modules**.

2. Click the name of your JMS module (for example, **IDP_JMSModule**).

3. Click the **Subdeployments** tab and then click **New**.

4. In the **Subdeployment Name** box, enter the name for your subdeployment.

    **Note:** Ensure that the subdeployment name is unique (for example, enter IDP_JMS_SubDeployment).

5. Click **Next** and select the check box beside the cluster you want to deploy.

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

# 7 | Installing LiveCycle ES Solution Components

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

Before you install the solution components, you must ensure that your environment includes the software and hardware required to run LiveCycle ES. You should also understand the installation options and have prepared the environment as required.

This chapter covers the following topics:

## Installing the product files

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

- The serial numbers for the solution component(s) you are installing.
- The type of installation and configuration you are performing.

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

- (Windows) C:\Adobe\LiveCycle8\
- (Linux and UNIX) /opt/adobe/livecycle8/

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 meets the minimum requirements as outlined in the *Preparing to Install LiveCycle* document. The temporary directory is one of the following locations:

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

The InstallShield command line parameter `-is:tempdir` is not supported with the LiveCycle ES installer. To work around this limitation, use the **TMP** environment variable to point to a disk that has the required amount of free space.

On Linux systems, you can install, configure, and deploy as a non-root user.:

- (Solaris) /var/tmp

**Note:** When you are installing the solution component on a Linux or UNIX operating system, you must be logged in as the root user to successfully install the solution component to the default location /opt/adobe/LiveCycle8/. If you are logged in as a non-root user, you must change the installation directory to one for which you have permissions (read-write-execute privileges) (for example, /home/adobe/livecycle8).

## Installing to a Windows staging platform for deployment on Linux or UNIX

LiveCycle ES can be installed and configured on Windows for deployment on a Linux or UNIX platform. You can use this functionality for installing to a locked-down Linux environment. For example, a locked-down environment would not have XWindows installed. When you run the installation program on Windows, you can choose a Linux or UNIX operating system as the target platform for deploying LiveCycle ES. The installation program installs binaries for AIX, Linux, or Solaris, which are also used by LiveCycle Configuration Manager when you configure the product.

The computer running Windows can then be used as a staging location for the deployable objects, which can be copied to a computer running Linux or UNIX for deployment to the application server. The application server that you are targeting must be consistent with what you choose during installation and configuration, regardless of the operating system.

**Note:** If your installation includes LiveCycle PDF Generator ES and you will use the native application conversion feature, the host and target operating system must be Windows.

## Installing LiveCycle ES

This section covers the initial installation of LiveCycle ES. For information about configuration and deployment, see "Configuring and Deploying LiveCycle ES" on page 57.

**Note:** To avoid permission issues during the deployment, ensure that you are logged in as the user who will run the WebLogic process when you run the LiveCycle ES installer and LiveCycle Configuration Manager.

➤ **To install LiveCycle ES:**

1. Navigate to the /livecycle_server/8.0 directory of the installation media.

2. Start the installation program:
   - (Windows) Double-click the win_livecycle8_setup.exe file.
   - (Linux, UNIX) From a command prompt, type the file name appropriate for your operating system:

     ```
     ./aix_livecycle8_setup.bin
     ```

     ```
     ./linux_livecycle8_setup.bin
     ```

     ```
     ./solaris_livecycle8_setup.bin
     ```

3. When prompted, select the language you want the installation to use, and then click **OK**.

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

5. On the Destination screen, accept the default directory as listed or click **Browse** and navigate to the directory where you want to install the solution component, and then click **Next**.

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

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

6. Type a serial number in the text box and click **Add**. If you have licensed multiple solution components, repeat this step for each serial number you have.

7. After you add all the serial numbers to the list, click **Next**.

   **Tip:** To reset the serial number, click **Clear**.

8. Read the Product License Agreement. If you agree, select **I accept the terms of the license agreement**, and then click **Next**.

9. (Windows staging only) Select the operating system for which you plan to configure LiveCycle ES, and click **Next.**

   **Note:** At this point, you can specify to use Windows as a staging platform for your deployment. You can select Linux or a UNIX operating system as the target for deployment even if you are installing on Windows. For more details see "Installing to a Windows staging platform for deployment on Linux or UNIX" on page 51.

10. ***(PDF Generator ES for Windows only)*** Select the appropriate option on the PDF Generator ES screen:

    ● If you select **Yes, enable native application support for PDF Generator ES**, the software will check the version of Acrobat you have installed. If you do not have Acrobat installed, accept the prompt to install Acrobat 8.1 now.

      If have an unsupported version of Acrobat installed, complete the LiveCycle ES installation, uninstall Acrobat, and then install Acrobat 8.1 from the LiveCycle ES installation media by following the instructions in the procedure "Configuring Acrobat 8.1 for PDF Generator ES" on page 64.

    ● If you are installing in a clustered environment, select **No, do not enable native application support for PDF Generator ES** and go to step 12.

11. ***(PDF Generator ES for Windows only)*** Confirm you have Acrobat installed, and 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 solution component installation is completed.

13. Review the release notes that are displayed and click **Next**.

14. Select **Start LiveCycle Configuration Manager** and then click **Finish**.

    **Note:** Selecting the **Start** option starts LiveCycle Configuration Manager, which allows you to complete your configuration and deployment immediately. If you are not ready to run LiveCycle Configuration Manager immediately, ensure that the **Start LiveCycle Configuration Manager** option is not selected when you click **Finish**. You can start LiveCycle Configuration Manager when you are ready. (See "Configuring LiveCycle ES EAR files" on page 58.)

    When you are ready to proceed with the configuration and deployment, see "Configuring and Deploying LiveCycle ES" on page 57.

