

NAME

digup – **D**igest file **u**psiding program.

SYNOPSIS

digup [*OPTIONS*]

DESCRIPTION

digup is a tool to update md5sum or shasum digest files. It will read existing digest files, check the current directory for new, updated, modified, renamed or deleted files and query the user with a summary of changes. After reviewing the updates, they can be written back to the digest file.

One of the envisioned applications of digup is to update and verify incremental archives like chronological data storages or music collections. By using digup possibly undesired changes or lost files can easily be detected while quickly adding new files.

Another purpose could be to automatically verify the integrity of hard disk copies of archives, as backups to hard disks are becoming increasingly popular. Using a full file digest scan even slowly creeping bad blocks on old hard disks can be detected. By using a crontab entry, this check can be performed unattended and routinely.

In normal operation only touched files with newer modification times are fully read and their digest compared. Optionally a full scan can be performed to test all file contents against their expected digests.

Symbolic links are supported by either following the link and reading the target's digest or by saving only the link target path and verifying it against the old one.

The digest files written by digup are compatible with those generated and read by md5sum and similar programs from the coreutils package. Additional information like file size and modification time or symlink targets are stored on comment lines.

Four digest algorithms are supported: MD5, SHA1, SHA256 and SHA512. The digest file itself is also checksummed using CRC32 against unintentional changes. A fast red-black binary tree is used for the internal file list, allowing fast operation on a large number of files.

OPTIONS**-b, --batch**

Enable non-interactive batch processing mode as needed when run unattended e.g. from cron. This option also decreases verbosity by one level (--quiet). The returned error code is set to 1 if any changed, renamed, moved, deleted files or read errors occur.

-c, --check

Perform a full digest scan of all file contents, thus ignoring file modification times. Without this option files with equal size and modification time are skipped.

-d, --directory=<path>

Change into this directory before looking for digest files or performing a recursive scan.

--exclude-marker=<file>

Sets a marker file, often called ".nobackup" in other programs. If this marker file is found in a directory, the directory itself and all sub-directories are excluded from the digest scan.

This option is persistent. It is saved in the digest file and will be applied to all future scans performed to check or update digests.

-f, --file=<file>

Check this file for existing digests and write updates to it. Depending on the selected digest --type the following file names are used by default: "md5sum.txt", "sha1sum.txt", "sha256sum.txt" or "sha512sum.txt".

-l, --links

When this flag is enabled, symbolic links (if supported on the platform) are followed. Otherwise, by default, only the symbolic link's target path is saved and verified.

-m, --modified

Print only modified, changed, copied, renamed or deleted files. Unchanged files lines are suppressed. If the whole digest file is clean, then no summary output is printed at all. This option is useful for crontabs in combination with `--batch`.

--modify-window=<integer>

Consider modification time deltas of up to this value to be unchanged (the default is zero). This option is very useful for checking backups on FAT filesystems, as FAT stores modification times with a precision of only 2 seconds.

-q, --quiet

Reduces the level of verbosity by one.

-r, --restrict=<substring>

Restricts the digest check to filepaths containing the given substring pattern, other files are skipped. Does NOT imply `-c / --check`; specify it additionally to run a full digest check of specific files.

-t, --type=<digest-type>

Select the digest type for newly created digest files. This is not needed for updating existing one, as the type can be inferred from the digest length.

digest-type must be md5, sha1, sha256 or sha512. The default digest type is sha1.

-u, --update

Automatically update the digest file in batch mode. Requires `--batch`, which is not automatically activated by this option.

-v, --verbose

Increase the level of verbosity by one.

At level 0 only read errors are printed.

At level 1 the status of each processed file is printed.

And at level 2 (the default) additionally a progress indicator is printed while reading each file (one dot per megabyte).

-w, --windows

Ignores modification time deltas of just 1 second (equivalent to `--modify-window=1`). Useful for checking backups on FAT filesystems.

EXAMPLES

To update or create a SHA1 digest file in current directory just run plain
`digup`

Alternatively update or create a SHA512 digest file in `/mnt/disk1`
`digup -t sha512 -d /mnt/disk1/archive`

A common call for a crontab line is
`digup -bmd /mnt/disk1/archive`

To run a full scan every midnight add a line similar to the following to your crontab
`0 * * * * digup -bcmd /mnt/disk1/archive`

AUTHORS

Timo Bingmann (tb (at) idlebox <dot> net)

SEE ALSO

md5sum(1), sha1sum(1), sha256sum(1), sha512sum(1)