NAME

rhash − calculate/check CRC32, MD5, SHA1, GOST, TTH, BTIH or other message digests.

SYNOPSIS

rhash [*option* **]... [** *file* **]...**

DESCRIPTION

RHash (Recursive Hasher) computes and verifies various message digests and checksums of files. Supported hash algorithms include CRC32, CRC32C, MD4, MD5, SHA1, SHA256, SHA512, SHA3, Tiger, DC++ TTH, BTIH, AICH, ED2K, GOST R 34.11−*, RIPEMD−160, HAS−160, BLAKE2s/BLAKE2b, BLAKE3, EDON−R 256/512, Whirlpool, Snefru−128/256.

The program can create and verify Magnet links and eDonkey ed2k:// links, see −−magnet and −−ed2k−link options.

A dash string parameter '−' is interpreted as the standard input stream (stdin).

By default **rhash** prints sums in SFV format with CRC32 checksum only. The format can be changed by options −−bsd, −−magnet, −−simple, −−printf, −−template. To output all sums use the '−a' option.

PROGRAM MODE OPTIONS

The default mode is to print checksums for all files and directory trees specified by command line. The mode can be changed by the following options.

−c, −−check
: Check hash files specified by command line. RHash can verify hash files in SFV and BSD formats, MD5 and SHA1 files format and text files containing magnet or ed2k links (one link per line). Empty lines and lines starting with ';' or '#' are ignored.

  RHash can verify hash files generated without −−printf and −−template formatting options.

  If the hash algorithm is not specified by command line options then RHash tries to detect algorithm from the extension of the hash file. If detection fails, then all hash function of the same hash length are calculated, and that significally slows down files verification. To speed up verification, in such case, explicitly specify the hash algorithm in the command line.

−u, −−update=<hash-file>
: Update the specified hash file by computing message digests for files listed in command arguments that aren't present in it. New digests are appended in the format specified by formatting options.

  With −−recursive, updates the hash file using files from entire directory trees. With −−check, verifies existing digests before updating.

−−missing=<hash-file>
: List files from the hash file that are missing or inaccessible in the filesystem.

−−unverified=<hash-file>
: List command-line files missing from the given hash file (unverified files). With −−recursive, checks entire directory trees for unverified files.

−k, −−check−embedded
: Verify files by crc32 sum embedded in their names.

−−torrent
: Create a torrent file for each processed file.

−h, −−help
: Help: print help screen and exit.

−V, −−version
: Version: print version and exit.

−B, −−benchmark
: Run benchmark for the selected hash algorithm(s).

HASH ALGORITHMS OPTIONS

−C, −−crc32
: CRC32: Select CRC32 checksum algorithm.

−−crc32c
: CRC32C: Select CRC32C checksum algorithm.

−−md4 MD4: Select MD4 hash function.

−M, −−md5
: MD5: Select MD5 hash function.

−H, −−sha1
: SHA1: Select SHA1 hash function.

−−sha224, −−sha256, −−sha384, −−sha512
: Select specified SHA2 hash function.

−−sha3-224, −−sha3-256, −−sha3-384, −−sha3-512
: Select specified SHA3 hash function.

−−tiger Tiger: Select Tiger hash function.

−T, −−tth
: TTH: Select DC++ TTH hash function.

−−btih BTIH: Select BitTorrent Info Hash.

−A, −−aich
: AICH: Select AICH hash function.

−E, −−ed2k
: ED2K: Select eDonkey 2000 hash function.

−W, −−whirlpool
: Whirlpool: Select Whirlpool hash function.

−G, −−gost12-256
: GOST−2012: Select 256-bit GOST R 34.11−2012, the Russian GOST standard hash function.

−−gost12-512
: GOST−2012: Select 512-bit GOST R 34.11−2012, the Russian GOST standard hash function.

−−gost94
: GOST−94: Select GOST R 34.11−94, the deprecated Russian hash function.

−−gost94−cryptopro
: GOST−94−CRYPTOPRO: Select the CryptoPro version of the deprecated Russian GOST R 34.11−94 hash function.

−−ripemd160
: RIPEMD−160: Select RIPEMD−160 hash function.

−−has160
: HAS−160: Select HAS−160 hash function.

−−snefru128, −−snefru256
: SNEFRU: Select SNEFRU−128/256 hash function.

−−edonr256, −−edonr512
: EDON−R: Select EDON−R 256/512 hash function.

