DocPu Redmine plugin

Christian Raschko

November 27, 2010

Contents

1 DocPu, Document publishing plugin . . . 1

1.1 Installation . . . 2

1.1.1 Installing RedCloth4 . . . 2

1.1.2 Installing LaTeX . . . 3

1.1.3 Installing DocPu plugin . . . 3

2 DocPu wiki syntax . . . 4

2.1 Font styles and enumeration . . . 4

2.2 Images . . . 5

2.3 Tables . . . 6

2.4 Code highlighting . . . 6

2.5 Wiki page references and footnotes . . . 7

3 DocPu sample page . . . 7

List of Figures

1 DocPu settings . . . 4

2 I am a floating image . . . 5

3 I am a fixed image . . . 5

1 DocPu, Document publishing plugin

This plugin enables *PDF* export and generation for Redmine wiki pages using the LaTeX typesetting system. But it is even more than an simple

PDF exporter since it is uses the LaTeX text and figure layout engine which produces extremely nice rendered page layouts.

Features:

- Merge multiple wiki pages to one single *PDF* document
- Automatically generate title page and table of content
- Automatically generate an index from all **strong** or *italic* words
- Wiki page references are translated to *PDF* links
- Code highlighting is supported for Redmine code block
- Solid table boarders can be drawn
- Add chapter for every wiki page
- Remove Redmine macros, eg. `{{toc}}`
- Convert wiki footnotes to LaTeX footnotes
- Use floating or fixed images
- ...

DocPu uses and extends the RedCloth4 LaTeX export module, which is a converter for the Textile markup language.

1.1 Installation

Since DocPu uses LaTeX and RedCloth4, you have to install them first.

1.1.1 Installing RedCloth4

Currently Redmine uses RedCloth3, so you have to additionally install RedCloth4. The simplest solution is by typing `gem install RedCloth`, but this only works if you have a compiler set-up, since some parts are written in C code. For windows users a pre compiled gem package can be downloaded from the repository. Download it first and install it via eg. `gem install RedCloth-4.2.2-x86-mswin32-60.gem`

1.1.2 Installing LaTeX

On Windows machines you can download and install the MikTex LaTeX distribution. Four all other platforms see http://www.latex-project.org/ftp.html.

DocPu Requires the following LaTeX packages, be sure you have them installed as well:

- `listings`
- `ulem`
- `graphicx`
- `float`
- `multirow`
- `makeidx`
- `hyperref`

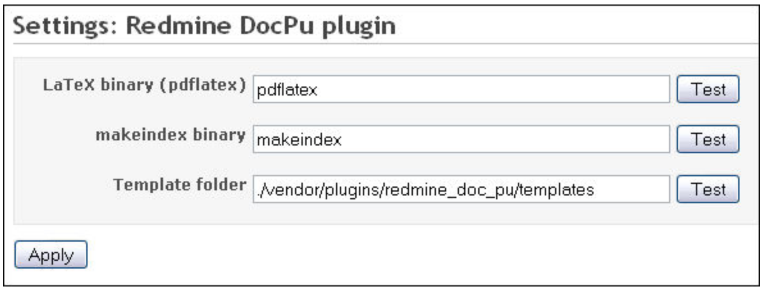
Testing LaTeX It's a good idea to test the LaTeX installation before using it with DocPu. Create an empty folder and copy the `article_en_tcfi.tex` file from the plugin template directory(`redmine_doc_pu/templates`) to it. Now create an empty file named `document.tex`, afterwards you can compile the document with `pdflatex article_en_tcfi.tex` (or path_to_latex/pdflatex). The file should compile successful with the message:

```
Output written on article_en_tcfi.pdf (1 page, 28874 bytes).
```

1.1.3 Installing DocPu plugin

To install DocPu, simple download it and extract its content into the Redmine plugin directory. Now run `rake db:migrate_plugins` from the Redmine directory and restart server. The more detailed installation guide can be found here.

Configure DocPu plugin Goto Redmine administration and select *Plugins*. You should see the DocPu plugin installed. Now select *Configure* and fill in the correct LaTeX and makeindex binary paths.



