Fangfrisch

Fangfrisch (German for "freshly caught") is a sibling of the Clam Anti-Virus freshclam utility. It allows downloading virus definition files that are not official ClamAV canon, e.g. from Sanesecurity and URLhaus. Fangfrisch was designed with security in mind, to be run by an unprivileged user only. For those curious about the development history, a changelog is available online.

Table of Contents

1. License

General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

Fangfrisch is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with Fangfrisch. If not, see https://www.gnu.org/licenses/.

2. Update strategy

Fangfrisch is expected to run periodically, e.g. using cron. Fangfrisch will attempt to download digests first (if available upstream), and only retrieve corresponding signature files when their recorded digest changes, minimising transfer volumes.

I recommend running Fangfrisch at 10 minute intervals. There is no need to worry about overburdening your machine, the network or remote signature providers. Each individual download is recorded in the database, and whenever Fangfrisch is run, it checks those recorded timestamps against the preconfigured provider intervals. No network connections are made for downloaded signature files which are considered "too young to update".

Imagine a provider interval setting of 45 minutes. Even if launched every 10 minutes, Fangfrisch will not attempt to download related signature files before 45 minutes have passed since the last recorded successful download. Provider intervals are set to reasonable defaults internally, but you can override them if necessary.

3. Installation

Fangfrisch requires Python 3.8 version or newer. While Fangfrisch was initially written for Python 3.7, security support (the final lifecycle stage) for that version was terminated by the Python Software Foundation in 2023.

The recommended installation method is using the pip command in a virtual Python environment. Here is an example listing of commands for BASH, to be executed as root, assuming that you will be running Fangfrisch as an unprivileged user who is member of the **clamav** group:

3.1. Create home directory

```
mkdir -m 0770 -p /var/lib/fangfrisch
chgrp clamav /var/lib/fangfrisch
```

This will grant group members the necessary write access to create the database (see Section 6).

3.2. Prepare and activate venv

```
cd /var/lib/fangfrisch
python3 -m venv venv
source venv/bin/activate
```

3.3. Install via PyPI

```
pip install fangfrisch
```

This step will also create an executable launcher script `venv/bin/fangfrisch`.

4. Installation packages

As an alternative to pip-based installation, there are packages available for the following Linux distributions:

- Arch Linux: packages/python-fangfrisch. Support contact: Arch package maintainer.
- Clear Linux: clearlinux-pkgs/fangfrisch. Support contact: Clear Linux package maintainers.
- Debian Linux: fangfrisch (utils). Support contact: Debian package maintainers.
- Gentoo Linux: app-antivirus/fangfrisch. Support contact: Gentoo package maintainers.
- NixOS: fangfrisch Support contact: NixOS package maintainers.

5. Configuration

A configuration file is mandatory, uses an INI-File-like structure and must contain a `db_url` entry. All other settings are optional. However, unless you enable one signature file provider section, Fangfrisch naturally won't do much.

Use the `--conf` command line argument (see Section 7) to specify the path to your configuration file. Note that there is no default location.

```
# Minimal example configuration, meant for testing.

[DEFAULT]
db_url = sqlite:////var/lib/fangfrisch/db.sqlite
local_directory = /var/lib/clamav

[urlhaus]
enabled = yes
```

- **cleanup**: Cleanup method used for provider sections. Default: `automatic`, alternative: `disabled`.

In automatic mode, Fangfrisch will attempt to delete obsolete virus definition files whenever you disable a provider section. Should you disable this option, orphaned files will be left behind, and you need to ensure cleanup by different means.

