

# CGITRC(5)

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

---

## Contents

<b>1</b>	<b>NAME</b>	<b>1</b>
<b>2</b>	<b>SYNOPSIS</b>	<b>1</b>
<b>3</b>	<b>LOCATION</b>	<b>1</b>
<b>4</b>	<b>GLOBAL SETTINGS</b>	<b>1</b>
<b>5</b>	<b>REPOSITORY SETTINGS</b>	<b>6</b>
<b>6</b>	<b>REPOSITORY-SPECIFIC CGITRC FILE</b>	<b>8</b>
<b>7</b>	<b>FILTER API</b>	<b>8</b>
<b>8</b>	<b>MACRO EXPANSION</b>	<b>10</b>
<b>9</b>	<b>CACHE</b>	<b>10</b>
<b>10</b>	<b>SIGNATURES</b>	<b>11</b>
<b>11</b>	<b>EXAMPLE CGITRC FILE</b>	<b>11</b>
<b>12</b>	<b>BUGS</b>	<b>14</b>
<b>13</b>	<b>AUTHOR</b>	<b>14</b>

---

## 1 NAME

cgitrc - runtime configuration for cgit

## 2 SYNOPSIS

Cgitrc contains all runtime settings for cgit, including the list of git repositories, formatted as a line-separated list of NAME=VALUE pairs. Blank lines, and lines starting with #, are ignored.

## 3 LOCATION

The default location of cgitrc, defined at compile time, is /etc/cgitrc. At runtime, cgit will consult the environment variable CGIT\_CONFIG and, if defined, use its value instead.

## 4 GLOBAL SETTINGS

### about-filter

Specifies a command which will be invoked to format the content of about pages (both top-level and for each repository). The command will get the content of the about-file on its STDIN, the name of the file as the first argument, and the STDOUT from the command will be included verbatim on the about page. Default value: none. See also: "FILTER API".

### agefile

Specifies a path, relative to each repository path, which can be used to specify the date and time of the youngest commit in the repository. The first line in the file is used as input to the "parse\_date" function in libgit. Recommended timestamp-format is "yyyy-mm-dd hh:mm:ss". You may want to generate this file from a post-receive hook. Default value: "info/web/last-modified".

### auth-filter

Specifies a command that will be invoked for authenticating repository access. Receives quite a few arguments, and data on both stdin and stdout for authentication processing. Details follow later in this document. If no auth-filter is specified, no authentication is performed. Default value: none. See also: "FILTER API".

### branch-sort

Flag which, when set to "age", enables date ordering in the branch ref list, and when set to "name" enables ordering by branch name. Default value: "name".

### cache-about-ttl

Number which specifies the time-to-live, in minutes, for the cached version of the repository about page. See also: "CACHE". Default value: "15".

### cache-dynamic-ttl

Number which specifies the time-to-live, in minutes, for the cached version of repository pages accessed without a fixed SHA1. See also: "CACHE". Default value: "5".

### cache-repo-ttl

Number which specifies the time-to-live, in minutes, for the cached version of the repository summary page. See also: "CACHE". Default value: "5".

### cache-root

Path used to store the cgit cache entries. Default value: "/var/cache/cgit". See also: "MACRO EXPANSION".

### cache-root-ttl

Number which specifies the time-to-live, in minutes, for the cached version of the repository index page. See also: "CACHE". Default value: "5".

---

**cache-scanrc-ttl**

Number which specifies the time-to-live, in minutes, for the result of scanning a path for git repositories. See also: "CACHE". Default value: "15".

**case-sensitive-sort**

Sort items in the repo list case sensitively. Default value: "1". See also: repository-sort, section-sort.

**cache-size**

The maximum number of entries in the cgит cache. When set to "0", caching is disabled. See also: "CACHE". Default value: "0"

**cache-snapshot-ttl**

Number which specifies the time-to-live, in minutes, for the cached version of snapshots. See also: "CACHE". Default value: "5".

