Perceptive Forms Server

Installation and Setup Guide

Version: Foundation 23.1

Written by: Documentation Team, R&D
Date: June 2023

Hyland™

Documentation Notice

Table of Contents

Overview

Perceptive Forms Server enables you to view a form in your browser using a URL. After you install and set up Forms Server, refer to the Access the forms URL section of this document. You can only access forms published as stand-alone documents or folders.

This document explains how to download and deploy Forms Server, and assumes you have Perceptive Integration Server installed. For installation instructions, refer to the *Perceptive Integration Server Installation Guide*.

This document also assumes you are installing Forms Server for the first time or that you have no earlier versions running on your computer. To update Forms Server from a previous version, refer to the *Perceptive Content Update Guide*. Sections of the update guide may reference procedures in this installation guide. You must also have Apache Tomcat running prior to installing Forms Server.

Note Starting with version 7.1.2, Java is no longer required to view forms from Forms Server. For Forms Server eForms viewing requirements, and all other technical and system requirements for Forms Server, refer to the 7.1.x *Perceptive Content Technical Specifications*.

Secure environment recommendations

The following recommendations are intended to help secure the installation environment and apply to all product installations. These recommendations should be followed as a minimum requirement for all Hyland products. The policies of your organization may have additional or more robust requirements that should also be followed.

Hyland products may have additional recommendations described in the specific documentation for that product. In some cases, the recommendations may change or may only apply when using certain Hyland products together in a solution.

- Use TLS for all HTTP traffic, including private network segments. TLS ciphers have to be maintained to stay current over time.

- Use Secure FTP instead of standard FTP for all FTP traffic.

- End-to-end encryption is recommended for all data in transport, independent of a network segment. Note that some regulatory compliance requirements may require end-to-end encryption.

- Change all default passwords before activation of the production system. This applies to Hyland products as well as third-party products used by Hyland products (such as a database server).

- Authorization rules should be configured and tested before activation of the production system. This applies to Hyland products as well as to file system folders and database user accounts.

- Use database encryption for all sensitive data persisted in the database.

- Use file system encryption for all sensitive data and content persisted on the file system.

- Enable encryption when available as part of a subsystem configuration. For example, since ODBC provides the capability to use strong encryption for data, it is recommended to have that option enabled.

Side-channel risks

Consider the following to mitigate side-channel risks:

- Ensure the latest application and operating system patches are applied.
- Ensure the latest firmware patches are applied for any hardware on-premises.

Prepare to install

During this installation, you perform the following procedures.

- Download the Forms Server file
- Deploy Forms Server
- Deploy Forms Server on Tomcat 10.1.x
- Configure Forms Server
- License Forms Server

Forms Server installation

Download the Forms Server file

To obtain Perceptive product installation files, contact the Hyland Software Technical Support group. For a list of Technical Support phone numbers, go to hyland.com/pswtscontact.

Deploy Forms Server

You must have Apache Tomcat installed as the web application server before you deploy Forms Server. To deploy the downloaded Forms Server, complete the following steps.

1. Stop Apache Tomcat.
2. Move the **psw-content-forms-server** WAR file to the **${TOMCAT_HOME}/webapps** directory.
3. Start Apache Tomcat, which automatically unzips and deploys the psw-content-forms-server WAR file.

   Note You can use other procedures to deploy a WAR file to Apache Tomcat. The subdirectory name depicts the URL used to navigate to a form. For example,
   http://<hostname>:<port#>/<subdirectory_name>/fs?form=<Document_Type_Name>.

Deploy Forms Server on Tomcat 10.1.x

You must have Apache Tomcat installed as the web application server before you deploy Forms Server. To deploy the downloaded Forms Server on Tomcat 10.1.x, complete the following steps.

1. Stop the Apache Tomcat server.
2. Locate the **tomcat-psw-content-forms-server_jakartaee9_xxx.war**. For Tomcat 10.1.x, the required .war file contains _jakartaee9 in the file name. This file can be renamed to **psw-content-forms-server.war** to match earlier installation steps.

Configure Forms Server

After deploying the psw-content-forms-server WAR file, you must make some changes to the imagenowforms.settings configuration file. You can use the settings file to customize how Forms Server works.

Note When you make changes to the Forms Server settings file, you must restart Tomcat for the changes to be applied to your environment.

1. Navigate to the [path]/${TOMCAT_HOME}/webapps/<subdirectory_name>/WEB-INF directory.

2. Using a text editor, open the **imagenowforms.settings** file.

3. Configure the following parameters under the **Perceptive Content Server** settings section.

    1. In **host**, enter the host name or IP address of the Perceptive Content server.

    2. In **port**, enter the port number of the Perceptive Content Server.

4. Under the **Application server settings** section, in the **integration-server-path** setting, enter the full path to the Integration Server.

    Note For more information regarding the communication between Forms Server and Integration Server, refer to the Cross-domain requests section of this document.

5. Save and then close the file.

Configure agent token authentication

To configure Integration Server to use token based authentication for Forms Server requests, complete the following steps.

1. On the Perceptive Content server machine, generate an authentication token for Forms Server by running the following command.

   ```
   intool --cmd create-authentication-token --lictype ImageNow Forms Server --file
   FormsServerToken.txt
   ```

2. On the Integration Server machine, navigate to the [path]/${TOMCAT_HOME}/webapps/<integration server subdirectory>/WEB-INF directory.

3. Using a text editor, open the **integrationserver.ini** file.

4. Under the **SSO** section, set the **forms.server.authentication.token** setting to the contents of the **FormsServerToken.txt** file, as shown in the following example.

   ```
   forms.server.authentication.token=[authentication token]
   ```

5. Save and then close the file.

6. Navigate to the [path]/${TOMCAT_HOME}/webapps/<subdirectory_name>/WEB-INF directory.

7. Using a text editor, open the **imagenowforms.settings** file

8. Under the **Perceptive Content Server** settings section, set the **forms.server.authentication.token** setting to the contents of the **FormsServerToken.txt** file, as shown in the following example.