> **Note:** ***(PDF Generator ES for Windows only)*** If you did not choose to install Acrobat 8.1 using the LiveCycle ES installer, you must complete the steps listed in <u>"Configuring Acrobat 8.1 for PDF Generator ES" on page 64</u>.

# 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 *[LiveCycleES root]* directory.

# Configuring Server Start arguments

You must perform the following configuration tasks on each server instance of the cluster to ensure that LiveCycle ES will start properly:

- Configure the LiveCycle ES properties files.
- Configure the Server Startarguments

➤ **To configure the LiveCycle ES properties files:**

1. Locate the adobe-prof.properties file in *[LiveCycleES root]*/LiveCycle _ES_SDK/misc/Foundation/ Server-Configuration.

2. Copy the file to the directory *[appserver root]*/init-info.

3. Using a text editor, make the following changes:

   - adobe.pof.cache.swarm.cluster-tcp-config=TCP(bind_addr=*<Machine1>*;start_port=7800):TCPPING(initial_hosts=*<Machine2>*[7800];port_range=5;timeout=3000;num_initial_members=5):FD_SOCK: VERIFY_SUSPECT(timeout=1500;down_thread=false;up_thread=false):pbcast.NAKACK(gc_lag=100 ;retransmit_timeout=600,1200,2400,4800;up_thread=true;down_thread=true):pbcast.STABLE(desired_avg_gossip=20000;max_bytes=0;stability_delay=1000;up_thread=false;down_thread=false):pbcast.GMS(join_timeout=5000;join_retry_timeout=2000;shun=false;print_local_addr=true;down_thread=true;up_thread=true):pbcast.STATE_TRANSFER(up_thread=true;down_thread=true)

   > **Note:** Change *<Machine1>* and*<Machine2>* to your machine names and add any additional machines in the cluster to the "initial_hosts" section, separated by commas. For a vertical cluster, use the same TCP settings but different ports.

4. Locate the idp_scheduler_quartz.properties file in *[LiveCycleES root]*/LiveCycle _ES_SDK/misc/ Foundation/Server-Configuration.

5. Copy the file to the directory *[appserver root]*/init-info.

6. Repeat steps <u>1</u> to <u>5</u> for each server instance in the cluster.

➤ **To configure the Server Start for each server:**

1. In the WebLogic Server Administration Console, under Domain Structure, click **Environment** > **Servers** and, in the right pane, click the name of your server.

2. Click the **Configuration** tab and then click **Server Start**.

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

4. In the Arguments box, enter the following text:

```
-XX:MaxPermSize=256m -Xms512m -Xmx1024m
-Dadobeidp.RootDirectory=[appserver domain]
-Djava.net.preferIPv4Stack=true -Dfile.encoding=utf8
-Dadobe-pof.properties=[appserver domain]/init-info/adobe-pof.properties
-Dadobe.idp.scheduler.properties=[appserver
domain]/init-info/idp_scheduler_quartz.properties
-Djava.security.policy=[appserver domain]/server/lib/weblogic.policy
```

**Note:** Copy the above text block to a text editor, ensure that all line breaks are removed and replace all occurrences of `[appserver domain]` with your application server domain path.

**Note:** If you are installing LiveCycle Output ES or LiveCycle Forms ES, the current default value for PoolMax is 4. The actual value to set would depend on the hardware configuration and the expected usage in your environment. For optimal use, it is recommended that the lower limit of PoolMax must not be less than the number of CPUs available, and the upper limit must be determined by the load pattern on your server. Generally, the upper limit should be set to twice the number of CPUs on the server. To modify the existing PoolMax value, enter the following text at the end of the arguments added above.

```
-Dcom.adobe.convertpdf.bmc.POOL_MAX=<value>
-Dcom.adobe.xmlform.bmc.POOL_MAX=<value>
```

5. Click **Save**.

6. Repeat steps 2 to 5 for each server in your cluster.

**Note:** If you are setting up a vertical cluster (all member servers on the same physical computer), include the following text to the above "Arguments" field:

```
-Djava.io.tmpdir=<usr_temp>/<server_name>, where <usr_temp> is the OS
user's temp directory, e.g. /usr/tmp/
```

## Configuring LiveCycle Output ES and LiveCycle Forms ES

The current default value for PoolMax is 4. The actual value to set would depend on the hardware configuration and the expected usage in your environment.

For optimal use, it is recommended that the lower limit of PoolMax must not be less than the number of CPUs available, and the upper limit must be determined by the load pattern on your server. Generally, the upper limit should be set to twice the number of CPUs on the server.

# Configuring the class path of the managed server

➤ **To copy the necessary JAR files:**

1. Create a directory named *idplib* in *[apserver domain]*.

2. Copy the pop3.jar file from *[LiveCycle root]*/lib/weblogic to *[appserver domain]*/idplib on the WebLogic node.

➤ **To copy the necessary database drivers:**

1. Locate the database drivers at *[LiveCycleES root]*/lib/db.

2.  Copy the appropriate database JAR file to *[appserver domain]*/idplib on the WebLogic node

➤ **To determine the current class path of the managed server:**

1.  Using Node Manager, start each application server in the WebLogic cluster.

2.  On startup, the application server writes the current class path to the application server log file at *[appserver domain]*/servers/*[appserver]*/logs/*[appserver]*.log.

3.  Search the log file for "java.class.path" and copy the value for this property. This is the current class path.

➤ **To modify the class path of a managed server:**

1.  In the WebLogic Server Administration Console, under Domain Structure, click **Environment** > **Servers**.

2.  Click **Lock & Edit**, and then click the name of your server.

3.  Click the **Configuration** tab and then click **Server Start**.

4.  In the **Class Path** box, enter the class path from step 3 in the section "To determine the current class path of the managed server:" on page 55. Add the location and file name for the pop3.jar file (*[appserver domain]*/idplib/pop3.jar) to the beginning of the class path.