- **connection_timeout**: Timeout in seconds, used when establishing network connections to download signature files. Default: `30`. Be aware that longer timeouts increase the risk of multiple Fangfrisch instances running concurrently. This could, in turn, cause database access problems, and network socket congestion. If your Internet connection is fast, I even suggest testing timeout values lower than the default 30 seconds.
- **db_url**: Database URL in SQLAlchemy syntax. Mandatory, no default. Typically, a local SQLite database will suffice.
- **enabled**: Scan this section for URLs? Default: `false`.
- **integrity_check**: Mechanism for integrity checks. Default: `sha256`. You can use `disabled` if the signature file provider offers no checksums.
- **interval**: Interval between downloads. Defaults are provider-dependent. Values can be expressed in human-readable form (e.g. `12h` or `45m`). Please respect the limits set by each provider.
- **local_directory**: Downloaded files are stored here. No default, so the current working directory of the Python process is used. As this can vary depending on how you launch Fangfrisch, it is highly recommended to define an absolute path like `/var/lib/clamav` instead. You can override this option in provider sections to separate downloads based on origin.
- **log_format**: See Formatter class documentation for details. Fangfrisch uses sensible defaults depending on the selected log method.
- **log_level**: Choose one of `DEBUG`, `INFO`, `WARNING` (default), `ERROR` or `FATAL`.
- **log_method**: Either `console` (default, meaning stdout/stderr) or `syslog`. For the latter, you can also specify a **log_target**.
- **log_target**: The syslog target address. Typical values are `/dev/log` (local Linux domain socket), `localhost` or `host.domain.tld:udpport`. If no target is specified, `localhost` is assumed. The UDP port number defaults to 514.
- **max_size**: Maximum expected file size. The default is `10MB`, but all predefined providers have individual size limits (see Appendix A). Values are can be expressed in human-readable form (e.g. `250KB` or `3MB`). Fangfrisch attempts to inspect the content length before downloading virus signature files so as not to download files larger than the defined limit. If providers don't respond with content length information, Fangfrisch will log a warning but download the data anyway.
- **on_update_exec**: If any files were downloaded during a pass, a command can be executed in after the pass finishes. No default. A typical value is `clamdscan --reload`. Starting with Fangfrisch 1.9.0, this option may be overridden in individual sections, permitting per-provider actions.
- **on_update_timeout**: Timeout for the `on_update_exec` command, in seconds. Default: 30. This option may also be overridden in individual sections.

See here for details about the configuration parser and extended interpolation. Section 5.3 provides additional information on how configuration options are interpreted.

5.1. Default providers

Fangfrisch contains internal defaults for the following providers, listed in alphabetical order:

- InterServer
- Malwarepatrol
- Sanesecurity
- SecuriteInfo
- URLhaus

I have also included "Fangfrisch News" (see Appendix C). Providers can be enabled by specifiying `enabled = yes` in the desired configuration file sections.

IMPORTANT	Please make sure to examine the respective settings for yourself before enabling providers.

The defaults may not suit your personal preferences, and some providers require additional configuration, e.g. an access token.

```
# Example configuration

[DEFAULT]
db_url = sqlite:////var/lib/fangfrisch/db.sqlite

# The following settings are optional. Other sections inherit
# values from DEFAULT and may also overwrite values.

local_directory = /var/lib/clamav
max_size = 5MB
on_update_exec = clamdscan --reload
on_update_timeout = 42

[interserver]
enabled = yes

[malwarepatrol]
enabled = yes
# Replace with your personal Malwarepatrol receipt
receipt = abcd1234

[sanesecurity]
enabled = yes
# Use a non-default mirror
prefix = https://mirror.rollernet.us/sanesecurity/

[securiteinfo]
enabled = yes
# Replace with your personal SecuriteInfo customer ID
```

```
customer_id = abcdef123456

[urlhaus]
enabled = yes
max_size = 2MB
```

5.2. User-defined providers

Fangfrisch is of course not limited to the internal defaults. You can define as many additional virus definition providers as you like. The following defines a fictional provider:

```
[fictionalprovider]
enabled = yes
integrity_check = md5
interval = 90m
prefix = http://fictional-provider.tld/clamav-unofficial/

# Reference the defined prefix in URL definitions. Values in
# other sections can be referenced using ${section:option}.
url_eggs = ${prefix}eggs.ndb
url_spam = ${prefix}spam.hdb

# Override local file name for url_spam
filename_spam = spam_spam_spam_lovely_spam.db

# Execute command after each fresh download from url_eggs
on_update_eggs = echo Fresh eggs in {path}
```

5.3. Semantics

Fangfrisch will scan enabled sections for lines prefixed with `url_` to determine download sources for virus definition files.

- The value of `integrity_check` determines both the expected filename suffix for digests and the hashing mechanism used for verification.
- Local file names will be determined by parsing URLs, but can be manually overridden. To change the file name for `url_xyz`, set `filename_xyz` to the desired value.
- To launch a command after data was downloaded for `url_xyz`, define `on_update_xyz`. The command string may contain a `{path}` placeholder, which will be substituted with the full path of the downloaded file.