**cache-static-ttl**

Number which specifies the time-to-live, in minutes, for the cached version of repository pages accessed with a fixed SHA1. See also: "CACHE". Default value: -1".

**clone-prefix**

Space-separated list of common prefixes which, when combined with a repository url, generates valid clone urls for the repository. This setting is only used if `repo.clone-url` is unspecified. Default value: none.

**clone-url**

Space-separated list of clone-url templates. This setting is only used if `repo.clone-url` is unspecified. Default value: none. See also: "MACRO EXPANSION", "FILTER API".

**commit-filter**

Specifies a command which will be invoked to format commit messages. The command will get the message on its STDIN, and the STDOUT from the command will be included verbatim as the commit message, i.e. this can be used to implement bugtracker integration. Default value: none. See also: "FILTER API".

**commit-sort**

Flag which, when set to "date", enables strict date ordering in the commit log, and when set to "topo" enables strict topological ordering. If unset, the default ordering of "git log" is used. Default value: unset.

**css**

Url which specifies the css document to include in all cgит pages. Default value: "/cgит.css".

**email-filter**

Specifies a command which will be invoked to format names and email address of committers, authors, and taggers, as represented in various places throughout the cgит interface. This command will receive an email address and an origin page string as its command line arguments, and the text to format on STDIN. It is to write the formatted text back out onto STDOUT. Default value: none. See also: "FILTER API".

**embedded**

Flag which, when set to "1", will make cgит generate a html fragment suitable for embedding in other html pages. Default value: none. See also: "noheader".

**enable-blame**

Flag which, when set to "1", will allow cgит to provide a "blame" page for files, and will make it generate links to that page in appropriate places. Default value: "0".

**enable-commit-graph**

Flag which, when set to "1", will make cgит print an ASCII-art commit history graph to the left of the commit messages in the repository log page. Default value: "0".

**enable-filter-overrides**

Flag which, when set to "1", allows all filter settings to be overridden in repository-specific cgитrc files. Default value: none.

**enable-follow-links**

Flag which, when set to "1", allows users to follow a file in the log view. Default value: "0".

---

**enable-git-config**

Flag which, when set to "1", will allow cgkit to use git config to set any repo specific settings. This option is used in conjunction with "scan-path", and must be defined prior, to augment repo-specific settings. The keys gitweb.owner, gitweb.category, gitweb.description, and gitweb.homepage will map to the cgkit keys repo.owner, repo.section, repo.desc, and repo.homepage respectively. All git config keys that begin with "cgkit." will be mapped to the corresponding "repo." key in cgkit. Default value: "0". See also: scan-path, section-from-path.

**enable-http-clone**

If set to "1", cgkit will act as a dumb HTTP endpoint for git clones. You can add "http://\$HTTP\_HOST\$SCRIPT\_NAME/\$CGIT\_REPO\_NAME" to clone-url to expose this feature. If you use an alternate way of serving git repositories, you may wish to disable this. Default value: "1".

**enable-html-serving**

Flag which, when set to "1", will allow the /plain handler to serve mimetype headers that result in the file being treated as HTML by the browser. When set to "0", such file types are returned instead as text/plain or application/octet-stream. Default value: "0". See also: "repo.enable-html-serving".

**enable-index-links**

Flag which, when set to "1", will make cgkit generate extra links for each repo in the repository index (specifically, to the "summary", "commit" and "tree" pages). Default value: "0".

**enable-index-owner**

Flag which, when set to "1", will make cgkit display the owner of each repo in the repository index. Default value: "1".

**enable-log-filecount**

Flag which, when set to "1", will make cgkit print the number of modified files for each commit on the repository log page. Default value: "0".

**enable-log-linecount**

Flag which, when set to "1", will make cgkit print the number of added and removed lines for each commit on the repository log page. Default value: "0".

