NAME

git-fsck − Verifies the connectivity and validity of the objects in the database

SYNOPSIS

git fsck [−−tags] [−−root] [−−unreachable] [−−cache] [−−no−reflogs]
[−−[no−]full] [−−strict] [−−verbose] [−−lost−found]
[−−[no−]dangling] [−−[no−]progress] [−−connectivity−only]
[−−[no−]name−objects] [<object>*]

DESCRIPTION

Verifies the connectivity and validity of the objects in the database.

OPTIONS

<object>
: An object to treat as the head of an unreachability trace.

  If no objects are given, *git fsck* defaults to using the index file, all SHA−1 references in **refs** namespace, and all reflogs (unless −−no−reflogs is given) as heads.

−−unreachable
: Print out objects that exist but that aren't reachable from any of the reference nodes.

−−[no−]dangling
: Print objects that exist but that are never *directly* used (default). **−−no−dangling** can be used to omit this information from the output.

−−root
: Report root nodes.

−−tags
: Report tags.

−−cache
: Consider any object recorded in the index also as a head node for an unreachability trace.

−−no−reflogs
: Do not consider commits that are referenced only by an entry in a reflog to be reachable. This option is meant only to search for commits that used to be in a ref, but now aren't, but are still in that corresponding reflog.

−−full
: Check not just objects in GIT_OBJECT_DIRECTORY ($GIT_DIR/objects), but also the ones found in alternate object pools listed in GIT_ALTERNATE_OBJECT_DIRECTORIES or $GIT_DIR/objects/info/alternates, and in packed Git archives found in $GIT_DIR/objects/pack and corresponding pack subdirectories in alternate object pools. This is now default; you can turn it off with −−no−full.

−−connectivity−only
: Check only the connectivity of tags, commits and tree objects. By avoiding to unpack blobs, this speeds up the operation, at the expense of missing corrupt objects or other problematic issues.

−−strict
: Enable more strict checking, namely to catch a file mode recorded with g+w bit set, which was created by older versions of Git. Existing repositories, including the Linux kernel, Git itself, and sparse repository have old objects that triggers this check, but it is recommended to check new projects with this flag.

−−verbose
: Be chatty.

−−lost−found
: Write dangling objects into .git/lost−found/commit/ or .git/lost−found/other/, depending on type. If the object is a blob, the contents are written into the file, rather than its object name.

−−name−objects
: When displaying names of reachable objects, in addition to the SHA−1 also display a name that describes **how** they are reachable, compatible with **git-rev-parse(1)**, e.g. **HEAD@{1234567890}~25^2:src/**.

−−[no−]progress
: Progress status is reported on the standard error stream by default when it is attached to a terminal, unless −−no−progress or −−verbose is specified. −−progress forces progress status even if the standard error stream is not directed to a terminal.

DISCUSSION

git−fsck tests SHA−1 and general object sanity, and it does full tracking of the resulting reachability and everything else. It prints out any corruption it finds (missing or bad objects), and if you use the **−−unreachable** flag it will also print out objects that exist but that aren't reachable from any of the specified head nodes (or the default set, as mentioned above).

Any corrupt objects you will have to find in backups or other archives (i.e., you can just remove them and do an *rsync* with some other site in the hopes that somebody else has the object you have corrupted).

If core.commitGraph is true, the commit−graph file will also be inspected using *git commit−graph verify*. See **git-commit-graph(1)**.

EXTRACTED DIAGNOSTICS

expect dangling commits − potential heads − due to lack of head information
: You haven't specified any nodes as heads so it won't be possible to differentiate between un−parented commits and root nodes.

missing sha1 directory *<dir>*
: The directory holding the sha1 objects is missing.

unreachable <type> <object>
: The <type> object <object>, isn't actually referred to directly or indirectly in any of the trees or commits seen. This can mean that there's another root node that you're not specifying or that the tree is corrupt. If you haven't missed a root node then you might as well delete unreachable nodes since they can't be used.

missing <type> <object>
: The <type> object <object>, is referred to but isn't present in the database.

dangling <type> <object>
: The <type> object <object>, is present in the database but never *directly* used. A dangling commit could be a root node.

sha1 mismatch <object>
: The database has an object who's sha1 doesn't match the database value. This indicates a serious data integrity problem.

ENVIRONMENT VARIABLES

GIT_OBJECT_DIRECTORY
: used to specify the object database root (usually $GIT_DIR/objects)

GIT_INDEX_FILE
: used to specify the index file of the index

GIT_ALTERNATE_OBJECT_DIRECTORIES
: used to specify additional object database roots (usually unset)

GIT

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