You can disable refresh operations for selected URLs by assigning either an empty value or setting it to `url_xyz = disabled`. Note that disabling URLs in this manner does *not* delete any previously downloaded files.

5.4. Proxy support

Fangfrisch relies on the *requests* library to download files, which supports environment variables like `HTTPS_PROXY`. Please refer to section Advanced Usage, subsection Proxies in the *requests* online documentation for details.

6. Preparing the database

After completing the configuration, make sure to create the database structure by running the `initdb` command in a root shell as shown below. Running `--force initdb` will drop existing database tables. For SQLite, deleting the database file is a viable alternative.

```
sudo -u clamav -- fangfrisch --conf /etc/fangfrisch.conf initdb
```

IMPORTANT	Fangfrisch need never be run as root. Choose an unprivileged user instead (typically **clamav**).

7. Usage

You can display command line arguments as follows:

```
$ fangfrisch --help

usage: fangfrisch [-h] [-c CONF] [-f] [-p PROVIDER]
                  {dumpconf,dumpmappings,initdb,refresh}

Update and verify unofficial ClamAV signatures.

positional arguments:
  {dumpconf,dumpmappings,initdb,refresh}

options:
  -h, --help            show this help message and exit
  -c CONF, --conf CONF  configuration file
  -f, --force           force action (default: False)
  -p PROVIDER, --provider PROVIDER
                        provider name filter (regular expression)
```

You can choose among following actions:

- **dumpconf**: Dump the effective configuration to stdout, combining both internal defaults and your own settings. The effective configuration for the example shown in Section 5 is available in Appendix B.
- **dumpmappings**: Dump URL-to-filepath mappings, as recorded in the database refresh log, to stdout. See Appendix D for details.

- **initdb**: Create the database structure. This needs to be run only once, before the first refresh. Using the `--force` option will drop existing tables from the database.
- **refresh**: Refresh the configured URLs. The `--force` option can be used to override download interval settings.

As stated before, Fangfrisch is typically run using cron. Depending on your host OS, you may use a systemd timer as an alternative. An example crontab looks like this:

```
HOME=/var/lib/fangfrisch
LOG_LEVEL=INFO
# minute hour day-of-month month day-of-week user command
*/10 * * * * clamav sleep $((RANDOM \% 42)); venv/bin/fangfrisch --conf
/etc/fangfrisch.conf refresh
```

8. Support

The project is hosted on GitHub. Before opening tickets or contacting the author, *always* check existing issues first, including closed ones. This is not meant to discourage you; it just saves time and effort for all involved. Please contact the author Ralph Seichter only after having done your "research". Thank you.

8.1. Reporting possible problems

If you experience problems, please start by trying to figure out underlying issues on your own. Running with DEBUG level logging helps with that. Should your efforts fail, consider filing a GitHub issue. Each issue needs to answer the questions listed below.

If you answer question number 1, 2 or 3 with "no", do not file an issue. Please answer all other questions as detailed as you can, within reason.

1. Have you checked the documentation?
2. Have you checked all existing issues, including closed ones?
3. Have you done your personal best to resolve the issue on your own?
4. What exactly did you do?
5. What did you expect to happen?
6. What happened instead?
7. What was your exact setup (operating system, Python core version, Python module versions)?

8.2. Offering suggestions

The list of questions is shorter, but important nonetheless:

1. Have you checked the documentation?

2. Have you checked all existing issues, including closed ones?
3. Do you consider the suggested feature helpful for more people than just yourself?

If you answered "yes" for all questions, please explain your idea in a sufficiently thorough manner. Use examples, graphics, and whatever else you think would help others to understand your suggestion.

8.3. Discussion precedes contribution

I do not accept any code or documentation contributions which have not been previously agreed upon. This means the order of steps is as follows:

1. Open an issue or discussion topic first.
2. Have a conversation about the matter at hand.
3. Reach a consensus about both form and content of the contribution.
4. Wait for a yay/nay signal from me.

Please do *not* submit any pull requests before I ask for them, as they will be closed. This outlined, proven practice will save time for all involved. See Heather McNamee's blog Always start with an issue for more information.

