eForms Forms Server Installation and Setup Guide

Environments: AIX-UX, HP-UX, Linux, and Sun Solaris

Web Application Server: Tomcat

Version: 6.6.x

Written by: Product Documentation, R&D
Date: July 2012

ImageNow and CaptureNow are registered trademarks of Perceptive Software, Inc. All other products produced by Perceptive Software, Inc., including WebNow, are Perceptive Software trademarks. All other brands and product names mentioned in this document are trademarks or registered trademarks of their respective owners.

Copyright© 2012 Perceptive Software, Inc. All rights reserved.

Table of Contents

About Forms Server Installation **3**
Install Forms Server on UNIX using Tomcat 3
Download eForms *3*
Deploy Forms Server *4*
Set up SSL *4*
Configure the Forms Server environment *4*
Install Sun JRE on clients 5
Automatically install Sun JRE through Enterprise Application Deployment *5*
Manually install Sun JRE *5*
About Forms Server Configuration **6**
Sharing web applications on the same web application server 6
About web application installation *6*
Configure your web application server to run multiple web applications *6*
Test Forms Server 6
Appendix: Forms Server Configuration Files **7**
web.xml parameters 7
Imagenowforms.settings parameters 11
Set up single sign-on 12
imagenowforms.xml 13
About Data and Configuration Parameters 14
Attachments *14*
Data parameters *14*
Configuration parameters *15*
About Transform Parameters 28
Transform Parameters *28*
Document Form Definition Example *29*
Project Form Definition Example *29*
Index **30**

About Forms Server Installation

Forms Server is a thin-client product that runs on a web application server. For product technical specifications and system requirements, go to www.perceptivesoftware.com, click **Customer Portal**, and then log in. The *Technical Specifications* document for this product version is available on the Product Documentation tab.

The following steps outline the high-level procedures required to install and configure Forms Server.

1. **Install and test ImageNow Server and at least one ImageNow Client**. Do this prior to installing the web application server. You use the ImageNow Management Console to set up user authentication for Forms Server users. For information on installing the ImageNow Server and Client, refer to the *ImageNow Installation Guide*.
2. **Download the individual imagenowforms.war file** appropriate for your web application server and operating system. The WAR.war file contains the form application files. You can download the files from the Perceptive Software Customer Portal at www.perceptivesoftware.com.
3. **Copy the INDesJNI65 file** to enable Forms Server communications and security. Forms Server uses Data Encryption Standard (DES) encryption to pass data between the web application server and the ImageNow Server, and uses a built-in data encryption method to pass data between the client browser and web application server. You must download and copy the libINDesJni65.so or libINDesJni65.sl file that is appropriate for your UNIX environment.
4. **Install, start, and test** your web application server.

   Note Apache Tomcat is the only web application server supported by Forms Server.
5. **Extract** the WAR file from the zipped file.
6. **Edit the imagenowforms.settings file** to configure the Forms Server environment. The Forms Server installation wizard performs this step for some application servers.
7. **Install Sun JRE** on all client computers. See the *Product Technical Specifications* document for supported JREs.

Install Forms Server on UNIX using Tomcat

To install Forms Server on a UNIX environment using Tomcat, you must perform the following tasks:

- Download Forms Server.
- Configure the Forms Server environment.

In the Forms Server installation instructions, environment variables represent the default installation path to a product because the actual path can vary. For example, $TOMCAT_HOME represents the default installation path to Tomcat. When you encounter an environment variable, replace the variable with the default installation path to that product.

Download eForms

1. Go to the Perceptive Software website at www.perceptivesoftware.com, click **Customer Portal**, enter your user name and password, and then click **Downloads**.
2. In the **Product Downloads** page, download the following files to a temporary directory on your computer:
   - The product installer
   - The product readme
3. Review the product readme.

4. Unzip and untar the files. Note that the tar contains WAR files for all web application servers. You can remove the WAR files that do not apply to your web application server.

   Note Apache Tomcat is the only web application server supported by Forms Server.

5. Move the **tomcat-imagenowforms.war** file to the **[*path*]/${TOMCAT_HOME}/webapps/imagenowforms** directory. Create the imagenowforms folder if it does not exist.

   Note This directory is recommended, but not required. If you want to deploy the imagenowforms.war file in another directory, move the file to the desired directory.

