



Configuring Adobe® LiveCycle® Launch Pad 10

A Guide for Administrators and Developers

March 2012

Launch Pad

Version 10

Legal Notices

For more information, see http://help.adobe.com/en_US/legalnotices/index.html.

Contents

About This Document

Introduction

Deploying Launch Pad

Enabling or Disabling Components

Adding new components

Create the SWF file	7
Build the Launch Pad code	8
Deploy the component	8
Test the component in Launch Pad	9

About This Document

Adobe® LiveCycle® Enterprise Suite 3 (ES3) Launch Pad is an Adobe® AIR client application that provides access from your desktop to the services on your organization's LiveCycle server. You access the services using the components available within Launch Pad.

This document is intended for administrators and developers responsible for managing access to LiveCycle using LiveCycle Launch Pad components.

Conventions used in this document

Name	Default value	Description
[<i>LiveCycle root</i>]	Windows: C:\Adobe\Adobe LiveCycle ES3 Linux and UNIX: /opt/adobe/livecycle/	The installation directory that is used for all LiveCycle modules. The installation directory contains subdirectories for Adobe® LiveCycle® Configuration Manager. This directory also includes directories relating to third-party products.
[<i>Launch Pad root</i>]	[<i>LiveCycle root</i>]/sdk/misc/LaunchPad	The directory where all Launch Pad source files are saved.

Additional information

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

For information about	See
Launch Pad	Launch Pad Help
An overview of LiveCycle	LiveCycle Services
Additional LiveCycle documents and resources	LiveCycle Documentation
Adobe Flash® Builder™	Flash Documentation
Other services and products that integrate with LiveCycle	www.adobe.com
Patch updates, technical notes, and additional information on this product version	Adobe Enterprise Support

Introduction

Adobe LiveCycle Launchpad 10 provides access from your desktop to the services on your organization's LiveCycle server. The following components are included with the SDK and appear in Launch Pad providing the corresponding services are enabled:

- Convert to PDF
- Export PDF
- Create a PDF Package
- Rights Protect Document

Administrators can also set up and enable additional components for end users based on other LiveCycle functionality.

Note: *The service that you want to access by using a component must be installed on the LiveCycle server. Otherwise, the component does not appear within Launch Pad.*

By combining Flex remoting and the services-oriented architecture of LiveCycle server, Launch Pad configures itself each time it restarts. Whenever you start Launch Pad, it queries the LiveCycle server for changes to existing components or for new components to display.

Using Launch Pad benefits all types of users:

- End users can access on-demand services, such as the Generate PDF service from the LiveCycle server without affecting desktop performance. Files are created according to enterprise standards instead of using settings that are selected.
- Administrators can supply services to the desktop without desktop management and without end-user intervention. The administrators deploy their Launch Pad assets, which, in turn, appear on the desktop.
- Developers have a platform for creating end-user values through an SOA programming interface.

Deploying Launch Pad

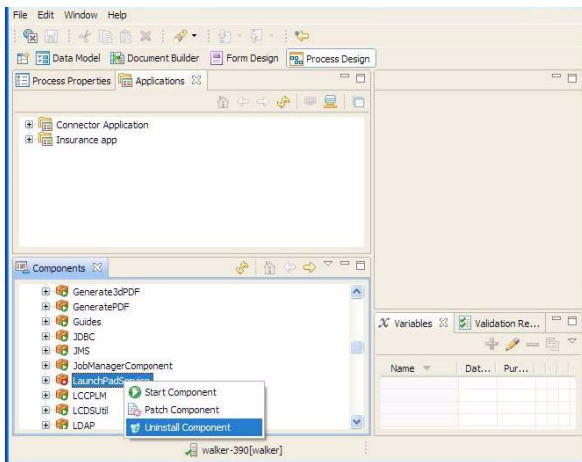
After you complete the LiveCycle installation, you can install and deploy Launch Pad. The following files are used to deploy Launch Pad:

adobe-launchpad-dsc.jar Contains the LiveCycle component called *LaunchPadService*

adobe-launchpad-assets.lca and adobe-livecycle-launchpad.lca LiveCycle archive (LCA) files that contain process definitions for the desktop components

