Hardening and Security for LiveCycle® ES2
# Contents

**About This Document** ......................................................................................................................... 5
  - Who should read this document? ........................................................................................................ 5
  - Conventions used in this document .................................................................................................. 5
  - Additional information ...................................................................................................................... 6

## 1 General Security Considerations .................................................................................................... 7
  - Vendor-specific security information ............................................................................................... 7
  - Operating system security information ............................................................................................ 7
  - Application server security information ........................................................................................... 8
  - Database security information ......................................................................................................... 8
  - Configuring JBoss to use a non-default HTTP port .......................................................................... 10
  - LiveCycle ES2 security considerations .......................................................................................... 10
  - Email credentials not encrypted in database .................................................................................. 11
  - Sensitive content for LiveCycle Rights Management ES2 in the database ................................... 11
  - Password in clear text format in adobe-ds.xml .............................................................................. 11

## 2 Hardening Your Environment ......................................................................................................... 12
  - Preinstallation ..................................................................................................................................... 12
  - Network layer security .................................................................................................................... 13
  - Operating system security ............................................................................................................... 14
  - Installation .......................................................................................................................................... 14
  - Post-installation steps ....................................................................................................................... 15
  - LiveCycle ES2 server security ......................................................................................................... 15
    - Restricting LiveCycle Content Services ES2 user data check-in quotas .................................... 18
    - Application server security ............................................................................................................ 19
    - Using JMX Console on JBoss ......................................................................................................... 20
    - Database security ........................................................................................................................... 21
      - Configuring integrated security on Windows ............................................................................. 21
    - Protecting access to sensitive content in the database ............................................................... 22
    - LDAP security ............................................................................................................................... 23
    - Auditing and logging ...................................................................................................................... 24
    - LiveCycle ES2 Unix system library dependencies ...................................................................... 24
      - Convert PDF service .................................................................................................................... 24
      - XMLForms .................................................................................................................................. 25
    - Configuring LiveCycle ES2 for access beyond the enterprise .................................................... 27
      - Setting up a reverse proxy for web access .................................................................................. 27
  - Secure network configuration ........................................................................................................... 29
    - LiveCycle ES2 physical architecture ............................................................................................... 29
    - Network protocols used by LiveCycle ES2 .................................................................................. 30
    - Ports for application servers ......................................................................................................... 30
    - Configuring SSL ............................................................................................................................. 32
    - Configuring SSL redirect .............................................................................................................. 32
    - Windows-specific security recommendations ............................................................................... 32
      - JBoss Service accounts ............................................................................................................... 33
      - File system security .................................................................................................................... 33
      - JBoss-specific security recommendations .................................................................................. 33
3 Configuring Secure Administration Settings ................................................................. 36
Disabling non-essential remote access to services............................................................ 36
Disabling non-essential anonymous access to services..................................................... 37
Remove sample user and role assignments ...................................................................... 38
Changing the default global time-out ............................................................................... 39
Disabling LiveCycle 7.x backwards-compatibility API access ........................................ 39
### About This Document

This document contains information about how to maximize the security of the Adobe® LiveCycle® ES2 (Enterprise Suite) version 9.0 production environment.

Additional security information for LiveCycle ES2 is available at the [LiveCycle Developer Center](#).

Security advisories and bulletins for LiveCycle ES2 are available at the [Security bulletins and advisories](#) site.

### Who should read this document?

This document is intended for consultants, security specialists, systems architects, and IT professionals who are responsible for planning application or infrastructure development and deployment of LiveCycle ES2. These roles include the following common roles:

- IT and operations' engineers who must deploy secure web applications and servers in their own or customer organizations
- Architects and planners who are responsible for planning the architectural efforts for the clients in their organizations
- IT security specialists who focus on providing security across the platforms within their organizations
- Consultants from Adobe and partners who require detailed resources for customers and partners

### Conventions used in this document

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="#" alt="LiveCycle ES2 root" /></td>
<td>C:\Adobe\Adobe LiveCycle ES2\</td>
<td>The installation directory that is used for all LiveCycle ES2 solution components. This directory contains subdirectories for LiveCycle Configuration Manager, the LiveCycle ES2 SDK, and each installed LiveCycle ES2 solution component.</td>
</tr>
<tr>
<td><img src="#" alt="JBoss_ES2 root" /></td>
<td>C:\Adobe\Adobe LiveCycle ES2\jboss</td>
<td>The home directory of the application server that runs LiveCycle ES2</td>
</tr>
</tbody>
</table>
Additional information

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

<table>
<thead>
<tr>
<th>For information about</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>LiveCycle ES2, the solution components and development tools</td>
<td>LiveCycle ES2 Overview</td>
</tr>
</tbody>
</table>
| Preparing your environment for installing or upgrading to LiveCycle ES2 | Preparing to Install LiveCycle ES2  
Preparing to Upgrade to LiveCycle ES2 |
| Installing LiveCycle ES2 | Installing and Deploying LiveCycle ES2 for JBoss  
Installing and Deploying LiveCycle ES2 for WebSphere  
Installing and Deploying LiveCycle ES2 for WebLogic |
| Upgrading to LiveCycle ES2 using the non-turnkey method | Upgrading to LiveCycle ES2 for JBoss  
Upgrading to LiveCycle ES2 for WebSphere  
Upgrading to LiveCycle ES2 for WebLogic |
| Installing LiveCycle Workbench ES2 | Installing Your Development Environment |
| Performing general administrative tasks for LiveCycle ES2 | LiveCycle ES2 Administration Help |
| Other services and products that integrate with LiveCycle ES2 | Adobe LiveCycle ES2 |
| Patch updates, technical notes, and additional information about this product version | LiveCycle Support Center |
| LiveCycle ES2 terminology | LiveCycle ES2 Glossary |
General Security Considerations

This section provides introductory information that helps you prepare for hardening your LiveCycle ES2 environment. It includes prerequisite information about LiveCycle ES2, operating system, application server, and database security. You should review this information before you continue to lock down your environment.

Vendor-specific security information

This section contains security-related information about operating systems, application servers, and databases that are incorporated into your LiveCycle ES2 enterprise solution.

Use the links in this section to find vendor-specific security information for your operating system, database, and application server.

Operating system security information

When securing your operating system, carefully consider implementing the measures described by your operating system vendor, including these:

- Defining and controlling users, roles, and privileges
- Monitoring logs and audit trails
- Removing unnecessary services and applications
- Backing up files

For security information about operating systems that LiveCycle ES2 supports, see the resources in the this table.

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Security Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM® AIX® 5.3 and 6.1</td>
<td>IBM AIX Security Benefits</td>
</tr>
<tr>
<td>Microsoft® Windows® XP SP 2 (for non-production environments only)</td>
<td>Windows XP Security Guide</td>
</tr>
<tr>
<td>Microsoft Windows 7, 32-bit and 64-bit (for non-production environments only)</td>
<td>Windows 7 Security Guide</td>
</tr>
<tr>
<td>Microsoft Vista™ SP1, all flavors, 32-bit and 64-bit (for non-production environments only)</td>
<td>Search for “Windows Vista Security Guide” at <a href="http://www.microsoft.com">www.microsoft.com</a></td>
</tr>
<tr>
<td>Red Hat® Linux® AP or ES</td>
<td>Red Hat Enterprise Linux Security Guide</td>
</tr>
</tbody>
</table>
When securing your application server, you should carefully consider implementing the measures described by your server vendor, including these:

- Using non-obvious administrator user name
- Disabling unnecessary services
- Securing the console manager
- Enabling secure cookies
- Closing unneeded ports
- Limiting clients by IP addresses or domains
- Using the Java™ Security Manager to programmatically restrict privileges

For security information about application servers that LiveCycle ES2 supports, see the resources in this table.