6. Consult the table below to determine the JNI file you need and the library path environment variable that stores the location for the libINDesJNI65 file for your environment and then perform the following tasks:

   1. Locate the appropriate SL or SO file from the tar file you downloaded, and move that file in the path referenced by the environment variable listed as the Library path in the following table.
   2. If the file name for your environment is different than the name shown below, rename the file to match the File name shown in the following table.

Environment	Library Path	File Name
Solaris & Linux	LD_LIBRARY_PATH	libINDesJni65.so
AIX	LIBPATH	libINDesJni65.so
HP-UX	SHLIB_PATH	libINDesJni65.sl

Deploy Forms Server

- Extract the tomcat-imagenowforms.war file into the **[*path*]/${TOMCAT_HOME}/webapps/imagenowforms** directory.

Set up SSL

If you want to configure SSL to use with Forms Server, you must do so on your Tomcat web application server. SSL configuration cannot be completed within Forms Server. For more information, refer to the Tomcat help documentation on the Apache website.

Configure the Forms Server environment

After deploying the imagenowforms.war file, you must make some changes to the **imagenowforms.settings** configuration file.

1. Navigate to the **imagenowforms/WEB-INF** directory.
2. Using a text editor, open the **imagenowforms.settings** file.
3. Configure the following parameters:
   1. In **host**, enter the host name or IP address of the web application server.
   2. In **port**, enter the port number of the ImageNow Server.
4. Save and then close the file.

5. Navigate to the **tomcat/bin** directory, open the **startup.sh** file in a text editor and then type the following lines in the file before starting Tomcat.

```
Replace <location of libINDesJNI65 file> with the path to this file on
your system:

LD_LIBRARY_PATH=<location of libINDesJNI65 file>:$LD_LIBRARY_PATH

export LD_LIBRARY_PATH
```

6. Start Tomcat using **./startup.sh**, located in **tomcat/bin**.

Install Sun JRE on clients

Forms Server is a Java-based application. Clients running Forms Server require Sun Microsystems Java Runtime Environment (JRE). Regardless of which the application server you use, the Sun JRE must exist on every client computer that accesses Forms Server. The JRE provides the Java APIs, Java virtual machine, and other components needed to run applets and applications written in the Java programming language. Because the version of Sun JRE you need to install depends on the client's environment, refer to the *Product Technical Specifications* document for JRE version information.

Before attempting to download JRE from the Sun website or from an intranet site, make sure that the user has access to the web or your intranet. Also, make sure that the user has permission to save the JRE on the client computer. Use one of the following procedures to install Sun JRE on your client computers.

Automatically install Sun JRE through Enterprise Application Deployment

You might want to push the Sun JRE out to all users at once. You can do this using SMS, Active Directory, or other similar "push" software. The following steps do not describe how to use these types of software. Instead, these steps provide general information you use to push the software.

1. Download Sun JRE from http://java.sun.com/javase/downloads/index.jsp. Make sure you download the appropriate operating system version. Make sure you download the "Offline Installation." For example, for Windows, download Windows Offline Installation, Multi-language (jre-1_6_0_01-windows-i586-p.exe, 15.08 MB).

   Note If you plan to install 64-bit Forms Server, download the 64-bit Sun JRE install package.

2. After you download the Sun JRE, incorporate the JRE installer with your company's application deployment software. We recommend that you install the JRE in "Silent" mode. The following links describe the options for a silent or unattended installation.

   For information on Sun JRE silent installation, go to: http://java.sun.com/javase/downloads/index.jsp.

Manually install Sun JRE

If you do not want to install Sun JRE automatically, you can install JRE manually.

1. Download Sun JRE from http://java.sun.com/javase/downloads/index.jsp. Download the appropriate version for your operating system. Make sure you download the "Offline Installation."
2. Run the JRE installer and follow the installation instructions. During installation, we recommend that you select **Typical**.
3. After the JRE installation finishes, Forms Server automatically launches.

About Forms Server Configuration

Parameters in the imagenowforms.settings file allow you to configure your individual Forms Server implementation without shutting down the web application server. Refer to the “imagenowforms.settings parameters” section of this document.

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

Sharing web applications on the same web application server

You can configure your web application server to run multiple web applications on the same server. For example, you can run WebNow and Forms Server on the same Tomcat server. Any web application that uses the INDesJNI65.dll file must share the encryption.jar file to run an additional web application on the same web application server.

