NAME

virt−filesystems − List filesystems, partitions, block devices, LVM in a virtual machine or disk image

SYNOPSIS

```
virt−filesystems [−−options] −d domname

virt−filesystems [−−options] −a disk.img [−a disk.img ...]
```

DESCRIPTION

This tool allows you to discover filesystems, partitions, logical volumes, and their sizes in a disk image or virtual machine. It is a replacement for **virt−list−filesystems** (1) and **virt−list−partitions** (1).

One use for this tool is from shell scripts to iterate over all filesystems from a disk image:

```
for fs in $(virt−filesystems −a disk.img); do
  # ...
done
```

Another use is to list partitions before using another tool to modify those partitions (such as **virt−resize** (1)). If you are curious about what an unknown disk image contains, use this tool along with **virt−inspector** (1).

Various command line options control what this program displays. You need to give either *−a* or *−d* options to specify the disk image or libvirt guest respectively. If you just specify that then the program shows filesystems found, one per line, like this:

```
$ virt−filesystems −a disk.img
/dev/sda1
/dev/vg_guest/lv_root
```

If you add *−l* or *−−long* then the output includes extra information:

```
$ virt−filesystems −a disk.img −l
Name                  Type         VFS   Label  Size
/dev/sda1             filesystem   ext4  boot   524288000
/dev/vg_guest/lv_root  filesystem   ext4  root   10212081664
```

If you add *−−extra* then non-mountable (swap, unknown) filesystems are shown as well:

```
$ virt−filesystems −a disk.img −−extra
/dev/sda1
/dev/vg_guest/lv_root
/dev/vg_guest/lv_swap
/dev/vg_guest/lv_data
```

If you add *−−partitions* then partitions are shown instead of filesystems:

```
$ virt−filesystems −a disk.img −−partitions
/dev/sda1
/dev/sda2
```

Similarly you can use *−−logical−volumes*, *−−volume−groups*, *−−physical−volumes*, *−−block−devices* to list those items.

You can use these options in combination as well (if you want a combination including filesystems, you have to add *−−filesystems*). Notice that some items fall into several categories (eg. */dev/sda1* might be both a partition and a filesystem). These items are listed several times. To get a list which includes absolutely everything that virt-filesystems knows about, use the *−−all* option.

UUIDs (because they are quite long) are not shown by default. Add the*−−uuid* option to display de vice and filesystem UUIDs in the long output.

−−all −−long −−uuid is a useful combination to display all possible information about everything.

```
$ virt-filesystems -a win.img --all --long --uuid -h
Name      Type       VFS  Label           Size Parent   UUID
/dev/sda1 filesystem ntfs System Reserved 100M -        F81C92571C92112C
/dev/sda2 filesystem ntfs -               20G  -        F2E8996AE8992E3B
/dev/sda1 partition  -    -               100M /dev/sda -
/dev/sda2 partition  -    -               20G  /dev/sda -
/dev/sda  device     -    -               20G  -        -
```

For machine-readable output, use *--csv* to get Comma-Separated Values.

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).

--all
Display everything. This is currently the same as specifying these options:*--filesystems*, *--extra*, *--partitions*, *--block-devices*, *--logical-volumes*, *--volume-groups*, *--physical-volumes*. (More may be added to this list in future).

See also *--long*.

--blkdevs
--block-devices
Display block devices.

--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).

-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-filesystems 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
This causes filesystems that are not ordinary, mountable filesystems to be displayed. This category includes swapspace, and filesystems that are empty or contain unknown data.

This option implies *--filesystems*.

--filesystems
Display mountable filesystems. If no display option was selected then this option is implied.

With *--extra*, non-mountable filesystems are shown too.

--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-filesystems --format=raw -a disk.img
```

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

```
virt-filesystems --format=raw -a disk.img --format -a another.img
```

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
In *--long* mode, display sizes in human-readable format.

--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.

-l
--long
Display extra columns of data (‘‘long format’’).

A title row is added unless you also specify *--no-title*.

The extra columns displayed depend on what output you select, and the ordering of columns may change in future versions. Use the title row, *--csv* output and/or **csvtool** (1) to match columns to data in external programs.

Use *-h* if you want sizes to be displayed in human-readable format. The default is to show raw numbers of *bytes*.

Use *--uuid* to display UUIDs too.

--lvs
--logvols
--logical-volumes
Display LVM logical volumes. In this mode, these are displayed irrespective of whether the LVs contain filesystems.

--no-title
In *--long* mode, don’t add a title row.

Note that the order of the columns is not fixed, and may change in future versions of virt-filesystems,

so using this option may give you unexpected surprises.

−−parts
−−partitions
Display partitions. In this mode, these are displayed irrespective of whether the partitions contain filesystems.

−−pvs
−−physvols
−−physical−volumes
Display LVM physical volumes.

−−uuid
−−uuids
In *−−long* mode, display UUIDs as well.

−v
−−verbose
Enable verbose messages for debugging.

−V
−−version
Display version number and exit.

−−vgs
−−volgroups
−−volume−groups
Display LVM volume groups.

−x Enable tracing of libguestfs API calls.

COLUMNS

Note that columns in the output are subject to reordering and change in future versions of this tool.

Name
The filesystem, partition, block device or LVM name.

For device and partition names these are displayed as canonical libguestfs names, so that for example */dev/sda2* is the second partition on the first device.

If the *−−long* option is **not** specified, then only the name column is shown in the output.

Type
The object type, for example `filesystem`, `lv`, `device` etc.

VFS
If there is a filesystem, then this column displays the filesystem type if one could be detected, eg. `ext4`.

Label
If the object has a label (used for identifying and mounting filesystems) then this column contains the label.

MBR
The partition type byte, displayed as a two digit hexadecimal number. A comprehensive list of partition types can be found here: http://www.win.tue.nl/~aeb/partitions/partition_types−1.html

This is only applicable for DOS (MBR) partitions.

Size
The size of the object in bytes. If the *−−human* option is used then the size is displayed in a human-readable form.

Parent

The parent column records the parent relationship between objects.

For example, if the object is a partition, then this column contains the name of the containing device. If the object is a logical volume, then this column is the name of the volume group.

If there is more than one parent, then this column is (internal to the column) a comma-separated list, eg. `/dev/sda,/dev/sdb`.

UUID

If the object has a UUID (used for identifying and mounting filesystems and block devices) then this column contains the UUID as a string.

The UUID is only displayed if the *−−uuid* option is given.

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−df** (1), **virt−list−filesystems** (1), **virt−list−partitions** (1), **csvtool** (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.