    **Note:** Ensure that the pop3.jar file is listed before the weblogic.jar file classes and the various JAR files are separated with a colon (:) (Linux/UNIX) or a semicolon (;) (Windows)

5.  Click Save.

6.  Repeat steps 2 to 4 for all servers in the cluster.

7.  Under Change Center, click **Activate Changes**.

# Creating a shared global document storage directory

All nodes in the cluster must share a common global document storage directory in the same way that they share a common database (that is, to store long-lived documents that are passed between services).

You must create the global document storage directory before initializing the LiveCycle ES database. Create a shared directory on any computer that is accessible by each node in the cluster, and ensure that all nodes of the cluster have read and write permissions for the directory.

Access to the global document storage directory must be secure. The long-lived documents in this directory may contain sensitive user information, such as information that requires special credentials when accessed using the LiveCycle ES SDK or user interfaces. Use a security method appropriate to your operating system. It is recommended that only the operating system account used to run the application server has read and write access to this directory.

Record the location where you create this shared directory for later use when you configure LiveCycle ES using LiveCycle Configuration Manager.

# Configuring the font directories

You must configure the font directories for each computer in the cluster, including the LiveCycle ES fonts installed in the *[LiveCycleES root]*\fonts directory.

The fonts must exist in the same path on all computers in the cluster. This can be accomplished by using a shared directory that is accessible by all computers in the cluster or by making the directory available on each computer in the cluster in an identical path.

Record the location where you create these shared directories for later use when you configure LiveCycle ES using LiveCycle Configuration Manager.

**Caution:** The font directories must be distinct from the global document storage directory; however, they may be distinct sibling subdirectories of a single shared parent directory.

# 8 | Configuring and Deploying LiveCycle ES

This chapter describes:

- LiveCycle Configuration Manager (see <u>"About LiveCycle Configuration Manager" on page 57</u>)
- Configuring LiveCycle ES solution components in an EAR file (See <u>"Configuring LiveCycle ES EAR files" on page 58.</u>)
- Manually deploying the LiveCycle ES EAR files to the application server (See <u>"Deploying LiveCycle ES EAR files" on page 59.</u>)
- Starting the application server (See <u>"Starting the application" on page 59.</u>)
- Initializing the LiveCycle ES database (See <u>"Initializing the LiveCycle ES database and deploying LiveCycle ES" on page 60.</u>)
- <u>See "Configuring LiveCycle PDF Generator ES" on page 61.</u>
- (Redeployment) Removing previously deployed EAR files (See <u>"Uninstalling EAR files" on page 63.</u>)
- Next steps (See <u>"Next steps" on page 63.</u>)

This chapter assumes that you have prepared your environment for hosting LiveCycle ES and installed the solution component files. If you have not done this, see <u>"Introduction" on page 9</u> and <u>"Installing LiveCycle ES Solution Components" on page 50</u>.

## About LiveCycle Configuration Manager

LiveCycle Configuration Manager is a wizard-like tool used to configure, deploy, and validate LiveCycle ES components for deployment to the application server.

LiveCycle Configuration Manager is installed with the solution component files when you run the LiveCycle ES installation program. When you run LiveCycle Configuration Manager, you specify the LiveCycle ES solution components you are configuring, as well as the tasks you want LiveCycle Configuration Manager to perform.

You can start LiveCycle Configuration Manager from the installation program to configure solution components during the installation process, or you can start LiveCycle Configuration Manager any time after the installation.

LiveCycle Configuration Manager provides an interface that prompts you for information so that it can automatically accomplish the following tasks:

- Configure the LiveCycle ES EAR files for deployment to the application server.

  **Note:** In a cluster environment, you must deploy the EAR files manually, not by using LiveCycle Configuration Manager.

- Initialize the LiveCycle ES database using LiveCycle Configuration Manager.
- Deploy LiveCycle ES components.
- Validate the LiveCycle ES component deployment.

# Configuring LiveCycle ES EAR files

When you run LiveCycle Configuration Manager, you can select the tasks that you want the program to perform automatically.

**Tip:** LiveCycle Configuration Manager verifies the values specified on each screen when you click Next. If it cannot validate a value, a warning appears, the property on the screen turns red, and you cannot proceed until you enter a valid value.

➤ **To configure the LiveCycle ES EAR files using LiveCycle Configuration Manager:**

1. Start LiveCycle Configuration Manager by navigating to the *[LiveCycleES root]*/configurationManager/bin directory and entering the following command:

    - (Windows) `ConfigurationManager.bat`

    - (Linux, UNIX) `./ConfigurationManager.sh`

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

3. On the Upgrade Selection screen, perform the appropriate task from the following choices:

    - If you are configuring the LiveCycle ES EAR files for an initial deployment of LiveCycle ES 8.x, ensure that **Upgrade LiveCycle 7.x** is deselected and then click **Next**.

    - If you are configuring the LiveCycle ES EAR files to upgrade from an existing LiveCycle ES 7.x deployment, select **Upgrade LiveCycle 7.x** and **Apply LiveCycle 7.x configuration to LiveCycle ES**, deselect **Migrate Data Essential to LiveCycle ES Operation** and **Migrate Remaining Data**, and then click **Next**. For information about upgrading your existing LiveCycle 7.x installation, see the *Upgrading to LiveCycle ES* document for you application server at http://www.adobe.com/go/learn_lc_documentation.

4. On the Solution Component screen, select the LiveCycle ES components you want to configure, deselect all other components, and then click **Next**.

5. On the Task Selection screen, select only **Configure LiveCycle ES**, deselect all other task options, and then click **Next**.

