

Perceptive Experience Content Apps

Installation and Setup Guide

Version: 2.2

Written by: Product Knowledge, R&D

Date: Wednesday, April 19, 2017

© 2014-2017 Lexmark. All rights reserved.

Lexmark is a trademark of Lexmark International Inc., registered in the U.S. and/or other countries. All other trademarks are the property of their respective owners. No part of this publication may be reproduced, stored, or transmitted in any form without the prior written permission of Lexmark.

Table of Contents

About Perceptive Experience	4
Prepare to install Perceptive Experience	4
Installation overview	4
Download and place files	4
Set up SSL	5
Disable HTTP pipelining	5
Configure Perceptive Experience for single sign-on	5
X.509 SSO Certificate Support	6
About using cache headers with Perceptive Experience	6
Enable Expires header	6
Start Perceptive Experience	7
View package version information	7
Choose a logging level	8
About using Apache Tomcat compression settings	8
config.json properties	8
Update config.json properties	15
	15

About Perceptive Experience

Perceptive Experience allows you to view and manage your structured and unstructured content with the use of Apps on any device.

With the flexibility of Perceptive Experience you can take content residing in Perceptive Content and further manage or view it from a desktop browser or mobile device.

Prepare to install Perceptive Experience

These instructions assume that you are installing Perceptive Experience for the first time or that you have no earlier versions of Perceptive Experience running on your computer.

The following steps outline the high-level procedures that you need to perform to install and configure Perceptive Experience, and the information you need to verify before installation.

Installation overview

During the installation of Perceptive Experience, you perform the following procedures.

1. Install prerequisite applications. To work with Perceptive Content related Apps you must have Integration Server and Apache Tomcat installed.
2. Download and copy files to the web server directory.
3. Enable SSL.
4. Disable HTTP pipelining
5. Configure SSO
6. Enable `Expires` header in the web application server.

Download and place files

To download and place files in the web application server directory, complete the following steps.

Prerequisite Stop the web application server service.

1. Go to www.lexmark.com and log in to the **Lexmark Enterprise Software Customer Portal**.
2. Click **Downloads**, and search for all downloadable items for this product.
3. Download the application *ZIP* file to a temporary directory on your computer.
4. Create a subdirectory in the *webapps* directory of the web application server. You may name the subdirectory with any name.

Note: The default location for the *webapps* directory is *[drive:]\Program Files\Apache Software Foundation\[Tomcat Installation]\webapps*

5. Extract the files to the subdirectory you created in the previous steps.

Next Restart the web application server service.

Set up SSL

You must configure use of SSL when using Perceptive Experience with Perceptive Integration Server version 7.1.5 or higher. To configure use of SSL you must do so on your web application server.

SSL configuration cannot be completed within Perceptive Experience. For more information refer to your specific web server help documentation.

Disable HTTP pipelining

You should disable HTTP pipelining for the Apache Tomcat web server used to access Perceptive Experience Apps. If pipelining is used by a browser accessing the Experience environment, users may encounter errors when loading packages in your environment. To disable HTTP pipelining, complete the following steps.

1. Navigate to the Tomcat *server.xml* file.

The default location for the system configuration file is `[drive:]\Program Files\Apache Software Foundation\[Tomcat Installation]\conf`

2. Add either the `maxKeepAliveRequests` or `restrictedUserAgents` attributes within the `Connector` section of the Tomcat *server.xml* file

Attribute	Valid value
<code>maxKeepAliveRequests</code>	1
<code>restrictedUserAgents</code>	A comma separated list of regular expressions identifying HTTP clients that should not use HTTP/1.1 or HTTP/1.0 Keep Alive even if client support is specified.

Configure Perceptive Experience for single sign-on

Single sign-on (SSO) allows users to access the system by authenticating with an external identity provider. Currently Perceptive Experience is the only Perceptive product that supports Integration Server's SSO configuration. To enable single sign-on for Perceptive Experience, complete the following steps.

Prerequisite You must configure the Integration Server for single sign-on. For Integration Server configuration instructions see the *Perceptive Integration Server Installation and Setup Guide*.

