NAME

virt−edit − Edit a file in a virtual machine

SYNOPSIS

```
virt−edit [−−options] −d domname file [file ...]

virt−edit [−−options] −a disk.img [−a disk.img ...] file [file ...]

virt−edit [−d domname|−a disk.img] file −e 'expr'
```

Old-style:

```
virt−edit domname file

virt−edit disk.img [disk.img ...] file
```

WARNING

Using `virt−edit` on live virtual machines, or concurrently with other disk editing tools, can be dangerous, potentially causing disk corruption. The virtual machine must be shut down before you use this command, and disk images must not be edited concurrently.

DESCRIPTION

`virt−edit` is a command line tool to edit `file` where each `file` exists in the named virtual machine (or disk image).

Multiple filenames can be given, in which case they are each edited in turn. Each filename must be a full path, starting at the root directory (starting with '/').

If you want to just view a file, use **virt−cat** (1).

For more complex cases you should look at the **guestfish** (1) tool (see ''USING GUESTFISH'' below).

`virt−edit` cannot be used to create a new file. **guestfish** (1) can do that and much more.

EXAMPLES

Edit the named files interactively:

```
virt−edit −d mydomain /boot/grub/grub.conf

virt−edit −d mydomain /etc/passwd
```

For Windows guests, some Windows paths are understood:

```
virt−edit −d mywindomain 'c:\autoexec.bat'
```

If Perl is installed, you can also edit files non-interactively (see ''NON-INTERACTIVE EDITING'' below). To change the init default level to 5:

```
virt−edit −d mydomain /etc/inittab −e 's/^id:.*/id:5:initdefault:/'
```

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).

−b EXTENSION
−−backup EXTENSION
Create a backup of the original file *in the guest disk image*. The backup has the original filename with `extension` added.

Usually the first character of `extension` would be a dot `.` so you would write:

```
virt−edit −b .orig [etc]
```

By default, no backup file is made.

−−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, then libvirt is not used at all.

−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-edit 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.

−e EXPR
−−edit EXPR
−−expr EXPR
Instead of launching the external editor, non-interactively apply the Perl expression `EXPR` to each line in the file. See ‘‘NON-INTERACTIVE EDITING’’ below.

Be careful to properly quote the expression to prevent it from being altered by the shell.

Note that this option is only available when Perl 5 is installed.

−−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−edit −−format=raw −a disk.img file
```

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

```
virt−edit −−format=raw −a disk.img −−format −a another.img file
```

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).

−−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.

-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-edit allowed you to write either:

```
virt-edit disk.img [disk.img ...] file
```

or

```
virt-edit guestname file
```

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.

NON-INTERACTIVE EDITING

`virt-edit` normally calls out to `$EDITOR` (or vi) so the system administrator can interactively edit the file.

There are two ways also to use `virt-edit` from scripts in order to make automated edits to files. (Note that although you *can* use `virt-edit` like this, it's less error-prone to write scripts directly using the libguestfs API and Augeas for configuration file editing.)

The first method is to temporarily set `$EDITOR` to any script or program you want to run. The script is invoked as `$EDITOR tmpfile` and it should update `tmpfile` in place however it likes.

The second method is to use the *-e* parameter of `virt-edit` to run a short Perl snippet in the style of **sed** (1). For example to replace all instances of `foo` with `bar` in a file:

```
virt-edit -d domname filename -e 's/foo/bar/'
```

The full power of Perl regular expressions can be used (see **perlre** (1)). For example to delete root's password you could do:

```
virt-edit -d domname /etc/passwd -e 's/^root:.*?:/root::/'
```

What really happens is that the snippet is evaluated as a Perl expression for each line of the file. The line, including the final `\n`, is passed in `$_` and the expression should update `$_` or leave it unchanged.

To delete a line, set `$_` to the empty string. For example, to delete the `apache` user account from the password file you can do:

```
virt-edit -d mydomain /etc/passwd -e '$_ = "" if /^apache:/'
```

To insert a line, prepend or append it to `$_`. However appending lines to the end of the file is rather difficult this way since there is no concept of "last line of the file" - your expression just doesn't get called again. You might want to use the first method (setting `$EDITOR`) if you want to do this.

The variable `$lineno` contains the current line number. As is traditional, the first line in the file is number `1`.

The return value from the expression is ignored, but the expression may call `die` in order to abort the whole program, leaving the original file untouched.

Remember when matching the end of a line that `$_` may contain the final `\n`, or (for DOS files) `\r\n`, or if the file does not end with a newline then neither of these. Thus to match or substitute some text at the end of a line, use this regular expression:

```
/some text(\r?\n)?$/
```

Alternately, use the perl `chomp` function, being careful not to chomp `$_` itself (since that would remove all newlines from the file):

```
my $m = $_; chomp $m; $m =~ /some text$/
```

WINDOWS PATHS

`virt-edit` has a limited ability to understand Windows drive letters and paths (eg. *E:\foo\bar.txt*).

If and only if the guest is running Windows then:

- Drive letter prefixes like `C:` are resolved against the Windows Registry to the correct filesystem.
- Any backslash (\) characters in the path are replaced with forward slashes so that libguestfs can process it.
- The path is resolved case insensitively to locate the file that should be edited.

There are some known shortcomings:

- Some NTFS symbolic links may not be followed correctly.
- NTFS junction points that cross filesystems are not followed.

USING GUESTFISH

guestfish (1) is a more powerful, lower level tool which you can use when `virt-edit` doesn't work.

Using `virt-edit` is approximately equivalent to doing:

```
guestfish --rw -i -d domname edit /file
```

where `domname` is the name of the libvirt guest, and */file* is the full path to the file.

The command above uses libguestfs's guest inspection feature and so does not work on guests that libguestfs cannot inspect, or on things like arbitrary disk images that don't contain guests. To edit a file on a disk image directly, use:

```
guestfish --rw -a disk.img -m /dev/sda1 edit /file
```

where *disk.img* is the disk image, */dev/sda1* is the filesystem within the disk image to edit, and */file* is the full path to the file.

`virt-edit` cannot create new files. Use the guestfish commands `touch`, `write` or `upload` instead:

```
guestfish --rw -i -d domname touch /newfile

guestfish --rw -i -d domname write /newfile "new content"

guestfish --rw -i -d domname upload localfile /newfile
```

ENVIRONMENT VARIABLES

`EDITOR`

If set, this string is used as the editor. It may contain arguments, eg. `"emacs -nw"`

If not set, `vi` is used.

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-in** (1), **virt-tar-in** (1), http://libguestfs.org/, **perl** (1), **perlre** (1).

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.