Perceptive Experience Content Apps

Installation and Setup Guide

Written by: Product Knowledge, R&D
Date: Tuesday, January 19, 2016

Lexmark™ Enterprise Software
formerly Perceptive Software

Table of Contents

What Is 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-based 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 `Expires` header in the web application server.
4. Enable SSL.

If you want to configure SSL for use when running Apps within a web browser, you must do so on your Tomcat web application server. For more information, refer to the Tomcat help documentation on the Apache website.

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 the Enterprise Software website at http://www.perceptivesoftware.com and log in to the **Customer Portal**.
2. In the **Products Downloads** page, 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

It is recommended that you configure use of SSL when using Perceptive Experience. If you want 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.

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 the icon.
2. Select the **Settings** icon.

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

Configure Perceptive Experience for single sign-on

Single sign-on (SSO) allows users to access Integration Server 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 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. Open the *config.json* file in a text editor.
4. Add the following setting.

> **Note** When adding settings to the config.json file, add a comma between each setting.

```
connection: core:SsoConnection
```

5. 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.

About using Tomcat compression settings

You can implement Tomcat 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.

Choose a logging level

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

1. Select the 👤 icon.
2. Select the **Settings** ⚙ icon.
3. Under **Settings**, select **Global** and 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**.

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 Perceptive Experience 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.

connection

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

Valid values can include `core:SsoConnection`

desktop-service-connector

Specifies the URL at which to connect the desktop master service. Use the `endpoint` option to designate the URL of the desktop service connector.

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.

serverUrl

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

sessionKeepAliveHeartBeatInterval

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

sessionTimeoutSeconds

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.

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.

loggingLevel

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

Valid entries include the following values.

- VERBOSE
- INFO
- WARNING
- ERROR
- CRITICAL

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.

modules

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

Shell

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

globalSearch

Specifies the object that will serve as the shell’s global search widget.

document

The properties you can define in the document section of the config.json file are listed below. These properties support print functionality.

enableannotations

Specifies the annotation types that are displayed.

Valid values include `true` or `false`.

> **Note** When set to false, the document is printed without annotations.

pagesize

Specifies the paper size printed.

Valid values include `A4`, `A3`, and `letter`.

> **Note** The default value is A4.

maxwidth

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

maxheight

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

hidden

Specifies the visibility configuration of the document toolbar icons. For example,

```
"document": { "viewer": { "buttons": {"DELETE": "hidden"}} }
```

> **Note** Values of the `hidden` property are case-sensitive.

disabled

Specifies the functionality configuration of the document toolbar icons. For example,

```
"document": { "viewer":{"buttons":{"SAVE":"disabled"}} }
```

> **Note** Values of the `disabled` property are case-sensitive.

hideannotationtypes

Specifies the annotation types that are not displayed.

Valid values include `all`, `false`.

> **Note** When set to all, any annotation type the user sets to hide is not displayed, provided the user has the `canHide` privilege.

> **Note** If no parameters are set in the *config.json* file, by default, default page size is letter and the `enableannotations` value is `true`.

Indocumentsource

The properties you can define in the 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 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.