NAME

git−qnew − create a new patch and add it to the patch series

SYNOPSIS

git qnew [**−e**] [[[**−m** <*text*>] **...**] | [**−F** <*file*>]] [**−a**] <*patch*>
git qnew [**−e**] [[[**−m** <*text*>] **...**] | [**−F** <*file*>]] <*patch*> [<*path*> **...**]

DESCRIPTION

The **git qnew** command creates a new patch, based on a new git commit, and pushes it on to the top of the **Git−MQ** patch series stack, immediately following the topmost currently-applied patch, (if any). The <*patch*> argument specifies the user's chosen name for the patch; this will also be used as a tag name for the associated patch commit, so must be chosen to comply with git's rules for naming tags.

By default, the content of the patch is defined by changes which are staged in git's index; this behaviour may be modified, either by use of the **−a** (or **−−all**) option, or by specifying a list of *path* name arguments, following the *patch* name, (but never by both together).

If the **−a** (or **−−all**) option is specified, any local changes in tracked files are staged before creating the git commit on which the patch is based; this is effectively equivalent to the effect of, and achieved by, adding the **−−all** option to those specified for the **git commit** command whence the patch is to be derived.

If a list of files is specified, the state of git's index is ignored, and only the specified files are included in the commit on which the patch is based. All files which are specified must already exist within the git working tree, and must be known to, and tracked by git; any which do not exist, or which lie outside the working tree, will be diagnosed, and execution of **git qnew** will be aborted; any which are otherwise unknown to git will be silently filtered out of the list.

The **−e** (or **−−edit**), **−F** (or **−−file**), and **−m** (or **−−message**) options may be used to specify a commit message, which will also be recorded in the patch header. If none of these options is specified, no commit message is recorded in the patch header, and the default message, “[Git−MQ]: <patch>”", is assigned for the associated commit.

OPTIONS

−a, **−−all**
Stage all local modifications to tracked files, such that they will be incorporated into the commit which is to be associated with the created patch, and whence its content is derived.

−e, **−−edit**
Open an editor, to facilitate modification of the commit message, (which has been specified by use of **−−file**, or **−−message** options, or by default), before completing the commit whence the patch is derived; (effectively, this option is simply passed on to the **git commit** command which creates this commit).

−F <*name*>, **−−file=**<*name*>
−l <*name*>, **−−logfile=**<*name*>
Read the text, to be used as the 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 qnew** usage.

−m <*text*>, **−−message=**<*text*>
Use <*text*> for the 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.

−D, **−−currentdate**
Use the current date as the commit date, (as would be the normal case); 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.

−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.

−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.

−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.

EXIT STATUS

On successful completion, **git qnew** 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 qnew** command differs from that of its **hg qnew** counterpart in the following respects:—

- The **−−include**, and **−−exclude** options, which are supported by **hg qnew**, are not (currently) supported by **git qnew**.
- The **−−all** option, described above, is neither supported by, nor required by **hg qnew**; 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 qnew −−all ...** <*patchname*>” will deliver behaviour which is effectively equivalent to that of the “**hg qnew ...** <*patchname*>” command.

AUTHOR

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 distibruted 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), **git−qrefresh**(1)