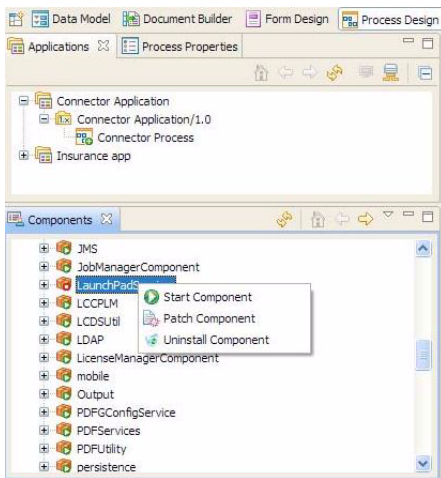
Install LaunchPadService:

- 1 In Adobe® LiveCycle® Workbench, navigate to the Components view.
- 2 Search the component list to see whether LaunchPadService is running. If LaunchPadService is running, stop it, and then uninstall it.



- 3 Right-click the Components view and install the adobe-launchpad-dsc.jar file from the deploy folder.

4 Right-click LaunchPadService and select Start Component.



LaunchpadService is now running. *LaunchpadService* is the container application that interacts with the client Adobe AIR application. LaunchpadService is responsible for identifying and providing the available user interface components to the Adobe AIR application.

To import the LiveCycle archive files:

1 In Adobe® LiveCycle® Administration Console, click Services > Application and Services > Application Management.

Import two archives: adobe.-launchpad-assets.lca and adobe-livecycle-launchpad.lca. The Launch Pad assets archive file contains the SWF application modules and icons that are downloaded to Launch Pad. The Launch Pad services archive file contains the LiveCycle processes that run when Launch Pad makes a call to the LiveCycle server.

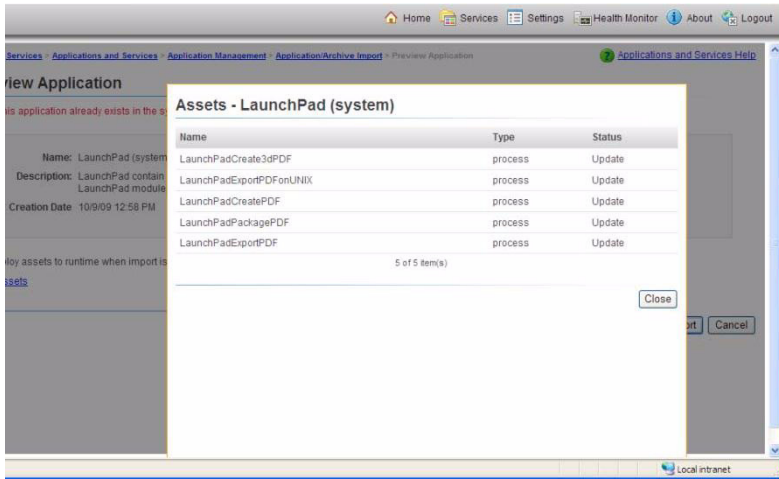
2 Click Import.

3 Select adobe-livecycle-launchpad.lca, and then click Preview.

The adobe-livecycle-launchpad.lca file deploys the processes that are responsible for all the Launch Pad activities.

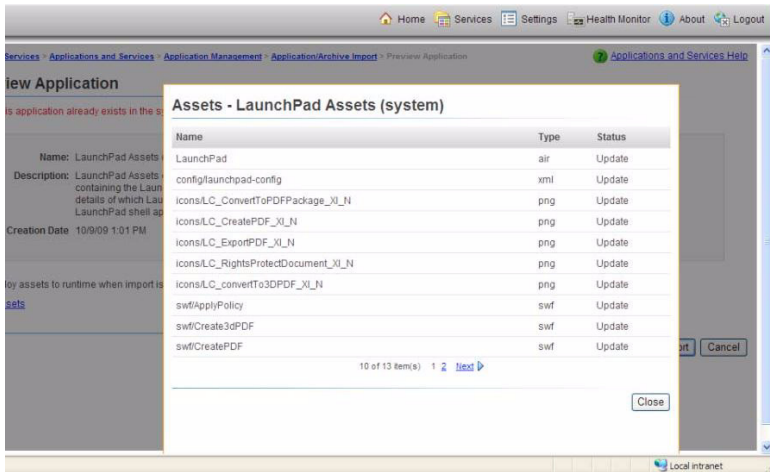
4 To deploy the assets to the run time, select Deploy Assets To Runtime When Import Is Complete.

