NAME

git-fast-export - Git data exporter

SYNOPSIS

git fast-export [options] | git fast-import

DESCRIPTION

This program dumps the given revisions in a form suitable to be piped into *git fast-import*.

You can use it as a human-readable bundle replacement (see **git-bundle(1)**), or as a kind of an interactive *git filter-branch*.

OPTIONS

--progress=<n>

Insert *progress* statements every <n> objects, to be shown by *git fast-import* during import.

--signed-tags=(verbatim|warn|warn-strip|strip|abort)

Specify how to handle signed tags. Since any transformation after the export can change the tag names (which can also happen when excluding revisions) the signatures will not match.

When asking to *abort* (which is the default), this program will die when encountering a signed tag. With *strip*, the tags will silently be made unsigned, with *warn-strip* they will be made unsigned but a warning will be displayed, with *verbatim*, they will be silently exported and with *warn*, they will be exported, but you will see a warning.

--tag-of-filtered-object=(abort|drop|rewrite)

Specify how to handle tags whose tagged object is filtered out. Since revisions and files to export can be limited by path, tagged objects may be filtered completely.

When asking to *abort* (which is the default), this program will die when encountering such a tag. With *drop* it will omit such tags from the output. With *rewrite*, if the tagged object is a commit, it will rewrite the tag to tag an ancestor commit (via parent rewriting; see **git-rev-list(1)**)

-M, -C

Perform move and/or copy detection, as described in the **git-diff(1)** manual page, and use it to generate rename and copy commands in the output dump.

Note that earlier versions of this command did not complain and produced incorrect results if you gave these options.

--export-marks=<file>

Dumps the internal marks table to <file> when complete. Marks are written one per line as :markid SHA-1. Only marks for revisions are dumped; marks for blobs are ignored. Backends can use this file to validate imports after they have been completed, or to save the marks table across incremental runs. As <file> is only opened and truncated at completion, the same path can also be safely given to --import-marks. The file will not be written if no new object has been marked/exported.

--import-marks=<file>

Before processing any input, load the marks specified in <file>. The input file must exist, must be readable, and must use the same format as produced by --export-marks.

Any commits that have already been marked will not be exported again. If the backend uses a similar --import-marks file, this allows for incremental bidirectional exporting of the repository by keeping the marks the same across runs.

--fake-missing-tagger

Some old repositories have tags without a tagger. The fast-import protocol was pretty strict about that, and did not allow that. So fake a tagger to be able to fast-import the output.

--use-done-feature

Start the stream with a *feature done* stanza, and terminate it with a *done* command.

--no-data

Skip output of blob objects and instead refer to blobs via their original SHA-1 hash. This is useful when rewriting the directory structure or history of a repository without touching the contents of individual files. Note that the resulting stream can only be used by a repository which already contains the necessary objects.

--full-tree

This option will cause fast-export to issue a deleteall directive for each commit followed by a full list of all files in the commit (as opposed to just listing the files which are different from the commit's first parent).

--refspec

Apply the specified refspec to each ref exported. Multiple of them can be specified.

[<git-rev-list-args>...]

A list of arguments, acceptable to *git rev-parse* and *git rev-list*, that specifies the specific objects and references to export. For example, master~10..master causes the current master reference to be exported along with all objects added since its 10th ancestor commit.

EXAMPLES

$ git fast-export --all | (cd /empty/repository && git fast-import)

This will export the whole repository and import it into the existing empty repository. Except for reencoding commits that are not in UTF-8, it would be a one-to-one mirror.

$ git fast-export master~5..master |
sed s|refs/heads/master|refs/heads/other| |
git fast-import

This makes a new branch called *other* from *master~5..master* (i.e. if *master* has linear history, it will take the last 5 commits).

Note that this assumes that none of the blobs and commit messages referenced by that revision range contains the string *refs/heads/master*.

LIMITATIONS

Since *git fast-import* cannot tag trees, you will not be able to export the linux.git repository completely, as it contains a tag referencing a tree instead of a commit.

GIT

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