NAME

virt−sysprep − Reset, unconfigure or customize a virtual machine so clones can be made

SYNOPSIS

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

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

WARNING

Using `virt−sysprep` 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-sysprep can reset or unconfigure a virtual machine so that clones can be made from it. Steps in this process include removing SSH host keys, removing persistent network MAC configuration, and removing user accounts. Virt-sysprep can also customize a virtual machine, for instance by adding SSH keys, users or logos. Each step can be enabled or disabled as required.

Virt-sysprep 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*. See ''COPYING AND CLONING'' below.

You do *not* need to run virt-sysprep as root. In fact we'd generally recommend that you don't. The time you might want to run it as root is when you need root in order to access the disk image, but even in this case it would be better to change the permissions on the disk image to be writable as the non-root user running virt-sysprep.

''Sysprep'' stands for ''system preparation'' tool. The name comes from the Microsoft program *sysprep.exe* which is used to unconfigure Windows machines in preparation for cloning them. Having said that, virt-sysprep does *not* currently work on Microsoft Windows guests. We plan to support Windows syspreping in a future version, and we already have code to do it.

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).

−−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.

−−enable operations
Choose which sysprep operations to perform. Give a comma-separated list of operations, for example:

```
 --enable ssh-hostkeys,udev-persistent-net
```

would enable ONLY `ssh−hostkeys` and `udev−persistent−net` operations.

If the *−−enable* option is not given, then we default to trying most sysprep operations (see *−−list−operations* to show which are enabled).

Regardless of the *−−enable* option, sysprep operations are skipped for some guest types.

Use *−−list−operations* to list operations supported by a particular version of virt-sysprep.

See "OPERATIONS" below for a list and an explanation of each operation.

−−operation operations
−−operations operations
Choose which sysprep operations to perform. Give a comma-separated list of operations, for example:

```
 --operations ssh-hostkeys,udev-persistent-net
```

would enable ONLY `ssh−hostkeys` and `udev−persistent−net` operations.

−−operations allows you to enable and disable any operation, including the default ones (which would be tried when specifying neither *−−operations* nor *−−enable*) and all the available ones; prepending a `−` in front of an operation name removes it from the list of enabled operations, while the meta-names `defaults` and `all` represent respectively the operations enabled by default and all the available ones. For example:

```
 --operations firewall-rules,defaults,-tmp-files
```

would enable the `firewall−rules` operation (regardless whether it is enabled by default), all the default ones, and disable the `tmp−files` operation.

−−operations can be specified multiple times; the first time the set of enabled operations is empty, while any further *−−operations* affects the operations enabled so far.

If the *−−operations* option is not given, then we default to trying most sysprep operations (see *−−list−operations* to show which are enabled).

Regardless of the *−−operations* option, sysprep operations are skipped for some guest types.

Use *−−list−operations* to list operations supported by a particular version of virt-sysprep.

See "OPERATIONS" below for a list and an explanation of each operation.

−−echo−keys
When prompting for keys and passphrases, virt-sysprep 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-sysprep --format raw -a disk.img
```

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

```
virt-sysprep --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.

−−list−operations

List the operations supported by the virt-sysprep program.

These are listed one per line, with one or more single-space-separated fields, eg:

```
$ virt-sysprep --list-operations
bash-history * Remove the bash history in the guest
cron-spool * Remove user at-jobs and cron-jobs
dhcp-client-state * Remove DHCP client leases
dhcp-server-state * Remove DHCP server leases
[etc]
```

The first field is the operation name, which can be supplied to *−−enable*. The second field is a `*` character if the operation is enabled by default or blank if not. Subsequent fields on the same line are the description of the operation.

Before libguestfs 1.17.33 only the first (operation name) field was shown and all operations were enabled by default.

−−mount−options mp:opts[;mp:opts;...]

Set the mount options used when libguestfs opens the disk image. Note this has no effect on the guest. It is used when opening certain guests such as ones using the UFS (BSD) filesystem.

Use a semicolon-separated list of `mountpoint:options` pairs. You may need to quote this list to protect it from the shell.

