Adobe LiveCycle

Article

Overview of Automated LC ES Deployment-Mechanisms for Large Scale Deployments

Adobe Systems GmbH
Adobe Consulting
Frankfurter Str. 181a
63263 Neu-Isenburg

AUTOR

Thomas Wenzel

twenzel@adobe.com

VERSION

1.0

Table of Contents

1. LiveCycle ES – forms, documents and business process management

Using the service oriented Adobe LiveCycle ES suite of solutions, enterprises can create and deploy services and solutions which allow creating forms and customer facing documents, automate complex business processes by using state of the art design tools and methods or offer document conversion services such as automated office document to PDF/A conversion. These service and processes can involve employees, partners, customers, etc… who complete forms, digitally sign documents, submit completed forms back into the process or submit incoming documents to be converted.

All participants can communicate with the system using the ubiquitous Adobe clients that are available not on nearly every PC, but also on a large number of handheld and mobile devices: Adobe Reader and Adobe Flash Player.

In Germany, one of Europe's biggest computing centres is currently running a farm of independent Adobe LiveCycle 7.2.2 servers providing the services to generate customer facing documents and reader enable PDF forms. The system today serves about 400000 users providing more than 70000 documents per hour. In the coming month, the customer wants to extend the services provided by the system to also allow users to convert existing documents into PDF/A (PDF Generator). On the other hand, the computing centre wants to migrate all the current existing applications from IBM WebSpehre 6.02 to IBM WebSphere 6.1.

As a consequence of this, we have to develop a smooth mechanism to migrate our application from Adobe LiveCycle 7.2.2 to Adobe LiveCycle ES.

Unfortunately, Adobe LiveCycle ES requires additional things to set-up in order to run, such as an own database, JMS messaging, etc…

2. Migrating from LC 7.2.2 to LC ES

Today, Adobe LiveCycle 7.2.2 is used as a core component in the Document Management Solution for generating and processing of PDF forms, documents, mass mailings, archival documents, mass printing jobs, etc…

It is currently in production on AIX 5.3 running on IBM WebSphere 6.02. A migration to WebSphere 6.1 is planned.

It is required that the Adobe solution components of the DMS solution will be migrated to Adobe LiveCycle ES (8.2) to also provide more services such as the PDF/A generation and conversion services. Today, only TIFF documents are archived and the TIFF generation process if error prone and already caused a lot of problems. As a consequence of this, a requirement to move to PDF/A as the archival document standard was submitted to the DMS team.

Today, we deploy and use Adobe LiveCycle 7.2.2 Forms and Reader Extensions Server components. These components are completely re-packaged. The final EAR files to be deployed do include all assets such as fonts, server dependant configuration settings, etc… Only one shared library is needed for running LC 7.2.2. This shared library is distributed and configured by the production server team during the IBM WebSphere deployment. As soon as the application server is configured and running, the DMS EAR files (including the LC 7.2.2 components) are automatically deployed to a farm of more than 60 individual servers.

Adobe LiveCycle ES also requires an application server to run – like the previous versions of LiveCycle. Unfortunately, additional components and configurations are required such as a mandatory system database for LiveCycle ES to allow it to save configuration settings, jobs, DSC components, etc…

A requirement for a system database also existed for Adobe LiveCycle 7.2.2, but was only really required if the web based user interface of Reader Extensions or other services such as PDF/G were utilized. Since our applications did only use the APIs of LiveCyclc Forms and Reader Extensions, we were able to run without a dedicated LiveCycle ES database.

So when migrating from LC 7.2.2 to LC ES, the requirement for a system database is no longer optional, but mandatory.

The new solution components and application architecture of LiveCyclc ES also use JMS messaging provided by IBM WebSphere. JMS queues and messages are usually managed and stored by IBM WebSphere using a set of database tables which are usually created in the LC ES system database.

LC ES can be installed in an IBM WebSphere clustered environment allowing the individual LC ES instances to communicate with each other to share load or take over jobs when an instance fails.