Appendix A: Default configuration

Fangfrisch contains the following internal configuration settings as defaults. As a safety measure, all sections are disabled by default. Entries with the `!url_` prefix are included for reference only. These represent data sources which are not recommended for general use. Possible reasons: Some sources have an elevated risk of false positives, are not free to use, or contain legacy data. Enabling a section will not enable these specially prefixed entries.

```
[DEFAULT]
cleanup = automatic
enabled = no
integrity_check = sha256
log_level = WARNING
log_method = console
max_size = 10MB

[fangfrischnews]
interval = 12h
local_directory = /tmp
max_size = 100KB
script = /path/to/fangfrisch-has-news.sh
on_update_exec = [ ! -x ${script} ] || ${script} ${local_directory}
prefix = https://www.seichter.de/fangfrisch/
url_alerts = ${prefix}fangfrisch_alerts.txt
url_news = ${prefix}fangfrisch_news.txt
```

```
[interserver]
interval = 1h
integrity_check = disabled
max_size = 5MB
prefix = http://sigs.interserver.net/
!url_shell_hdb = ${prefix}shell.hdb
!url_shellb_db = ${prefix}shellb.db
url_interserver256 = ${prefix}interserver256.hdb
url_shell_ldb = ${prefix}shell.ldb
filename_shell_ldb = interservershell.ldb
url_topline = ${prefix}interservertopline.db
url_whitelist_fp = ${prefix}whitelist.fp
filename_whitelist_fp = interserverwhitelist.fp

[malwarepatrol]
interval = 1d
integrity_check = disabled
product = 8
receipt = you_forgot_to_configure_receipt
prefix =
https://lists.malwarepatrol.net/cgi/getfile?product=${product}&receipt=${receipt}&list
=
url_clamav_basic = ${prefix}clamav_basic
filename_clamav_basic = malwarepatrol.db

[sanesecurity]
interval = 1h
prefix = http://mirror.sentries.org/sanesecurity/
!url_foxhole_all_cdb = ${prefix}foxhole_all.cdb
!url_foxhole_all_ndb = ${prefix}foxhole_all.ndb
!url_foxhole_links = ${prefix}foxhole_links.ldb
!url_foxhole_mail = ${prefix}foxhole_mail.cdb
!url_winnow_phish_complete = ${prefix}winnow_phish_complete.ndb
url_badmacro = ${prefix}badmacro.ndb
url_blurl = ${prefix}blurl.ndb
url_bofhland_cracked_url = ${prefix}bofhland_cracked_URL.ndb
url_bofhland_malware_attach = ${prefix}bofhland_malware_attach.hdb
url_bofhland_malware_url = ${prefix}bofhland_malware_URL.ndb
url_bofhland_phishing_url = ${prefix}bofhland_phishing_URL.ndb
url_foxhole_filename = ${prefix}foxhole_filename.cdb
url_foxhole_generic = ${prefix}foxhole_generic.cdb
url_foxhole_js_cdb = ${prefix}foxhole_js.cdb
url_foxhole_js_ndb = ${prefix}foxhole_js.ndb
url_hackingteam = ${prefix}hackingteam.hsb
url_junk = ${prefix}junk.ndb
url_jurlbl = ${prefix}jurlbl.ndb
url_jurlbla = ${prefix}jurlbla.ndb
url_lott = ${prefix}lott.ndb
url_malwareexpert_fp = ${prefix}malware.expert.fp
url_malwareexpert_hdb = ${prefix}malware.expert.hdb
url_malwareexpert_ldb = ${prefix}malware.expert.ldb
```

```
url_malwareexpert_ndb = ${prefix}malware.expert.ndb
url_malwarehash = ${prefix}malwarehash.hsb
url_phish = ${prefix}phish.ndb
url_phishtank = ${prefix}phishtank.ndb
url_porcupine = ${prefix}porcupine.ndb
url_rogue = ${prefix}rogue.hdb
url_scam = ${prefix}scam.ndb
url_shelter = ${prefix}shelter.ldb
url_sigwhitelist = ${prefix}sigwhitelist.ign2
url_spamattach = ${prefix}spamattach.hdb
url_spamimg = ${prefix}spamimg.hdb
url_spear = ${prefix}spear.ndb
url_spearl = ${prefix}spearl.ndb
url_ssftm = ${prefix}sanesecurity.ftm
url_winnow_attachments = ${prefix}winnow.attachments.hdb
url_winnow_bad_cw = ${prefix}winnow_bad_cw.hdb
url_winnow_extended_malware = ${prefix}winnow_extended_malware.hdb
url_winnow_extended_malware_links = ${prefix}winnow_extended_malware_links.ndb
url_winnow_malware = ${prefix}winnow_malware.hdb
url_winnow_malware_links = ${prefix}winnow_malware_links.ndb
url_winnow_phish_complete_url = ${prefix}winnow_phish_complete_url.ndb
url_winnow_spam_complete = ${prefix}winnow_spam_complete.ndb

[securiteinfo]
customer_id = you_forgot_to_configure_customer_id
interval = 1h
max_size = 20MB
prefix = https://www.securiteinfo.com/get/signatures/${customer_id}/
!url_0hour = ${prefix}securiteinfo0hour.hdb
!url_old = ${prefix}securiteinfoold.hdb
!url_securiteinfo_mdb = ${prefix}securiteinfo.mdb
!url_securiteinfo_pdb = ${prefix}securiteinfo.pdb
!url_securiteinfo_yara = ${prefix}securiteinfo.yara
url_android = ${prefix}securiteinfoandroid.hdb
url_ascii = ${prefix}securiteinfoascii.hdb
url_html = ${prefix}securiteinfohtml.hdb
url_javascript = ${prefix}javascript.ndb
url_pdf = ${prefix}securiteinfopdf.hdb
url_securiteinfo = ${prefix}securiteinfo.hdb
url_securiteinfo_ign2 = ${prefix}securiteinfo.ign2
url_spam_marketing = ${prefix}spam_marketing.ndb

[urlhaus]
interval = 10m
url_urlhaus = https://urlhaus.abuse.ch/downloads/urlhaus.ndb
```