About web application installation

When you install multiple web applications using the same web application server, you store each web application in a directory on the web application server. The default file location varies for each web application server. Examples are outlined below.

Web Application Server	Sample Web Application	Default Installed Directory
Apache Tomcat	WebNow	[*path*]/${TOMCAT_HOME}/webapps/webnow
Apache Tomcat	Forms Server	[*path*]/${TOMCAT_HOME}/webapps/imagenowforms

Configure your web application server to run multiple web applications

1. Stop the web application server and then close all Internet browsers.
2. Navigate to the **[*path*]/${TOMCAT_HOME}/webapps/imagenowforms/WEB-INF** directory.
3. Open the **WEB-INF** directory for the **imagenowforms** web application.
4. Open the **lib** directory and delete the **encryption.jar** file.
5. Open the **WEB-INF** directory for the **webnow** web application.
6. Open the **lib** directory and cut the **encryption.jar** file.
7. Navigate to the shared **lib** directory for the web application server and paste the **encryption.jar** file.

Test Forms Server

1. Using Form Designer, create a test form. Refer to Form Designer Help for more information.
2. Start your web application server.
3. Open your Internet browser.
4. In the **Address** box, type **http://<*machinename*>:8080/imagenowforms/fs?form=<*FormName*>**.

Appendix: Forms Server Configuration Files

This section outlines the configuration files associated with Forms Server.

web.xml parameters

The following table lists the parameters and sample values for the web.xml file. You must restart the web application server for changes to the web.xml file to take place.

Group	Parameter Name	Sample Value	Description
Context parameters	native-viewer	TRUE FALSE	When set to **true**, you can open.pdf documents inside WebNow Viewer. When set to false, these files open in a separate window native to their applications. The default is **false**. When set to **true**, the download size of WebNow increases by 400k. If you are not using Adobe Acrobat Reader documents, we recommend that you keep this set to **false**.

Group	Parameter Name	Sample Value	Description

Context parameters
(continued)
	java-plugin-codebase	• **JRE version 1.5.0_xx**
		`<context-param>`
		`<param-name>java-plugin-codebase</param-name>`
		`<param-value>`**`http://java.sun.com/update/1.5.0/jinstall-1_5_0-windows-i586.cab#Version=1,5`**`</param-value>`
		`</context-param>`
		• **JRE version 1.6.0_xx**
		`<context-param>`
		`<param-name>java-plugin-codebase</param-name>`
		`<param-value>`**`http://java.sun.com/products/plugin/autodl/jinstall-1_6_0_10-windows-i586.cab#Version=1,6,0,10`**`</param-value>`
		`</context-param>`
			The java-plugin-codebase determines the installation directory from which you install the Sun JRE. By default, it installs from Sun's website. However, we suggest you change this setting to download from your internal web server. Doing so provides quicker download times and avoids network proxy issues. For further information, see the "Install Sun JRE" section.
			The combination of java-classid and java-version determines which Sun JRE is downloaded (if not already installed) and used by WebNow. Both parameters must be in sync.

Group	Parameter Name	Sample Value	Description

Context parameters
(continued)
	java-classid	• **JRE version 1.5.0_xx**
		`<context-param>`
		`<param-name>java-classid</param-name>`
		`<param-value>`**`8AD9C840-044E-11D1-B3E9-00805F499D93`**
		`</param-value>`
		`</context-param>`
		• **JRE version 1.6.0_xx**
		`<context-param>`
		`<param-name>java-classid</param-name>`
		`<param-value>`**`CAFEEFAC-0014-0002-0007-ABCDEFFEDCBA`**
		`</param-value>`
		`</context-param>`
			The java-plugin-codebase determines the installation directory from which you install the Sun JRE. By default, it installs from Sun's website. However, we suggest you change this setting to download from your internal web server. Doing so provides quicker download times and avoids network proxy issues. For further information, see the "Install Sun JRE" section.
			The combination of java-classid and java-version determines which Sun JRE is downloaded (if not already installed) and used by WebNow. Both parameters must be in sync.

	java-version	• **JRE version 1.5.0_xx**
		`<context-param>`
		`<param-name>java-version</param-name>`
		`<param-value>`**`1.5`**
		`</param-value>`
		`</context-param>`
		• **JRE version 1.6.0_xx**
		`<context-param>`
		`<param-name>java-version</param-name>`
		`<param-value>`**`1.6.0_10`**
		`</param-value>`
		`</context-param>`	

Servlet and Servlet mapping	servlet-name	com.imagenow.webnow.rss	The web application server uses this parameter. It should not be changed.