However, this configuration may not be beneficial for our application, because the application itself runs on "groups" of two servers with a hardware load-balancer supporting several thousand users. Each server of a group of two servers works completely independent and the application is completely stateless, so the load-balancer forwards requests to any server of a group. All servers of a group use the very same application database to store templates, documents, converted files, jobs, etc… So a request to render 10 forms for one business process request can effectively distributed among all member servers of a group.

LC ES supports clustered environments, but it may be easier from a configuration, deployment and production perspective to install the individual LC ES servers as "little islands" just as described above.

Each LC ES island is an independent instance running the Adobe LiveCyclc ES components and the custom DMS application. Unfortunately, each "island" will require its own system database.

3. Pitfalls and Solutions: Deployment of Adobe LiveCycle ES

The following chapters will describe the new requirements for deploying and running Adobe LiveCycle ES. If issues are encountered where the deployment and production processes compared to the previous version of Adobe LiveCycle (7.2.2) need to be changed, the issues and potential solutions will be discussed in detail.

3.1. Software-Requirements (OS, APPServer, DB)

The document „Preparing to Install LiveCycle® ES (Single Server)“ lists the following software requirements to run Adobe LC ES on AIX / WebSphere:

IBM AIX
LiveCycle ES supports the following IBM AIX operating system:

AIX 5L 5.3 (64-bit and 32-bit architectures)
Note: It is recommended using the 64-bit architecture for AIX even when using the 32 bit JDK.
This table lists the supported application server, JDK, and database configurations for this operating System.

3.1.1. Supported Combinations of OS+APPServer+DB

Pure 64 bit system on AIX 5L 5.3:

Component	
Operating System	AIX 5L 5.3
Application Server	IBM WebSphere 6.1.0.7 (Base & Network Deployment Edition) JDK 1.5 SR4 available for Download from IBM 64-bit OS and 64-bit JVM
System Database	IBM DB2 8.2, 9.1 or Oracle 9i, 10g

Or 64 / 32 bit mixed configuration:

Component	
Operating System	AIX 5L 5.3
Application Server	IBM WebSphere 6.1.0.7 (Base & Network Deployment Edition) JDK 1.5 SR4 available for Download from IBM 64-bit OS and 32-bit JVM
System Database	IBM DB2 8.2, 9.1 or Oracle 9i, 10g

3.1.2. Hardware-Requirements

The minimal hardware-requirements to run Adobe LC ES on AIX are:
IBM AIX 5L 5.3 P4 pSeries 615 (Model 6C3) 7029-6C3, 1.2 GHz processor
RAM:

- 3GB (64-bit OS with 64 bit JVM)
- 2GB (64-bit OS with 32 bit JVM)

Free disk space: 5.4 GB of temporary space plus 3.4 GB for LiveCycle ES

3.1.3. Additional requirements for AIX / UNIX Systems

According to the product documentation, the Adobe LiveCycle ES Installer requires the bc tool, which is usually available in AIX / UNIX environments:

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

3.2. Pitfall: Requirements for the System-Datenbase

For the desired configuration running LC ES on AIX 5.3 and IBM WebSphere 6.1.x, Oracle 9i or 10g Datenbank can be perfectly used – including the Standard- and Enterprise editions.

3.2.1. Database Driver

The following driver must be used for deployment and production:
Oracle 9i and 10g: ojdbc14.jar, Release 2 (10.2.0.2 Thin or later update of 10.2)

3.2.2. Required Database Privileges for Deployment and Configuration

An Adobe LiveCycle ES installation is not only a pure installation or deployment of EAR files, but includes a phase called „Bootstrapping“. This phase is discussed later on in detail. For this initialization phase of the system database, Adobe LiveCycle ES required additional database privileges. During this phase, the database structure is created and the database is populated with configuration parameters and solution components and services.

For an Oracle based systemdatabase, the following rights are required:

CREATE SESSION
CREATE CLUSTER
CREATE TABLE
CREATE VIEW
CREATE SEQUENCE
UNLIMITED TABLE SPACE

(according to “Preparing to Install LiveCycle® ES (Single Server)”.)

