



Write once, publish anywhere

TEXT MAGIC

How would you like write text using whatever text editor you like and then publish the results in a variety of formats, including (X)HTML, MoinMoin wiki, LaTeX, and SGML? txt2tags allows you to write once and publish anywhere. **BY DMITRI POPOV**

Txt2tags is a so-called document generator that can convert a text document containing special formatting markup into different file formats [1]. Although most word processors like OpenOffice.org, KWord, and AbiWord can save documents in different formats, txt2tags offers a few significant advantages:

- Txt2tags is an application-independent tool – you can use virtually any text editor to write documents.
- Because all your documents are stored as plain text files, you don't have to think about compatibility issues.
- Txt2tags can output documents in formats that some word processors don't support (e.g., MoinMoin wiki, man pages, and MagicPoint).

Getting Started

The txt2tags processing engine consists of a single 155KB file written in Python. This means that you have to install Python on your machine to be able to use txt2tags. Fortunately, most mainstream

Linux distributions come with Python preinstalled, so txt2tags should work right out of the box. Txt2tags installs one of two ways on your system. Some Linux distributions – for example, Ubuntu – have a prepackaged version of txt2tags in their repositories, so you can easily install it using your distro's package manager. The downside is that the txt2tags package includes an older version of the software. If you want to get the latest release of txt2tags, you should download the current tarball package from the txt2tags website, unpack the tarball in a directory of your choice, and start txt2tags by running the *python txt2tags* command. To use the txt2tags graphical interface, you might also need to install the python-tk package.

Txt2tags uses a lightweight markup language to identify and properly convert text formatting. Although the idea of learning a new markup language might sound like a daunting proposition, txt2tags' markup is extremely easy to grasp. Better yet, the tarball package

contains a few useful tools that add the support for txt2tags to popular text editors, including Emacs, gVim, Kate, and nano. For example, the *txt2tags.vim* plugin adds the txt2tags syntax highlighting to gVim, whereas the *gvim-menu.vim* plugin adds a txt2tags toolbar menu, so you can use txt2tags in gVim without learning the markup rules.

To install and enable syntax highlighting in gVim, move the *txt2tags.vim* file to the *~/.vim/syntax* directory (you might need to create it if it doesn't exist), then run the *:syntax enable* and *:set syntax = txt2tags* commands. To install the txt2tags menu in gVim, simply move the *gvim-menu.vim* file into the *~/.vim/plugin* directory.

If you prefer not to leave the comfort of your word processor, you might want to try the txt2tags extension for OpenOffice.org Writer created by yours truly [2]. To install the extension, download the latest release, choose *Tools | Extension Manager* in OpenOffice.org, select the *My Extensions* section, press *Add*, and

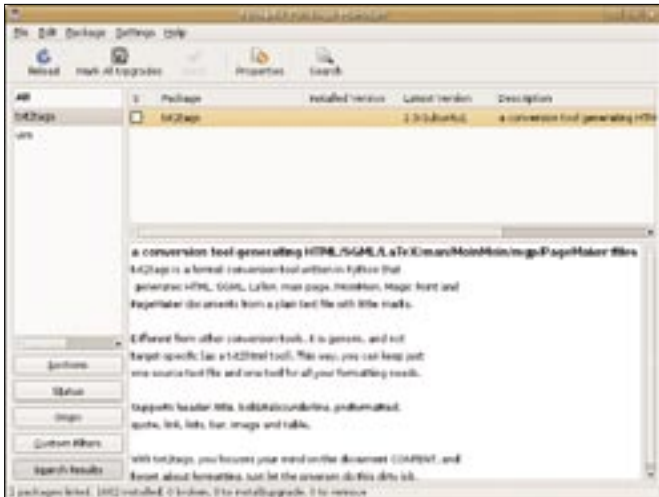


Figure 1: You can install txt2tags using your distro's package manager.



Figure 2: Txt2tags comes with some extra tools that add support for formatting markup to applications like gVim.

select the downloaded .txt file. The extension adds the txt2tags menu to the Main toolbar, and you can apply formatting by selecting the appropriate formatting command. Although the included tools support only some text editors, you can use whatever editor you like – you only need to add formatting manually.

Once you've added txt2tags support to your favorite text editor, you can start working on your documents using the txt2tags formatting. The txt2tags markup syntax is rather straightforward and looks a lot like wiki markup. Although txt2tags markup is lightweight, it provides all the basics, such as different text formatting options (bold, italic, underline, raw, etc.), inline images, hyperlinks, lists, headings, and tables. You can see an example of txt2tags-formatted text in Listing 1.

Macros and Filters

Besides the formatting markup, txt2tags includes two powerful features: filters and macros. The latter allows you to insert commands directly into the text, and txt2tags performs an appropriate action during the conversion process. For example, if you insert a `%toc` macro somewhere in the document, txt2tags generates a table of contents during the conversion. Need to insert the current date or modification time? Use the `%date` and `%mtime` macros. If you are working on a long document, you might find the `%infile` macro particularly useful. With this macro, you can insert the contents of another file into your document, for example: `%infile chapter1.t2t`.