   ```
   forms.server.authentication.token=[authentication token]
   ```

9. Save and then close the file.

10. Restart Integration Server and Forms Server.

Configure Integration Server for token-based agent authentication

After you configure agent token authentication, you must also configure Integration Server to support token-based authentication for agent connections. For more information on importing a token signing certificate and configuring Integration Server, refer to the Perceptive Integration Server on Tomcat Installation Guide or Perceptive Integration Server on WebSphere Installation Guide.

Obtain the license files

To generate a system fingerprint, complete the following steps.

Note You must be the admin user on Windows to obtain the hardware information for the Perceptive Content Server.

1. Click Start > All Programs > Perceptive Content > Perceptive Content Management Console.

2. In the **Login** page, click **License Manager**.

3. In the License Management dialog box, select Save system fingerprint and click OK.

4. In the **Save As** dialog box, enter a name for the file and then navigate to the location where you want to save the report. Click **Save**.

5. Contact your Enterprise Software representative for instructions on where to send the system fingerprint file to obtain your license. The system fingerprint file has a SYSFP extension.

6. When you receive the license files, store the license files in a temporary directory on the Perceptive Content Server computer.

Install Perceptive Content product licenses

To upload licenses, complete the following steps.

Note Before entering your licenses, you must have installed the Perceptive Content Server and at least one Perceptive Content Client. You must have Service Administrator privileges to install Perceptive Content licenses. The Client must be available on a Windows machine to install the product licenses.

1. When you receive the license files from your Enterprise Software representative, copy them to a temporary folder where you can access them with a Perceptive Content Client.

2. Click **Start**, point to **All Programs**, and then select **Perceptive Content**.

3. In the login page, click **License Manager**.

4. In the **License Management** dialog box, select **Upload Licenses** and click **OK**.

5. Navigate to the folder where you stored the Perceptive Content license files, select the LIC files to upload, and click **Open**.

6. Enter the User Name, Password, and Server Location and click **OK**.

7. Optional. The **License Upload** dialog box lets you view the type name, actual license code, and status of each license upload. To display detailed information for a specific license, select the appropriate row.

8. Click **OK**.

9. Restart Perceptive Content Server to create the **imagenowforms** user automatically.

Deploy forms

The following steps outline the high-level procedures you perform to deploy forms accessed with a URL in Forms Server. When you use Forms Designer to publish the form, the imagenowforms user automatically receives access to view and submits a form accessible with a URL.

Note If you choose to deploy forms accessed with a URL, you must ensure that your network is configured with the appropriate security.

1. Create the form using Forms Designer. For more information about creating forms, refer to "Create a customized form section" in **Manage > Forms Designer Help**.

2. Publish the form using Forms Designer. For more information about publishing forms, refer to "About publishing forms" in **Manage > Forms Designer Help**.

3. Grant any of the following form privileges to users or groups. For steps to set form privileges, refer to "Set form privileges" in **Manage > Forms Designer Help**.

    - To allow users or groups to view the form and its data, grant the **View** privilege.

    - To allow users or groups to modify the data for that form, grant the **Modify** privilege.

    - To allow users or groups to enter the initial data into that form, grant the **Create** privilege.

    - To allow users or groups to clear all form data, grant the **Delete** privilege.

    For more information on granting privileges, refer to Administrator Help.

4. To access the form, ensure the user or group has access to the document type, folder type, and drawer as necessary.

5. Optional. Associate the form to the workflow queue where the **imagenowforms.xml** is configured to route the forms. To learn more about routes, refer to Workflow Designer Help.

6. Embed the forms URL in your portal or business application. For more information, refer to the Access the Forms URL section.

Access the forms URL

You can create URLs that point to specific form types. Each time a user launches a URL, a new instance of the specified form type appears, allowing a user to view, complete, and submit the form into the Perceptive Content document or folder repository.

The URL contains two sections. The first section points to the Forms Server application domain followed by a question mark. The second section contains the name of the document or folder type being accessed along with an optional collection of name-value pairs separated by ampersands (&).

- **<host>** Supply the host name of the Forms Server application server. You can specify the host name or the IP address.

- **<port>** Supply the port number of the web application. By default, the web application utilizes port 8080.

Below is an example of a URL.

```
http://localhost:8080/psw-content-forms-
server/fs?form=DocFormDemo&barColor=blue&saveText=Submit&success=Your%20form%20was%20s
uccessfully%20submitted%20Thank%20you!
```

Prevent Forms URL from appearing on browser status bar

You can prevent a Forms URL from showing in the status bar of your browser when you integrate Forms with another web application. After performing these steps, when you move your mouse over the Forms link, the status bar displays "Complete Form" rather than the URL associated with the Forms link.

1. Open the web application that uses the forms link.

2. Replace the Forms link that you want to hide with the following HTML code.

```
<a href="http://forms url here" onMouseOver="window.status='Complete Form'; return
true" onMouseOut="window.status=''; return true">View</a>
```

Appendix A: Cross-domain requests

Use a proxy server or Cross-Origin Resource Sharing (CORS) to allow Forms Server to make requests to a server on a domain other than the one where Forms Server is installed.

Web browsers implement a security mechanism known as the "same origin security policy," which only allows a script to interact with a server on the same domain as the origin of the script. If Forms Server attempts to make a cross-domain request, the user's web browser may not allow the request to complete.

If a server provides data to Forms Server, and it is possible that the server is accessed by a different domain than Forms Server, additional configuration steps may be required to install Forms Server. Alternatively, the server providing the data may be configured in a way that allows cross-domain requests in a secure manner.

Cross-domain request options

The options for allowing cross-domain requests include the use of a proxy server or implementation of CORS.