3.2.3. Required Database Privileges for Production

After completing the initialization (Bootstrapping) phase, Adobe LiveCycle ES requires less database privileges to access the system database:

CREATE SESSION
CREATE CLUSTER
UNLIMITED TABLE SPACE

(according to “Preparing to Install LiveCycle® ES (Single Server)”.)

3.2.4. Creation of an Oracle-based Adobe LC ES Systemdatabase

Create a new database by using the Database Configuration Assistant tool. LiveCycle ES supports Oracle 9i and Oracle 10g with WebSphere 6.1.0.7.

These requirements are for creating your Oracle database:

- The initial database size must be a minimum of 1175 MB [1](You must increase this initial size if you are deploying LiveCycle ES Content Services).
- The database must support UTF-8 characters.
- The Database Character Set is Unicode (AL32UTF8), and the National Character Set is AL16UTF16
- (Unicode UTF-16 universal character set).
- The database setting NLS_LENGTH_SEMANTICS is set to BYTE. The database initialization fails if you set any other value.
- You must install Oracle using Transaction Processing and set the connection mode for the server to dedicated Processing.

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

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

[1] In discussions with friends from R&D in Canada, we’ve leared that the minial database size of 1175 MB (which is 1.2 GB) may be a bit over-sized and really depends on the services used. For the services used by our application and for the known workflows we support today, a database size of about 100 MB is expected by R&D. We will verify this as soon as possible in the customers environment.

- CREATE SESSION

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

3.3. Pitfall: Customer Specific Fonts

During the packaging of the Adobe LiveCycle 7.2.2 solution components, all required fonts including Adobe standard fonts and additional fonts provided by the customer, were packaged in the EAR files generated by the configuration procedure.

As a consequence of this we've created a self-containing application archive which can easily deployed to our 60 servers.

Unfortunately with Adobe LiveCycle ES (8.x) there is a change in product behaviour: According to the documentation, fonts are not longer contained in the packaged EAR files, but need to be distributed as „file sets" onto the target server environment.

As a consequence of this, the fonts need to either be manually or automatically copied or distributed to a folder on each target app server of copied onto a network share (high performance and high avail – such as a SAN powered storage) accessible to all serves.

The standard Adobe LiveCycle ES installation procedure copies the fonts into a folder of the target app server.

In order to support the customers' existing fully automated deployment and production procedures, one of the following two options should be implemented:

- Implementation of a mechanism to package and include all the required fonts in an EAR file. When the application starts, the fonts could be extracted to a folder specified in the Adobe LiveCycle ES configuration. On IBM WebSphere this font-EAR extraction could be implemented using a special deployment descriptor or by implementing a small java function which is invoked upon start of the web application, i.e. in a servlet's onInit method. We must take care of the fonts folder which LC ES uses. It may be a fixed folder or the mechanism also needs to change the instance's configuration setting.

- Define where the fonts are to be provided on the target app server. Manually copy and distribute the fonts onto all >60 servers and configure the Adobe LiveCycle EAR files accordingly.

3.4. Pitfall: Global document storage directory

Unfortunately, Adobe LiveCycle ES requires a special folder called „Global document storage directory“ to store solution components, long-lived process information and documents:

The global document storage (GDS) directory is used to store long-lived files that are used within a process as well as critical LiveCycle ES product components. The lifetime of long-lived files is intended to span multiple restarts of a LiveCycle ES system, and can span for days and even years. These files may include PDF files, policies, or form templates. Long-lived files are a critical part of the overall state of many LiveCycle ES deployments. If some or all long-lived documents are lost or corrupted, the LiveCycle ES Server may become unstable. Input documents for asynchronous job invocation are also stored in the GDS Directory and must be available in order to process requests. Therefore, it is important that you consider the reliability of the file system that hosts the GDS directory.

Unfortunately, the GDS is used as a kind of extension to the system database and the combination of the two (database and GDS) need to be in sync and backed-up in sync.

3.4.1. Location of the global document storage directory

One can configure the location of your GDS directory with LiveCycle Configuration Manager after installing LiveCycle ES. The GDS directory specifyed should be highly available and should have low access time to enhance performance.

When running Adobe LC ES clusters, it is advised to place the GDS on a SAN powered storage system, because this kind of storage offers ultimate performance and high availability.