Appendix B: Effective configuration

The following effective configuration is the result of combining internal defaults (see Appendix A)

with the example settings shown in Section 5.

```
[DEFAULT]
cleanup = automatic
enabled = no
integrity_check = sha256
log_level = WARNING
log_method = console
max_size = 5MB
db_url = sqlite:////var/lib/fangfrisch/db.sqlite
local_directory = /var/lib/clamav
on_update_exec = clamdscan --reload
on_update_timeout = 42

[fangfrischnews]
interval = 12h
local_directory = /tmp
max_size = 100KB
script = /path/to/fangfrisch-has-news.sh
on_update_exec = [ ! -x ${script} ] || ${script} ${local_directory}
prefix = https://www.seichter.de/fangfrisch/
url_alerts = ${prefix}fangfrisch_alerts.txt
url_news = ${prefix}fangfrisch_news.txt

[interserver]
interval = 1h
integrity_check = disabled
max_size = 5MB
prefix = http://sigs.interserver.net/
!url_shell_hdb = ${prefix}shell.hdb
!url_shellb_db = ${prefix}shellb.db
url_interserver256 = ${prefix}interserver256.hdb
url_shell_ldb = ${prefix}shell.ldb
filename_shell_ldb = interservershell.ldb
url_topline = ${prefix}interservertopline.db
url_whitelist_fp = ${prefix}whitelist.fp
filename_whitelist_fp = interserverwhitelist.fp
enabled = yes

[malwarepatrol]
interval = 1d
integrity_check = disabled
product = 8
receipt = abcd1234
prefix =
https://lists.malwarepatrol.net/cgi/getfile?product=${product}&receipt=${receipt}&list
=
url_clamav_basic = ${prefix}clamav_basic
filename_clamav_basic = malwarepatrol.db
enabled = yes
```

```
[sanesecurity]
interval = 1h
prefix = https://mirror.rollernet.us/sanesecurity/
!url_foxhole_all_cdb = ${prefix}foxhole_all.cdb
!url_foxhole_all_ndb = ${prefix}foxhole_all.ndb
!url_foxhole_links = ${prefix}foxhole_links.ldb
!url_foxhole_mail = ${prefix}foxhole_mail.cdb
!url_winnow_phish_complete = ${prefix}winnow_phish_complete.ndb
url_badmacro = ${prefix}badmacro.ndb
url_blurl = ${prefix}blurl.ndb
url_bofhland_cracked_url = ${prefix}bofhland_cracked_URL.ndb
url_bofhland_malware_attach = ${prefix}bofhland_malware_attach.hdb
url_bofhland_malware_url = ${prefix}bofhland_malware_URL.ndb
url_bofhland_phishing_url = ${prefix}bofhland_phishing_URL.ndb
url_foxhole_filename = ${prefix}foxhole_filename.cdb
url_foxhole_generic = ${prefix}foxhole_generic.cdb
url_foxhole_js_cdb = ${prefix}foxhole_js.cdb
url_foxhole_js_ndb = ${prefix}foxhole_js.ndb
url_hackingteam = ${prefix}hackingteam.hsb
url_junk = ${prefix}junk.ndb
url_jurlbl = ${prefix}jurlbl.ndb
url_jurlbla = ${prefix}jurlbla.ndb
url_lott = ${prefix}lott.ndb
url_malwareexpert_fp = ${prefix}malware.expert.fp
