PDF Forms

The PDF Forms plugin can be used to publish PDF Forms through the Application Server. User can **open** the PDF Form, **fill in** the fields and **submit** the PDF Form. The submitted PDF Form is sent to the plugin and the values of the fields are **extracted** from the PDF Form and **stored** in the database.

For the plugin to receive the filled out PDF Form and to extract the data, the following is required:

- The fields need to be uniquely named
- The PDF Form required to have a 'submit' button , which sends the form data back to the plugin.

The PDF Forms plugin is a 100% server side operating plugin and interacts directly with the database for retrieving the form and storing the filled in field values.

Basic Workflow

1. Actual PDF Forms are uniquely stored as a record in the pdf_template table.
2. For each time that an actual PDF Form needs to be filled out a pdf_action record is created
3. Links to open the PDF Form for a specific action are provided to the people that need to fill out the form (for example through an application or through email)
4. The user opens the PDF Form (in a browser), fills in the fields and submits the PDF Form
5. The plugin receives the submitted form, reads the values from the fields and saves those as individual records with the field name and value in the pdf_form_values table, related to the pdf_action record.
6. The user gets redirected to the redirect URL
7. The pdf_action record is marked as closed

Creating PDF Forms

The PDF Forms plugin supports PDF Forms (both FDF and XFA) created using Adobe Acrobat .

> ⚠ The plugin supports XFA forms since Servoy 5.2.5 release.

In order for the PDF Forms plugin to work with a specific PDF Form, it is required that the PDF Form has:

- Uniquely named fields
- A submit button
- A hidden field called 'servoy_pdf_submit_url'
- A hidden field called 'servoy_action_id'

Uniquely named fields
When the user has filled out a form and submits it, the PDf Forms plugin retrieves all the fields and their values and stores them in the database. For each field name, it stores only one value. This means that unless each field is uniquely named, values will get lost.

Submit button
The PDF form needs to contain a submit button to allow the user to post back the filled in form back to the PDF Forms plugin on the Application Server. Using Adobe Acrobat or Adobe LiveCycle, create a submit button on the form. For FDF form add the following JavaScript code to the mouse-up action on the button:

```
var ef = this.getField("servoy_pdf_submit_url");
this.submitForm(ef.value);
```

For XFA form add the following JavaScript code to the preSubmit action on the button:

```
this.resolveNode("#event").submit.target =
event.target.getField("Form[0].Page[0].servoy_pdf_submit_url[0]").value+"?action_id="+event.target.getField
("Form[0].Page[0].servoy_action_id[0]").value;
```

For XFA form the form must be submitted as XML using UTF-8 encoding. Also in this javascript code Form and Page must be replaced with the actual names present in the PDF document.

Automatic submit button injection
The PDF Forms plugin contains logic to automatically insert the submit button into the PDF Form. The automatic insert of the submit button is only supported when the PDF Form is opened by users using Adobe Acrobat, not when opened through Adobe Reader. Also, it is not supported for XFA forms.

To enable automatic insertion of the submit button, the 'skip_placing_submit_button' field on the pdf_template record needs to be set to one ('1'). When set to zero ('0') the button will not be inserted automatically.

Hidden fields
The PDF Form needs to contain two hidden fields, one named 'servoy_pdf_submit_url' and one named 'servoy_action_id'. These fields will be automatically populated by the PDF Forms plugin and are used to retrieve the filled in data back.

For XFA forms, these hidden fields are also used in order to put the form to readonly. This is needed because the XFA form cannot be flattened out-of-the-box, just as the FDF form. So, this javascript must be added in initialize event of the form:

```
if (xfa.resolveNode("Form.Page.servoy_pdf_submit_url").rawValue==null) {xfa.resolveNode ("Form.Page.
submitButton").presence="invisible";Form.access = "nonInteractive";}
```

In this javascript code Form, Page and submiButton must be replaced with the actual names present in the PDF document. Also this code only works for PDF version 9 and later when access property was introduced for form. In order to support earlier versions all fields in form must be parsed and put to nonInteractive (in javascript code).

Accessing the PDF Forms

The PDF Forms plugin exposes two dynamic url's for accessing the PDF Forms:

- <serverUrl>/servoy-service/pdf_forms/pdf_template?template_id={template_id}
  Through this URL, the PDF Forms can be viewed, but filling and submitting the form will not result in storing the data to the database
- <serverUrl>/servoy-service/pdf_forms/pdf_form/load.fdf?action_id={action_id}
  Through this URL, a PDF Form linked to a specific pdf_action record can be opened. When submitted, the filled in form data will be stored in the database, as records in the pdf_form_values table, linked to the specific pdf_action.

pdf_action.action_type

The pdf_template record contains one column that affects if the PDF Form can be submitted or not. If the action_type column is set to VIEW (value=0) the PDf Form will open in read-only mode. The fields will be read-only and the submit button will be disabled. When the action_type column is set to EDIT (value=1) the PDF Form will open in edit-mode, allowing the fields to be edited and the PDf Form to be submitted.

To view PDF Forms in the browser, Adobe Acrobat Reader 7 (or higher) - including the default web browser plugin - must be installed. Acrobat Reader 7 can be downloaded (at no charge) from Adobe: http://www.adobe.com

Redirect url

After submitting the PDF form, the browser will redirect to the redirect_url, if specified. If no redirect_url is specified on the specific action, the PDF Forms plugin will lookup the redirect_url on the pdf_template record linked to the pdf_action. If both the pdf_action and pdf_template do not specify a redirect_url, the browser will be redirected to a default page that displays the text 'Data successfully stored,close this window' and will attempt to close the window.