**enable-remote-branches**

Flag which, when set to "1", will make cgkit display remote branches in the summary and refs views. Default value: "0". See also: "repo.enable-remote-branches".

**enable-subject-links**

Flag which, when set to "1", will make cgkit use the subject of the parent commit as link text when generating links to parent commits in commit view. Default value: "0". See also: "repo.enable-subject-links".

**enable-tree-linenumbers**

Flag which, when set to "1", will make cgkit generate linenummer links for plaintext blobs printed in the tree view. Default value: "1".

**favicon**

Url used as link to a shortcut icon for cgkit. It is suggested to use the value "/favicon.ico" since certain browsers will ignore other values. Default value: "/favicon.ico".

**footer**

The content of the file specified with this option will be included verbatim at the bottom of all pages (i.e. it replaces the standard "generated by..." message. Default value: none.

**head-include**

The content of the file specified with this option will be included verbatim in the html HEAD section on all pages. Default value: none.

**header**

The content of the file specified with this option will be included verbatim at the top of all pages. Default value: none.

**include**

Name of a configfile to include before the rest of the current config- file is parsed. Default value: none. See also: "MACRO EXPANSION".

---

**local-time**

Flag which, if set to "1", makes cgит print commit and tag times in the servers timezone. Default value: "0".

**logo**

Url which specifies the source of an image which will be used as a logo on all cgит pages. Default value: "/cgит.png".

**logo-link**

Url loaded when clicking on the cgит logo image. If unspecified the calculated url of the repository index page will be used. Default value: none.

**max-atom-items**

Specifies the number of items to display in atom feeds view. Default value: "10".

**max-blob-size**

Specifies the maximum size of a blob to display HTML for in KBytes. Default value: "0" (limit disabled).

**max-commit-count**

Specifies the number of entries to list per page in "log" view. Default value: "50".

**max-message-length**

Specifies the maximum number of commit message characters to display in "log" view. Default value: "80".

**max-repo-count**

Specifies the number of entries to list per page on the repository index page. Default value: "50".

**max-repodesc-length**

Specifies the maximum number of repo description characters to display on the repository index page. Default value: "80".

**max-stats**

Set the default maximum statistics period. Valid values are "week", "month", "quarter" and "year". If unspecified, statistics are disabled. Default value: none. See also: "repo.max-stats".

**mimetype.<ext>**

Set the mimetype for the specified filename extension. This is used by the `plain` command when returning blob content.

**mimetype-file**

Specifies the file to use for automatic mimetype lookup. If specified then this field is used as a fallback when no "mimetype.<ext>" match is found. If unspecified then no such lookup is performed. The typical file to use on a Linux system is `/etc/mime.types`. The format of the file must comply to:

- a comment line is an empty line or a line starting with a hash (#), optionally preceded by whitespace
- a non-comment line starts with the mimetype (like `image/png`), followed by one or more file extensions (like `jpg`), all separated by whitespace. Default value: none. See also: "mimetype.<ext>".

**module-link**

Text which will be used as the formatstring for a hyperlink when a submodule is printed in a directory listing. The arguments for the formatstring are the path and SHA1 of the submodule commit. Default value: none.

**noplainemail**

If set to "1" showing full author email addresses will be disabled. Default value: "0".

**noheader**

Flag which, when set to "1", will make cgит omit the standard header on all pages. Default value: none. See also: "embedded".

**owner-filter**

Specifies a command which will be invoked to format the Owner column of the main page. The command will get the owner on STDIN, and the STDOUT from the command will be included verbatim in the table. This can be used to link to additional context such as an owners home page. When active this filter is used instead of the default owner query url. Default value: none. See also: "FILTER API".

---

**project-list**

A list of subdirectories inside of scan-path, relative to it, that should be loaded as git repositories. This must be defined prior to scan-path. Default value: none. See also: scan-path, "MACRO EXPANSION".

