

# Welcome to the org.apache.forrest.plugin.output.pdf Plugin

## Table of contents

1 Apache Forrest - org.apache.forrest.plugin.output.pdf Plugin.....	2
2 Configuration options.....	2
3 Font Family Configuration.....	2
3.1 Preparations.....	2
3.2 Font family overrides/specifications.....	3
4 Other presentation options.....	4
5 I18n Configuration.....	4
5.1 Font Family Configuration.....	4
5.2 L10n of label text.....	4

## 1 Apache Forrest - org.apache.forrest.plugin.output.pdf Plugin

This plugin (hereafter just called the PDF plugin) enables any Forrest page to be output as a PDF document.

To use it simply request a page with the pdf extension.

### Note:

By default this plugin is included in a new Forrest project. You will see a PDF link in the upper right corner of the body section of each page, this will, of course, link to the PDF version of that page.

## 2 Configuration options

The PDF plugin can be configured in several ways:

- font family configuration
- other presentation changes: page size, copyright notes, etc.
- i18n configuration, which is actually two subtopics:
  - font family configuration
  - l10n of label text

Each of these points will be covered in separate sections below.

## 3 Font Family Configuration

The font family can be specified in different steps of granularity. As of version 0.3 of the PDF plugin, all of the configuration can be done on a per project basis. There is no need to touch the plugin itself, as it used to be.

### 3.1 Preparations

If you want to change the actual font family, not only where to use the generic serif, sans-serif and monospace families, then you also need to take these preparatory steps:

1. Follow the instructions at <http://xmlgraphics.apache.org/fop/1.0/configuration.html> to create a FOP configuration file at `$PROJECT_HOME/src/documentation/resources/fonts` (you may have to create the `fonts` folder). The FOP configuration file can a.o. give you access to locally installed fonts.
2. Specify the absolute path to the FOP configuration file you created in step 1 as the value of the entity `&pdf-config-file;` in the project entity file. The file is located at: `$PROJECT_HOME/src/documentation/resources/schema/symbols-project-v10.ent`. The result should look something like:

```
<!ENTITY pdf-config-file "/Users/foo/forrestproject/src/documentation/resources/fonts/config.xml">
```

(the above would work for a user named `foo` on a typical MacOS X system - adapt to your local OS as needed, but the path needs to be absolute)

A minimal FOP configuration file to give you access to all fonts installed on the local OS looks like:

```
<fop>
  <renderers>
    <renderer mime="application/pdf">
      <fonts>
        <auto-detect/>
      </fonts>
    </renderer>
  </renderers>
```

```
</fop>
```

See the link above for more details.

Continue with the configuration in your `forrest.properties.xml` file as described below.

### 3.2 Font family overrides/specifications

For the easiest type of changing the font family, add the following lines to your `forrest.properties.xml` file:

```
<property name="output.pdf.fontFamily.serif" value="serif" />
<property name="output.pdf.fontFamily.sansSerif" value="sans-serif" />
<property name="output.pdf.fontFamily.monospace" value="monospace" />
```

and replace the default values above with your preferred font family, e.g. DeJaVuSans, etc. Using these properties, you will change the font family of whole classes of text types at once. If you need proper i18n support for your language (more below) in the pdf files, this is usually the easiest way to go - just replace the generic families with your chosen font families.

If you need to override or specify the font family for a certain text type, add one or more of the following properties to your `forrest.properties.xml` file:

```
<property name="output.pdf.fontFamily.rootFontFamily" value="serif" />
<property name="output.pdf.fontFamily.headerFooterFontFamily" value="sans-serif" />
<property name="output.pdf.fontFamily.documentTitleFontFamily" value="sans-serif" />
<property name="output.pdf.fontFamily.versionFontFamily" value="sans-serif" />
<property name="output.pdf.fontFamily.authorsFontFamily" value="sans-serif" />
<property name="output.pdf.fontFamily.TOCTitleFontFamily" value="sans-serif" />
<property name="output.pdf.fontFamily.TOCFontFamily" value="serif" />
<property name="output.pdf.fontFamily.sectionTitleFontFamily" value="sans-serif" />
<property name="output.pdf.fontFamily.sourceFontFamily" value="monospace" />
<property name="output.pdf.fontFamily.codeFontFamily" value="monospace" />
<property name="output.pdf.fontFamily.warningTitleFontFamily" value="sans-serif" />
<property name="output.pdf.fontFamily.noteTitleFontFamily" value="sans-serif" />
<property name="output.pdf.fontFamily.fixmeTitleFontFamily" value="sans-serif" />
<property name="output.pdf.fontFamily.abstractFontFamily" value="serif" />
<property name="output.pdf.fontFamily.noticeFontFamily" value="serif" />
```

The given values are defaults, and should be replaced with what you want.

#### Note:

The value specified for the properties should be the font family name. If the font family name contains spaces, enclose the name in single quotes.

#### Note:

If you are using Dispatcher (in the whiteboard), the property `output.pdf.fontFamily.versionFontFamily` does not have any effect. Hopefully this will be corrected in the future.

If you want, you can further specify the font family for variants of the `output.pdf.fontFamily.headerFooterFontFamily` property using the following properties instead:

```
<property name="output.pdf.fontFamily.firstFooterFontFamily" value="sans-serif" />
<property name="output.pdf.fontFamily.evenHeaderFontFamily" value="sans-serif" />
<property name="output.pdf.fontFamily.evenFooterFontFamily" value="sans-serif" />
<property name="output.pdf.fontFamily.oddHeaderFontFamily" value="sans-serif" />
<property name="output.pdf.fontFamily.oddFooterFontFamily" value="sans-serif" />
```

It is possible to do further customisations of the pdf rendering by editing the stylesheets found in ...

## 4 Other presentation options

---

Presentation changes like page size, copyright notes, etc. are presently specified in the `skinconfig.xml` file. See separate documentation for further info.

## 5 I18n Configuration

---

### 5.1 Font Family Configuration

---

For some languages, the default font families (`serif`, `sans-serif`, `monospace`) do not render all characters, giving squares, question marks or nothing instead in the rendered pdf document. This is known to be an issue with CJK languages, as well as with lesser used languages such as Sámi. If this is the case for your language(s), please follow the steps above to configure the pdf plugin to use a font family known to contain all characters needed. For many alphabetic languages, the free and open source [DejaVu font family](#) nowadays has quite good coverage.

### 5.2 L10n of label text

---

[TO BE WRITTEN]