

1 NAME

blazeintro - an introduction to BlazeBlogger

2 SYNOPSIS

```
$ blaze-init
$ blaze-add
$ blaze-make
```

3 DESCRIPTION

This document tries to provide a brief introduction on how to create, configure, add new content to, and generate a static content from the BlazeBlogger repository.

Although each command described below is actually an autonomous utility, on most Unix systems, BlazeBlogger should be shipped with a command wrapper, making it possible to use `blaze-command` and more usual `blaze command` interchangeably. In this text, however, we will strictly use the first form, as it is the most portable variant.

Creating a New Blog

To create a new blog, you have to create a repository first. Similarly to Git, BlazeBlogger repository is just a (usually hidden) directory named `.blaze`, where all the blog data and configuration are stored and from which the static content, i.e. the HTML pages, is generated. In most cases, it is placed in the same directory as the generated content.

Assuming you are already in the directory where the blog repository is to be placed, you can create a new one typing:

```
$ blaze-init
```

If nothing goes wrong, you should get a response similar to this:

```
Created a BlazeBlogger repository in .blaze.
```

As the message suggests, you can use the same command any time later to return the repository to a consistent state while leaving the rest of your data (e.g. configuration, blog posts and pages) intact. Optionally, you can force the utility to restore the configuration, theme, style and language files to their initial state by adding the following option:

```
$ blaze-init --force
```

Configuring the Blog

Although the defaults try to be reasonable enough and ready for an immediate use, there are few options you might at least want to know about.

The first thing probably to be changed is an external text editor you will be using for writing your posts. Unless this option is set, BlazeBlogger tries to use system wide settings by looking for EDITOR environment variable, and if neither of these options is supplied, vi is used as a considerably reasonable option. To change it to, lets say, pico, type:

```
$ blaze-config core.editor pico
```

Another things you might want to change are the heading and subheading of your blog, and the default user name; for example:

```
$ blaze-config blog.title My Blog
$ blaze-config blog.subtitle yet another blog
$ blaze-config user.name Your Name
```

Furthermore, if you intend to generate RSS feed, you have to specify the base URL:

```
$ blaze-config blog.url http://blog.example.com/
```

Finally, if you prefer coloured output in your terminal window, you can enable it individually for each utility that supports it:

```
$ blaze-config color.list true
$ blaze-config color.log true
```

For the complete list of available options, see **blaze-config(1)** manual page. To see the current setting, simply type the option name without the value:

```
$ blaze-config user.name
```

You can also edit all options at once opening the configuration file in the external text editor:

```
$ blaze-config --edit
```

Managing the Blog Posts

To add a new blog post to the repository, type:

```
$ blaze-add
```

This will open the external editor with a pre-formatted header and empty body for you to write. Although the most of the header lines are devoted to the exhaustive explanation of what exactly you should and should not fill in, let's take a closer look at the five header options just to make sure everything is perfectly clear:

title

The title of your post; you do not want to omit that one, do you?

author

The post author. The user name specified in the configuration is used by default, but you are free to change it to whatever name you wish.

date

Date of the publishing of the post. It has to be in the YYYY-MM-DD form (i.e. 17 July 2009 would look like 2009-07-17 etc.) and it has to be filled. The default option is current date.

tags

Comma separated list of categories the post belongs to. The exact number of attached tags is not limited as long as they are all listed on the same line. Since tags are completely optional, you can safely leave this option empty, or fill it any time later.

url

The use of term 'URL' in this context is quite vague and simply means the part of the URL representing the particular post. As it is derived from the post title, you usually do not have to bother with it at all. There are however two cases when it might prove useful: (1) when your title contains special characters (typically foreign diacritical marks) that BlazeBlogger strips away and you still want your URL look pretty, or (2) in case you want your URL different from the title, for example because of the title length. Remember that the URL should consist of alphanumeric characters, hyphens and underscores only.

Keeping that in mind, your first post might look somehow like the one below:

```
# This and following lines beginning with '#' are the post header.
# Please take your time and replace these options with desired values.
# Just remember that the date has to be in an YYYY-MM-DD form,
# tags are a comma separated list of categories the post (pages ignore
# these) belong to, and the url, if provided, should consist of
# alphanumeric characters, hyphens and underscores only. Specifying
# your own url is especially recommended when you use non-ASCII
# characters in your post title.
#
# title: The Hobbit, or There and Back Again
# author: David
# date: 2009-07-17
# tags: childrens literature, fantasy
# url: the-hobbit
#
# The header ends here. The rest is the content of your post.
```

```
<p>&ldquo;In a hole in the ground there lived a hobbit.&rdquo;
&mdash; these are the words that introduces us to the incredible
world of nowadays classic Tolkien's novel. But what is so
fascinating about it that generation after generation keep returning
to its charms?</p>
```

```
<!-- break -->
```

```
<p><em>The Hobbit</em> is a story about Bilbo Baggins, a hobbit ...
```

Note that you can use `<!-- break ->` to mark the end of the part to be displayed on index page. You can also use a special placeholder, `%root%`, as a relative path to the base URL (directory where the blog is placed) in your links and images. For example, to link the RSS feed from your blog post, write:

```
<a href="%root%/index.rss">RSS Feed</a>
```

When you are done, save your changes and close the editor. You should get a confirmation message with the ID of the new post.

