

NAME

`rdiff-backup` – local/remote mirror and incremental backup

SYNOPSIS

rdiff-backup [*options*] [[[*user@*]*host1.foo*]::*source_directory*] [[[*user@*]*host2.foo*]::*destination_directory*]

rdiff-backup {{ `-l` | `--list-increments` } | `--remove-older-than` *time_interval* | `--list-at-time` *time* | `--list-changed-since` *time* | `--list-increment-sizes` | `--verify` | `--verify-at-time` *time*} [[[*user@*]*host2.foo*]::*destination_directory*]

rdiff-backup `--calculate-average` *statfile1* *statfile2* ...

rdiff-backup `--test-server` [*user1*]*@host1.net1*::*path* [[*user2*]*@host2.net2*::*path*] ...

DESCRIPTION

rdiff-backup is a script, written in **python**(1) that backs up one directory to another. The target directory ends up a copy (mirror) of the source directory, but extra reverse diffs are stored in a special subdirectory of that target directory, so you can still recover files lost some time ago. The idea is to combine the best features of a mirror and an incremental backup. **rdiff-backup** also preserves symlinks, special files, hardlinks, permissions, uid/gid ownership, and modification times.

rdiff-backup can also operate in a bandwidth efficient manner over a pipe, like **rsync**(1). Thus you can use ssh and **rdiff-backup** to securely back a hard drive up to a remote location, and only the differences will be transmitted. Using the default settings, **rdiff-backup** requires that the remote system accept ssh connections, and that **rdiff-backup** is installed in the user's PATH on the remote system. For information on other options, see the section on **REMOTE OPERATION**.

Note that you **should not write to the mirror directory** except with **rdiff-backup**. Many of the increments are stored as reverse diffs, so if you delete or modify a file, you may lose the ability to restore previous versions of that file.

Finally, this man page is intended more as a precise description of the behavior and syntax of **rdiff-backup**. New users may want to check out the `examples.html` file included in the **rdiff-backup** distribution.

OPTIONS**-b, --backup-mode**

Force backup mode even if first argument appears to be an increment or mirror file.

--calculate-average

Enter calculate average mode. The arguments should be a number of statistics files. **rdiff-backup** will print the average of the listed statistics files and exit.

--carbonfile

Enable backup of MacOS X carbonfile information.

--check-destination-dir

If an **rdiff-backup** session fails, running **rdiff-backup** with this option on the destination dir will undo the failed directory. This happens automatically if you attempt to back up to a directory and the last backup failed.

--compare

This is equivalent to '`--compare-at-time now`'

--compare-at-time *time*

Compare a directory with the backup set at the given time. This can be useful to see how archived data differs from current data, or to check that a backup is current. This only compares metadata,

in the same way rdiff-backup decides whether a file has changed.

--compare-full

This is equivalent to '**--compare-full-at-time now**'

--compare-full-at-time *time*

Compare a directory with the backup set at the given time. To compare regular files, the repository data will be copied in its entirety to the source side and compared byte by byte. This is the slowest but most complete compare option.

--compare-hash

This is equivalent to '**--compare-hash-at-time now**'

--compare-hash-at-time *time*

Compare a directory with the backup set at the given time. Regular files will be compared by computing their SHA1 digest on the source side and comparing it to the digest recorded in the metadata.

--create-full-path

Normally only the final directory of the destination path will be created if it does not exist. With this option, all missing directories on the destination path will be created. Use this option with care: if there is a typo in the remote path, the remote filesystem could fill up very quickly (by creating a duplicate backup tree). For this reason this option is primarily aimed at scripts which automate backups.

--current-time *seconds*

This option is useful mainly for testing. If set, rdiff-backup will use it for the current time instead of consulting the clock. The argument is the number of seconds since the epoch.

--exclude *shell_pattern*

Exclude the file or files matched by *shell_pattern*. If a directory is matched, then files under that directory will also be matched. See the **FILE SELECTION** section for more information.

--exclude-device-files

Exclude all device files. This can be useful for security/permissions reasons or if rdiff-backup is not handling device files correctly.

--exclude-fifos

Exclude all fifo files.

--exclude-filelist *filename*

Excludes the files listed in *filename*. If *filename* is handwritten you probably want **--exclude-globbing-filelist** instead. See the **FILE SELECTION** section for more information.