1. Stop the web application server.
2. Navigate to the root folder on the installation directory for **Perceptive Experience**.
3. Protect the `sso` folder found at `rootpath/packages/framework-core/sso`.
4. Open the `config.json` file in a text editor.

The default location for the Perceptive Experience configuration file is `[drive:]\Program Files\Apache Software Foundation\[Tomcat Installation]\webapps\<subdirectory>`.

5. Add the `clientSession` parameter to specify the server connection. The valid entry for `sso` is `framework-core:SsoClientSession`.

Example

```
"framework-core": {  
    "clientSession": "framework-core:SsoClientSession",  
    "loggingMethod": "Console"  
},
```

Note: When adding parameters to the config.json file, add a comma between each parameter.

6. Restart the application server.

X.509 SSO Certificate Support

Perceptive Experience Apps utilizing Perceptive Integration Server support LDAP authentication with X.509 client certificates.

Using Perceptive Experience Apps in a browser supports the use of multiple client certificates. Perceptive Experience for Windows Desktop supports the import of a single X.509 client certificate.

About using cache headers with Perceptive Experience

To ensure that the web application displayed in a web browser is current, configure the web application server to send an `Expires` header.

Web browsers cache data fetched from the web application server on the initial web page load. On subsequent page loads, the server determines if the browser has the most current content saved in the cache. If the content is current, the server returns a 304 response, and the browser continues to use the cached content. If data on the server has changed since the last request, the content in the browser's cache is considered stale. The server then returns new content and the browser updates the cache with the new content.

Some browsers do not make a call to the server to check for new data unless the cached version is considered expired. These browsers only consider data expired if there is an indicator in the web server response, such as an `Expires` header.

To avoid displaying old content when a new version of an application is deployed, the web server should send an `Expires` header in the response. Browsers that rely on an `Expires` header make a call to the server after the specified expiration time and retrieve any updates to an application file.

Enable Expires header

You must use an `Expires` header to ensure the application runs correctly and invalid content is not displayed to users. To configure Apache Tomcat to use an `Expires` header, complete the following steps.

1. Navigate to the Apache Tomcat `conf` directory. The default location is `[drive:]\Program Files\Apache Software Foundation\[Tomcat Installation]\conf`
2. Open the `web.xml` file with a text editor and add the following code to the file.

```
<web-app>  
  
    <filter>
```

```

<filter-name>ExpiresFilter</filter-name>

<filter-class>org.apache.catalina.filters.ExpiresFilter</filter-
class>

<init-param>
    <param-name>ExpiresDefault</param-name>
    <param-value>now plus 0 minutes</param-value>
</init-param>
</filter>

<filter-mapping>
    <filter-name>ExpiresFilter</filter-name>
    <url-pattern>/*</url-pattern>
    <dispatcher>REQUEST</dispatcher>
</filter-mapping>

...

```

Result In the example above, content expires immediately, so all browsers will always ask if the cached copies of files need updating. You can modify the `param-value` content to suppress the check for a set amount of time after the web browser saves the data, but the browser will not display updates to the application until the amount of time defined by `param-value` has passed. During the time that the check for new content is suppressed, it is possible that an application will stop functioning properly.

Start Perceptive Experience

To start Perceptive Experience complete the following steps.

Prerequisite Ensure that you have the latest Internet browser installed.

1. Open your browser and enter the **Perceptive Experience** URL. The URL format is `http://<server name>:<port number>/<subdirectory>`, where `<server name>` and `<port number>` are placeholders for the name of the computer and the port number where you installed **Perceptive Experience**. The subdirectory is the location within the web application server where you installed **Perceptive Experience**.


Note: The default port number for the Apache Tomcat web application server is 8080.

2. Enter your `user name` and `password`.
3. Select **Connect**.

View package version information

To view the packages and dependencies installed, complete the following steps.



1. Select **Profile** .

2. Select **Settings** .
3. Select **About**.

Under **Packages and Dependencies**, the name and version number of packages and dependencies are listed.

Choose a logging level

To set the logging level for the current user, complete the following steps.

1. Select **Profile** .
2. Select **Settings** .
3. Under **Global** select one of the following logging levels.
 - To set the maximum logging level, select **All Messages**.
 - To set the default logging level, select **Warnings and Errors**.
 - To set the minimum logging level, select **Errors Only**.