If you prefer to write your posts in advance (e.g. work with concepts), you can also add any number of existing files without opening the text editor like this:

```
$ blaze-add file1 file2 file3
```

To edit the existing post, e.g. with ID number 3, type:

```
$ blaze-edit 3
```

To completely remove the post with ID 3 from the repository, use the following command:

```
$ blaze-remove 3
```

Do not forget that the utility has no conscience and remorselessly deletes anything you tell it to, with no way to take it back (unless you have backup, of course). To prevent the accidental loss of potentially valuable data, you might want to consider running it in an interactive mode using the `-interactive` (or shorter `-i`) option and thus having to confirm every deletion.

Managing the Pages

Pages are the right place for all those information that have to be immediately accessible from every page on your website (typically the About, Contact, Downloads or Screenshots sections). They are usually kept in a relatively small number and generally do not tend to vary much. In the default theme, they are being listed in the horizontal menu below the blog title.

Similarly to blog posts, you can manage pages using the same commands with additional `-page` (or `-p` for short) option. Adding new page would therefore look like this:

```
$ blaze-add --page
```

Unlike posts, pages do not respect tags, nor do they display their author and the date of publishing. The latter is however still taken in account in the sorting process, making it possible to alter the order of pages in the menu according to your taste.

To edit existing page with ID 3, use:

```
$ blaze-edit --page 3
```

Finally, to remove pages with ID 2, type:

```
$ blaze-remove --page 2
```

Browsing the Blog Repository

To display all blog posts in the repository, e.g. to look up an ID of the particular post, use following command:

```
$ blaze-list
```

You can also narrow the results by providing the additional specifications like date, tag or author. For example, to find all posts from 19 December 2008 written by David, type:

```
$ blaze-list --author David --year 2008 --month 12 --day 19
```

To browse pages instead of blog posts, simply add the `-pages` option:

```
$ blaze-list --pages
```

Generating the Static Content

To generate the static content from your repository in the current working directory, with all posts, pages, monthly and yearly archives, tags and RSS feed included, type:

```
$ blaze-make
```

or if you wish to see a list of files as they are created:

```
$ blaze-make --verbose
```

You can now check the appearance of your web presentation in the browser and/or copy the files to the hosting. Note that you can force the full paths creation to make it easier to browse the off-line content:

```
$ blaze-make --full-paths
```

Furthermore, if the public web directory is located on the same server, you can create the content directly in that location by specifying the destination directory:

```
$ blaze-make --destdir ~/public_html
```

For the complete list of all command line options, see **blaze-make**(1) manual page.

Observing the Repository History

Since every significant change in the repository (i.e. its initialization and recovery, as well as post/page addition, editing and removal) is automatically logged along with the exact time of when it happened, you can view the whole history typing:

```
$ blaze-log
```

or if you want to save space:

```
$ blaze-log --short
```

Changing the Blog Theme

Changing the theme of your blog is rather straightforward. First, extract the content of the theme package to the directory where the static content is being generated; for example, to install the theme called VectorLover:

```
$ tar xzf vectorlover-1.0.tar.gz
```

Open the template file, in this case `vectorlover.html`, in your favourite text editor and replace the sidebar sections like ‘About’ or ‘Links’ with appropriate information. Then place both template file and the corresponding stylesheet to the `.blaze/theme/` and `.blaze/style/` directories respectively:

```
$ mv vectorlover.html .blaze/theme/  
$ mv vectorlover.css .blaze/style/
```

Finally, change the configuration to use this theme:

```
$ blaze-config blog.theme vectorlover.html  
$ blaze-config blog.style vectorlover.css
```

Run **blaze-make**(1) to re-create the static content with the new outfit.

For more information on how to modify existing theme or create a new one from scratch, see **blazetheme**(7) manual page.

Changing the Blog Localization

To change the blog language to, let's say, Czech, copy the file `cs_CZ` to the `.blaze/lang/` directory and change the appropriate configuration option:

```
$ blaze-config blog.lang cs_CZ
```

Next time you invoke `blaze-make(1)`, all generated strings should be properly translated.

4 FILES

.blaze/config

BlazeBlogger configuration file.

.blaze/log

BlazeBlogger repository log file.

.blaze/theme/

BlazeBlogger themes directory.

.blaze/style/

BlazeBlogger stylesheets directory.

.blaze/lang/

BlazeBlogger language files directory.

5 SEE ALSO

`blaze-init(1)`, `blaze-config(1)`, `blaze-add(1)`, `blaze-edit(1)`, `blaze-remove(1)`, `blaze-list(1)`, `blaze-make(1)`, `blaze-log(1)`.

6 AUTHOR

Written by Jaromir Hradilek <jhradilek@gmail.com>

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included as a file called `FDL` in the main directory of the BlazeBlogger source package.