

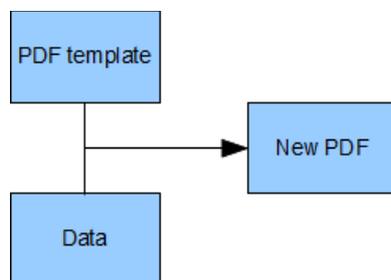
**PDF Exporter
Xpages Custom Control
Documentation**

1	What is this custom control and what it does.....	3
1.1	PDF template.....	3
1.2	How to use Open Office Impress.....	4
2	Technical overview.....	4
3	Installation – how to use the custom control.....	5
4	Custom Control properties.....	6
5	Known limitations.....	8

1 What is this custom control and what it does

With this Xpages custom control you are able to export data to a fully customized PDF document and view it with a browser or with PDF document reader (Adobe Reader, FoxIt reader etc). The source can be a Notes Document or almost what ever.

As a result you get a new PDF document which is a mix of your own PDF template file and your own data. This custom control handles only the data part. This gives you the flexibility to focus only on the data and separate the PDF UI design elsewhere. In really complex PDF documents the design would be a real pain to create “on the fly” e.g. with iText. As a result you get a new copy of the PDF template containing your data.



1.1 PDF template

From the technical view the PDF template is a Adobe Acro Form, a normal PDF document containing just some Acro Fields with unique names (IDs). Really short definition of Acro Form may be that they were originally designed to be like a interactive web forms. Instead of letting users to enter the values in the PDF interactive PDF document, this custom control sets the values by code using the named Acro Fields in the PDF template.

Here is one definition of the Acro Forms:

http://en.wikipedia.org/wiki/Portable_Document_Format#AcroForms

PDF templates (Acro Forms) can be made with several tools:

- Adobe Acrobat Professional
- Foxit Reader Form designer (foxit reader plugin)
- Open Office, libre office (impress)
- Lotus Symphony Presentation
- etc

So you have several tools to choose from. From the list above only the **Open Office Impress** and **Lotus Symphony Presentation** have been tested so far. Mainly the control has been tested with Open Office Impress. With this free WYSIWYG editor

you can design your PDF template by adding your custom design and the Acro Text Fields not forgetting all the other design elements (images, text, background etc).

Best part is that Lotus Notes 8.5. is shipped with one editor with you you can create PDF templates, Lotus Symphony Presentation (optional applications on client installation).

1.2 How to use Open Office Impress

The basic design of the template file is based on a normal Impress slides. There is nothing special in that. The slide(s) can contain any number of design elements, images and other resources. The template can have one or more slides (pages) aswell.

The "magic" is been made with the Text Field Controls which must be added to show any data in the PDF document. To be able to add controls, you need to open the Form Controls toolbar by selecting it from View ==> Toolbars ==> Form Controls menu.

After this you just set the form to design mode and draw the text fields where ever you like on your design.

One important thing is that you must give a **unique** name to each Text Field you add. This name is been used when the data is been pushed by the custom control.

External images uses the Text Field controls also. The Text Field is been used only on positioning and resizing the image. Technically the image is a "floating" element on top of the Text Field.

After you are done with your design, final step is to export the template as a PDF file (File ==> export as PDF).

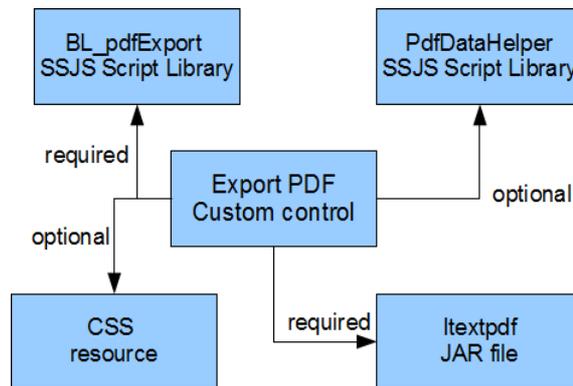
2 Technical overview

Custom Control requires Lotus Notes / Domino 8.5.2 client / server (or newer). So far it has been tested only on 8.5.2. Custom Control does not require any extension libraries.

Custom control works both in Notes Client (8.5.2) and in browser (server) – it's Xpages right? :)

As been told already this custom control is based on the Adobe Acro Forms technique. The actual implementation uses the iText PDF Open Source library (Java) as the Acro Forms implementation. You can download the full binay distribution and view the iText documentation from their web site <http://itextpdf.com/>

This custom control consists of following design elements that are shipped in the example database.



