

**NAME**

git-symbolic-ref – Read, modify and delete symbolic refs

**SYNOPSIS**

*git symbolic-ref* [-m <reason>] <name> <ref>

*git symbolic-ref* [-q] [--short] <name>

*git symbolic-ref* --delete [-q] <name>

**DESCRIPTION**

Given one argument, reads which branch head the given symbolic ref refers to and outputs its path, relative to the **.git/** directory. Typically you would give **HEAD** as the <name> argument to see which branch your working tree is on.

Given two arguments, creates or updates a symbolic ref <name> to point at the given branch <ref>.

Given **--delete** and an additional argument, deletes the given symbolic ref.

A symbolic ref is a regular file that stores a string that begins with **ref: refs/**. For example, your **.git/HEAD** is a regular file whose contents is **ref: refs/heads/master**.

**OPTIONS**

-d, --delete

Delete the symbolic ref <name>.

-q, --quiet

Do not issue an error message if the <name> is not a symbolic ref but a detached HEAD; instead exit with non-zero status silently.

--short

When showing the value of <name> as a symbolic ref, try to shorten the value, e.g. from **refs/heads/master** to **master**.

-m

Update the reflog for <name> with <reason>. This is valid only when creating or updating a symbolic ref.

**NOTES**

In the past, **.git/HEAD** was a symbolic link pointing at **refs/heads/master**. When we wanted to switch to another branch, we did **ln -sf refs/heads/newbranch .git/HEAD**, and when we wanted to find out which branch we are on, we did **readlink .git/HEAD**. But symbolic links are not entirely portable, so they are now deprecated and symbolic refs (as described above) are used by default.

*git symbolic-ref* will exit with status 0 if the contents of the symbolic ref were printed correctly, with status 1 if the requested name is not a symbolic ref, or 128 if another error occurs.

**GIT**

Part of the **git(1)** suite