url_malwareexpert_hdb = ${prefix}malware.expert.hdb
url_malwareexpert_ldb = ${prefix}malware.expert.ldb
url_malwareexpert_ndb = ${prefix}malware.expert.ndb
url_malwarehash = ${prefix}malwarehash.hsb
url_phish = ${prefix}phish.ndb
url_phishtank = ${prefix}phishtank.ndb
url_porcupine = ${prefix}porcupine.ndb
url_rogue = ${prefix}rogue.hdb
url_scam = ${prefix}scam.ndb
url_shelter = ${prefix}shelter.ldb
url_sigwhitelist = ${prefix}sigwhitelist.ign2
url_spamattach = ${prefix}spamattach.hdb
url_spamimg = ${prefix}spamimg.hdb
url_spear = ${prefix}spear.ndb
url_spearl = ${prefix}spearl.ndb
url_ssftm = ${prefix}sanesecurity.ftm
url_winnow_attachments = ${prefix}winnow.attachments.hdb
url_winnow_bad_cw = ${prefix}winnow_bad_cw.hdb
url_winnow_extended_malware = ${prefix}winnow_extended_malware.hdb
url_winnow_extended_malware_links = ${prefix}winnow_extended_malware_links.ndb
url_winnow_malware = ${prefix}winnow_malware.hdb
url_winnow_malware_links = ${prefix}winnow_malware_links.ndb
url_winnow_phish_complete_url = ${prefix}winnow_phish_complete_url.ndb
url_winnow_spam_complete = ${prefix}winnow_spam_complete.ndb
enabled = yes
```

```
[securiteinfo]
customer_id = abcdef123456
interval = 1h
max_size = 20MB
prefix = https://www.securiteinfo.com/get/signatures/${customer_id}/
!url_0hour = ${prefix}securiteinfo0hour.hdb
!url_old = ${prefix}securiteinfoold.hdb
!url_securiteinfo_mdb = ${prefix}securiteinfo.mdb
!url_securiteinfo_pdb = ${prefix}securiteinfo.pdb
!url_securiteinfo_yara = ${prefix}securiteinfo.yara
url_android = ${prefix}securiteinfoandroid.hdb
url_ascii = ${prefix}securiteinfoascii.hdb
url_html = ${prefix}securiteinfohtml.hdb
url_javascript = ${prefix}javascript.ndb
url_pdf = ${prefix}securiteinfopdf.hdb
url_securiteinfo = ${prefix}securiteinfo.hdb
url_securiteinfo_ign2 = ${prefix}securiteinfo.ign2
url_spam_marketing = ${prefix}spam_marketing.ndb
enabled = yes

[urlhaus]
interval = 10m
url_urlhaus = https://urlhaus.abuse.ch/downloads/urlhaus.ndb
enabled = yes
max_size = 2MB
```

Appendix C: Fangfrisch News

This pseudo provider does not offer antivirus signature files. It is only meant for me to pass news about Fangfrisch to you, the user, in lieu of a mailing list. Don't expect this to happen on a regular basis. I am only thinking about release notifications at this point in time. Still, I kindly ask you to please enable this service by adding the following to your configuration file:

```
[fangfrischnews]
enabled = yes
# Uncomment/adapt the following to trigger a script in case of news.
# script = /path/to/script/fangfrisch-has-news.sh
```