**readme**

Text which will be used as default value for "repo.readme". Multiple config keys may be specified, and cgit will use the first found file in this list. This is useful in conjunction with scan-path. Default value: none. See also: scan-path, repo.readme.

**remove-suffix**

If set to "1" and scan-path is enabled, if any repositories are found with a suffix of ".git", this suffix will be removed for the url and name. This must be defined prior to scan-path. Default value: "0". See also: scan-path.

**renamelimit**

Maximum number of files to consider when detecting renames. The value "-1" uses the compiletime value in git (for further info, look at `man git-diff`). Default value: "-1".

**repository-sort**

The way in which repositories in each section are sorted. Valid values are "name" for sorting by the repo name or "age" for sorting by the most recently updated repository. Default value: "name". See also: section, case-sensitive-sort, section-sort.

**robots**

Text used as content for the "robots" meta-tag. Default value: "index, nofollow".

**root-desc**

Text printed below the heading on the repository index page. Default value: "a fast webinterface for the git dscm".

**root-readme**

The content of the file specified with this option will be included verbatim below the "about" link on the repository index page. Default value: none.

**root-title**

Text printed as heading on the repository index page. Default value: "Git Repository Browser".

**scan-hidden-path**

If set to "1" and scan-path is enabled, scan-path will recurse into directories whose name starts with a period (.). Otherwise, scan-path will stay away from such directories (considered as "hidden"). Note that this does not apply to the ".git" directory in non-bare repos. This must be defined prior to scan-path. Default value: 0. See also: scan-path.

**scan-path**

A path which will be scanned for repositories. If caching is enabled, the result will be cached as a cgitrc include-file in the cache directory. If project-list has been defined prior to scan-path, scan-path loads only the directories listed in the file pointed to by project-list. Be advised that only the global settings taken before the scan-path directive will be applied to each repository. Default value: none. See also: cache-scanrc-ttl, project-list, "MACRO EXPANSION".

**section**

The name of the current repository section - all repositories defined after this option will inherit the current section name. Default value: none.

**section-sort**

Flag which, when set to "1", will sort the sections on the repository listing by name. Set this flag to "0" if the order in the cgitrc file should be preserved. Default value: "1". See also: section, case-sensitive-sort, repository-sort.

**section-from-path**

A number which, if defined prior to scan-path, specifies how many path elements from each repo path to use as a default section name. If negative, cgit will discard the specified number of path elements above the repo directory. Default value: "0".

**side-by-side-diffs**

If set to "1" shows side-by-side diffs instead of unidiffs per default. Default value: "0".

---

**snapshots**

Text which specifies the default set of snapshot formats that `cgit` generates links for. The value is a space-separated list of zero or more of the values "tar", "tar.gz", "tar.bz2", "tar.xz" and "zip". The special value "all" enables all snapshot formats. Default value: none.

**source-filter**

Specifies a command which will be invoked to format plaintext blobs in the tree view. The command will get the blob content on its STDIN and the name of the blob as its only command line argument. The STDOUT from the command will be included verbatim as the blob contents, i.e. this can be used to implement e.g. syntax highlighting. Default value: none. See also: "FILTER API".

**summary-branches**

Specifies the number of branches to display in the repository "summary" view. Default value: "10".

**summary-log**

Specifies the number of log entries to display in the repository "summary" view. Default value: "10".

**summary-tags**

Specifies the number of tags to display in the repository "summary" view. Default value: "10".

**strict-export**

Filename which, if specified, needs to be present within the repository for `cgit` to allow access to that repository. This can be used to emulate `gitweb`'s `EXPORT_OK` and `STRICT_EXPORT` functionality and limit `cgit`'s repositories to match those exported by `git-daemon`. This option must be defined prior to `scan-path`.

