

NAME

git-qrefresh – update the current patch

SYNOPSIS

```
git qrefresh [-e] [[[-m <text>] ...] | [-F <file>]] [-s] [<path> ...]
git qrefresh [-e] [[[-m <text>] ...] | [-F <file>]] [-a]
```

DESCRIPTION

The **git qrefresh** command updates the topmost currently-applied patch, to incorporate changes staged in git's index, and optionally, when the **-a** (or **--all**) option is specified, unstaged changes to tracked files in the working tree.

If the **--all** option is *not* specified, and a list of one or more *<path>* arguments *is*, either explicitly as such, or implicitly by means of specifying the **-s** (or **--short**) option, (which has the effect of implicitly naming all files which are already referenced within the patch), the updated content of the patch is restricted to include only the named files.

OPTIONS**-a, --all**

Stage any currently unstaged changes in tracked files, before updating the patch; this option cannot be combined with the **-s** (or **--short**) option, or with any explicit list of *<path>* arguments.

-e, --edit

Open an editor, to facilitate further modification of the amended commit message, (which has been specified by use of **--file**, or **--message** options, or retrieved by default, from the current commit whence the patch is derived), before amending the current commit, (effectively, this option is simply passed on to the **git commit --amend** command which refreshes this commit).

-F <name>, --file=<name>**-l <name>, --logfile=<name>**

Read the text, to be used as the amended commit message, from the file specified by *<name>*; if *<name>* is specified as **-**, read the commit message text from standard input. By convention, git expects the **-F** (or **--file**) variant of this option; the **-l** (or **--logfile**) alias is accepted for compatibility with **hg qrefresh** usage.

-m <text>, --message=<text>

Use *<text>* for the amended commit message. If more than one instance of this option is specified, each individual *<text>* block is appended to the commit message, as a separate paragraph. This feature *cannot* be used in conjunction with the **-F** (or **--file**) option.

-s, --short

Short-list files which are already referenced within the patch, for inclusion in a restricted content update; this has the effect of implicitly adding the path names for such files to the (possibly empty) list of explicitly specified *<path>* name arguments.

-D, --currentdate

Use the current date as the commit date, (as would be the normal case, if no commit date has been recorded previously); record it within the patch header, such that it will be preserved as the patch creation date, unless explicitly reset, during any subsequent **git qrefresh**, or **git qimport** of this patch; this has the effect of overriding any commit date which may have been recorded previously, within the patch header.

-d <datespec>, --date=<datespec>

Set the commit date, and hence the patch creation date, to *<datespec>*, (which may be specified using any date format recognized by **git commit**); record this within the patch header, such that it will be preserved as the patch creation date, unless explicitly reset, during any subsequent **git qrefresh**, or **git qimport** of this patch; this has the effect of overriding any commit date which may have been recorded previously, within the patch header.

-U, --currentuser

Record the identity of the current user as the author of the patch, and of its associated commit, such that this will be preserved, unless explicitly reset, during any subsequent **git qrefresh**, or **git qimport** of this patch; this has the effect of overriding any author identification which may have been recorded previously, within the patch header.

-u <userspec>, --user=<userspec>

Record the identity of the user specified by *<userspec>*, (which may be specified in any form recognized by **git commit**), as the author of the patch, and of its associated commit, such that this will be preserved, unless explicitly reset, during any subsequent **git qrefresh**, or **git qimport** of this patch; this has the effect of overriding any author identification which may have been recorded previously, within the patch header.

EXIT STATUS

On successful completion, **git qrefresh** reports an exit status code of 0. Any non-zero exit status code indicates that an error occurred.

COMPARISON WITH MERCURIAL QUEUES

Although inspired by, and for the most part based on behavioural observation of **Mercurial's MQ** extension, the behaviour of the **git qrefresh** command differs from that of its **hg qrefresh** counterpart in the following respects:—

- The **--include**, and **--exclude** options, which are supported by **hg qrefresh**, are not (currently) supported by **git qrefresh**.
- The **--all** option, described above, is neither supported by, nor required by **hg qrefresh**; it is provided, as a **Git-MQ** specific extension to the **Mercurial Queues** model, to achieve more flexible interaction with git's index. The command: “**git qrefresh --all**” will deliver behaviour which is effectively equivalent to that of the “**hg qrefresh**” command.

AUTHOR

Copyright (C) 2019, by Keith Marshall

This man page was written by Keith Marshall <keith@users.osdn.me>, to accompany the **Git-MQ** program suite. It is published under the terms of the GNU Free Documentation Licence, 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.

The **Git-MQ** program suite itself is distributed under the terms of the GNU General Public Licence, version 3, (or any later version published by the Free Software Foundation).

Copies of the GNU Free Documentation Licence, and of the GNU General Public Licence, are included within the **Git-MQ** source distribution, in the files **FDL-1.3**, and **LICENCE**, respectively.

SEE ALSO

git-commit(1), **git-qimport(1)**