Automatic pdf_action closing

When the user submits a PDF Form, the related pdf_action record will be marked as closed. When the user would try to re-open a PDF Form that was previously submitted, thus the pdf_action record is marked as closed, instead a page will be shown in the browser, displaying the text 'Security violation, use the pdf system to edit pdfs'.

Re-using data

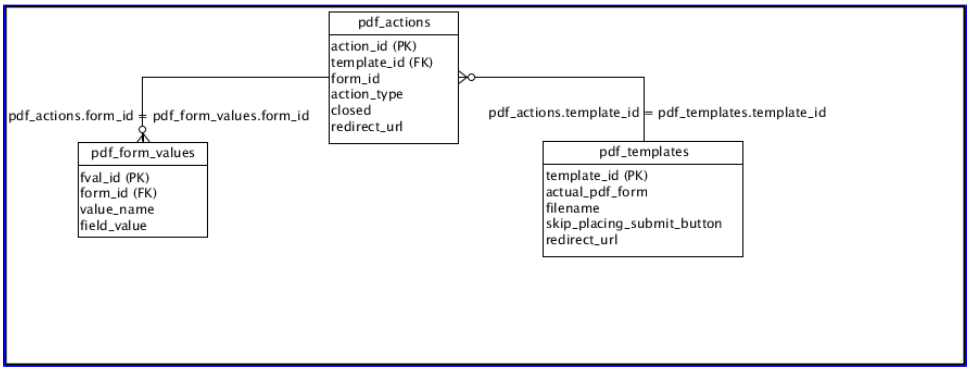
When the PDF Forms plugin opens a PDF Form linked to an pdf_action, it will fill any field in the PDF Form with data that is already stored in the pdf_form_values table, related to the pdf_action record.

As can be seen in the ERD below, the link between the data stored in the pdf_form_values table and the pdf_action table is not done based on the primary key of the pdf_actions table, but on a separate form_id column. This allows for the following:

- Link multiple actions, possibly linked to different pdf_templates to the same set of data
- Provide default values for fields, by pre-populating the pdf_form_values table

Entity Relation Diagram

The ERD below is the minimum required by the plugin to operate. The data model needs to be present in one of the Database Servers. By default, the DataBase Server named 'pdf_forms' will be used, but this can be overwritten using the 'pdf_forms_plugin_servername' setting of the plugin, available on the Admin page under Server Plugin Settings .



pdf_templates
The pdf_templates table stores the the actual PDF Form:

- template_id: the primary key for the table
- actual_pdf_form: a media field containing the actual PDF Form
- filename: a descriptive name (excluding the .pdf extention), used in the URL's the plugin generates
- skip_placing_submit_button: flag to indicate if the plugin should automatically insert a 'submit' button into the PDF Form (see Submit button above)
- redirect_url: the URL where the user is redirected to after submitting the form

pdf_actions

- action_id: the primary key for the table
- template_id: foreign key to the pdf_templates table
- form_id: identifier to link with the pdf_form_values table
- action_type: identifier indication if the PDF form is editable or not. (VIEW=0, EDIT=1)
- closed: identifier indicating if the action is closed. Automatically set when the user submits the form. When set to closed, it overrides the action_type EDIT. (OPEN=0/null,CLOSED=1)
- redirect_url: the URL where the user is redirected to after submitting the form. If not specified, the plugin will lookup and use the redirect_url of the linked pdf_template record

> ⚠ **Additional column required**
>
> Before Servoy 5.1, the plugin also requires the presence of a 'user_id' column, although not used

pdf_form_values

- fval_id: the primary key for the table
- form_id: foreign key to the pdf_forms table
- value_name: the name of the field that was filled
- field_value: the value of the field that was filled

PDF Template location options

When a PDF Form is shown to the user a related PDF Form template is automatically retrieved. This PDF Form Template is by default loaded from the "actual_pdf_form" column from the pdf_templates table.

There are two options to override the location from where the template is retrieved:

1. Override template location for individual PDF Forms
   By appending an extra parameter named "overrideTemplateLocation" to the URL with as value the location of the PDF Form Template, the PDF Form Template will be loaded from the specified location.
   Example: <serverUrl>/servoy-service/pdf_forms/pdf_form/load.fdf?action_id={action_id}&overrideTemplateLocation=<serverUrl>/template.pdf
2. Override template location for all PDF Forms
   The PDF Form Template location can also be overridden on plugin level by setting the 'pdf_forms_plugin_template_location' property on the plugin. This will override the PDF Form location for all PDF Forms. As this setting affects all PDF Forms, the value of the property should only specify the directory location. The name of the specific PDF Template is be taken from the 'filename' column of the 'pdf_templates' table.
   Example: if property is set to <serverUrl> the location will be: <serverUrl>/template.pdf where template.pdf is the template filename

The overrideTemplateLocation can be any location that is accessible from the machine where the Servoy Web Client is running, for example .. /application_server/server/webapps/ROOT/servoy-webclient/..

The order of precedence for PDF Form Template location is:

1. if specified the 'overrideTemplateLocation' parameter in the url is used
2. Otherwise if the 'pdf_forms_plugin_template_location' is set on the plugin that will be used
3. else the default will be taken from the database

Server Property Summary

Server Property Details

pdf_forms_plugin_servername

pdf_forms_plugin_template_location