Group	Parameter Name	Sample Value	Description
	servlet-class	com.imagenow.webnow.rss	The web application server uses this parameter. It should not be changed.
Servlet and Servlet mapping (continued)	url-pattern	/WebNowRSS	The web application server uses this parameter. It should not be changed.
MIME mapping	extension	docx	The file type WebNow opens using the application specified in the mime-mapping parameter. For example: • docx • pptx • xlsx
	mime-type	application/msword	The application WebNow uses to open files of the file type specified in the extension parameter. For example: • application/msword • application/vnd.ms-powerpoint • application/excel

Note If you use Microsoft Office 2007 to view native documents, use Mime Mapping parameters to specify how WebNow opens files. Each extension and mime-mapping parameter pairings must be enclosed in <mime-mapping> and </mime-mapping> parameters, for example:

```
<mime-mapping>
     <extension>xlsx</extension>
     <mime-type>application/excel</mime-type>
</mime-mapping>
```

Imagenowforms.settings parameters

The following table lists the parameters and sample values for the Imagenowforms.settings file.

Group	Parameter name	Sample value	Description
[ImageNow] ImageNow Server settings	host	168.154.21.54	The host name of the ImageNow Forms application server. You can specify the host name or the IP address. The default is localhost.
	port	6000	The port number of ImageNow Server. By default, ImageNow utilizes port 6000.
	heartbeat	TRUE FALSE	Whether or not 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 ImageNow Server in seconds.
	sso	TRUE FALSE	Enables users to log into Forms Server directly from your SSO provider logon. After setting up single sign-on, you can log into your SSO provider for direct access to Forms Server.
	sso.key	tokenpassword	When single sign-on is enabled, this setting specifies the token WebNow uses with ImageNow 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.

Group	Parameter name	Sample value	Description
[ImageNow] Application server settings	log	TRUE FALSE	Whether to log ImageNow Forms 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 ImageNow Forms stores logged error and warning messages. This path must exist.
	log-unencrypted	TRUE FALSE	Whether the log files are encrypted. The default is TRUE, which means the log files are not encrypted.
	log-transformer-warnings	TRUE FALSE	Whether to log warning messages associated with rendering ImageNow Forms. Set this parameter to true when developing a new form. The default is false.
	worksheet-presentation-cache-timeout	60	The socket timeout for caching form files in seconds. The default setting is 60.
AppletParams	Varies, see description	Varies, see description	This optional section enables you to customize the behavior of the Java applet 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 ImageNow language pack is available.

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 into your SSO provider for direct access to Forms Server.

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

1. Complete the following steps to configure settings in the **ImageNowForms.settings** file:
    1. On the Forms Server computer, navigate to the **[*drive:*][*path*]\ImageNowForms\WEB-INF** folder.
    2. Using a text editor, open the **ImageNowForms.settings** file.

3. Under the **ImageNow 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 ImageNow Server settings section, set **sso** to **TRUE**.
5. Save and close the **ImageNowForms.settings** file.
6. Restart your web server.

2. Optional. Complete the following substeps to populate a form field with SSO username information:
   1. Navigate to the **[*drive:*][*path*]\INServer6\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

To locate the imagenowforms.xml file, navigate to the \inserver6\etc directory. Use this file to create form definitions. The imagenowforms.xml file contains some configuration examples. You can have as many form definitions in this file as you need, but you must have at least one form definition within the file for each form you plan to deploy. There are two types of form definitions: the document form definition and the project form definition. You can’t deploy a single form definition for both a project type and a document type. The imagenowforms.xml file contains examples of each form definition.

A form definition can store two types of attributes for a form. You can use either a literal or dynamic value for the attribute. The dynamic value is only valid for project on the project name element.