**Caution:** Do not select the tasks **Configure application server**, **Validate application server configuration**, or **Deploy LiveCycle ES EARs**. These tasks are not supported by LiveCycle Configuration Manager in a clustered configuration and must always be deselected.

6. (Upgrade) To import and extract the LiveCycle 7.x configuration data, see the *Upgrading to LiveCycle ES* document for your application server.

7. On the Configure LiveCycle ES screen, click **Configure** to generate the base EAR files for WebLogic Server and, when completed, click **Next**.

8. On the Configure LiveCycle ES (continued) screen, enter the required information:

    - Locations of the font directories that you created in "Configuring the font directories" on page 56.

    - Global document storage root directory that you created in "Configuring the font directories" on page 56.

    - (Optional) Values for the other configuration options on this screen. For more information, see *Installing and Configuring LiveCycle ES for WebLogic* or the context-sensitive Help for LiveCycle Configuration Manager.

9. Click **Configure** and, when configuration is complete, click **Next**.

   When LiveCycle Configuration Manager completes the configuration, it places the files to be deployed to the application server (adobe-livecycle-native-weblogic-*[OS]*.ear, adobe-livecycle-weblogic.ear and adobe-workspace-client.ear) in the following directory:

   - (Windows) *[LiveCycleES root]*\configurationManager\export
   - (Linux, UNIX) *[LiveCycleES root]*/configurationManager/export

10. Click **Exit**.

# Deploying LiveCycle ES EAR files

You must manually deploy your LiveCycle ES solution EAR files.

**Note:** If this deployment is a redeploy of the LiveCycle ES EAR files, you must first undeploy the previously deployed LiveCycle EAR files. (See "Uninstalling EAR files" on page 63.)

➤ **To deploy the LiveCycle ES EAR files:**

1. In the WebLogic Server Administration Console, under Domain Structure, click **Deployments** > **Install**.

2. Browse to *[LiveCycle root]*/configurationManager/export, select one of the EAR files, and click **Next**.

3. Select **Install this deployment as an application** and click **Next**.

4. Select the name of your cluster, select **All Servers in the Cluster**, and click **Next**.

5. Click **Next** to accept the default settings and then click **Finish**.

6. Under Change Center. click **Activate Changes**.

7. Repeat steps 1 to 6 for the each of the other EAR files.

# Starting the application

After deploying the EAR files, you must start the LiveCycle ES application. When the state changes to Active, the application has been deployed and started successfully.

WebLogic displays an error message if it cannot start the application. For information about WebLogic error messages, see the WebLogic documentation.

➤ **To restart the Managed Servers:**

1. In the WebLogic Server Administration Console, under Domain Structure, click **Environment** > **Clusters**.

2. In the **Clusters** table, click the name of your cluster and then click the **Control** tab.

3. Select the servers in your cluster and click **Shutdown**, and then click **Force Shutdown Now**.

   **Note:** You can monitor the shutdown progress of each server in the cluster by reloading the page. The status column in the table will update.

4.  When the server shutdowns are complete, return to the WebLogic Server Administration Console and, under Domain Structure, click **Environment** > **Clusters**.

5.  In the **Clusters** table, click the name of your cluster.

6.  Click the **Control** tab and confirm that your servers are in **Shutdown** state.

7.  Select the servers in your cluster and click **Start**.

    **Note:** View the server log files to monitor the progress of your Managed Server startup. The message "`<Notice> <WebLogicServer> <BEA-000360> <Server started in RUNNING mode>`" confirms a successful startup. Startup can take up to 5-10 minutes.

8.  Click **Deployments** and confirm that the EAR files you deployed are displayed as Active in the state column in the table on the right

# Initializing the LiveCycle ES database and deploying LiveCycle ES

You must initialize the LiveCycle ES database and deploy the LiveCycle ES components by using LiveCycle Configuration Manager. You can also validate that your LiveCycle ES components deployed properly.

➤ **To initialize the LiveCycle ES database and deploy LiveCycle ES components:**

1.  Ensure that the WebLogic cluster is started and the LiveCycle ES enterprise applications are deployed and started.

2.  Start LiveCycle Configuration Manager by navigating to the *[LiveCycle ES root] /*configurationManager/ bin directory and entering the following command:
    *   (Windows) **ConfigurationManager.bat**
    *   (Linux, UNIX) **./ConfigurationManager.sh**

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

4.  On the Upgrade Selection screen, perform the appropriate task from the following choices:
    *   If you are configuring the LiveCycle ES EAR files for an initial deployment of LiveCycle ES, ensure that **Upgrade LiveCycle 7.x** is deselected, and then click **Next**.
    *   If you are configuring the LiveCycle ES EAR files to upgrade from an existing LiveCycle 7.x deployment, select **Upgrade LiveCycle 7.x**, **Migrate Data Essential to LiveCycle ES Operation**, and **Migrate Remaining Data** (if enabled), deselect **Apply LiveCycle 7.x configuration to LiveCycle ES**, and then click **Next**. For information about upgrading your existing LiveCycle 7.x installation, see the *Upgrading to LiveCycle ES* document for your application server.

5.  On the Solution Component screen, select the LiveCycle ES solution components that you selected when configuring the LiveCycle ES EAR files and then click **Next**.

6.  On the Task Selection screen, select **Initialize LiveCycle ES database**, **Deploy LiveCycle ES components**, and **Validate LiveCycle ES component deployment**, and then click **Next**.

**Caution:** Ensure that all other options are deselected.

7.  (Upgrade) If you are prompted to provide the connection information for your LiveCycle 7.x database, see the *Upgrading to LiveCycle ES* document for your application server for information about upgrading your existing LiveCycle 7.x database.

