git-show-ref

git-show-ref(1) ==========

NAME ---- git-show-ref - List references in a local repository

SYNOPSIS --------

> 'git show-ref' [--head] [-d | --dereference]
> [-s | --hash[=<n>]] [--abbrev[=<n>]] [--branches] [--tags]
> [--] [<pattern>...]
> 'git show-ref' --verify [-q | --quiet] [-d | --dereference]
> [-s | --hash[=<n>]] [--abbrev[=<n>]]
> [--] [<ref>...]
> 'git show-ref' --exclude-existing[=<pattern>]
> 'git show-ref' --exists <ref>

DESCRIPTION -----------

Displays references available in a local repository along with the associated commit IDs. Results can be filtered using a pattern and tags can be dereferenced into object IDs. Additionally, it can be used to test whether a particular ref exists.

By default, shows the tags, heads, and remote refs.

The `--exclude-existing` form is a filter that does the inverse. It reads refs from stdin, one ref per line, and shows those that don't exist in the local repository.

The `--exists` form can be used to check for the existence of a single references. This form does not verify whether the reference resolves to an actual object.

Use of this utility is encouraged in favor of directly accessing files under the `.git` directory.

OPTIONS -------

--head

```
Show the HEAD reference, even if it would normally be filtered out.
```

--branches

--tags

```
Limit to local branches and local tags, respectively.  These options
are not mutually exclusive; when given both, references stored in
"refs/heads" and "refs/tags" are displayed.  Note that `--heads`
is a deprecated synonym for `--branches` and may be removed
in the future.
```

-d

--dereference

```
Dereference tags into object IDs as well. They will be shown with `^{}`
appended.
```

-s

--hash[=<n>]

```
Only show the OID, not the reference name. When combined with
`--dereference`, the dereferenced tag will still be shown after the OID.
```

--verify

```
Enable stricter reference checking by requiring an exact ref path.
Aside from returning an error code of 1, it will also print an error
message if `--quiet` was not specified.
```

--exists

```
Check whether the given reference exists. Returns an exit code of 0 if
it does, 2 if it is missing, and 1 in case looking up the reference
failed with an error other than the reference being missing.
```

--abbrev[=<n>]

```
Abbreviate the object name.  When using `--hash`, you do
not have to say `--hash --abbrev`; `--hash=n` would do.
```

-q

--quiet

```
Do not print any results to stdout. Can be used with `--verify` to
silently check if a reference exists.
```

--exclude-existing[=<pattern>]

```
Make `git show-ref` act as a filter that reads refs from stdin of the
form `^(?:<anything>\s)?<refname>(?:\^{})?$`
and performs the following actions on each:
(1) strip `^{}` at the end of line if any;
(2) ignore if pattern is provided and does not head-match refname;
(3) warn if refname is not a well-formed refname and skip;
(4) ignore if refname is a ref that exists in the local repository;
(5) otherwise output the line.
```

<pattern>...

```
Show references matching one or more patterns. Patterns are matched from
the end of the full name, and only complete parts are matched, e.g.
'master' matches 'refs/heads/master', 'refs/remotes/origin/master',
'refs/tags/jedi/master' but not 'refs/heads/mymaster' or
'refs/remotes/master/jedi'.
```

OUTPUT ------

The output is in the format:

```
<oid> SP <ref> LF
```

For example,

```
$ git show-ref --head --dereference
832e76a9899f560a90ffd62ae2ce83bbeff58f54 HEAD
832e76a9899f560a90ffd62ae2ce83bbeff58f54 refs/heads/master
832e76a9899f560a90ffd62ae2ce83bbeff58f54 refs/heads/origin
3521017556c5de4159da4615a39fa4d5d2c279b5 refs/tags/v0.99.9c
6ddc0964034342519a87fe013781abf31c6db6ad refs/tags/v0.99.9c^{}
055e4ae3ae6eb344cbabf2a5256a49ea66040131 refs/tags/v1.0rc4
423325a2d24638ddcc82ce47be5e40be550f4507 refs/tags/v1.0rc4^{}
...
```

When using `--hash` (and not `--dereference`), the output is in the format:

```
<oid> LF
```

For example,

```
$ git show-ref --branches --hash
2e3ba0114a1f52b47df29743d6915d056be13278
185008ae97960c8d551adcd9e23565194651b5d1
03adf42c988195b50e1a1935ba5fcbc39b2b029b
...
```

EXAMPLES --------

To show all references called "master", whether tags or heads or anything else, and regardless of how deep in the reference naming hierarchy they are, use:

```
    git show-ref master
```

This will show "refs/heads/master" but also "refs/remote/other-repo/master", if such references exist.

When using the `--verify` flag, the command requires an exact path:

```
    git show-ref --verify refs/heads/master
```

will only match the exact branch called "master".

If nothing matches, `git` `show-ref` will return an error code of 1, and in the case of verification, it will show an error message.

For scripting, you can ask it to be quiet with the `--quiet` flag, which allows you to do things like

```
    git show-ref --quiet --verify -- "refs/heads/$headname" ||
        echo "$headname is not a valid branch"
```

to check whether a particular branch exists or not (notice how we don’t actually want to show any results, and we want to use the full refname for it in order to not trigger the problem with ambiguous partial matches).

To show only tags, or only proper branch heads, use `--tags` and/or `--branches` respectively (using both means that it shows tags and branches, but not other random references under the refs/ subdirectory).

To do automatic tag object dereferencing, use the `-d` or `--dereference` flag, so you can do

```
    git show-ref --tags --dereference
```

to get a listing of all tags together with what they dereference.

FILES ----- `.git/refs/*`, `.git/packed-refs`

SEE ALSO -------- linkgit:git-for-each-ref[1], linkgit:git-ls-remote[1], linkgit:git-update-ref[1], linkgit:gitrepository-layout[5]

GIT --- Part of the linkgit:git[1] suite

□□□ #1
□ □□□ □□□ 2026-10-31 20:13:04 UTC
□ □□□ □□□ 2026-10-31 20:13:04 UTC