Settings: Redmine DocPu plugin

LaTeX binary (pdflatex): pdflatex [Test]

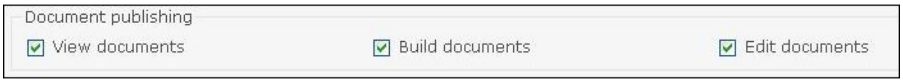
makeindex binary: makeindex [Test]

Template folder: ./vendor/plugins/redmine_doc_pu/templates [Test]

[Apply]

Figure 1: DocPu settings

In this example, LaTeX and makindex binary paths are set in PATH environment variable. You can also specify the full path, eg. `C:\Program Files\MiKTeX 2.7\miktex\bi` on windows.



Document publishing

☑ View documents ☑ Build documents ☑ Edit documents

Now it's time to set *Roles and permissions* for the DocPu plugin in Redmine. Therefore, goto administration and select *Roles and permissions*. You will see a new category called *Document publishing*, within that category there are 3 different security settings:

- **View documents:** Shows all documents assigned to a project, if disabled the project menu *Publish* is not displayed.
- **Build documents:** Build a document, enables the permission to execute the LaTeX binary.
- **Edit documents:** Edit document, this also involves creating, editing and deleting documents.

The last step is to enable the *Document publishing* module in the project settings.

2 DocPu wiki syntax

DocPu mainly uses the Redmine/Textile wiki syntax with some minor extensions.

2.1 Font styles and enumeration

You can use all supported font styles from the Redmine wiki such as **bold**, *italic*, underlined or ~~deleted~~ text. Links such as link are also supported.

- An enumeration

- Another enumeration
    - A sub enumeration
    - Another sub enumeration

And all enumerations are supported.

1. Enumeration 1
2. Enumeration 2
3. Enumeration 3

- **Description:** A simple description text.
- **More description:** Another longer description text. Another longer description text. Another longer description text. Another longer description text. Another longer description text.

2.2 Images

The image position can be fixed or floated. Fixed images occur at the text position where they are defined. Floating images will be laid out by LaTeX to best fit the page layout guidelines. Also a image caption is supported by using the alternative text field.

Figure 2: I am a floating image

- **Fixed images:** `!image.jpg!`
- **Floating images:** `!>image.jpg!`
- **Fixed images with caption:** `!image.jpg(Caption text goes here)!`

Figure 3: I am a fixed image

You can also create a reference to an image by writing `{{!image.jpg!}}`, see figure 2.

2.3 Tables

DocPu supports the Redmine/Textile table syntax with table span and heading.

```
|_.UserID|_.Name|_.Group|
|\3=.IT|
|\2=.IT|Test|
|1|Artur Pirozhkov|/2.Users|
|2|Vasya Rogov|
|3|John Smith|Admin|
```

Example: This table was generated from the code above.

UserID	**Name**	**Group**
IT		
IT		Test
1	Artur Pirozhkov	Users
2	Vasya Rogov	
3	John Smith	Admin

2.4 Code highlighting

Code highlighting is also supported using the LaTeX listings package. All CodeRay languages are supported.

¡/pre¿

Example: This listing was generated from the code above.

```
# The Greeter class
class Greeter
  def initialize(name)
    @name = name.capitalize
  end

  def salute
    puts "Hello #{@name}!"
  end
end
```

2.5 Wiki page references and footnotes

Another feature of DocPu[1] is the generation of wiki page references. Simply write `[[WikiPage]]` or `[[WikiPage|My page name]]` to add a wiki page reference. For an example see section 3 or My page name 3.

3 DocPu sample page

This is a empty sample page!

[1] This is a foonote.

Index

bold, 4
Build documents:, 4

Configure, 3

Description:, 5
Document publishing, 4

Edit documents:, 4

Fixed images with caption:, 5
Fixed images:, 5
Floating images:, 5

italic, 2, 4

More description:, 5

PDF, 1, 2
Plugins, 3
Publish, 4

Roles and permissions, 4

strong, 2

View documents:, 4