For example, the attribute isPath is dynamic. You use this attribute with any document key value (Drawer, Folder, Tab, Field 3, Field 4, or Field 5). To use this attribute, set the value to true and then add an XPath for the Folder element value. The isPath attribute is not required. The default value is false. isUnique is another attribute where the value is dynamic. If you set the value to true for this attribute, it generates a unique key, similar to Document IDs. Ensure that the ImageNow Forms group has privileges for the appropriate drawer for each form configured in imagenowforms.xml.

Note: The FormName element must match the document type or project type.

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. If a sequence with that name already exists in ImageNow, that sequence is used and the definition of it and its attributes in this element for the form definition are ignored. In other words, when there are two sequences with the same name, the existing one is used first. If the sequence does not exist elsewhere, a new sequence is created.

You can map parameters in the URL to the form data definition .xml file before the system renders the form.

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 successfully enter data parameters in the URL, you must also enter the parameter in the imagenowforms.xml file.

Configuration parameters give you the ability to manipulate the presentation or behavior of the form in the Forms Viewer applet. 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 imagenowforms.xml:

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

Attachments

When you add the attachmentsVisible configuration parameter to a form, an Attachments button appears in the Forms Viewer applet. When a user adds attachments and submits the form, the attachment pages display as additional pages in ImageNow or WebNow. 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.

Note The maximum size for attachments is 5 MB.

Data parameters

You must add data parameters to the imagenowforms.xml file and the embedded URL. The imagenowforms.xml file specifies the data parameter name and the field where the information is populated in the form. The URL parameter specifies the data parameter name and information to be populated on the form. The examples that appear in the following table show a format for the .xml file and the URL.

Data Parameter	Value	Description
<User>	JSmith	An example of a data parameter is a user ID. This parameter specifies the user name to be passed into a form field titled Student ID. Examples: • `<DataParam name="userID" path="/form/student/studentID"/>` • http://localhost:8080/imagenowforms/fs?form=DocFormTest&userID=JSmith
<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
printText	Any textUses an intro set of high-level steps to lead into 7 level 2 sections.	Specifies the text for the Print button that appears on the applet 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/imagenowforms/fs?form=FormName&printText=Print%20Form
saveText	Any text	Specifies the text for the Save button that appears on the applet 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/imagenowforms/fs?form=FormName&saveText=Save%20Form
resetText	Any text	Specifies the text for the Reset button that appears on the applet 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/imagenowforms/fs?form=FormName&resetText=Clear

Config Parameter	Value	Description
attachments Text	Any text	Specifies the text for the Attachments button that appears on the applet 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/imagenowforms/fs?form=FormName&attachmentsText=Add%20File
width	Pixels or percentage	Specifies the width of the applet 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 applet size larger can make the buttons inaccessible outside of the viewing area. Examples: • `<ConfigParams><ConfigParam name="width" value="250px"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&width=250px
height	Pixels or percentage	Supplies the height of the applet 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 applet size larger can make the buttons inaccessible outside of the viewing area. Examples: • `<ConfigParams><ConfigParam name="height" value="250px"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&height=250px
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/imagenowforms/fs?form=FormName&sucess=Success!

Config Parameter	Value	Description
error	Any text	Specifies the error message that displays when there is an error submitting a form. However, if the error is from the ImageNow Server, the text from the Forms Server text does not appear. Examples: • `<ConfigParams><ConfigParam name="error" value="Error submitting form"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&error=Error%20submitting%20form
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. The default message is “Save form? Click OK to exit without saving”. However, the examples below show the message, “Save form now?" Examples: • `<ConfigParams><ConfigParam name="prompt" value="Save form now?"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&prompt=Save%20form%20now?
barColor	Color name or an RGB value	Specifies the color of the applet button bar that appears at the bottom of the form. The default color is gray. Valid names for the colors are: black, blue, cyan, darkGray, darkGrey, gray, grey, green, lightGray, lightGrey, magenta, orange, pink, red, white, yellow. If you supply an RGB value, separate each color section with commas, for example 255, 255, 255. Examples: • `<ConfigParams><ConfigParam name="barColor" value="green"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&barColor=green
username	imagenowforms	The user name does not have to exist as a parameter. If the user name does exist in the URL, then it must be equal to imagenowforms. If the user name does not exist, it is set to imagenowforms by default.

