git-map(1) Manual Page

NAME

git-map - Display history of all branches in a colorized terminal format.

SYNOPSIS

> *git map* [<extra_args>…]

DESCRIPTION

Git map formats the output of `git log --graph` from all refs such that:

- Current branch is cyan.
- Local branches are green.
- Remote branches are red.

- Tags are magenta.
- Merge Base markers are white.
- The currently checked out commit is highlighted with a blue background.

The output is automatically piped through the `less` pager command, even on windows.

OPTIONS

<extra_args>...

Extra parameters to pass to the internal git-log(1) invocation. This can be used to restrict what refs *git map* operates on, etc.

If you run git map with a series of fixed arguments frequently, you can use the depot-tools.map-extra configuration variable to pre-set arguments (See `CONFIGURATION VARIABLES`)

CONFIGURATION VARIABLES

depot-tools.map-extra

Each value of the *depot-tools.map-extra*config variable is applied as an additional argument to `git log` during the execution of git map. If you wish to configure this, use git `config --add depot-tools.map-extra <value>` to do so.

EXAMPLE

Running *git map* would result in an output something like:

```
$ git map
* d0fb9c7          (HEAD ->
frozen_branch) 2014-04-10 ~
FREEZE.unindexed
* f48f415          2014-04-10 ~
modfile
* 4c5b9c0          2014-04-10 ~ a
deleted file
* f7ce1e4          (fixit) 2014-
04-10 ~ Add neat feature
<(frozen_branch)
* 3726937          2014-04-10 ~
Epic README update
| * 14db2e5        (cool_feature)
2014-04-10 ~ Respond to CL
comments
| | * ee3f972      (subfeature)
```

```
2014-04-10 ~ integrate with
CoolService
| | * 4f9f508     2014-04-10 ~
slick commenting action
| |/
| * 7d84f85       2014-04-10 ~
another improvement
<(subfeature)
| * 27abea4       (spleen_tag)
2014-04-10 ~ Refactor spleen
| * d8abe73       2014-04-10 ~
Add widget
|/
* beec6f4
(origin/master, origin/HEAD)
2014-04-10 ~ Make ReflectorImpl
use mailboxes
<(cool_feature, fixit)
* 41290e0         2014-04-10 ~
don't use glibc-specific
execinfo.h on uclibc builds
* a76fde7         2014-04-10 ~
[fsp] Add requestUnmount()
method together with the
request manager.
* 9de7a71         2014-04-10 ~
linux_aura: Use system
configuration for middle
clicking the titlebar.
* 073b0c2         2014-04-10 ~
ContentView->ContentViewCore in
```

```
ContentViewRenderView
* 2250f53         2014-04-10 ~
ozone: evdev: Filter devices by
path
* 33a7a74         2014-04-10 ~
Always output seccomp error
messages to stderr
```

As you can see, the structure of the commit history is visible, particularly what the parents of each commit are. In order to see the *upstream* relationships of the branches (i.e. which branch is tracking which other branch), use the git-map-branches(1) command.

SEE ALSO

git-map-branches(1)