

fnews_post_proc

Post-Processor for the fnews Newsreader

SYNOPSIS

The post-processor is invoked by piping an article into it. This typically occurs automatically when the "post_proc" variable is set to **fnews_post_proc** in the fnews configuration file.

For testing purposes, you can also pipe locally stored articles into the post-processor using a command like:

```
fnews_post_proc < article
```

.or:

```
cat article.txt | fnews_post_proc
```

DESCRIPTION

The fnews newsreader fully supports Usenet access, enabling the reception and reading of articles from newsgroups, as well as composing and posting them.

When comparing various news clients, differences will often influence your choice of preferred client. Fnews distinguishes itself by not only offering control over its internal behavior, but also by allowing you to modify outgoing posts just before they are sent to the configured NNTP server.

The **fnews_post_proc** program can be used to modify and enhance a post in ways that fnews alone cannot. It is configurable and may meet the needs of some Usenet users, though it is primarily intended as a starting point for creating your own custom solutions.

Limitations of Basic Newsreaders

While fnews creates posts that are generally complete and ready for submission, there are instances where users might find the output unsatisfactory for arbitrary reasons:

- ▶ If you're posting to different newsgroups in multiple languages, the introductory line referencing a prior post can only be set once in the fnews configuration file. As a result, posts to a French newsgroup, for example, may start with an introduction in English. The post-processor can address this by setting specific introductory lines for each newsgroup.

- ▶ Similar issues arise when you set a standard signature and want it to vary based on the newsgroup. The post-processor allows different signatures to be used for different newsgroups, and you can even randomize the selection from a list of signatures stored in a file.
- ▶ Custom headers, such as GnuPG key IDs or your language proficiency, can also be added to posts, though signatures may be a better alternative. Custom headers can be defined in the configuration file and will be added to outgoing posts.
- ▶ Headers like "Archive" and "X-No-Archive" are sometimes set to prevent an article from being indexed by search engines (e.g., Google). For example, test posts likely don't need to be included in search results. The post-processor can apply both headers to all posts sent to certain newsgroups.

NOTE: As of 2024, the "X-No-Archive" header has limited effectiveness, and it is up to server operators whether or not to honor it.

- ▶ Posts that include many references to other posts or URLs can become cluttered. The post-processor can convert marked text into footnotes, similar to the <ref/> link format used in Wikipedia, with the delimiter configurable in the configuration file. Example (using %=): "This is an object %=and this becomes the footnote, describing the object further=%". See also the hints under [EDITING FOOTNOTES](#), below.

Dialog for Overriding Settings

IMPORTANT: As of version 1.72, YAD or Zenity cannot be used for the dialog interface.

Before the post-processor runs, a dialog can be displayed to allow users to **disable** certain [CONFIGURATION](#) options. If Whiptail or xterm are available, you can choose to disable the following options. You **cannot** enable options that haven't already been set in the configuration.

- ▶ Signatures (if set in the [CONFIGURATION](#)) can be **ignored**. A default signature will be used, or no signature at all.
- ▶ Custom headers can be **omitted**.
- ▶ The "Archive" and "X-No-Archive" headers can be **ignored** if set for the current newsgroup.
- ▶ Logging can be **disabled**, if enabled.

Pressing Esc or the Cancel button on the dialog will interrupt the process, and flnews will not post the article.

To disable the dialog entirely (and apply all configured options without further interaction), set the **OVERRIDE_CONFIG** option.

CONFIGURATION

Upon first execution of the program, a copy of the original configuration file is created in `/home/[user]/flnews_post_proc.conf`. This file is used thereafter. Deleting it will cause the file to be recreated, but any custom changes will be lost.

The configuration file uses YAML syntax and includes detailed explanations. The variables within it fall into two categories:

1. Variables that represent values originally set by flnews, which can be used or replaced. These elements are typically captured using regular expressions.
2. Variables that define new or modified content.

Configuration Options

FUP_NAME

A regular expression that matches the name of the previous poster in a quoted post. This string is recognized in the original article and used with the matching element from GROUP_INTROS.

Leave this field empty to retain the default value from the FLNews configuration.

CONTENT: A string representing a regular expression.

DEFAULT: EMPTY

EXAMPLE1: "On \\d+\\.\\d+\\.\\d{2,4} at \\d+:\\d+ (.*) wrote:"

EXAMPLE2: "(.*) wrote:"

FUP_GROUP

A regular expression that matches the newsgroup where the quoted post was published.

Leave this field empty to ignore the exact group.

CONTENT: A string representing a regular expression.

DEFAULT: EMPTY

EXAMPLE: "wrote in (.*):"

GROUP_INTROS

Introductory strings referring to the previous poster in a quoted post. When the newsgroup matches `FUP_GROUP`, these variables are used in the resulting introduction.

CONTENT: A newsgroup or regular expression per line, followed by a colon, space, and string.