For small (individual or single) Adobe LC ES installations, it is possible to place the GDS on the app server itself. This also is the product default behavior:

If one leaves the location setting empty during installation, the location defaults to a directory under the Application server installation:

- (WebSphere) appserver root/installedApps/adobe/<server>/DocumentStorage”

Usually the production team of a big enterprise customer will not be happy with this method to create the GDS, but will want to specify the exact location. It will then be assured that this GDS folder is secure, correctly sized, highly available and highly performing. Also in a WebSphere ND environment, the appserver root/installed Apps/adobe folder may be the absolutely wrong place to create files, because the appserver root folder used by our LC components may point to the ND installation, but not to the server instance and as a consequence of this, the folder may not even be accessible from the server instance.

3.4.2. Size of the global document storage directory

The size of the shared directory depends on expected Adobe LiveCycle ES usage factors for the deployment. According to the documentation, one should allocate a minimum of 10 GB of disk space for the GDS directory. However, the following factors affect the sizing:

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

It seems that the GDS is primarily used by long-lives processes which are not completed in memory. The services used by DMS today are short-lived, but only a test of the application can show how the GDS is used and a load test may unveil what sizing factors must be taken into account.

3.4.3. Backup of the global document storage directory

The global document storage directory should be backed up to allow administrators to restore an LiveCycle ES system in case of failure.
(See Administering LiveCycle ES.)
If the GDS directory becomes unavailable or is lost due to a failure, Adobe LiveCycle ES will not run until the GDS and database are restored by a consistent back up or Adobe LiveCycle ES is reinitialized.”

3.5. Adobe LiveCycle ES Deployment on IBM WebSphere 6.1

The deployment and configuration mechanisms of Adobe LiveCyclc ES on IBM WebSphere 6.1 is explained in detail in the „Installing and Deploying LiveCycle® ES For WebSphere®“ document.

Of major importance are the requirements to be provided/fulfilled by the application server for the phases of deployment, bringing into service and production.

After installing the software, the product support manual and automated mechanisms for the configuration, deployment and initialization (bootstrapping) steps.

3.5.1. Installation and Configuration

Beginning with Forms and Reader Extensions Release 6.0 until 7.2.2, we’ve used a Finanz Informatik AIX development server to install, configure and repackage the Adobe software.

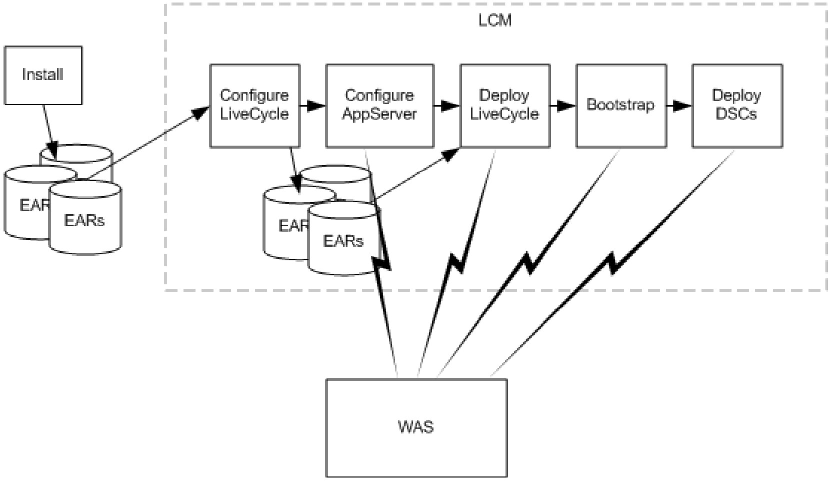
This was required to allow us to comply to the various rules of the Finanz Informatik automated deployment mechanisms.

This approach is also suggested when installing and configuring Adobe LiveCycle ES for the first time for Finanz Informatik.

However, the current version of Adobe LiveCycle ES allows to install the software on a Windows System to carry out the configuration and prepare the deployment to the AIX systems.