<table>
<thead>
<tr>
<th>Application Server</th>
<th>Security Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle WebLogic® 10g R3</td>
<td>Search for Understanding WebLogic Security at <a href="http://download.oracle.com/docs/">http://download.oracle.com/docs/</a></td>
</tr>
<tr>
<td>IBM WebSphere® 6.1 or 7.0</td>
<td>Securing applications and their environment (version 6.1)</td>
</tr>
<tr>
<td></td>
<td>Securing applications and their environment (version 7.0)</td>
</tr>
<tr>
<td>Red Hat® JBoss® 4.2.0 or 4.2.1</td>
<td>Security on JBoss</td>
</tr>
</tbody>
</table>

When securing your database, you should consider implementing the measures described by your database vendor, including these:

- Restricting operations with access control lists (ACLs)
- Using non-standard ports
- Hiding the database behind a firewall
- Encrypting sensitive data before writing it to the database (see the database manufacturer’s documentation)

For security information about databases that LiveCycle ES2 supports, see the resources in this table.
This table describes the default ports that are required to be open during your LiveCycle ES2 configuration process. If you are connecting over https, adjust your port information and IP addresses accordingly. For more information about configuring ports, see the Installing and Deploying LiveCycle ES2 document for your application server.

<table>
<thead>
<tr>
<th>Database</th>
<th>Security Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM DB2® 9.1 or 9.5</td>
<td>DB2 Product Family</td>
</tr>
<tr>
<td>MySQL 5</td>
<td>MySQL 5.0 General Security Issues MySQL 5.1 General Security Issues</td>
</tr>
<tr>
<td>Oracle® 10g or 11g</td>
<td>Security Considerations and Requirements (version 10g) See the Security chapter in the Oracle 11g Documentation Library</td>
</tr>
</tbody>
</table>

This table describes the default ports that are required to be open during your LiveCycle ES2 configuration process. If you are connecting over https, adjust your port information and IP addresses accordingly. For more information about configuring ports, see the Installing and Deploying LiveCycle ES2 document for your application server.

<table>
<thead>
<tr>
<th>Product or service</th>
<th>Port number</th>
</tr>
</thead>
<tbody>
<tr>
<td>JBoss</td>
<td>8080</td>
</tr>
<tr>
<td>WebLogic</td>
<td>7001</td>
</tr>
<tr>
<td>WebLogic Managed Server</td>
<td>Set by administrator during configuration</td>
</tr>
<tr>
<td>WebSphere</td>
<td>9060, if Global Security is enabled the default SSL port value is 9043. 9080</td>
</tr>
<tr>
<td>BAM Server</td>
<td>7001</td>
</tr>
<tr>
<td>SOAP</td>
<td>8880</td>
</tr>
<tr>
<td>MySQL</td>
<td>3306</td>
</tr>
<tr>
<td>Oracle</td>
<td>1521</td>
</tr>
<tr>
<td>DB2</td>
<td>50000</td>
</tr>
<tr>
<td>SQL Server</td>
<td>1433</td>
</tr>
<tr>
<td>LDAP</td>
<td>The port on which the LDAP server is running. The default port is typically 389. However, if you select the SSL option, the default port is typically 636. You must confirm with your LDAP administrator which port to specify.</td>
</tr>
</tbody>
</table>

Configuring JBoss to use a non-default HTTP port

JBoss Application Server uses 8080 as the default HTTP port. JBoss also has pre-configured ports 8180, 8280, and 8380, which are commented out in the jboss-service.xml file. If you have an application on your computer that already uses this port, change the port that LiveCycle ES2 uses by following these steps:
1. Open the jboss-service.xml file in an editor.
   JBoss turnkey install: [JBossES2 root]/server/lc_turnkey/conf/
   JBoss manual install: [appserver root]/server/all/conf/

