NAME

p2v−building − How to build virt−p2v from source

DESCRIPTION

This manual page describes how to build virt−p2v from source.

The main steps are:

- Install the requirements.
- Build, either from the git repository or from a tarball.
- Run the tests.
- Run the tools from the source directory, or install.

REQUIREMENTS

Full list of requirements

qemu-img ≥ 2.2.0
: *Required*.

Gcc or Clang
: *Required*. We use `__attribute__((cleanup))` which is a GCC extension also supported by Clang.

Perl
: *Required*. Various build steps and tests are written in Perl. Perl is not needed at runtime.

Perl `Pod::Man`
Perl `Pod::Simple`
: *Required*. Part of Perl core.

autoconf
automake
: *Required* if compiling from git. Optional if compiling from tarball.

Perl-compatible Regular Expressions (PCRE) library
: *Required*.

libxml2
: *Required*.

xz *Required*.

qemu-nbd
nbdkit
: Optional. qemu-nbd is used for testing.

  virt−p2v (1) requires either qemu-nbd or nbdkit, but these only need to be present on the virt−p2v ISO, they do not need to be installed at compile time.

Gtk ≥ 2.24, or 3
: *Required*.

  Either Gtk 2 or Gtk 3 can be used. If you want to select a specific version of Gtk, use `./configure--with-gtk=2` or `./configure--with-gtk=3`.

D−Bus
: Optional.

  If the D−Bus low level C API is available, virt−p2v can send a D−Bus message to logind to inhibit power saving (sleep, suspend, etc) during P2V conversions.

  If this API is not available at build time, then very long conversions might be interrupted if the physical machine goes to sleep.

valgrind
: Optional. For testing memory problems.

bash-completion
: Optional. For tab-completion of commands in bash.

BUILDING FROM GIT

You will need to install additional dependencies `autoconf`, and `automake` when building from git.

```
git clone https://github.com/libguestfs/virt-p2v
cd virt-p2v
./autogen.sh
make
```

BUILDING FROM TARBALLS

Tarballs are downloaded from http://download.libguestfs.org/. Stable tarballs are signed with the GnuPG key for `rich@annexia.org`, see https://pgp.mit.edu/pks/lookup?op=vindex&search=0x91738F73E1B768A0. The fingerprint is`F777 4FB1 AD07 4A7E 8C87 67EA 9173 8F73 E1B7 68A0`.

Download and unpack the tarball.

```
cd virt-p2v-1.xx.yy
./configure
make
```

RUNNING THE TESTS

DO NOT run the tests as root! Virt-p2v can be built and tested as non-root. Running the tests as root could even be dangerous, don't do it.

To run the tests, do:

```
make check
```

There are many more tests you can run. See **p2v-hacking**(1) for details.

INSTALLING

DO NOT use `make install`! You'll end up with conflicting versions of virt-p2v installed, and this causes constant headaches for users. See the next section for how to use the *./run* script instead.

Distro packagers can use:

```
make DESTDIR=[temp-build-dir] install
```

THE ./run SCRIPT

You can test **virt-p2v**(1) and the other tools without needing to install them by using the *./run* script in the top directory. This script works by setting several environment variables.

For example:

```
./run virt-p2v-make-disk [usual virt-p2v-make-disk args ...]
```

The *./run* script adds every virt-p2v binary to the `$PATH`, so the above example run virt-p2v-make-disk from the build directory (not the globally installed virt-p2v-make-disk if there is one).

SELECTED ./configure SETTINGS

There are many `./configure` options. Use:

```
./configure --help
```

to list them all. This section covers some of the more important ones.

--disable-gnulib-tests
: On some platforms the GNUlib test suite can be flaky. This disables it, since errors in the GNUlib test suite are often not important.

−−enable−werror
This turns compiler warnings into errors (ie. `-Werror`). Use this for development, especially when submitting patches. It should generally *not* be used for production or distro builds.

−−with−extra="*distroname=version,...***"**
−−with−extra="local"
This option appends a text to the version of the virt−p2v tools. It is a free text field, but a good idea is to encode a comma-separated list of facts such as the distro name and version, and anything else that may help with debugging problems raised by users.

For custom and/or local builds, this can be set to `local` to indicate this is *not* a distro build.

−−with−gtk=3
This option forces virt−p2v to be built against Gtk 3, which is currently the most widely tested configuration.

USING CLANG (LLVM) INSTEAD OF GCC

```
export CC=clang
./configure
make
```

BUILDING i686 32 BIT VIRT−P2V

(This section only applies on the x86−64 architecture.)

Building a 32 bit virt−p2v (i686) binary improves compatibility with older hardware. See **virt−p2v−make−disk** (1) for details. Although virt−p2v is a simple Gtk application, it is not especially easy to build just virt−p2v as a 32 bit application on a 64 bit host. Usually the simplest way is to use a 32 bit chroot or even a 32 bit virtual machine to build virt−p2v.

On Fedora you can use the **mock** (1) tool. For example:

```
fedpkg mockbuild −−root fedora−23−i386
```

This will result in a *virt−v2v−*.i686.rpm* file which can be unpacked to extract the 32 bit virt−p2v binary.

The binary may be compressed to either *virt−p2v.i686.xz*, or *`$libdir`/virt−p2v/virt−p2v.i686.xz* or *`$VIRT_P2V_DATA_DIR`/virt−p2v.i686.xz* as appropriate. This enables the **virt−p2v−make−disk** (1) *−−arch* option.

SEE ALSO

p2v−hacking (1), **p2v−release−notes** (1), http://libguestfs.org/.

AUTHORS

Richard W.M. Jones (`rjones at redhat dot com`)

COPYRIGHT

LICENSE

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.