NAME

virt−make−fs − Make a filesystem from a tar archive or files

SYNOPSIS

```
 virt−make−fs [−−options] input.tar output.img

 virt−make−fs [−−options] input.tar.gz output.img

 virt−make−fs [−−options] directory output.img
```

DESCRIPTION

Virt-make-fs is a command line tool for creating a filesystem from a tar archive or some files in a directory. It is similar to tools like **mkisofs** (1), **genisoimage** (1) and **mksquashfs** (1). Unlike those tools, it can create common filesystem types like ext2/3 or NTFS, which can be useful if you want to attach these filesystems to existing virtual machines (eg. to import large amounts of read-only data to a VM).

To create blank disks, use **virt−format** (1). To create complex layouts, use **guestfish** (1).

Basic usage is:

```
 virt−make−fs input output.img
```

where `input` is either a directory containing files that you want to add, or a tar archive (either uncompressed tar or gzip-compressed tar); and *output.img* is a disk image. The input type is detected automatically. The output disk image defaults to a raw ext2 sparse image unless you specify extra flags (see ''OPTIONS'' below).

FILESYSTEM TYPE

The default filesystem type is `ext2`. Just about any filesystem type that libguestfs supports can be used (but *not* read-only formats like ISO9660). Here are some of the more common choices:

ext3
: Note that ext3 filesystems contain a journal, typically 1−32 MB in size. If you are not going to use the filesystem in a way that requires the journal, then this is just wasted overhead.

ntfs or *vfat*
: Useful if exporting data to a Windows guest.

minix
: Lower overhead than `ext2`, but certain limitations on filename length and total filesystem size.

EXAMPLE

```
 virt−make−fs −−type=minix input minixfs.img
```

TO PARTITION OR NOT TO PARTITION

Optionally virt-make-fs can add a partition table to the output disk.

Adding a partition can make the disk image more compatible with certain virtualized operating systems which don't expect to see a filesystem directly located on a block device (Linux doesn't care and will happily handle both types).

On the other hand, if you have a partition table then the output image is no longer a straight filesystem. For example you cannot run **fsck** (8) directly on a partitioned disk image. (However libguestfs tools such as **guestfish** (1) and **virt−resize** (1) can still be used).

EXAMPLE

Add an MBR partition:

```
 virt−make−fs −−partition −− input disk.img
```

If the output disk image could be terabyte-sized or larger, it's better to use an EFI/GPT−compatible partition table:

```
 virt−make−fs −−partition=gpt −−size=+4T −−format=qcow2 input disk.img
```

EXTRA SPACE

Unlike formats such as tar and squashfs, a filesystem does not "just fit" the files that it contains, but might have extra space. Depending on how you are going to use the output, you might think this extra space is wasted and want to minimize it, or you might want to leave space so that more files can be added later. Virt-make-fs defaults to minimizing the extra space, but you can use the *−−size* flag to leave space in the filesystem if you want it.

An alternative way to leave extra space but not make the output image any bigger is to use an alternative disk image format (instead of the default "raw" format). Using *−−format=qcow2* will use the native qemu/KVM qcow2 image format (check your hypervisor supports this before using it). This allows you to choose a large *−−size* but the extra space won't actually be allocated in the image until you try to store something in it.

Don't forget that you can also use local commands including **resize2fs** (8) and **virt−resize** (1) to resize existing filesystems, or rerun virt-make-fs to build another image from scratch.

EXAMPLE

```
 virt−make−fs −−format=qcow2 −−size=+200M input output.img
```

OPTIONS

−−help
Display brief help.

−−blocksize=512
−−blocksize=4096
This parameter sets the sector size of the output disk image.

The default is `512` bytes.

See also "guestfs_add_drive_opts" in **guestfs** (3).

−−floppy
Create a virtual floppy disk.

Currently this preselects the size (1440K), partition type (MBR) and filesystem type (VFAT). In future it may also choose the geometry.

−−size=N
−−size=+N
−s N
−s +N
Use the *−−size* (or *−s*) option to choose the size of the output image.

If this option is *not* given, then the output image will be just large enough to contain all the files, with not much wasted space.

To choose a fixed size output disk, specify an absolute number followed by b/K/M/G/T/P/E to mean bytes, Kilobytes, Megabytes, Gigabytes, Terabytes, Petabytes or Exabytes. This must be large enough to contain all the input files, else you will get an error.

To leave extra space, specify + (plus sign) and a number followed by b/K/M/G/T/P/E to mean bytes, Kilobytes, Megabytes, Gigabytes, Terabytes, Petabytes or Exabytes. For example: *−−size=+200M* means enough space for the input files, and (approximately) an extra 200 MB free space.

Note that virt-make-fs estimates free space, and therefore will not produce filesystems containing precisely the free space requested. (It is much more expensive and time-consuming to produce a filesystem which has precisely the desired free space).

−−format=FMT
−F FMT
Choose the output disk image format.

The default is `raw` (raw sparse disk image).

−−type=FS
−t FS
Choose the output filesystem type.

The default is `ext2`.

Any filesystem which is supported read-write by libguestfs can be used here.

−−label=LABEL
Set the filesystem label.

−−partition
−−partition=PARTTYPE
If specified, this flag adds an MBR partition table to the output disk image.

You can change the partition table type, eg. *−−partition=gpt* for large disks.

For MBR, virt-make-fs sets the partition type byte automatically.

−v
−−verbose
Enable debugging information.

−V
−−version
Display version number and exit.

−x Enable libguestfs trace.

SEE ALSO

guestfish (1), **virt−format** (1), **virt−resize** (1), **virt−tar−in** (1), **mkisofs** (1), **genisoimage** (1), **mksquashfs** (1), **mke2fs** (8), **resize2fs** (8), **guestfs** (3), http://libguestfs.org/.

AUTHOR

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

COPYRIGHT

Copyright (C) 2010−2020 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, 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.