--exclude-filelist-stdin

Like **--exclude-filelist**, but the list of files will be read from standard input. See the **FILE SELECTION** section for more information.

--exclude-globbing-filelist *filename*

Like **--exclude-filelist** but each line of the filelist will be interpreted according to the same rules as **--include** and **--exclude**.

--exclude-globbing-filelist-stdin

Like **--exclude-globbing-filelist**, but the list of files will be read from standard input.

--exclude-other-filesystems

Exclude files on file systems (identified by device number) other than the file system the root of the source directory is on.

--exclude-regexp *regex*

Exclude files matching the given regexp. Unlike the **--exclude** option, this option does not match files in a directory it matches. See the **FILE SELECTION** section for more information.

- exclude-special-files**
Exclude all device files, fifo files, socket files, and symbolic links.
- exclude-sockets**
Exclude all socket files.
- exclude-symbolic-links**
Exclude all symbolic links. This option is automatically enabled if the backup source is running on native Windows to avoid backing-up NTFS reparse points.
- exclude-if-present *filename***
Exclude directories if *filename* is present. This option needs to come before any other include or exclude options.
- force**
Authorize a more drastic modification of a directory than usual (for instance, when overwriting of a destination path, or when removing multiple sessions with **--remove-older-than**). `rdiff-backup` will generally tell you if it needs this. **WARNING:** You can cause data loss if you mis-use this option. Furthermore, do NOT use this option when doing a restore, as it will DELETE FILES, unless you absolutely know what you are doing.
- group-mapping-file *filename***
Map group names and ids according to the group mapping file *filename*. See the **USERS AND GROUPS** section for more information.
- include *shell_pattern***
Similar to **--exclude** but include matched files instead. Unlike **--exclude**, this option will also match parent directories of matched files (although not necessarily their contents). See the **FILE SELECTION** section for more information.
- include-filelist *filename***
Like **--exclude-filelist**, but include the listed files instead. If *filename* is handwritten you probably want **--include-globbing-filelist** instead. See the **FILE SELECTION** section for more information.
- include-filelist-stdin**
Like **--include-filelist**, but read the list of included files from standard input.
- include-globbing-filelist *filename***
Like **--include-filelist** but each line of the filelist will be interpreted according to the same rules as **--include** and **--exclude**.
- include-globbing-filelist-stdin**
Like **--include-globbing-filelist**, but the list of files will be read from standard input.
- include-regexp *regexp***
Include files matching the regular expression *regexp*. Only files explicitly matched by *regexp* will be included by this option. See the **FILE SELECTION** section for more information.
- include-special-files**
Include all device files, fifo files, socket files, and symbolic links.
- include-symbolic-links**
Include all symbolic links.
- list-at-time *time***
List the files in the archive that were present at the given time. If a directory in the archive is specified, list only the files under that directory.
- list-changed-since *time***
List the files that have changed in the destination directory since the given time. See **TIME FORMATS** for the format of *time*. If a directory in the archive is specified, list only the files under that directory. This option does not read the source directory; it is used to compare the contents of two

different rdiff-backup sessions.

-l, --list-increments

List the number and date of partial incremental backups contained in the specified destination directory. No backup or restore will take place if this option is given.

--list-increment-sizes

List the total size of all the increment and mirror files by time. This may be helpful in deciding how many increments to keep, and when to `--remove-older-than`. Specifying a subdirectory is allowable; then only the sizes of the mirror and increments pertaining to that subdirectory will be listed.

--max-file-size *size*

Exclude files that are larger than the given size in bytes

--min-file-size *size*

Exclude files that are smaller than the given size in bytes

--never-drop-acls

Exit with error instead of dropping acls or acl entries. Normally this may happen (with a warning) because the destination does not support them or because the relevant user/group names do not exist on the destination side.

--no-acls

No Access Control Lists - disable backup of ACLs

--no-carbonfile

Disable backup of MacOS X carbonfile information

--no-compare-inode

This option prevents rdiff-backup from flagging a hardlinked file as changed when its device number and/or inode changes. This option is useful in situations where the source filesystem lacks persistent device and/or inode numbering. For example, network filesystems may have mount-to-mount differences in their device number (but possibly stable inode numbers); USB/1394 devices may come up at different device numbers each remount (but would generally have same inode number); and there are filesystems which don't even have the same inode numbers from use to use. Without the option rdiff-backup may generate unnecessary numbers of tiny diff files.