Config Parameter	Value	Description
saveColor	Color name or an RGB value	Specifies the color for the Save button that appears on the applet button bar at the bottom of the form. The default value is gray. Valid names for the colors are: black, blue, cyan, darkGray, darkGrey, gray, grey, green, lightGray, lightGrey, magenta, orange, pink, red, white, yellow. If you supply an RGB value, separate each color section with commas, for example 255, 255, 255. Examples: • `<ConfigParams><ConfigParam name="saveColor" value="red"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&saveColor=red
printColor	Color name or an RGB value	Specifies the color for the Print button that appears on the applet button bar at the bottom of the form. The default value is gray. Valid names for the colors are: black, blue, cyan, darkGray, darkGrey, gray, grey, green, lightGray, lightGrey, magenta, orange, pink, red, white, yellow. If you supply an RGB value, separate each color section with commas, for example 255, 255, 255. Examples: • `<ConfigParams><ConfigParam name="printColor" value="red"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&printColor=red
resetColor	Color name or an RGB value	Specifies the color for the Reset button that appears on the applet button bar at the bottom of the form. The default value is gray. Valid names for the colors are: black, blue, cyan, darkGray, darkGrey, gray, grey, green, lightGray, lightGrey, magenta, orange, pink, red, white, yellow. If you supply an RGB value, separate each color section with commas, for example 255, 255, 255. Examples: • `<ConfigParams><ConfigParam name="resetColor" value="red"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&resetColor=red

Config Parameter	Value	Description
attachmentsColor	Color name or an RGB value	Specifies the color for the Attachment button that appears on the applet button bar at the bottom of the form. The default value is gray. Valid names for the colors are: black, blue, cyan, darkGray, darkGrey, gray, grey, green, lightGray, lightGrey, magenta, orange, pink, red, white, yellow. If you supply an RGB value, separate each color section with commas, for example 255, 255, 255. Examples: • `<ConfigParams><ConfigParam name="attachmentsColor" value="red"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&attachmentsColor=red
saveTextColor	Color name or an RGB value	Specifies the color for the text on the Save button that appears on the applet button bar at the bottom of the form. The default value is black. Valid names for the colors are: black, blue, cyan, darkGray, darkGrey, gray, grey, green, lightGray, lightGrey, magenta, orange, pink, red, white, yellow. If you supply an RGB value, separate each color section with commas, for example 255, 255, 255. Examples: • `<ConfigParams><ConfigParam name="saveTextColor" value="red"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&saveTextColor=red

Config Parameter	Value	Description
printText Color	Color name or an RGB value	Specifies the color for the text on the Print button that appears on the applet button bar at the bottom of the form. The default value is black. Valid names for the colors are: black, blue, cyan, darkGray, darkGrey, gray, grey, green, lightGray, lightGrey, magenta, orange, pink, red, white, yellow. If you supply an RGB value, separate each color section with commas, for example 255, 255, 255. Examples: • `<ConfigParams><ConfigParam name="printTextColor" value="red"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&printTextColor=red
resetText Color	Color name or an RGB value	Specifies the color for the text on the Reset button that appears on the applet button bar at the bottom of the form. The default value is black. Valid names for the colors are: black, blue, cyan, darkGray, darkGrey, gray, grey, green, lightGray, lightGrey, magenta, orange, pink, red, white, yellow. If you supply an RGB value, separate each color section with commas, for example 255, 255, 255. Examples: • `<ConfigParams><ConfigParam name="resetTextColor" value="red"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&resetTextColor=red

Config Parameter	Value	Description
attachments TextColor	Color name or an RGB value	Specifies the color for the text on the Attachment button that appears on the applet button bar at the bottom of the form. The default value is black. Valid names for the colors are: black, blue, cyan, darkGray, darkGrey, gray, grey, green, lightGray, lightGrey, magenta, orange, pink, red, white, yellow. If you supply an RGB value, separate each color section with commas, for example 255, 255, 255. Examples: • `<ConfigParams><ConfigParam name="attachmentsTextColor" value="red"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&attachmentsTextColor=red
savePosition	An integer between 1 and 4	Specifies the placement of the button in regard to the other buttons. 1 represents the first button in a row and 4 represents the last. The default value is 1. Examples: • `<ConfigParams><ConfigParam name="savePosition" value="2"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&savePosition=2
resetPosition	An integer between 1 and 4	Specifies the placement of the button in regard to the other buttons. 1 represents the first button in a row and 4 represents the last. The default value is 2. Examples: • `<ConfigParams><ConfigParam name="resetPosition" value="1"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&resetPosition=1