8.  On the LiveCycle ES Database Initialization screen, set these properties and click **Initialize**:

    ● In the **Host** field, enter the host name of one of the cluster nodes.

    ● In the HTTP port field, enter the HTTP port for that node.

    ● In the WebLogic Admin User ID box, type the user name of the administration user.

    ● In the WebLogic Admin Password field, type the password for the administration user.

9.  When the initialization is complete, click **Next**.

10. On the LiveCycle ES Server Information screen, enter the host and HTTP port values. These value must be the same ones used when initializing the LiveCycle ES database.

11. Enter the LiveCycle ES user name and password, and then click **Next**.

    **Note:** The default user name is *administrator* and the default password is *password*.

12. On the LiveCycle ES Component Deployment screen, click **Deploy** and, when deployment is complete, click **Next**.

13. Click **Validate** to validate the deployment of the LiveCycle ES components and, when validation is complete, click **Next**.

14. (Upgrade) Complete any additional data migration tasks.

15. Click **Exit** to close LiveCycle Configuration Manager.

16. Restart all instances of WebLogic Serverin the cluster.

# Configuring LiveCycle PDF Generator ES

LiveCycle PDF Generator ES is capable of doing multiple PDF conversions simultaneously for certain types of input files. This is enforced through the use of stateless session beans.

This section describes advanced tuning for PDF Generator ES. This section should be completed only on a production system by an advanced application server administrator.

## Configuring the EJB pool size

Four different stateless session beans exist for enforcing independent pool sizes for the following types of input files:

● PostScript® and Encapsulated PostScript files.

● Image files (such as BMP, TIFF, PNG, JPEG, and so on.)

● OpenOffice files.

● All other file types (except HTML files) such Microsoft Office, Adobe Photoshop, PageMaker, and FrameMaker files.

The pool size for HTML to PDF conversions is not managed through the use of stateless session beans.

The default pool size for PostScript and Encapsulated PostScript files, and for Image files, is set to 3, and the default pool size for OpenOffice and other file types (except HTML) is set to 1.

You can configure the PS/EPS and Image pool size to a different value based on your server hardware configuration, such as the number of CPUs, the number of cores within each CPU, and so on. However, it is mandatory for the OpenOffice and other file types pool size to be left unchanged (that is, 1) for proper functioning of PDF Generator.

This section describes how to configure the pool size for PS2PDF and Image2PDF.

The text below assumes that the following two LiveCycle ES application EAR files have been deployed on the application server:

- adobe-livecycle-weblogic.ear

- adobe-livecycle-native-weblogic-*<platform>*.ear

Here, *<platform>* should be replaced with one of the strings "x86_win32", "x86_linux", "sparc_sunos" or "powerpc_aix", depending on whether you are running your application server on Windows, Linux, SunOS or AIX operating system.

**Note:** The following procedure describes how to make the necessary changes to the EAR files using WinRAR. You may use any other equivalent tool (such as the Java jar command), but notice that the procedural steps for extracting and repackaging will be different for other tools.

➤ **To configure the EJB pool size:**

1. Open `adobe-livecycle-weblogic.ear` in WinRAR.

2. Navigate to the following path, opening nested JAR files in WinRAR successively:
   `adobe-pdfg-bmc-invoker-ejb.jar > META-INF`.

3. Extract `weblogic-ejb-jar.xml` to any suitable location on the file system and open it in a plain text editor such as Notepad or vi.

4. In the XML, navigate to the node with the following path (expressed in the standard XPath notation):
   `weblogic-ejb-jar / weblogic-enterprise-bean`

5. Choose the `weblogic-enterprise-bean` node for which the `ejb-name` node is set to the value `ImageToPDFInvoker`.

6. Further navigate down to `stateless-session-descriptor / pool` and set the `max-beans-in-free-pool and initial-beans-in-free-pool` nodes to the appropriate value (that is, the new pool size for Image2PDF conversions). This new value is dependant on the hardware your application server runs on.

7. Repeat steps 5 and 6 for PS2PDF pool size (if required). The `ejb-name` node for PS2PDF conversions is set to `PSToPDFInvoker`.

8. Save your changes, and repackage the entire EAR structure in the original form.

9. Redeploy the modified EAR file in WebLogic Server. You can use WebLogic Server Administration Console to perform this task. You will be prompted to restart WebLogic.

# Uninstalling EAR files

To redeploy a LiveCycle ES solution component, you must first uninstall the LiveCycle ES-related applications from the application server.

For example, if you have assembled LiveCycle ES solution components again to configure solution component properties differently and the previous adobe-livecycle-weblogic.ear file is already deployed, you must uninstall it before deploying the newly assembled EAR files.

➤ **To uninstall the EAR files:**

1. Start the Administration Server and the Managed Server.

2. Log in to WebLogic Server Administration Console by typing `http://localhost:7001/console` in the URL line of a web browser.

3. Under Domain Structure, click **Deployments** and then click **Lock and Edit**.

4. On the **Control** tab, select the LiveCycle ES applications you want to remove.

5. Click **Stop** > **Force Stop Now**.

6. Click **Delete** for the stopped application, click **Yes**, and then click **Continue**.

7. Repeat steps 4 to 6 for the other deployed LiveCycle ES components.

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

9. Stop and start the Administration Server and the Managed Server.

After uninstalling the EAR file, you may need to reconfigure your EAR file and redeploy, see "Configuring LiveCycle ES EAR files" on page 58.

# Next steps

If you used LiveCycle Configuration Manager to configure and deploy LiveCycle ES, you can now do the following tasks:

● Access LiveCycle Administration Console. (See "Accessing LiveCycle Administration Console" on page 65.)

● Configure LiveCycle ES solution components to access LDAP. (See "Configuring LiveCycle ES to access LDAP" on page 67.)