--no-compression

Disable the default gzip compression of most of the .snapshot and .diff increment files stored in the rdiff-backup-data directory. A backup volume can contain compressed and uncompressed increments, so using this option inconsistently is fine.

--no-compression-regexp *regexp*

Do not compress increments based on files whose filenames match regexp. The default includes many common audiovisual and archive files, and may be found in Globals.py.

--no-eas

No Extended Attributes support - disable backup of EAs.

--no-file-statistics

This will disable writing to the file_statistics file in the rdiff-backup-data directory. rdiff-backup will run slightly quicker and take up a bit less space.

--no-hard-links

Don't replicate hard links on destination side. If many hard-linked files are present, this option can drastically decrease memory usage. This option is enabled by default if the backup source or restore destination is running on native Windows.

--null-separator

Use nulls (\0) instead of newlines (\n) as line separators, which may help when dealing with file-names containing newlines. This affects the expected format of the files specified by the `--{include|exclude}-filelist[-stdin]` switches as well as the format of the directory statistics file.

--parsable-output

If set, rdiff-backup's output will be tailored for easy parsing by computers, instead of convenience for humans. Currently this only applies when listing increments using the **-l** or **--list-increments** switches, where the time will be given in seconds since the epoch.

--override-chars-to-quote

If the filesystem to which we are backing up is not case-sensitive, automatic 'quoting' of characters occurs. For example, a file 'Developer.doc' will be converted into ';068developer.doc'. To override this behavior, you need to specify this option.

--preserve-numerical-ids

If set, rdiff-backup will preserve uids/gids instead of trying to preserve unames and gnames. See the **USERS AND GROUPS** section for more information.

--print-statistics

If set, summary statistics will be printed after a successful backup. If not set, this information will still be available from the session statistics file. See the **STATISTICS** section for more information.

-r, --restore-as-of *restore_time*

Restore the specified directory as it was as of *restore_time*. See the **TIME FORMATS** section for more information on the format of *restore_time*, and see the **RESTORING** section for more information on restoring.

--remote-cmd *cmd*

Deprecated. Please use **--remote-schema** instead

--remote-schema *schema*

Specify an alternate method of connecting to a remote computer. This is necessary to get rdiff-backup not to use ssh for remote backups, or if, for instance, rdiff-backup is not in the PATH on the remote side. See the **REMOTE OPERATION** section for more information.

--remote-tempdir *path*

Adds the **--tempdir** option with argument *path* when invoking remote instances of rdiff-backup.

--remove-older-than *time_spec*

Remove the incremental backup information in the destination directory that has been around longer than the given time. *time_spec* can be either an absolute time, like "2002-01-04", or a time interval. The time interval is an integer followed by the character s, m, h, D, W, M, or Y, indicating seconds, minutes, hours, days, weeks, months, or years respectively, or a number of these concatenated. For example, 32m means 32 minutes, and 3W2D10h7s means 3 weeks, 2 days, 10 hours, and 7 seconds. In this context, a month means 30 days, a year is 365 days, and a day is always 86400 seconds.

rdiff-backup cannot remove-older-than and back up or restore in a single session. In order to both backup a directory and remove old files in it, you must run rdiff-backup twice.

By default, rdiff-backup will only delete information from one session at a time. To remove two or more sessions at the same time, supply the **--force** option (rdiff-backup will tell you if **--force** is required).

Note that snapshots of deleted files are covered by this operation. Thus if you deleted a file two weeks ago, backed up immediately afterwards, and then ran rdiff-backup with **--remove-older-than 10D** today, no trace of that file would remain. Finally, file selection options such as **--include** and **--exclude** don't affect **--remove-older-than**.

--restrict *path*

Require that all file access be inside the given path. This switch, and the following two, are intended to be used with the **--server** switch to provide a bit more protection when doing automated remote backups. They are **not intended as your only line of defense** so please don't do

something silly like allow public access to an rdiff-backup server run with `--restrict-read-only`.

--restrict-read-only *path*

Like `--restrict`, but also reject all write requests.

--restrict-update-only *path*

Like `--restrict`, but only allow writes as part of an incremental backup. Requests for other types of writes (for instance, deleting *path*) will be rejected.