Config Parameter	Value	Description
printPosition	An integer between 1 and 4	Specifies the placement of the button in regard to the other buttons. 1 represents the first button in a row and 4 represents the last. The default value is 3. Examples: • `<ConfigParams><ConfigParam name="printPosition" value="1"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&printPosition=1
attachments Position	An integer between 1 and 4	Specifies the placement of the button in regard to the other buttons. 1 represents the first button in a row and 4 represents the last. The default value is 4. Examples: • `<ConfigParams><ConfigParam name="attachmentsPosition" value="1"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&attachmentsPosition=1
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 save and then reset the form. The default value is save. Examples: • `<ConfigParams><ConfigParam name="saveActions" value="print,save"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&saveActions=print,save
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 Save button can save and then reset the form. The default value is reset. Examples: • `<ConfigParams><ConfigParam name="resetActions" value="print,reset"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&resetActions=print,reset

Config Parameter	Value	Description
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 Save button can save and then reset the form. The default value is print. Examples: • `<ConfigParams><ConfigParam name="printActions" value="print,save"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&printActions=print,save
attachments Actions	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 save and then reset the form. The default value is attachments. Examples: • `<ConfigParams><ConfigParam name="attachmentsActions" value="print, attachments"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&attachmentsActions=print,attachments
saveVisible	TRUE FALSE	Specifies whether or not a button appears on the applet button bar at the bottom of the form. The default value is TRUE. Examples: • `<ConfigParams><ConfigParam name="saveVisible" value="FALSE"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&saveVisible=FALSE
resetVisible	TRUE FALSE	Specifies whether or not a button appears on the applet button bar at the bottom of the form. The default value is TRUE. Examples: • `<ConfigParams><ConfigParam name="resetVisible" value="FALSE"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&resetVisible=FALSE

Config Parameter	Value	Description
printVisible	TRUE FALSE	Specifies whether or not a button appears on the applet button bar at the bottom of the form. The default value is TRUE. Examples: • `<ConfigParams><ConfigParam name="printVisible" value="FALSE"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&printVisible=FALSE
attachments Visible	TRUE FALSE	Specifies whether or not a button appears on the applet button bar at the bottom of the form. The default value is TRUE. Examples: • `<ConfigParams><ConfigParam name="attachmentsVisible" value="FALSE"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&attachmentsVisible=FALSE
saveTooltip	Any text	Specifies the text that appears when a user places the pointer over the button. The default value is Save Form. Examples: • `<ConfigParams><ConfigParam name="saveTooltip" value="save and submit"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&saveTooltip=save%20and%20submit
resetTooltip	Any text	Specifies the text that appears when a user places the pointer over the button. The default value is Reset Form. Examples: • `<ConfigParams><ConfigParam name="resetTooltip" value="Clear"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&resetTooltip=Clear

Config Parameter	Value	Description
printTooltip	Any text	Specifies the text that appears when a user places the pointer over the button. The default value is Print Form. Examples: • `<ConfigParams><ConfigParam name="printTooltip" value="Print"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&printTooltip=Print
attachmentsTooltip	Any text	Specifies the text that appears when a user places the pointer over the button. The default value is Attachments (0). Examples: • `<ConfigParams><ConfigParam name="attachmentsTooltip" value="Add Files"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&attachmentsTooltip=Add%20Files
nativeLnF	TRUE FALSE	Determines whether to use the default display for the operating system. True uses the display from the operating system. False uses the display from the Javascript. If you select True, other parameters may vary based on the operating system. The default value is FALSE. Examples: • `<ConfigParams><ConfigParam name="nativeLnF" value="TRUE"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&nativeLnF=TRUE
promptOnClose	TRUE FALSE	Determines whether a message appears when a user attempts to close the window without saving. The default value is TRUE. The following is the default message: “Save form? Click OK to exit without saving.” Examples: • `<ConfigParams><ConfigParam name="promptOnClose" value="TRUE"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&promptOnClose=TRUE