**virtual-root**

Url which, if specified, will be used as root for all `cgit` links. It will also cause `cgit` to generate *virtual urls*, i.e. urls like `/cgit/tree/README` as opposed to `?r=cgit&p=tree&path=README`. Default value: none. NOTE: `cgit` has recently learned how to use `PATH_INFO` to achieve the same kind of virtual urls, so this option will probably be deprecated.

## 5 REPOSITORY SETTINGS

**repo.about-filter**

Override the default about-filter. Default value: none. See also: "enable-filter-overrides". See also: "FILTER API".

**repo.branch-sort**

Flag which, when set to "age", enables date ordering in the branch ref list, and when set to "name" enables ordering by branch name. Default value: "name".

**repo.clone-url**

A list of space-separated urls which can be used to clone this repo. Default value: none. See also: "MACRO EXPANSION".

**repo.commit-filter**

Override the default commit-filter. Default value: none. See also: "enable-filter-overrides". See also: "FILTER API".

**repo.commit-sort**

Flag which, when set to "date", enables strict date ordering in the commit log, and when set to "topo" enables strict topological ordering. If unset, the default ordering of "git log" is used. Default value: unset.

**repo.defbranch**

The name of the default branch for this repository. If no such branch exists in the repository, the first branch name (when sorted) is used as default instead. Default value: branch pointed to by HEAD, or "master" if there is no suitable HEAD.

**repo.desc**

The value to show as repository description. Default value: none.

**repo.email-filter**

Override the default email-filter. Default value: none. See also: "enable-filter-overrides". See also: "FILTER API".

---

**repo.enable-commit-graph**

A flag which can be used to disable the global setting 'enable-commit-graph'. Default value: none.

**repo.enable-html-serving**

A flag which can be used to override the global setting `enable-html-serving`. Default value: none.

**repo.enable-log-filecount**

A flag which can be used to disable the global setting 'enable-log-filecount'. Default value: none.

**repo.enable-log-linecount**

A flag which can be used to disable the global setting 'enable-log-linecount'. Default value: none.

**repo.enable-remote-branches**

Flag which, when set to "1", will make cgkit display remote branches in the summary and refs views. Default value: `<enable-remote-branches>`.

**repo.enable-subject-links**

A flag which can be used to override the global setting 'enable-subject-links'. Default value: none.

**repo.extra-head-content**

This value will be added verbatim to the head section of each page displayed for this repo. Default value: none.

**repo.hide**

Flag which, when set to "1", hides the repository from the repository index. The repository can still be accessed by providing a direct path. Default value: "0". See also: "repo.ignore".

**repo.homepage**

The value to show as repository homepage. Default value: none.

**repo.ignore**

Flag which, when set to "1", ignores the repository. The repository is not shown in the index and cannot be accessed by providing a direct path. Default value: "0". See also: "repo.hide".

**repo.logo**

Url which specifies the source of an image which will be used as a logo on this repo's pages. Default value: global logo.

**repo.logo-link**

Url loaded when clicking on the cgkit logo image. If unspecified the calculated url of the repository index page will be used. Default value: global logo-link.

**repo.module-link**

Text which will be used as the formatstring for a hyperlink when a submodule is printed in a directory listing. The arguments for the formatstring are the path and SHA1 of the submodule commit. Default value: `<module-link>`

**repo.module-link.<path>**

Text which will be used as the formatstring for a hyperlink when a submodule with the specified subdirectory path is printed in a directory listing. The only argument for the formatstring is the SHA1 of the submodule commit. Default value: none.

**repo.max-stats**

Override the default maximum statistics period. Valid values are equal to the values specified for the global "max-stats" setting. Default value: none.

**repo.name**

The value to show as repository name. Default value: `<repo.url>`.

**repo.owner**

A value used to identify the owner of the repository. Default value: none.

**repo.owner-filter**

Override the default owner-filter. Default value: none. See also: "enable-filter-overrides". See also: "FILTER API".

**repo.path**

An absolute path to the repository directory. For non-bare repositories this is the `.git-directory`. Default value: none.

