

**NAME**

`git-qprev` – identify the predecessor of the current patch

**SYNOPSIS**

`git qprev` [`-v` | `--verbose`] [`-s` | `--summary`]

**DESCRIPTION**

The `git qprev` command displays the identity of the most recently applied patch, if any, which precedes the topmost currently-applied patch in a **Git-MQ** patch series; this is the patch, if any, which will become topmost, following a “`git qpop`” operation with no arguments specified.

**OPTIONS**

`-s`, `--summary`

Include the summary line from the patch header, if available.

`-v`, `--verbose`

Include the patch sequence number, and the status flag, (which should always be ‘A’), as described for the `git qseries` command.

**EXIT STATUS**

On successful completion, `git qprev` reports an exit status code of 0, when an applied predecessor patch is identified, or of 1 if fewer than one patch has been applied. Any other 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 qprev` command exhibits fundamentally the same behaviour as its `hg qprev` 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-qnext(1)`, `git-qpop(1)`, `git-qseries(1)`, `git-qtop(1)`