For example:

```
--mount-options "/:noatime"
```

will mount the root directory with `notime`. This example:

```
--mount-options "/:noatime;/var:rw,nodiratime"
```

will do the same, plus mount */var* with `rw,nodiratime`.

−q
−−quiet
Don't print log messages.

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

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

In virt-sysprep, the network is *disabled* by default. You must use *−−network* to enable it, in order that options such as *−−install* or *−−update* will work.

virt−builder (1) has more information about the security advantages of disabling the network.

−v
−−verbose
Enable verbose messages for debugging.

−V
−−version
Display version number and exit.

−x Enable tracing of libguestfs API calls.

−−append−line FILE:LINE (see `customize` below)
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 (see `customize` below)
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 (see `customize` below)
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 (see `customize` below)
Copy files or directories recursively inside the guest.

Wildcards cannot be used.

--copy-in LOCALPATH:REMOTEDIR (see `customize` below)
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 (see `customize` below)
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 (see `customize` below)
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 (see `customize` below)
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' (see `customize` below)
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.. (see `customize` below)
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 (see `customize` below)
Set the hostname of the guest to `HOSTNAME`. You can use a dotted hostname.domainname (FQDN) if you want.

−−install PKG,PKG.. (see `customize` below)
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*.

−−keep−user−accounts USERS (see `user−account` below)
The user accounts to be kept in the guest. The value of this option is a list of user names separated by comma, where specifying an user means it is going to be kept. For example:

```
 −−keep−user−accounts mary
```

would keep the user account `mary`.

This option can be specified multiple times.

−−link TARGET:LINK[:LINK..] (see `customize` below)
Create symbolic link(s) in the guest, starting at `LINK` and pointing at `TARGET`.

−−mkdir DIR (see `customize` below)
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 (see `customize` below)
Move files or directories inside the guest.

Wildcards cannot be used.

−−no−logfile (see `customize` below)
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 (see `customize` below)
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 (see `customize` below)
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).

−−remove−user−accounts USERS (see `user−account` below)
The user accounts to be removed from the guest. The value of this option is a list of user names separated by comma, where specifying an user means it is going to be removed. For example:

```
 −−remove−user−accounts bob,eve
```

would only remove the user accounts `bob` and `eve`.

This option can be specified multiple times.

−−root−password SELECTOR (see `customize` below)
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 (see `customize` below)
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' (see `customize` below)
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*.

−−script SCRIPT (see `script` below)
Run the named `SCRIPT` (a shell script or program) against the guest. The script can be any program on the host. The script's current directory will be the guest's root directory.

Note: If the script is not on the `$PATH`, then you must give the full absolute path to the script.

−−scriptdir SCRIPTDIR (see `script` below)
The mount point (an empty directory on the host) used when the `script` operation is enabled and one or more scripts are specified using *−−script* parameter(s).

Note: `SCRIPTDIR` **must** be an absolute path.

If *--scriptdir* is not specified then a temporary mountpoint will be created.

--scrub FILE (see `customize` below)
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 (see `customize` below)
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 (see `customize` below)
Attach to a pool using `subscription-manager`.

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

--sm-credentials SELECTOR (see `customize` below)
Set the credentials for `subscription-manager`.

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

--sm-register (see `customize` below)
Register the guest using `subscription-manager`.

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

--sm-remove (see `customize` below)
Remove all the subscriptions from the guest using `subscription-manager`.

--sm-unregister (see `customize` below)
Unregister the guest using `subscription-manager`.

--ssh-inject USER[:SELECTOR] (see `customize` below)
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 (see `customize` below)
Set the default timezone of the guest to `TIMEZONE`. Use a location string like `Europe/London`

--touch FILE (see `customize` below)
This command performs a **touch**(1)-like operation on `FILE`.

--truncate FILE (see `customize` below)
This command truncates `FILE` to a zero-length file. The file must exist already.

--truncate-recursive PATH (see `customize` below)
This command recursively truncates all files under `PATH` to zero-length.

