NAME

git-help − Display help information about Git

SYNOPSIS

git help [−a|−−all [−−[no−]verbose]]
            [[−i|−−info] [−m|−−man] [−w|−−web]] [COMMAND|GUIDE]
git help [−g|−−guides]
git help [−c|−−config]

DESCRIPTION

With no options and no COMMAND or GUIDE given, the synopsis of the *git* command and a list of the most commonly used Git commands are printed on the standard output.

If the option −−**all** or −**a** is given, all available commands are printed on the standard output.

If the option −−**guides** or −**g** is given, a list of the Git concept guides is also printed on the standard output.

If a command, or a guide, is given, a manual page for that command or guide is brought up. The *man* program is used by default for this purpose, but this can be overridden by other options or configuration variables.

If an alias is given, git shows the definition of the alias on standard output. To get the manual page for the aliased command, use **git COMMAND −−help**.

Note that **git −−help ...** is identical to **git help ...** because the former is internally converted into the latter.

To display the **git**(1) man page, use **git help git**.

This page can be displayed with *git help help* or **git help −−help**

OPTIONS

−a, −−all
    Prints all the available commands on the standard output. This option overrides any given command or guide name.

−−verbose
    When used with −−**all** print description for all recognized commands. This is the default.

−c, −−config
    List all available configuration variables. This is a short summary of the list in **git-config**(1).

−g, −−guides
    Prints a list of the Git concept guides on the standard output.

−i, −−info
    Display manual page for the command in the *info* format. The *info* program will be used for that purpose.

−m, −−man
    Display manual page for the command in the *man* format. This option may be used to override a value set in the **help.format** configuration variable.

    By default the *man* program will be used to display the manual page, but the **man.viewer** configuration variable may be used to choose other display programs (see below).

−w, −−web
    Display manual page for the command in the *web* (HTML) format. A web browser will be used for that purpose.

The web browser can be specified using the configuration variable **help.browser**, or **web.browser** if the former is not set. If none of these config variables is set, the *git web−−browse* helper script (called by *git help*) will pick a suitable default. See **git-web--browse**(1) for more information about this.

CONFIGURATION VARIABLES

help.format

If no command−line option is passed, the **help.format** configuration variable will be checked. The following values are supported for this variable; they make *git help* behave as their corresponding command− line option:

- "man" corresponds to −*m*/−−*man*,
- "info" corresponds to −*i*/−−*info*,
- "web" or "html" correspond to −*w*/−−*web*.

help.browser, web.browser and browser.<tool>.path

The **help.browser**, **web.browser** and **browser.<tool>.path** will also be checked if the *web* format is chosen (either by command−line option or configuration variable). See −*w*/−−*web* in the OPTIONS section above and **git-web--browse**(1).

man.viewer

The **man.viewer** configuration variable will be checked if the *man* format is chosen. The following values are currently supported:

- "man": use the *man* program as usual,
- "woman": use *emacsclient* to launch the "woman" mode in emacs (this only works starting with emacsclient versions 22),
- "konqueror": use *kfmclient* to open the man page in a new konqueror tab (see *Note about konqueror* below).

Values for other tools can be used if there is a corresponding **man.<tool>.cmd** configuration entry (see below).

Multiple values may be given to the **man.viewer** configuration variable. Their corresponding programs will be tried in the order listed in the configuration file.

For example, this configuration:

```
[man]
        viewer = konqueror
        viewer = woman
```

will try to use konqueror first. But this may fail (for example, if DISPLAY is not set) and in that case emacs' woman mode will be tried.

If everything fails, or if no viewer is configured, the viewer specified in the **GIT_MAN_VIEWER** environment variable will be tried. If that fails too, the *man* program will be tried anyway.

man.<tool>.path

You can explicitly provide a full path to your preferred man viewer by setting the configuration variable **man.<tool>.path**. For example, you can configure the absolute path to konqueror by setting *man.konqueror.path*. Otherwise, *git help* assumes the tool is available in PATH.

man.<tool>.cmd

When the man viewer, specified by the **man.viewer** configuration variables, is not among the supported ones, then the corresponding **man.<tool>.cmd** configuration variable will be looked up. If this variable exists then the specified tool will be treated as a custom command and a shell eval will be used to run the

command with the man page passed as arguments.

Note about konqueror

When *konqueror* is specified in the **man.viewer** configuration variable, we launch *kfmclient* to try to open the man page on an already opened konqueror in a new tab if possible.

For consistency, we also try such a trick if *man.konqueror.path* is set to something like **A_PATH_TO/konqueror**. That means we will try to launch **A_PATH_TO/kfmclient** instead.

If you really want to use *konqueror*, then you can use something like the following:

```
[man]
        viewer = konq

[man "konq"]
        cmd = A_PATH_TO/konqueror
```

Note about git config −−global

Note that all these configuration variables should probably be set using the **−−global** flag, for example like this:

```
$ git config −−global help.format web
$ git config −−global web.browser firefox
```

as they are probably more user specific than repository specific. See **git-config**(1) for more information about this.

GIT

Part of the **git**(1) suite