NAME

virt−ls − List files in a virtual machine

SYNOPSIS

```
virt−ls [−−options] −d domname directory [directory ...]

virt−ls [−−options] −a disk.img [−a disk.img ...] directory [directory ...]
```

Old style:

```
virt−ls [−−options] domname directory

virt−ls [−−options] disk.img [disk.img ...] directory
```

DESCRIPTION

`virt−ls` lists filenames, file sizes, checksums, extended attributes and more from a virtual machine or disk image.

Multiple directory names can be given, in which case the output from each is concatenated.

To list directories from a libvirt guest use the *−d* option to specify the name of the guest. For a disk image, use the *−a* option.

`virt−ls` can do many simple file listings. For more complicated cases you may need to use **guestfish** (1), or write a program directly to the **guestfs** (3) API.

EXAMPLES

Get a list of all files and directories in a virtual machine:

```
virt−ls −R −d guest /
```

List all setuid or setgid programs in a Linux virtual machine:

```
virt−ls −lR −d guest / | grep '^− [42]'
```

List all public-writable directories in a Linux virtual machine:

```
virt−ls −lR −d guest / | grep '^d ...7'
```

List all Unix domain sockets in a Linux virtual machine:

```
virt−ls −lR −d guest / | grep '^s'
```

List all regular files with filenames ending in ‘.png’:

```
virt−ls −lR −d guest / | grep −i '^−.*\.png$'
```

To display files larger than 10MB in home directories:

```
virt−ls −lR −d guest /home | awk '$3 > 10*1024*1024'
```

Find everything modified in the last 7 days:

```
virt−ls −lR −d guest −−time−days / | awk '$6 <= 7'
```

Find regular files modified in the last 24 hours:

```
virt−ls −lR −d guest −−time−days / | grep '^−' | awk '$6 < 1'
```

DIFFERENCES IN SNAPSHOTS AND BACKING FILES

Although it is possible to use virt-ls to look for differences, since libguestfs ≥ 1.26 a new tool is available called **virt−diff** (1).

OUTPUT MODES

`virt−ls` has four output modes, controlled by different combinations of the *−l* and *−R* options.

SIMPLE LISTING

A simple listing is like the ordinary **ls** (1) command:

```
$ virt-ls -d guest /
bin
boot
[etc.]
```

LONG LISTING

With the *-l* (*--long*) option, the output is like the `ls -l` command (more specifically, like the `guestfs_ll` function).

```
$ virt-ls -l -d guest /
total 204
dr-xr-xr-x.   2 root root   4096 2009-08-25 19:06 bin
dr-xr-xr-x.   5 root root   3072 2009-08-25 19:06 boot
[etc.]
```

Note that while this is useful for displaying a directory, do not try parsing this output in another program. Use ‘‘RECURSIVE LONG LISTING’’ instead.

RECURSIVE LISTING

With the *-R* (*--recursive*) option, `virt-ls` lists the names of files and directories recursively:

```
$ virt-ls -R -d guest /tmp
foo
foo/bar
[etc.]
```

To generate this output, `virt-ls` runs the `guestfs_find0` function and converts `\0` characters to `\n`.

RECURSIVE LONG LISTING

Using *-lR* options together changes the output to display directories recursively, with file stats, and optionally other features such as checksums and extended attributes.

Most of the interesting features of `virt-ls` are only available when using *-lR* mode.

The fields are normally space-separated. Filenames are **not** quoted, so you cannot use the output in another program (because filenames can contain spaces and other unsafe characters). If the guest was untrusted and someone knew you were using `virt-ls` to analyze the guest, they could play tricks on you by creating filenames with embedded newline characters. To **safely** parse the output in another program, use the *--csv* (Comma-Separated Values) option.

Note that this output format is completely unrelated to the `ls -lR` command.

```
$ virt-ls -lR -d guest /bin
d 0555       4096 /bin
- 0755        123 /bin/alsaunmute
- 0755      28328 /bin/arch
l 0777          4 /bin/awk -> gawk
- 0755      27216 /bin/basename
- 0755     943360 /bin/bash
[etc.]
```

These basic fields are always shown:

type
: The file type, one of: `-` (regular file), `d` (directory), `c` (character device), `b` (block device), `p` (named pipe), `l` (symbolic link), `s` (socket) or `u` (unknown).

permissions
: The Unix permissions, displayed as a 4 digit octal number.

size
: The size of the file. This is shown in bytes unless *-h* or *--human-readable* option is given, in which case this is shown as a human-readable number.

path
: The full path of the file or directory.

link
: For symbolic links only, the link target.

