

 ***dalibo***
L'alternative Open Source



Author : Dimitri Fontaine <dimitri@dalibo.com>
Version : 0.4.4 - 21/08/2006
Type : Technical presentation
Abstract : In this document you'll learn how and why this dalidoc software was written, then how to use it.
We'll talk about how to launch the dalidoc command, but also how to customize the given output, which can be some PDF, plain HTML or HTML for presentation (S5).
Licence : Creative Commons Attribution-NonCommercial-NoDerivs 2.5
You are free to copy, distribute, display, and perform the work under the following conditions :
Attribution. You must attribute the work in the manner specified by the author or licensor.
Non commercial.. You may not use this work for commercial purposes.
No Derivative Works. You may not alter, transform, or build upon this work.
For any reuse or distribution, you must make clear to others the license terms of this work. Any of these conditions can be waived if you get permission from the copyright holder.
Your fair use and other rights are in no way affected by the above.
This is a human-readable summary of the Legal Code. The full license is available here :
<http://creativecommons.org/licenses/by-nc-nd/2.5/legalcode>

Revisions history

Version	Date	Libellé
0.1	12/17/2005	Version initiale
0.4	03/28/2006	Rewrite in english, conformance to new code version
0.4.3	05/07/2006	Some typos, documentation licence update

Table des matières

1	So, what about <i>dalidoc</i> ?	4
2	How to use <i>dalidoc</i>	5
2.1	<i>dalidoc</i> configuration file	5
2.2	<i>dalidoc</i> CLI	7
2.3	Examples	7
3	<i>dalidoc</i> inner working	9
3.1	rest file creation	9
3.2	pdf authoring	10
3.3	pdf bullet lists and latex	10
3.4	Standard html output	11
3.5	Slides output, in html	11
4	Conclusion	12

1

So, what about *dalidoc* ?

The *dalidoc* software was first an attempt to produce nice looking financial proposals to **dalibo** clients without having to fire up **OpenOffice.org**.

The reason is that the author of *dalidoc* finds that producing high quality documents respecting a common rule set with a *WYSIWYG* application is a somewhat clumsy process, and that using `text/plain` editing with simple tagging and an automated tool is far more efficient.

The input format was easy to choose : after having tried **ReST**, you just don't want to mess up with any other. The **docutils** package comes with some handy tools for `html` and `pdf` output generation, the latter using LaTeX allows high quality output.

We still needed some nice looking papers, hence we integrated some **PSTricks** to obtain a nice booklet cover, and then some LaTeX style hacking for better looking setup, including footer with company logo.

2

How to use *dalidoc*

dalidoc is about using some other tools together, it comes with a simple command line interface and a user configuration file to edit.

The way *dalidoc* uses the other tools is quite simple, and you may want to know more about it : this section will provide you the needed keys to understand or even tweak *dalidoc* internals.

dalidoc allows you to :

- create a `rest` file, optionnaly using a template ;
- produce a `pdf` file from a source `rest` file ;
- produce a plain `html` file from a source `rest` file ;
- produce a `s5.html` file from a slide source `rest` file.

For *dalidoc* to work, you first need to provide it a configuration file. A default one is given in `example/dalidoc.conf` in the source tree, you may find it in `/usr/share/dalidoc/examples/dalidoc.conf` (after install *debian* package for instance).

2.1 *dalidoc* configuration file

Here's a sample `dalidoc.conf` file content :

```
[dalidoc]
author   = Your Name <your@address.com>
template = note
locale   = fr_FR
verbose  = False
```

```
[paths]
make      = /usr/bin/make
texinputs = /usr/share/dalidoc/latex/ :
lib       = /usr/lib/dalidoc
tmpdir    = /tmp/dalidoc
s5theme   = /usr/lib/site-
python/docutils/writers/s5_html/themes/default

[logos]
footer = /usr/share/dalidoc/examples/default_logo.small.eps
cover  = /usr/share/dalidoc/examples/default_logo.eps
bg     = /usr/share/dalidoc/examples/default_background.eps

# if you don't want the default logo to appear you can
# either uncomment the lines below
# or create your own eps files
#
#footer = /usr/share/dalidoc/examples/blank_logo.small.eps
#cover  = /usr/share/dalidoc/examples/blank_logo.eps
#bg     = /usr/share/dalidoc/examples/blank_background.eps

[style]
latex = /usr/share/dalidoc/style/dalibo.sty
cover = /usr/share/dalidoc/style/cover.sty
css   = /usr/share/dalidoc/style/dalibo.css

[latex]
quotchap = /usr/share/dalidoc/latex/quotchap.sty

[templates]
note      = /usr/share/dalidoc/templates/note.rest
propale   = /usr/share/dalidoc/templates/propale.rest
slides    = /usr/share/dalidoc/templates/slides.rest
```

The configuration file format is the well known `.INI` style. You have to edit the first entry, that use Author name (and optionnaly email address).

Temporary and intermediates files (as latex and dvi output) are to be generated into the `paths.tmpdir` directory, which *dalidoc* will try to create for you if it does not exists.

We use some latex extensions styles such as `quotchap.sty` for our rendering, those are to be prepended to the latex environment variable `TEXINPUTS`. This path must end with a colon `' : '` character, *dalidoc* code will assure this in case you don't do it yourself.

