

NAME

virt-p2v-make-kickstart – Build the virt-p2v kickstart

SYNOPSIS

```
virt-p2v-make-kickstart [-o p2v.ks] [--proxy=http://...] repo [repo...]
```

DESCRIPTION

virt-p2v (1) converts a physical machine to run virtualized on KVM, managed by libvirt, OpenStack, oVirt, Red Hat Enterprise Virtualisation (RHEV), or one of the other targets supported by **virt-v2v** (1).

Kickstart is a format used by Red Hat-derived distributions (such as Fedora, Red Hat Enterprise Linux, CentOS, Scientific Linux, and others) to describe how to make live CDs, install the distro, make “Spins” and so on. It is driven by a kickstart file.

virt-p2v-make-kickstart builds a kickstart file which can be used to build a bootable P2V ISO, live CD, USB key, or PXE image. This tool only builds the kickstart file, but this manual page describes some of the ways you can use the kickstart file.

The root user on the disk image uses p2v as its initial password.

BUILDING THE KICKSTART FILE

Using virt-p2v-make-kickstart is very simple:

```
virt-p2v-make-kickstart fedora
```

will build a kickstart file for Fedora. The kickstart file will be called *p2v.ks* and located in the current directory.

The parameters are a list of one or more repositories. Some built-in repositories are available: *fedora*, *rawhide*, *koji* or *rhel-VERSION* (eg. *rhel-7.1*). You can also use a URL as a parameter to point to a repository, for example:

```
virt-p2v-make-kickstart https://dl.fedoraproject.org/pub/fedora/linux/releases/2
```

To control the name of the output file, use the *-o* parameter. To tell kickstart to use a proxy server or web cache to download files, use the *--proxy* parameter.

BUILDING A LIVE CD / ISO

Once you have the kickstart file, you can use **livecd-creator** (8) to make a live CD:

```
sudo livecd-creator p2v.ks
```

Before running this note that you should probably run **livecd-creator** in a disposable virtual machine for these reasons:

- You have to disable SELinux when running the tool.
- This tool has to be run as root, and has some nasty failure modes.
- You can only create the exact same Live CD distro as the host distro. Cross-builds will fail in strange ways (eg. RHBZ#1092327).

BUILDING A FEDORA SPIN USING KOJI

This requires **spin-livecd** permissions on Koji, which are not given out usually, even to Fedora packagers. However assuming you have been given these permissions (or have your own Koji instance, I guess), then you can do:

```
koji spin-livecd [--scratch] virt-p2v 1.XX.YY rawhide x86_64 p2v.ks
```

- Add the *--scratch* option to do a scratch build (recommended for testing).
- 1.XX.YY should match the libguestfs version
- Instead of *rawhide* you can use any Koji target.

BUILDING A BOOTABLE USB KEY

Use the **livecd-iso-to-disk** (8) program to convert the ISO created above to a USB key:

```
sudo livedcd-iso-to-disk livedcd-p2v.iso /dev/sdX
```

BUILDING A PXE BOOT IMAGE

Use the `livedcd-iso-to-pxeboot` program to convert the ISO created above to a PXE boot image.

```
sudo livedcd-iso-to-pxeboot livedcd-p2v.iso
```

This creates a `tftpboot` subdirectory under the current directory containing the files required to PXE boot `virt-p2v`:

```
$ ls -lR tftpboot/
tftpboot/:
initrd0.img
pxelinux.0
pxelinux.cfg/
vmlinuz0

tftpboot/pxelinux.cfg:
default
```

32 OR 64 BIT VIRT-P2V?

`Virt-p2v` can convert any 32 or 64 bit guest, regardless of whether `virt-p2v` itself is built as a 32 or 64 bit binary. The only restriction is that 64 bit `virt-p2v` cannot run on 32 bit hardware.

Old `virt-p2v` 0.9 was always built as a 32 bit (i686) ISO. This meant that the CD could be booted on any 32- or 64-bit i686 or x86-64 hardware, and could convert any guest. The old `virt-p2v` ISO shipped by Red Hat was based on Red Hat Enterprise Linux (RHEL) 6.

Since RHEL 7 dropped support for 32 bit machines, current `virt-p2v` on RHEL can only be built for 64 bit. It cannot run on old 32 bit only hardware.

Fedora `virt-p2v` ISOs are generally built for 32 bit, so like the old RHEL 6-based `virt-p2v` 0.9 they can boot on any hardware.

TESTING VIRT-P2V USING QEMU

TESTING THE P2V ISO USING QEMU

You can use `qemu` to test-boot the P2V ISO:

```
qemu-kvm -m 1024 -hda /tmp/guest.img -cdrom /tmp/livedcd-p2v.iso -boot d
```

Note that `-hda` is the (virtual) system that you want to convert (for test purposes). It could be any guest type supported by **virt-v2v** (1), including Windows or Red Hat Enterprise Linux.