Proxy server

If you configure a proxy server and direct all requests to that server, the proxy server can delegate specified requests to servers on other domains.

CORS

CORS is a mechanism supported by some browsers that allows a script to make requests to a domain other than the domain where the script originated. This option requires that the client request and server response conform to a specific format.

Appendix B: Forms Configuration files

imagenowforms.settings parameters

The following table contains a list of the groups, parameters, and sample values for the imagenowforms.settings file.

Group	Parameter name	Sample value	Description
[PerceptiveContent Forms] Perceptive Content Server settings	host	168.154.21.54	The host name of the Perceptive Content server. You can specify the host name or the IP address.
	port	6000	The port number of the Perceptive Content Server. By default, Perceptive Content utilizes port 6000.
	heartbeat	TRUE FALSE	Specifies whether to send a heartbeat message to the server at a regular interval to maintain the session when no information is sent. By default, this parameter is set to FALSE. If set to TRUE, the Forms session will never time out. The recommended setting is FALSE.
	logontimeout	30	The socket timeout for the logon process in seconds.
	defaulttimeout	30	The socket timeout for communications with Perceptive Content Server in seconds.
	sso	TRUE FALSE	Enables users to log in to Forms Server directly from your SSO provider logon. After setting up single sign-on, you can log in to your SSO provider for direct access to Forms Server.
	sso.key	tokenpassword	When single sign-on is enabled, this setting specifies the token Forms Server uses with Perceptive Content Server to indicate successful authentication.
	sso.method	com.imagenowservlet.auth.RemoteUserAuthenticator	Based on the provider, this setting specifies the type of single sign-on used.
	connection.socket.keep.alive.enabled	TRUE FALSE	Sets the SO_KEEPALIVE socket option on server connections. This enables keep-alive packets for the socket connection. The default is FALSE.
	forms.server.authentication.token	Any valid token	Optional. Specifies an authentication token for Forms Server to use when making connections. If unspecified, legacy agent authentication is used.

Group	Parameter name	Sample value	Description
[PerceptiveContent Forms] Application server settings	log	TRUE FALSE	Specifies whether to log Forms Server errors and warning messages. When the value is TRUE, logging is enabled. When the value is FALSE, logging is disabled. The recommended setting is TRUE.
	logdirectory	C:\logs\	The file directory in which Forms Server stores logged error and warning messages. This path must exist.
	log-unencrypted	TRUE FALSE	Specifies whether the log files are encrypted. The default is FALSE, which means the log files are encrypted. **Note** The common servlet class defaults to TRUE if the setting is blank.
	form-presentation-cache-timeout	60	The socket timeout for caching form files in seconds. The default setting is 60.
	Integration-server-path	http:// 168.154.21.86:8080/int egrationserver	Specifies the path to the Integration Server.
[Forms]	forms.transform.output.method	html xml	Specifies the output method to use for forms transform. html = forms transform uses the html output method. xml = forms transform uses the xml output method. The default is xml.
	forms.fallback.transform.output.method	html xml	Specifies the output method to use if forms transform fails. The fallback only occurs if this value differs from forms.transform.output.method. html = forms transform uses the html output method. xml = forms transform uses the xml output method. The default is xml.
	forms.xsltcache.filesystem.size	0 or any positive number	Specifies the number of compiled xslt revisions the system saves to the file system for each presentation or form. Enabling a file system cache allows the system to use the cache between restarts of the service. Setting the value to 0 disables the file system cache. Any positive number enables the cache. We recommend setting the value to 2 to help performance when xslts are updated. The default is 0.

Group	Parameter name	Sample value	Description
	forms.xsltcache.memory.size	0 or any positive number	Specifies the number of compiled xslt revisions the system keeps in memory for each presentation or form. Enabling the memory cache reduces the file system IO when file system cache is enabled. It can also be used as the only cache. Setting the value to 0 disables the memory cache. Any positive number enables the cache. We recommend setting the value to 2 to help performance when xslts are updated. The default is 2.
	forms.xsltcache.background.enabled	TRUE FALSE	Specifies if the system should compile the xslt forms in the background when a compiled version already exists. This may return old xslt forms when they have been updated. The default is FALSE.
	forms.xsltcache.directory	Any valid directory (relative paths only)	Specifies in which directory the system builds the xslt caches. The cache may create several subdirectories such as xsltcache/forms/{presentation_id}/{unique_id} The default is xsltcache.
[Locale]	Varies, see description	Varies, see description	This optional section enables you to customize the behavior of the application to perform actions such as changing the language to one different from the operating system's locale. For more information on changing the Forms Server language, refer to Change the Forms Server language. This feature is available when the associated Perceptive Content language pack is available.

Change the Forms Server language

You can use this procedure to change the language displayed in Forms Server. By default, Forms Server uses English as the language. However, if the setting is empty, the user's browser setting determines the language. This feature is available when the associated Perceptive Content language package is available.

1. Stop Apache Tomcat.

2. In the **imagenowforms.settings** file, under **Locale**, type a language setting and one of the following supported language codes according to the example below.

   - To change the language to English, type **en**.

   - To change the language to French, type **fr**.

   - To change the language to Spanish, type **es**.

- To change the language to Dutch, type **nl**.

- To change the language to Portuguese, type **pt**.

- To change the language to German, type **de**.

- To change the language to Chinese, type **zh**.

- To change the language to Japanese, type **ja**.

- To change the language to Korean, type **ko**.

    Example   `Language=pt`

3. If you want to refine the language based on a specific country, type a **country** setting and one of the following two-letter ISO 3166-1 country codes for the country you want to specify according to the example below.

    Note This step is required to view **Forms Server** in Dutch, Brazilian Portuguese, and simplified Chinese.

- To change the country code to the Netherlands, type **NL**.

