Troubleshooting LiveCycle® ES2

Adobe® LiveCycle® ES2

February 10, 2011

Version 9

Troubleshooting Adobe® LiveCycle® ES2
February 10, 2011

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

Contents

About This Document

This document contains information about how to troubleshoot many installation, configuration and administration issues that may arise with an Adobe® LiveCycle® ES2 (Enterprise Suite) version 9.0 production environment.

Who should read this document?

This guide provides information for administrators or developers who are responsible for installing, configuring, administering, or deploying LiveCycle ES2 components. The information provided is based on the assumption that anyone reading this guide is familiar with the following:

- J2EE application servers
- Microsoft Windows, AIX, Linux, or Solaris operating systems
- MySQL, Oracle®, DB2®, or SQL Server database servers
- Web environments

Conventions used in this document

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

Name	Description	Default value
[LiveCycleES2 root]	The installation directory that is used for all LiveCycle ES2 modules. The installation directory contains subdirectories for Adobe LiveCycle Configuration Manager, the LiveCycle ES2 SDK, and each LiveCycle ES2 module installed (along with the product documentation). This directory also includes directories relating to third-party technologies.	Windows: C:\Adobe\Adobe LiveCycle ES2\ Linux and UNIX: /opt/adobe/adobe livecycle es2/
[LiveCycle8x root]	The installation directory that is used for all LiveCycle ES (8.x) version 8.0 or version 8.2 solution components.	C:\Adobe\LiveCycle ES\ (for version 8.0) or C:\Adobe\LiveCycle8.2 (for version 8.2)
[JBossES2 root]	*(JBoss Turnkey)* The home directory of the application server that runs LiveCycle ES2.	C:\Adobe\Adobe LiveCycle ES2\jboss
[Adobe_JAVA_HOME]	*(JBoss Turnkey)* The home directory of the Java JDK installed by the LiveCycle ES2 turnkey.	C:\Adobe\Adobe LiveCycle ES2\Java

Name	Description	Default value
[appserver root]	The home directory of the application server that runs the LiveCycle ES2 services.	JBoss on Windows: C:\jboss JBoss on Linux, Solaris: /opt/jboss WebSphere on Windows: C:\Program Files\IBM\WebSphere\AppServer WebSphere on Linux and Solaris: /opt/IBM/WebSphere/AppServer WebSphere on AIX: /usr/IBM/WebSphere/AppServer, or, /opt/IBM/WebSphere/AppServer WebLogic on Windows: C:\bea\wlserver_10.3 WebLogic on Linux and UNIX: /bea/wlserver_10.3
WL_HOME	The install directory for WebLogic as specified for the `WL_HOME` environment variable.	WebLogic on Windows C:\bea WebLogic on Linux and UNIX: /opt/bea
[appserverdomain]	The domain that you configured on WebLogic. The default domain is called *base_domain*.	WebLogic on Windows: C:\bea\user_projects\domains\base_domain WebLogic on Linux and UNIX: /opt/bea/user_projects/domains/base_domain
[dbserver root]	The location where the LiveCycle ES2 database server is installed.	Depends on the database type and your specification during installation.

Additional information

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

For information about	See
LiveCycle ES2, the solution components and development tools	*LiveCycle ES2 Overview*
Preparing your environment for installing or upgrading to LiveCycle ES2	*Preparing to Install LiveCycle ES2 (Single Server)* *Preparing to Install LiveCycle ES2 (Server Cluster)* *Preparing to Upgrade to LiveCycle ES2*
Installing LiveCycle ES2 (Single Server)	*Installing and Deploying LiveCycle ES2 Using Turnkey* *Installing and Deploying LiveCycle ES2 for JBoss* *Installing and Deploying LiveCycle ES2 WebSphere* *Installing and Deploying LiveCycle ES2 for WebLogic*
Configuring LiveCycle ES2 (Server Cluster)	*Configuring LiveCycle ES2 Application Server Clusters Using JBoss* *Configuring LiveCycle ES2 Application Server Clusters Using WebLogic* *Configuring LiveCycle ES2 Application Server Clusters Using WebSphere*
Upgrading to LiveCycle ES2 using the non-turnkey method	*Upgrading to LiveCycle ES2 from 8.x (for JBoss Turnkey)* *Upgrading to LiveCycle ES2 for JBoss* *Upgrading to LiveCycle ES2 for WebLogic* *Upgrading to LiveCycle ES2 for WebSphere*
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*

Getting help

This section describes the steps you should take before you contact Adobe Enterprise Support. If, after reviewing the LiveCycle ES2 documentation, you have not resolved your issues, contact Adobe Enterprise Support. To help expedite your service, have the following information available:

- What were you doing when the problem occurred?
- Can you repeat the problem?
- Did you see any error message when the problem occurred? Did you see anything else?
- If you deselect the Show Friendly HTTP Error Messages check box in Internet Explorer (Tools > Internet Options > Advanced), do the errors persist?

Troubleshooting Installation and Deployment

This section discusses issues you may encounter when installing and deploying LiveCycle ES2, and suggests steps for avoiding or working around them.

Installation considerations

If you have problems installing, configuring, or deploying LiveCycle ES2, make sure that you carefully followed the instructions in these LiveCycle ES2 documents listed in "Additional information" on page 9.

If you installed and configured everything according to the documentation, review the following sections for issues similar to those you are experiencing.

See also

"Installation, Configuration and Deployment Issues" on page 20

"Troubleshooting Error Messages" on page 50

System requirements considerations

Temporary directory space issues

LiveCycle installer fails to install on a system which has no space available on system temporary directory. In such cases, you get the error message in the console window that */tmp* does not have enough disk space.

To avoid this issue, ensure that your temporary directory has enough space. See *Preparing to Install LiveCycle ES2 (Single Server)* guide for information about system requirements.

Slow performance of SharePoint Connector APIs on Windows Server 2008

If your LiveCycle ES2 is deployed on Windows Server 2008, and the SharePoint Server it is communicating with is hosted on a 64-bit Windows machine, SharePoint Content Repository Connector API calls made from Workbench ES2 or from LiveCycle ES2 SDK show slow performance.

To avoid this issue, disable automatic tuning of TCP Receive Window on the SharePoint server or set the finetuning level to `highlyrestricted`.

Use one of the following commands:

```
netsh interface tcp set global autotuninglevel=highlyrestricted
```

```
netsh interface tcp set global autotuninglevel=disabled
```

See *Microsoft knowledge base* for more information.

File size limitations for LiveCycle ES2 Output service

To generate large files on UNIX systems using the LiveCycle ES2 Output Service, you must set the upper limit for file size to greater than or equal to the file being generated.

To ensure errors when generating PDF documents from large files, change the value for `fsize` in /etc/security/limits on UNIX systems to a value large enough to cover all file sizes.

Application server considerations

Verify the following application server settings before you contact Adobe Enterprise Support:

- **Total transaction lifetime timeout:** 300
- **Initial heap size:** 256
- **Maximum heap size**: 1024 MB
- **Prepared statement cache**: 100
- **Database connection pool maximum**:
  - IDP_DS is 30
  - RM_DS is 20
  - BAM metadata database is 20
- **Database connection pool minimum**:
  - IDP_DS is 1
  - RM_DS is 1
- **Connection pool maximum connections**: 50

See also

"Troubleshooting Error Messages" on page 50

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:

```
<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-bin
dings.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.

Configuring Solaris 10 memory requirements

Make the following memory configurations to avoid StuckThread issues on a Solaris environment:

- Add or increase the rlim values in the /etc/system file.
- Increase the swap space to at least twice the total RAM.

➤ **Modify the `rlim` values:**

1. Locate and open the /etc/system file.

2. Locate and modify the `rlim` values as follows:

   `set rlim_fd_cur`: The initial (soft) maximum number of file descriptors per process. Set this value to `8192` or more.

   `set rlim_fd_max`: The hard maximum number of file descriptors per process. Set this value to `8192` or more. (This modification is required only if the default value is lower than 8192). You must have super user privileges to change this value.

   Note: The `rlim_fd_max` value must be equal to or greater than the `rlim_fd_cur` value.

3. Save and close the file.

4. Restart your computer.

➤ **Add additional swap space:**

1. Increase the swap space so that the total swap space limit exceeds twice the total RAM amount. For example, if you have 8 GB of RAM, configure the swap space to exceed 16 GB. (See the tech tip at the Sun community portal.)

2. Restart the computer if required.

➤ **Verify the updated settings:**

1. Launch a new shell.

2. Type `ulimit -n` and press **Enter**.

3. Verify the value returned matches the `rlim` values you have set.

4. Type `swap -s` and press **Enter**.

5. Verify the value returned matches the new swap space value.

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

Database initialization considerations

If you are having problems initializing the LiveCycle ES2 server, consider the following possibilities:

- Database instances must contain only alphanumeric characters in their names.
- (Linux and UNIX) Database instances must not exceed the platform-specific threshold of 8 characters.

If the initialization fails at the beginning of the process, check the following:

- (Non-turnkey installation) The LiveCycle ES2 database is already created and the user has full rights to it.
- The database server is accessible when you ping it.
- The database is empty; that is, it has no tables, sequences, views, or index tables.
- The JNDI name for `IDP_DS` is created.

If initialization fails while writing to the registry, check the application server logs for errors that pertain to the queues and topics. If errors exist, verify that the queues and topics are configured properly. (See *Installing and Deploying LiveCycle ES2* for your application server.)

See also

"Database Error Messages" on page 60

Addressing a high number of concurrent calls

A high number of concurrent calls can negatively impact database performance. To avoid poor performance, increase the maximum connection pool size for the datasource (IDP_DS) to be more than the expected (or actual) number of concurrent calls.

Trust store initialization fails

(SQL Server only) When you create an SQL Server user name (for example, admin1) and you create a schema for the database (for example, admin2), and configure it to be the default schema for the SQL Server account, trust store initialization fails and a message similar to the one below appears:

```
[7/5/07 10:46:35:251 IST] 00000027 TrustStoreBoo E POF is not installed,
Trust Store cannot bootstrap unless POF is installed.
```

Ensure that the SQL Server user name (for example, admin1) and database schema name (for example, admin1) match, in order for the trust store to initialize successfully.

Security Considerations

XMLForm.exe crashes when SELinux security is in "enforcing" mode

You may encounter problems running LiveCycle ES2 on a server where NSA Security Enhanced Linux (SELinux) is in *enforcing* mode. With that configuration, a LiveCycle ES2 process that calls XMLForm.exe will not run. Users will not be able to open the associated form in LiveCycle Workspace ES2, and an error message similar to this one appears in your application server log file:

```
Service XMLFormService: Process ProcessResource(name=XMLForm.exe,pid=0)
terminated abnormally with error code {3}
```

To correct this problem, change SELinux security to *permissive* mode.

Upgrade Considerations

Problem when restoring LiveCycle ES2 to a machine with a different host name

Imported LiveCycle 8.x processes may use a Document Form variable and the Submit PDF Form service. For such processes, users are not able to submit a form to complete a task under the following circumstances:

1. The process is started and then a backup of the database and GDS is performed.

2. LiveCycle ES2 is restored on another machine with a different host name. Under the GDS directory, the "backup" folder is renamed to "restore", and the database is restored.

3. The user submits the form in Workspace.

When the user submits the form, a Security Warning dialog box displays this message: "This document has accessed more than one site and might be trying to share data among different locations. This could be a privacy concern. Are you sure you want to continue?" When the user clicks Yes, this error message appears: "This page cannot be displayed".

To avoid this problem, use the same host name when restoring LiveCycle ES2 from a backup.

Content Services ES2 EAR fails to deploy during upgrade

While upgrading from LiveCycle ES Update 1 (8.2.1.x) to LiveCycle ES2, deployment of the Content Services ES2 EAR may fail with the following exception message:

```
SchemaBootstr E org.alfresco.util.LogUtil error Schema auto-update failed
org.alfresco.error.AlfrescoRuntimeException: A previous schema upgrade failed
or was not completed. Revert to the original database before attempting the
upgrade again.
```

This issue occurs due to one of the following reasons:

- WebSphere application resource utilization is high
- The ALF_BOOTSTRAP_LOCK table is present in the LiveCycle ES Update 1 database

EAR deployment failure due to high WebSphere resource utilization

Content Services ES2 deployment may fail during upgrade if the WebSphere application server resource utilization is high.

To resolve this issue, perform the following procedures.

Restoring the contents of lccs_data and the database from backup

1. Revert to the original Content Services ES database by restoring the tables beginning with `alf` and `avm` from the backup copy of the database.

2. Restore the contents of the Content Store root location from the backup copy.

Increasing the total transaction lifetime timeout value

1. (WebSphere 6.x) In the WebSphere Administrative Console, click **Servers > Application servers** and then click the server name.

   (WebSphere 7.x) In the WebSphere Administrative Console, click **Servers > Server Types > WebSphere application servers** and then click the server name.

2. Click **Container Settings > Container Services > Transaction Service.**

3. On the Configuration tab, set the value of the **Total transaction lifetime timeout** setting to 900 seconds.

4. Click **Apply** or **OK**.

5. Restart the application server.

Deploying the EARs in a different order

Initiate the upgrade again. Now, in LiveCycle Configuration Manager, deploy the Content Services ES2 EAR before deploying the other EARs.

1. On the Deploy LiveCycle ES2 EARs screen, select `adobe-contentservices.ear` and then click **Deploy**.
2. Once the Content Services ES2 EAR has deployed successfully, deselect `adobe-contentservices.ear`, select the other EARs, and click **Deploy**.

EAR deployment failure due to the ALF_BOOTSTRAP_LOCK database table

If the ALF_BOOTSTRAP_LOCK table is present in the LiveCycle ES Update 1 database, Content Services ES2 EAR deployment may fail during upgrade.

To resolve this issue, follow these steps:

1. Restore the contents of lccs_data and the database from backup. See "Restoring the contents of lccs_data and the database from backup" on page 16.
2. Delete the ALF_BOOTSTRAP_LOCK table from the LiveCycle ES Update 1 database and reinitiate the upgrade.
3. Deploy the Content Services ES2 EAR and then deploy the other EARs. See "Deploying the EARs in a different order" on page 17.

Content Services EAR file is not deployed to all nodes during upgrade

When upgrading to Content Services ES2 from LiveCycle ES version 8.2 on a cluster, the Content Services ES2 EAR file is deployed to the first node but not to the other cluster nodes. The following two workarounds resolve this issue, but each has its drawbacks. Review each workaround to determine which is the best solution for your environment.

- During upgrade, while configuring the Content Services ES2 EAR file using LiveCycle Configuration Manager, point the Index Root directory for LiveCycle ES2 to a location different from what was specified for version 8.2. This workaround allows you start all the nodes in the cluster directly from LiveCycle Configuration Manager.

  Note: With this option, the LiveCycle ES2 server can take a long time to start up if you have a lot of content saved in the Content Services ES2 repository. This is because each node of the cluster attempts to recreate the indexes.

- While deploying the EAR files, make sure that only one of the nodes of the cluster is started and specify the details pertaining to only that node during the entire upgrade process. This step ensures that the LiveCycle ES2 server only updates the indexes rather than recreating them.

  Once the node starts successfully, manually copy the indexes directory from that node to the other nodes of the cluster where you do not plan to run LiveCycle Configuration Manager. Now, start the other nodes of the cluster. The Content Services ES2 EAR file will now be successfully deployed to all cluster nodes.

Note: Although this workaround is time-consuming to implement, it ensures minimal server downtime during startup.

Some components are not upgraded correctly

If an upgrade attempt from LiveCycle 8.x to LiveCycle ES2 fails due to any reason, some components may not be upgraded correctly during a subsequent upgrade even after you've reversed your setup to a pre-upgrade state. For example, Reader Extensions ES2.

To resolve this issue, you must delete the `sharedData` file in the <*LiveCycle ES2 root*>/configurationManager/working/upgrade directory before reinitiating upgrade.

This step is in addition to reversing the following entities to the pre-upgrade state before reinitiating upgrade:

- Database
- lccs_data
- GDS
- Other relevant folders

"Bad version number in .class file" error during database verification

(Oracle database only)

When upgrading to a LiveCycle ES2 platform that uses JDK 1.5 and will connect to an Oracle database, you must use the ojdbc5.jar file. If the ojdbc6.jar file (default) is used, you will see the error message "Bad version number in .class file" when verifying the database connection on the LiveCycle ES2 Database screen.

If you see this error, exit LiveCycle Configuration Manager and then restart it. On the LiveCycle ES2 Database screen, provide the ojdbc5.jar file.

Note: On a LiveCycle ES2 platform that uses JDK 1.6, you can input the database driver as either the ojdbc5.jar or ojdbc6.jar file.

Flex applications fail to render after upgrading to LiveCycle ES2

Flex applications in LiveCycle ES or LiveCycle ES update 1 are built using Flex 3.0.1 SDK. However, they are incompatible with Flex 3.4 SDK used by LiveCycle ES2. Therefore, if you are upgrading to LiveCycle ES2, you must recompile these applications with Flex 3.4 SDK.

You can install Flex 3.4 SDK from the LiveCycle ES2 installation media:

<*LiveCycle_DVD*>/additional/flex_sdk

PDF Generator ES2 considerations

The following topics address issues encountered with PDF Generator ES2 and their respective resolutions.

Configuring PDF Generator ES2 to convert MS Office files to PDF format

It is a known issue that on some LiveCycle ES2 servers, a DCOM permission policy must be modified to use PDF Generator ES2 to convert Microsoft Office application files successfully. Otherwise, the following error message appears:

```
INFO [STOUT] com.jniwrapper.win32.com.ComException: COM object method returns
error code: 0x80004005; E_FAIL (Unspecified error)
```

Microsoft has documented the required steps in a knowledge base article on the Microsoft Help and Support site.

Known issue when a LiveCycle ES2 server is accessed over a Telnet session

Native to PDF conversions fail with an error when you access a LiveCycle ES2 server running on Windows Server 2008 using the Telnet service and start the application server using a batch script.

To prevent this issue, start the Telnet service with the **Allow service to interact with desktop** option selected. You can select this option when you modify the properties of the **Services** panel in Windows Server 2008.

Resolving paths when converting Excel files to PDF file

When you use LiveCycle PDF Generator ES2 to convert an Excel file to a PDF file, and the Excel file contains a file name and path function (`&[Path]&[File]`) in the header or footer, unexpected results may occur. When converting the file, PDF Generator ES2 copies it to a temporary location on the LiveCycle ES2 server and performs the conversion on that copy. As a result, the file name and path functions in the Excel document resolve to the temporary file name and location, and those values appear in the generated PDF file.

Installation, Configuration and Deployment Issues

General issues

LiveCycle Configuration Manager does not start in command line interface

This error occurs when you haven't set the ADOBE_JAVA_HOME environment variable. To ensure successful installation, always ensure that the required parameters are set before you launch the installer and LiveCycle Configuration Manager.

install.bin fails to launch on Red Hat Enterprise Linux

When you try to run the install.bin executable from the LiveCycle ES2 DVD on a machine running

Red Hat® Enterprise Linux®, the following error message appears:

```
/bin/sh:bad interpreter: Permission denied
```

This occurs because Red Hat Enterprise Linux auto-mounts the DVD with noexec permissions. To resolve this issue and start the LiveCycle ES2 installation, perform these steps:

1. Unmount the drive by entering the following command:

   ```
   umount /media/CDROM
   ```

2. Remount the drive manually by completing the following tasks:
   - Create a directory named CDROM under the /media folder:

     ```
     mkdir /media/CDROM
     ```
   - Mount the LiveCycle ES2 DVD under the /media/CDROM folder:

     ```
     mount /dev/hda /media/CDROM
     ```
   - Change to the directory on which the DVD is mounted and run `./install.bin`

Configuration issues

LiveCycle Configuration Manager maintains state for incompatible driver jar on WebSphere

If an incompatible .jar file is specified in the **JDBC driver in application server lib** field on the **Datasource Configuration** screen, the location of the correct .jar file can be specified only after restarting LiveCycle Configuration Manager.

`ojdbc5.jar` is the correct .jar file for WebSphere 6.x, while `ojdbc6.jar` is the correct .jar file for WebSphere 7.x.

LiveCycle Configuration Manager picks a different JDK than is expected

If you have JDKs from multiple vendors installed on your system, LiveCycle Configuration Manager may pick a JDK other than the JDK specified in the PATH environment variable.

To resolve this issue, launch LiveCycle Configuration Manager using `ConfigurationManager.bat` instead of `ConfigurationManager.exe`.

Tip: You can examine the LiveCycle Configuration Manager logs to ensure that it is using the correct JDK.

Deployment issues

LiveCycle Configuration Manager fails to deploy LCA files when using express mode

LiveCycle Configuration Manager may fail to deploy LCAs when running in Express mode. If this error occurs, LiveCycle Configuration Manager will not allow you to proceed and the Next button will be disabled. To troubleshoot this issue, run the LiveCycle Configuration Manager again, and select Custom mode. Then, select Configure Components and Import samples. You may encounter this issue when configuring LiveCycle Content Services 9.0 without LiveCycle Process Management 9.0.

Failure to deploy EAR files

Depending on the LiveCycle ES2 services you are installing and your system configuration, you may receive errors when deploying the EAR files. If this occurs, increase the `MaxPermSize` on your application server from `256` to `512`. For specific instructions on setting this value on your application server, see *Preparing to Install LiveCycle ES2*. For example, in WebLogic, you can change the `MaxPermSize` here:

C:\bea\user_projects\domains\base_domain\bin\setDomainEnv.cmd

Null pointer exception on redeploying Content Services ES2 EARs

(WebSphere 7.0.0.7 only) After deploying Content Services ES2 EAR, restart the application server; failing which a Null pointer exception is thrown when Content Services ES2 EAR is redeployed. This is due to a known issue with WebSphere 7.0.0.7.

Incorrect reporting of failure to deploy Content Services ES2 EARs

If the system does not meet minimum system requirements or is heavily loaded, during installation or upgrade to LiveCycle ES2, LiveCycle Configuration Manager incorrectly reports failure in deploying the Content Services ES2 EAR. This can occur while installing LiveCycle ES2 or upgrading LiveCycle from 8.x to LiveCycle ES2.

In addition, LiveCycle Configuration Manager logs contain the error message: “Failed to make SOAP RPC call: invoke”. As a result of this incorrect reporting, LiveCycle Workspace ES2 EAR is not deployed and you are unable to proceed to the next step in the LiveCycle Configuration Manager.

If this error appears, do the following:

1. Verify that the SOAP request time out is correctly set per the documentation. See the section “Configuring WebSphere time-out settings” in the *Installing and Deploying LiveCycle ES2 for WebSphere* or *Upgrading to LiveCycle ES2 from 8.x for WebSphere* guide.

2. From the WebSphere Administrative Console, verify if Content Services ES2 EAR is correctly deployed and running. If you determine that Content Services ES2 EAR is deployed correctly, do one of the following:

- In LiveCycle Configuration Manager, deselect the EARs that are already deployed and running and select only those EARs that are yet to be deployed (in this case, adobe-workspace-client.ear), and click **Deploy**.
- Deploy the Workspace ES2 EAR file (adobe-workspace-client.ear) manually from WebSphere Administrative Console. After this step, run the LiveCycle Configuration Manager again, and then select the tasks that follow the deployment of EARs in LiveCycle Configuration Manager.

Note: When LiveCycle Configuration Manager fails to deploy any EAR files, you can deploy them manually from the administrative console of your application server. Follow this general outline for manually deploying EAR files and continuing the configuration tasks in LiveCycle Configuration Manager:

- Manually deploy the required EAR files from the administrative console of your application server.
- Close the currently running instance of LiveCycle Configuration Manager.
- Launch LiveCycle Configuration Manager again, and select the tasks that follow the deployment of EARs that you completed manually.

Deploying LiveCycle ES2 in a distributed environment

If the application server instance (WebSphere) or Managed Server (WebLogic) is on a different server, and not on the server with Deployment Manager (WebSphere) or Admin Server (WebLogic), you must manually deploy the LiveCycle ES2 EAR files. If you attempt to deploy by using LiveCycle Configuration Manager, you receive an error message indicating that deployment failed.

If you are experiencing this error when using LiveCycle Configuration Manager to deploy the EAR files, complete the following procedure.

➤ **To deploy the LiveCycle ES2 EAR files:**

1. Exit LiveCycle Configuration Manager.
2. Manually deploy the EAR files. (See the "Manually Deploying" chapter in *Installing and Deploying LiveCycle ES2* for your application server.)
3. Run LiveCycle Configuration Manager and select the tasks only after you deploy LiveCycle ES2 EAR files.

LiveCycle ES2 module deployment validation failure on WebSphere

While manually configuring WebSphere for LiveCycle ES2, you may see this error:

```
Component deployment failed validation. Cannot connect to server container
```

If so, complete the following procedure.

➤ **To increase the maximum transaction time-out and ORB service values:**

1. In the WebSphere Administrative Console navigation tree, click **Servers** > **Application Servers** and, in the right pane, click the server name.
2. Under Container Settings, click **Container Services** > **Transaction Service**.
3. Increase all of these time-out values to address this LiveCycle ES2 error:
   - **Total transaction lifetime timeout:** `1800`

- **Async response timeout:** `1800`
- **Maximum transaction timeout:** `1800`

4. Restart WebSphere Application Server.

Could not start/create deployment errors on Solaris

Download the GNU tar tool and use it to extract all files on a Solaris environment. Do not use the Solaris tar command to extract files otherwise errors such as missing files and log entries similar to the following will occur:

```
ERROR[org.apache.catalina.core.ContainerBase.[jboss.web].[localhost].[/invok
er]] Exception starting filter ReadOnlyAccessFilter

java.lang.ClassNotFoundException:
org.jboss.invocation.http.servlet.ReadOnlyAccessFilter
```

Application Server configuration validation fails when running LiveCycle Configuration Manager again

When you run LiveCycle Configuration Manager again after running configuration and validation once, if you skip deploying the adobe-lcm-lcvalidation.ear, an error message is displayed that validation of the application server configuration failed. This error occurs even though you had deployed the adobe-lcm-lcvalidation.ear file and validated the configuration.

To avoid this error, select the task to deploy adobe-lcm-lcvalidation.ear and validate the application server configuration on all subsequent runs of LiveCycle Configuration Manager.

Troubleshooting your application server

This section describes possible issues you may encounter with your application server and how to troubleshoot them using log files.

Application server does not start

If the server does not start, perform these checks:

- Check the application server log file.
- Check whether the server is already running. If so, it continues to run but fails to initialize. Stop and restart the application server.
- Check whether another process is using any of the ports configured for LiveCycle ES2.
- See the manufacturer's documentation.

JBoss Application Server issues

Content Services ES2 deployment fails after a JBoss server restart

If deployment of Content Services ES2 fails after a JBoss Application Server restart, add Java processes and *mysqld-max-nt.exe* to the safe-processes list of your anti-virus software.

The error code 13 that appears in the logs is a permission-denied error.

WebLogic Server issues

LiveCycle Configuration Manager hangs when redeploying an EAR file

LiveCycle Configuration Manager may hang at an incomplete stage when redeploying a LiveCycle ES2 EAR file, and fail to deploy the new EAR file.

This problem occurs occasionally when WebLogic Server does not release a currently deployed EAR file. Perform the following procedure to correct this problem:

1. Stop LiveCycle Configuration Manager,
2. Manually remove the existing version of the LiveCycle ES2 EAR file that you want to update.
3. Stop and restart the administrative and managed WebLogic Servers.
4. Run LiveCycle Configuration Manager to deploy the new EAR file.

WebSphere Application Server issues

Problems deleting directory tree on WebSphere

If you are unable to delete a WebSphere directory tree, it may be because the path is too long. For example, you may be unable to delete the following file, which is used by LiveCycle Rights Management ES2:

```
C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\installedApps\
Server1Node01Cell\LiveCycleES2.ear\rest-webservice.war\WEB-INF\classes\com\
adobe\rightsmanagement\webservices\rest\serialization\objects\
RestPolicyInfoSerializer$InternalPolicyInfoStruct.class
```

To resolve this problem, map a drive to the location from which you want to delete files (in this example, `C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\installedApps`) and then delete the files from that mapped drive.

➤ **To map a drive location and delete files from the command line:**

1. Map a drive to the location you want to delete from. For example:

   ```
   net use L: \\<hostname>\C$\Program Files\IBM\WebSphere\AppServer\profiles
   \AppSrv01\installedApps
   ```

2. Change to the mapped drive. For example:

   ```
   L:
   ```

3. Delete the files. For example:

   ```
   del /s /q *
   ```

4. Change back to the original drive. For example:

   ```
   C:
   ```

5. Delete the drive mapping. For example:

```
net use delete L:
```

Alternatively, you can rename a directory in the path to one-character name to make the path shorter and then delete the files.

OutOfMemory PermGen space error (WebSphere on Solaris)

If you are using WebSphere application server on Solaris, LiveCycle Configuration Manager might fail during LiveCycle ES2 component deployment with the `OutOfMemory PermGen space` error.

To avoid this error, add the following parameter in application server's Generic JVM Arguments section:

```
-XX:MaxPermSize=512m
```

Restart the application server and then proceed normally. For more information, see *Preparing to Install LiveCycle ES2 (Singer Server)*.

Application server stops responding (WebSphere and DB2 on AIX)

If multiple users use Workbench ES2 at the same time in a WebSphere+DB2 configuration on AIX, LiveCycle ES2 server might stop responding when a user tries to deploy an application.

To avoid this error, do the following:

1. Open *[WorkbenchES2_HOME]*\workbench.ini file using a text editor.
2. Locate the following line, and change the value to `1`.

```
-Dcom.adobe.workbench.unsupported.service.cache.batch.threads=5
```

3. Save and close the file.

NullPointerException while accessing Contentspace ES2

If you encounter a NullPointerException while trying to access Contentspace ES2, restart the application server and access Contentspace ES2 again.

This issue is observed only on WebSphere 7.x and not on WebSphere 6.x.

Troubleshooting your LiveCycle ES2 database

This section describes possible issues you may encounter with your LiveCycle ES2 database and suggests steps for avoiding or working around them.

If your database is failing to bootstrap, perform the following check:

- The database has adequate disk space to grow.
- The database configuration meets minimum database configuration requirements. For configuration requirements for your database type, see *Preparing to Install LiveCycle ES2 (Singer Server)*.
- See the manufacturer's documentation.

Note: If your database administrator cannot successfully bootstrap the database after these checks, the database manufacturer needs to be contacted immediately.

To ensure continuous availability and performance of your LiveCycle ES2 database, do the tasks:

- Continuously monitor the database as it is running for performance related problems.
- Continuously monitor database growth to ensure adequate disk space is available at all times.
- Consider LiveCycle ES2 component usage: Intense LiveCycleProcess Management ES2 applications will grow the database more substantially than intense PDF Generator ES2 applications.
- Review manufacturers database performance documentation.

IBM DB2 configuration settings

If you are running LiveCycle ES2 with a DB2 database and the computer stops responding, check the server log files for deadlock-related messages. If such messages are in the log files, change your DB2 configuration parameters:

- Set the `LOCKTIMEOUT` parameter to `15`.
- Double the values for the `APPLHEAPSZ`, `STMTHEAP`, and `SORTHEAP` parameters.

You must then restart the database and application server.

Troubleshooting with log files

This section describes how to troubleshoot LiveCycle ES2 using the log files.

LiveCycle ES2 log file

By default, the LiveCycle ES2 log file is located in the *[LiveCycleES2 root]* directory and is named log.txt. This log file is useful for LiveCycle ES2 failure analysis and may be required when dealing with Adobe Enterprise Support.

LiveCycle Configuration Manager log file

By default, the LiveCycle Configuration Manager log file is in *[LiveCycleES2 root]*\ConfigurationManager\log and may be named lcm.0.log. The log files are useful for LiveCycle Configuration Manager failure analysis and may be required when dealing with Adobe Enterprise Support.

Troubleshooting your application server using log files

Information in the application server log files can be used to help troubleshoot problems you are experiencing with your LiveCycle ES2 implementation. If the log files do not provide enough information to help you troubleshoot problems, you can enable verbose logging to increase logging details. Verbose logging should be enabled only during troubleshooting; otherwise, it slows system performance and consumes additional disk space for log files.

Note: It is recommended that you work with Adobe Enterprise Support to troubleshoot problems when using verbose log files.

JBoss log file

By default, the JBoss log files are located in *[LiveCycleES2 root]*\jboss\server\all\log and are named boot.log and server.log. The log files are useful for JBoss Application Server and LiveCycle ES2 bootstrapping failure analysis, and may be required when dealing with Adobe Enterprise Support.

If the log files do not provide enough information to help you troubleshoot problems, you can enable TRACE logging to increase logging details by modifying the *[appserver root]*/conf file.

Note: Ensure that you back up the *[appserver root]*/conf file before you modify it.

➤ **To enable TRACE logging in JBoss:**

1. From a command prompt, go to the *[appserver root]*/conf directory.
2. Edit the `log4j.xml` configuration file using a text editor.
3. Locate the `<root>` logger element in the file and change it as follows:

```
<root>
     <priority value=”INFO” />
<appender-ref ref=”FILE” />
</root>
```

4. Above the `<root>` logger element, type the following text:

```
<category name="org.jboss.ejb">
   <priority value="TRACE" class="org.jboss.logging.XLevel"/>
   <!--Comment the line below if you want to disable tracing -->
   <appender-ref ref=”TRACE_FILE” />
   <appender-ref ref=”FILE” />
</category>
```

5. Locate `<appender name=”FILE”` in the file and change or enter the following line:

```
<param name=”Threadhold” value=”DEBUG” />
```

6. Locate `<!-- A size based file rolling appender` in the file and paste the appender in the line below:

```
<appender name="TRACE_FILE"
   class="org.jboss.logging.appender.RollingFileAppender">
   <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
   <param name="File" value="${jboss.server.home.dir}/log/trace.log"/>
   <param name="Append" value="false"/>
   <param name="MaxFileSize" value="5MB"/>
   <param name="MaxBackupIndex" value="2"/>
   <layout class="org.apache.log4j.PatternLayout">
   <param name="ConversionPattern" value="%d %-5p [%c] %m%n"/>
   </layout>
</appender>
```

7. Save and close the `log4j.xml` file.

➤ **To disable TRACE logging in JBoss:**

1. From a command prompt, go to the *[appserver root]*/conf directory.

2. Edit the `log4j.xml` configuration file using a text editor.

3. Locate the `<root>` logger element in the file and change it as follows:

```
<root>
   <priority value=”INFO” />
   <appender-ref ref=”FILE” />
</root>
```

4. Above the `<root>` logger element, enter the following text:

```
<category name="org.jboss.ejb">
   <priority value="TRACE" class="org.jboss.logging.XLevel"/>
   <!--Comment the line below if you want to disable tracing -->
   <appender-ref ref=”TRACE_FILE” />
   <appender-ref ref=”FILE” />
</category>
```

5. Locate `<appender name=”FILE”` in the file and change or enter the following line:

```
<param name=”Threadhold” value=”DEBUG” />
```

6. Locate `<!-- A size based file rolling appender` in the file and paste the appender in the line below:

```
<appender name="TRACE_FILE"
class="org.jboss.logging.appender.RollingFileAppender">
   <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
   <param name="File" value="${jboss.server.home.dir}/log/trace.log"/>
   <param name="Append" value="false"/>
   <param name="MaxFileSize" value="5MB"/>
   <param name="MaxBackupIndex" value="2"/>
   <layout class="org.apache.log4j.PatternLayout">
   <param name="ConversionPattern" value="%d %-5p [%c] %m%n"/>
   </layout>
</appender>
```

7. Save and close the `log4j.xml` file.

WebLogic log file

By default, the WebLogic log file is located in `/var/log/httpd/error_log`. The log files are useful for WebLogic Server and LiveCycle ES2 bootstrapping failure analysis, and may be required when dealing with Adobe Enterprise Support.

If the log file does not provide enough information to help you troubleshoot problems, you can specify the level of tracing in the log file to increase logging details. To do this, modify the `LogLevel` parameter in the *[appserver root]*/conf/httpd.conf file. `LogLevel` sets how verbose the error messages in the error logs are. `LogLevel` can be set (from least verbose to most verbose) to `emerg`, `alert`, `crit`, `error`, `warn`, `notice`, `info`, or `debug`. The default `LogLevel` is `warn`.

Note: Ensure that you back up the *[appserver root]*/conf/httpd.conf file before you modify it.

➤ **To enable debug LogLevel in WebLogic:**

1. From a command prompt, navigate to the *[appserver root]*/conf directory.

2. Edit the `httpd.conf` configuration file using a text editor.
3. Locate the `LogLevel` in the file and change it as follows:

   ```
   LogLevel debug
   ```

4. Save and close the `httpd.conf` file.

When you have completed troubleshooting, repeat steps 1 to 4 but change the `LogLevel` to `warn`.

WebSphere log file

By default, the WebSphere log file is located in *[appserver root]*/logs/server1/trace.log. The log files are useful for WebSphere Application Server and LiveCycle ES2 bootstrapping failure analysis, and may be required when dealing with Adobe Enterprise Support.

If the log files do not provide enough information to help you troubleshoot problems, in the WebSphere Administrative Console, you can enable TRACE logging to increase logging details.

➤ **To enable TRACE in WebSphere:**

1. Log in to WebSphere Administrative Console and, in the navigation tree, click **Troubleshooting** > **Logs and Trace** and, in the list of servers, click **server1**, and then click **Change Log Detail Levels**.
2. Select **Enable Trace** and, in the **Trace Specification** box, type `com.adobe.*=all=enabled:com.adobe.framework.UITools=all=disabled`.

   [appserver root]/profiles/*[profile_name]*/logs/*[server name]*

Viewing the JVM system output and error logs

The JVM system output and error logs are valuable tools for troubleshooting problems with your server.

➤ **To view the JVM system output and error logs:**

1. Log in to WebSphere Administrative Console and, in the navigation tree, click **Troubleshooting** > **Logs and Trace**.
2. Click the name of the application server, and then click **JVM Logs**.
3. Click the **Runtime** tab and, under either System.out (to view the JVM system output log) or System.err (to view the error log). click **View**. If any of the selections are unavailable, you can view them from the **Configuration** tab by specifying the SystemOut.log and SystemErr.log file names. By default the files are located in the following location:

   [appserver root]/profiles/*[profile_name]*/logs/*[server name]*

➤ **To prevent Java core dumps from appearing during EAR deployment or when you restart the server:**

Ensure that JAVA_HOME_32 is set only as an environment variable and is not included in the PATH.

➤ **To prevent repetitive "reindexImpl started" error messages in the WebSphere server logs:**

After Content Services ES2 is deployed, you may observe the following error message appearing repetitively in SystemOut.log:

```
IndexTransact I org.alfresco.repo.node.index.IndexTransactionTracker
reindexImpl reindexImpl started:
org.alfresco.repo.node.index.IndexTransactionTracker@290c290c
```

To resolve the issue, follow these steps:

1. *(WebSphere 6.1)* In the WebSphere navigation tree, click **Servers > Application servers.**

   (WebSphere 7.x) In the WebSphere navigation tree, click **Servers > Server Types > Websphere application servers.**

2. Click an application server listed in the right pane.

3. Click **Troubleshooting > Change Log Level Details.**

4. In the **Components** list, navigate to the `org.alfresco.repo.node.index.IndexTransactionTracker` package.

5. Click the `org.alfresco.repo.node.index.IndexTransactionTracker` package and select **No Logging**.

6. Repeat steps 1-5 for the **Configuration** and **Runtime** tabs, and for all nodes in the cluster.

➤ **To prevent repetitive "Failed job" error messages originating from the Quartz scheduler:**

If you are using the SOAP port for any of your LiveCycle ES2 services, you may encounter an issue that causes repetitive "Failed job" error messages, originating from the Quartz scheduler, to appear in the WebSphere log file for all nodes in the cluster.

These error messages continue to appear even after the node servicing the request has been shut down and another node has completed the pending job.

To avoid this issue, use the admin console to change the log configuration for all nodes in the WebSphere cluster. Set the log level for the following packages to `severe`:

- org.quartz.impl.jdbcjobstore
- com.adobe.idp.scheduler.jobstore.DSCJobStoreTX

Deleting the application server transaction log file

If the component solutions fail to deploy for any reason, the application server that hosts LiveCycle ES2 does not restart because it attempts to recover what it interprets as rolled back transactions but fails to do so. To resolve this issue, locate and delete the application server transaction log file and restart the application server.

Troubleshooting Administration Tasks

This chapter discusses possible issues with your deployed LiveCycle ES2 environment and with the LiveCycle Administration Console interface, as well as when to contact Adobe Enterprise Support.

See also

"Getting help" on page 10

"Additional information" on page 9

Note: Refer to the *LiveCycle ES2 Error Code Reference* for a list of errors that you may encounter while using LiveCycle ES2, their causes, and the actions that you can take to resolve them.

Login issues

If you cannot access any of the LiveCycle ES2 web applications, such as LiveCycle Administration Console or LiveCycle Workspace ES2 web pages, ensure that the following conditions exist:

- The LiveCycle ES2 database tables were created and the user has full rights to the database.
- The database server is accessible when you ping it.

Note: Only administrators with the appropriate roles can access the Workspace ES2 application. For information about roles and permissions, see *LiveCycle ES2 Administration Help*.

If you cannot log in to LiveCycle Administration Console as a user with administrator privileges, do the following tasks:

- Try to log in as Super Administrator, that is, use Administrator as the user ID. This user always checks in to the local database before going to any other authentication provider.
- Ensure that the custom SPI conditions described below exist.
- Check whether the administrator user has all the required roles. If your LDAP tree has an administrator name, the LiveCycle ES2 roles may have been overwritten. Contact Adobe Enterprise Support.

If you cannot log in to the Workspace ES2 web pages, ensure that the following conditions exist:

- The host file contains the Workspace ES2 server name.
- The Workspace ES2 server is accessible when you ping it.
- Neither the client nor the Workspace ES2 server are blocked by a firewall.
- The QLC file has the correct settings, such as the Workspace ES2 server name, JNDI, or URL provider port.

If you are using a custom SPI and cannot log in, ensure that the following conditions exist:

- Check the config.xml file to ensure that the association between the domain and its authentication provider is correct. If it is incorrect or absent, login authentication will fail. The domain must be configured in the config.xml file as follows:

```
<node name="Domains">
   <map/>
   <node name="<Domain_Name">
      <map>
         <entry key="description" value="any suitable discription"/>
         <entry key="name" value="<Domain_Name>"/>
         <entry key="isLocal" value="true/false"/>
      </map>
      <node name="AuthConfigs">
         <map/>
            <node name="<Profile_Name">
               <map>
                  <entry key="authProviderNode"
value="/Adobe/LiveCycle/Config/UM/AuthProviders/<AuthenticationProvider"/>
               </map>
            </node>
      </node>
      <node name="DirectoryConfigs">
```

- Every domain keeps a reference of the authentication provider it uses for authentication. Ensure that the authentication provider is configured in the config.xml file as follows:

```
<root type="system">
    <map/>
    <node name="Adobe">
      <map/>
      <node name="LiveCycle">
        <map/>
        <node name="Config">
          <map/>
          <node name="UM">
            <map/>
            <node name="AuthProviders">
              <map/>
              <node name="Authentication Provider">
                <map>
                  <entry key="configured" value="true"/> SHOULD BE TRUE
                  <entry key="visibleInUI" value="false"/>
                  <entry key="enabled" value="true"/>
                  <entry key="allowMultipleConfigs" value="false"/>
                  <entry key="className"
value="com.adobe.idp.um.provider.authentication.CertificateAuthProviderImpl"
/> SHOULD BE NON NULL
                  <entry key="order" value="5"/>
```

Accessing the Services page in LiveCycle Administration Console on JBoss

If you go to the Services page in LiveCycle Administration Console and the page appears blank, perform this workaround to ensure that the page displays correctly:

➤ **To display the Services page correctly:**

1. Start JBoss Application Server manually (not using the Windows service) using the command:

   ```
   run -b localhost -c all
   ```

2. In the Windows hosts file located in the C:\windows\system32\drivers\etc. . . \hosts directory, add the IP address and host name of the LiveCycle ES2 server.

Login pages appear even after SSO authentication

The Workspace ES2 and LiveCycle Administration Console login pages appear even after SSO has been configured and the user has authenticated using it.

To resolve this issue, create a new realm that filters to /um/login and add it to the policy.

Example resolution for SiteMinder

To resolve this issue for SiteMinder 6.0, perform these steps while configuring SiteMinder with LiveCycle ES2:

1. Create a realm named UM Login that filters to /um/login. All authentications will route through this realm.
2. Create a rule for the new realm. While doing so, specify '*' as the resource.
3. In the Response Properties dialog box, specify **Auth Response** as the name and the add an attribute with the value **ADB_USER=<%userattr="cn"%>**.
4. Add UM Login to the policy as a rule.

Performance considerations

If you are experiencing performance issues with LiveCycle ES2, consider the following:

Synchronization issues: If many threads are waiting at the same time in the same part of the code, obtain a thread dump when the congestion passes.

Caution: Thread dumps may disable the JVM.

Slow external resources: If many threads are waiting for a return message from an external source, obtain a thread dump to find threads that are waiting for sources such as databases or LDAP servers.

Slow GC collection: If `verbosegc` performs compaction frequently, reduce the amount of garbage generated by the application by introducing object pooling or caching. If the log shows long garbage collection cycles in `verbosegc`, reduce the maximum heap size.

High user CPU: If your CPU is running at 75% or higher, consider these options:

- Reduce the pool size of the web container or ORB threads.
- Reduce the number of database connections on the database server.
- If you experience consistently high CPU usage, consider adding processing resources.
- If the CPU is on the database server, reduce the datasource maximum connection setting.

Improving performance during asynchronous service invocation

For improving performance during asynchronous invocation of services, set the following JVM arguments:

```
-Dadobe.work-manager.queue-refill-interval=1
-Dadobe.workmanager.memory-control.enabled=false
```

For JBoss, add these arguments to the run.bat file (Windows) or the run.sh file (UNIX).

See "Configuring the JVM arguments" in the *Installing and Deploying LiveCycle ES2 for WebSphere* guide for information on setting JVM arguments for WebSphere.

See "Configuring the JVM arguments" in the *Installing and Deploying LiveCycle ES2 for WebLogic* guide for information on setting JVM arguments for WebLogic.

Remote invocation fails with application servers on pure IPv6

If your LiveCycle ES2 server is implemented in a pure IPv6 environment, remote invocation of services on the LiveCycle ES2 server might fail. This is an issue with Sun JDK used with the clients. To avoid this error, use the IBM JDK with clients when LiveCycle ES2 is deployed on application servers in a pure IPv6 environment.

Process Management ES2 performance issue on Oracle

New for 9.0.0.2

Process Management ES2 throughput for Oracle databases is sometimes observed to deteriorate over time. The LiveCycle development team has made some SQL*Plus scripts available to help resolve this issue. These scripts improve performance in scenarios having a large number of users.

You can contact Adobe Customer Support and ask for scripts associated with the TechNote titled "Process Management ES2 performance issue on Oracle" (document ID *cpsid_85089*).

Improving Windows Server performance with LDAP

Using connection pooling on the search connection can decrease the number of ports needed by as much as 50% because that connection always uses the same credentials for a given domain, and the context and related objects are closed explicitly.

➤ **To configure your Windows Server for connection pooling:**

1. Start the registry editor by selecting **Start** > **Run** and, in the **Open** box, type `regedit`, and then click **OK**.
2. Go to the registry key
   `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters`
3. In the right pane of the registry editor, look for the **TcpTimedWaitDelay** value name. If the name does not appear, select **Edit** > **New** > **DWORD Value** to add it.
4. In the name box, type `TcpTimedWaitDelay`.

5. If you do not see an insertion point and **New Value #** inside the box, right-click inside the right panel, select **Rename** from the menu and then, in the **name** box, type `TcpTimedWaitDelay`.

6. Repeat steps 4 and 5for the following value names: `MaxUserPort`, `MaxHashTableSize`, and `MaxFreeTcbs`.

7. Double-click inside the right pane to set the **TcpTimedWaitDelay** value; under Base, select **Decimal** and, in the **Value** box, type `30`.

8. Double-click inside the right pane to set the **MaxUserPort** value; under Base, select **Decimal** and, in the **Value** box, type `65534`.

9. Double-click inside the right pane to set the **MaxHashTableSize** value; under Base, select **Decimal** and, in the **Value** box, type `65536`.

10. Double-click inside the right pane to set the **MaxFreeTcbs** value; under Base, select **Decimal** and, in the **Value** box, type `16000`.

Caution: Serious problems may occur if you modify the registry incorrectly by using Registry Editor or by another method. These problems may require that you reinstall your operating system. Modify the registry at your own risk.

Scheduler service configuration for nondefault JNDI URLs

(Non-clustered environments only)

To function correctly, the Scheduler service may require some additional configuration.

JBoss

On JBoss, if the JNDI URL differs from the default JNDI URL for the application server (that is, for JBoss: jnp://localhost:1099), this is the JNDI URL for the IDP_DS that is managed by your application server:

```
org.quartz.dataSource.idp.java.naming.provider.url
```

➤ **To set the scheduler properties:**

1. Create a new file named dscscheduler.properties.

2. Set the values of the above properties as necessary for the app server node, such as the following text:

```
org.quartz.dataSource.idp.java.naming.provider.url =
   jnp://localhost:1099/
org.quartz.jobstore.isClustered = true
org.quartz.scheduler.instanceId = AUTO
```

3. Add the JVM argument `-Dadobe.idp.scheduler.properties=`*`[Path to this file]`*`/dscscheduler.properties` to the application server startup scripts/configuration.

WebSphere

On WebSphere, if the JNDI URL differs from the default JNDI URL for the application server (that is, for WebSphere: iiop://localhost:2809), this is the JNDI URL for the IDP_DS that is managed by your application server:

```
org.quartz.dataSource.idp.java.naming.provider.url
```

➤ **To set the scheduler properties:**

1. Create a new file called dscscheduler.properties
2. Set the values of the above properties as necessary for the app server node, such as the following values:

```
org.quartz.dataSource.idp.java.naming.provider.url =
   iop://localhost:2809/
org.quartz.jobstore.isClustered = true
org.quartz.scheduler.instanceId = AUTO
```

3. Add the JVM argument `-Dadobe.idp.scheduler.properties=`*`[Path to this file]`*`/dscscheduler.properties` to the application server startup scripts/configuration.

WebLogic

On WebLogic, if the JNDI URL differs from the default JNDI URL for the application server (that is, for WebLogic: t3://localhost:7001), this is the JNDI URL for the IDP_DS that is managed by your application server:

```
org.quartz.dataSource.idp.java.naming.provider.url
```

➤ **To set the scheduler properties**

1. Create a new file named dscscheduler.properties
2. Set the values of the above properties as necessary for the app server node, such as the following values:

```
org.quartz.dataSource.idp.java.naming.provider.url =
   t3://localhost:7001/
org.quartz.jobstore.isClustered = true
org.quartz.scheduler.instanceId = AUTO
```

3. Add the JVM argument `-Dadobe.idp.scheduler.properties=`*`[Path to this file]`*`/dscscheduler.properties` to the application server startup scripts/configuration.

FileNet API excessive logging and performance issues on WebLogic

For LiveCycle ES2 installed on a WebLogic Server, the FileNet APIs might experience performance issues owing to the excessive logging. To improve the performance in such cases, you should change the logging level from 'debug' to 'Fatal'.

1. On FileNet Content Server, navigate to the <Install_DIR>\FileNet\ContentEngine\config\samples directory.
2. Copy the log4j.properties.client file to the LiveCycle ES2 server machine and rename it to `log4j.properties`.
3. Open the log4j.properties file and comment out the entries for the appenders FileNetTraceAppender and FileNetTraceRollingAppender:

```
#=== FileNetTraceAppender

log4j.appender.FileNetTraceAppender=org.apache.log4j.FileAppender
log4j.appender.FileNetTraceAppender.File=/p8_api_trace.log
# This is the layout that the TraceLoggingConfiguration framework on the server
```

```
uses.
# To use this layout , jace.jar must be present in the classpath.
#log4j.appender.FileNetTraceAppender.layout=com.filenet.apiimpl.util.TraceLayout
# Comment out the following lines if using the FileNet TraceLayout
log4j.appender.FileNetTraceAppender.layout=org.apache.log4j.PatternLayout
log4j.appender.FileNetTraceAppender.layout.ConversionPattern=%d %5p [%t] -
%m\r\n
#=== FileNetTraceRollingAppender
log4j.appender.FileNetTraceRollingAppender=org.apache.log4j.RollingFileAppender
log4j.appender.FileNetTraceRollingAppender.File=/p8_api_trace.log
log4j.appender.FileNetTraceRollingAppender.MaxFileSize=100MB
log4j.appender.FileNetTraceRollingAppender.MaxBackupIndex=1
# This is the layout that the TraceLoggingConfiguration framework on the server
uses.
# To use this layout , jace.jar must be present in the classpath.
#log4j.appender.FileNetTraceRollingAppender.layout=com.filenet.apiimpl.util.Trac
eLayout
# Comment out the following lines if using the FileNet TraceLayout
log4j.appender.FileNetTraceRollingAppender.layout=org.apache.log4j.PatternLayout
log4j.appender.FileNetTraceRollingAppender.layout.ConversionPattern=%d %5p [%t]
- %m\r\n
```

- Set FileNet logger to FATAL and remove FileNetTraceAppender and FileNetTraceRollingAppender from the logger:

  Replace

  ```
  log4j.logger.filenet_error = error, FileNetConsoleAppender,
  FileNetErrorRollingAppender, FileNetTraceRollingAppender
  ```

  with

  ```
  log4j.logger.filenet_error = fatal, FileNetErrorRollingAppender
  ```

- Set FileNet API logger to FATAL:

  Replace