3.5.2. LiveCycle Configuration Manager

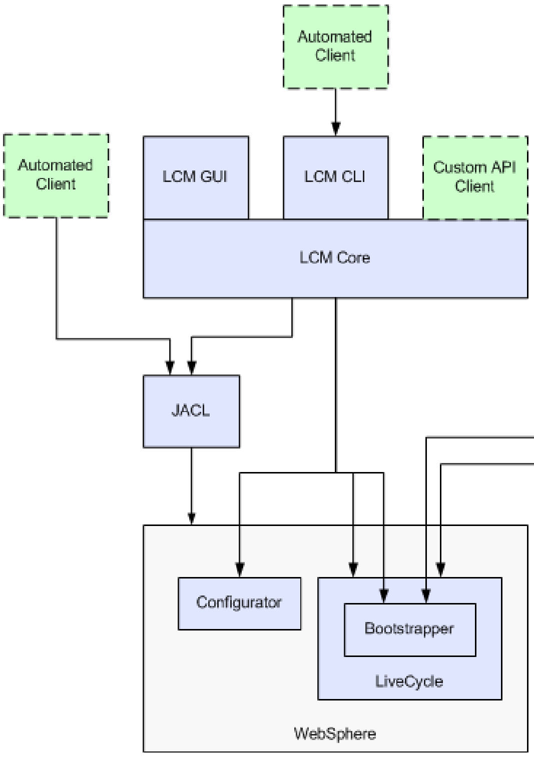
Beside the solution components like Forms, Reader Extensions, Output, PDF/G, additional tools such as the configuration manager are installed which guide the admin through the configuration process:



The Configuration Manager not only packages the EAR files based on the configuration settings given by the administrator, but also addresses a lot of work of the deployment and initialization of Adobe LiveCycle ES. There is a GUI based and a CLI (Command Line Interface) based version of the Configuration Manager available, the latter one should provide mechanisms to automate the configuration and deployment of multiple installations using Property Files.

There are several ways to invoke the functions which initialize and bootstrap the database (bootstrapper) and which do deploy the Adobe LiveCycle ES solution components such as DSC and LCA components.

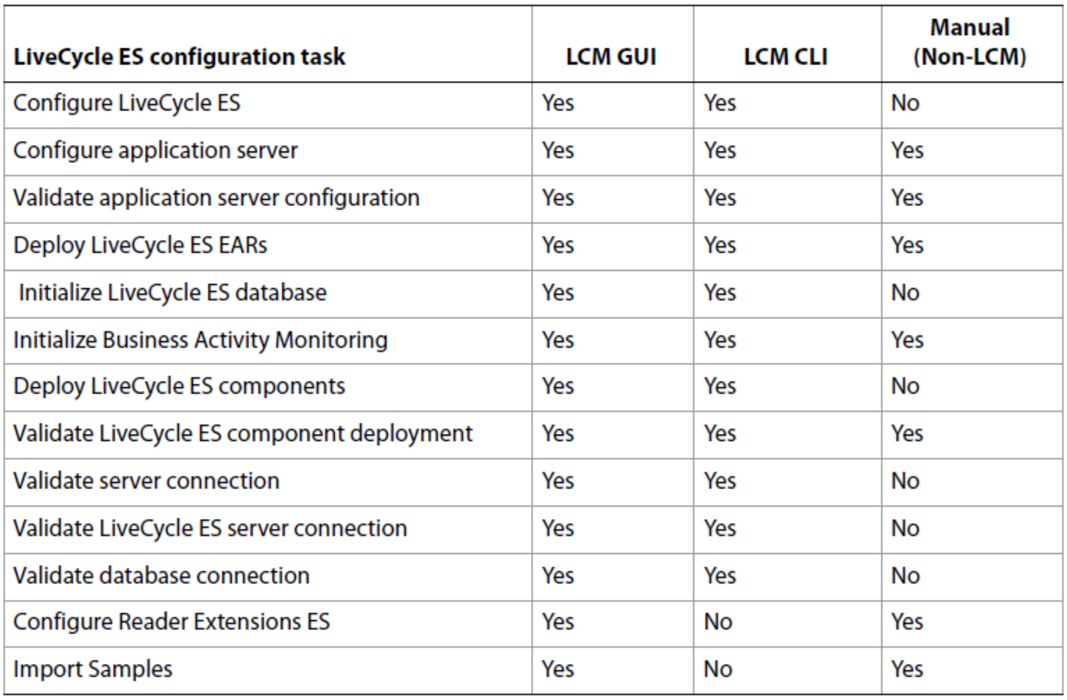
Normally, in a big enterprise environment with many servers, one will configure the EAR files using the configuration manager and will then deploy the pre-configured EAR files to the AppServers. Then after the EAR files have been deployed, the bootstrapping and deployment of the DSC and LCA components will be carried out, usually be using the LCM – livecycle configuration manager:



LiveCycle Configuration Manager offers a graphical user interface (LCM GUI) and a command-line based interface (LCM CLI). The graphical user interface is usually used to configure the EAR files, but it also supports the whole deployment process. However an administrator will normally not tolerate that a 3rd party tool such as LCM is used to configure the WebSphere AppServer settings. At least the admin wants to see what exactly is configured. In our scenario, the admins would like to manually configure their AppServers and expect that we provide the EAR files to deploy on their configured servers. Settings such as JVM arguments, heap size, timeouts, datasources, JMS configuration, etc… are things they want to control and they will not allow a 3rd party tool to modify these settings.

The LCM CLI tool provides nearly the very same functions as the GUI based tool, but due to its nature it can be used to automate the deployment and configuration when there is a need to configure or deploy more than one server or LC ES installation.

The following table lists the steps which need to be carried out during a typical Adobe LiveCycle ES installation. It also show which steps/tasks can be fulfilled using LCM GUI, CLI or by manually configuring the appserver:



LiveCycle ES configuration task	LCM GUI	LCM CLI	Manual (Non-LCM)
Configure LiveCycle ES	Yes	Yes	No
Configure application server	Yes	Yes	Yes
Validate application server configuration	Yes	Yes	Yes
Deploy LiveCycle ES EARs	Yes	Yes	Yes
Initialize LiveCycle ES database	Yes	Yes	No
Initialize Business Activity Monitoring	Yes	Yes	Yes
Deploy LiveCycle ES components	Yes	Yes	No
Validate LiveCycle ES component deployment	Yes	Yes	Yes
Validate server connection	Yes	Yes	No
Validate LiveCycle ES server connection	Yes	Yes	No
Validate database connection	Yes	Yes	No
Configure Reader Extensions ES	Yes	No	Yes
Import Samples	Yes	No	Yes

If the deployment must be mainly carried out „manually“ on a WebSphere application server, the following tasks should be carried out after configuring the EAR files using LCM CLI or LCM GUI:
The tasks are described in detail in the document „Installing and Deploying LiveCycle® ES For WebSphere®“:

- “Setting directory permissions” on page 27 (described below as “Native Files – adobeidp.RootDirectory”)
- “Configuring the JVM arguments” on page 28
- “Updating ejbdeploy.sh for UNIX/Linux” on page 29 (described below as “Deployment: Increase MaxPermSize / heap size”)
- “Configuring WebSphere time-out settings” on page 29 (described below as “Deployment: Increase SOAP Timeout”)
- “Configuring J2C authentication for JMS” on page 46
- “Configuring throttling for LiveCycle PDF Generator ES” on page 30
- “Configuring the LiveCycle ES database connectivity” on page 31
- “Configuring JMS resources for WebSphere” on page 41

After these tasks have been carried out, the database could be initialized and bootstrapped using LCM to create a working Adobe LiveCycle ES system.

3.5.3. Using LCM

LCM can be used to configure a locally running WebSphere instance, but can also be used to configure, bootstrap and deploy Adobe LiveCycle ES to remote WebSphere application servers.

To configure a remote WebSphere application server, LCM utilizes a local WebSphere application server installation. There is no need to start the local instance, it is sufficient that it is installed and available so that LCM can use it’s library files.

3.5.4. Bootstrap using a custom J2EE application

It is advised to use LCM to bootstrap the LC ES database and deploy the DSC and LCA solution components. However, it is possible to create a custom J2EE application or add functions to an existing application which could initialize and configure LC ES automatically, when the application starts for the very first time after deployment.