--server

Enter server mode (not to be invoked directly, but instead used by another rdiff-backup process on a remote computer).

--ssh-no-compression

When running ssh, do not use the `-C` option to enable compression. `--ssh-no-compression` is ignored if you specify a new schema using `--remote-schema`.

--tempdir *path*

Sets the directory that rdiff-backup uses for temporary files to the given path. The environment variables `TMPDIR`, `TEMP`, and `TMP` can also be used to set the temporary files directory. See the documentation of the Python `tempfile` module for more information.

--terminal-verbosity *[0-9]*

Select which messages will be displayed to the terminal. If missing the level defaults to the verbosity level.

--test-server

Test for the presence of a compatible rdiff-backup server as specified in the following `host::filename` argument(s). The filename section will be ignored.

--user-mapping-file *filename*

Map user names and ids according to the user mapping file *filename*. See the **USERS AND GROUPS** section for more information.

-v*[0-9]*, **--verbosity** *[0-9]*

Specify verbosity level (0 is totally silent, 3 is the default, and 9 is noisiest). This determines how much is written to the log file.

--verify

This is short for `--verify-at-time now`

--verify-at-time *now*

Check all the data in the repository at the given time by computing the SHA1 hash of all the regular files and comparing them with the hashes stored in the metadata file.

-V, --version

Print the current version and exit

RESTORING

There are two ways to tell rdiff-backup to restore a file or directory. Firstly, you can run rdiff-backup on a mirror file and use the `-r` or `--restore-as-of` options. Secondly, you can run it on an increment file.

For example, suppose in the past you have run:

```
rdiff-backup /usr /usr.backup
```

to back up the `/usr` directory into the `/usr.backup` directory, and now want a copy of the `/usr/local` directory the way it was 3 days ago placed at `/usr/local.old`.

One way to do this is to run:

```
rdiff-backup -r 3D /usr.backup/local /usr/local.old
```

where above the "3D" means 3 days (for other ways to specify the time, see the **TIME FORMATS** section). The `/usr.backup/local` directory was selected, because that is the directory containing the current

version of /usr/local.

Note that the option to **—restore-as-of** always specifies an exact time. (So "3D" refers to the instant 72 hours before the present.) If there was no backup made at that time, rdiff-backup restores the state recorded for the previous backup. For instance, in the above case, if "3D" is used, and there are only backups from 2 days and 4 days ago, /usr/local as it was 4 days ago will be restored.

The second way to restore files involves finding the corresponding increment file. It would be in the /backup/rdiff-backup-data/increments/usr directory, and its name would be something like "local.2002-11-09T12:43:53-04:00.dir" where the time indicates it is from 3 days ago. Note that the increment files all end in ".diff", ".snapshot", ".dir", or ".missing", where ".missing" just means that the file didn't exist at that time (finally, some of these may be gzip-compressed, and have an extra ".gz" to indicate this). Then running:

```
rdiff-backup /backup/rdiff-backup-data/increments/usr/local.<time>.dir /usr/local.old
```

would also restore the file as desired.

If you are not sure exactly which version of a file you need, it is probably easiest to either restore from the increments files as described immediately above, or to see which increments are available with `-l/--list-increments`, and then specify exact times into `-r/--restore-as-of`.

TIME FORMATS

rdiff-backup uses time strings in two places. Firstly, all of the increment files rdiff-backup creates will have the time in their filenames in the w3 datetime format as described in a w3 note at <http://www.w3.org/TR/NOTE-datetime>. Basically they look like "2001-07-15T04:09:38-07:00", which means what it looks like. The "-07:00" section means the time zone is 7 hours behind UTC.

Secondly, the `-r`, **—restore-as-of**, and **—remove-older-than** options take a time string, which can be given in any of several formats:

1. the string "now" (refers to the current time)
2. a sequences of digits, like "123456890" (indicating the time in seconds after the epoch)
3. A string like "2002-01-25T07:00:00+02:00" in datetime format
4. An interval, which is a number followed by one of the characters s, m, h, D, W, M, or Y (indicating seconds, minutes, hours, days, weeks, months, or years respectively), or a series of such pairs. In this case the string refers to the time that preceded the current time by the length of the interval. For instance, "1h78m" indicates the time that was one hour and 78 minutes ago. The calendar here is unsophisticated: a month is always 30 days, a year is always 365 days, and a day is always 86400 seconds.
5. A date format of the form YYYY/MM/DD, YYYY-MM-DD, MM/DD/YYYY, or MM-DD-YYYY, which indicates midnight on the day in question, relative to the current timezone settings. For instance, "2002/3/5", "03-05-2002", and "2002-3-05" all mean March 5th, 2002.
6. A backup session specification which is a non-negative integer followed by 'B'. For instance, '0B' specifies the time of the current mirror, and '3B' specifies the time of the 3rd newest increment.