About using Apache Tomcat compression settings

You can implement Apache Tomcat web server compression options to save bandwidth as files are delivered to the browser. Saving bandwidth during file transmission can decrease the initial load time of the framework and packages.

There are several settings you can modify within the Tomcat *server.xml* file to implement file compression. The default location for the *server.xml* file is *[drive:]\Program Files\Apache Software Foundation\[Tomcat Installation]\conf*.

For more information, refer to the Tomcat help documentation on the Apache website.

config.json properties

The following list provides information about the properties defined in the *config.json* file. The *config.json* file contains runtime parameters that can be defined by an administrator.

The default location for the system configuration file is *[drive:]\Program Files\Apache Software Foundation\[Tomcat Installation]\webapps\<subdirectory>*.

The properties listed are in the order they appear within the *config.json* file.

helpUrl

Specifies the URL for online help. Valid values include any URL.

helpPage

Specifies the help landing page for the active module. This page information is appended to the `helpUrl` to form a complete context sensitive URL address.

disableSslWarning

By default, this property is not defined. Add this property and populate it with a `true` value to disable the warning that displays when the application uses a non-secure connection. Setting this property to `true` transmits unencrypted user names and passwords.

framework-runtime

modules

Specifies the modules currently loaded into the application that users can access from the launcher.

lesrdl-content-integrationserver

serverUrl

Specifies the URL to Integration Server, which should be a relative path.

heartbeatIntervalInSeconds

Specifies how often, in seconds, the application verifies its connection to ImageNow Server.

lesrdl-content-search-controls

vsl-extended-operators-enabled

Specifies whether Perceptive Content adds the “contains” and “ends with” operators in the quick search options.

The default value is `false`.

Note: When set to true, Perceptive Content includes the additional search operators.

framework-core

clientSession

By default, this property is not defined. Specifies the current server connection that the application is using.

Valid values for `clientSession` include `framework-core:SsoClientSession`, `lesrdl-content-integrationserver:ClientSession`.

sessionTimeoutInSeconds

Specifies the duration of time, in seconds, that a session can remain inactive before the system logs the current user out of the application.

Valid values are any integer. If set to 0 or -1, the system disables automatic logout.

loggingMethod

Specifies the way users can review log contents.

Valid entries include the following values.

- `externalwindow` - Use this property for web-based applications to display the log in a new browser tab or window.
- `console` - Use this property for web-based applications to display the log in the browser console.

loggingLevel

Specifies the messages that displays to the user in the application log.

Valid entries include the following values.

- VERBOSE
- INFO
- WARNING
- ERROR
- CRITICAL

logoutRedirect

Specifies the URL the system redirects to upon log out for systems utilizing single sign on. The URL can be a relative or absolute path.

contentCapture

The properties you can define in the contentCapture section of the config.json file are listed below.

postSaveScriptName

Specifies the post-save iScript file.

toolkitDownloadLocation

Specifies the URL of the Perceptive Web Scan installer.

sourceProfileToScannerPageSizeMapping

Specifies the Perceptive Content source profile page size available for mapping.

```
"contentCapture": { "sourceProfileToScannerPageSizeMapping": [
"letter:/letter/i", "legal:/legal/i", "a4:/a4/i", "a5:/a5/i", "a6:/a6/i",
"b5:/b5/i", "b6:/b6/i", "executive:/executive/i", "us check:/check/i",
"custom:/custom/i", "business card:/card/i", "scanner maximum:/maximum/i"
] }
```

profileNameToSource

Specifies mapping between a capture profile name and capture source identifier.

les-rdk-fpa-viewer

The property you can define within les-rdk-fpa-viewer in the lesrdl-content-documents section of the config.json file is listed below.

buttons

Specifies if the buttons are disabled or hidden in the viewer. Buttons include ANNOTATION, SAVE, DELETE, ADDREPLACE, and EXIT. Valid values include `disabled` and `hidden`.