DEFAULT: As configured in FLNews.

EXAMPLE: alt.test: "Thus spoke %fup_name% on that baleful %fup_date%:"

GROUP_SIGS

Signature lines for each newsgroup or a path to a file containing signatures, which should be formatted and separated by one empty line. A random signature is picked from the file for each newsgroup. You can also source another file within the signature file by starting a line with ". /path/to/other/file".

CONTENT: A newsgroup or regular expression per line, followed by a colon, space, and string.

DEFAULT: As configured in flnews.

EXAMPLE:

```
alt.test: "Signature for alt.test\r\nsecond line"  
comp.*: /home/[user]/.my_sigs
```

CUSTOM_HEADERS

Additional custom headers to be added to outgoing articles.

CONTENT: One line per header, starting with a dash and space, followed by the header name and value.

DEFAULT: Undefined

EXAMPLE:

```
- 'X-My-Header: nothing fancy'  
- 'X-Another-Header: care not!'
```

NO_ARCHIVE_GROUPS

Newsgroups where the "Archive: no" and "X-No-Archive: YES" headers should be set.

CONTENT: A line with a dash and space, followed by the group name or a regular expression.

DEFAULT: Empty

EXAMPLE:

```
- "alt.test"  
- "^news.*"
```

DEBUG_LOG

The name of the file where debug messages are logged. If specified, logging will be enabled. If empty, logging is disabled.

CONTENT: The name of a writable file, which will be created if it doesn't exist and overwritten if necessary.

DEFAULT: Empty

EXAMPLE: `"/tmp/a_log-file.txt"`

LOG_LEVEL

One of: debug, fatal, error, info, warn.

REFERENCES_SEPARATOR

A symbol or sequence marking the end of the message body and the beginning of a footnote list. If not defined, footnotes are appended directly after the message body.

CONTENT: A quoted symbol or sequence.

DEFAULT: Empty

EXAMPLE: `"-----"`

REFERENCES_DELIMITER

A sequence marking the beginning and end of text to be used as a footnote or reference. The reversed sequence is used to mark the end.

CONTENT: A quoted symbol or sequence.

DEFAULT: None (empty)

EXAMPLE: `"%?"`

REFERENCE_FORMAT

A format string using %s for a number to replace the reference text in the body.

DEFAULT: `" %s)"` → becomes 1) ... 2) ... 3)

EXAMPLE: `"(%s)" # becomes (1) ... (2) ... (3)`

VFY_URLS

A boolean constant determining whether URLs should be verified and corrected if necessary. URL manipulations include adding angular brackets ('<', '>') and slashes after "http(s):".

CONTENT: One of YES, yes, NO, no (case insensitive).

DEFAULT: yes

EXAMPLE: ... I leave this to your discretion.

OVERRIDE_CONFIG

A boolean constant that controls whether configuration options can be overridden before posting.

Set this to "no" to disable the dialog.

DEFAULT: yes

EXAMPLE: No

Other Information

Editing footnotes

Writing footnote text (or URL references) inline while composing an article may be tempting, but it will likely break your line formatting. It's difficult to predict line lengths after footnotes are automatically moved to the end. Instead, write your complete article first, then add footnotes just before posting. This lets you anticipate how the final article will look after footnote markers replace the inline text (see [REFERENCE_FORMAT](#), above).

Signatures in Supersedes

When superseding an existing post, the post-processor will NOT automatically add a signature. This preserves the original post content as much as possible.

To retain the signature from the original post:

1. In flnews, select the entire article body with the mouse—from the first line (including any introductions) down to the last line of the signature.
2. From the Article menu, select **Supersede**.
3. Edit the article text as needed. The signature will be included in the editable text (normally, flnews strips signatures before opening the editor).
4. Send the article.

Note: The [override dialog](#) (see above) does not affect signatures in superseding posts.

Testing

You can verify the post-processor's effect by using one of two methods:

1. Pipe a saved post into the program:
`~$ /usr/local/bin/[post-processor] < [test-article]`
This will show the modified article, and you can pipe it into another file for further testing.
2. Post directly to a test newsgroup (e.g., alt.test). This is essential before posting to real newsgroups, especially when the post-processor will alter the article.

Source Code

The gem file for the program, available via the gem utility or rubygems.org, contains all the source code and documentation (this page included). To extract it, do the following:

1. Extract the gem file: `tar -xf flnews_post_proc-0.1.gem`
2. Uncompress the data archive: `gunzip data.gz`
3. Extract the data archive: `tar -xf data.tar`

This will create the directories bin, doc, and lib.

License

The **flnews_post_proc** is distributed under the [WTFPL-2.0](http://www.wtfpl.net) or later License. For the complete license text, visit <http://www.wtfpl.net/txt/copying/> or refer to the license file in the documentation directory of the gem.

Author

flnews_post_proc has been developed by Michael Uplawski
<michael.uplawski@uplawski.eu>