- To change the country code to Brazil, type **BR**.

- To change the country code to simplified Chinese, type **CN**.

    Example   `Country=CN`

4. Save and close the **imagenowforms.settings** file.

5. Start Apache Tomcat.

Set up single sign-on

Single sign-on (SSO) enables users to log in to Forms Server directly from your SSO provider logon. After setting up single sign-on, users can log in to your SSO provider for direct access to Forms Server. To configure settings in the imagenowforms.settings file, complete the following steps.

Important Single sign-on functionality is not intended to be used for authentication purposes. Use this functionality only for populating form fields.

1. Stop Apache Tomcat.

2. Using a text editor, open the **imagenowforms.settings** file.

3. Under the **Perceptive Content Server** settings section, make the following modifications to the specified values.

- Set **sso.method** to the single sign-on method used by your SSO provider. The sso.method value is:

    ```
    com.imagenow.servlet.auth.RemoteUserAuthenticator
    ```

- For a custom SSO solution that uses a named HTTP header to send information, set **sso.method** to **com.imagenow.servlet.auth.HTTPHeaderAuthenticator** and create a new setting, **sso.httpheader.name**. Then set the value to the name of the header that Forms Server uses to read the user name from the SSO provider.

4. Under the **Perceptive Content Server** settings section, set **sso** to **TRUE**.

5. Save and close the **imagenowforms.settings** file.

6. Restart your web server.

7. Optional. Complete the following substeps to populate a form field with SSO user name information.

   1. Navigate to the [*drive:*][*path*]\inserver\etc folder.

   2. Using a text editor, open the **imagenowforms.xml** file.

   3. Enter the **username** data parameter as defined in the Data parameters section of this document.

imagenowforms.xml

The imagenowforms.xml file, located in the [*drive:*]\inserver\etc directory, is used to create and modify form definitions available for use by Forms Server. You can have as many form definitions in the imagenowforms.xml file as you need. However, you must have a form definition in the imagenowforms.xml file for each form you plan to deploy. There are two types of form definitions: the document form definition and the folder form definition. You cannot deploy a single form definition for both a folder type and a document type.

Note The `FormName` element must match the document type or folder type.

A form definition can store two types of attributes for a form: a literal value or dynamic value. Note that when defining a folder form, only the `FolderName` element of the folder form definition can accept a dynamic value. This is the only field on the folder form definition that obeys dynamic fields.

To ensure correct XML validation, you must have the form definition elements in the required order. The required order for document form definition elements is Drawer, Field1, Field2, Field3, Field4, Field5, DataParams, ConfigParams, TransformParams. The required order for folder form definition elements is FolderName, DataParams, ConfigParams.

Sequence is a node that takes literal values for its attributes. You can put a sequence in any of the document key elements. The attributes for the Sequence node are name, width, increment, prefix and suffix. The prefix and suffix attributes are not required. The increment attribute starts with a value of 001, and then increments that value by one. A new sequence is only created if the name does not already exist. Otherwise, the existing sequence is used.

About Data and Configuration parameters

You can enter parameters in the URL field of your browser or in the imagenowforms.xml file. Data parameters allow information to be passed to the form. To enter data parameters in the URL, you must also enter the parameter in the imagenowforms.xml file.

Configuration parameters give you the ability to configure the application that displays the form in the browser. If a parameter is not listed in the XML file, you can modify the parameter's value from the URL. If the parameter is listed in the XML file, you cannot override the value from the URL. However, you can add a urlOverride value in the XML file to change this security feature.

The following example shows a configuration parameter that has an override value added in the imagenowforms.xml file.

```
<ConfigParams>
<Param name="savePosition" value="1" urlOverride="TRUE"/>
</ConfigParams>
```

Attachments

When you add the attachmentsVisible configuration parameter to a form, an attachments button appears on the button bar. When a user adds attachments and submits the form, the attachment pages display as additional pages in Perceptive Content. A number appears on the attachments button to represent the number of attachments a user adds. The Attachments button is only available for forms that are published as document types.

Data parameters

You use data parameters to pre-populate form fields. The imagenowforms.xml file specifies the data parameter name and the field where the information is populated in the form. You can reference this information from the URL or single sign-on functionality. The URL parameter specifies the data parameter name and information to be populated on the form. Refer to the Set up single sign-on section of this document for more information on configuring the SSO functionality.

Data Parameter	Value	Description
<First_Name>	John	An example of a data parameter that uses a URL is a user ID. This parameter specifies the user name to be passed into a form field named First Name. Examples: • `<DataParam name="First_Name" path="/form/student/first_name"/>` • `http://localhost:8080/psw-content-forms-server/fs?form=DocFormTest&First_Name=John`
<username>	Jsmith	An example of a data parameter that uses single sign-on is username. This parameter specifies the user name to be passed into a form field. Example: • `<DataParam name="username" path="/form/student/username"/>`

Configuration parameters

You can add configuration parameters to the imagenowforms.xml file or the embedded URL. The examples that appear in the following table show a format for the configuration parameter in the XML file and in the URL.