REMOTE OPERATION

In order to access remote files, rdiff-backup opens up a pipe to a copy of rdiff-backup running on the remote machine. Thus rdiff-backup must be installed on both ends. To open this pipe, rdiff-backup first splits the filename into `host_info::pathname`. It then substitutes `host_info` into the remote schema, and runs the resulting command, reading its input and output.

The default remote schema is `'ssh -C %s rdiff-backup --server'` where `host_info` is substituted for `'%s'`. So if the `host_info` is `user@host.net`, then rdiff-backup runs `'ssh user@host.net rdiff-backup --server'`. Using `—remote-schema`, rdiff-backup can invoke an arbitrary command in order to open up a remote pipe.

For instance,

```
rdiff-backup --remote-schema 'cd /usr; %s' foo `rdiff-backup --server':bar
is basically equivalent to (but slower than)
rdiff-backup foo /usr/bar
```

Concerning quoting, if for some reason you need to put two consecutive colons in the `host_info` section of a `host_info:pathname` argument, or in the pathname of a local file, you can quote one of them by prepending a backslash. So in `'a\::b::c'`, `host_info` is `'a::b'` and the pathname is `'c'`. Similarly, if you want to refer to a local file whose filename contains two consecutive colons, like `'strange::file'`, you'll have to quote one of the colons as in `'strange\::file'`. Because the backslash is a quote character in these circumstances, it too must be quoted to get a literal backslash, so `'foo\::\bar'` evaluates to `'foo::\bar'`. To make things more complicated, because the backslash is also a common shell quoting character, you may need to type in `'\\'` at the shell prompt to get a literal backslash (if it makes you feel better, I had to type in 8 backslashes to get that in this man page...). And finally, to include a literal % in the string specified by `--remote-schema`, quote it with another %, as in `% %`.

Although ssh itself may be secure, using `rdiff-backup` in the default way presents some security risks. For instance if the server is run as root, then an attacker who compromised the client could then use `rdiff-backup` to overwrite arbitrary server files by "backing up" over them. Such a setup can be made more secure by using the `sshd` configuration option `command="rdiff-backup --server"` possibly along with the `--restrict*` options to `rdiff-backup`. For more information, see the web page, the wiki, and the entries for the `--restrict*` options on this man page.

FILE SELECTION

`rdiff-backup` has a number of file selection options. When `rdiff-backup` is run, it searches through the given source directory and backs up all the files matching the specified options. This selection system may appear complicated, but it is supposed to be flexible and easy-to-use. If you just want to learn the basics, first look at the selection examples in the `examples.html` file included in the package, or on the web at <http://rdiff-backup.nongnu.org/examples.html>

`rdiff-backup`'s selection system was originally inspired by `rsync(1)`, but there are many differences. (For instance, trailing backslashes have no special significance.)

The file selection system comprises a number of file selection conditions, which are set using one of the following command line options: `--exclude`, `--exclude-filelist`, `--exclude-device-files`, `--exclude-fifos`, `--exclude-sockets`, `--exclude-symbolic-links`, `--exclude-globbing-filelist`, `--exclude-globbing-filelist-stdin`, `--exclude-filelist-stdin`, `--exclude-regexp`, `--exclude-special-files`, `--include`, `--include-filelist`, `--include-globbing-filelist`, `--include-globbing-filelist-stdin`, `--include-filelist-stdin`, and `--include-regexp`. Each file selection condition either matches or doesn't match a given file. A given file is excluded by the file selection system exactly when the first matching file selection condition specifies that the file be excluded; otherwise the file is included. When backing up, if a file is excluded, `rdiff-backup` acts as if that file does not exist in the source directory. When restoring, an excluded file is considered not to exist in either the source or target directories.