−−blake2b, −−blake2s
: BLAKE2: Select BLAKE2b/BLAKE2s hash function.

−−blake3
: BLAKE3: Select BLAKE3 hash function.

−a, −−all
: Calculate all supported hash functions.

−−list−hashes
: List names of all supported hash functions, one per line.

MISCELLANEOUS OPTIONS

−r, −−recursive
: Recursively process directories, specified by command line.

−−follow
: Follow symbolic links when processing files or directories recursively.

−m, −−message=<text>
: Calculate message digests of the given text message.

−−file−list=<file>
: Process given file as a file-list. Lines of this file are interpreted as paths to files to be processed. Multiple file lists can be specified at command line.

−v, −−verbose
: Be verbose.

−−brief
: Print brief form of verification report (without a header and footer), when verifying a hash file.

−P, −−percents
: Show percents, while calculating or checking sums

−−skip−ok
: Don’t print OK messages for successfully verified files.

−−ignore−missing
: Ignore missing files, while verifying a hash file.

−i, −−ignore−case
: Ignore case of filenames when updating crc files.

−−speed
: Print per−file and the total processing speed.

−e, −−embed−crc
: Rename files by inserting crc32 sum into name.

−−embed−crc−delimiter=<delimiter>
: Insert specified <delimiter> before a crc sum in the −−embed−crc mode, default is white space. The <delimiter> can be a character or empty string.

−−path−separator=<separator>
: Use specified path separator to display paths. Only slash (‘/’) and backslash (‘\’) values are accepted.

−q, −−accept=<list>
: Set a comma-delimited list of extensions of the files to process.

−−exclude=<list>
: Set a comma-delimited list of extensions of the files to exclude from processing.

−t, −−crc−accept=<list>
: Set a comma-delimited list of extensions of the hash files to verify.

−−max−depth=<levels>
: Descend at most <levels> (a non-negative integer) levels of directories below the command line arguments. '−−max−depth 0' means only apply the tests and actions to the command line arguments.

−o, −−output=<file−path>
: Set the file to output calculated message digests or verification results to.

−l, −−log=<file−path>
: Set the file to log errors and verbose information to.

−−openssl=<list>
: Specify which hash functions should be calculated using the OpenSSL library. The <list> is a comma delimited list of hash function names, but only those supported by openssl are allowed: md4, md5, sha1, sha2*, ripemd160 and whirlpool.

−−gost−reverse
: Reverse bytes in hexadecimal output of a GOST hash functions. The most significant byte of the message digest will be printed first. Default order is the least significant byte first.

−−bt−batch=<file−path>
: Turn on torrent batch mode (implies torrent mode). Calculates batch-torrent for the files specified at command line and saves the torrent file to the file−path. The option −r <directory> can be useful in this mode.

−−bt−private
: Generate torrent file or BTIH for a private BitTorrent tracker.

−−bt−transmission
: Generate torrent file or BTIH compatible with Transmission torrent client.

−−bt−piece−length
: Set the *piece length* value for torrent file.

−−bt−announce=<announce−url>
: Add a tracker announce URL to the created torrent file(s). Several URLs can be passed by specifying the option mutltiple times. This option doesn't change the BTIH message digest.

−−benchmark−raw
: Switch benchmark output format to be a machine-readable tab-delimited text with hash function name, speed, cpu clocks per byte. This option works only if the −−benchmark option was specified.

−−no−detect−by−ext
: Do not detect hash function by an extension of hash file, in the −−check mode.

−−no−path−escaping
: Turn off escape characters in file paths. The option can be specified in the default, check or update modes.

−− (double dash)
: Mark the end of command line options. All parameters following the double dash are interpreted as files or directories. It is typically used to process filenames starting with a dash '−'. Alternatively you can specify './' or full path before such files, so they will not look like options anymore.

OUTPUT FORMAT OPTIONS

−−sfv Print message digests in the SFV (Simple File Verification) output format (default). But unlike common SFV file, not only CRC32, but any message digests specified by options can be printed.

−g, −−magnet
Print message digests formatted as magnet links.

−−bsd Use BSD output format. Each message digest is printed on a separate line after hash function name and file's path, enclosed in parentheses.

−−simple
Use simple output format. Each line will consist of filename and message digests specified by options.

−−one−hash
Print one message digest per line without additional file information.

−L, −−ed2k−link
Print eDonkey link containing file name, file size, ED2K and AICH message digests.