For example, "lesrdl-content-documents": { "les-rdk-fpa-viewer": { "buttons": { "ANNOTATION": "disabled", "DELETE": "hidden", "SAVE": "disabled", "ADDREPLACE": "hidden", "EXIT": "disabled", "TASK": "hiDDen"}}}

Note: Inclusion of this property is optional and is subject to the requirement of the consuming module.

psw-content-forms

The property you can define in the psw-content-forms section of the config.json file is listed below.

loadByDefault

Specify whether the first form in the list displays by default when viewing the Forms pane. The default setting is `false`.

les-rdk-fpa-indocumentsource

The properties you can define in the les-rdk-fpa-indocumentsource section of the config.json file are listed below. These properties are not defined by default but can be set when an App utilizes the les-rdk-fpa-indocumentsource package.

maxWidth

Specifies the maximum width in pixels of thumbnails that are displayed.

maxHeight

Specifies the maximum height in pixels of thumbnails that are displayed.

Note: The values of `maxHeight` and/or `maxWidth` can only be positive integers.

noRenditionFileTypes

Helps bypass rendition calls for certain file types. If a bypassed file type is not directly supported in the browser the document is viewed in, some features of the viewer, such as annotations, becomes unavailable.

Note: Microsoft Internet Explorer and Microsoft Edge are the only browsers that natively support the rendering of TIFFs.

In the example mentioned below, rendition call is bypassed for JPEG and JPG files.

```
"noRenditionFileTypes" : ["JPEG", "Jpg"]
```

Note: Values of this property are not case-sensitive.

les-rdk-fpa-print

The property you can define in the `les-rdk-fpa-print` section of the `config.json` file is listed below.

pdfConversion

Helps print a document in the viewer using PDF plugin. The viewer converts all the pages into a single PDF file and displays it in your web browser. Valid values include `true` or `false`. For example,

```
"print":{ "pdfConversion" : "true"}
```

Note: Ensure that Adobe Acrobat Reader plugin is installed and enabled in all the web browsers except Google Chrome. In Google Chrome, use the default PDF plugin. When printing in Internet Explorer 10 and 11, set the margin to "0" on the print page setup and ensure that the page size is as desired.

les-rdk-fpa-action-manager

Helps manage all the actions registered by different modules of an application. To ensure that the actions are loaded during the application loading, include the action modules in the `preModules` section of the `config.json` file. For more information, see [les-rdk-fpa-action-manager API documentation](#).

les-rdk-fpa-action-manager Actions

The actions you can perform under `les-rdk-fpa-action-manager` are listed below.

les-rdk-fpa-action-open-in-workflow

Opens any Perceptive Content document in a workflow context if the document is added to a workflow. The property you can define in the `les-rdk-fpa-action-open-in-workflow` section of the `config.json` file is mentioned below.

instanceName

Specifies the name of the instance of the loaded action manager. For example:

```
"les-rdk-fpa-action-open-in-workflow":{ "instanceName" :  
"NameOfTheActionManagerInstance" }
```

les-rdk-fpa-action-print-document

The properties you can define in the les-rdk-fpa-action-print-document section of the config.json file are listed below.

preferenceWindow

Specifies the printing preferences. Displays a window for taking print preferences from the user. The default value is `true`.

Note: Other print configuration options are overlooked if preferenceWindow value is `true`.

enableAnnotations

Specifies if the annotations are printed. Valid values include `true` or `false`.

pageSize

Specifies the paper size printed. The default value is `letter`.

maxWidth

Specifies the maximum width in pixels of pages that are printed.

maxHeight

Specifies the maximum height in pixels of pages that are printed.

hideAnnotationTypes

Specifies the annotation types that are not displayed.

coverPage

Specifies if the cover page of a document is to be printed. The default value is `false`.

pdfConversion

Converts all the pages into a single PDF file and displays it in your web browser. The default value is `false`.

les-rdk-fpa-action-email-workflow

The properties you can define in the les-rdk-fpa-action-email-workflow section of the config.json file are listed below. This action allows the user to email a link to a workflow item.

to

Specifies the email address of the recipient.

cc

Specifies a carbon copy of an email when the user wants to include someone other than the direct recipient.

bcc

Specifies a blind carbon copy of an email when the user does not want the other recipients to see who else is included in the email.

