NAME

virt−customize − Customize a virtual machine

SYNOPSIS

```
 virt−customize
    [ −a disk.img [ −a disk.img ... ] | −d domname ]
    [−−attach ISOFILE] [−−attach−format FORMAT]
    [ −c URI | −−connect URI ] [ −n | −−dry−run ]
    [ −−format FORMAT] [ −m MB | −−memsize MB ]
    [ −−network | −−no−network ]
    [ −q | −−quiet ] [−−smp N] [ −v | −−verbose ] [−x]
    [−−append−line FILE:LINE] [−−chmod PERMISSIONS:FILE]
    [−−commands−from−file FILENAME] [−−copy SOURCE:DEST]
    [−−copy−in LOCALPATH:REMOTEDIR] [−−delete PATH] [−−edit FILE:EXPR]
    [−−firstboot SCRIPT] [−−firstboot−command 'CMD+ARGS']
    [−−firstboot−install PKG,PKG..] [−−hostname HOSTNAME]
    [−−install PKG,PKG..] [−−link TARGET:LINK[:LINK..]] [−−mkdir DIR]
    [−−move SOURCE:DEST] [−−password USER:SELECTOR]
    [−−root−password SELECTOR] [−−run SCRIPT]
    [−−run−command 'CMD+ARGS'] [−−scrub FILE] [−−sm−attach SELECTOR]
    [−−sm−register] [−−sm−remove] [−−sm−unregister]
    [−−ssh−inject USER[:SELECTOR]] [−−truncate FILE]
    [−−truncate−recursive PATH] [−−timezone TIMEZONE] [−−touch FILE]
    [−−uninstall PKG,PKG..] [−−update] [−−upload FILE:DEST]
    [−−write FILE:CONTENT] [−−no−logfile]
    [−−password−crypto md5|sha256|sha512] [−−selinux−relabel]
    [−−sm−credentials SELECTOR]


 virt−customize [ −V | −−version ]
```

WARNING

Using `virt−customize` 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-customize can customize a virtual machine (disk image) by installing packages, editing configuration files, and so on.

Virt-customize modifies the guest or disk image *in place*. The guest must be shut down. If you want to preserve the existing contents of the guest, *you must snapshot, copy or clone the disk first*.

You do *not* need to run virt-customize as root. In fact we'd generally recommend that you don't.

Related tools include: **virt−sysprep** (1) and **virt−builder** (1).

OPTIONS

−−help

Display brief help.

−a file
−−add file

Add *file* which should be a disk image from a virtual machine.

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. The URI format is compatible with guestfish. See "ADDING REMOTE STORAGE" in **guestfish** (1).

−−attach ISOFILE
The given disk is attached to the libguestfs appliance. This is used to provide extra software repositories or other data for customization.

You probably want to ensure the volume(s) or filesystems in the attached disks are labelled (or use an ISO volume name) so that you can mount them by label in your run-scripts:

```
 mkdir /tmp/mount
 mount LABEL=EXTRA /tmp/mount
```

You can have multiple *−−attach* options, and the format can be any disk format (not just an ISO).

−−attach−format FORMAT
Specify the disk format for the next *−−attach* option. The `FORMAT` is usually `raw` or `qcow2`. Use `raw` for ISOs.

−−colors
−−colours
Use ANSI colour sequences to colourize messages. This is the default when the output is a tty. If the output of the program is redirected to a file, ANSI colour sequences are disabled unless you use this option.

−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.

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

−n
−−dry−run
Perform a read-only "dry run" on the guest. This runs the sysprep operation, but throws away any changes to the disk at the end.

−−echo−keys
When prompting for keys and passphrases, virt-customize 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.

−−format raw|qcow2|..
−−format auto
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 auto* switches back to auto-detection for subsequent *−a* options.