Config Parameter	Value	Description
attachmentsActions	Any combination of the following action names separated by a comma: save, reset, print, or attachments	Specifies the action the system performs when a user clicks the button. A button can perform multiple tasks. For example, the Attachments button can launch the attachments dialog and then save the form. The default value is *attachments*. Examples: • `<ConfigParams><ConfigParam name="attachmentsActions" value="attachments,save"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server/fs?form=FormName&attachmentsActions=attachments,save`

Config Parameter	Value	Description
attachmentsBorder Color	CSS color name or an RGB value	Specifies the border color of the Attachments button that appears on the button bar at the bottom of the form. The default value is #C8C8C8. You can supply any CSS color name or RGB value recognized by the browser. For example, white, (255, 255, 255), #FFFFFF. Forms Server recognizes any three decimal values, separated with commas, as an RGB value. For example 255,255,255. Examples: • `<ConfigParams><ConfigParam name="attachmentsBorderColor" value="red"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server /fs?form=FormName&attachmentsBorderColor=red`
attachmentsColor	CSS color name or an RGB value	Specifies the color for the Attachments button that appears on the button bar at the bottom of the form. The default value is #F0F0F0. You can supply any CSS color name or RGB value recognized by the browser. For example, white, (255, 255, 255), #FFFFFF. Forms Server recognizes any three decimal values, separated with commas, as an RGB value. For example 255,255,255. Examples: • `<ConfigParams><ConfigParam name="attachmentsColor" value="red"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server /fs?form=FormName&attachmentsColor=red`
attachmentsMaximum Count	Any non-negative 32-bit integer	Specifies the maximum number of files that can be attached to the form, and must be equal to, or larger than, attachmentsMinimumCount. This does not include any PDFs sent as attachments when the saveTransformAsAttachment parameter is set to true. The default is no limit. Examples: • `<ConfigParams><ConfigParam name="attachmentsMaximumCount" value="5"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server /fs?form=FormName&attachmentsMaximumCount=5`

Config Parameter	Value	Description
attachmentsMaximum Size	Any non-negative 32-bit integer	Specifies the maximum file size for each attachment, in kilobytes. This does not include any PDFs sent as attachments when the saveTransformAsAttachment parameter is set to true. The default is no limit. Examples: • `<ConfigParams><ConfigParam name="attachmentsMaximumSize" value="2147483646"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server /fs?form=FormName&attachmentsMaximumSize=2147483 646`
attachmentsMinimum Count	Any non-negative 32-bit integer	Specifies the minimum number of files that must be attached to submit the form, and must be equal to, or smaller than, attachmentsMaximumCount. This does not include any PDFs sent as attachments when the saveTransformAsAttachment parameter is set to true. The default is no limit. Examples: • `<ConfigParams><ConfigParam name="attachmentsMinimumCount" value="1"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server /fs?form=FormName&attachmentsMinimumCount=1`
attachments Position	An integer between 1 and 4	Specifies the placement of the button in regard to the other buttons. The first button in a row is represented by 1, while 4 represents the last. The default value is 4. Examples: • `<ConfigParams><ConfigParam name="attachmentsPosition" value="1"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server /fs?form=FormName&attachmentsPosition=1`
attachmentsText	Any text	Specifies the text for the Attachments button that appears on the button bar at the bottom of the form. **Note** If you remove all text from the button, an icon appears to represent the attachments action. Examples: • `<ConfigParams><ConfigParam name="attachmentsText" value="AddFile"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server /fs?form=FormName&attachmentsText=Add%20File`

Config Parameter	Value	Description
attachments TextColor	CSS color name or an RGB value	Specifies the color for the text on the Attachments button that appears on the button bar at the bottom of the form. The default value is #F0F0F0. You can supply any CSS color name or RGB value recognized by the browser. For example, white, (255, 255, 255), #FFFFFF. Forms Server recognizes any three decimal values, separated with commas, as an RGB value. For example 255,255,255. Examples: • `<ConfigParams><ConfigParam name="attachmentsTextColor" value="red"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server /fs?form=FormName&attachmentsTextColor=red`
attachmentsTooltip	Any text	Specifies the text that appears when a user places the pointer over the Attachments button. The default value is Attachments. Examples: • `<ConfigParams><ConfigParam name="attachmentsTooltip" value="Add Files"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server /fs?form=FormName&attachmentsTooltip=Add%20Files`
attachmentsTypes	File type extension	Specifies the file extensions permitted for attachments, separated by commas. Do not include a leading period. Examples: • `<ConfigParams><ConfigParam name="attachmentsTypes" value="pdf,jpg,png"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server /fs?form=FormName&attachmentsTypes=pdf,jpg,png`
attachments Visible	TRUE FALSE	Specifies if the Attachments button appears on the button bar at the bottom of the form. The default value is TRUE. Examples: • `<ConfigParams><ConfigParam name="attachmentsVisible" value="FALSE"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server /fs?form=FormName&attachmentsVisible=FALSE`

Config Parameter	Value	Description
barColor	CSS color name or an RGB value	Specifies the color of the button bar that appears at the bottom of the form. The default color is #888888. You can supply any CSS color name or RGB value recognized by the browser. For example, white, (255, 255, 255), #FFFFFF. Forms Server recognizes any three decimal values, separated with commas, as an RGB value. For example 255,255,255. Examples: • `<ConfigParams><ConfigParam name="barColor" value="green"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server /fs?form=FormName&barColor=green`
displayStatus	TRUE FALSE	Determines whether the system displays the form information at the bottom of the screen. The default value is TRUE. Examples: • `<ConfigParams><ConfigParam name="displayStatus" value="FALSE"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server /fs?form=FormName&displayStatus=FALSE`
error	Any text	Specifies the error message displayed when an error occurs submitting a form. **Note** If the error is from the Perceptive Content Server, the custom text from Forms Server does not appear. Examples: • `<ConfigParams><ConfigParam name="error" value="Error%20submitting%20form"/></ConfigParam s>` • `http://localhost:8080/psw-content-forms-server /fs?form=FormName&error=Error%20submitting%20for m`
fullScreenSuccess	TRUE FALSE	Determines if the confirmation message after you submit a form is displayed on a new screen. If the value is FALSE, the message appears on the button bar of the submitted form. The default value is TRUE. Examples: • `<ConfigParams><ConfigParam name="fullScreenSuccess" value="TRUE"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server /fs?form=FormName&fullScreenSuccess=TRUE`