In *−lR* mode, additional command line options enable the display of more fields.

With the *−−uids* flag, these additional fields are displayed before the path:

uid
gid
: The UID and GID of the owner of the file (displayed numerically). Note these only make sense in the context of a Unix-like guest.

With the *−−times* flag, these additional fields are displayed:

atime
: The time of last access.

mtime
: The time of last modification.

ctime
: The time of last status change.

The time fields are displayed as string dates and times, unless one of the *−−time−t*, *−−time−relative* or *−−time−days* flags is given.

With the *−−extra−stats* flag, these additional fields are displayed:

device
: The device containing the file (displayed as major:minor). This may not match devices as known to the guest.

inode
: The inode number.

nlink
: The number of hard links.

rdev
: For block and char special files, the device (displayed as major:minor).

blocks
: The number of 512 byte blocks allocated to the file.

With the *−−checksum* flag, the checksum of the file contents is shown (only for regular files). Computing file checksums can take a considerable amount of time.

OPTIONS

−−help
: Display brief help.

−a file
−−add file
: Add *file* which should be a disk image from a virtual machine. If the virtual machine has multiple block devices, you must supply all of them with separate *−a* options.

  The format of the disk image is auto-detected. To override this and force a particular format use the *−−format=*.. option.

−a URI
−−add URI
: Add a remote disk. See ‘‘ADDING REMOTE STORAGE’’ in **guestfish** (1).

--blocksize=512
--blocksize=4096
--blocksize
This parameter sets the sector size of the disk image. It affects all explicitly added subsequent disks after this parameter. Using *--bloc ksize* with no argument switches the disk sector size to the default value which is usually 512 bytes. See also ''guestfs_add_drive_opts'' in **guestfs**(3).

--checksum
--checksum=crc|md5|sha1|sha224|sha256|sha384|sha512
Display checksum over file contents for regular files. With no argument, this defaults to using *md5*. Using an argument, you can select the checksum type to use.

This option only has effect in *-lR* output mode. See ''RECURSIVE LONG LISTING'' above.

-c URI
--connect URI
If using libvirt, connect to the given *URI*. If omitted, then we connect to the default libvirt hypervisor.

If you specify guest block devices directly (*-a*), then libvirt is not used at all.

--csv
Write out the results in CSV format (comma-separated values). This format can be imported easily into databases and spreadsheets, but read ''NOTE ABOUT CSV FORMAT'' below.

-d guest
--domain guest
Add all the disks from the named libvirt guest. Domain UUIDs can be used instead of names.

--echo-keys
When prompting for keys and passphrases, virt-ls normally turns echoing off so you cannot see what you are typing. If you are not worried about Tempest attacks and there is no one else in the room you can specify this flag to see what you are typing.

--extra-stats
Display extra stats.

This option only has effect in *-lR* output mode. See ''RECURSIVE LONG LISTING'' above.

--format=raw|qcow2|..
--format
The default for the *-a* option is to auto-detect the format of the disk image. Using this forces the disk format for *-a* options which follow on the command line. Using *--format* with no argument switches back to auto-detection for subsequent *-a* options.

For example:

```
virt-ls --format=raw -a disk.img /dir
```

forces raw format (no auto-detection) for *disk.img*.

```
virt-ls --format=raw -a disk.img --format -a another.img /dir
```

forces raw format (no auto-detection) for *disk.img* and reverts to auto-detection for *another.img*.

If you have untrusted raw-format guest disk images, you should use this option to specify the disk format. This avoids a possible security problem with malicious guests (CVE-2010-3851).

-h
--human-readable
Display file sizes in human-readable format.

This option only has effect in *-lR* output mode. See ''RECURSIVE LONG LISTING'' above.

--key SELECTOR
Specify a key for LUKS, to automatically open a LUKS device when using the inspection. `ID` can be either the libguestfs device name, or the UUID of the LUKS device.

--key `ID`:key:KEY_STRING
Use the specified `KEY_STRING` as passphrase.

--key `ID`:file:FILENAME
Read the passphrase from *FILENAME*.

--keys-from-stdin
Read key or passphrase parameters from stdin. The default is to try to read passphrases from the user by opening */dev/tty*.

If there are multiple encrypted devices then you may need to supply multiple keys on stdin, one per line.

-m dev[:mountpoint[:options[:fstype]]]
--mount dev[:mountpoint[:options[:fstype]]]
Mount the named partition or logical volume on the given mountpoint.

If the mountpoint is omitted, it defaults to */*.

Specifying any mountpoint disables the inspection of the guest and the mount of its root and all of its mountpoints, so make sure to mount all the mountpoints needed to work with the filenames given as arguments.