For example:

```
 virt−customize −−format raw −a disk.img
```

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

```
 virt−customize −−format raw −a disk.img −−format auto −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).

−−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 MB
−−memsize MB
Change the amount of memory allocated to *−−run* scripts. Increase this if you find that *−−run* scripts or the *−−install* option are running out of memory.

The default can be found with this command:

```
 guestfish get−memsize
```

−−network
−−no−network
Enable or disable network access from the guest during the installation.

Enabled is the default. Use*−−no−network* to disable access.

The network only allows outgoing connections and has other minor limitations. See ''NETWORK'' in **virt−rescue** (1).

If you use *−−no−network* then certain other options such as *−−install* will not work.

This does not affect whether the guest can access the network once it has been booted, because that is controlled by your hypervisor or cloud environment and has nothing to do with virt-customize.

Generally speaking you should *not* use *−−no−network*. But here are some reasons why you might want to:

1. Because the libguestfs backend that you are using doesn't support the network. (See: ''BACKEND'' in **guestfs** (3)).
2. Any software you need to install comes from an attached ISO, so you don't need the network.
3. You don't want untrusted guest code trying to access your host network when running virt-customize. This is particularly an issue when you don't trust the source of the operating system templates. (See ''SECURITY'' below).
4. You don't have a host network (eg. in secure/restricted environments).

−q
−−quiet
Don't print log messages.

To enable detailed logging of individual file operations, use *−x*.

−−smp N
Enable N ≥ 2 virtual CPUs for *−−run* scripts to use.

−v

−−verbose
Enable verbose messages for debugging.

−V
−−version
Display version number and exit.

−x Enable tracing of libguestfs API calls.

Customization options

−−append−line FILE:LINE
Append a single line of text to the `FILE`. If the file does not already end with a newline, then one is added before the appended line. Also a newline is added to the end of the `LINE` string automatically.

For example (assuming ordinary shell quoting) this command:

```
 −−append−line '/etc/hosts:10.0.0.1 foo'
```

will add either `10.0.0.1 foo` or `10.0.0.1 foo` to the file, the latter only if the existing file does not already end with a newline.

 represents a newline character, which is guessed by looking at the existing content of the file, so this command does the right thing for files using Unix or Windows line endings. It also works for empty or non-existent files.

To insert several lines, use the same option several times:

```
 −−append−line '/etc/hosts:10.0.0.1 foo'
 −−append−line '/etc/hosts:10.0.0.2 bar'
```

To insert a blank line before the appended line, do:

```
 −−append−line '/etc/hosts:'
 −−append−line '/etc/hosts:10.0.0.1 foo'
```

−−chmod PERMISSIONS:FILE
Change the permissions of `FILE` to `PERMISSIONS`.

Note: `PERMISSIONS` by default would be decimal, unless you prefix it with `0` to get octal, ie. use `0700` not `700`.

−−commands−from−file FILENAME
Read the customize commands from a file, one (and its arguments) each line.

Each line contains a single customization command and its arguments, for example:

```
 delete /some/file
 install some−package
 password some−user:password:its−new−password
```

Empty lines are ignored, and lines starting with # are comments and are ignored as well. Furthermore, arguments can be spread across multiple lines, by adding a \ (continuation character) at the of a line, for example

```
 edit /some/file:\
   s/^OPT=.*/OPT=ok/
```

The commands are handled in the same order as they are in the file, as if they were specified as *−−delete /some/file* on the command line.

−−copy SOURCE:DEST
Copy files or directories recursively inside the guest.

Wildcards cannot be used.

−−copy−in LOCALPATH:REMOTEDIR

Copy local files or directories recursively into the disk image, placing them in the directory `REMOTEDIR` (which must exist).

Wildcards cannot be used.

−−delete PATH

Delete a file from the guest. Or delete a directory (and all its contents, recursively).

You can use shell glob characters in the specified path. Be careful to escape glob characters from the host shell, if that is required. For example:

```
 virt-customize --delete '/var/log/*.log'.
```

See also: *−−upload*, *−−scrub*.

−−edit FILE:EXPR

Edit `FILE` using the Perl expression `EXPR`.

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.

See ‘‘NON-INTERACTIVE EDITING’’ in **virt−edit**(1).

−−firstboot SCRIPT

Install `SCRIPT` inside the guest, so that when the guest first boots up, the script runs (as root, late in the boot process).

The script is automatically chmod +x after installation in the guest.

The alternative version *−−firstboot−command* is the same, but it conveniently wraps the command up in a single line script for you.

You can have multiple *−−firstboot* options. They run in the same order that they appear on the command line.

Please take a look at ‘‘FIRST BOOT SCRIPTS’’ in **virt−builder**(1) for more information and caveats about the first boot scripts.

See also *−−run*.

−−firstboot−command ’CMD+ARGS’

Run command (and arguments) inside the guest when the guest first boots up (as root, late in the boot process).

You can have multiple *−−firstboot* options. They run in the same order that they appear on the command line.

Please take a look at ‘‘FIRST BOOT SCRIPTS’’ in **virt−builder**(1) for more information and caveats about the first boot scripts.

See also *−−run*.

−−firstboot−install PKG,PKG..

Install the named packages (a comma-separated list). These are installed when the guest first boots using the guest’s package manager (eg. apt, yum, etc.) and the guest’s network connection.

For an overview on the different ways to install packages, see ‘‘INSTALLING PACKAGES’’ in **virt−builder**(1).

−−hostname HOSTNAME

Set the hostname of the guest to `HOSTNAME`. You can use a dotted hostname.domainname (FQDN) if you want.

−−install PKG,PKG..

Install the named packages (a comma-separated list). These are installed during the image build using the guest's package manager (eg. apt, yum, etc.) and the host's network connection.

For an overview on the different ways to install packages, see "INSTALLING PACKAGES" in **virt−builder**(1).

See also *−−update*, *−−uninstall*.

−−link TARGET:LINK[:LINK..]

Create symbolic link(s) in the guest, starting at `LINK` and pointing at `TARGET`.

−−mkdir DIR

Create a directory in the guest.

This uses `mkdir−p` so any intermediate directories are created, and it also works if the directory already exists.

−−move SOURCE:DEST

Move files or directories inside the guest.

Wildcards cannot be used.

−−no−logfile

Scrub `builder.log` (log file from build commands) from the image after building is complete. If you don't want to reveal precisely how the image was built, use this option.

See also: "LOG FILE".

−−password USER:SELECTOR

Set the password for `USER`. (Note this option does*not* create the user account).

See "USERS AND PASSWORDS" in **virt−builder**(1) for the format of the `SELECTOR` field, and also how to set up user accounts.

−−password−crypto md5|sha256|sha512

When the virt tools change or set a password in the guest, this option sets the password encryption of that password to `md5`, `sha256` or `sha512`.

`sha256` and `sha512` require glibc ≥ 2.7 (check **crypt**(3) inside the guest).

`md5` will work with relatively old Linux guests (eg. RHEL 3), but is not secure against modern attacks.

The default is `sha512` unless libguestfs detects an old guest that didn't have support for SHA−512, in which case it will use `md5`. You can override libguestfs by specifying this option.

Note this does not change the default password encryption used by the guest when you create new user accounts inside the guest. If you want to do that, then you should use the *−−edit* option to modify `/etc/sysconfig/authconfig` (Fedora, RHEL) or `/etc/pam.d/common−password` (Debian, Ubuntu).

−−root−password SELECTOR

Set the root password.

See "USERS AND PASSWORDS" in **virt−builder**(1) for the format of the `SELECTOR` field, and also how to set up user accounts.

Note: In virt-builder, if you *don't* set *−−root−password* then the guest is given a *random* root password.

−−run SCRIPT

Run the shell script (or any program) called `SCRIPT` on the disk image. The script runs virtualized inside a small appliance, chrooted into the guest filesystem.

The script is automatically chmod +x.

If libguestfs supports it then a limited network connection is available but it only allows outgoing network connections. You can also attach data disks (eg. ISO files) as another way to provide data (eg. software packages) to the script without needing a network connection (*−−attach*). You can also upload data files (*−−upload*).

You can have multiple *−−run* options. They run in the same order that they appear on the command line.

See also: *−−firstboot*, *−−attach*, *−−upload*.

−−run−command 'CMD+ARGS'
Run the command and arguments on the disk image. The command runs virtualized inside a small appliance, chrooted into the guest filesystem.