Config Parameter	Value	Description
height	Pixels or percentage	Supplies the height of the web application within the parent web page. You can use either pixels or percentages for the value. For pixels, an example is width=250px. For percentage, an example is height=50%25. The %25 is the URL escape code for %. **Note** Scaling the application size larger can make the buttons inaccessible outside of the viewing area. Examples: • `<ConfigParams><ConfigParam name="height" value="250px"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server /fs?form=FormName&height=250px`
printActions	Any combination of the following action names separated by a comma: save, reset, print, or attachments	Specifies the action the system performs when a user clicks the button. A button can perform multiple tasks. For example, the Print button can print and then save the form. The default value is *print*. Examples: • `<ConfigParams><ConfigParam name="printActions" value="print,save"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server/fs?form=FormName&printActions=print,save`
printBorderColor	CSS color name or an RGB value	Specifies the color for the Print button that appears on the button bar at the bottom of the form. The default value is #C8C8C8. You can supply any CSS color name or RGB value recognized by the browser. For example, white, (255, 255, 255), #FFFFFF. Forms Server recognizes any three decimal values separated with commas as an RGB value. For example 255, 255, 255. Examples: • `<ConfigParams><ConfigParam name="printBorderColor" value="red"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server/fs?form=FormName&printBorderColor=red`
printColor	CSS color name or an RGB value	Specifies the color for the Print button that appears on the button bar at the bottom of the form. The default value is #F0F0F0. You can supply any CSS color name or RGB value recognized by the browser. For example, white, (255, 255, 255), #FFFFFF. Forms Server recognizes any three decimal values separated with commas as an RGB value. For example 255, 255, 255. Examples: • `<ConfigParams><ConfigParam name="printColor" value="red"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server/fs?form=FormName&printColor=red`

Config Parameter	Value	Description
printPosition	Any integer between 1 and 4	Specifies the placement of the button in regard to the other buttons. The first button in a row is represented by 1, while 4 represents the last. The default value is 3. Examples: • `<ConfigParams><ConfigParam name="printPosition" value="1"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server/fs?form=FormName&printPosition=1`
printText	Any text	Specifies the text for the Print button that appears on the button bar at the bottom of the form. **Note** If you remove all text from the button, an icon appears to represent the print action. Examples: • `< ConfigParams><ConfigParam name="printText" value="PrintForm"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server/fs?form=FormName&printText=Print%20Form`
printTextColor	CSS color name or an RGB value	Specifies the color for the text on the Print button that appears on the button bar at the bottom of the form. The default value is black. You can supply any CSS color name or RGB value recognized by the browser. For example, white, (255, 255, 255), #FFFFFF. Forms Server recognizes any three decimal values separated with commas as an RGB value. For example 255, 255, 255. Examples: • `<ConfigParams><ConfigParam name="printTextColor" value="red"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server/fs?form=FormName&printTextColor=red`
printTooltip	Any text	Specifies the text that appears when a user places the pointer over the Print button. The default value is Print Form. Examples: • `<ConfigParams><ConfigParam name="printTooltip" value="Print"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server/fs?form=FormName&printTooltip=Print`

Config Parameter	Value	Description
printTransform	TRUE FALSE	Determines whether the system generates and opens a PDF in a new browser window when a user initiates a print action. **Notes** • If the PDF file does not generate successfully, the Print dialog box appears in order to complete the print action. • PDF files display in a new browser window. To view, you must disable pop-up blockers. Examples: • `<ConfigParams><ConfigParam name="printTransform" value="TRUE"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server/fs?form=FormName&printTransform=TRUE`
printVisible	TRUE FALSE	Specifies if the Print button appears on the button bar at the bottom of the form. The default value is TRUE. Examples: • `<ConfigParams><ConfigParam name="printVisible" value="FALSE"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server/fs?form=FormName&printVisible=FALSE`
prompt	Any text	Specifies the text in the message that displays when a user attempts to close the window without saving changes to a form. **Note** The promptOnClose parameter must be set to TRUE in order for the prompt parameter's text to display. The default message is "The form might contain unsaved data." Examples: • `<ConfigParams><ConfigParam name="prompt" value="Save%20form%20now?"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server/fs?form=FormName&prompt=Save%20form%20now?`
promptOnClose	TRUE FALSE	Determines whether the message from the prompt parameter appears when a user attempts to close the window without saving. The default value is FALSE. Examples: • `<ConfigParams><ConfigParam name="promptOnClose" value="TRUE"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server/fs?form=FormName&promptOnClose=TRUE`

Config Parameter	Value	Description
redirectUrl	Any valid URL	After a user clicks the button to save the form, this parameter specifies a URL address to redirect the user. Save must be the last action configured on the button. **Note** The URL must include the "http://" or "https://" preface. You must use encodings to escape special characters. Examples: • `<ConfigParams><ConfigParam name= "redirectUrl" value="http://www.perceptivesoftware.com"/> </ConfigParams>` • `http://localhost:8080/psw-content-forms-server/fs?form=FormName&redirectUrl=http://www.perceptivesoftware.com`
resetActions	Any combination of the following action names separated by a comma: save, reset, print, or attachments	Specifies the action the system performs when a user clicks the button. A button can perform multiple tasks. For example, the Reset button can save and then reset the form. The default value is *reset*. Examples: • `<ConfigParams><ConfigParam name="resetActions" value="save,reset"/></ConfigParams>` • `http://localhost:8080/psw-conent-forms-server/fs?form=FormName&resetActions=save,reset`
resetBorderColor	CSS color name or an RGB value	Specifies the border color for the Reset button that appears on the button bar at the bottom of the form. The default value is #C8C8C8. You can supply any CSS color name or RGB value recognized by the browser. For example, white, (255, 255, 255), #FFFFFF. Forms Server recognizes any three decimal values separated with commas as an RGB value. For example 255, 255, 255. Examples: • `<ConfigParams><ConfigParam name="resetBorderColor" value="red"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server/fs?form=FormName&resetBorderColor=red`