5 To view a list of the assets that are contained in the file, click View Assets.



6 Click Import. The archive file is deployed.

7 Repeat steps 4-6 but choose the adobe-launchpad-assets.lca file.



The adobe-launchpad-assets.lca file is responsible for deploying the following Adobe AIR client assets:

- The icons for the components
- The SWF file code that runs within the Adobe AIR application and defines the user experience
- The version of the Adobe AIR application used on the desktop
- The configuration file that binds everything together

Launch Pad is now ready to be run, and the components are located on the client desktop.

Enabling or Disabling Components

You control which LiveCycle components are available in Launch Pad by editing a copy of the Launch Pad configuration file (launchpad-config.xml). This configuration file is deployed with Launch Pad. You can place the copy of the configuration file in the Launch Pad Applications view within Workbench.

To enable extending the number of Launch Pad components deployed to the desktop, add a new folder structure to the application assets. The folder structure is `.../ext/config/`. Create this folder structure and copy the launchpad-config.xml file into it. Then, edit the copy of the configuration file. (See “[Deploying Launch Pad](#)” on page 3)

Note: The `/ext/config/launchpad-config.xml` file takes precedence over the `/launchpad-config.xml` file. If a component is disabled in the `/ext/config/launchpad-config.xml` file but enabled in the `/launchpad-config.xml` file, that component does not appear in Launch Pad. It is recommended that you change only the `ext/launchpad-config.xml` file.

To disable a component:

- ❖ Edit the `/ext/config/launchpad-config.xml` file to change the enabled tag from `true` to `false`.

To enable a component:

- ❖ Edit the `/ext/config/launchpad-config.xml` file to change the enabled tag from `false` to `true`.

Adding new components

In addition to the default components, you can create and enable components for other LiveCycle services.

Before you can begin creating a component for Launch Pad, you must have access to the following items:

- **Launch Pad** Includes the `adobe-launchpad-module.swc` library for developing components.
- **Adobe Flash® Builder™ 4.0 or later** Includes the tools to create a SWF file for the component.
- **Workbench** Used to create processes and deploy the component.
- **Adobe® LiveCycle® Foundation** Provides the underlying foundation and services. Launch Pad must be connected to a LiveCycle server. Each component depends on having the associated LiveCycle service available.

The following steps are for adding a new component:

- Create the SWF file that runs within the desktop client.
- Deploy the component.
- Test the component's accessibility in Launch Pad.

Create the SWF file

The first step for adding a new component is to create the SWF file.

- 1 Copy the Launch Pad library file (`adobe-launchpad-module.swc`) from `[LiveCycle root]/sdk/client-libs/common` directory to the `/library` folder of the project.

The library contains the `com.adobe.launchpad.ClientHubComponent` class. The components that you develop must be a subclass of the `com.adobe.launchpad.ClientHubComponent` class, which has the following properties:

- `serverProtocol`
- `serverName`
- `serverPort`
- `userName`
- `password`

Components use these properties to interact or load the services on the LiveCycle server.

- 2 Copy the following files from `[Launch Pad root]/pdfg/src/com/adobe/launchpad/ui/common/effect/` to the project under a package:
 - `AppearEffect.mxml`
 - `DisappearEffect.mxml`
 - `ClarifyEffect.mxml`

Be sure to edit the package declaration line of these files accordingly.

- 3 In the project's main MXML file, change the root element of the project from `WindowedApplication/Application` to `LaunchpadComponent`. The initial code of the MXML looks similar to the following code:

```
<ch:LaunchpadModule xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute"
xmlns:ch="com.adobe.launchpad.*" creationComplete="initData(this)"
xmlns:effect="com.adobe.launchpad.ui.common.effect.*"
xmlns:common="com.adobe.launchpad.ui.common.*" hideEffect="{disappear}" showEffect="{appear}"/>
<effect:AppearEffect id="appear"/>
```

```
<effect:DisappearEffect id="disappear"/>
<effect:ClarifyEffect id="clarify"/>
```

This code enables the sliding effect for showing and hiding a component in Launch Pad to the new component.