# 9 | Post-Deployment Activities

This chapter describes how to verify the deployment by accessing LiveCycle Administration Console and checking the application server log files. It also describes how to get started using LiveCycle ES solution components and services after they are installed, configured, and deployed to your application server:

After you configure the settings in this chapter, see *Administering LiveCycle ES* at http://www.adobe.com/go/learn_lc_administration for additional information about configuring your LiveCycle ES environment for development and production.

## Configuring Acrobat 8.1 for PDF Generator ES

**Note:** This functionality is supported only on the Windows platform.

If you did not choose to install Acrobat 8.1 using the LiveCycle ES installer, the following steps need to be completed to set up Acrobat 8.1 for use with PDF Generator ES.

➤ **To configure Acrobat 8.1 for use with PDF Generator ES:**

This procedure is required only if you upgraded to or installed Acrobat 8.1 after completing the LiveCycle ES installation. It can be completed before or after you run LiveCycle Configuration Manager and deploy LiveCycle ES to the application server.

1.  If a previous version (8.0 or earlier) of Acrobat is installed, uninstall Acrobat using Add or Remove Programs in the Windows Control Panel.

2.  Navigate to one of the following folders on the LiveCycle ES installation media:
    - additional\acrobat\efg\
    - additional\acrobat\jpn\

3.  Install Acrobat 8.1 by running the AutoPlay.exe file.

4.  On the LiveCycle ES installation medium, navigate to the additional\scripts folder.

5.  Run the following batch file:

```
Acrobat_for_PDFG_Configuration.bat [LiveCycleES root]
```

➤ **To validate the Acrobat 8.1 installation:**

1. Navigate to a PDF file on your system and double-click it to open it in Acrobat.

2. If the PDF file opens successfully, Acrobat 8.1 is installed correctly. If the PDF does not open correctly, uninstall Acrobat and reinstall it.

**Note:** Ensure that you dismiss all the Acrobat dialog boxes that appear after the Acrobat installation is completed and disable the automatic updates for Acrobat.

Set the environment variable `Acrobat_PATH` to point to Acrobat.exe (for example, C:\Program Files\Adobe\Acrobat 8.0\Acrobat\Acrobat.exe).

# Final setup for LiveCycle Rights Management ES

LiveCycle Rights Management ES requires the application server to be configured to use SSL. For details, see *Administering LiveCycle ES* at http://www.adobe.com/go/learn_lc_administration.

# Verifying the deployment

You can verify the deployment by logging in to LiveCycle Administration Console. If you can log in, LiveCycle ES is running on the application server and the default user has been created in the database.

You can review the application server log files to ensure that components were deployed correctly or to determine the cause of any deployment issues you may encounter.

## Accessing LiveCycle Administration Console

LiveCycle Administration Console is the web-based portal for accessing a variety of configuration pages that let you set run-time properties that control the way LiveCycle ES operates. When you log in to LiveCycle Administration Console, you can access User Management, Watched Folder and Email client configuration, and administrative configuration options for other services. LiveCycle Administration Console also provides access to Archive Administration, which administrators use for managing archives and deploying services to a production environment.

The default user name and password for logging in to LiveCycle Administration Console is *administrator* and *password*. After you log in the first time, access User Management and change the password.

Before you access LiveCycle Administration Console, LiveCycle ES must be deployed and running on your application server.

For information about using LiveCycle Administration Console, see *Archive Administration Help* (available from the Help menu of the LiveCycle Administration Console window).

➤ **To access LiveCycle Administration Console:**

1. Type the following URL in a web browser:

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

The default port number for WebLogic Server is 7001. (When you created a new managed server, you may have set a different port.)

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 service administration pages or **Settings** to access the pages on which you can administer settings for different solution components.

## Viewing the log files

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

# Accessing solution component web applications

After LiveCycle ES is deployed, you can access the web applications associated with the following solution components:

*   LiveCycle Reader Extensions ES
*   LiveCycle Workspace ES
*   LiveCycle Rights Management ES

After accessing the web applications using the default administrator permissions to ensure that they are accessible, you can create additional users and roles so that others can log in and use the applications. For information, see *User Management Help*, accessible through the LiveCycle Administration Console.

**Note:** To access the Reader Extensions ES web application, you must apply a Reader Extensions ES credential and apply the user roles for a new user. For more information, see the "Configuring SSL" chapter in *Administering LiveCycle ES* at http://www.adobe.com/go/learn_lc_administration.

➤  **To access the Reader Extensions ES web application:**

1.  Open a web browser and enter the URL http://*[server name]*/ReaderExtensions.

2.  Log in using the default user name and password:

    **User name**: administrator

    **Password**: password

**Note:** You must have administrator or superuser privileges to log in using the default user name and password. To allow other users to access the Reader Extensions ES web application, you must create the users in User Management and grant them the role "Reader Extensions Web Application".

➤  **To access Workspace ES:**

1.  Open a web browser and enter the URL http://*[server name]*/workspace.

2.  Log in using the default user name and password:

    **User name**: administrator

    **Password**: password

## Accessing Rights Management ES

You must create a user account with the role "RM end user console" in User Management and log in to the Rights Management ES administrator or end-user applications using the login information associated with that user.

### ➤ To access the Rights Management ES end-user web application

1. Open a web browser and enter the URL http://*[server name]:[port]*/edc/Login.do.

   Where *[port]* is the port assigned to the Managed WebLogic Server.

   **Note:** The administrator user cannot access the Rights Management ES end-user web application. You can create new users using the WebLogic Server Administration Console.

### ➤ To access the Rights Management ES administration web application

1. Open a web browser and enter the URL http://*[server name]:[port]*/adminui.

   Where *[port]* is the port assigned to the Managed WebLogic Server.

