How to use rst2pdf

Author: Roberto Alsina

Version: 0.3

Contents

Introduction

This document explains how to use rst2pdf version 0.3. Here is the very short version:

```
rst2pdf.py mydocument.txt -o mydocument.pdf
```

That will, as long as mydocument.txt is a valid Restructured Text (ReST) document, produce a file called mydocument.pdf which is a PDF version of your document.

Of course, that means you just used default styles and settings. If it looks good enough for you, then you may stop reading this document, because you are done with it. If you are reading this in a PDF, it was generated using those default settings.

However, if you want to customize the output, or are just curious to see what can be done, let´s continue.

Headers and Footers

ReST supports headers and footers, using the header and footer directive:

```
.. header::

   This will be at the top of every page.
```

Often, you may want to put a page number there, or a section name.The following magic tokens will be replaced (More will be added as rst2pdf evolves):

###Page###: Replaced by the current page number.

Styles

You can style paragraphs with a style using the class directive:

```
.. class:: special

This paragraph is special.

This one is not.
```

Or inline styles using custom interpreted roles:

```
.. role:: redtext

I like color :redtext:`red`.
```

For more information about this, please check the ReST docs.

The only special thing about using rst2pdf here is the syntax of the stylesheet.

You can make rst2pdf print the default stylesheet:

    rst2pdf --print-stylesheet

If you want to add styles, just take the standard stylesheet, modify it and pass it with the -s option:

```
rst2pdf mydoc.txt -s mystyles.txt
```

StyleSheet Syntax

It´s a JSON file with three elements in it.

fontsAlias

By default, uses some of the standard PDF fonts:

```
"fontsAlias" : {
  "stdFont": "Helvetica",
  "stdBold": "Helvetica-Bold",
  "stdItalic": "Helvetica-Oblique",
  "stdBoldItalic": "Helvetica-BoldOblique",
  "stdMono": "Courier"
},
```

This defines the fonts used in the styles. You can use, for example, Helvetica directly in a style, but if later you want to use another font all through your document, you will haveto change it in each style. So, I suggest you use aliases.

The standard PDF fonts are these:

    Times_Roman Times-Bold Times-Italic Times-Bold-Italic Helvetica Helvetica_Bold Helvetica-Oblique Helvetica-Bold-Oblique Courier Courier-Bold Courier-Oblique Courier-Bold-Oblique Symbol Zapf-Dingbats

Style Definition

Then you have a 'styles' which is a list of [stylename, styleproperties]. For example:

```
["normal" , {
  "parent": "base",
  "language": "EN"
}],
```

This means that the style called "normal" inherits style "base". So, each property not defined in the normal style will be taken from the base style. Also, it defines the language to be english, only useful if you enable hyphenation.

I suggest you do not remove any style from the default stylesheet. Add or modify at will, though.

If your document requires a style that is not defined in your styleheet, it will print a warning and use bodytext instead.

Also, the order of the styles is important: if styleA is the parent of styleB, styleA should be earlier in the stylesheet.

Font Embedding

There are thousands of excelent free True Type fonts available on the web, and you can use them in your documents by declaring them in your stylesheet.

The last element is called "embeddedFonts" and handles embedding True Type fonts in your PDF.

Usually, it's empty, because with the default styles you are not using any font beyond the standard PDF fonts:

```
"embeddedFonts" : [ ],
```

Suppose you want to use the nice public domain Tuffy font (included in rst2pdf's source distribution).

First, you need to tell rst2pdf that you want it embedded in your PDF, by using the embeddedFonts property of the stylesheet:

```
"embeddedFonts" : [ ["Tuffy.ttf","Tuffy_Bold.ttf","Tuffy_Italic.ttf","Tuffy_Bold_Italic.ttf"] ],
```

This will provide your styles with fonts called "Tuffy" "Tuffy_Bold" and so on. Also, if you use *italics* in a paragraph whose style uses the Tuffy font, it will use Tuffy_Italic. That's why it's better if you use fonts that provide the four variants, and that is the order in which you should put them. If your font lacks a variant, use the "normal" variant instead. For example, if you only had Tuffy.ttf:

```
"embeddedFonts" : [ ["Tuffy.ttf","Tuffy.ttf","Tuffy.ttf","Tuffy.ttf"] ],
```

However, that means that italics and bold in styles using Tuffy will not work (they will display as regular text).

If you want to use this as the base font for your document, you should change the fontsAlias section accordingly. For example:

```
"fontsAlias" : {
  "stdFont": "Tuffy",
  "stdBold": "Tuffy_Bold",
  "stdItalic": "Tuffy_Italic",
  "stdBoldItalic": "Tuffy_Bold_Italic",
  "stdMono": "Courier"
},
```

If, on the other hand, you only want a specific style to use the Tuffy font, don't change the fontAlias, and set the fontName properties for that style. For example:

```
["heading1" , {
  "parent": "normal",
  "fontName": "Tuffy_Bold",
  "bulletFontName": "Tuffy_Bold",
  "fontSize": 18,
  "bulletFontSize": 18,
  "leading": 22,
  "keepWithNext": true,
  "spaceAfter": 6
}],
```

By default, rst2pdf will search for the fonts in its fonts folder and in the current folder. You can make it search another folder by passing the --font-folder option, or you can use absolute paths in your stylesheet.

Syntax Highlighting

Rst2pdf adds a non-standard directive, called code-block, which produces syntax highlighted for many languages using Pygments.

For example, if you want to include a python fragment:

```
.. code-block:: python

   def myFun(x,y):
     print x+y
```

Notice that you need to declare the language of the fragment. Here's a list of the currently supported.