Config Parameter	Value	Description
resetColor	CSS color name or an RGB value	Specifies the color for the Reset button that appears on the button bar at the bottom of the form. The default value is #F0F0F0.
		You can supply any CSS color name or RGB value recognized by the browser. For example, white, (255, 255, 255), #FFFFFF.
		Forms Server recognizes any three decimal values separated with commas as an RGB value. For example 255, 255, 255.
		Examples:
		• `<ConfigParams><ConfigParam name="resetColor" value="red"/></ConfigParams>`
		• `http://localhost:8080/psw-content-forms-server/fs?form=FormName&resetColor=red`
resetPosition	Any integer between 1 and 4	Specifies the placement of the button in regard to the other buttons. The first button in a row is represented by 1, while 4 represents the last. The default value is 2.
		Examples:
		• `<ConfigParams><ConfigParam name="resetPosition" value="1"/></ConfigParams>`
		• `http://localhost:8080/psw-content-forms-server/fs?form=FormName&resetPosition=1`
resetText	Any text	Specifies the text for the Reset button that appears on the button bar at the bottom of the form.
		Note If you remove all text from the button, an icon appears to represent the reset action.
		Examples:
		• `<ConfigParams><ConfigParam name="resetText" value="Clear"/></ConfigParams>`
		• `http://localhost:8080/psw-content-forms-server/fs?form=FormName&resetText=Clear`
resetTextColor	CSS color name or an RGB value	Specifies the color for the text on the Reset button that appears on the button bar at the bottom of the form. The default value is black.
		You can supply any CSS color name or RGB value recognized by the browser. For example, white, (255, 255, 255), #FFFFFF.
		Forms Server recognizes any three decimal values separated with commas as an RGB value. For example 255, 255, 255.
		Examples:
		• `<ConfigParams><ConfigParam name="resetTextColor" value="red"/></ConfigParams>`
		• `http://localhost:8080/psw-content-forms-server/fs?form=FormName&resetTextColor=red`

Config Parameter	Value	Description
resetTooltip	Any text	Specifies the text that appears when a user places the pointer over the Reset button. The default value is Reset Form. Examples: • `<ConfigParams><ConfigParam name="resetTooltip" value="Clear"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server/fs?form=FormName&resetTooltip=Clear`
resetVisible	TRUE FALSE	Specifies if the Reset button appears on the button bar at the bottom of the form. The default value is TRUE. Examples: • `<ConfigParams><ConfigParam name="resetVisible" value="FALSE"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server/fs?form=FormName&resetVisible=FALSE`
saveActions	Any combination of the following action names separated by a comma: save, reset, print, or attachments	Specifies the action the system performs when a user clicks the button. A button can perform multiple tasks. For example, the Save button can print and then save the form. The default value is *save*. Examples: • `<ConfigParams><ConfigParam name="saveActions" value="print,save"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server/fs?form=FormName&saveActions=print,save`
saveBorderColor	CSS color name or an RGB value	Specifies the color for the Save button that appears on the button bar at the bottom of the form. The default value is #C8C8C8. You can supply any CSS color name or RGB value recognized by the browser. For example, white, (255, 255, 255), #FFFFFF. Forms Server recognizes any three decimal values, separated with commas, as an RGB value. For example, 255,255,255. Examples: • `<ConfigParams><ConfigParam name="saveBorderColor" value="red"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server/fs?form=FormName&saveBorderColor=red`

Config Parameter	Value	Description
saveColor	CSS color name or an RGB value	Specifies the color for the Save button that appears on the button bar at the bottom of the form. The default value is #F0F0F0. You can supply any CSS color name or RGB value recognized by the browser. For example, white, (255, 255, 255), #FFFFFF. Forms Server recognizes any three decimal values, separated with commas, as an RGB value. For example, 255,255,255. Examples: • `<ConfigParams><ConfigParam name="saveColor" value="red"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server /fs?form=FormName&saveColor=red`
savePosition	An integer between 1 and 4	Specifies the placement of the button in regard to the other buttons. The first button in a row is represented by 1, while 4 represents the last. The default value is 1. Examples: • `<ConfigParams><ConfigParam name="savePosition" value="2"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server /fs?form=FormName&savePosition=2`
saveText	Any text	Specifies the text for the Save button that appears on the button bar at the bottom of the form. **Note** If you remove all text from the button, an icon appears to represent the save action. Examples: • `<ConfigParams><ConfigParam name="saveText" value="SaveForm"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server /fs?form=FormName&saveText=Save%20Form`
saveText Color	CSS color name or an RGB value	Specifies the color for the text on the Save button that appears on the button bar at the bottom of the form. The default value is black. You can supply any CSS color name or RGB value recognized by the browser. For example, white, (255, 255, 255), #FFFFFF. Forms Server recognizes any three decimal values, separated with commas, as an RGB value. For example 255,255,255. Examples: • `<ConfigParams><ConfigParam name="saveTextColor" value="red"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server /fs?form=FormName&saveTextColor=red`

Config Parameter	Value	Description
saveTooltip	Any text	Specifies the text that appears when a user places the pointer over the Save button. The default value is Save Form. Examples: • `<ConfigParams><ConfigParam name="saveTooltip" value="save and submit"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server /fs?form=FormName&saveTooltip=save%20and%20submit`
saveTransformAsAttachment	TRUE FALSE	Determines whether the system sends a PDF as an attachment to Perceptive Content when a user performs a save action. Examples: • `<ConfigParams><ConfigParam name="saveTransformAsAttachment" value="TRUE"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server/fs?form=FormName&saveTransformAsAttachment=TRUE`
saveVisible	TRUE FALSE	Specifies if the Save button appears on the button bar at the bottom of the form. The default value is TRUE. Examples: • `<ConfigParams><ConfigParam name="saveVisible" value="FALSE"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server /fs?form=FormName&saveVisible=FALSE`
success	Any text	Specifies the text message that displays when a form is successfully submitted. Examples: • `<ConfigParams><ConfigParam name="success" value="Success!"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server /fs?form=FormName&sucess=Success!`

