NAME

git−qimport − import patches or existing commits

SYNOPSIS

git qimport [**−efP**] [**−n** <*name*>] [[**−r** <*commit*>] **...**] [<*patch*>] **...**]

DESCRIPTION

The **git qimport** command may be used to add new patches, or existing unregistered patches, to the **Git−MQ** patch series; it may also be used to convert existing committed revisions to patches, thus bringing them under **Git−MQ** control.

When importing patches, the <*patch*> argument specifies the name of the patch file which is to be imported. If the **−e** (or **−−existing**) option is specified, the patch file should already be present in the active **Git−MQ** patch directory, and <*patch*> should specify a bare file name; conversely, if the **−e** (or **−−existing**) option is not specified, <*patch*> should specify a path name for the patch file.

If <*patch*> is specified as '−', the patch will be read from standard input; in this case, the **−n**<*name*> (or **−−name**=<*name*>) must be used to assign a name for the imported patch.

Patches must be named to conform with git's rules for naming tags; to import a patch with a non-conforming file name, the **−n**<*name*> (or **−−name**=<*name*>) option must be used to assign a conforming patch name.

If the **−P** (or **−−push**) option is specified, the patch(es) will be pushed (applied) after import. Without this option, patches will be copied to the **Git−MQ** patch directory, if necessary, and will be registered in the series file, but they will not be applied.

If the **−r**<*commit*> (or **−−rev**=<*commit*>) option is specified, then one or more existing commits will be converted to **Git−MQ** patches. The <*commit*> option argument is required; it must specify a (possibly abbreviated) commit ID, or a range of such commit IDs, and aggregate of all such specified ranges must represent one contiguous range, which must not include, or span, any merge commit, and must end at the parent commit of the first applied **Git−MQ** patch, or at the **HEAD** commit, if no **Git−MQ** patches have been applied.

When converting existing commits to patches, the **−P** (or **−−push**) option is ignored; such patches are *always* marked as *applied*.

OPTIONS

−e, **−−existing**
: Register patches for which the patch files are already present in the **Git−MQ** patch directory, but for which no entry yet exists in the series file.

−f, **−−force**
: Overwrite patch files which already exist in the **Git−MQ** patch directory.

−n <*name*>, **−−name**=<*name*>
: Specify the name, under which the patch is to be registered; (this may differ from the base name specified in the <*patch*> argument). This option cannot be used if importing more than one patch, or if converting more than one commit, (or if both importing and converting, in a single invocation of the **git qimport** command).

−r <*commit*>, **−−rev**=<*commit*>
: Specify a single commit ID, or a range of commit IDs, for conversion to **Git−MQ** patches; this may be specified using any form of reference which is acceptable to git, for identification of a single commit, or a range of commits.

−P, **−−push**
: Apply each patch, after it has been imported; this option is ignored for converted commits, (which are always registered as *applied*).

EXIT STATUS

On successful completion, **git qimport** 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 qimport** command exhibits fundamentally the same behaviour as its **hg qimport** counterpart.

AUTHOR

SEE ALSO

git−qnew(1), **git−qpush**(1)