--uninstall PKG,PKG.. (see `customize` below)
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 (see `customize` below)
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 (see `customize` below)
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 (see `customize` below)
Write `CONTENT` to `FILE`.

OPERATIONS

If the *--enable*/*--operations* option is *not* given, then most sysprep operations are enabled.

Use `virt-sysprep --list-operations` to list all operations for your virt-sysprep binary. The ones which are enabled by default are marked with a `*` character. Regardless of the *--enable*/*--operations* options, sysprep operations are skipped for some guest types.

Operations can be individually enabled using the *--enable*/*--operations* options. Use a comma-separated list, for example:

```
 virt-sysprep --operations ssh-hostkeys,udev-persistent-net [etc..]
```

Future versions of virt-sysprep may add more operations. If you are using virt-sysprep and want predictable behaviour, specify only the operations that you want to have enabled.

`*` = enabled by default when no *--enable*/*--operations* option is given.

abrt-data *
Remove the crash data generated by ABRT.

Remove the automatically generated ABRT crash data in `/var/spool/abrt/`.

backup-files *
Remove editor backup files from the guest.

The following files are removed from anywhere in the guest filesystem:

- *.bak
- *~

On Linux and Unix operating systems, only the following filesystems will be examined:

- /etc
- /root
- /srv
- /tmp
- /var

bash-history *
Remove the bash history in the guest.

Remove the bash history of user ‘‘root’’ and any other users who have a `.bash_history` file in their home directory.

Notes on bash-history

Currently this only looks in `/root` and `/home/*` for home directories, so users with home directories in other locations won’t have the bash history removed.

blkid-tab *
Remove blkid tab in the guest.

ca-certificates
Remove CA certificates in the guest.

In case any certificate is removed, the system CA store is updated.

crash-data *
Remove the crash data generated by kexec-tools.

Remove the automatically generated kdump kernel crash data.

cron-spool *
Remove user at-jobs and cron-jobs.

customize *
Customize the guest.

Customize the guest by providing **virt−customize**(1) options for installing packages, editing files and so on.

dhcp-client-state *
Remove DHCP client leases.

dhcp-server-state *
Remove DHCP server leases.

dovecot-data *
Remove Dovecot (mail server) data.

firewall-rules
Remove the firewall rules.

This removes custom firewall rules by removing `/etc/sysconfig/iptables` or custom firewalld configuration in `/etc/firewalld/*/*`.

Note this is *not* enabled by default since it may expose guests to exploits. Use with care.

flag-reconfiguration
Flag the system for reconfiguration.

For Linux guests, this touches `/.unconfigured`, which causes the first boot to interactively query the user for settings such as the root password and timezone.

fs-uuids
Change filesystem UUIDs.

On guests and filesystem types where this is supported, new random UUIDs are generated and assigned to filesystems.

Notes on fs-uuids

The fs-uuids operation is disabled by default because it does not yet find and update all the places in the guest that use the UUIDs. For example `/etc/fstab` or the bootloader. Enabling this operation is more likely than not to make your guest unbootable.

See: https://bugzilla.redhat.com/show_bug.cgi?id=991641

ipa-client *
Remove the IPA files.

Remove all the files related to an IPA (Identity, Policy, Audit) system. This effectively unenrolls the guest from an IPA server without interacting with it.

This operation does not run `ipa−client`.

kerberos-data
Remove Kerberos data in the guest.

kerberos-hostkeytab *
Remove the Kerberos host keytab file in the guest.

logfiles *
Remove many log files from the guest.