−−hex Print message digests in hexadecimal format.

−−base32
Print message digests in Base32 format.

−b, −−base64
Print message digests in Base64 format.

−−uppercase
Print message digests in upper case.

−−lowercase
Print message digests in lower case.

−−template=<file>
Read printf-like template from given <file>. See the −−printf option.

−p, −−printf=<format>
Format: print *format* string the standard output, interpreting '\' escapes and '%' directi ves. The escapes and directives are:

\n Newline.

\r Carriage return.

\t Horizontal tab.

\\ A literal backslash ('\').

\0 ASCII NUL.

\NNN The character which octal ASCII code is NNN.

\xNN The character which hexadecimal ASCII code is NN.

A '\' character followed by any other character is treated as an ordinary character, so they both are printed.

%% A literal percent sign.

%p File's path.

%f File's name.

%d File's directory.

%u or %U
Prefix used to print a filename, file path or base64/raw message digest as an URL-encoded string. For example: '%up', '%ud', '%uf', '%uBm', '%u@h'. Use %u for

lowercase and %U for uppercase hexadecimal characters.

%s File's size in bytes.

%{mtime}
File's last modification time.

%a or %A
AICH message digest.

%c or %C
CRC32 checksum. Use %c for lowercase and %C for uppercase characters.

%g or %G
GOST R 34.11−2012 256-bit message digest.

%h or %H
SHA1 message digest.

%e or %E
ED2K message digest.

%l or %L
EDonkey ed2k://... link.

%m or %M
MD5 message digest.

%r or %R
RIPEMD-160 message digest.

%t or %T
TTH message digest.

%w or %W
Whirlpool message digest.

%{crc32}, %{crc32c}, %{md4}, %{md5}, %{sha1}, %{tiger}, %{tth}, %{btih}, %{ed2k}, %{aich}, %{whirlpool}, %{ripemd160}, %{has160}, %{gost94}, %{gost94−cryptopro}, %{gost12−256}, %{gost12−512}, %{sha−224}, %{sha−256}, %{sha−384}, %{sha−512}, %{sha3−224}, %{sha3−256}, %{sha3−384}, %{sha3−512}, %{edon−r256}, %{edon−r512}, %{blake2s}, %{blake2b}, %{blake3}, %{snefru128}, %{snefru256}
Print the specified message digest. It is printed in uppercase, if the hash function name starts with a capital letter, e.g. %{TTH}, %{Sha-512}.

%x<hash>, %b<hash>, %B<hash>, %@<hash>
Use one of these prefixes to output a message digest in hexadecimal, base32, base64 or raw (binary) format respectively, e.g. %b{md4}, %BH or %xT.

The default output format can also be changed by renaming the program or placing a hardlink/symlink to it with a filename containing strings 'crc32', 'crc32c', 'md4', 'md5', 'sha1', 'sha224' 'sha256', 'sha384' 'sha512', 'sha3−256', 'sha3−512', 'sha3−224', 'sha3−384', 'tiger', 'tth', 'btih', 'aich', 'ed2k', 'ed2k−link', 'gost12−256', 'gost12−512', 'gost94', 'gost94−cryptopro', 'rmd160', 'has160', 'whirlpool', 'edonr256', 'edonr512', 'blake2s', 'blake2b', 'blake3', 'snefru128', 'snefru256', 'sfv' , 'bsd' or 'magnet'.

CONFIG FILE

RHash looks for a config file at $XDG_CONFIG_HOME/rhash/rhashrc, $HOME/.config/rhash/rhashrc, $XDG_CONFIG_DIRS/rhash/rhashrc, $HOME/.rhashrc and /etc.

The config file consists of lines formatted as
variable = value

where the *variable* can be a name of any command line option, like *magnet, printf, percents, etc.* A boolean variable can be set to true by a value ‘on’, ‘yes’ or ‘true’, any other value sets the variable to false.

Empty lines and lines starting with ‘#’ or ‘;’ are ignored.

Example config file:
# This is a comment line
percents = on
crc-accept = .sfv,.md5,.sha1,.sha256,.sha512,.tth,.magnet

AUTHOR

Aleksey Kravchenko <rhash.admin@gmail.com>

SEE ALSO

md5sum(1) **cksfv**(1) **ed2k_hash**(1)

BUGS

Bug reports are welcome! Post them to the GitHub issues page *https://github.com/rhash/RHash/issues*