  ```
  #log4j.logger.filenet_error.api = warn
  ```

  with

  ```
  log4j.logger.filenet_error.api = fatal
  ```

4. Save the log4j.properties file.
5. Add the path of the folder containing the log4j.properties file to the FileNet Component ID entry present in the *adobe-component-ext.properties* file placed in the LiveCycle application server. In WebLogic application server, this file is in *[WL_HOME]*/user_projects/domains/<domain name>.

   For example, if log4j.properties file is stored at location: 'C:/log4j_file/log4j.properties', then add "C:/log4j_file" to the adobe-component-ext.properties file.

6. Restart the application server.

Sporadic out-of-memory errors with WebLogic/JRockit

If you observe sporadic out-of-memory errors with WebLogic 10gR3/JRockit JVM R26.4 (or later), do one of the following tasks:

- Increase the JVM heap-size to a value greater than 4GB.

- Add the following argument to the JVM start-up options:

```
-XXcompressedRefs=false
```

Cache item expiry warnings when many concurrent users are accessing Content Services ES2

New for 9.0.0.2

The following warning message may appear in the application server logs when multiple concurrent users are accessing Content Services ES2:

```
ReadWriteCach W org.hibernate.cache.ReadWriteCache handleLockExpiry An item
was expired by the cache while it was locked (increase your cache timeout):
org.alfresco.repo.domain.hibernate.NodeImpl.properties
```

To resolve this issue, set the following JVM argument in the application server startup scripts to use the Least Recently Used (LRU) algorithm instead of the default heuristic algorithm to refresh `ehcache`:

```
-Dnet.sf.ehcache.use.classic.lru=true
```

Uninstall issues

Removing JBoss/MySQL services, folders, and files

If you choose to retain JBoss or MySQL while uninstalling LiveCycle ES2, you may not be able to uninstall these applications later using uninstallers in the *[LiveCycleES2_root]*\Uninstall_Adobe JBoss for Adobe LiveCycle ES2 folder or the *[LiveCycleES2_root]*\Uninstall_Adobe MySQL for Adobe LiveCycle ES2 folder.

Do the following to uninstall JBoss/MySQL manually:

1. Delete the services with the name "JBOSS_FOR_ADOBE_LIVECYCLE_ES2" or "MySQL for Adobe LiveCycle ES2" using the following command:

```
sc delete <service name>
```

2. Remove JBoss/MySQL folders and files.

Turnkey mode is disabled after an incomplete install or uninstall

The Turnkey mode of installation becomes unavailable after an unsuccessful attempt to install or uninstall LiveCycle ES2.

To resolve this issue, check if services with the name "JBOSS_FOR_ADOBE_LIVECYCLE_ES2" or "MySQL for Adobe LiveCycle ES2" already exist on the system. If these services exist, delete them and their related registry entries before initiating a fresh Turnkey install.

Use the following command to delete the services:

```
sc delete <service name>
```

Troubleshooting output errors

Some output files are not converted from a watched folder

Some LiveCycle ES2 servers (most commonly those running on UNIX) invokes the conversion process before all the associated files are copied to the watched folder, and therefore missing some of the files. To avoid this issue, create a folder outside the watched folder hierarchy, copy all of the required files into the folder, and then copy the entire folder to the watched folder root.

OutputIVS Printer not found

If you see this error, then follow the steps below:

1. In the URL \\<*servername*>\<*printername*>, specify the full name of the printer.
2. Add the printer to the windows account before starting JBoss.
3. Enable JBoss to run in the context of a valid user. To perform this task, change the properties of the JBoss service by clicking the **Log On** tab and selecting **This account**. Supply a valid user name and password and restart the service for this change to take effect.
4. The server that hosts LiveCycle ES2 needs to have permissions to access the printer; the connection fails if it does not have the necessary permissions.

Diagnosing cache related problems

If you have an older form that is being cached and re-used instead of a newer or changed version of the form, you will see unexpected results after generation. Make the following updates to resolve this problem:

- Ensure that the cache validation settings are set so that the template you are using had been revalidated. If an update is made and the form is re-rendered within a few seconds, it is possible that the revalidation checkpoint time has not yet occurred and the updated form will not be picked up. To avoid this issue, set the validation method to **Unconditionally** or manually reset the cache.
- Ensure that uniquely generated templates being passed by value have empty UUID strings, so that they are treated as non cacheable.

If you have a cacheable form but are not seeing the performance benefit of caching, review the following settings:

- Ensure that the Designer ES2 settings allow for caching rendered forms.
- Ensure that the render options allow the render cache to operate. In general, flat PDF document renderings cannot be cached, whereas render-at-client forms benefit from the render cache.
- Search within the physical directory structure of the cache to find the cached template and rendered elements of your document to confirm that the form is being physically cached.
- Make a copy of the form and manually removed the form's UUID. This creates a version of the form that will always bypass the cache. Then compare the performance of the non-cached and cached version of the form to determine what benefit the cache is yielding.

Some output files are lost when a clustered WebSphere Application Server shuts down

Some expected output files may be lost when one WebSphere Application Server of a cluster shuts down. One possible cause is that invocation requests from a watched folder cannot, for various reasons related to the shutdown, access files that are placed in the staging folder.

Complete the following procedure to recover the lost files.

➤ **To recover files from a staging folder:**

1. Ensure that the node is restarted.
2. Log in to LiveCycle Administration Console and click **Services** > **Applications and Services** > **Endpoint Management**.
3. In the Provider list, select **WatchedFolder**, and then click **Filter** to display the endpoints for the watched folder.
4. Select the check box for the service name endpoint, and then click **Disable**. The watched folder is now disabled from processing new files.
5. Wait for LiveCycle ES2 to recover and process any files it can access. The waiting time depends on the time that is required to process the operation being called and the number of files being recovered.
6. Check the date-time stamp of the files that remain in the staging directory to identify which files are old enough to be files that were lost due to the shutdown.
7. Copy the lost files to the input directory.
8. Re-enable the watched folder to process new input files by repeating steps 2 to 4 and selecting **Enable**.

Password encryption error

When Federal Information Processing Standards (FIPS) mode is enabled in LiveCycle ES2 (set either during the LiveCycle ES2 configuration process or manually in the Core Systems Settings web pages within LiveCycle Administration Console), password encryption will not be applied to any document. If you attempt to encrypt the password on a FIPS-enabled document, the error "Password encryption is not permitted in FIPS mode" is displayed.

PDF output contains an unwanted orange watermark

A PDF file generated from a Microsoft Word document may have an unwanted orange (not gray) watermark "Adobe LiveCycle PDF Generator Evaluation". (A gray watermark results from other LiveCycle ES2 settings.) This error is typically caused when the example.ps file becomes corrupted (for example, by multiple installations of LiveCycle ES2 on the same server). To correct this problem, delete the C:\Documents and Settings*[user]*\Application Data\Adobe\Adobe PDF\Distiller\Startup\example.ps file from the application server that LiveCycle ES2 runs on and then restart the server. In a clustered environment, delete the file from all servers where it occurs and then restart each server that you deleted the file from.

Converting multiple 3D assemblies causes "No transaction" error

LiveCycle PDF Generator 3D ES2 may experience conversion failure with the exception "IllegalStateException: No transaction". This error occurs when converting a large number of 3D assemblies from a watched folder. To avoid this problem, perform the procedure below that is appropriate to your application server.

Note: You must perform the procedure on each application server instance to correct this error in a cluster configuration.

➤ **To correct a PDF Generator 3D ES2 "No transaction" error on JBoss Application Server:**

1. Open the *[JBoss root]*\server\all\deploy\jms\hajndi-jms-ds.xml file in a text editor.
2. Locate the `tx-connection-factory` element defined for `jndi-name adobe_JmsQueueXA` and insert the element in bold as follows:

```
<tx-connection-factory>
   <track-connection-by-tx>true</track-connection-by-tx>
   <jndi-name>adobe_JmsQueueXA</jndi-name>
   <max-pool-size>100</max-pool-size>
   <xa-transaction/>
   <rar-name>jms-ra.rar</rar-name>
```

3. Save the file.
4. Open the *[JBoss root]*\server\all\deploy\adobe-ds.xml file in a text editor and change the following settings:
   - Locate the `min-pool-size` element for `jndi-name IDP_DS` and change the value to `10`.
   - Locate the `idle-timeout-minutes` element and change the value to `20`.
5. Save the file and restart JBoss Application Server.
6. Log in to the LiveCycle Administration Console and click **Services** > **Applications and Services** > **Service Management**.
7. Find and click the **Generate3dPDFService: 1.0** service.
8. In the **Server Maximum Conversion Timeout** box, change the value to `2700`, and then click **Save**.

➤ **To correct a PDF Generator 3D ES2 "No transaction" error on WebLogic Server:**

1. In the WebLogic Server Administration Console, under Domain Structure, click **Services** > **JDBC** > **Data Sources**.
2. Under Change Center, click **Lock & Edit**, and then click **IDP_DS**.
3. Under the Configuration tab, on the Connection Pool page, change the **Initial Capacity** value to 10, and then click **Save**.
4. At the bottom of the page, click **Advanced**, and change the **Inactive Connection Timeout** value to 0.
5. Click **Save** and then click **Activate Changes**.
6. In the WebLogic Server Administration Console, under Domain Structure, click **Services** > **JTA** and, in the **Timeout Seconds** box, change the value to `2900`.

7. Click **Save** and then click **Activate Changes**.
8. Restart the WebLogic Server.
9. Open the LiveCycle Administration Console, and then click **Services** > **Applications and Services** > **Service Management**.
10. Find and then click the **Generate3dPDFService: 1.0** service.
11. In the **Server Maximum Conversion Timeout** box, change the value to `2700`, and then click **Save**.

➤ **To correct a PDF Generator 3D ES2 "No transaction" error on WebSphere Application Server:**

1. In the WebSphere Administrative Console, click **Servers** > **Application servers**, and then click the name of the server.
2. Under Container Settings, click **Container Services** > **Transaction Service**, make the following configuration changes, and then click **OK**:
   - In the **Total transaction lifetime timeout** box, type `2900`.
   - In the **Maximum transaction timeout** box, type `2900`.
3. Under Container Settings, click **Container Services** > **ORB Service**, in the **Request timeout** box, type `2900`, and then click **OK**.
4. In the navigation tree, click **Resources** > **JMS** > **Queue connection factories**, and then click **JobManagerQueueConnectionFactory**.
5. Under **Additional Properties**, click **Connection pool properties**, make the following configuration changes, and then click **OK**:
   - In the **Connection timeout** box, type `1800`.
   - In the **Maximum connections** box, type `100`.
   - In the **Reap time** box, type `1800`.
   - In the **Unused timeout** box, type `1200`.
6. In the navigation tree, click **Resources** > **JMS** > **Queue connection factories**, and then click **QueueConnectionFactory**.
7. Under **Additional Properties**, click **Connection pool properties**, make the following configuration changes, and then click **OK**:
   - In the **Connection timeout** box, type `1800`.
   - In the **Maximum connections** box, type `100`.
   - In the **Reap time** box, type `1800`.
   - In the **Unused timeout** box, type `1200`.
8. In the navigation tree, click **Resources** > **JDBC** > **Data sources** and then, in the right pane, click the name of a data source (for example, click **Livecycle - DB2 - IDP_DS**).
9. Under **Additional Properties**, click **Connection pool properties**, make the following configuration changes, and then click **OK**:
   - In the **Connection timeout** box, type `1800`.
   - In the **Minimum connections** box, type `10`.

- In the **Reap time** box, type `1200`.
- In the **Unused timeout** box, type `1200`.

10. Click **Save directly to the master configuration**, and then restart the WebSphere Application Server.
11. Open LiveCycle Administration Console and click **Services** > **Applications and Services** > **Service Management**.
12. Find and then click the **Generate3dPDFService: 1.0** service.
13. In the **Server Maximum Conversion Timeout** box, change the value to `2700`, and then click **Save**.

Browser cache may interfere with HTML rendition

If you are rendering to HTML an XDP form that contains references to images (either through links or embedded images) or contains data with links or image data, you should disable browser caching to avoid having cached data interfere with image display.

"Failure to create directory" error on Windows

On a Windows environment, you may encounter an error when converting PRN files to PDF format from a watched folder endpoint. This error is dependent on the Output parameter that is set in LiveCycle Administration Console.

When a watched folder endpoint's Output parameter is set to `%E/%F.pdf`, an output directory named with the originating file name extension is created to receive the converted files (`%E` = file name extension, `%F` = file name). For example, when the file *example.prn* is converted to PDF, a new directory is created (if it does not already exist) and the file is stored there. The result would be //prn/example.pdf.

On a Windows operating system, *prn* is reserved for system directories only and an attempt to create it generates the "Failure to create directory" error.

To avoid this error, set the value of the Output parameter for the PRN file to PDF endpoint to `%E_/%F or %E_Files.%F`. (See *LiveCycle ES2 Administration Help*.)

"Error while converting image to PDF" on Windows

On a Windows environment, you may encounter any one of a set of related errors when you convert an image file to PDF. To address this error, configure the Generate PDF service to use Acrobat image conversion.

➤ **To configure Acrobat image conversion:**

1. In LiveCycle Administration Console, navigate to **Services** > **Applications and Services** > **Service Management** and click **GeneratePDFService: 1.1**.
2. On the **Configuration** tab, in the Use Acrobat Image Conversion (Windows Only) box, type `true`, and then click **Save**.

Conversion of OpenOffice.org files to PDF fails on Solaris

Conversion of OpenOffice.org files to PDF may fail with a timeout error if a long temp directory name is specified on Solaris. The temp directory name should be 20 characters long at the maximum.

Do the following to resolve this issue:

1. Navigate to **Settings > Core System > Core Configurations** in the LiveCycle Administration Console.
2. Enter a shorter directory name in the **Location of temp directory** field. For example, `/tmp` or `/usr/temp`.
3. Click **OK**.

"Content URL 404 error" using template in LiveCycle Contentspace ES2

When using the doc_info.php template in LiveCycle Contentspace ES2, you may encounter an error message indicating a Content URL error. To address this error, replace the contents in the doc_info.php template.

➤ **To replace the contents:**

1. Log in to Contentspace ES2 as *administrator* and click **Company Home** > **Data Dictionary**,
2. In the right pane, click **Presentation Templates** and then click the **doc_info.php** content item.
3. Replace the contents of the **doc_info.php** file with the following text:

```
<?php

$document = $_ALF_MODEL["document"];

if ($document != null)
{
?>
   <h4>Current Document Info:</h4>
   <b>Name:</b> <?php echo($document->cm_name); ?><br>
   <b>Ref:</b> <?php echo($document->__toString()); ?><br>
   <b>Type:</b> <?php echo($document->type); ?><br>
   <b>DBID:</b> <?php echo($document->sys_node_dbid); ?><br>
   <b>Content:</b> <a href="<?php $pos1 =  strpos ( $document->url  ,
'ticket');echo ( substr( $document->url, 0, $pos1 -1  )) ?>"><?php $pos1 =
strpos ( $document->url  , 'ticket');echo ( substr( $document->url, 0, $pos1 -1
)) ?></a><br>

<?php
   if ($document->hasAspect("cm_countable") == true)
   {
?>
     <b>Counter:</b> <?php echo($document->cm_counter); ?><br>
<?php
   }
?>
   <b>Aspects:</b>
   <table>
<?php
   foreach ($document->aspects as $aspect)
   {
?>
     <tr><td><?php echo($aspect); ?></td></tr>
<?php
   }
?>
   </table>

   <b>Properties:</b>
   <table>
<?php
   foreach ($document->properties as $name=>$value)
   {
```

```
      echo("<tr><td>".$name." = ".$value."</td></tr>");
   }
?>
   </table>

<?php
}
else
{
   echo "No document found!";
}
?>
```

4. Save the file.

Bell Gothic font deprecated in LiveCycle ES2 and Designer ES2

The LiveCycle ES2 server and Designer ES2 installations no longer include the Bell Gothic font. If you are upgrading and have forms that use this font, either substitute it with one available on your system, or purchase it from the Adobe Store.

How a form using Bell Gothic is affected during runtime:

LiveCycle ES2 server: If this font is missing from the LiveCycle ES2 server, PDF Generator ES2 will automatically substitute the font in the generated PDF output file.

Designer ES2: When opening the form in Designer ES2, the Missing Fonts dialog appears and prompts the user to substitute the missing font.

"Cannot retrieve the resource from Repository Path" error

After upgrading from LiveCycle 8.0.x, exceptions such as the following warning appear in the server logs while retrieving images from the LiveCycle repository. Inspite of the exception, the PDF file generates as expected.

```
00000041 FormServerExc W com.adobe.livecycle.formsservice.logging.FormsLogger
logMessage
```

```
ALC-OUT-002-058: Cannot retrieve the resource from Repository Path. Authority
component retrieved is "<somestring>". Path component retrieved is
"/somefolder/someimage.jpg".Underlying Exception is : ALC-REP-018-000:
Resource [/somefolder/someimage.jpg] does not exist or you do not have
sufficient rights to access it.
```

This exception occurs due to the use of incorrect syntax in the URL to the image repository where resources are cached. If "repository://" is used instead of "repository:///", the resource is not cached. Because resources were not cached in LiveCycle 8.0.x, the exceptions were not seen on that system.

To resolve this issue, modify any instances of repository:// to repository:/// in forms migrated from LiveCycle 8.0.x.

Miscellaneous errors

adobe-livecycle-websphere.ear fails to deploy

If you are deploying LiveCycle ES2 components to WebSphere on a localized instance of the Windows operating system, the LiveCycle Configuration Manager deployment process reaches approximately 7% completion and then `adobe-livecycle-websphere.ear` fails to deploy.

Follow these steps to avoid this problem:

1. Use LiveCycle Configuration Manager to configure the LiveCycle ES2 EAR files, but do not deploy them.
2. Without exiting LiveCycle Configuration Manager, open the WebSphere Administrative Console and remove the following JVM argument from the server configuration:

   ```
   -Dfile.encoding=utf8
   ```

3. Restart WebSphere.
4. In LiveCycle Configuration Manager, deploy the `adobe-livecycle-websphere.ear` file.
5. When complete, return to the WebSphere Administrative Console.
6. Add the `-Dfile.encoding=utf8` JVM argument back.
7. Restart WebSphere.
8. Return to LiveCycle Configuration Manager and deploy the remaining EAR files.

Application model features are unavailable

If you select an ECM repository as the working repository on the **ECM Connectors Configuration Settings** screen in the LiveCycle Administration Console, LiveCycle ES2-specific application model features become unavailable. For example, you may not be able to create, execute, or delete LiveCycle ES2 applications.

However, you will be able to work with LiveCycle ES processes as before.

Level setting for HTML to PDF is lost

The level setting on the **Home > Services > LiveCycle PDF Generator ES2 > File Type Settings > New > HTML to PDF** screen in the LiveCycle Administration Console is lost. For example, change the level setting from **Get only 1 level** to **Get only 2 levels**. Now, click some other option (for example, **PDF optimizer**) and then expand **HTML to PDF** again. The value for the setting changes to **Get only 1 level** from what you set.

To resolve this issue, set the value for this setting again to **Get only 2 levels** (or any other desired value) immediately before clicking **Save**.

Note that if you expand the **HTML to PDF** section again, you will still see **Get only 1 level** selected instead of **Get only 2 levels**. Do not worry about this user interface discrepancy since LiveCycle ES2 has correctly stored your selected value.

If you want to confirm that your selected value has indeed been stored, do the following tasks:

1. In LiveCycle Administration Console, click **Home > Services > LiveCycle PDF Generator ES2 > Configuration Files > Export Configuration File** to export the configuration file.
2. In the exported configuration file, verify that the value of the `<levels>` tag under `<html2pdfSettings>` is correct.

English error messages instead of localized strings on some screens

LiveCycle Administration Console displays English error messages instead of the localized strings on some screens. To resolve this issue, change the JVM locale to have server-side components generate localized error messages. For example, add the following JVM argument to set the JVM locale to French:

```
-Duser.language=fr -Duser.region=FR
```

See http://java.sun.com/developer/technicalArticles/J2SE/locale for more information.

Exception in server log when removing applications

From Workbench ES2, when you remove an application which has a datatype which is referenced in a process, an exception is recorded in the server log. However, the required application is undeployed and removed successfully. You can ignore this error.

WebSphere stops responding when many concurrent threads are running

New for 9.0.0.2

When the `storeContent` operation for Content Services ES2 is invoked with more than 30 threads, the WebSphere Application Server may stop responding.

Follow these steps to resolve the issue:

1. In WebSphere Administrative Console, click **Servers > Server Types > WebSphere application servers** and then click a server name.
2. In the right pane, click **Thread pools** under Additional Properties.
3. Click **WebContainer** and, on the Configuration page, increase the value of the **Maximum Size** field by double the number of threads that you need to run. For example, increase the value of the **Maximum Size** field by 80 if you want to run 40 threads.
4. Click **Apply** or **OK**.
5. Click **Save directly to the master configuration**.
6. Restart the WebSphere Application Server.

Repeated messages in the WebSphere logs for objects created using Asset Manager

*** New for 9.5 ***

This issue is relevant for the Correspondence Management solution accelerator.

For objects created using Asset Manager, you may observe that the following message appears repeatedly in the WebSphere logs:

```
ADMLuceneInde I org.alfresco.repo.search.impl.lucene.ADMLuceneIndexerImpl
indexProperty Not indexed: Content Missing
node: workspace://SpacesStore/aeb5e45f-971f-4f31-8f2b-5ed4f2a15070 at
/{http://www.alfresco.org/model/application/1.0}company_home/{http://www.alf
resco.org/model/content/1.0}DataStore/{http://www.alfresco.org/model/content
/1.0}ObjectWithExtendedPropertiesMap/{http://www.alfresco.org/model/content/
1.0}ObjectWithExtendedPropertiesMap-1282274200933
```

Follow these steps to prevent this message from appearing:

1. (WebSphere 7.x) In WebSphere Administrative Console, click **Servers > Websphere application servers** and click the server name.

   (WebSphere 6.x) In WebSphere Administrative Console, click **Servers > Application Servers** and click the server name.

2. On the Configuration tab, under Troubleshooting, click **Change log level details.**

3. Click the Runtime tab and then expand the **All Components** list.

4. Navigate to the **org.alfresco.repo.search.impl.*** category.

5. Click the **org.alfresco.repo.search.impl.lucene.*** category and select **Message and Trace Levels > warning**.

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

Troubleshooting Error Messages

This section describes the problems and solutions associated with the LiveCycle ES2 log file error messages.

General Error Messages

This section describes error messages that are not specific to LiveCycle ES2 and how to resolve the underlying problems.

OutOfMemoryError

This type of error is typically caused by one of the following issues:

- "Running out of threads" on page 50
- "Threads and memory allocation" on page 50
- "Running the Document Management service for Content Services ES2 on basic hardware" on page 52

Running out of threads

There are many types of threads; however, essentially they fall into two categories: Java threads and native threads. All the threads running within a JVM are Java threads (`java.lang.Thread` class inside Java). The native code (C++/C) creates native threads that are scheduled and managed by the operating system. Here are the key differences between the two types:

- Java threads are created and managed by LiveCycle ES2 code, application server, or the JVM itself.
- Operating system tools (such as perfmon or Task Manager) know only about native threads.

Because the operating system has no visibility into Java threads, when you monitor threads using operating system tools such as perfmon, you are monitoring only native threads. The only way to get details into Java threads is to get a Java thread dump. The process to get a Java thread dump varies depending on your application server and JVM. See the manufacturer's documentation.

The implementation of the JVM is done in C/C++ code and that JVM code maps Java threads to native threads. This mapping can be either 1:1 (1 Java thread to 1 native thread) or N:1 (multiple Java threads to 1 native thread). The details of how this mapping works are specific to the JVM vendor, but 1:1 mapping is a typical default. This mapping means that each Java thread has a corresponding native thread. The number of Java threads has no limit; however, because 1:1 mapping is typical and the number of native threads is limited, you can run out of Java threads as well. This limit applies per process (JVM being a single process) and varies with each operating system. You can assume that the limit is in the thousands, but less than 10,000. Regardless of this number, having many hundreds of threads is a performance problem because the operating system has to schedule up to that many threads.

Threads and memory allocation

Another common issue for threads pertains to memory allocations. When a new Java thread is allocated, a fixed amount of memory is required for the thread's stack. This thread stack space is a parameter (`-Xss`

option for Sun™ JVM), and the default is ~512 KB. Therefore, if you have 1000 threads, 500 MB of memory is required just for the thread's stacks. This memory will compete with all the other memory allocations being done in the JVM, such as what LiveCycle ES2 allocates, and will create memory allocation issues.

In practice, when the JVM cannot allocate memory or create threads, it throws an `OutOfMemory` exception back to the caller. Along with this exception is a stack trace and a reason for throwing the exception. This reason is very important to note; it will give you further clues to what may be wrong.

The following code is an example of a message that displays two errors and their associated reason codes:

```
"unable to create new native thread: java.lang.OutOfMemoryError: unable to
create new native thread java.lang.OutOfMemoryError: Java heap space"
```

These errors mean that the JVM could not create more threads for one of these reasons:

- The per-process thread limit was reached.
- The thread stack cannot be allocated.

To determine the exact cause, you must get a thread dump (also known as *Java jump*). A thread dump is typically called javacore.xxxx.txt and resides under an application server's log directories. A lot of information is inside the thread dump, but you can quickly determine the number of threads by counting the occurrences of the `TID:` token on the list. A typical entry looks like this:

```
"Thread-1227" (TID:0x106948F0, sys_thread_t:0x78996DA0, state:R, native
ID:0x191C) prio=5
4XESTACKTRACE at java.net.SocketInputStream.socketRead0(Native Method)
4XESTACKTRACE at
java.net.SocketInputStream.read(SocketInputStream.java(Compiled Code))
4XESTACKTRACE at
java.io.BufferedInputStream.fill(BufferedInputStream.java(Compiled Code))
4XESTACKTRACE at
java.io.BufferedInputStream.read1(BufferedInputStream.java(Compiled Code))
4XESTACKTRACE at
java.io.BufferedInputStream.read(BufferedInputStream.java(Compiled Code))
4XESTACKTRACE at com.sun.jndi.ldap.Connection.run(Connection.java(Compiled
Code))
4XESTACKTRACE at java.lang.Thread.run(Thread.java:567)
```

If you find thousands of threads, you are probably running out of threads. Developers should be able to identify obvious culprits by scanning the stack traces of these threads.

Note: Thread dumps are typically intrusive and require that you restart the application server afterwards.

If the thread count is in the hundreds, the reason for the `java.lang.OutOfMemory` error is not the thread limit. Reduce the thread stack size (`-Xss` option mentioned above), rerun LiveCycle ES2, and see if the problem disappears.

OutOfMemoryError: Java heap space error

LiveCycle ES2 can require transactions that run for longer than the default application server transaction time-out value. For example, processing large PDF documents can be very time-intensive. These errors can appear in the application server log when LiveCycle Workbench ES2 users drag large files to the Resources view.

If you see `OutOfMemoryError` messages in the application server log, you must increase the transaction time-out value. The recommended value is 300 seconds (5 minutes). On WebLogic, the time-out value must be higher than the value configured at the Job Source through the WebLogic Server Administration

Console. On WebSphere, the time-out value must be higher than the value configured for the maximum transaction time out.

➤ **To configure the JBoss transaction time-out:**

1. Open *[appserver root]*/server/all/conf/jboss.service.xml using a text editor.
2. Locate the `attribute` element that has the `name` attribute with the value `TransactionTimeout`:

   ```
   <attribute name="TransactionTimeout">300</attribute>
   ```

3. Modify the text in the `attribute` element to be a larger number, as required.
4. Save jboss.service.xml.

➤ **To configure the WebLogic transaction time-out:**

1. Log in to the WebLogic Server Administration Console and, under Domain Structure, click **Environment** > **Servers.**
2. In the right pane, click your server, and then click the **Server Start** tab.
3. Click **Lock & Edit**.
4. In the left pane, click *`[domain name]`* and, in the right pane, click the **JTA** tab.
5. In the **Timeout Seconds** box, type `300` (or higher).
6. Click **Save** and then click **Activate Changes**.

➤ **To configure the WebSphere transaction time-out:**

1. In the WebSphere Administrative Console navigation tree, click **Servers** > **Application servers** > *[server name]*.
2. Under Container Settings, click **Container Services** > **Transaction Service**.
3. Under General Properties, in the **Total transaction lifetime timeout** box, type `300` (or higher).
4. Under General Properties, ensure that the value for **Maximum transaction timeout** is greater than or equal to the value you specified for the **Total transaction lifetime timeout** property.
5. Click **OK**.

Running the Document Management service for Content Services ES2 on basic hardware

Content Services ES2 features various in-memory caches that significantly improve performance, but consume considerable Java heap memory. You may encounter OutOfMemory exceptions if you run the the Document Management service for Content Services ES2 on hardware that only meets the minimum hardware requirements.

You can control memory usage by setting the JVM arguments `-Dhibernate.cache.use_second_level_cache=false` and `-Dhibernate.cache.use_query_cache=false`.

➤ **To control Content Services ES2 memory usage on JBoss Application Server:**

1. Open the following file in a text editor:
   - (Windows) *[appserver root]*/bin/run.bat
   - (UNIX) *[appserver root]*/bin/run.sh
2. In the JAVA_OPTS line, add or change the following arguments:
   - `-Dhibernate.cache.use_second_level_cache=false`
   - `-Dhibernate.cache.use_query_cache=false`
3. Save the edited file.

➤ **To control Content Services ES2 memory usage on WebLogic Server:**

1. In the WebLogic Server Administration Console, under Domain Structure, click **Environment** > **Servers** and, in the right pane, click the name of the LiveCycle ES2 server.
2. Click the **Configuration** tab > **Server Start**.
3. Under Change Center, click **Lock & Edit**.
4. In the Arguments box, add or change the following JVM arguments:
   - `-Dhibernate.cache.use_second_level_cache=false`
   - `-Dhibernate.cache.use_query_cache=false`
5. Click **Save** and then click **Activate Changes**.

➤ **To control Content Services ES2 memory usage on WebSphere Application Server:**

1. Log in to the WebSphere Administrative Console and, in the navigation tree, click **Servers** > **Application servers** and then, in the right pane, click the server name.
2. Under Server Infrastructure, click **Java and Process Management** > **Process Definition**.
3. Under Additional Properties, click **Java Virtual Machine** and, in the Generic JVM arguments box, add or change the following JVM arguments:
   - `-Dhibernate.cache.use_second_level_cache=false`
   - `-Dhibernate.cache.use_query_cache=false`
4. Click **Apply** and then click **Save directly to the master configuration**.

404 File not found

If you see the `404 File not found` error, perform these checks:

- Confirm the problem in the browser's access log.
- Confirm that the EAR file deployed properly and that the application initialized.
- If the URL is intended for the HTTP server, check that the file exists. Look in the error_log or error.log file for the full file name that the web server is looking for.
- (JBoss) Because it is case sensitive, ensure that the URL uses the correct case.

- (JBoss) Ensure that the web application context root (first part of the URL) exists in the uriworkermap.properties file of the JK plug-in configuration.
- (JBoss) If it is a JSP, ensure that the file exists in the EAR file. This option is confirmed by the absence of an entry in the HTTP server's error log file.

Class not found

If you see the `Class not found` error, check whether any of these problems exist:

- The class path setting is invalid or missing.
- The JAR file is obsolete.
- A compilation problem exists in the class.

JNDI name not found

If the symptom is an exception stack trace showing `javax.naming.NameNotFoundException: jdbc/<badName>`, check that the expected name is spelled correctly. If it is not, you must fix the code.

➤ **To correct most common JNDI exceptions:**

1. Check the JNDI tree on the LiveCycle ES2 application server. Does the name used appear in the tree?
   - If yes, it is most likely that your code did not properly set up the `InitialContext` object being used for the look-up, and the look-up is being done on a JNDI tree that is not the one that the resource is listed in. For the property values to use, see *Installing and Deploying LiveCycle ES2* for your application server.
   - If no, continue to step 2.
2. Does the resource appear in the JNDI tree under a name other than that listed in the look-up?
   - If yes, you are using the incorrect look-up name. Provide the correct name.
   - If no, continue to step 3.
3. Review the application server logs during startup. If the application server was configured to make this resource available but something is going wrong, an exception appears here. Is there an exception?
   - If yes, review the exception and stack trace. If the `NameNotFoundException` is a symptom of another problem based on your investigation of the server logs, go to the troubleshooting steps for that problem.
   - If no, continue to step 4.
4. If the resource is not listed in the JNDI tree, and no exception appears at startup to explain why it is not available, the most probable issue is that the application server was not configured properly to make that resource available. Review the application server configuration. Was it configured to make this resource available?
   - If no, see *Installing and Deploying LiveCycle ES2* for your application server.
   - If yes, this problem is not one of the common ones that cause this issue. Contact Adobe Enterprise Support.

JBoss Application Server error messages

org.jboss.logging.appender.FileAppender object issue

(Known issue) If ECM Connector for EMC Documentum is included in your LiveCycle ES2 for JBoss installation, the following error message appears in the server logs every time you restart the server:

```
An org.jboss.logging.appender.FileAppender object is not assignable to an
org.apache.log4j.Appender variable
```

IBM FileNet messages appear in JBoss Application Server log file

o stop unnecessary ERROR and WARNING log messages, generated by IBM FileNet, from appearing in theJBoss Application Server log file, make the following modification to the log4j.xml file located at *[jboss_root]*/server/all/conf.

1. Locate the log4j.xml file and open it in an editor.
2. Add the following text to the `[Category]` section:

```
<category name="com.filenet">
  <priority value="FATAL"/>
</category>
```

3. Save and close the file.
4. Restart the application server.

WebLogic Server error messages

WebLogic JTA time-out error

You have a WebLogic time-out issue if you receive the following error message:

```
<Warning> <com.adobe.workflow.AWS> <ap-sun4> <Server_127> <[ACTIVE]
ExecuteThread: '17' for queue: 'weblogic.kernel.Default (self-tuning)'>
<<anonymous>> <BEA1-58E59A31956BB0D8F0AB> <> <1178316054656> <000000>
<javax.ejb.TransactionRolledbackLocalException: EJB Exception: ; nested
exception is: javax.ejb.TransactionRolledbackLocalException: EJB Exception: ;
nested exception is: weblogic.transaction.internal.TimedOutException:
Transaction timed out after 299 seconds
```

To resolve this issue, increase the WebLogic JTA time-out value to a value greater than 300 seconds. (See "Configuring the WebLogic transaction time-out" in *Preparing to Install LiveCycle ES2*.

Failure to deploy adobe-livecycle-weblogic.ear

You have a WebLogic EAR file deployment issue if you receive the following error message:

```
  Could not start application adobe-livecycle-weblogic.
  com.adobe.livecycle.cdv.CDVException[ALC-LCM-030-113]: Failed to deploy
EAR.
```

To resolve this issue, check the WebLogic Server Administration Console to ensure that it is not locked, which is indicated by the Lock & Edit button being selected. If it is locked, LiveCycle Configuration Manager shows the deployment process as 16% complete and the WebLogic Server Administration Console shows the EAR file as deployed but in an installed state. If the WebLogic Server Administration Console is not locked, LiveCycle Configuration Manager can deploy the EAR files.

To resolve this issue, go to the WebLogic Server Administration Console, ensure that it is unlocked, and redeploy the EAR files.

Failure to deploy due to PermGen Space error

You have a WebLogic EAR file deployment issue if you receive the following error message:

```
java.lang.OutOfMemoryError: PermGen space
```

To resolve this issue, increase the `PermGen space` from `256` to `512`. You can change this value from the WebLogic Server Administration Console.

Failure to deploy LiveCycle ES2 modules on WebLogic/Windows

There is a known issue that WebLogic Server running on Windows fails to deploy LiveCycle ES2 modules because the server time-out setting of 5 seconds is too short. You must manually configure this setting as follows:

- Go to *[appserverdomain]* and open startWeblogic.cmd in an editor.
- Locate `[ADOBE_JAVA_HOME]` and add the following parameter:

```
-Dweblogic.client.socket.ConnectTimeout = <timeout value>
```

Note: LiveCycle Configuration Manager does not support using and admin server if a managed server is not configured.

WebSphere Application Server error messages

Failure to deploy adobe-livecycle-websphere.ear file

This section explains how to correct a failed deployment if you receive the following error message when attempting to deploy the adobe-livecycle-websphere.ear file:

```
Could not deploy adobe-livecycle-websphere.ear.
com.adobe.livecycle.cdv.CDVException[ALC-LCM-030-112]: Failed to deploy EAR.
Could not deploy adobe-livecycle-websphere.ear.
```

➤ **To correct a WebSphere failed deployment:**

1. Run the `limit -n` command in the command window.
2. If a value of `1024` is returned, increase the value to `2048` in the wasadmin.sh script.
3. Open the *[appserver root]*/bin/wsadmin.sh script in a text editor. After the file's comment block header, add the `ulimit -n 2048` line:
4. Restart WebSphere and deploy the adobe-livecycle-websphere.ear file by using LiveCycle Configuration Manager.

J2CA0294W warning messages

To avoid receiving warning messages in the SystemOut.log file that are related to the deprecated usage of direct JNDI lookup, you can modify the WebSphere logging level.

To suppress the warning message `J2CA0294W` from the SystemOut.log, you can change the logging level to `*=info:com.ibm.ejs.j2c.ConnectionFactoryBuilderImpl=severe`.

➤ **To change the logging levels:**

1. Log in to WebSphere Administrative Console through the URL http://*[hostname]*:9060/admin and, in the navigation tree, click **Troubleshooting** > **Logs and Trace**.
2. In the right pane, click the name of the application server and then click **Change Log Detail Levels**.
3. Click the Configuration tab and type the following string:

   ```
   *=info:com.ibm.ejs.j2c.ConnectionFactoryBuilderImpl=severe
   ```

4. Click **OK** and the click **Save directly to the master configuration**.

Verbose log messages in WebSphere installation

To avoid the WebSphere installation from logging several unnecessary log messages, you can increase the logging level to "Warning" so that messages at lower level are not logged.

➤ **To change the logging levels:**

1. Log in to WebSphere Administrative Console through the URL http://*[hostname]*:9060/admin and
2. In the navigation tree, click **Troubleshooting** and select **Logs and Trace**.
3. In the right pane, click the name of the application server and then click **Change Log Detail Levels**.
4. Select **Runtime** and enter `org.apache.xml.security.*`
5. Click **Message And Trace Levels**, and select **Warning**.
6. Select **Save runtime changes to configuration** check box.
7. Click **OK**.

Adobe-livecycle-websphere.ear file not deploying to 64-bit WebSphere 6.1

When deploying the adobe-livecycle-websphere.ear file manually or by using LiveCycle Configuration Manager to a 64-bit WebSphere Application Server, the following error message may appear:

```
Error executing deployment: java.lang.IllegalStateException. Error is Unable
to acquire application service. Ensure that the org.eclipse.core.runtime
bundle is resolved and started (see config.ini)..
java.lang.IllegalStateException: Unable to acquire application service. Ensure
that the org.eclipse.core.runtime bundle is resolved and started (see
config.ini).
at
org.eclipse.core.runtime.internal.adaptor.EclipseAppLauncher.start(EclipseAp
pLauncher.java:65)
```

If this error occurs in the WebSphere Administrative Console, it is recommended that you update your application server to WebSphere 6.1 Fixpack 15 and SR6.

Unnecessary SystemOut messages during LiveCycle ES2 module deployment on WebSphere 6.1

There is a known issue that when deploying the LiveCycle ES2 modules on WebSphere Application Server, unnecessary SystemOut messages are written to the system.out log file, such as these messages:

```
   [2/26/08 19:16:34:303 CST] 00000022 SystemOut
Oflex.management.runtime.messaging.services.remoting.RemotingDestinationCont
rol
[2/26/08 19:16:34:304 CST] 00000022 SystemOut     O 9
[2/26/08 19:16:34:305 CST] 00000022 SystemOut     O
flex.management.runtime.messaging.services.remoting.adapters.JavaAdapterCont
rol
```

IBM provides a patch in their Available Fixes list to resolve this issue.

Exception: No trusted certificate found

Your WebSphere Application Server may give exceptions similar to the ones described below.

Exceptions seen from LiveCycle Administration Console:

```
 Could not connect to Inbox. Error message: com.ibm.jsse2.util.h:
   No trusted certificate found; nested exception is:
      javax.net.ssl.SSLHandshakeException:
com.ibm.jsse2.util.h: No trusted certificate found
```

Exceptions seen in WebSphere Application Server log files:

```
[5/28/08 13:15:30:283 CDT] 00000025 SystemOut     O
CWPKI0022E: SSL HANDSHAKE FAILURE:  A signer with SubjectDN
"CN=imap.gmail.com, O=Google Inc, L=Mountain View, ST=California, C=US"
was sent from target host:port "null:null". The signer may need to be
added to local trust store "D:/servers/websphere6.1/profiles/AppSrv01
/config/cells/MN-TOBIKONode01Cell/nodes/MN-TOBIKONode01/trust.p12"
located in SSL configuration alias "NodeDefaultSSLSettings" loaded from
SSL configuration file "security.xml".  The extended error message from
the SSL handshake exception is: "No trusted certificate found".
[5/28/08 13:15:30:283 CDT] 00000025 SystemOut     O
[5/28/08 13:15:30:283 CDT] 00000025 ExceptionUtil E
CNTR0020E: EJB threw an unexpected (non-declared) exception during
invocation of method "doSupports" on bean "BeanId(adobe-core-websphere
#adobe-dscf.jar#EjbTransactionCMTAdapter, null)". Exception data:
java.lang.RuntimeException: Could not connect to Inbox. Error message:
com.ibm.jsse2.util.h: No trusted certificate found;
   nested exception is:
      javax.net.ssl.SSLHandshakeException:
com.ibm.jsse2.util.h: No trusted certificate found
```

This problem arises when the WebSphere key store does not contain a required certificate. Note that the default WebSphere key store contains only a limited set of certificates. Use the following procedure to add a new certificate to the WebSphere key store.

➤ **To add a new certificate to the WebSphere key store:**

1. Obtain the appropriate certificate from the email service.
2. Copy the certificate to *[appserver root]*\profiles*[server name]*\etc.. . .
3. Log in to the WebSphere Administrative Console and click **Security** > **SSL certificate and key management**.
4. Under Related Items, click **Key stores and certificates**, and then click **CellDefaultTrustStore**.
5. Under Additional Properties, click **Signer certificates**, and then click **Add**.
6. In the **Alias** box, type an appropriate alias for the certificate you are importing.
7. In the **File name** box, type the location where you installed the certificate at step 2, and then click **OK**.
8. Click **Save directly to the master configuration**. The certificate you just added should now be listed as a Signer certificate.
9. Restart the WebSphere Application Server.

Java NameNotFoundException exception

While bootstrapping User Manager components on WebSphere Application Server, the following exception message will appear only once after the application is started:

```
00000043 javaURLContex E   NMSV0310E: A JNDI operation on a "java:" name cannot
be completed because the server runtime is not able to associate the
operation's thread with any J2EE application component. This condition can
occur when the JNDI client using the "java:" name is not executed on the thread
of a server application request. Make sure that a J2EE application does not
execute JNDI operations on "java:" names within static code blocks or in
threads created by that J2EE application. Such code does not necessarily run on
the thread of a server application request and therefore is not supported by
JNDI operations on "java:" names. Exception stack trace:
javax.naming.ConfigurationException [Root exception is
javax.naming.NameNotFoundException: Name comp/env/ejb not found in context
"java:".]
```

This error can be safely ignored.

Unexpected exception during DSC invocation

When a DSC is invoked from within a transaction started by another application deployed as an EAR on the same instance of WebSphere 7.x on which LiveCycle ES2 is deployed, the DSC call fails with the following error message:

```
LocalException E   CNTR0020E: EJB threw an unexpected (non-declared) exception
during invocation of method "getObjectType" on bean
"BeanId(LiveCycleES2#adobe-pof.jar#adobe_POFDataDictionaryLocalEJB, null)".
Exception data: com.ibm.websphere.csi.CSIException: Global tx resume failed;
```

This error is encountered for WebSphere 7.x only when the `adobe-utilities.jar` file is used and `Platform.UTIL.getTransactionManager()` is the user that starts the transaction manager.

To resolve this issue, do not use `adobe-utilities.jar` to start the transaction manager. Rather, use the following code to create the UserTransaction:

```
InitialContext initialContext = new InitialContext();
UserTransaction ut =
(UserTransaction)initialContext.lookup("java:comp/UserTransaction");
ut.begin();
```

Database Error Messages

This section describes error messages relating to your LiveCycle ES2 database.

Package JDBC modules into LiveCycle ES2 EARs screen freezes

On the Package JDBC modules into LiveCycle ES2 EARs screen, if your **DB Type** is ***oracleRAC*** the screen freezes and only the **Next**, **Back**, and **Exit** buttons stay enabled. This also means that the option to select any other **DB Type** is also disabled. Using the Back button traverse to Configure Datasource JDBC Driver Classpath screen and you can see that it has frozen too. This is a known issue and can be resolved using any of the two workarounds listed below:

➤ **Method 1**

1. Exit from your LiveCycle Configuration Manager.
2. Navigate to *[LiveCycleES2 root]/* configurationManager/config folder.
3. Open the **userValues.properties** file and make the following changes:
   - Change the **CDVDataSource.dbType** value to any other DB Type.
     For example: oracle, sqlserver, etc
   - Set **CDVDataSource.manualconfiguration** value to **False**.
4. Save the file and relaunch LiveCycle Configuration Manager.

➤ **Method 2**

1. Use the **Back** button to navigate to the Task Selection screen and deselect the **Package JDBC modules into LiveCycle ES2 EARs** option.

2. Use the **Next** button to navigate to the Application Server Configuration screen, select **Globally-scoped Datasources** option.

3. Use the **Next** button to navigate to the Datasource Configuration screen, deselect the **Manual Configuration** option.

4. Now, navigate back to the Task Selection screen and reselect **Package JDBC modules into LiveCycle ES2 EARs** option.

Exceptions thrown when initializing the database multiple times

When you initialize the LiveCycle ES2 database after it is already initialized, exceptions may be thrown indicating that the POF schema was initialized.

This error can be safely ignored.

MySQL "Too many connections" error

Under heavy load conditions, MySQL may generate "Too many connections" errors. This problem can be alleviated by increasing the number of parallel connections allowed by MySQL.

1. Go to the MySQL install directory.
2. Locate the my.ini file and open it in an editor.
3. Locate the `max_connections` parameter and set it to the required value. The default value is `100`.
4. Restart MySQL.

J2CA0081E: Method cleanup error on SQL Server

If you see the error "J2CA0081E: Method cleanup failed while trying to execute method cleanup...", the stack trace appears as follows:

```
"00000057 MCWrapper E J2CA0081E: Method cleanup failed while trying to execute
method cleanup on ManagedConnection WSRdbManagedConnectionImpl@6d826d82 from
resource SP_DS. Caught exception: com.ibm.ws.exception.WsException:
DSRA0080E: An exception was received by the Data Store Adapter. See original
exception message: This operation is not supported.. with SQL State : null SQL
Code : 0"
```

You must update to SQL Server 2005 JDBC driver version 1.2. Obtain this driver from the Microsoft Download Center.

Database initialization fails on WebLogic

You may receive the following exception while initializing the LiveCycle ES2 database on WebLogic Server:

```
Servlet failed with Exception java.lang.ClassCastException:
weblogic.jndi.factories.java.ReadOnlyContextWrapper at
com.adobe.livecycle.bootstrap.framework.BootstrapServlet.checkDatasource(Boo
tstrapServlet.java:216)
```

To resolve this problem, modify the IDP_DS-3079-jdbc.xml file.

➤ **To modify the IDP_DS-3079-jdbc.xml file:**

1. Locate the IDP_DS-3079-jdbc.xml file in the *[appserverdomain]*/config/jdbc directory and open it in an editor.
2. Locate the line `<jndi-name></jndi-name>` and delete it.
3. Save and close the file, and reinitialize the database.

Memory issues on DB2 server

DB2 configuration page displays the following error message:

```
SQL1585N A temporary table space with sufficient page size does not exist.
```

```
Following is the SQLCODE and SQLSTATE value specific to memory issue on DB2
server.
sqlcode: -1584
sqlstate: 57055
SQL1585N A temporary table space with sufficient page size does not exist.
```

For information on this error, see the IBM knowledge base article at http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp?topic=/com.ibm.db2.luw.messages.sql.doc/doc/msql01585n.html.

To avoid this issue, do the following:

- Increase the system RAM. See the Preparing to Install guide for recommendations.
- Provide at least 2 GB of space to each DB2 instance. If you are installing LiveCycle Content Services ES2 for use with a DB2 database, you must have a minimum of 2GB of RAM on the computer that hosts the LiveCycle ES2 database.

LiveCycle ES2 Error Messages

This section describes error messages relating to your LiveCycle ES2 installation.

Output error messages

Converting native files fails on JBoss/Windows

PDF Generator ES2 or PDF Generator 3D ES2 running in a Windows operating system may fail when converting native files, such as from Microsoft Word to PDF, giving an exception in the log file similar to this:

```
INFO [PDF Generator] Application server started as user: SYSTEM
```

This problem occurs if you did not use the same user account for Microsoft Office, PDF Generator ES2, Acrobat for PDF Generator ES2, and your application server process.

To correct this problem, change the user for the JBoss for Adobe LiveCycle service to be the same as the user for Microsoft Office.

➤ **To change the user for the JBoss for Adobe LiveCycle service:**

1. Select **Start** > **Control Panel** > **Administrative Tools** > **Computer Management** > **Services and Applications** > **Services**.
2. Double click the **JBoss for Adobe LiveCycle** service, and then select the **Log On** tab.
3. Select **This account**, type the user name and password that runs Microsoft Office, and then click **OK**.

Converting HTML files fails on JBoss/Solaris

PDF Generator ES2 running on JBoss in a Solaris operating system may fail when converting HTML files using the HTMLToPDF or HTMLURLToPDF operations, giving an exception in the log file similar to this:

```
com.adobe.livecycle.generatepdf.client.ConversionException:
ALC-PDG-052-016-The conversion has timed out.
at com.adobe.pdfg.GeneratePDFImpl.htmlURLToPdf
```

This problem occurs when your system is missing the libiconv.so.2 library.

To correct this problem, install the libiconv.so.2 library in the /usr/lib directory.

XMLForm.exe permission error on UNIX or Linux

A UNIX and Linux permission issue exists with Forms ES2 if the following error message is logged:

```
Cannot add execute permission on file /[path_to_XMLForm.exe]
```

To resolve this issue, ensure that the swap space on UNIX and Linux servers is at least 3 GB.

Unexpected end of file error on WebLogic

You have an issue with LiveCycle Forms ES2 and LiveCycle Output ES2 if you see the following error message:

```
com.adobe.idp.DocumentError: java.net.SocketException: Unexpected end of
file from server at
com.adobe.idp.Document.passivateInitData(Document.java:867)
```

To resolve this issue, ensure that the Document `MaxInlineSize` parameter is set to a value that is smaller than the IIOP message size parameter defined in WebLogic Server.

Client-side Error Messages

CORBA COMM_FAILURE exception on WebLogic/Solaris

If you encounter a CORBA COMM_FAILURE exception from your client while running WebLogic Server on Solaris, you must pass the following additional property to the client-side JVM:

```
-Dcom.sun.CORBA.transport.ORBTCPReadTimeouts=1:60000:300:1
```

The value in bold is a colon-delimited list of the time-out values in milliseconds that should be set according to your system requirements. The values from left to right are as follows:

- Initial wait time if a transport TCP read returns 0 bytes
- Maximum cumulative wait time if a transport TCP read returns 0 bytes
- Maximum cumulative wait time if a transport TCP read of a GIOP header returns 0 bytes
- Backoff percentage used to compute the amount of time to wait on a subsequent transport TCP read of 0 bytes

No settings are required on the server side.

Miscellaneous Error Messages

Locator is already running

If you use TCP for caching, you may encounter the following exception while starting the TCP locator:

```
Exception in thread "main" com.gemstone.gemfire.SystemIsRunningException:
Locator "/usr/prod/lbs/domains/lbsdomain/idplib/caching" is already running.

   at com.gemstone.gemfire.internal.ManagerInfo.setManagerStarting
   (ManagerInfo.java:65)
   at com.gemstone.gemfire.internal.ManagerInfo.setLocatorStarting
   (ManagerInfo.java:61)
   at com.gemstone.gemfire.internal.DistributionLocator.main
   (DistributionLocator.java:87)
   at com.gemstone.gemfire.distributed.Locator.main(Locator.java:359)

[info 2009/08/26 16:18:29.770 PDT <Thread-0> nid=0x15d56d5] Locator stopped
```

To resolve this issue, check for the existence of the `.locator` lock file in the `caching` folder and delete the file.

com.adobe.idp.DocumentError

If some cluster nodes are unable to access the global document storage (GDS) directory, you may encounter the following error:

```
com.adobe.idp.DocumentError: The document pointing to the file
"B:\lc9_share\docm1245494450089\c5c0f6c0a324c0d45396deb69b6e11db" has
expired. Consider increasing the document disposal timeout
```

Ensure that:

- All nodes in the cluster are time-synchronized
- All nodes have access to the GDS

If you still encounter this issue, check for network and latency problems.

IDPSchedulerService is not in a RUNNING state

`IDPSchedulerService` is not restarted although Workbench ES2 indicates that it is in the started state. The following exception appears in the server logs:

```
ALC-DSC-020-000: com.adobe.idp.dsc.InvalidStateException: Service:
IDPSchedulerService is not in a RUNNING state
```

To resolve this issue, start `IDPSchedulerService` explicitly when you need to restart Scheduler from within Workbench ES2.

"Failed to delete a directory" warning during PDF conversions

The following warning may appear in the application server logs during conversions to PDF:

```
WARN [Document] DOCS007: Failed to delete a directory
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\AdobeDocumentStorage\local\removeOn2006Y
08M31D18h16m15s.1157028375000" after 60 attempts. The file(s) under this
directory may still be locked."
```

If this message appears consistently in the logs, restart the application server.