On Linux the following files are removed:

- /etc/Pegasus/*.cnf
- /etc/Pegasus/*.crt
- /etc/Pegasus/*.csr
- /etc/Pegasus/*.pem
- /etc/Pegasus/*.srl
- /root/anaconda−ks.cfg
- /root/anaconda−post.log
- /root/initial−setup−ks.cfg
- /root/install.log
- /root/install.log.syslog
- /root/original−ks.cfg
- /var/cache/fontconfig/*
- /var/cache/gdm/*
- /var/cache/man/*
- /var/lib/AccountService/users/*
- /var/lib/fprint/*
- /var/lib/logrotate.status
- /var/log/*.log*
- /var/log/BackupPC/LOG
- /var/log/ConsoleKit/*
- /var/log/anaconda.syslog
- /var/log/anaconda/*
- /var/log/apache2/*_log
- /var/log/apache2/*_log−*
- /var/log/apt/*
- /var/log/aptitude*
- /var/log/audit/*
- /var/log/btmp*
- /var/log/ceph/*.log
- /var/log/chrony/*.log
- /var/log/cron*
- /var/log/cups/*_log*

- /var/log/debug*
- /var/log/dmesg*
- /var/log/exim4/*
- /var/log/faillog*
- /var/log/firewalld*
- /var/log/gdm/*
- /var/log/glusterfs/*glusterd.vol.log
- /var/log/glusterfs/glusterfs.log
- /var/log/grubby*
- /var/log/httpd/*log
- /var/log/installer/*
- /var/log/jetty/jetty−console.log
- /var/log/journal/*
- /var/log/lastlog*
- /var/log/libvirt/libvirtd.log
- /var/log/libvirt/libxl/*.log
- /var/log/libvirt/lxc/*.log
- /var/log/libvirt/qemu/*.log
- /var/log/libvirt/uml/*.log
- /var/log/lightdm/*
- /var/log/mail/*
- /var/log/maillog*
- /var/log/messages*
- /var/log/ntp
- /var/log/ntpstats/*
- /var/log/ppp/connect−errors
- /var/log/rhsm/*
- /var/log/sa/*
- /var/log/secure*
- /var/log/setroubleshoot/*.log
- /var/log/spooler*
- /var/log/squid/*.log
- /var/log/syslog*
- /var/log/tallylog*
- /var/log/tuned/tuned.log
- /var/log/wtmp*
- /var/log/xferlog*
- /var/named/data/named.run

lvm-uuids *
Change LVM2 PV and VG UUIDs.

On Linux guests that have LVM2 physical volumes (PVs) or volume groups (VGs), new random UUIDs are generated and assigned to those PVs and VGs.

machine-id *
Remove the local machine ID.

The machine ID is usually generated from a random source during system installation and stays constant for all subsequent boots. Optionally, for stateless systems it is generated during runtime at boot if it is found to be empty.

mail-spool *
Remove email from the local mail spool directory.

net-hostname *
Remove HOSTNAME and DHCP_HOSTNAME in network interface configuration.

For Fedora and Red Hat Enterprise Linux, this is removed from `ifcfg-*` files.

net-hwaddr *
Remove HWADDR (hard-coded MAC address) configuration.

For Fedora and Red Hat Enterprise Linux, this is removed from `ifcfg-*` files.

pacct-log *
Remove the process accounting log files.

The system wide process accounting will store to the pacct log files if the process accounting is on.

package-manager-cache *
Remove package manager cache.

pam-data *
Remove the PAM data in the guest.

passwd-backups *
Remove /etc/passwd− and similar backup files.

On Linux the following files are removed:

· /etc/group−

· /etc/gshadow−

· /etc/passwd−

· /etc/shadow−

· /etc/subgid−

· /etc/subuid−

puppet-data-log *
Remove the data and log files of puppet.

rh-subscription-manager *
Remove the RH subscription manager files.

rhn-systemid *
Remove the RHN system ID.

rpm-db *
Remove host-specific RPM database files.

Remove host-specific RPM database files and locks. RPM will recreate these files automatically if needed.

samba-db-log *

Remove the database and log files of Samba.

script *

Run arbitrary scripts against the guest.

The `script` module lets you run arbitrary shell scripts or programs against the guest.

Note this feature requires FUSE support. You may have to enable this in your host, for example by adding the current user to the `fuse` group, or by loading a kernel module.

Use one or more −−*script* parameters to specify scripts or programs that will be run against the guest.

The script or program is run with its current directory being the guest's root directory, so relative paths should be used. For example: `rm etc/resolv.conf` in the script would remove a Linux guest's DNS configuration file, but `rm /etc/resolv.conf` would (try to) remove the host's file.