---

**repo.readme**

A path (relative to <repo.path>) which specifies a file to include verbatim as the "About" page for this repo. You may also specify a git refspec by head or by hash by prepending the refspec followed by a colon. For example, "master:docs/readme.mkd". If the value begins with a colon, i.e. ":docs/readme.rst", the default branch of the repository will be used. Sharing any file will expose that entire directory tree to the "/about/PATH" endpoints, so be sure that there are no non-public files located in the same directory as the readme file. Default value: <readme>.

**repo.section**

Override the current section name for this repository. Default value: none.

**repo.snapshots**

A mask of snapshot formats for this repo that cgit generates links for, restricted by the global "snapshots" setting. Default value: <snapshots>.

**repo.snapshot-prefix**

Prefix to use for snapshot links instead of the repository basename. For example, the "linux-stable" repository may wish to set this to "linux" so that snapshots are in the format "linux-3.15.4" instead of "linux-stable-3.15.4". Default value: <empty> meaning to use the repository basename.

**repo.source-filter**

Override the default source-filter. Default value: none. See also: "enable-filter-overrides". See also: "FILTER API".

**repo.url**

The relative url used to access the repository. This must be the first setting specified for each repo. Default value: none.

## 6 REPOSITORY-SPECIFIC CGITRC FILE

When the option "scan-path" is used to auto-discover git repositories, cgit will try to parse the file "cgitrc" within any found repository. Such a repo-specific config file may contain any of the repo-specific options described above, except "repo.url" and "repo.path". Additionally, the "filter" options are only acknowledged in repo-specific config files when "enable-filter-overrides" is set to "1".

Note: the "repo." prefix is dropped from the option names in repo-specific config files, e.g. "repo.desc" becomes "desc".

## 7 FILTER API

By default, filters are separate processes that are executed each time they are needed. Alternative technologies may be used by prefixing the filter specification with the relevant string; available values are:

**exec:**

The default "one process per filter" mode.

**lua:**

Executes the script using a built-in Lua interpreter. The script is loaded once per execution of cgit, and may be called multiple times during cgit's lifetime, making it a good choice for repeated filters such as the *email filter*. It responds to three functions:

***filter\_open(argument1, argument2, argument3, ...)***

This is called upon activation of the filter for a particular set of data.

***filter\_write(buffer)***

This is called whenever cgit writes data to the webpage.

***filter\_close()***

This is called when the current filtering operation is completed. It must return an integer value. Usually 0 indicates success.

Additionally, cgit exposes to the Lua the following built-in functions:

***html(str)***

Writes *str* to the webpage.

***html\_txt(str)***

HTML escapes and writes *str* to the webpage.

***html\_attr(str)***

HTML escapes for an attribute and writes "str" to the webpage.

***html\_url\_path(str)***

URL escapes for a path and writes *str* to the webpage.

***html\_url\_arg(str)***

URL escapes for an argument and writes *str* to the webpage.

***html\_include(file)***

Includes *file* in webpage.

Parameters are provided to filters as follows.

**about filter**

This filter is given a single parameter: the filename of the source file to filter. The filter can use the filename to determine (for example) the type of syntax to follow when formatting the readme file. The about text that is to be filtered is available on standard input and the filtered text is expected on standard output.

**auth filter**

The authentication filter receives 12 parameters:

- filter action, explained below, which specifies which action the filter is called for
- http cookie
- http method
- http referer
- http path
- http https flag
- cgit repo
- cgit page
- cgit url
- cgit login url When the filter action is "body", this filter must write to output the HTML for displaying the login form, which POSTs to the login url. When the filter action is "authenticate-cookie", this filter must validate the http cookie and return a 0 if it is invalid or 1 if it is invalid, in the exit code / close function. If the filter action is "authenticate-post", this filter receives POST'd parameters on standard input, and should write a complete CGI response, preferably with a 302 redirect, and write to output one or more "Set-Cookie" HTTP headers, each followed by a newline.