For instance,

```
rdiff-backup --include /usr --exclude /usr /usr /backup
is exactly the same as
rdiff-backup /usr /backup
```

because the include and exclude directives match exactly the same files, and the `--include` comes first, giving it precedence. Similarly,

```
rdiff-backup --include /usr/local/bin --exclude /usr/local /usr /backup
would backup the /usr/local/bin directory (and its contents), but not /usr/local/doc.
```


The **include**, **exclude**, **include-globbing-filelist**, and **exclude-globbing-filelist** options accept *extended shell globbing patterns*. These patterns can contain the special patterns `*`, `**`, `?`, and `[...]`. As in a normal shell, `*` can be expanded to any string of characters not containing `/`, `?` expands to any character except `/`, and `[...]` expands to a single character of those characters specified (ranges are acceptable). The new special pattern, `**`, expands to any string of characters whether or not it contains `/`. Furthermore, if the pattern starts with "ignorecase:" (case insensitive), then this prefix will be removed and any character in the string can be replaced with an upper- or lowercase version of itself.

If you need to match filenames which contain the above globbing characters, they may be escaped using a backslash `"\"`. The backslash will only escape the character following it so for `**` you will need to use `"**"` to avoid escaping it to the `*` globbing character.

Remember that you may need to quote these characters when typing them into a shell, so the shell does not interpret the globbing patterns before `rdiff-backup` sees them.

The `--exclude pattern` option matches a file iff:

1. `pattern` can be expanded into the file's filename, or
2. the file is inside a directory matched by the option.

Conversely, `--include pattern` matches a file iff:

1. `pattern` can be expanded into the file's filename,
2. the file is inside a directory matched by the option, or
3. the file is a directory which contains a file matched by the option.

For example,

```
--exclude /usr/local
```

matches `/usr/local`, `/usr/local/lib`, and `/usr/local/lib/netcape`. It is the same as `--exclude /usr/local --exclude '/usr/local/**'`.

```
--include /usr/local
```

specifies that `/usr`, `/usr/local`, `/usr/local/lib`, and `/usr/local/lib/netcape` (but not `/usr/doc`) all be backed up. Thus you don't have to worry about including parent directories to make sure that included subdirectories have somewhere to go. Finally,

```
--include ignorecase:'/usr/[a-z0-9]foo/**.py'
```

would match a file like `/usr/5fOO/hello/there/world.py`. If it did match anything, it would also match `/usr`. If there is no existing file that the given pattern can be expanded into, the option will not match `/usr`.

The `--include-filelist`, `--exclude-filelist`, `--include-filelist-stdin`, and `--exclude-filelist-stdin` options also introduce file selection conditions. They direct `rdiff-backup` to read in a file, each line of which is a file specification, and to include or exclude the matching files. Lines are separated by newlines or nulls, depending on whether the `--null-separator` switch was given. Each line in a filelist is interpreted similarly to the way *extended shell patterns* are, with a few exceptions:

1. Globbing patterns like `*`, `**`, `?`, and `[...]` are not expanded.
2. Include patterns do not match files in a directory that is included. So `/usr/local` in an include file will not match `/usr/local/doc`.
3. Lines starting with `"+"` are interpreted as include directives, even if found in a filelist referenced by `--exclude-filelist`. Similarly, lines starting with `"-"` exclude files even if they are found within an include filelist.

For example, if the file `"list.txt"` contains the lines:

```

/usr/local
- /usr/local/doc
/usr/local/bin
+ /var
- /var

```

then "`--include-filelist list.txt`" would include `/usr`, `/usr/local`, and `/usr/local/bin`. It would exclude `/usr/local/doc`, `/usr/local/doc/python`, etc. It neither excludes nor includes `/usr/local/man`, leaving the fate of this directory to the next specification condition. Finally, it is undefined what happens with `/var`. A single file list should not contain conflicting file specifications.

The `--include-globbing-filelist` and `--exclude-globbing-filelist` options also specify filelists, but each line in the filelist will be interpreted as a globbing pattern the way `--include` and `--exclude` options are interpreted (although "+" and "-" prefixing is still allowed). For instance, if the file "globbing-list.txt" contains the lines:

```

dir/foo
+ dir/bar
- **

```

Then "`--include-globbing-filelist globbing-list.txt`" would be exactly the same as specifying "`--include dir/foo --include dir/bar --exclude **`" on the command line.