If you don't know what filesystems a disk image contains, you can either run guestfish without this option, then list the partitions, filesystems and LVs available (see ''list-partitions'', ''list-filesystems'' and ''lvs'' commands), or you can use the **virt-filesystems** (1) program.

The third (and rarely used) part of the mount parameter is the list of mount options used to mount the underlying filesystem. If this is not given, then the mount options are either the empty string or `ro` (the latter if the *--ro* flag is used). By specifying the mount options, you override this default choice. Probably the only time you would use this is to enable ACLs and/or extended attributes if the filesystem can support them:

```
-m /dev/sda1:/:acl,user_xattr
```

Using this flag is equivalent to using the `mount-options` command.

The fourth part of the parameter is the filesystem driver to use, such as `ext3` or `ntfs`. This is rarely needed, but can be useful if multiple drivers are valid for a filesystem (eg: `ext2` and `ext3`), or if libguestfs misidentifies a filesystem.

-l
--long
-R
--recursive
Select the mode. With neither of these options, `virt-ls` produces a simple, flat list of the files in the named directory. See ''SIMPLE LISTING''.

`virt-ls -l`produces a ' 'long listing'', which shows more detail. See ''LONG LISTING''.

`virt-ls -R`produces a recursi ve list of files starting at the named directory. See ''RECURSIVE LISTING''.

`virt-ls -lR`produces a recursi ve long listing which can be more easily parsed. See ''RECURSIVE LONG LISTING''.

--times
Display time fields.

This option only has effect in *-lR* output mode. See ''RECURSIVE LONG LISTING'' above.

−−time−days
Display time fields as days before now (negative if in the future).

Note that `0` in output means ‘‘up to 1 day before now’’, or that the age of the file is between 0 and 86399 seconds.

This option only has effect in *−lR* output mode. See ‘‘RECURSIVE LONG LISTING’’ above.

−−time−relative
Display time fields as seconds before now (negative if in the future).

This option only has effect in *−lR* output mode. See ‘‘RECURSIVE LONG LISTING’’ above.

−−time−t
Display time fields as seconds since the Unix epoch.

This option only has effect in *−lR* output mode. See ‘‘RECURSIVE LONG LISTING’’ above.

−−uids
Display UID and GID fields.

This option only has effect in *−lR* output mode. See ‘‘RECURSIVE LONG LISTING’’ above.

−v
−−verbose
Enable verbose messages for debugging.

−V
−−version
Display version number and exit.

−x Enable tracing of libguestfs API calls.

OLD-STYLE COMMAND LINE ARGUMENTS

Previous versions of virt-ls allowed you to write either:

```
virt−ls disk.img [disk.img ...] /dir
```

or

```
virt−ls guestname /dir
```

whereas in this version you should use *−a* or *−d* respectively to avoid the confusing case where a disk image might have the same name as a guest.

For compatibility the old style is still supported.

NOTE ABOUT CSV FORMAT

Comma-separated values (CSV) is a deceptive format. It *seems* like it should be easy to parse, but it is definitely not easy to parse.

Myth: Just split fields at commas. Reality: This does *not* work reliably. This example has two columns:

```
"foo,bar",baz
```

Myth: Read the file one line at a time. Reality: This does *not* work reliably. This example has one row:

```
"foo
bar",baz
```

For shell scripts, use `csvtool` (https://github.com/Chris00/ocaml−csv also packaged in major Linux distributions).

For other languages, use a CSV processing library (eg. `Text::CSV` for Perl or Python’s built-in csv library).

Most spreadsheets and databases can import CSV directly.

EXIT STATUS

This program returns 0 if successful, or non-zero if there was an error.

SEE ALSO

guestfs (3), **guestfish** (1), **virt−cat** (1), **virt−copy−out** (1), **virt−diff** (1), **virt−tar−out** (1), http://libguestfs.org/.

AUTHOR

Richard W.M. Jones http://people.redhat.com/~rjones/

COPYRIGHT

LICENSE

BUGS

To get a list of bugs against libguestfs, use this link: https://bugzilla.redhat.com/buglist.cgi?component=libguestfs&product=Virtualization+Tools

To report a new bug against libguestfs, use this link: https://bugzilla.redhat.com/enter_bug.cgi?component=libguestfs&product=Virtualization+Tools

When reporting a bug, please supply:

- The version of libguestfs.
- Where you got libguestfs (eg. which Linux distro, compiled from source, etc)
- Describe the bug accurately and give a way to reproduce it.
- Run **libguestfs−test−tool** (1) and paste the **complete, unedited** output into the bug report.