# GJax PDF Printing

**From Intranet**

**Table of contents**



## PDF Generation

To generate PDF documents on the server you need to install FOP from Apache. See XSL-FO to PDF for details of how to do this.

Within VC/m you need to make the following setting:

```
s ^%vcvc("fopCommand")="C:\progra~1\apache~1\fop-0.20.5\fop.bat"
```

Where the path points to the file fop.bat (or fop.sh on Linux) in the directory where you installed FOP. Note that it doesn't appear to work if there are spaces in the path name.

To create a pdf document, first generate a file containing your xml dataset, then call call $$generate^%vcPDF passing in the name of the xml file and an xsl template. The layout of the document is determined by the xsl template.

### gjsReport_Basic.xsl

This is a basic template that provides a banner page, standard page headers and footers and content in a simple tabular layout. It is formatted to fit A4 paper in landscape orientation (TODO: format to *US Legal* size paper if user's locale is US).

To use this template your xml file should have the following structure:

```
<report>
  <user user='georgej' name='George James' .../>
  <system siteTitle='Patient Leaflets (LIVE)' currentDateTime='20050515125901' .../>
  your data here
  .
  .
  .
</report>
```

Call reportStart^%vcPDF(title) to generate the starting <report> tag and the <user> and <system> elements. Call reportEnd^%vcPDF() to generate the ending </report> tag. See *report^%vcFunction* for an example of how to do this.

The user and system elements contain data that is likely to be useful for printing in the headers and footers of any report so these should always be included even if it doess't seem to apply to your document layout at the moment.

Create an xsl file for your report that <xsl:import>'s gjsReport_Basic.xsl. Add a template named *reportTable* and within that generate <fo:table-row> elements for each row of data using normal xsl techniques. See *Function_Report.xsl* for an example.

If your document layout doesn't fit the gjsReport_Basic.xsl template, then you can copy'n'paste it's contents into your template and adapt as required. If you are producing a series of documents with a similar structure you might want to consider creating a standard template for the page structure, headers and footers and include it in each document specific template.

**WARNING:** Do not be tempted to over-generalise the template to the point where you are doing layout related processing at the server end. The moment you find yourself generating xml attributes like *columnHeading="Description"* stop and reconsider your design. Static layout information goes in your xsl template not in xml dataset.

### Tips

- The PDF generation will fail if the pdf document you are generating already exists and is open in Adobe Acrobat viewer. If you are tweaking your document layout you will be quite likely to have it open. You have to remember to close the old pdf file in Acrobat before you can generate a new version.
- Write the code to generate your xml dataset first and get it working (if you are using standard xml production functions then this should be pretty trivial). Then you can concentrate on design of the document template without being plagued by errors in the xml generation part.

# Client Side Printing

# Server Side Printing

### Windows

The general approach to server side printing is to create an output file in pdf format and to then invoke Adobe's Acrobat Reader from the command line specifying the source document and a target printer.

#### Pre-requisites

- You must have a printer installed on the same server as Cache.
- You must have Adobe's free Acrobat Reader installed on the same server as Cache.

Note: If the printer is installed on a different server then it appears that the Acrobat Reader process doesn't terminate after printing and you will end up with 20 Acrobat processes running after printing 20 documents. Eventually you will run out of memory...

#### Cache

From Cache you need to issue the following command to the windows:

```
<path to acrobat>\acrord32.exe /t "document to print.pdf" "name of printer"
```

This doesn't seem to work to well at the moment - seems to be some problem with parsing of the command line (or maybe its a permissions thing).

Adobe is normally installed in c:\Program Files\Adobe\Acrobat 7.0\Reader\acrord32.exe

This should be a setup parameter.

There's also a problem that Acrobat doesn't terminate, but this can be addressed using $zf(-2), I think.

**Printer Setup**

You need to add one printer to Windows for each different printer configuration that you need. For example, you might have:

- \\yak\yeats portrait
- \\yak\yeats landscape
- \\yak\yeats portrait two copies
- \\yak\yeats portrait letter heading

Configure the printer preferences for each printer appropriately.

## Linux

http://www.cups.org/documentation.php



- This page was last modified 00:33, 21 Dec 2005.