NAME

git-show-branch − Show branches and their commits

SYNOPSIS

git show−branch [−a|−−all] [−r|−−remotes] [−−topo−order | −−date−order]
[−−current] [−−color[=<when>] | −−no−color] [−−sparse]
[−−more=<n> | −−list | −−independent | −−merge−base]
[−−no−name | −−sha1−name] [−−topics]
[(<rev> | <glob>)...]
git show−branch (−g|−−reflog)[=<n>[,<base>]] [−−list] [<ref>]

DESCRIPTION

Shows the commit ancestry graph starting from the commits named with <rev>s or <glob>s (or all refs under refs/heads and/or refs/tags) semi−visually.

It cannot show more than 29 branches and commits at a time.

It uses **showbranch.default** multi−valued configuration items if no <rev> or <glob> is given on the command line.

OPTIONS

<rev>
Arbitrary extended SHA−1 expression (see **gitrevisions**(7)) that typically names a branch head or a tag.

<glob>
A glob pattern that matches branch or tag names under refs/. For example, if you have many topic branches under refs/heads/topic, giving **topic/*** would show all of them.

−r, −−remotes
Show the remote−tracking branches.

−a, −−all
Show both remote−tracking branches and local branches.

−−current
With this option, the command includes the current branch to the list of revs to be shown when it is not given on the command line.

−−topo−order
By default, the branches and their commits are shown in reverse chronological order. This option makes them appear in topological order (i.e., descendant commits are shown before their parents).

−−date−order
This option is similar to **−−topo−order** in the sense that no parent comes before all of its children, but otherwise commits are ordered according to their commit date.

−−sparse
By default, the output omits merges that are reachable from only one tip being shown. This option makes them visible.

−−more=<n>
Usually the command stops output upon showing the commit that is the common ancestor of all the branches. This flag tells the command to go <n> more common commits beyond that. When <n> is negative, display only the <reference>s given, without showing the commit ancestry tree.

−−list
Synonym to **−−more=−1**

−−merge−base
Instead of showing the commit list, determine possible merge bases for the specified commits. All

merge bases will be contained in all specified commits. This is different from how **git-merge-base**(1) handles the case of three or more commits.

−−independent
Among the <reference>s given, display only the ones that cannot be reached from any other <reference>.

−−no−name
Do not show naming strings for each commit.

−−sha1−name
Instead of naming the commits using the path to reach them from heads (e.g. "master~2" to mean the grandparent of "master"), name them with the unique prefix of their object names.

−−topics
Shows only commits that are NOT on the first branch given. This helps track topic branches by hiding any commit that is already in the main line of development. When given "git show−branch −−topics master topic1 topic2", this will show the revisions given by "git rev−list ^master topic1 topic2"

−g, −−reflog[=<n>[,<base>]] [<ref>]
Shows <n> most recent ref−log entries for the given ref. If <base> is given, <n> entries going back from that entry. <base> can be specified as count or date. When no explicit <ref> parameter is given, it defaults to the current branch (or **HEAD** if it is detached).

−−color[=<when>]
Color the status sign (one of these: *** ! + −**) of each commit corresponding to the branch it's in. The value must be always (the default), never, or auto.

−−no−color
Turn off colored output, even when the configuration file gives the default to color output. Same as **−−color=never**.

Note that −−more, −−list, −−independent and −−merge−base options are mutually exclusive.

OUTPUT

Given N <references>, the first N lines are the one−line description from their commit message. The branch head that is pointed at by $GIT_DIR/HEAD is prefixed with an asterisk ***** character while other heads are prefixed with a **!** character.

Following these N lines, one−line log for each commit is displayed, indented N places. If a commit is on the I−th branch, the I−th indentation character shows a **+** sign; otherwise it shows a space. Merge commits are denoted by a **−** sign. Each commit shows a short name that can be used as an extended SHA−1 to name that commit.

The following example shows three branches, "master", "fixes" and "mhf":

```
$ git show−branch master fixes mhf
* [master] Add 'git show−branch'.
 ! [fixes] Introduce "reset type" flag to "git reset"
  ! [mhf] Allow "+remote:local" refspec to cause −−force when fetching.
−−−
  + [mhf] Allow "+remote:local" refspec to cause −−force when fetching.
  + [mhf~1] Use git−octopus when pulling more than one heads.
 +  [fixes] Introduce "reset type" flag to "git reset"
  + [mhf~2] "git fetch −−force".
  + [mhf~3] Use .git/remote/origin, not .git/branches/origin.
  + [mhf~4] Make "git pull" and "git fetch" default to origin
  + [mhf~5] Infamous 'octopus merge'
  + [mhf~6] Retire git−parse−remote.
```

```
 + [mhf~7] Multi−head fetch.
 + [mhf~8] Start adding the $GIT_DIR/remotes/ support.
*++ [master] Add 'git show−branch'.
```

These three branches all forked from a common commit, [master], whose commit message is "Add 'git show−branch'". The "fixes" branch adds one commit "Introduce "reset type" flag to "git reset"". The "mhf" branch adds many other commits. The current branch is "master".

EXAMPLES

If you keep your primary branches immediately under **refs/heads**, and topic branches in subdirectories of it, having the following in the configuration file may help:

```
[showbranch]
    default = −−topo−order
    default = heads/*
```

With this, **git show−branch** without extra parameters would show only the primary branches. In addition, if you happen to be on your topic branch, it is shown as well.

```
$ git show−branch −−reflog="10,1 hour ago" −−list master
```

shows 10 reflog entries going back from the tip as of 1 hour ago. Without **−−list**, the output also shows how these tips are topologically related with each other.

GIT

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