The logo section allows you to indicate three logo, the first one will be on each page footer and the two others only found on the first page, the booklet front page.

2.2 *dalidoc* CLI

The command line provides some options to be used :

```
usage : dalidoc.py [option] create|clean <filename>
```

options :

```
-h, --help          show this help message and exit
-cCONFIG, --config=CONFIG
                    dalidoc configuration file
-aAUTHOR, --author=AUTHOR
                    dalidoc author    (create seulement)
-tTITLE, --title=TITLE
                    Titre du document (create seulement)
-pTEMPLATE, --template=TEMPLATE
                    Type de document  (create seulement)
-lLOGO, --logo=LOGO  Logo à utiliser sur la couverture
-sSLOGO, --smallogo=SLOGO
                    Logo à utiliser dans le pied de page
-v, --verbose       mode verbeux
```

You may want to use the `-v` switch as soon as you include some graphics in your document, or modify your logos, to follow any LaTeX output.

2.3 Examples

Here's the more simple and more common usage of *dalidoc*, the pre-generation of a `test.rest` file from which to obtain a nice `test.pdf` output file :

```
$ dalidoc.py -t 'This is a test' -p 'note' test.rest
$ $EDITOR test.rest
$ dalidoc.py test.pdf
```

You can also obtain a `test.html` file from the same source, which will reference your configured `style.css` stylesheet :

```
$ dalidoc.py test.html
```

If you want to write down some slides, you'll have to conform to their specificities and type in something like :

```
$ dalidoc.py -p slide -t 'Slide example' slides.rest
$ $EDITOR slides.rest
$ dalidoc.py slides.s5.html
```

3

dalidoc inner working

As a user of the *dalidoc* tool, you may want to know more about its internal working, thus being able to customize its output, add some new output format support or even fix some bugs.

The *dalidoc* tool uses docutils provided writers to transform the rest sources you feed it, and manage creating output by using a *Makefile*.

The main *python* script is used to parse templates resources and style files (using the *Cheetah* *python* solution) and preparing the build environment (in the `paths.tmpdir` configured directory) where to invoke *make*.

3.1 rest file creation

By default, the template `/usr/share/dalidoc/templates/note.rest` is named *note* in your configuration file and used if no `-p` switch was used.

Its used as a *Cheetah* template where some fields are replaced by configuration file and command line switches values. You can use any of these option in your own template files and add as many templates file as you need to, either in the shared system directory or your own user resource templates directory.

Available *dalidoc* templates options are :

- `$doc.title`
Document title, to be given by the `-t` command line switch.
- `$title_frame`

This variable is to be used as title underlining in your template documents, it will be replaced with as much = characters as four title length.

- `$doc.author`
The document author variable will be replaced by the one given in the *dalidoc* configuration file (the `dalidoc.author` configuration option) or the one given with the `-a` command line switch. The command line switch takes precedence.
- `$doc.type`
The document type, used in the cover page.
- `$doc.date` et `$doc.raw_date`
Those dates are generated by the *dalidoc* software, the former version figures expanded month name (using configured locale information), the latter a numeric date format : `jj/mm/aaaa`. You're now able to prepare as many *dalidoc* templates as needed!

3.2 pdf authoring

The pdf output generation is done with the tools `rst2latex` then `latex`. The former one comes from the *python docutils* distribution and produce `.tex` file from a `.rest` file; and the latter command is the well known latex processor which generates a `.dvi` file from some LaTeX input. Then `dvipdf` allows PDF rendering.

All those steps are controlled by the `/usr/lib/dalidoc/dalidoc.Makefile` makefile.

The `dalidoc.sty` latex stylesheet is generated by *dalidoc* from the following two templates files : `dalibo.sty` and `cover.sty`. Those stylesheets are templates to be able to use the title and type to be found in the source `.rest` document, and the logos configured.

The `cover.sty` template file uses some **PSTricks** code to obtain some nice effects and place the logos on the cover page, here you can tweak the booklet cover output to suit your company documentation format, for example.

3.3 pdf bullet lists and latex

The default latex bullet lists are not that nice, but are easy enough to accomodate. *dalidoc* authors made a choice upon some symbols to represent the first for bullet lists levels.

The latex command to use, for example `\renewcommand\labelitemi{\star$}`, has to be placed either before or after the `\begin{document}` macro, depending on which latex packages you happen to include.

The packages used by *dalidoc* implies placing those bullet lists renew commands after the document beginning, thus you'll find them in the `cover.sty` file.

Here are a sample bullet list, just to show off *dalidoc* choices as soon as in the README documentation :

- first bullet level
 - ⇒ second level
 - ◇ third level
 - ▷ fourth level
 - ◇ some more third level
 - ◇ where we stay
 - ⇒ then some second level entry
- and here we can finish this stupid bullet list example

3.4 Standard html output

The standard html output is obtained directly with the *docutils rst2html* command, and uses the given CSS stylesheet.

3.5 Slides output, in html

Slides are obtained by using the *docutils rst2s5* command. It generates some S5 slides using the configured `paths.s5theme` stylesheet. S5 is Eric Meyer *Simple Standard-base Slide Show System*.

4

Conclusion

To conclude, *dalidoc* tool allow its author to simply edit its business and technical papers with the same editor, and avoiding the *WYSIWYG* some people tend to prefer.

Enjoy *dalidoc*, enjoy free software !