2. Locate and uncomment the following mbean:
   
   ```xml
   <mbean code="org.jboss.services.binding.ServiceBindingManager"
   name="jboss.system:service=ServiceBindingManager">
   <attribute name="ServerName">ports-01</attribute>
   
   <attribute name="StoreURL">${jboss.home.url}/docs/examples/binding-manager/sample-bindings.xml</attribute>
   
   <attribute name="StoreFactoryClassName">
       org.jboss.services.binding.XMLServicesStoreFactory
   </attribute>
   </mbean>
   
3. Save and close the file.

4. Restart JBoss.

   JBoss is now configured to use port 8180. If you need to use either 8280 or 8380, modify the ServerName attribute value to use one of the following alternative ports:
   
   For 8280: ports-02
   For 8380: ports-03

   If you need to configure a port number other than those pre-configured for JBoss, perform the following steps:
   
   1. Locate and open the deploy/jboss-web.deployer file in [JBossES2 root] (turnkey) or [appserver root] (JBoss manual install).
   
   2. Locate and uncomment the mbean from step 2 above.
   
   3. Modify the ServerName value to the port number to use.
   
   4. Save and close the file.
   
   5. Restart JBoss.

LiveCycle ES2 security considerations

This section describes some LiveCycle ES2-specific security issues that you should know about.

Email credentials not encrypted in database

The email credentials stored by LiveCycle ES2 applications are not encrypted before they are stored in the LiveCycle ES2 database. When you configure a service endpoint to use email, any password information used as part of that endpoint configuration is not encrypted when it is stored in the database.
Sensitive content for LiveCycle Rights Management ES2 in the database

LiveCycle ES2 uses the LiveCycle ES2 database to store sensitive document key information as well as other cryptographic material that is used for policy documents. Securing the database against intrusion helps to protect this sensitive information.

Password in clear text format in adobe-ds.xml

The application server that is used to run LiveCycle ES2 requires its own configuration for access to your database through a data source that is configured on the application server. You should ensure that your application server does not expose your database password in clear text in its data source configuration file.

The adobe-ds.xml file contains passwords in clear text format. Consult your application server vendor about how to encrypt these passwords for your application server. For example, the JBoss® instructions are at Encrypting DataSource Passwords.

Note: The LiveCycle ES2 JBoss turnkey installer encrypts the database password.

IBM WebSphere Application Server and Oracle WebLogic Server may encrypt data source passwords by default. However, you should confirm with your application server documentation to ensure that this is happening.
Hardening Your Environment

This section describes recommendations and best practices for securing servers that run LiveCycle ES2. This is not a comprehensive host-hardening document for your operating system and application server. Instead, this section describes a variety of security-hardening settings that you should implement to enhance the security of LiveCycle ES2 that is running within a corporate intranet. To ensure that the LiveCycle ES2 application servers stay secure, however, you should also implement security monitoring, detection, and response procedures.

This section describes hardening techniques that should be applied during the following stages during the installation and configuration life cycle:

- **Preinstallation**: Use these techniques before you install the LiveCycle ES2 software.
- **Installation**: Use these techniques during the LiveCycle ES2 software installation process.
- **Post-installation**: Use these techniques after installation and periodically thereafter.

LiveCycle ES2 is highly customizable and can work in many different environments. Some of the recommendations may not fit your organization’s needs.

**Preinstallation**

Before installing LiveCycle ES2, you can apply security solutions to the network layer and operating system. This section describes some issues and makes recommendations for reducing security vulnerabilities in these areas.

**Installation and configuration on UNIX and Linux**

You should not install or configure LiveCycle ES2 using a root shell. By default, files are installed under the /opt directory, and the user who performs the installation needs all file permissions under /opt. Alternatively, an installation can be performed under an individual user’s /user directory where they already have all file permissions.

**Installation and configuration on Windows**

You should perform the installation on Windows as an administrator if you are installing LiveCycle ES2 on JBoss by using the turnkey method or if you are installing LiveCycle PDF Generator ES2. Also, when installing PDF Generator ES2 on Windows with native application support, you must run the installation as the same Windows user who installed Microsoft Office. For more information about installation privileges, see the Installing and Deploying LiveCycle ES2 document for your application server.
Network layer security

Network security vulnerabilities are among the first threats to any Internet-facing or intranet-facing application server. This section describes the process of hardening hosts on the network against these vulnerabilities. It addresses network segmentation, Transmission Control Protocol/Internet Protocol (TCP/IP) stack hardening, and the use of firewalls for host protection.

The following table describes common processes that reduce network security vulnerabilities.

<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demilitarized zones (DMZs)</td>
<td>Deploy LiveCycle ES2 servers within a demilitarized zone (DMZ). Segmentation should exist in at least two levels with the application server used to run LiveCycle ES2 placed behind the inner firewall. Separate the external network from the DMZ that contains the web servers, which in turn must be separated from the internal network. Use firewalls to implement the layers of separation. Categorize and control the traffic that passes through each network layer to ensure that only the absolute minimum of required data is allowed.</td>
</tr>
<tr>
<td>Private IP addresses</td>
<td>Use Network Address Translation (NAT) with RFC 1918 private IP addresses on LiveCycle ES2 application servers. Assign private IP addresses (10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16) to make it more difficult for an attacker to route traffic to and from a NAT’d internal host through the Internet.</td>
</tr>
<tr>
<td>Firewalls</td>
<td>Use the following criteria to select a firewall solution:</td>
</tr>
<tr>
<td></td>
<td>● Implement firewalls that support proxy servers and/or stateful inspection instead of simple packet-filtering solutions.</td>
</tr>
<tr>
<td></td>
<td>● Use a firewall that supports a deny all services except those explicitly permitted security paradigm.</td>
</tr>
<tr>
<td></td>
<td>● Implement a firewall solution that is dual-homed or multihomed. This architecture provides the greatest level of security and helps to prevent unauthorized users from bypassing the firewall security.</td>
</tr>
<tr>
<td>Database ports</td>
<td>Do not use default listening ports for databases (MySQL - 3306, Oracle - 1521, MS SQL - 1433). For information about changing database ports, see your database documentation.</td>
</tr>
<tr>
<td></td>
<td><strong>Caution:</strong> Using a different database port affects the overall LiveCycle ES2 configuration. If you change default ports, you must make corresponding modifications in other areas of configuration, such as the data sources for LiveCycle ES2.</td>
</tr>
<tr>
<td></td>
<td>For information about configuring data sources in LiveCycle ES2, see <em>Installing and Deploying LiveCycle ES2</em> or <em>Upgrading to LiveCycle ES2</em> for your application server, at Adobe LiveCycle ES2 Documentation.</td>
</tr>
</tbody>
</table>
Operating system security

The following table describes some potential approaches to minimizing security vulnerabilities found in the operating system.

<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security patches</td>
<td>There is an increased risk that an unauthorized user may gain access to the application server if vendor security patches and upgrades are not applied in a timely fashion. Test security patches before you apply them to production servers. Also, create policies and procedures to check for and install patches on a regular basis.</td>
</tr>
<tr>
<td>Virus protection software</td>
<td>Virus scanners can identify infected files by scanning for a signature or watching for unusual behavior. Scanners keep their virus signatures in a file, which is usually stored on the local hard drive. Because new viruses are discovered often, you should frequently update this file for the virus scanner to identify all current viruses.</td>
</tr>
<tr>
<td>Network Time Protocol (NTP)</td>
<td>For forensic analysis, keep accurate time on LiveCycle ES2 servers. Use NTP to synchronize the time on all systems that are connected directly to the Internet.</td>
</tr>
</tbody>
</table>

For additional security information for your operating system, see “Operating system security information” on page 7.

Installation

This section describes techniques you can use during the LiveCycle ES2 installation process to reduce security vulnerabilities. In some cases, these techniques use options that are part of the installation process. The following table describes these techniques.

<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Privileges</td>
<td>Use the least amount of privileges necessary to install the software. Log in to your computer by using an account that is not in the Administrators group. On Windows, you can use the Run As command to run the LiveCycle ES2 installer as an administrative user. On UNIX and Linux systems, use a command such as sudo to install the software.</td>
</tr>
<tr>
<td>Software source</td>
<td>Do not download or run LiveCycle ES2 from untrusted sources. Malicious programs can contain code to violate security in several ways, including data theft, modification and deletion, and denial of service. Install LiveCycle ES2 from the Adobe DVD or only from a trusted source.</td>
</tr>
<tr>
<td>Disk partitions</td>
<td>Place LiveCycle ES2 on a dedicated disk partition. Disk segmentation is a process that keeps specific data on your server on separate physical disks for added security. Arranging data in this way reduces the risk of directory traversal attacks. Plan to create a partition that is separate from the system partition on which you can install the LiveCycle ES2 content directory. (On Windows, the system partition contains the system32 directory, or boot partition.)</td>
</tr>
</tbody>
</table>
Post-installation steps

After you successfully install LiveCycle ES2, it is important to periodically maintain the environment from a security perspective.

The following section describes in detail the different tasks that are recommended to secure the deployed LiveCycle ES2 server.

LiveCycle ES2 server security

The following recommended settings apply to the LiveCycle ES2 server outside of the administrative web application. To reduce the security risks to the server, apply these settings immediately after installing LiveCycle ES2.

Security patches

There is an increased risk that an unauthorized user might gain access to the application server if vendor security patches and upgrades are not applied in a timely fashion. Test security patches before you apply them to production servers to ensure compatibility and availability of LiveCycle ES2 applications. Also, create policies and procedures to check for and install patches on a regular basis. LiveCycle ES2 updates are on the Enterprise products download site.