2. Navigate to the Rights Management ES administration web application.

# Accessing User Management

User Management allows administrators to maintain a database of all users and groups, synchronized with one or more third-party user directories. User Management provides authentication, authorization, and user management for LiveCycle ES solution components, including Reader Extensions ES, Workspace ES, Rights Management ES, Process Management ES, and Forms ES.

### ➤ To access User Management:

1. Open a web browser and enter the URL http://*[server name]:[port]*/adminui.

   Where *[port]* is the port assigned to the Managed WebLogic Server.

2. On the LiveCycle Administration Console Home page of, click **Settings**.

3. 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.

# Configuring LiveCycle ES to access LDAP

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

### ➤ To configure User Management with LDAP (Enterprise Domain):

1. Open a web browser, navigate to `http://[host name]:[port]/adminui`, and log in. (See .)

2. Click **Settings** > **User Management** > **Domain Management** and then click **New Enterprise 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**.

6.  Click **OK** and then click **Add Directory**.

7.  Under Profile Name, choose a name for your LDAP profile and then click **Next**.

8.  Specify values in the **Server**, **Port**, **SSL**, and **Binding** boxes as required. For details about the settings, Navigate to **Managing Domains** > **Configuring directories** > **Directory settings** in *User Management Help*.

9.  Under **Populate Page With**, select a directory settings option, such as **Default Sun ONE values**.

10. Click **Next** and configure the **User Settings** as required. For details about the settings, see "Directory settings" in *User Management Help*.

11. Click **Next** and configure the **Group Settings** as required. For details on the settings, see "Directory settings" in *User Management Help*.

12. (Optional) Test your configuration:

    ●  Click **Test**.

    ●  In 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 details will be displayed, and then click **Back**.

13. Click **Finish** to exit the New Directory page and then click **OK** to exit.

➤ **To configure User Management with LDAP (Local Domain):**

1.  Open a web browser, navigate to `http://[host name]:[port]/adminui`, and log in. (See .)

2.  Click **Settings** > **User Management** > **Domain Management** and then click **New Local Domain**.

# Setting PDF Generator ES Watched Folder performance parameters

To avoid java.io.IOException error messages indicating that not enough disk space is available to perform PDF conversions using a watched folder, you can modify the settings for PDF Generator ES in LiveCycle Administration Console.

➤ **To set performance parameters for PDF Generator ES:**

1.  Log in to LiveCycle Administration Console and click **Services** > **Archive Administration** > **Service Management**.

2.  In the list, click **PDFGConfigService** and set the following values:

    **PDFG Cleanup Scan Seconds**: 1800

    **Job Expiration Seconds**: 6000

3. Increase **Server conversion timeout** from the default of 270 to a higher value, such as 450.

4. Click **Save** and then restart the cluster.

# Configuring FIPS mode

LiveCycle ES provides a FIPS mode to restrict data protection to Federal Information Processing Standard (FIPS) 140-2 approved algorithms using the RSA BSAFE Crypto-C 2.1 encryption module.

If you did not enable this option with LiveCycle Configuration Manager during the LiveCycle ES configuration or if you enabled it but want to turn it off, you can change this setting through LiveCycle Administration Console.

Modifying FIPS mode requires you to restart the server.

FIPS mode does not support Acrobat versions earlier than 7.0. If FIPS mode is enabled, LiveCycle ES processes that contain encryption operations configured to support Acrobat versions earlier than 7.0 will fail.

In general, when FIPS is enabled, the Assembler service will not apply password encryption to any document.  If this is attempted, a FIPSModeException is thrown indicating that "`Password encryption is not permitted in FIPS mode.`" Additionally, the Document Description XML (DDX) `PDFsFromBookmarks` element is not supported in FIPS mode when the base document is password-encrypted.

Additionally, the Document Description XML (DDX) `PDFsFromBookmarks` element is not supported in FIPS mode when the base document is password-encrypted.

➤ **To turn FIPS mode on or off:**

1. Log in to LiveCycle Administration Console.

2. Click **Settings** > **Core System Settings** > **Core Configurations** > **Configurations**.

3. Select **Enable FIPS** to enable FIPS mode or deselect it to disable FIPS mode.

4. Click **OK**.

5. Restart the cluster.

**Note:** LiveCycle ES software does not validate code to ensure FIPS compatibility. It provides a FIPS operation mode so that FIPS-approved algorithms are used for cryptographic services from the FIPS-approved libraries (RSA).

# Configuring HTML digital signature

To use the HTML digital signature feature of LiveCycle Forms ES, you must complete the following procedure.

➤ **To enable HTML digital signature:**

1. Manually deploy the *[LivecycleES root]*/deploy/adobe-forms-ds.ear file to your application server.

2. Log in to LiveCycle Administration Console.

3. Click **Service > Livecycle Forms ES** and select **HTML Digital Signature Enabled**.

# Modifying the SAML token threshold attribute

You must modify the SAML token threshold to a suitable value for a clustered environment.

➤ **To modify the SAML token threshold attribute:**

1. Log in to LiveCycle Administration Console and click **Settings** > **User Management** > **Configuration** > **Import and export configuration files**.

2. Click **Export** to download the config.xml file and then, in a text editor, open the downloaded file.

3. In the file, find the string `assertionThresholdInSeconds` and modify it as follows:

   ```
   <entry key="assertionThresholdInSeconds" value="60"/>
   ```

4. Save the edited config.xml file.

5. In LiveCycle Administration Console, click **Settings** > **User Management** > **Configuration** > **Import and export configuration files**.

6. Browse to the location of the edited config.xml file, select the file, and click **Import**.

   A message confirms a successful import of the modified config.xml file.