TESTING PXE SUPPORT USING QEMU

- Unpack the `tftpboot` directory into `/tmp` (so it appears as `/tmp/tftpboot`).
- Copy `pxelinux.0` and `ldlinux.c32` from `syslinux` (usually from `/usr/share/syslinux`) into `/tmp/tftpboot`.
- Adjust the `APPEND` line in `/tmp/tftpboot/pxelinux.cfg/default` if required. See “KERNEL COMMAND LINE CONFIGURATION” in **virt-p2v** (1).
- Run `qemu` like this so that it acts as a TFTP and BOOTP server, emulating a netboot:

```
qemu-kvm \
-m 4096 -hda /tmp/guest.img \
-boot n \
-netdev user,id=unet,tftp=/tmp/tftpboot,bootfile=pxelinux.0 \
-device virtio-net-pci,netdev=unet \
-serial stdio
```

Note that this requires considerably more memory because the PXE image is loaded into memory. Also that `qemu`’s TFTP server is very slow and the `virt-p2v` PXE image is very large, so it can appear to “hang” after `pxelinux` starts up.

ADDING EXTRA PACKAGES

You can install extra packages using the `--install` option. This can be useful for making a more fully-featured virt-p2v disk with extra tools for debugging and troubleshooting. Give a list of packages, separated by commas. For example:

```
virt-p2v-make-kickstart [...] --install tcpdump,traceroute
```

ADDING AN SSH IDENTITY

You can inject an SSH identity (private key) file to the kickstart and hence into the ISO using the `--inject-ssh-identity` option. Note that you *cannot* inject a key once the ISO has been built.

First create a key pair. It must have an empty passphrase:

```
ssh-keygen -t rsa -N '' -f id_rsa
```

This creates a private key (`id_rsa`) and a public key (`id_rsa.pub`) pair. The public key should be appended to the `authorized_keys` file on the virt-v2v conversion server (usually to `/root/.ssh/authorized_keys`).

The private key should be added to the kickstart file and then discarded:

```
virt-p2v-make-kickstart [...] --inject-ssh-identity id_rsa
rm id_rsa
```

The ISO can then be built from the kickstart in the usual way (see above), and it will contain the embedded SSH identity (`/var/tmp/id_rsa`).

When booting virt-p2v, specify the URL of the injected file like this:

```
|           User name: [root_____] |
|           Password: [  <leave this field blank>  ] |
| SSH Identity URL: [file:///var/tmp/id_rsa_____] |
```

or if using the kernel command line, add:

```
p2v.identity=file:///var/tmp/id_rsa
```

For more information, see “SSH IDENTITIES” in **virt-p2v**(1).

OPTIONS

--help

Display help.

--inject-ssh-identity id_rsa

Add an SSH identity (private key) file into the kickstart. See “ADDING AN SSH IDENTITY” above.

--install pkg, pkg,...

Add extra packages to the kickstart `%packages` section. See “ADDING EXTRA PACKAGES” above.

-o OUTPUT

--output OUTPUT

Write kickstart to OUTPUT. If not specified, the default is `p2v.ks` in the current directory.

--proxy URL

Tell the kickstart to use a proxy server or web cache for downloads.

-v

--verbose

Enable verbose output. Use this if you need to debug problems with the script or if you are filing a bug.

-V

--version

Display version number and exit.

FILES

\$libdir/virt-p2v/virt-p2v.xz

The **virt-p2v** (1) binary which is copied into the kickstart file.

The location of the binary can be changed by setting the `VIRT_P2V_DATA_DIR` environment variable.

\$datadir/virt-p2v/issue

\$datadir/virt-p2v/launch-virt-p2v.in

\$datadir/virt-p2v/p2v.ks.in

\$datadir/virt-p2v/p2v.service

Various data files that are used to make the kickstart.

The location of these files can be changed by setting the `VIRT_P2V_DATA_DIR` environment variable.

ENVIRONMENT VARIABLES

`VIRT_P2V_DATA_DIR`

The directory where `virt-p2v-make-kickstart` looks for data files and the `virt-p2v` binary (see “FILES” above). If not set, a compiled-in location is used.

SEE ALSO

virt-p2v (1), **virt-p2v-make-disk** (1), **virt-v2v** (1), **livecd-creator** (8), **livecd-iso-to-disk** (8), <http://libguestfs.org/>.

AUTHORS

Richard W.M. Jones <http://people.redhat.com/~rjones/>

COPYRIGHT

Copyright (C) 2009–2019 Red Hat Inc.

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, see [<https://www.gnu.org/licenses/>](https://www.gnu.org/licenses/).

BUGS

To get a list of bugs against libguestfs (which include `virt-p2v`), 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 `virt-p2v`.
- Where you got `virt-p2v` (eg. which Linux distro, compiled from source, etc)
- Describe the bug accurately and give a way to reproduce it.