Config Parameter	Value	Description
width	Pixels or percentage	Specifies the width of the web application within the parent web page. You can use either pixels or percentages for the value. For pixels, an example is width=250px. For percentage, an example is width=50%25. The %25 is the URL escape code for %. **Note** Scaling the application size larger can make the buttons inaccessible outside of the viewing area. Examples: • `<ConfigParams><ConfigParam name="width" value="250px"/></ConfigParams>` • `http://localhost:8080/psw-content-forms-server /fs?form=FormName&width=250px`

About Transform parameters

Adding the transform parameter to the imagenowforms.xml gives you the ability to generate a PDF from the form data. You can choose to generate a PDF when a user performs a print action, a save action, or both.

Note You must add either the printTransform or saveTransformAsAttachment configuration parameter to add a transform parameter.

Transform Parameters

Transform Parameter	Value	Description
transformProvider	reader xsl-fo	**reader** – Sends the form data to an existing PDF reader template. **xsl-fo** – Leverages an existing template that specifies customized formatting and layout details. **Note** When you choose to use reader or xsl-fo providers, you are responsible for creating the templates that Perceptive Content will leverage.
transformTemplate	The file name relative to the web application root.	The file name of the template the system uses to generate formatting and style of the PDF file. **Note** The xsl-fo and reader transformProvider parameters require a template.

Form Definition examples

Document Form Definition example

The following data definition example shows a form that is configured as a document. In this form, the Attachments button appears, but you can add parameters in the URL to remove it.

```
<FormMap xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
              xsi:noNamespaceSchemaLocation="http://www.imagenow.com/imagenowforms/">

    <DocumentForm>
    <FormName>DocFormTest</FormName>
    <QueueName/>
    <Drawer>default</Drawer>
    <Field1 isPath="true">/InvoiceCoding/TaxCode</Field1>
    <Field2 isUnique="true"></Field2>
    <Field3/>
    <Field4/>
    <Field5/>
    <DataParams/>
       <ConfigParams>
       <ConfigParam name="attachmentsVisible" urlOverride="TRUE"
value="true"/>
             <ConfigParam name="barColor" value="green"/>
       </ConfigParams>
  </DocumentForm>

</FormMap>
```

Folder Form Definition example

For folder types, the QueueName and FolderName elements are required. The QueueName element must match a workflow queue. The FolderName element requires a dynamic value. In this form, you customized the text on the Reset button in the imagenowforms.xml file, but you can add parameters to the URL to change the text.

```
 <FormMap xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:noNamespaceSchemaLocation="http://www.imagenow.com/imagenowforms/">

 <FolderForm>
   <FormName>FolderFormTest</FormName>
   <QueueName>Begin</QueueName>
   <FolderName isUnique="true"/>
   <DataParams/>
   <ConfigParams>
      <ConfigParam name="resetText" value="Clear Form" urlOverride="TRUE"/>
   </ConfigParams>
 </FolderForm>

</FormMap>
```

Transform Parameter examples

The following examples show document type forms that utilize each type of transform parameter.

Reader

```
<DocumentForm>

    <FormName>HR</FormName>

    <QueueName>HR Generalist 1</QueueName>

    <Drawer>HR</Drawer>

    <Field1 isPath="true">/form/_301YT69_0000PW72F000002_</Field1>

    <Field2
isPath="true">/form/_301YT69_0000PK72F000001_/_301YT69_0000PH72F000005_</Field2>
```

```
    <Field3 isPath="true">/form/_301YT69_0000PP72F000002_</Field3>

    <Field4 isTimeStamp="true"/>

    <Field5/>

    <DataParams>

      <DataParam name="username" path="/form/_301YT69_0000PW72F000002_"/>

    </DataParams>

    <ConfigParams>

        <ConfigParam name="printTransform" urlOverride="false" value="true"/>

        <ConfigParam name="saveTransformAsAttachment" urlOverride="false"
value="true"/>

    </ConfigParams>

  <TransformParams transformProvider="reader" transformInput="xml"
transformTemplate="c:\inserver\temp\pdf\enabled_form.pdf">

            <TransformParam name="form1[0].#subform[0].Body[0].Comments[0]"
value="/form/_ta_301YT69_0000PM72F000001_" />

            <TransformParam name="form1[0].#subform[0].Body[0].EmployeeName[0]"
value="/form/_301YT69_0000PK72F000001_/_301YT69_0000PH72F000005_" />

      </TransformParams>

 </DocumentForm>
```

XSL-FO

```
<DocumentForm>

    <FormName>HR</FormName>

    <QueueName>HR Generalist 1</QueueName>

    <Drawer>HR</Drawer>

    <Field1 isPath="true">/form/_301YT69_0000PW72F000002_</Field1>

    <Field2
isPath="true">/form/_301YT69_0000PK72F000001_/_301YT69_0000PH72F000005_</Field2 >

    <Field3 isPath="true">/form/_301YT69_0000PP72F000002_</Field3>

    <Field4 isTimeStamp="true"/>

    <Field5/>

    <DataParams>

      <DataParam name="username" path="/form/_301YT69_0000PW72F000002_"/>

    </DataParams>

    <ConfigParams>

        <ConfigParam name="printTransform" urlOverride="false" value="true"/>

        <ConfigParam name="saveTransformAsAttachment" urlOverride="false"
value="true"/>

    </ConfigParams>

  <TransformParams transformProvider="xsl-fo" transformInput="xml"
transformTemplate="c:\inserver\temp\pdf\ApacheFOP.xsl">
```

```
        </TransformParams>
</DocumentForm>
```