This will only enable Fangfrisch to occasionally download small text files, nothing more. You will not need to read the content, nor will anything else happen, unless you modify the `script` option. Here is an example script which you could use to have downloaded news items mailed to you. It needs to be readable for the user account running Fangfrisch, and the sender/recipient addresses obviously have to be adapted. For your convenience, the latest version of the script can be downloaded here.

```
#!/usr/bin/env bash
# vim: ts=4 sw=4 noet ft=sh
```

```
#
# Example script to process Fangfrisch News.

declare -r MAILFROM="noreply"
declare -r MAILTO="alice@example.com"
declare -r SUBJECT="Fangfrisch News are available"

# Choose one of the options below and uncomment the 'declare'
# statements. Until this is done, the script will fail with
# an error message.

# Option 1: Mutt
#declare -r MAILAPP="mutt"
#declare -r MAILAPP_OPT=( "-s" "$SUBJECT" "$MAILTO" )

# Option 2: sendmail
#declare -r MAILAPP="sendmail"
#declare -r MAILAPP_OPT=( "-t" )
#export PATH="$PATH:/usr/sbin"

# Option 3: swaks
#declare -r MAILAPP="swaks"
#declare -r MAILAPP_OPT=( "-d" "-" "-f" "$MAILFROM" "-t" "$MAILTO" )

### No changes required below this line ###

set -euo pipefail

die() {
    echo >&2 "$@"
    exit 1
}

usage() {
    die "Usage: $(basename "$0") {directory}"
}

gen_header() {
    cat <<EOT
From: Fangfrisch News <$MAILFROM>
To: $MAILTO
Subject: $SUBJECT

EOT
    # Mail header must end with an empty line!
}

declare -a NEWSITEMS=()

report_news() {
    local dir=$1 ni
```

```
    [ -d "$dir" ] || die "$dir is not a directory"
    while IFS= read -r -d '' ni; do
        if [ ${#NEWSITEMS[*]} -eq 0 ] && [ "$MAILAPP" != mutt ]; then
            # Mutt does not need the header, others do.
            gen_header
        fi
        NEWSITEMS+=("$ni")
        echo -e "\n### $(basename "$ni"):\n"
        cat "$ni"
    done < <(find "$dir" -maxdepth 1 -type f -name "fangfrisch*.txt" -print0)
}

main() {
    local t
    [ -n "$MAILAPP" ] || die "MAILAPP is undefined, exiting."
    if tty -s; then
        # Running in a terminal session
        t=$(mktemp)
        # shellcheck disable=SC2064
        trap "rm $t" EXIT
        report_news "$@" | tee "$t" || exit 1
        [ ! -s "$t" ] || "$MAILAPP" "${MAILAPP_OPT[@]}" >/dev/null <"$t"
    else
        report_news "$@" 2>&1 | "$MAILAPP" "${MAILAPP_OPT[@]}" >/dev/null
        [ ${#NEWSITEMS[*]} -eq 0 ] || rm -v "${NEWSITEMS[@]}"
    fi
}

[ $# -ge 1 ] || usage
main "$@"
```

Appendix D: Database structure

While users can technically access the Fangfrisch backend database directly, its structure and content are considered **private**. They may change at any time, without notice.

D.1. Accessing mappings

In contrast to direct database access, the **dumpmappings** action allows accessing selected parts of database entries in a stable manner. Specifically, it returns 3-tuples (provider name, URL, local file path). Elements are separated by horizontal tabulators to facilitate piping the output into `awk` or similar utilities. If specified, the `provider` option is interpreted as a regular expression, and only DB records with matching provider column are returned. That means if you have providers *foo* and *foobar*, you need to use anchoring (e.g. `^foo$`) if you only wish to match entries for the former provider. Make sure to use quoting as required by your shell. Example usage:

```
# Print all recorded mappings for the [example] provider section.
```

```
fangfrisch --conf /etc/fangfrisch.conf --provider '^example$' dumpmappings
```

```
# Delete all files that were downloaded by Fangfrisch.
# DON'T EXECUTE THIS UNLESS YOU REALLY MEAN IT!
fangfrisch --conf /etc/fangfrisch.conf dumpmappings | awk '{print $3}' | xargs /bin/rm
```