Please see `'filters/simple-authentication.lua'` for a clear example script that may be modified.

**commit filter**

This filter is given no arguments. The commit message text that is to be filtered is available on standard input and the filtered text is expected on standard output.

**email filter**

This filter is given two parameters: the email address of the relevant author and a string indicating the originating page. The filter will then receive the text string to format on standard input and is expected to write to standard output the formatted text to be included in the page.

**owner filter**

This filter is given no arguments. The owner text is available on standard input and the filter is expected to write to standard output. The output is included in the Owner column.

---

**source filter**

This filter is given a single parameter: the filename of the source file to filter. The filter can use the filename to determine (for example) the syntax highlighting mode. The contents of the source file that is to be filtered is available on standard input and the filtered contents is expected on standard output.

All filters are handed the following environment variables:

- CGIT\_REPO\_URL (from repo.url)
- CGIT\_REPO\_NAME (from repo.name)
- CGIT\_REPO\_PATH (from repo.path)
- CGIT\_REPO\_OWNER (from repo.owner)
- CGIT\_REPO\_DEFBRANCH (from repo.defbranch)
- CGIT\_REPO\_SECTION (from repo.section)
- CGIT\_REPO\_CLONE\_URL (from repo.clone-url)

If a setting is not defined for a repository and the corresponding global setting is also not defined (if applicable), then the corresponding environment variable will be unset.

## 8 MACRO EXPANSION

The following cgitrc options support a simple macro expansion feature, where tokens prefixed with "\$" are replaced with the value of a similarly named environment variable:

- cache-root
- include
- project-list
- scan-path

Macro expansion will also happen on the content of \$CGIT\_CONFIG, if defined.

One usage of this feature is virtual hosting, which in its simplest form can be accomplished by adding the following line to `/etc/cgitrc`:

```
include=/etc/cgitrc.d/$HTTP_HOST
```

The following options are expanded during request processing, and support the environment variables defined in "FILTER API":

- clone-url
- repo.clone-url

## 9 CACHE

All cache ttl values are in minutes. Negative ttl values indicate that a page type will never expire, and thus the first time a URL is accessed, the result will be cached indefinitely, even if the underlying git repository changes. Conversely, when a ttl value is zero, the cache is disabled for that particular page type, and the page type is never cached.

---

## 10 SIGNATURES

Cgit can host .asc signatures corresponding to various snapshot formats, through use of git notes. For example, the following command may be used to add a signature to a .tar.xz archive:

```
git notes --ref=refs/notes/signatures/tar.xz add -C "$(
  gpg --output - --armor --detach-sign cgit-1.1.tar.xz |
  git hash-object -w --stdin
)" v1.1
```

If it is instead desirable to attach a signature of the underlying .tar, this will be linked, as a special case, beside a .tar.\* link that does not have its own signature. For example, a signature of a tarball of the latest tag might be added with a similar command:

```
tag="$(git describe --abbrev=0) "
git notes --ref=refs/notes/signatures/tar add -C "$(
  git archive --format tar --prefix "cgit-${tag#v}/" "$tag" |
  gpg --output - --armor --detach-sign |
  git hash-object -w --stdin
)" "$tag"
```

Since git-archive(1) is expected to produce stable output between versions, this allows one to generate a long-term signature of the contents of a given tag.