However, it is required that the database user used to connect to the LC ES datasources has the required privileges to carry out the bootstrapping, i.e. create tables, views, etc…

The implementation of this kind of automatic bootstrapping and DSC / LCA deployment is not easy. The custom J2EE bootstrapper will need to invoke the original LC ES bootstrap functions and then configure the Reader Extensions certificate, change the admin account password, deploy and configure the DSC and LCA components, etc…

During our discussions with engineering, we've learned that most of the java API's required would be available to carry out these various tasks, but still one has to develop a java application and it may be required to change this custom bootstrapper/deployment tool for upcoming service packs and LiveCycle ES versions.

3.5.5. Native Files – adobeidp.RootDirectory

Adobe LiveCycle ES still uses native files and programs to implement some functions. Usually the system extracts those native files of the services to a folder below appserver root/installedApps, which is problematic. Instead of using this folder, one can specify a custom property named „adobeidp.RootDirectory" which will define the path where the native components will be extracted and executed.

3.5.6. Deployment: Increase SOAP Timeout

It is required to increase the SOAP timeout for deployment and production use as pointed out below. Additionally, the parameters CORBA timeout and TRANSACTION timeout should be changed according to the dokumentation „Installing and Deploying LiveCycle ES for WebSphere" (page 29).

"Increasing the SOAP request time out

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

1. Go to the appserver root directory and search for all files named soap.client.props. Multiple files may have this name. For example, on a UNIX server, the following files exist:
   - appserver root/profileTemplates/default/documents/properties/soap.client.props
   - appserver root/profileTemplates/cell/default/documents/properties/soap.client.props
   - appserver root/cip/profileTemplates/minimal/documents/properties/soap.client.props
   - appserver root/profiles/AppSrv01/properties/soap.client.props
   - appserver root/profiles/AppSrv01/temp/soap.client.props
2. Open each soap.client.props file in a text editor, find the com.ibm.SOAP.requestTimeout property, and change the value from 180 to 1800.
3. Save each soap.client.props file.
4. In the navigation tree of the WebSphere Administrative Console, click Servers > Application Servers and, in the right pane, click the server name.
5. Under Server Infrastructure, click Administration > Administration Services.
6. Under Additional Properties, click JMX Connectors, and then click SOAPConnector.
7. On the next screen, click Custom properties, and then click requestTimeout.
8. On the next screen, in the Value box, change 600 to 1800 and then click OK or Apply.
9. In the Messages box, click Save directly to master configuration."

3.5.7. Deployment: Increase MaxPermSize / heap size

It is also required to change the MaxPermSize and heap size for deployments on UNIX systems. To do this, one has to change the ejbdeploy.sh script used by WebSphere:

"Increasing the MaxPermSize and heap size (UNIX only)
You must increase the MaxPermSize and heap size in the ejbdeploy.sh script to avoid time-out errors.

To increase the MaxPermSize and heap size:

1. Go to the appserver root/deploytool/itp/ directory and open ejbdeploy.sh for editing.
2. In the SunOS section, find the EJBDEPLOY_JVM_OPTIONS attribute and change the value of the –XX:PermSize option to 256m, and ensure that the value of the –Xverify option is none.

3. Change the heap size in the $JAVA_CMD section to the following value:

   * (32-bit OS) –Xms256m –Xmx1024m.
   * (64-bit OS) –Xms256m –Xmx1792m.”

3.6. Next steps

Currently, we are in discussion with the production department. We aim to find a solution to integrate our Adobe LiveCycle ES configuration and deployment mechanisms into their well defined set of procedures and mechanisms for deployment.

The goal is to fully automate the configuration and deployment including bootstrapping and LC ES configuration (deployment of DSC and RE cert) of the 40 x 2 = 80 servers which are currently running the application for the 400000 users.

We’ve unveiled some very interesting problems related to the configuration, deployment, but also regarding the daily production business by discussing this document. We are in contact with engineering and support to find answers and solutions and hope that during the coming weeks we can updated this document to provide our findings and solutions to others.