Config Parameter	Value	Description
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 applet button bar of the submitted form. The default value is TRUE. Examples: • `<ConfigParams><ConfigParam name="fullScreenSuccess" value="TRUE"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&fullScreenSuccess=TRUE
promptPage	TRUE FALSE	Determines whether, after a user initiates a print action, the system displays the Page Setup dialog box. The default value is TRUE. Examples: • `<ConfigParams><ConfigParam name="promptPage" value="FALSE"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&promptPage=FALSE
promptPrint	TRUE FALSE	Determines whether the Print dialog box is displayed after a user initiates a print action. The default value is TRUE. Examples: • `<ConfigParams><ConfigParam name="promptPrint" value="FALSE"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&promptPrint=FALSE
displayStatus	TRUE FALSE	Determines whether the system displays the form information in the status bar at the bottom of the screen. The default value is TRUE. Examples: • `<ConfigParams><ConfigParam name="displayStatus" value="FALSE"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&displayStatus=FALSE

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. **Note** If the PDF file does not generate successfully, the Print dialog box appears in order to complete the print action. The default value is FALSE. Examples: • `<ConfigParams><ConfigParam name="printTransform" value="TRUE"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&printTransform=TRUE
saveTransformAsAttachment	TRUE FALSE	Determines whether the system sends a PDF as an attachment to ImageNow or WebNow when a user performs a save action. The default value is FALSE. Examples: • `<ConfigParams><ConfigParam name="saveTransformAsAttachment" value="TRUE"/></ConfigParams>` • http://localhost:8080/imagenowforms/fs?form=FormName&saveTransformAsAttachment=TRUE

About Transform Parameters

Adding the transform parameter to the imagenowforms.xml gives you the ability to generate a PDF from the form HTML or XML 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
transformInput	xml html	**XML** – Use the data from the form as it is specified in the form data definition XML file. **Note** If you choose compact for the transformParameter value, xml is the recommended transformInput value. **HTML** – Use the HTML displayed on the screen. **Note** If you choose full for the transformParameter value, html is the recommended transformInput value.
transformProvider	compact full reader xsl-fo	**compact** – Provides the form data in a basic .pdf file without formatting or color. **full** – Resembles the format that is displayed on the screen in the applet. **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 ImageNow I leverages.
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** Only the xsl-fo and reader transformProvider parameters require a template.

Document Form Definition Example

The following data definition example shows a form that is configured as a document. In this form, the Attachment button appears in the applet, but you can add parameters in the URL to remove it. This form generates a PDF that leverages a customized template when the user initiates a print action.

```
<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>
    <Folder isPath="true">/InvoiceCoding/TaxCode</Folder>
    <Tab isUnique="true"></Tab>
    <Field3/>
    <Field4/>
    <Field5/>
    <DataParams/>
      <ConfigParams>
      <ConfigParam name="attachmentsVisible" urlOverride="TRUE"
value="true"/>
            <ConfigParam name="barColor" value="green"/>
            <ConfigParam name="printTransform" value="true"/>
      </ConfigParams>
      <TransformParams transformProvider="xsl-fo" transformInput="xml"
transformTemplate="template.xsl">
</TransformParams>
  </DocumentForm>

</FormMap>
```

Project Form Definition Example

For project types, The QueueName and ProjectName elements are required. The QueueName element must match a workflow queue. The ProjectName element requires a dynamic value. In this form, the text on the Reset button is customized in the applet, but you can add parameters in the URL to change it.

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

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

  <ProjectForm>
    <FormName>ProjFormTest</FormName>
    <QueueName>Begin</QueueName>
    <ProjectName isUnique="true"/>
    <DataParams/>
    <ConfigParams>
      <ConfigParam name="resetText" value=”Clear Form” urlOverride="TRUE"/>
    </ConfigParams>
  </ProjectForm>

</FormMap>
```

Index

attachment
  parameter 14
configuration parameter 15
data parameter 14
document form definition
  example 29
form definitions
  document 29
  project 29
FormName
  element 13
Forms Server Installation 3
  about 3
  on UNIX using Tomcat 3
imagenowforms.settings 3, 4, 6
ImageNowForms.settings
  parameters 11
imagenowforms.war 3, 4
INDesJNI65 3
isPath 13
isUnique 13
Mime Mapping 10
parameter
  attachment 14
  data 14
parameter, configuration 15
parameters 14
  configuration 14
  transform 28
  URL 15
project form definition
  example 29
sequence node 13
SSL 4
Sun JRE 3, 5
  installing 5
transform
  parameters 28
URL parameters 15
web.xml
  parameters 7