subject

Specifies a subject of an email where the user can add a short description of the main content.

preBody

Specifies the section where the user can add any content before a workflow item link.

postBody

Specifies the section on complimentary closure, such as "Thanks & regards".

les-rdk-fpa-action-email-document

The properties you can define in the les-rdk-fpa-action-email-document section of the config.json file are listed below. This action allows the user to email a link to a Perceptive Content document.

to

Specifies the email address of the recipient.

cc

Specifies a carbon copy of an email when the user wants to include someone other than the direct recipient.

bcc

Specifies a blind carbon copy of an email when the user does not want the other recipients to see who else is included in the email.

subject

Specifies a subject of an email where the user can add a short description of the main content.

preBody

Specifies the section where the user can add any content before a workflow item link.

postBody

Specifies the section on complimentary closure, such as “Thanks & regards”.

Update config.json properties

When you update your Perceptive Experience Apps to version 2.0 or later, you should update the following config.json properties.

The list below provides the existing property names in Perceptive Experience Content Apps versions 1.0 to 1.4 as well as the change following an upgrade to Perceptive Experience Apps 2.0 or later. Update your new config.json entries to match the new values.

preloadModules

These entries are required system settings. Do not modify entries in the `preloadModules` section of the `config.json` file.

modules

This entry is now in the `framework-runtime` section of the configuration file.

Specifies the modules currently loaded into the application that users can access from the launcher.

inactiveModules

Specifies any Apps that should not be available to users from the Experience Home page. If you do not want the icon for an App that is provided by default at installation to appear on the Home page, move the full name of the desired package from the `modules` section to this section.

sessionKeepAliveHeartBeatInterval

This entry is now in the `lesrdl-content-integrationserver` section of the configuration file and renamed `heartbeatIntervalInSeconds`.

Specifies how often, in seconds, the application verifies its connection to ImageNow Server.

serverUrl

This entry is now in the `lesrdl-content-integrationserver` section of the configuration file.

Specifies the URL to Integration Server, which should be a relative path.

connection

This entry is now in the `framework-core` section of the configuration file and is renamed to `clientSession`.

Valid values for `clientSession` include `framework-core:SsoClientSession`, `lesrdl-content-integrationserver:ClientSession`.

disableSslWarning

This entry should now be in the `framework-shell` section of the configuration file.

Valid values include `true` or `false`.

By default, this property is not defined. Add this property and populate it with a `true` value to disable the warning that displays when the application uses a non-secure connection. Setting this property to `true` transmits unencrypted user names and passwords.

sessionTimeoutInSeconds

This entry is now in the `framework-core` section of the configuration file.

Valid values are any integer. If set to 0 or -1, the system disables automatic logout.

loggingMethod

This entry is now in the `framework-core` section of the configuration file.

Valid entries include the following values.

- `externalwindow` - Use this property for web-based applications to display the log in a new browser tab or window.
- `console` - Use this property for web-based applications to display the log in the browser console.

loggingLevel

This entry is now in the `framework-core` section of the configuration file.

Valid entries include the following values.

- `VERBOSE`
- `INFO`
- `WARNING`
- `ERROR`
- `CRITICAL`

document

This property is now renamed `lesrdl-content-documents`. Refer to “`lesrdl-content-documents`” in the `config.json` properties section for a detailed information on its defined properties.

psw-content-forms

This property is now renamed `lesrdl-content-forms`. Refer to “lesrdl-content-forms” in the `config.json` properties section for a detailed information on its defined properties.

psw-content-indocumentsource

This property is now renamed `les-rdk-fpa-indocumentsource`. Refer to “les-rdk-fpa-indocumentsource” in the `config.json` properties section for a detailed information on its defined properties.

psw-content-print

This property is now renamed `les-rdk-fpa-print`. Refer to “les-rdk-fpa-print” in the `config.json` properties section for a detailed information on its defined properties.

viewer

This property is now renamed `les-rdk-fpa-viewer` in the `lesrdl-content-documents` section of the `config.json` file. Refer to “les-rdk-fpa-viewer” in the `config.json` properties section for a detailed information on its defined properties.