Finally, the `--include-regexp` and `--exclude-regexp` allow files to be included and excluded if their filenames match a python regular expression. Regular expression syntax is too complicated to explain here, but is covered in Python's library reference. Unlike the `--include` and `--exclude` options, the regular expression options don't match files containing or contained in matched files. So for instance

```

--include '[0-9]{7}(?!foo)'

```

matches any files whose full pathnames contain 7 consecutive digits which aren't followed by 'foo'. However, it wouldn't match `/home` even if `/home/ben/1234567` existed.

USERS AND GROUPS

There can be complications preserving ownership across systems. For instance the username that owns a file on the source system may not exist on the destination. Here is how `rdiff-backup` maps ownership on the source to the destination (or vice-versa, in the case of restoring):

1. If the `--preserve-numerical-ids` option is given, the remote files will always have the same uid and gid, both for ownership and ACL entries. This may cause unames and gnames to change.
2. Otherwise, attempt to preserve the user and group names for ownership and in ACLs. This may result in files having different uids and gids across systems.
3. If a name cannot be preserved (e.g. because the username does not exist), preserve the original id, but only in cases of user and group ownership. For ACLs, omit any entry that has a bad user or group name.
4. The `--user-mapping-file` and `--group-mapping-file` options override this behavior. If either of these options is given, the policy described in 2 and 3 above will be followed, but with the mapped user and group instead of the original. If you specify both `--preserve-numerical-ids` and one of the mapping options, the behavior is undefined.

The user and group mapping files both have the same form:

```

old_name_or_id1:new_name_or_id1
old_name_or_id2:new_name_or_id2

```

<etc>

Each line should contain a name or id, followed by a colon ":", followed by another name or id. If a name or id is not listed, they are treated in the default way described above.

When restoring, the above behavior is also followed, but note that the original source user/group information will be the input, not the already mapped user/group information present in the backup repository. For instance, suppose you have mapped all the files owned by *alice* in the source so that they are owned by *ben* in the repository, and now you want to restore, making sure the files owned originally by *alice* are still owned by *alice*. In this case there is no need to use any of the mapping options. However, if you wanted to restore the files so that the files originally owned by *alice* on the source are now owned by *ben*, you would have to use the mapping options, even though you just want the unames of the repository's files preserved in the restored files.

STATISTICS

Every session `rdiff-backup` saves various statistics into two files, the session statistics file at `rdiff-backup-data/session_statistics.<time>.data` and the directory statistics file at `rdiff-backup-data/directory_statistics.<time>.data`. They are both text files and contain similar information: how many files changed, how many were deleted, the total size of increment files created, etc. However, the session statistics file is intended to be very readable and only describes the session as a whole. The directory statistics file is more compact (and slightly less readable) but describes every directory backed up. It also may be compressed to save space.

Statistics-related options include `--print-statistics` and `--null-separator`.

Also, `rdiff-backup` will save various messages to the log file, which is `rdiff-backup-data/backup.log` for backup sessions and `rdiff-backup-data/restore.log` for restore sessions. Generally what is written to this file will coincide with the messages displayed to `stdout` or `stderr`, although this can be changed with the `--terminal-verbosity` option.

The log file is not compressed and can become quite large if `rdiff-backup` is run with high verbosity.

EXIT STATUS

If `rdiff-backup` finishes successfully, the exit status will be 0. If there is an unrecoverable (critical) error, it will be non-zero (usually 1, but don't depend on this specific value). When setting up `rdiff-backup` to run automatically (as from `cron(8)` or similar) it is probably a good idea to check the exit code.

BUGS

The `gzip` library in versions 2.2 and earlier of `python` (but fixed in 2.3a1) has trouble producing files over 2GB in length. This bug will prevent `rdiff-backup` from producing large compressed increments (snapshots or diffs). A workaround is to disable compression for large uncompressable files.

AUTHOR

Ben Escoto <ben@emerose.org>

Feel free to ask me questions or send me bug reports, but you may want to see the web page, mentioned below, first.

SEE ALSO

`python(1)`, `rdiff(1)`, `rsync(1)`, `ssh(1)`. The main `rdiff-backup` web page is at <http://rdiff-backup.nongnu.org/>. It has more information, links to the mailing list and CVS, etc.

