NAME

virt−builder−repository − Build virt−builder source repository easily

SYNOPSIS

```
virt−builder−repository /path/to/repository
   [−i|−−interactive] [−−gpg−key KEYID]
```

DESCRIPTION

Virt-builder is a tool for quickly building new virtual machines. It can be configured to use template repositories. However creating and maintaining a repository involves many tasks which can be automated. virt-builder-repository is a tool helping to manage these repositories.

Virt-builder-repository loops over the files in the directory specified as argument, compresses the files with a name ending by `qcow2`, `raw`, `img` or without extension, extracts data from them and creates or updates the `index` file.

Some of the image-related data needed for the index file can't be computed from the image file. virt-builder-repository first tries to find them in the existing index file. If data are still missing after this, they are prompted in interactive mode, otherwise an error will be triggered.

If a `KEYID` is provided, the generated index file will be signed with this GPG key.

EXAMPLES

Create the initial repository

Create a folder and copy the disk image template files in it. Then run a command like the following one:

```
virt−builder−repository −−gpg−key "joe@hacker.org" −i /path/to/folder
```

Note that this example command runs in interactive mode. To run in automated mode, a minimal index file needs to be created before running the command containing sections like this one:

```
[template_id]
file=template_filename.qcow.xz
```

The file value needs to match the image name extended with the `.xz` suffix if the *−−no−compression* parameter is not provided or the image name if no compression is involved. Other optional data can be prefilled. Default values are computed by inspecting the disk image. For more informations, see ''Creating and signing the index file'' in **virt−builder** (1).

Update images in an existing repository

In this use case, a new image or a new revision of an existing image needs to be added to the repository. Place the corresponding image template files in the repository folder.

To update the revision of an image, the file needs to have the same name than the existing one (without the `xz` extension).

As in the repository creation use case, a minimal fragment can be added to the index file for the automated mode. This can be done on the signed index even if it may sound a strange idea: the index will be signed again by the tool.

To remove an image from the repository, just remove the corresponding image file before running virt-builder-repository.

Then running the following command will complete and update the index file:

```
virt−builder−repository −−gpg−key "joe@hacker.org" −i /path/to/folder
```

virt-builder-repository works in a temporary folder inside the repository one. If anything wrong happens when running the tool, the repository is left untouched.

OPTIONS

−−help

Display help.

--gpg GPG
Specify an alternate **gpg** (1) (GNU Privacy Guard) binary. You can also use this to add gpg parameters, for example to specify an alternate home directory:

```
virt-builder-repository --gpg "gpg --homedir /tmp" [...]
```

This can also be used to avoid gpg asking for the key passphrase:

```
virt-builder-repository --gpg "gpg --passphrase-file /tmp/pass --batch" [...]
```

-K KEYID
--gpg-key KEYID
Specify the GPG key to be used to sign the repository index file. If not provided, the index will left unsigned. `KEYID` is used to identify the GPG key to use. This value is passed to gpg's *--default-key* option and can thus be an email address or a fingerprint.

NOTE: by default, virt-builder-repository searches for the key in the user's GPG keyring.

-i
--interactive
Prompt for missing data. Default values are computed from the disk image.

When prompted for data, inputting **-** corresponds to leaving the value empty. This can be used to avoid setting the default computed value.

--no-compression
Don't compress the template images.

--machine-readable
--machine-readable=format
This option is used to make the output more machine friendly when being parsed by other programs. See ''MACHINE READABLE OUTPUT'' below.

--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.

-q
--quiet
Don't print ordinary progress messages.

-v
--verbose
Enable debug messages and/or produce verbose output.

When reporting bugs, use this option and attach the complete output to your bug report.

-V
--version
Display version number and exit.

-x Enable tracing of libguestfs API calls.

MACHINE READABLE OUTPUT

The *--machine-readable* option can be used to make the output more machine friendly, which is useful when calling virt-builder-repository from other programs, GUIs etc.

Use the option on its own to query the capabilities of the virt-builder-repository binary. Typical output looks like this:

```
$ virt-builder-repository --machine-readable
virt-builder-repository
```

A list of features is printed, one per line, and the program exits with status 0.

It is possible to specify a format string for controlling the output; see ‘‘ADVANCED MACHINE READABLE OUTPUT’’ in **guestfs** (3).

EXIT STATUS

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

SEE ALSO

virt−builder (1) http://libguestfs.org/.

AUTHOR

Cédric Bosdonnat mailto:cbosdonnat@suse.com

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.