- 4 Update ChannelSet in the new component to communicate with the LiveCycle server:

```
remote_object>.channelSet = this.lcChannelSet as ChannelSet;
```

- 5 Compile the component by using the Adobe Flash[®] compiler.
- 6 Use the @Embed tag to embed images that are used in the component, in the SWF file.

The new component is ready to be deployed on the LiveCycle server.

Build the Launch Pad code

Perform the following steps to build the Launch Pad code:

- 1 Set the FLEX_HOME environment variable to the Flex SDK root directory.
- 2 Set the LiveCycle_ES_SDK environment variable to `[dep root]/sdk`.
- 3 Set the ANT_HOME environment variable to point to Ant (1.7 or later) root directory. Also, add %ANT_HOME%/bin to the system PATH variable.
- 4 Download the ant-contrib.jar file from <http://sourceforge.net/projects/ant-contrib/files/> and copy it to the `[ANT_HOME]/lib` directory.
- 5 Set the JAVA_HOME environment variable to the Java home directory. Also, add %JAVA_HOME%/bin to the system PATH variable.
- 6 Copy the jdom.jar file (jdom-b7.zip) from <http://jdom.org/dist/binary/archive/> to the `[dep root]/sdk/client-libs/thirdparty` directory.
- 7 Extract the `[Launch Pad root] /lc-launchpad-src.zip` file and follow the instructions to comment or uncomment the code lines in all build.xml files in the extracted directories.
Tip: Use the keyword "comment" to search for instructions.
- 8 Update the value of `flex.sdk.dir` in the build.properties file to point to the Flex-sdk-3.4 location.
- 9 Execute the ant command from the `[Launch Pad root]` directory to build the code.

Deploy the component

Applications are managed through the Application view in Workbench. Launch Pad has two applications: the system application and the Launch Pad Assets application, the application that contains assets and configuration information for Launch Pad.

- 1 To open the Launch Pad assets application from Workbench, click File > GetApplications.
- 2 Open Launch Pad Assets (system), and then select Launch Pad Assets (system)/1.0. The folder structure appears in the Applications view.
- 3 In Workbench, deploy the SWF file to the SWF file folder in the assets application in the Applications view.
- 4 Add the icon (32 X 32 pixels) that appears in the client to the Launchpadcomponent/icons folder in Workbench.
- 5 Create a folder called /ext/config in the Launchpadcomponent folder.
- 6 Copy the launchpad-config.xml file to this folder.
- 7 Edit the launchpad-config.xml file and create a component tag. The component tag supports the following tags:
 - **id** Identifies the component. If more than one component with the same ID is present in the configuration file, only the last occurrence is enabled in Launch Pad.
 - **name** Name of the component. This text is used as the tooltip and application title when the component is enabled.

- **component-url** URL of the component's SWF file in the repository.
- **description** Short description of the component.
- **icon-url** URL of the component's icon (png) in the repository.
- **enabled** Specifies whether this component is enabled. The default is true.
- **order** Specifies the order in which the component is shown in Launch Pad. The components are sorted alphabetically in Launch Pad.
- **services** A comma-separated list of services that the component uses. If a service is specified in this list, the component is sent to Launch Pad. The service must be installed on the LiveCycle server.
- **role** A comma-separated list of roles that are required to download and run the component. If a role is specified, the component appears only if the current user has those roles.

An example of a component tag

```
<updates>
  <enabled>true</enabled>
  <current-app-version>9.0.20090707.004745</current-app-version>
  <force-update>>false</force-update>
  <app-url>/Applications/LaunchpadAssets/1.0/Launchpad.air</app-url>
</updates>
<modules>
  <module>
    <id>IdOfTheNewcomponent</id>
    <name>New component</name>
    <module-url>/Applications/LaunchpadAssets/1.0/swf/Newcomponent.swf</component-url>
    <description>New component</description>
    <icon-url>/Applications/LaunchpadAssets/1.0/icons/Newcomponent.png</icon-url>
    <enabled>true</enabled>
    <order>1</order>
    <services>AssemblerService</services>
    <supported-operating-system>ALL</supported-operating-system>
  </module>
</modules>
```

Test the component in Launch Pad

You test the component's accessibility in Launch Pad.

- 1 Start Launch Pad. The new component appears in the list of components.
- 2 Click the component and confirm that the service starts as expected.