Service accounts (JBoss turnkey on Windows only)

LiveCycle ES2 installs a service, by default, by using the LocalSystem account. The built-in LocalSystem user account has a high level of accessibility; it is part of the Administrators group. If a worker-process identity runs as the LocalSystem user account, that worker process has full access to the entire system.

To run the application server on which LiveCycle ES2 is deployed, using a specific non-administrative account, follow these instructions:

1. In the Microsoft Management Console (MMC), create a local user for the LiveCycle ES2 service to log in as:
   - Select User cannot change password.
   - On the Member Of tab, ensure that the Users group is listed.
   
     **Note:** You cannot change this setting for PDF Generator ES2.


<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Components</td>
<td>Evaluate existing services and disable or uninstall any that are not required. Do not install unnecessary components and services. The default installation of an application server might include services that are not necessary for your use. You should disable all unnecessary services prior to deployment to minimize points of entry for an attack. For example, on JBoss, you can comment out unnecessary services in the META-INF/jboss-service.xml descriptor file.</td>
</tr>
<tr>
<td>Backward-compatibility</td>
<td>Do not enable LiveCycle 7.x backward compatibility if it is not required for your deployment.</td>
</tr>
</tbody>
</table>
3. Double-click the JBoss for Adobe LiveCycle ES2 service and stop the service.

4. On the **Log On** tab, select **This Account**, browse for the user account you created, and enter the password for the account.

5. In the MMC, open **Local Security Settings** and select **Local Policies > User Rights Assignment**.

6. Assign the following rights to the user account that LiveCycle ES2 server is running under:
   - Deny log on through Terminal Services
   - Deny log on locally
   - Log on as Service (should be already set)

7. Give the new user account the Read & Execute, List Folder Contents, and Read permissions for the LiveCycle ES2 web content directories item.

8. Start the LiveCycle ES2 Application Server service.

**Disabling the LiveCycle Configuration Manager bootstrap servlet**

LiveCycle Configuration Manager made use of a servlet deployed on your application server to perform bootstrapping of the LiveCycle ES2 database. Because LiveCycle Configuration Manager accesses this servlet before configuration is complete, access to it has not been secured for authorized users, and it should be disabled after you have successfully used LiveCycle Configuration Manager to configure LiveCycle ES2.

1. Unzip the adobe-livecycle-[appserver].ear file.

2. Open the META-INF/application.xml file.

3. Search for the adobe-bootstrapper.war section:

   ```xml
   <!-- bootstrapper start -->
   <module id="WebApp_adobe_bootstrapper">
   <web>
     <web-uri>adobe-bootstrapper.war</web-uri>
     <context-root>/adobe-bootstrapper</context-root>
   </web>
   </module>
   <module id="WebApp_adobe_lcm_bootstrapper_redirector">
   <web>
     <web-uri>adobe-lcm-bootstrapper-redirector.war</web-uri>
     <context-root>/adobe-lcm-bootstrapper</context-root>
   </web>
   </module>
   <!-- bootstrapper end-->
   ```

4. Comment out the adobe-bootstrapper.war and the adobe-lcm-bootstrapper-redirectory.war modules as follows:

   ```xml
   <!-- bootstrapper start -->
   <!--
   <module id="WebApp_adobe_bootstrapper">
   <web>
     <web-uri>adobe-bootstrapper.war</web-uri>
     <context-root>/adobe-bootstrapper</context-root>
   </web>
   </module>
   <module id="WebApp_adobe_lcm_bootstrapper_redirector">
   <web>
     <web-uri>adobe-lcm-bootstrapper-redirector.war</web-uri>
     <context-root>/adobe-lcm-bootstrapper</context-root>
   </web>
   <!--
   ```
5. Save and close the META-INF/application.xml file.

6. Zip the EAR file and redeploy it to the application server.

7. Type the URL into a browser to test the change and ensure that it no longer works.

**Lockdown remote access to the Trust Store**

LiveCycle Configuration Manager lets you upload a LiveCycle Reader Extensions ES2 credential to the LiveCycle ES2 trust store. This means that access to the Trust Store Credential Service over remote protocols (SOAP and EJB) has been enabled by default. This access is no longer necessary after you have uploaded the Rights credential using LiveCycle Configuration Manager or if you decide to use LiveCycle Administration Console later to manage credentials.

You can disable remote access to all of the Trust Store services by following the steps in the section “Disabling non-essential remote access to services” on page 36.

**Disable all non-essential anonymous access**

Some LiveCycle ES2 services have operations that may be invoked by an anonymous caller. If anonymous access to these services is not required, disable it by following the steps in “Disabling non-essential anonymous access to services” on page 37.

**Changing the administrator password**

When LiveCycle ES2 is installed, a single default user account is configured for user Super Administrator/login-id Administrator with a default password of password. You should immediately change this password using the LiveCycle Configuration Manager.

➤ To change the default administrator password:

1. Type the following URL in a web browser:
   http://[host name]:[port]/adminui

   The default port number is one of these:
   - **JBoss**: 8080
   - **WebLogic Server**: 7001
   - **WebSphere**: 9080.

2. In the **User Name** field, type administrator and, in the **Password** field, type password.

3. Click **Settings > User Management > Users and Groups**.

4. Type administrator in the **Find** field, and click **Find**.
5. Click **Super Administrator** from the list of users.

6. Click **Change Password** on the Edit User page.

7. Specify the new password and click **Save**.

**Disabling WSDL generation in production environment**

Web Service Definition Language (WSDL) generation should be enabled only for development environments, where WSDL generation is used by developers to build their client applications. You may choose to disable WSDL generation in a production environment to avoid exposing a service’s internal details.

➤ To disable WSDL generation:

1. Type the following URL in a web browser:
   ```
   http://[host name]:[port]/adminui
   ```

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

3. Uncheck **Enable WSDL** and click **OK**.

**Restricting LiveCycle Content Services ES2 user data check-in quotas**

By default, Content Services ES2 does not restrict on the amount of data a user can check in to the server at any one time. Large amounts of data are potentially threatening to the system as they leave the system without the resources to perform other operations. This situation can cause a denial of service to other incoming processes. Use JVM arguments to enable quota management in Content Services ES2.

**Caution:** These JVM arguments must be passed prior to synchronizing the users. This user quota cannot be modified once the users have been synchronized.

➤ **Enable quota management on Content Services ES2:**

**On JBoss**

1. Navigate to the `[jboss root]/bin` directory and open the startup script in a text editor:
   ```
   ● (Windows) run.bat
   ● (Linux and UNIX) run.sh
   ```

2. Add the following properties below the `Set JAVA_OPTS` argument:
   ```
   -Dsystem.usages.enableQuotaSize=true -Dsystem.usages.quota=[size in KB]
   ```

3. Save and close the file.

4. Restart the JBoss server before synchronizing the users.

**On WebLogic**

1. Access the WebLogic Server Administration Console, type `http://[host name]:[port]/console` in the URL line of a web browser, where `[port]` is the non-secure listening port. By default, this port value is 7001.

2. On the login screen, type your WebLogic user name and password and click **Log In**.
3. Under Change Center, click **Lock & Edit**.

4. Under Domain Structure, click **Environment > Servers** and, in the right pane, click the managed server name.

5. In the Settings for Server pane, click the **Configuration tab > Server Start tab**.

6. In the Arguments box, add the following arguments separated by a space delimiter:
   
   -Dsystem.usages.enableQuotaSize=true
   -Dsystem.usages.quota=[size in KB]
   
7. Click **Save** and then click **Activate Changes**.

8. Restart the WebLogic server before synchronizing the users.

**On WebSphere**

1. In the WebSphere Administrative Console navigation tree, do the following task for your application server:
   
   (WebSphere 6.x) Click **Servers > Application servers**
   
   (WebSphere 7.x) Click **Servers > Server Types > WebSphere application servers**

2. Click the server name in the right pane.


4. Under Additional Properties, click **Java Virtual Machine**.

5. In the **Generic JVM arguments** box, add `-Dsystem.usages.enableQuotaSize=true` and `-Dsystem.usages.quota=<size in KB>`, separated by commas, to the existing properties.

6. Click **OK** or **Apply**, and then click **Save directly to the master configuration**.

7. Restart the WebSphere server before synchronizing the users.

**Application server security**

The following table describes some techniques for securing your application server after the LiveCycle ES2 application is installed.

<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application server administrative console</td>
<td>After you install, configure, and deploy LiveCycle ES2 on your application server, you should disable access to the application server administrative consoles. See your application server documentation for details.</td>
</tr>
</tbody>
</table>
**Using JMX Console on JBoss**

When the Java Management Extensions (JMX) console is installed with JBoss, URLs can be constructed for use as cross-site scripting (XSS) exploits that can reveal sensitive information about your system.

If you installed LiveCycle ES2 by using the turnkey method and are using the version of JBoss that was included with the turnkey installation, the JBoss JMX Console is removed by default to ensure that security risks are minimized. However, if you need to use the JBoss JMX Console, reinstall it by following this procedures.

<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application server cookie settings</td>
<td>Application cookies are controlled by the application server. When deploying the application, the application server administrator can specify cookie preferences on a server-wide or application-specific basis. By default, the server settings take preference. You can restrict cookies to be sent using HTTPS-only. As a result, they are not sent unencrypted over HTTP. Application server administrators should enable secure cookies for the server on a global basis. For example, when using the JBoss Application Server, you can modify the connector element to <code>secure=true</code> in the server.xml file. See your application server documentation for more details.</td>
</tr>
</tbody>
</table>
| Directory browsing | When someone requests a page that does not exist or requests the name of a directory (the request string ends with a forward slash (/)), the application server should not return the contents of that directory. To prevent this, you can disable directory browsing on your application server. You should do this for the LiveCycle Administration Console application and for other applications running on your server.

For JBoss, set the value of the listings initialization parameter of the DefaultServlet property to false in the web.xml file, as shown by this example:

```xml
<servlet>
    <servlet-name>default</servlet-name>
    <servlet-class>
        org.apache.catalina.servlets.DefaultServlet
    </servlet-class>
    <init-param>
        <param-name>listings</param-name>
        <param-value>false</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>
```

For WebSphere, set the `directoryBrowsingEnabled` property in the `ibm-web-ext.xmi` file to false.

For WebLogic, set the index-directories properties in the `weblogic.xml` file to false, as shown by this example:

```xml
<container-descriptor>
    <index-directory-enabled>false
</index-directory-enabled>
</container-descriptor>
```
To enable JBoss JMX Console:

1. Download a copy of JBoss 4.2.0 (or later) from JBoss.org.
2. Stop the JBoss Application Server.
3. From the zipped archive file you downloaded, extract the files from JBoss root/deploy/jmx-console.war/.
4. Place the jmx-console.war/... files in the deploy directory of the JBoss installation directory.
5. Restart JBoss.
6. Go to the following URL to ensure that the JBoss JMX Console is available:
   http://localhost:8080/jmx-console

Database security

When securing your database, you should implement the measures described by your database vendor. You should allocate a database user with the minimum required database permissions granted for use by LiveCycle ES2. For example, do not use an account with database administrator privileges.

On Oracle, the database account that you use needs only the CONNECT, RESOURCE, and CREATE VIEW privileges. For similar requirements on other databases, see Preparing to Install LiveCycle ES2.

Configuring integrated security on Windows

This section applies to SQL Server database and LiveCycle ES2 running on a Windows Server.

On WebSphere, you can configure integrated security only when you use an external SQL Server JDBC driver, not the SQL Server JDBC driver that is embedded with WebSphere.

To use integrated security to make a trusted connection with SQL Server from JBoss:

1. Modify JBOSS_HOME\server\all\deploy\adobe-ds.xml to add integratedSecurity=true to the connection URL, as shown in this example:
   jdbc:sqlserver://<serverhost>:<port>;databaseName=<dbname>;integratedSecurity=true
2. Add the sqljdbc_auth.dll file to the Windows systems path on the computer that is running the application server. The sqljdbc_auth.dll file is located with the Microsoft SQL JDBC 1.2 driver installation (the default is InstallDir/sqljdbc_1.2/enu/auth/x86).
3. Modify JBoss Windows service (JBoss for Adobe LiveCycle ES2) property for Log On As from Local System to a login account that has LiveCycle ES2 database and a minimum set of privileges. If you are running JBoss from the command line instead of as a Windows service, you do not need to perform this step.
4. Set Security for SQL Server from Mixed mode to Windows Authentication only.

To use integrated security to make a trusted connection with SQL Server from WebLogic:

1. Start the WebLogic Server Administration Console by typing the following URL in the URL line of a web browser:
http://[host name]:7001/console

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

3. Under Domain Structure, click **[base_domain] > Services > JDBC > Data Sources** and, in the right pane, click **IDP_DS**.

4. On the next screen, on the **Configuration** tab, click the **Connection Pool** tab and, in the **Properties** box, type `integratedSecurity=true`.

5. Under Domain Structure, click **[base_domain] > Services > JDBC > Data Sources** and, in the right pane, click **RM_DS**.

6. On the next screen, on the **Configuration** tab, click the **Connection Pool** tab and, in the **Properties** box, type `integratedSecurity=true`.

7. Add the sqljdbc_auth.dll file to the Windows systems path on the computer that is running the application server. The sqljdbc_auth.dll file is located with the Microsoft SQL JDBC 1.2 driver installation (the default is `[InstallDir]/sqljdbc_1.2/enu/auth/x86`).

8. Set Security for SQL Server from **Mixed** mode to **Windows Authentication only**.

➤ To use integrated security to make a trusted connection with SQL Server from WebSphere:

1. Log in to the WebSphere Administrative Console.

2. In the navigation tree, click **Resources > JDBC > Data Sources** and, in the right pane, click **IDP_DS**.

3. In the right pane, under Additional Properties, click **Custom Properties**, and then click **New**.

4. In the **Name** box, type `integratedSecurity` and, in the **Value** box, type `true`.

5. In the navigation tree, click **Resources > JDBC > Data Sources** and, in the right pane, click **RM_DS**.

6. In the right pane, under Additional Properties, click **Custom Properties**, and then click **New**.

7. In the **Name** box, type `integratedSecurity` and, in the **Value** box, type `true`.

8. On the computer where WebSphere is installed, add the sqljdbc_auth.dll file to the Windows systems path (C:\Windows). The sqljdbc_auth.dll file is located with the Microsoft SQL JDBC 1.2 driver installation (default is `[InstallDir]/sqljdbc_1.2/enu/auth/x86`).

9. Select **Start > Control Panel > Services**, right-click the Windows service for WebSphere (IBM WebSphere Application Server <version> - <node>) and select **Properties**.

10. In the Properties dialog box, click the **Log On** tab.

11. Select **This Account** and provide the information required to set the login account you want to use.

12. Set Security on SQL Server from **Mixed** mode to **Windows Authentication only**.

Protecting access to sensitive content in the database

The LiveCycle ES2 database schema contains sensitive information about system configuration and business processes and should be hidden behind the firewall. The database should be considered within the same trust boundary as the LiveCycle ES2 server. To guard against information disclosure and theft of
business data, the database must be configured by the database administrator (DBA) to allow access only by authorized administrators.

As an added precaution, you should consider using database vendor-specific tools to encrypt columns in tables that contain the following data:

- Rights Management ES2 Document Keys
- Trust Store HSM PIN encryption key
- Local User Password Hashes

For information about vendor-specific tools, see “Database security information” on page 8.

**LDAP security**

A Lightweight Directory Access Protocol (LDAP) directory is typically used by LiveCycle ES2 as a source for enterprise user and group information, and a means to perform password authentication. You should ensure that your LDAP directory is configured to use Secure Socket Layer (SSL) and that LiveCycle ES2 is configured to access your LDAP directory using its SSL port.

**LDAP denial of service**

A common attack using LDAP involves an attacker deliberately failing to authenticate multiple times. This forces the LDAP Directory Server to lock out a user from all LDAP-reliant services.

You can set the number of failure attempts and subsequent lock-out time that LiveCycle ES2 implements when a user repeatedly fails to authenticate to LiveCycle ES2. In the LiveCycle Administration Console, choose low values. When selecting the number of failure attempts, it is important to understand that after all attempts are made, LiveCycle ES2 locks out the user before the LDAP Directory Server does.

➢ **To set automatic account locking settings:**

1. Log in to LiveCycle Administration Console.

2. Click **Settings > User Management > Domain Management**.

3. Under Automatic Account Locking Settings, set **Maximum Consecutive Authentication Failures** to a low number, such as 3.

4. Click **Save**.
Auditing and logging

The proper and secure use of application auditing and logging can help ensure that security and other anomalous events are tracked and detected as quickly as possible. Effective use of auditing and logging within an application includes such items as tracking successful and failed logins, as well as key application events such as the creation or deletion of key records.

You can use auditing to detect many types of attacks, including these:

- Brute force password attacks
- Denial of service attacks
- Injection of hostile input and related classes of scripting attacks

This table describes auditing and logging techniques you can use to reduce your server’s vulnerabilities.

<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log file ACLs</td>
<td>Set appropriate LiveCycle ES2 log file access control lists (ACLs). Setting the appropriate credentials helps prevent attackers from deleting the files. The security permissions on the log file directory should be Full Control for Administrators and SYSTEM groups. The LiveCycle ES2 user account should have Read and Write permissions only.</td>
</tr>
<tr>
<td>Log file redundancy</td>
<td>If resources permit, send logs to another server in real time that is not accessible by the attacker (write only) by using Syslog, Tivoli, Microsoft Operations Manager (MOM) Server, or another mechanism. Protecting logs this way helps prevent tampering. Also, storing logs in a central repository aids in correlation and monitoring (for example, if multiple LiveCycle ES2 servers are in use and a password-guessing attack is taking place across multiple computers where each computer is queried for a password).</td>
</tr>
</tbody>
</table>

LiveCycle ES2 Unix system library dependencies

The following information is intended to help you plan for a LiveCycle ES2 deployment on a UNIX environment.

Convert PDF service

The Convert PDF service that is part of LiveCycle ES2 requires the following minimum system libraries:

**Linux**