## 11 EXAMPLE CGITRC FILE

```
# Enable caching of up to 1000 output entries
cache-size=1000

# Specify some default clone urls using macro expansion
clone-url=git://foo.org/$CGIT_REPO_URL git@foo.org:$CGIT_REPO_URL

# Specify the css url
css=/css/cgit.css

# Show owner on index page
enable-index-owner=1

# Allow http transport git clone
enable-http-clone=1

# Show extra links for each repository on the index page
enable-index-links=1

# Enable blame page and create links to it from tree page
enable-blame=1

# Enable ASCII art commit history graph on the log pages
enable-commit-graph=1
```

---

```
# Show number of affected files per commit on the log pages
enable-log-filecount=1

# Show number of added/removed lines per commit on the log pages
enable-log-linecount=1

# Sort branches by date
branch-sort=age

# Add a cgkit favicon
favicon=/favicon.ico

# Use a custom logo
logo=/img/mylogo.png

# Enable statistics per week, month and quarter
max-stats=quarter

# Set the title and heading of the repository index page
root-title=example.com git repositories

# Set a subheading for the repository index page
root-desc=tracking the foobar development

# Include some more info about example.com on the index page
root-readme=/var/www/htdocs/about.html

# Allow download of tar.gz, tar.bz2 and zip-files
snapshots=tar.gz tar.bz2 zip

##
## List of common mimetypes
##

mimetype.gif=image/gif
mimetype.html=text/html
mimetype.jpg=image/jpeg
mimetype.jpeg=image/jpeg
mimetype.pdf=application/pdf
mimetype.png=image/png
mimetype.svg=image/svg+xml

# Highlight source code with python pygments-based highlighter
source-filter=/var/www/cgit/filters/syntax-highlighting.py
```

---

```
# Format markdown, restructuredtext, manpages, text files, and html files
# through the right converters
about-filter=/var/www/cgit/filters/about-formatting.sh

##
## Search for these files in the root of the default branch of repositories
## for coming up with the about page:
##
readme=:README.md
readme=:readme.md
readme=:README.mkd
readme=:readme.mkd
readme=:README.rst
readme=:readme.rst
readme=:README.html
readme=:readme.html
readme=:README.htm
readme=:readme.htm
readme=:README.txt
readme=:readme.txt
readme=:README
readme=:readme
readme=:INSTALL.md
readme=:install.md
readme=:INSTALL.mkd
readme=:install.mkd
readme=:INSTALL.rst
readme=:install.rst
readme=:INSTALL.html
readme=:install.html
readme=:INSTALL.htm
readme=:install.htm
readme=:INSTALL.txt
readme=:install.txt
readme=:INSTALL
readme=:install

##
## List of repositories.
## PS: Any repositories listed when section is unset will not be
## displayed under a section heading
## PPS: This list could be kept in a different file (e.g. '/etc/cgitrepos')
## and included like this:
## include=/etc/cgitrepos
##

repo.url=foo
repo.path=/pub/git/foo.git
repo.desc=the master foo repository
repo.owner=fooman@example.com
repo.readme=info/web/about.html

repo.url=bar
repo.path=/pub/git/bar.git
```

---

```
repo.desc=the bars for your foo
repo.owner=barman@example.com
repo.readme=info/web/about.html

# The next repositories will be displayed under the 'extras' heading
section=extras

repo.url=baz
repo.path=/pub/git/baz.git
repo.desc=a set of extensions for bar users

repo.url=wiz
repo.path=/pub/git/wiz.git
repo.desc=the wizard of foo

# Add some mirrored repositories
section=mirrors

repo.url=git
repo.path=/pub/git/git.git
repo.desc=the dscm

repo.url=linux
repo.path=/pub/git/linux.git
repo.desc=the kernel

# Disable adhoc downloads of this repo
repo.snapshots=0

# Disable line-counts for this repo
repo.enable-log-linecount=0

# Restrict the max statistics period for this repo
repo.max-stats=month
```

## 12 BUGS

Comments currently cannot appear on the same line as a setting; the comment will be included as part of the value. E.g. this line:

```
robots=index # allow indexing
```

will generate the following html element:

```
<meta name='robots' content='index # allow indexing' />
```

## 13 AUTHOR

Lars Hjemli <[hjemli@gmail.com](mailto:hjemli@gmail.com)> Jason A. Donenfeld <[Jason@zx2c4.com](mailto:Jason@zx2c4.com)>

---