If libguestfs supports it then a limited network connection is available but it only allows outgoing network connections. You can also attach data disks (eg. ISO files) as another way to provide data (eg. software packages) to the script without needing a network connection (*−−attach*). You can also upload data files (*−−upload*).

You can have multiple *−−run−command* options. They run in the same order that they appear on the command line.

See also: *−−firstboot*, *−−attach*, *−−upload*.

−−scrub FILE
Scrub a file from the guest. This is like *−−delete* except that:

- It scrubs the data so a guest could not recover it.
- It cannot delete directories, only regular files.

−−selinux−relabel
Relabel files in the guest so that they have the correct SELinux label.

This will attempt to relabel files immediately, but if the operation fails this will instead touch */.autorelabel* on the image to schedule a relabel operation for the next time the image boots.

You should only use this option for guests which support SELinux.

−−sm−attach SELECTOR
Attach to a pool using `subscription−manager`.

See "SUBSCRIPTION-MANAGER" in **virt−builder** (1) for the format of the `SELECTOR` field.

−−sm−credentials SELECTOR
Set the credentials for `subscription−manager`.

See "SUBSCRIPTION-MANAGER" in **virt−builder** (1) for the format of the `SELECTOR` field.

−−sm−register
Register the guest using `subscription−manager`.

This requires credentials being set using *−−sm−credentials*.

−−sm−remove
Remove all the subscriptions from the guest using `subscription−manager`.

−−sm−unregister
Unregister the guest using `subscription−manager`.

−−ssh−inject USER[:SELECTOR]
Inject an ssh key so the given `USER` will be able to log in over ssh without supplying a password. The `USER` must exist already in the guest.

See "SSH KEYS" in **virt−builder** (1) for the format of the `SELECTOR` field.

You can have multiple *−−ssh−inject* options, for different users and also for more keys for each user.

−−timezone TIMEZONE
Set the default timezone of the guest to `TIMEZONE`. Use a location string like `Europe/London`

−−touch FILE
This command performs a **touch** (1)−like operation on `FILE`.

−−truncate FILE
This command truncates `FILE` to a zero-length file. The file must exist already.

−−truncate−recursive PATH
This command recursively truncates all files under `PATH` to zero-length.

−−uninstall PKG,PKG..
Uninstall the named packages (a comma-separated list). These are removed during the image build using the guest's package manager (eg. apt, yum, etc.). Dependent packages may also need to be uninstalled to satisfy the request.

See also *−−install*, *−−update*.

−−update
Do the equivalent of `yum update`, `apt−get upgrade`, or whatever command is required to update the packages already installed in the template to their latest versions.

See also *−−install*, *−−uninstall*.

−−upload FILE:DEST
Upload local file `FILE` to destination `DEST` in the disk image. File owner and permissions from the original are preserved, so you should set them to what you want them to be in the disk image.

`DEST` could be the final filename. This can be used to rename the file on upload.

If `DEST` is a directory name (which must already exist in the guest) then the file is uploaded into that directory, and it keeps the same name as on the local filesystem.

See also: *−−mkdir*, *−−delete*, *−−scrub*.

−−write FILE:CONTENT
Write `CONTENT` to `FILE`.

SELINUX

For guests which make use of SELinux, special handling for them might be needed when using operations which create new files or alter existing ones.

For further details, see ''SELINUX'' in **virt−builder** (1).

EXIT STATUS

This program returns 0 on success, or 1 if there was an error.

ENVIRONMENT VARIABLES

`VIRT_TOOLS_DATA_DIR`
This can point to the directory containing data files used for Windows firstboot installation.

Normally you do not need to set this. If not set, a compiled-in default will be used (something like */usr/share/virt−tools*).

This directory may contain the following files:

rhsrvany.exe
This is the RHSrvAny Windows binary, used to install a ''firstboot'' script in Windows guests. It is required if you intend to use the *−−firstboot* or *−−firstboot−command* options with Windows guests.

See also: `https://github.com/rwmjones/rhsrvany`

pvvxsvc.exe
: This is a Windows binary shipped with SUSE VMDP, used to install a ''firstboot'' script in Windows guests. It is required if you intend to use the −−*firstboot* or −−*firstboot−command* options with Windows guests.

For other environment variables, see ''ENVIRONMENT VARIABLES'' in **guestfs** (3).

SEE ALSO

guestfs (3), **guestfish** (1), **virt−builder** (1), **virt−clone** (1), **virt−rescue** (1), **virt−resize** (1), **virt−sparsify** (1), **virt−sysprep** (1), **virsh** (1), **lvcreate** (8), **qemu−img** (1), **scrub** (1), http://libguestfs.org/, http://libvirt.org/.

AUTHORS

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.