

NAME

`git-qgoto` – push or pop patches until specified patch is topmost

SYNOPSIS

```
git qgoto [[-f [--no-backup]] | --keep-changes] <patch>
git qgoto [[-f [--no-backup]] | --keep-changes] <index>
```

DESCRIPTION

The **git qgoto** command will either push patches on to, or pop patches from the **Git-MQ** applied patch stack, until the patch specified by the <patch> argument (a patch *name*), or by the <index> argument (a patch *sequence number*, as reported by the **git qseries --verbose** command), becomes the topmost applied patch.

While pushing, or popping patches, **git qgoto** will report the identity of each patch processed; additionally, if the **--verbose** option (which is supported by *all* **Git-MQ** commands) is specified, while pushing patches, it will also report the identity of any guarded patches which are skipped.

Note that, unlike the **git qpop --all** command, **git qgoto** is unable to pop *all* applied patches; it *can* emulate the **git qpush --all** operation, but less conveniently, since it is necessary to pass the name, or sequence number of the last pushable patch, as argument, whereas **git qpush --all** determines this implicitly.

OPTIONS**-f, --force**

If any files, known to and tracked by git, have been locally modified in the working tree, or in git's index, and the **--no-backup** option has not been specified, first make backup backup copies of all locally modified files; discard all local modifications, then push or pop patches, as appropriate.

If the **--force** option is not specified, **git qgoto** will abort in the presence of local modifications.

--no-backup

Used only in conjunction with **--force**, (and silently ignored otherwise), discard all local modifications, *without* making backup copies.

--keep-changes

Attempt to preserve local modifications, while pushing or popping patches; if any local modification conflicts with patch content, abort the entire **git qgoto** operation.

(The **--keep-changes** option is currently unimplemented).

If there are local modifications to tracked files, either in the working tree, or staged in git's index, and if neither the **--force** option, nor the **--keep-changes** option is specified, then the **git qgoto** operation will be aborted.

EXIT STATUS

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

COMPARISON WITH MERCURIAL QUEUES

Inspired by, and for the most part based on behavioural observation of **Mercurial's MQ** extension, the **git qgoto** command exhibits fundamentally the same behaviour as its **hg qgoto** counterpart.

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-qpop(1), **git-qpush(1)**, **git-qseries(1)**