NAME

git-cat-file - Provide content or type and size information for repository objects

SYNOPSIS

git cat-file (-t | -s | -e | -p | <type> | --textconv) <object>
git cat-file (--batch | --batch-check) < <list-of-objects>

DESCRIPTION

In its first form, the command provides the content or the type of an object in the repository. The type is required unless *-t* or *-p* is used to find the object type, or *-s* is used to find the object size, or *--textconv* is used (which implies type blob).

In the second form, a list of objects (separated by linefeeds) is provided on stdin, and the SHA-1, type, and size of each object is printed on stdout.

OPTIONS

<object>
: The name of the object to show. For a more complete list of ways to spell object names, see the SPECIFYING REVISIONS section in **gitrevisions(7)**.

-t
: Instead of the content, show the object type identified by <object>.

-s
: Instead of the content, show the object size identified by <object>.

-e
: Suppress all output; instead exit with zero status if <object> exists and is a valid object.

-p
: Pretty-print the contents of <object> based on its type.

<type>
: Typically this matches the real type of <object> but asking for a type that can trivially be dereferenced from the given <object> is also permitted. An example is to ask for a tree with <object> being a commit object that contains it, or to ask for a blob with <object> being a tag object that points at it.

--textconv
: Show the content as transformed by a textconv filter. In this case, <object> has be of the form <tree-ish>:<path>, or :<path> in order to apply the filter to the content recorded in the index at <path>.

--batch, --batch=<format>
: Print object information and contents for each object provided on stdin. May not be combined with any other options or arguments. See the section BATCH OUTPUT below for details.

--batch-check, --batch-check=<format>
: Print object information for each object provided on stdin. May not be combined with any other options or arguments. See the section BATCH OUTPUT below for details.

OUTPUT

If *-t* is specified, one of the <type>.

If *-s* is specified, the size of the <object> in bytes.

If *-e* is specified, no output.

If *-p* is specified, the contents of <object> are pretty-printed.

If <type> is specified, the raw (though uncompressed) contents of the <object> will be returned.

BATCH OUTPUT

If --batch or --batch-check is given, cat-file will read objects from stdin, one per line, and print information about them. By default, the whole line is considered as an object, as if it were fed to **git-rev-parse(1)**.

You can specify the information shown for each object by using a custom <format>. The <format> is copied literally to stdout for each object, with placeholders of the form %(atom) expanded, followed by a newline. The available atoms are:

objectname
: The 40-hex object name of the object.

objecttype
: The type of of the object (the same as cat-file -t reports).

objectsize
: The size, in bytes, of the object (the same as cat-file -s reports).

objectsize:disk
: The size, in bytes, that the object takes up on disk. See the note about on-disk sizes in the CAVEATS section below.

deltabase
: If the object is stored as a delta on-disk, this expands to the 40-hex sha1 of the delta base object. Otherwise, expands to the null sha1 (40 zeroes). See CAVEATS below.

rest
: If this atom is used in the output string, input lines are split at the first whitespace boundary. All characters before that whitespace are considered to be the object name; characters after that first run of whitespace (i.e., the rest of the line) are output in place of the %(rest) atom.

If no format is specified, the default format is %(objectname) %(objecttype) %(objectsize).

If --batch is specified, the object information is followed by the object contents (consisting of %(objectsize) bytes), followed by a newline.

For example, --batch without a custom format would produce:

<sha1> SP <type> SP <size> LF
<contents> LF

Whereas --batch-check=%(objectname) %(objecttype) would produce:

<sha1> SP <type> LF

If a name is specified on stdin that cannot be resolved to an object in the repository, then cat-file will ignore any custom format and print:

<object> SP missing LF

CAVEATS

Note that the sizes of objects on disk are reported accurately, but care should be taken in drawing conclusions about which refs or objects are responsible for disk usage. The size of a packed non-delta object may be much larger than the size of objects which delta against it, but the choice of which object is the base and which is the delta is arbitrary and is subject to change during a repack.

Note also that multiple copies of an object may be present in the object database; in this case, it is undefined which copy's size or delta base will be reported.

GIT

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