```
/lib/
   libdl.so.2 (0x00964000)
   ld-linux.so.2 (0x007f6000)
/lib/tds/
   libc.so.6 (0x00813000)
   libm.so.6 (0x0093f000)
   libpthread.so.0 (0x000a5d000)
/usr/lib/libz.so.1 (0x0096a000)
/gcc410/lib/
   libgcc_s.so.1 (0x00fc0000)
```
libstdc++.so.6 (0x001111000)

**Solaris**

/usr/platform/SUNW,Sun-Fire-V210/lib/libc_psr.so.1  
/usr/lib/  
lib.so.1  
libdl.so.1  
libintl.so.1  
libm.so.1  
libmp.so.2  
libnsl.so.1  
libpthread.so.1  
libsocket.so.1  
libstdc++.so.6  
libthread.so.1

**AIX**

/usr/lib/  
libpthread.a(shr_comm.o)  
libpthread.a(shr_xpg5.o)  
libc.a(shr.o)  
librt1.a(shr.o)  
libpthread.a(shr_comm.o)  
libcrypt.a(shr.o)  
/aix5.2/lib/gcc/powerpc-ibm-aix5.2.0.0/4.1.0/libstdc++.a(libstdc++.so.6)  
/aix5.2/lib/gcc/powerpc-ibm-aix5.2.0.0/4.1.0/libgcc_s.a(shr.o)

**XMLForms**

XMLForms requires the following minimum system libraries:

**Linux**

/lib/  
libdl.so.2  
libpthread.so.0  
libm.so.6  
libgcc_s.so.1  
libc.so.6  
librt.so.1  
ld-linux.so.2  
/usr/X11R6/lib/  
libX11.so.6

**Solaris**

/usr/lib/  
libdl.so.1  
libpthread.so.1  
libintl.so.1  
libsocket.so.1  
libnsl.so.1  
libm.so.1  
libc.so.1  
librt.so.1
libX11.so.4
libmp.so.2
libmd5.so.1
libscf.so.1
libaio.so.1
libXext.so.0
libdoor.so.1
libuutil.so.1
libm.so.2
/usr/platform/SUNW,Sun-Fire-V210/lib/libpsr.so.1
/usr/platform/SUNW,Sun-Fire-V210/lib/libmd5_psr.so.1

**AIX 6.1**

/usr/lib/
  libpthread.a(shr_comm.o)
  libpthread.a(shr_xpg5.o)
  libc.a(shr.o)
  librt1.a(shr.o)
  libdl.a(shr.o)
  libX11.a(shr4.o)
  libiconv.a(shr4.o)
  libpthreads.a(shr_comm.o)

/unix
  /usr/lib/libcrypt.a(shr.o)
  /usr/lib/libIM.a(shr.o)
  /usr/lib/libpthreads.a(shr_xpg5.o)
Configuring LiveCycle ES2 for access beyond the enterprise

After you successfully install LiveCycle ES2, it is important to periodically maintain the security of your environment. This section describes the tasks that are recommended to maintain the security of your LiveCycle ES2 production server.

Setting up a reverse proxy for web access

A reverse proxy can be used to ensure that one set of URLs for LiveCycle ES2 web applications are available to both external and internal users. This configuration is more secure than allowing users to connect directly to the application server that LiveCycle ES2 is running on. The reverse proxy performs all HTTP requests for the application server that is running LiveCycle ES2. Users have only network access to the reverse proxy and can only attempt URL connections that are supported by the reverse proxy.

LiveCycle ES2 root URLs for use with reverse proxy server

The following application root URLs for each LiveCycle ES2 web application. You should configure your reverse proxy only to expose URLs for web application functionality that you want to provide to end users.

Certain URLs are highlighted as end-user-facing web applications. You should avoid exposing other URLs for LiveCycle Configuration Manager for access to external users through the reverse proxy.

<table>
<thead>
<tr>
<th>Root URL</th>
<th>Purpose and/or associated web application</th>
<th>Web-based interface</th>
<th>End-user access</th>
</tr>
</thead>
<tbody>
<tr>
<td>/ReaderExtensions/*</td>
<td>LiveCycle Reader Extensions ES2 end-user web application for applying usage rights to PDF documents</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>/edc/*</td>
<td>LiveCycle Rights Management ES2 end-user web application</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Root URL</td>
<td>Purpose and/or associated web application</td>
<td>Web-based interface</td>
<td>End-user access</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------------------------------------------------------------------------------------------------------</td>
<td>---------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>/edcws/*</td>
<td>Web service URL for Rights Management ES2</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>/pdfgui/*</td>
<td>LiveCycle PDF Generator ES2 administration web application</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>/workspace/*</td>
<td>LiveCycle Workspace ES2 end-user web application</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>/workspace-server/*</td>
<td>LiveCycle Workspace ES2 servlets and data services that the Workspace ES2 client application requires</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>/contentspace/*</td>
<td>LiveCycle Contentspace ES2 end-user web application</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>/adobe-bootstrapper/*</td>
<td>Servlet for bootstrapping the LiveCycle ES2 repository</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>/adobe-lcm-bootstrapper/*</td>
<td>Redirect to bootstrap servlet/redirects LiveCycle 7.x style bootstrap requests to /adobebootstrapper/</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>/soap/*</td>
<td>Information page for LiveCycle ES2 web services</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>/soap/services/*</td>
<td>Web service URL for all LiveCycle ES2 services</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>/edc/admin/*</td>
<td>LiveCycle Rights Management ES2 administration web application</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>/adminui/*</td>
<td>LiveCycle Administration Console home page</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>/TruststoreComponent/</td>
<td>Trust Store Management administration pages</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>secured/*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/FormsIVS/*</td>
<td>Forms ES2 IVS application for testing and debugging form rendering</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>/OutputIVS/*</td>
<td>Output ES2 IVS application for testing and debugging output service</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>/rmws/*</td>
<td>REST URL for Rights Management</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>/OutputAdmin/*</td>
<td>LiveCycle Output ES2 administration pages</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>/FormServer/*</td>
<td>LiveCycle Forms ES2 web application files</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>/FormServer/GetImage</td>
<td>Used for fetching JavaScript during HTML transformation</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Servlet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/FormServerAdmin/*</td>
<td>LiveCycle Forms ES2 administration pages</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>/repository/*</td>
<td>URL for WebDAV (debugging) access</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>/appstore/Forms/*</td>
<td>Compatibility: Redirect to repository WebDAV implementation for clients of LiveCycle Form Manager 7.x WebDAV</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
Secure network configuration

This section describes the protocols and ports that are required by LiveCycle ES2 and provides recommendations for deploying LiveCycle ES2 in a secure network configuration.

LiveCycle ES2 physical architecture

This image shows the components and protocols that are used in a typical LiveCycle ES2 deployment, including the appropriate firewall topology.
Network protocols used by LiveCycle ES2

When you configure a secure network architecture as described in the previous section, the following network protocols are required for interaction between LiveCycle ES2 and other systems in your enterprise network.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Use</th>
</tr>
</thead>
</table>
| HTTP     | ● Browser displays LiveCycle Configuration Manager and end-user web applications  
          ● All SOAP connections |
| SOAP     | ● Web service client applications, such as .NET applications  
          ● Adobe Reader® uses SOAP for LiveCycle ES2 web services  
          ● Adobe Flash® applications uses SOAP for LiveCycle ES2 web services  
          ● LiveCycle ES2 SDK calls when used in SOAP mode  
          ● LiveCycle Workbench ES2 design environment |
| RMI      | LiveCycle ES2 SDK calls when used in Enterprise JavaBeans (EJB) mode |
| IIOP     | LiveCycle 7.x applications (PDF Manipulation Module APIs) calling LiveCycle ES2 services through the CORBA Backwards Compatibility Layer. |
| IMAP / POP3 | ● Email-based input to a service (Email endpoint)  
               ● User task notifications over email |
| UNC File IO | LiveCycle ES2 monitoring of watched folders for input to a service (watched folder endpoint) |
| LDAP     | ● Synchronizations of organizational user and group information in a directory  
          ● LDAP authentication for interactive users |
| JDBC     | ● Query and procedure calls made to an external database during execution of a process using the JDBC service  
          ● Internal access LiveCycle ES2 repository |
| WebDAV   | Enables remote browsing of the LiveCycle ES2 design-time repository (forms, fragments, and so on) by any WebDAV client |
| AMF      | Adobe Flash applications, where LiveCycle ES2 services are configured with a Remoting endpoint |
| JMX      | LiveCycle ES2 exposes MBeans for monitoring using JMX |

Ports for application servers

This section describes the default ports (and alternate configuration ranges) for each type of application server supported. These ports must be enabled or disabled on the inner firewall, depending on the network functionality you want to allow for clients that connect to the application server running LiveCycle ES2.

**Note:** By default, the server exposes several JMX MBeans under the adobe.com namespace. Only information that is useful for server health monitoring is exposed. However, to prevent information
disclosure, you should prevent callers in an untrusted network from looking up JMX MBeans and accessing health metrics.

**JBoss ports**

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access to web applications</td>
<td>[JBoss root]/server/all/deploy/jbossweb-tomcat50.sar/server.xml</td>
</tr>
<tr>
<td></td>
<td>HTTP/1.1 Connector port 8080</td>
</tr>
<tr>
<td></td>
<td>AJP 1.3 Connector port 8009</td>
</tr>
<tr>
<td></td>
<td>SSL/TLS Connector port 8443</td>
</tr>
<tr>
<td>Access to LiveCycle ES2 services</td>
<td>[JBoss root]/server/all/conf/jboss-service.xml</td>
</tr>
<tr>
<td></td>
<td>WebService port 8083</td>
</tr>
<tr>
<td></td>
<td>NamingService Port 1099</td>
</tr>
<tr>
<td></td>
<td>RMI port from 1098</td>
</tr>
<tr>
<td></td>
<td>RMIOBJECTPORT from 4444</td>
</tr>
<tr>
<td></td>
<td>PooledInvoker ServerBindPort 4445</td>
</tr>
<tr>
<td>J2EE cluster support</td>
<td>[JBoss root]/server/all/deploy/cluster-service.xml</td>
</tr>
<tr>
<td></td>
<td>ha.jndi.HANamingService port from 1100</td>
</tr>
<tr>
<td></td>
<td>RmiPort 1101</td>
</tr>
<tr>
<td></td>
<td>RMIOBJECTPORT 4447</td>
</tr>
<tr>
<td></td>
<td>(clusters only) ServerBindPort 4446</td>
</tr>
<tr>
<td>CORBA support</td>
<td>[JBoss root]/server/all/conf/jacorb.properties</td>
</tr>
<tr>
<td></td>
<td>OAPort 3528</td>
</tr>
<tr>
<td></td>
<td>OASSLPort 3529</td>
</tr>
<tr>
<td>SNMP support</td>
<td>[JBoss root]/server/all/deploy/snmp-adaptor.sar/META-INF/jboss.</td>
</tr>
<tr>
<td></td>
<td>service. xml</td>
</tr>
<tr>
<td></td>
<td>ports 1161, 1162</td>
</tr>
<tr>
<td></td>
<td>[JBoss root]/server/all/deploy/snmp-adaptor.sar/managers.xml</td>
</tr>
<tr>
<td></td>
<td>port 1162</td>
</tr>
</tbody>
</table>

**WebLogic ports**

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access to web applications</td>
<td>● Admin Server listen port: default is 7001</td>
</tr>
<tr>
<td></td>
<td>● Admin Server SSL listen port: default is 7002</td>
</tr>
<tr>
<td></td>
<td>● Port configured for Managed Server, for example 8001</td>
</tr>
<tr>
<td>WebLogic administration ports not</td>
<td>● Managed Server listen port: Configurable from 1 to 65534</td>
</tr>
<tr>
<td>required for access to LiveCycle ES2</td>
<td>● Managed Server SSL listen port: Configurable from 1 to 65534</td>
</tr>
<tr>
<td></td>
<td>● Node Manager listen port: default is 5556</td>
</tr>
</tbody>
</table>
WebSphere 6.1 ports

For information about WebSphere 6.1 ports that LiveCycle ES2 requires, go to Port number settings in WebSphere Application Server versions.

WebSphere 7.0 ports


Configuring SSL

Referring to the physical architecture that is described in the section “LiveCycle ES2 physical architecture” on page 29, you should configure SSL for all of the connections that you plan to use. Specifically, all SOAP connections must be conducted over SSL to prevent exposure of user credentials on a network.

For instructions on how to configure SSL on JBoss, WebLogic, and WebSphere, see “Configuring SSL” in the LiveCycle ES2 Administration Help.

Configuring SSL redirect

After you configure your application server to support SSL, you must ensure that all HTTP traffic to LiveCycle ES2 applications and services are enforced to use the SSL port.

To configure SSL redirect for WebSphere or WebLogic, see your application server documentation.

➤ To configure SSL redirect for JBoss:

1. Navigate to the adobe-livecycle-jboss.ear and unzip it.
2. Extract the adminui.war file and open the web.xml file for editing.
3. Add the following code to the web.xml file:

```xml
<security-constraint>
  <web-resource-collection>
    <web-resource-name>app or resource name</web-resource-name>
    <url-pattern>*</url-pattern>
    <!-- define all url patterns that need to be protected-->
    <http-method>GET</http-method>
    <http-method>POST</http-method>
  </web-resource-collection>
  <user-data-constraint>
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>
  </user-data-constraint>
</security-constraint>
```

Windows-specific security recommendations

This section contains security recommendations that are specific to Windows when used to run LiveCycle ES2.
JBoss Service accounts

The LiveCycle ES2 turnkey installation sets up a service account, by default, using the Local System account. The built-in Local System user account has a high level of accessibility; it is part of the Administrators group. If a worker process identity runs as the Local System user account, that worker process has full access to the entire system.

➤ To run the LiveCycle ES2 application server using a specific non-administrative account:

1. In the Microsoft Management Console (MMC), create a local user for the LiveCycle ES2 service to log in as:
   - Select User cannot change password.
   - On the Member Of tab, ensure that the Users group is listed.
2. Select Settings > Administrative Tools > Services.
3. Double-click the the LiveCycle ES2 Application Server service and stop the service.
4. On the Log On tab, select This Account, browse for the user account you created, and enter the password for the account.
5. In the Local Security Settings window, under User Rights Assignment, give the following rights to the user account that LiveCycle ES2 server is running under:
   - Deny log on through Terminal Services
   - Deny log on locally
   - Log on as Service (should be already set)
6. Give the new user account Read & Execute, List Folder Contents, and Read permissions to LiveCycle ES2web content directories.
7. Start the the LiveCycle ES2 Application Server service.

File system security

LiveCycle ES2 uses the file system in the following ways:
- Stores temporary files that are used while processing document input and output
- Stores files in the global archive store that are used to support the solution components that are installed
- Watched folders store dropped files that are used as input to a service from a file system folder location

When using watched folders as a way to send and receive documents with a LiveCycle ES2 service, take extra precautions with file system security. When a user drops content in the watched folder, that content is exposed through the watched folder. In this case, the service does not authenticate the actual end user. Instead, it relies on ACL and Share level security to be set at the folder level to determine who can effectively invoke the service.

JBoss-specific security recommendations

This section contains application server configuration recommendations that are specific to JBoss 4.2.x when used to run LiveCycle ES2.
Disable JBoss Management Console and JMX Console

Access to the JBoss Management Console and JMX Console is already configured (JMX monitoring is disabled) when you install LiveCycle ES2 on JBoss by using the turnkey installation method. If you are using your own JBoss Application Server, ensure that access to the JBoss Management Console and JMX monitoring console are secured. Access to the JMX monitoring console is set in the JBoss configuration file called jmx-invoker-service.xml.

Disable directory browsing

After logging into LiveCycle Administration Console, it is possible to browse the console’s directory listing by modifying the URL. For example, if you change the URL to one of the following URLs, a directory listing may appear:

http://<servername>:8080/adminui/secured/
http://<servername>:8080/um/

To disable the directory listing, set the value of the listings initialization parameter of the DefaultServlet property to false as shown in bold in the [JBoss_ES2 root]/server/default/deploy/jbossweb-tomcatxxx.sar/conf/web.xml file, as shown in this example:

```xml
<servlet>
    <servlet-name>default</servlet-name>
    <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>
    <init-param>
        <param-name>listings</param-name>
        <param-value>false</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>
```

WebLogic-specific security recommendations

This section contains application server configuration recommendations for securing WebLogic 9.1 when running LiveCycle ES2.

Disable directory browsing

Set the index-directories properties in the weblogic.xml file to false, as shown by this example:

```xml
<container-descriptor>
    <index-directory-enabled>false</index-directory-enabled>
</container-descriptor>
```

Enable WebLogic SSL Port

By default, WebLogic does not enable the default SSL Listen Port, 7002. Enable this port in the WebLogic Server Administration Console before you configure SSL.
WebSphere-specific security recommendations

This section contains application server configuration recommendations for securing WebSphere running LiveCycle ES2.

Disable directory browsing

Set the `directoryBrowsingEnabled` property in the `ibm-web-ext.xml` file to `false`.

Enabling WebSphere administrative security

➤ To enable WebSphere administrative security:

1. Log in to the WebSphere Administrative Console.
2. In the navigation tree, go to one of the following links:
   - (WebSphere 6.1) **Security > Secure administration, applications, and infrastructure**
   - (WebSphere 7.0) **Security > Global Security**
3. Select **Enable administrative security**.
4. Deselect both **Enable application security** and **Use Java 2 security**.
5. Click **OK** or **Apply**.
6. In the **Messages** box, click **Save directly to the master configuration**.
Generally, developers do not use the LiveCycle ES2 production environment to build and test their applications. Therefore, you must administer user accounts and services that, although required in a private development environment, are not required in a production environment.

This section describes methods for reducing the overall attack surface through administration options that LiveCycle ES2 provides.

Disabling non-essential remote access to services

After LiveCycle ES2 is installed and configured, many LiveCycle ES2 services are available for remote invocation over SOAP, Enterprise JavaBeans™ (EJB), and LiveCycle Remoting. The term remote, in this case, refers to any caller that has network access to the SOAP, EJB, or Action Message Format (AMF) ports for the application server.

Although LiveCycle ES2 services require valid credentials to be passed for an authorized caller, you should allow only remote access to the services that you need to be remotely accessible. To achieve limited accessibility, you should reduce the set of remotely accessible services to the minimum possible for a functioning system and then enable remote invocation for the additional services that you need.

LiveCycle ES2 services always need at least SOAP access. These services are typically required for use by LiveCycle Workbench ES2 but also include services that are called by the LiveCycle Workspace ES2 web application.

Complete this procedure using the Applications and Services web page in LiveCycle Administration Console:

➤ To disable remote access to services:

1. Log in to LiveCycle Administration Console by typing the following URL in a web browser:
   
   http://[host name]:[port]/adminui

2. Click Services > Applications and Services > Preferences.

3. Set the Preferences to view up to 200 services and endpoints on the same page.

4. Click Services > Applications and Services > Endpoint Management.

5. Select EJB from the Provider list and then click Filter.

6. To disable all EJB endpoints, select the check box beside each one in the list and click Disable.

7. Click Next and repeat the previous step for all EJB endpoints. Ensure that EJB is listed in the Provider column before you disable endpoints.

8. Select SOAP from the Provider list and then click Filter.

9. To remove SOAP endpoints, select the check box beside each one in the list and click Remove. Do not remove the following endpoints:
Disabling non-essential anonymous access to services

Some LiveCycle ES2 services permit unauthenticated (anonymous) invocation for some operations. This means that one or more operations exposed by the service may be invoked as any authenticated user or as no authenticated user at all.

➤ To disable anonymous access to services:

1. Log in to LiveCycle Administration Console by typing the following URL in a web browser:
   http://[host name]:[port]/adminui

2. Click Services > Applications and Services > Service Management.

3. Click the name of the service that you want to disable (for example, AuthenticationManagerService).

4. Click the Security tab, deselect Anonymous Access Allowed, and click Save.

5. Complete steps 3 and 4 for the following services:
   - AuthenticationManagerService
   - EJB
   - Email
   - JobManager
   - WatchedFolder
   - UsermanagerUtilService
   - Remoting
   - RemoteEvents
   - RepositoryProviderService

10. Click Next and repeat the previous step for SOAP endpoints that are not in the above list. Ensure that SOAP is listed in the Provider column before you remove endpoints.
EMCDocumentumRepositoryProvider
IBMFilenetRepositoryProvider
FormAugmenter
TaskManagerService
TaskManagerConnector
TaskManagerQueryService
TaskQueueManager
TaskEndpointManager
LCMTMInvoker
UserService
WorkspaceSearchTemplateService
WorkspaceSignleSignOn
WorkspacePropertyService
OutputService
FormsService

If you intend to expose any of these services for remote invocation, you should also consider disabling anonymous access for these services. Otherwise, any caller with network access to this service may invoke the service without passing valid credentials.

Anonymous access should be disabled for any services that are not needed. Many internal services require anonymous authentication to be enabled because they need to be invoked by potentially any user in the system without being preauthorized.

Remove sample user and role assignments

You may have included sample users and roles when you installed LiveCycle ES2 (for example, Kel Varsen and the Finance Corp User Domain. Using the User Management administration pages, you should remove the sample user domain and sample roles.

➤ To remove sample users:

1. Log in to LiveCycle Administration Console by typing the following URL in a web browser:
   http://[host name]:[port]/adminui
2. Click Settings > User Management > Users and Groups.
3. Select the Sample Organization from the and domain list and click Find.
4. To disable all sample users, select the check box beside each one in the list and click Delete.

➤ To remove sample domains:

1. Log in to LiveCycle Administration Console by typing the following URL in a web browser:
   http://[host name]:[port]/adminui
2. Click Settings > User Management > Domain Management.
3. To delete all sample domains, select the check box beside each one in the list and click Delete.

4. Click Save.

Changing the default global time-out

End users can authenticate to LiveCycle ES2 through LiveCycle Workbench ES2, LiveCycle ES2 web applications, or custom applications that invoke LiveCycle ES2 services. One global time-out setting is used to specify how long such users can interact with LiveCycle ES2 (using a SAML-based Assertion) before they are forced to reauthenticate. The default setting is two hours. On a production environment, the amount of time needs to be reduced to the minimum number of minutes acceptable.

➤ To minimize reauthentication time limit:

1. Log in to LiveCycle Administration Console by typing the following URL in a web browser:
   http://[host name]:[port]/adminui

2. In LiveCycle Administration Console, click Settings > User Management > Configuration > Import And Export Configuration Files.

3. Click Export to produce a config.xml file with the existing LiveCycle ES2 settings.

4. Open the XML file in an editor and locate the following entry:
   <entry key="assertionValidityInMinutes" value="120"/>

5. Change the value to any number greater than 5 (in minutes) and save the file.

6. In LiveCycle Administration Console, navigate to the Import And Export Configuration Files page.

7. Enter the path to the modified config.xml file or click Browse to navigate to it.

8. Click Import to upload the modified config.xml file and then click OK.

Disabling LiveCycle 7.x backwards-compatibility API access

Applications that are developed by using the LiveCycle 7.x SDK do not invoke LiveCycle ES2 services by using an authenticated EJB or SOAP request. Instead, they make an unsecured CORBA invocation to a CORBA service that is deployed on the application server.

If you choose the upgrade option in LiveCycle Configuration Manager during the installation and configuration process, a CORBA service is deployed that allows existing LiveCycle 7.x applications that use the LiveCycle 7.x SDK to run when they are deployed on your application server. If you did not choose Upgrade, this CORBA service is not installed.

➤ To disable the LiveCycle 7.x backwards-compatibility CORBA service:

1. Locate the adobe-livecycle-[appserver].ear file in the [LiveCycleES root]/jboss/deploy directory and make a back-up copy of this EAR file.

2. Within the adobe-core-[appserver].ear file, locate the adobe-core-compat-7to8-[appserver].ear file. This EAR file is present if you already performed a LiveCycle ES2 configuration and deployment with the Upgrade option.
3. Within the adobe-core-compa7to8-{appserver}. ear file, locate the application.xml file.

4. Modify the application.xml file to comment out the following module:

   <!-- adobe-PDFManipulation start -->
   <module id = "WebApp_PDFManipulation">
     <web>
       <web-uri>adobe-PDFManipulation.war</web-uri>
       <context-root>/adobe-PDFManipulation</context-root>
     </web>
   </module>