This allows you to divide the long document into separate files and assemble them into a master document.

The first three lines of the document are reserved for txt2tags headers and compose the Header Area. During the conversion, you can choose whether you want txt2tags to print them or not. This feature can come in handy if you want to add some meta info about your document. For example:

```
Article title
Author
Modified: 2
%%mtime(%Y-%m-%d %H:%M)
```

The lines after the Header Area compose the Configuration Area, where you specify filter, macros, and other settings.

Filters are another powerful feature that allows you to specify the way txt2tags processes the source document. Txt2tags has two types of filters: `%preproc` filters are processed before the document is converted, whereas `%postproc` filters are handled after the conversion.

`%preproc` filters act as a “find and replace” command, and they can be used for a variety of pur-

poses. The most obvious purpose is to specify abbreviations. For example, typing “deoxyribonucleic acid” can be a bit of a bother, so instead, you can specify a filter such as `%preproc DNA deoxyribonucleic acid`.

Now you can use the DNA abbreviation in your document. When txt2tags processes the source file, it replaces all occurrences of DNA with the “deoxyribonucleic acid” string.

You can also create more complex filters. For example, you can define a filter that automatically adds a link to a specified word, as shown in Listing 2. `%postproc` filters work similarly to `%preproc`, but the filters are applied after the document is converted. This can be useful for fine-tuning the target file.

For example, the following filter changes the background color of the HTML page to blue:



Figure 3: With the txt2tags extension, you can use OpenOffice.org as a txt2tags editor.



Figure 4: Txt2tags features a simple and functional graphical interface.

```
%!postproc(html): '<BODY.*?>' ➔
'<BODY BGCOLOR="blue">'
```

Besides filters and macros, you can specify other settings directly in the document. For example, you can specify the target format right in the document using the `%!target` setting:

```
%!target: html
```

The filters, macros, and settings specified inside the source document apply only to that document. However, with a user-configuration file (also called RC file), you can specify options that apply

Listing 1: Example of txt2tags Markup

```
01 = txt2tags description =
02 Txt2tags supports **bold**, //
   italic//, __underlined__, and
   "___raw___" text formatting.
03 You can add hyperlinks (here
   is the link to Google (http://www.google.com) and inline
   images:
04
05 [Figure 1]
06 Txt2tags can also handle
   numbered (+) and bulleted (-)
   lists as well as tables:
07
08 || Table heading 1 | Table
   heading 2 |
09 | Column 1 | Column 2 ||
10
11 %%%You can also add comments,
   which will not appear in the
   converted document.
```



Figure 5: The txt2tags web interface allows you to convert text snippets directly in your browser.

to any document txt2tags processes. The tarball package comes with a sample RC file, which you can move into the `$HOME/.txt2tagsrc` directory and tweak to your liking.

When you are done formatting your document, you are ready to convert it. The easiest way to do this is to use txt2tags' graphical front-end, which you can launch with the `python txt2tags --gui` command. All of the GUI's options are self-explanatory, so you shouldn't have any problems using it.

Txt2tags also runs from the command line with different options. This can be particularly useful if you want to automate the conversion process or you want to perform a conversion in the background. Using the available options, you can control virtually any aspect of the conversion process. For example, the following command converts the source document into HTML format, skipping the headers and suppressing all output (except errors):

```
--target html ➔
--no-headers --quiet
```

The txt2tags manual provides a detailed overview of the available command options [3].

Listing 2: Filter

```
01 %!preproc: '(Google)'
   '[\1 http://google.com]'
02 %!preproc: 'LINUX Magazine'
   '[\1 http://linux-magazine.com]'
```

The txt2tags tarball packages also include a web-based front-end written in PHP. Drop the `txt2tags.php` file into the document root of your web server, and you can convert formatted text directly in your browser. Keep in mind that the script requires the *Save Mode* turned off. Although the web interface provides only the most basic features of txt2tags, it can be useful for quick-and-dirty conversion of small text snippets when you don't have access to the desktop tool.

Final Word

It is amazing what a 155KB Python utility can do. Besides providing a clean and easy-to-use markup language, txt2tags allows you to convert your source text in a variety of formats with minimum effort. Txt2tags is an elegant piece of software and a shining example of the KISS approach [4]. Although it's not overloaded with features, it's flexible enough for most tasks, including the complex ones. Albeit with some limitations, you can even use it to write books (<http://txt2tags.sourceforge.net/writing-book.html>) as well as do some other nifty stuff (<http://txt2tags.sourceforge.net/tips.html>). ■

INFO

- [1] txt2tags: <http://txt2tags.sourceforge.net/>
- [2] txt2tags extension: http://sourceforge.net/project/showfiles.php?group_id=179652&package_id=209734
- [3] txt2tags manual: <http://txt2tags.sourceforge.net/manpage.html>
- [4] KISS principle: http://en.wikipedia.org/wiki/KISS_principle