Export PDF Custom Control	The core design element of the system. Required design element.
BL_pdfExport Script Library	SSJS Script Library which contains the business logic of the whole system. Required design element.
PdfDataHelper Script Library	SSJS Script Library which contains some helper functions which can be used if wanted. Optional design element. The Custom Control itself does not require this. Use this in your own Xpages or elsewhere.
Itextpdf jar file	Itextpdf Open Source project JAR file (Java). See this documentation how to import this resource to the project. Tested version is 5.1.0.
CSS resource	Style settings for the button element. Optional design element.

3 Installation – how to use the custom control

Here is an overview of required phases to do before you can use this custom control in your own application. First thing is to copy all the required design elements, see the image and table in previous chapter for details of required and optional elements.

Copy paste is always easy. Copy the required and optional design elements to your application. The trickiest part is to set the JAR resource to you application build path. Here are quick instructions how and where to put it so that the custom control works properly:

- open Domino Designer (DDE)

- select Window ==> Open Perspective ==> Java
- find your project from the list and open the file structure
- open WebContent\WEB-INF folder and create a new "lib" subfolder under it.
 - right click and select New ==> Folder and enter the folder name (lib)
- select your "lib" folder, right click and select "Import" ==> "File system"
- find the itextpdf JAR file from your file system and import it to the lib folder
 - See the zip file (or download from itextpdf web site)
- select "Build Path" ==> "Configure Build Path" from right click menu
- select "libraries" tab from the dialog and click "add JARs..." button
- select your project and browse to your "lib" folder (/webContent/WEB-INF/lib) and select the JAR file you just imported.
- OK
- Window ==> Open Perspective ==> Domino Designer to go back to normal look of the IDE.

Now you have imported a JAR file to your project and you are ready to go. Select.

There's one known issue with JavaScript library loading. This custom control code is based on a Script Library and it is been loaded normally by the Custom Control. However the same JavaScript code is been used already by the Xpage where this custom control has been added. This causes a JavaScript error since the Xpage is not able to "see" the library and it's "pdfData" class yet (the library is been read afterwards by the custom control). This issue can be fixed by adding the "BL_pdfExport" script library as a resource to the xpage aswell. See the example page. This issue has been reported in IBM web site <https://www-304.ibm.com/support/docview.wss?uid=swg21380793>

I'm working on with this issue. There for the API of this custom control might change a little bit in the future.

4 Custom Control properties

After inserting the custom control to your xpage, you need to set some of the following custom control properties:

PdfData	<p>This property holds all the data (field values) that you like to push to the PDF document.</p> <p>Example</p> <pre>var pdfdata = new pdfData(); pdfdata.addNewData("name", "Jane Doe"); pdfdata.addNewData("id", "123456");</pre>
---------	--

	<pre>pdfdata.addNewData("email", "my@email.com"); pdfdata.addNewData("checkbox1", "Off"); // Values: Yes Off pdfdata.addNewData("RadioGroup1", "1"); pdfdata.addNewData("birthday", formatDate(new java.util.Date(), "dd.MM.yyyy")); // add image(s) pdfdata.addNewImage("image2", "c:/temp/kuva.jpg", true); pdfdata.addNewImage("image1", "http://itextpdf.com/img/iia.jpg", false); return pdfdata;</pre> <p>The “pdfData” is a JavaScript object which holds the Text Field values and Image data.</p>
PdfRenderType	<p>ComboBox choice of the rendering type</p> <ul style="list-style-type: none"> - File - OutputStream <p>File – Creates a File in to the file system OutputStream – Creates the PDF document and renders it to the Java Servlet output stream.</p>
TargetFilename	<p>Name of the final target PDF document.</p> <p>This property is been used only when “pdfRenderType” is been set as “File”.</p>
TemplateFilename	<p>Name of the PDF template. Can be a external URL address or a file in local file system (server / client).</p> <p>Note: Use http:// prefix if the template is an external file resource in the Net.</p> <p>Normal file is been named e.g. c/dev/yourtemplate.pdf</p> <p>You can also use the computed value to set a fully customized location of the template file.</p>
ButtonText	<p>Button text</p>
buttonStyleClass	<p>Button style class (optional).</p> <p>You can use your own CSS styles as you like.</p>
RedirectPage	<p>Redirect user to selected Xpage after the PDF file has been created.</p> <p>Affects only when pdfRenderType is FILE.</p>

5 Known limitations

All Text Fields in Acro Forms are always text type of fields. There for all data to be inserted in to the PDF document needs to be in text format.

Custom Control does not support (yet) Rich Text items and their value. The “problem” is that all Acro Fields are text fields.