NAME

guestfs−perl − How to use libguestfs from Perl

SYNOPSIS

```
use Sys::Guestfs;

my $g = Sys::Guestfs−>new ();
$g−>add_drive_opts ('guest.img', format => 'raw');
$g−>launch ();
$g−>mount ('/dev/sda1', '/');
$g−>touch ('/hello');
$g−>shutdown ();
$g−>close ();
```

DESCRIPTION

This manual page documents how to call libguestfs from the Perl programming language. This page just documents the differences from the C API and gives some examples. If you are not familiar with using libguestfs, you also need to read **guestfs** (3). To read the full Perl API, see **Sys::Guestfs** (3).

ERRORS

Errors from libguestfs functions turn into calls to `croak` (see **Carp** (3)).

EXAMPLE 1: CREATE A DISK IMAGE

```
#!/usr/bin/perl

# Example showing how to create a disk image.

use strict;
use warnings;
use Sys::Guestfs;

my $output = "disk.img";

my $g = new Sys::Guestfs ();

# Create a raw−format sparse disk image, 512 MB in size.
$g−>disk_create ($output, "raw", 512 * 1024 * 1024);

# Set the trace flag so that we can see each libguestfs call.
$g−>set_trace (1);

# Attach the disk image to libguestfs.
$g−>add_drive_opts ($output, format => "raw", readonly => 0);

# Run the libguestfs back−end.
$g−>launch ();

# Get the list of devices.  Because we only added one drive
# above, we expect that this list should contain a single
# element.
my @devices = $g−>list_devices ();
if (@devices != 1) {
    die "error: expected a single device from list−devices";
}

# Partition the disk as one single MBR partition.
$g−>part_disk ($devices[0], "mbr");
```

```
# Get the list of partitions.  We expect a single element, which
# is the partition we have just created.
my @partitions = $g->list_partitions ();
if (@partitions != 1) {
    die "error: expected a single partition from list-partitions";
}

# Create a filesystem on the partition.
$g->mkfs ("ext4", $partitions[0]);

# Now mount the filesystem so that we can add files.
$g->mount ($partitions[0], "/");

# Create some files and directories.
$g->touch ("/empty");
my $message = "Hello, world\n";
$g->write ("/hello", $message);
$g->mkdir ("/foo");

# This one uploads the local file /etc/resolv.conf into
# the disk image.
$g->upload ("/etc/resolv.conf", "/foo/resolv.conf");

# Because we wrote to the disk and we want to detect write
# errors, call $g->shutdown.  You don't need to do this:
# $g->close will do it implicitly.
$g->shutdown ();

# Note also that handles are automatically closed if they are
# reaped by reference counting.  You only need to call close
# if you want to close the handle right away.
$g->close ();
```

EXAMPLE 2: INSPECT A VIRTUAL MACHINE DISK IMAGE

```
#!/usr/bin/perl

# Example showing how to inspect a virtual machine disk.

use strict;
use warnings;
use Sys::Guestfs;

if (@ARGV < 1) {
    die "usage: inspect_vm disk.img"
}

my $disk = $ARGV[0];

my $g = new Sys::Guestfs ();

# Attach the disk image read-only to libguestfs.
# You could also add an optional format => ... argument here.  This is
# advisable since automatic format detection is insecure.
$g->add_drive_opts ($disk, readonly => 1);
```

```
# Run the libguestfs back−end.
$g−>launch ();

# Ask libguestfs to inspect for operating systems.
my @roots = $g−>inspect_os ();
if (@roots == 0) {
    die "inspect_vm: no operating systems found";
}

for my $root (@roots) {
    printf "Root device: %s\n", $root;

    # Print basic information about the operating system.
    printf "  Product name: %s\n", $g−>inspect_get_product_name ($root);
    printf "  Version:      %d.%d\n",
        $g−>inspect_get_major_version ($root),
        $g−>inspect_get_minor_version ($root);
    printf "  Type:         %s\n", $g−>inspect_get_type ($root);
    printf "  Distro:       %s\n", $g−>inspect_get_distro ($root);

    # Mount up the disks, like guestfish −i.
    #
    # Sort keys by length, shortest first, so that we end up
    # mounting the filesystems in the correct order.
    my %mps = $g−>inspect_get_mountpoints ($root);
    my @mps = sort { length $a <=> length $b } (keys %mps);
    for my $mp (@mps) {
        eval { $g−>mount_ro ($mps{$mp}, $mp) };
        if ($@) {
            print "$@ (ignored)\n"
        }
    }

    # If /etc/issue.net file exists, print up to 3 lines.
    my $filename = "/etc/issue.net";
    if ($g−>is_file ($filename)) {
        printf "--- %s ---\n", $filename;
        my @lines = $g−>head_n (3, $filename);
        print "$_\n" foreach @lines;
    }

    # Unmount everything.
    $g−>umount_all ()
}
```

SEE ALSO

Sys::Guestfs (3), **guestfs** (3), **guestfs−examples** (3), **guestfs−erlang** (3), **guestfs−gobject** (3), **guestfs−golang** (3), **guestfs−java** (3), **guestfs−lua** (3), **guestfs−ocaml** (3), **guestfs−python** (3), **guestfs−recipes** (1), **guestfs−ruby** (3), http://libguestfs.org/.

AUTHORS

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