Normally a temporary mount point for the guest is used, but you can choose a specific one by using the −−*scriptdir* parameter.

Note: This is different from −−*firstboot* scripts (which run in the context of the guest when it is booting first time). −−*script* scripts run on the host, not in the guest.

smolt-uuid *

Remove the Smolt hardware UUID.

ssh-hostkeys *

Remove the SSH host keys in the guest.

The SSH host keys are regenerated (differently) next time the guest is booted.

If, after cloning, the guest gets the same IP address, ssh will give you a stark warning about the host key changing:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
```

ssh-userdir *

Remove ".ssh" directories in the guest.

Remove the `.ssh` directory of user "root" and any other users who have a `.ssh` directory in their home directory.

Notes on ssh-userdir

Currently this only looks in `/root` and `/home/*` for home directories, so users with home directories in other locations won't have the ssh files removed.

sssd-db-log *

Remove the database and log files of sssd.

tmp-files *

Remove temporary files.

This removes temporary files under `/tmp` and `/var/tmp`.

udev-persistent-net *

Remove udev persistent net rules.

Remove udev persistent net rules which map the guest's existing MAC address to a fixed ethernet device (eg. eth0).

After a guest is cloned, the MAC address usually changes. Since the old MAC address occupies the old name (eg. eth0), this means the fresh MAC address is assigned to a new name (eg. eth1) and this is usually undesirable. Erasing the udev persistent net rules avoids this.

user-account

Remove the user accounts in the guest.

By default remove all the user accounts and their home directories. The ‘‘root’’ account is not removed.

See the *−−remove−user−accounts* parameter for a way to specify how to remove only some users, or to not remove some others.

utmp *

Remove the utmp file.

This file records who is currently logged in on a machine. In modern Linux distros it is stored in a ramdisk and hence not part of the virtual machine’s disk, but it was stored on disk in older distros.

yum-uuid *

Remove the yum UUID.

Yum creates a fresh UUID the next time it runs when it notices that the original UUID has been erased.

COPYING AND CLONING

Virt-sysprep can be used as part of a process of cloning guests, or to prepare a template from which guests can be cloned. There are many different ways to achieve this using the virt tools, and this section is just an introduction.

A virtual machine (when switched off) consists of two parts:

configuration

The configuration or description of the guest. eg. The libvirt XML (see `virsh dumpxml`), the running configuration of the guest, or another external format like OVF.

Some configuration items that might need to be changed:

- name
- UUID
- path to block device(s)
- network card MAC address

block device(s)

One or more hard disk images, themselves containing files, directories, applications, kernels, configuration, etc.

Some things inside the block devices that might need to be changed:

- hostname and other net configuration
- UUID
- SSH host keys
- Windows unique security ID (SID)
- Puppet registration

COPYING THE BLOCK DEVICE

Starting with an original guest, you probably wish to copy the guest block device and its configuration to make a template. Then once you are happy with the template, you will want to make many clones from it.

```
                       virt-sysprep
                            |
                            v
original guest --------> template ---------->
                                     \------> cloned
                                      \-----> guests
                                       \---->
```

You can, of course, just copy the block device on the host using **cp** (1) or **dd** (1).

```
                  dd                 dd
original guest --------> template ---------->
                                     \------> cloned
                                      \-----> guests
                                       \---->
```

There are some smarter (and faster) ways too:

```
                         snapshot
               template ---------->
                           \------> cloned
                            \-----> guests
                             \---->
```

You may want to run virt-sysprep twice, once to reset the guest (to make a template) and a second time to customize the guest for a specific user:

```
                   virt-sysprep        virt-sysprep
                     (reset)      (add user, keys, logos)
                        |                    |
                dd      v          dd        v
original guest ----> template ---------> copied ------> custom
                                         template       guest
```

- Create a snapshot using qemu-img:

  ```
  qemu-img create -f qcow2 -o backing_file=original snapshot.qcow
  ```

  The advantage is that you don't need to copy the original (very fast) and only changes are stored (less storage required).

  Note that writing to the backing file once you have created guests on top of it is not possible: you will corrupt the guests.

- Create a snapshot using `lvcreate --snapshot`.
- Other ways to create snapshots include using filesystems-level tools (for filesystems such as btrfs).

  Most Network Attached Storage (NAS) devices can also create cheap snapshots from files or LUNs.
- Get your NAS to duplicate the LUN. Most NAS devices can also duplicate LUNs very cheaply (they copy them on-demand in the background).
- Prepare your template using **virt-sparsify** (1). See below.

VIRT-CLONE

A separate tool, **virt-clone** (1), can be used to duplicate the block device and/or modify the external libvirt configuration of a guest. It will reset the name, UUID and MAC address of the guest in the libvirt XML.

virt-clone (1) does not use libguestfs and cannot look inside the disk image. This was the original motivation to write virt-sysprep.

SPARSIFY

```
             virt-sparsify
original guest --------> template
```

virt-sparsify (1) can be used to make the cloning template smaller, making it easier to compress and/or faster to copy.

Notice that since virt-sparsify also copies the image, you can use it to make the initial copy (instead of `dd`).

RESIZE

```
                  virt-resize
         template ---------->
                     \------> cloned
                      \-----> guests
                       \---->
```

If you want to give people cloned guests, but let them pick the size of the guest themselves (eg. depending on how much they are prepared to pay for disk space), then instead of copying the template, you can run **virt−resize** (1). Virt-resize performs a copy and resize, and thus is ideal for cloning guests from a template.

FIRSTBOOT VS SCRIPT

The two options *−−firstboot* and *−−script* both supply shell scripts that are run against the guest. However these two options are significantly different.

−−firstboot script uploads the file `script` into the guest and arranges that it will run, in the guest, when the guest is next booted. (The script will only run once, at the ''first boot'').

−−script script runs the shell `script` *on the host*, with its current directory inside the guest filesystem.

If you needed, for example, to `yum install` new packages, then you *must not* use *−−script* for this, since that would (a) run the `yum` command on the host and (b) wouldn't have access to the same resources (repositories, keys, etc.) as the guest. Any command that needs to run on the guest *must* be run via *−−firstboot*.

On the other hand if you need to make adjustments to the guest filesystem (eg. copying in files), then *−−script* is ideal since (a) it has access to the host filesystem and (b) you will get immediate feedback on errors.

Either or both options can be used multiple times on the command line.

SECURITY

Although virt-sysprep removes some sensitive information from the guest, it does not pretend to remove all of it. You should examine the ''OPERATIONS'' above and the guest afterwards.

Sensitive files are simply removed. The data they contained may still exist on the disk, easily recovered with a hex editor or undelete tool. The *−−scrub* option can be used to scrub files instead of just deleting them. **virt−sparsify** (1) is another way to remove this content. See also the **scrub** (1) command to get rid of deleted content in directory entries and inodes.

RANDOM SEED

(This section applies to Linux guests only)

For supported guests, virt-sysprep writes a few bytes of randomness from the host into the guest's random seed file.

If this is just done once and the guest is cloned from the same template, then each guest will start with the same entropy, and things like SSH host keys and TCP sequence numbers may be predictable.

Therefore you should arrange to add more randomness *after* cloning from a template too, which can be done by enabling just the customize module:

```
cp template.img newguest.img
virt−sysprep −−enable customize −a newguest.img
```

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).

WINDOWS 8

Windows 8 ''fast startup'' can prevent virt-sysprep from working. See ''WINDOWS HIBERNATION AND WINDOWS 8 FAST STARTUP'' in **guestfs** (3).

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−customize** (1), **virt−rescue** (1), **virt−resize** (1), **virt−sparsify** (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/

Wanlong Gao, Fujitsu Ltd.

COPYRIGHT

Copyright (C) 2011−2020 Red Hat Inc.

Copyright (C) 2012 Fujitsu Ltd.

LICENSE

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110−1301 USA.

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.