# Uninstalling LiveCycle ES

The uninstall program located in the *[LivecycleES root]* directory does not remove any files that you deployed to your application server.

**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. To invoke the uninstall program:

   - (Windows) Complete the following steps:
     - You can use Add or Remove Programs in the Windows Control Panel.
       - Remove Adobe LiveCycle ES
       - Remove Adobe Acrobat 8.1 Professional (if installed with PDF Generator ES)
     - Alternatively, you can complete these manual steps:
       - cd *[LivecycleES root]*/_uninst/server
       - Double-click the livecycle8_uninstall.exe file.
   - (Linux, UNIX) From a terminal, type: ./livecycle8_uninstall.bin (You may need to make this binary an executable file)

2. Follow the on-screen instructions in the uninstall program, and then click **Finish.**

# 10 | Configuring Load Balancing

You can configure your WebLogic Server cluster to provide load balancing.

With the domain configuration complete, using the administration server and clustered managed servers, you need a method to proxy in order to forward to different managed servers for load balancing. You can implement a proxy server for WebLogic by using one of these mechanisms:

- An instance of WebLogic Server and HttpClusterServlet (See http://e-docs.bea.com/wls/docs92/plugins/index.html.)
- A third-party proxy server, such as Apache, Microsoft IIS, or Netscape IPlanet, with a WebLogic proxy plug-in. (See http://e-docs.bea.com/wls/docs92/plugins/index.html.)
- A hardware-based load balancer such as F5's BigIP or Cisco's LocalDirector.

## Configuring an Apache server plug-in

You can install and configure an Apache HTTP server plug-in to provide load balancing in your cluster. The Apache HTTP server plug-in proxies requests from an Apache server to a WebLogic Server cluster.

Perform the following tasks:

- Install the Apache HTTP server plug-in. (See "Installing the Apache HTTP server plug-in" on page 71.)
- Configure the Apache HTTP server plug-in. (See "Configuring the Apache HTTP server plug-in" on page 71.)
- Test the Apache HTTP server plug-in. (See "Testing the Apache HTTP server plug-in" on page 72.)

### Installing the Apache HTTP server plug-in

These plug-ins are available as part of WebLogic Server installation. For Windows, the WebLogic plug-in files called mod_wl_20.so and mod_wl28.so are available in the *[appserverdomain]*\server\plugin\*<os>*\*<processor>* directory. If the plug-in directory is not visible, reinstall WebLogic and select the Web Server Plug-Ins component to install.

**Note:** The mod_wl28_20.so file is used for 128-bit encryption. To install the plug-in, copy the mod_wl_20.so (or mod_wl28.so) file to the *[APACHE_HOME]*\modules directory.

### Configuring the Apache HTTP server plug-in

To configure the Apache HTTP server plug-in, you need to modify the configuration file.

➤ **To configure the Apache HTTP server plug-in:**

1. Using a text editor, open *[APACHE_HOME]*\conf\httpd.conf.

2. Add the following line:

```
LoadModule weblogic_module modules\mod_wl_20.so
```

3.  Add an IfModule block as follows:

```
<IfModule mod_weblogic.c>
WebLogicCluster <Server1>:8001,<Server2>:8001
MatchExpression *
</IfModule>
```

> **Note:** If computer names do not work, use the IP addresses. Add any additional computer names to the list, separating each with a comma.

4.  Start/restart Apache HTTP server.

> **Note:** If your cluster implements SSL, see the WebLogic documentation at http://e-docs.bea.com/wls/docs92/plugins/index.html for information about configuring the Apache plug-in using SSL.

## Testing the Apache HTTP server plug-in

You can use LiveCycle ES to test the Apache HTTP server plug-in.

➤ **To test the Apache HTTP server plug-in:**

1.  Open a browser and enter the URL <URL of the Apache server>/adminui.

2.  Log in to LiveCycle ES and check the log files of the managed servers in the cluster for a response success message to determine which server of the cluster serviced the request.

3.  Using a new browser window and a different server of your cluster, successively repeat steps 1 to 2 to verify that requests are forwarded in round-robin manner to different servers of the cluster.

    The response success messages confirm that you have accessed the various servers in the cluster.

➤ **To test HTTP session replication:**

1.  Log in to LiveCycle ES and make note of the server that processes the request.

2.  Open a new browser window and log in to the WebLogic Server Administration Console.

3.  Under Domain Structure, click **Environment** > **Servers** and click the server name noted in step 1.

4.  In the Control Tab, click **Graceful Shutdown** for this server.

5.  Check the server log for a message indicating that the session replicated.

6.  After the server has successfully completed a graceful shutdown, return to the server page accessed in step 1 and click any link.

    Display of the page to which you linked confirms that the session replicated successfully.

## Setting up WebLogic for load balancing

For complete instructions on setting up WebLogic for load balancing, see http://e-docs.bea.com/wls/docs92/cluster/load_balancing.html.

# Configuring message-driven beans

The LiveCycle ES uses message-driven beans (MDBs) and a distributed JMS queue.

Each node maintains a pool of MDB instances that are available to process job requests in the queue, and attempts to keep the MDB pool fully utilized (that is, it will pull JMS requests for all of its idle MDB instances). Light load conditions can result in uneven load balancing across nodes of the cluster because a single node will process multiple requests to fully utilize all MDB instances available in its pool. Heavy load conditions will balance the load more evenly across the cluster because more nodes of the cluster will have fully utilized their MDB pool.

Reducing the MDB pool size to a small number will result in more even load distribution under light load conditions at the expense of reduced throughput under heavy loads. For additional information about MDB parameters and instructions to change them, see
http://e-docs.bea.com/wls/docs92/perform/mdbtuning.html.