Rocky Enterprise Linux 9.2 Manual Pages on command 'virt-viewer.1'

$ man virt-viewer.1

Virt-Viewer(1) Virtualization Support Virt-Viewer(1)

NAME

virt-viewer - display the graphical console for a virtual machine

SYNOPSIS

virt-viewer [OPTIONS] [ID|UUID|DOMAIN-NAME]

DESCRIPTION

virt-viewer is a minimal tool for displaying the graphical console of a virtual machine. The console is accessed using the VNC or SPICE protocol. The guest can be referred to based on its name, ID, or UUID. If the guest is not already running, then the viewer can be told to wait until it starts before attempting to connect to the console. The viewer can connect to remote hosts to lookup the console information and then also connect to the remote console using the same network transport.

In some circumstances the viewer may need to grab the mouse pointer. The default key sequence for releasing the grab is "Ctrl_L"+"Alt_L", however, this can be overridden using the "--hotkeys" argument documented below.

OPTIONS

The following options are accepted when running "virt-viewer":

-h, --help

Display command line help summary

-V, --version

Display program version number

-v, --verbose

Display information about the connection

-c URI, --connect=URI

Specify the hypervisor connection URI

-w, --wait

Wait for the domain to start up before attempting to connect to the console

-r, --reconnect

Automatically reconnect to the domain if it shuts down and restarts

-z PCT, --zoom=PCT

Zoom level of the display window in percentage. Range 10-400.

-d, --direct

Do not attempt to tunnel the console over SSH, even if the main connection URI used SSH.

-a, --attach

Instead of making a direct TCP/UNIX socket connection to the remote display, ask libvirt to provide a pre-connected socket for the display. This avoids the need to authenticate with the remote display server directly. This option will only work when connecting to a guest that is running on the same host as the virt-viewer program. If attaching to the guest via libvirt fails, virt-viewer will automatically fallback to trying a regular direct TCP/UNIX socket connection.

-f, --full-screen

Start with the window maximised to fullscreen

If supported, the remote display will be reconfigured to match the physical client monitor configuration, by enabling or disabling

extra monitors as necessary. This is currently implemented by the Spice backend only.

To specify which client monitors are used in fullscreen mode, see the CONFIGURATION section below.

--auto-resize <always|never>

Controls whether it is permitted to attempt to resize the remote framebuffer to match the local window size. This currently defaults to on, but note that not all servers will support this.

-s, --shared

Permitted a shared session with multiple clients

--cursor auto|local

Control how the mouse cursor is rendered. "auto" is the default behaviour, which will honour the behaviour requested by the remote server. This may involve the server remote rendering the cursor into the framebuffer, or sending the cursor details to the client to render. "local" overrides this default to request that the local desktop cursor is always rendered regardless of what the server requests. The latter is rarely needed, but can be used if the server has a bad configuration that results in its own cursor being hidden.

--debug

Print debugging information

-H HOTKEYS, --hotkeys HOTKEYS

Set global hotkey bindings. By default, keyboard shortcuts only work when the guest display widget does not have focus. Any actions specified in HOTKEYS will be effective even when the guest display widget has input focus. The format for HOTKEYS is <action1>=<key1>[+<key2>][,<action2>=<key3>[+<key4>]]. Key-names are case-insensitive. Valid actions are: toggle-fullscreen, release-cursor, zoom-in, zoom-out, zoom-reset, secure-attention, usb-device-reset, smartcard-insert and smartcard-remove. The "secure-attention" action sends a secure attention sequence (Ctrl+Alt+Del) to the guest. Examples:

--hotkeys=toggle-fullscreen=shift+f11,release-cursor=shift+f12

--hotkeys=release-cursor=ctrl+alt

Note that hotkeys for which no binding is given are disabled.

Although the hotkeys specified here are handled by the client, it is still possible to send these key combinations to the guest via a menu item.

-K, --keymap

Remap and/or block supplied keypresses to the host. All key identifiers are case-sensitive and follow the naming convention as defined in gdkkeysyms.h without the GDK_KEY_ prefix.

Running the application with --debug will display keypress symbols in the following way:

"Key pressed was keycode='0x63', gdk_keyname='c'"

"Key pressed was keycode='0xffeb', gdk_keyname='Super_L'"

The format for supplying a keymap is:

<srcKeySym1>=[<destKeySym1>][+<destKeySym2][,<srckeySym2>=[<destKeySym1]

To block a keypress simply assign an empty parameter to the srcKeySym.

Example:

--keymap=Super_L=,Alt_L=,1=Shift_L+F1,2=Shift_L+F2

This will block the Super_L (typically Windows Key) and ALT_L keypresses and remap key 1 to Shift F1, 2 to Shift F2.

-k, --kiosk

Start in kiosk mode. In this mode, the application will start in fullscreen with minimal UI. It will prevent the user from quitting or performing any interaction outside of usage of the remote desktop session.

Note that it can't offer a complete secure solution by itself. Your kiosk system must have additional configuration and security settings to lock down the OS. In particular, you must configure or disable the window manager, limit the session capabilities, use some restart/watchdog mechanism, disable VT switching etc.

--kiosk-quit <never|on-disconnect>

By default, when kiosk mode is enabled, virt-viewer will remain open when the connection to the remote server is terminated. By setting kiosk-quit option to "on-disconnect" value, virt-viewer will quit instead. Please note that --reconnect takes precedence over this option, and will attempt to do a reconnection before it quits.

--id, --uuid, --domain-name

Connect to the virtual machine by its id, uuid or name. These options are mutual exclusive. For example the following command may sometimes connect to a virtual machine with the id 2 or with the name 2 (depending on the number of running machines):

virt-viewer 2

To always connect to the virtual machine with the name "2" use the "--domain-name" option:

virt-viewer --domain-name 2

CONFIGURATION

A small number of configuration options can be controlled by editing the settings file located in the user configuration directory:

<USER-CONFIG-DIR>/virt-viewer/settings

This file is a text file in INI format, with application options in the [virt-viewer] group and per-guest options in a group identified by the guest's UUID. The application options should not be edited manually.

There is also a special [fallback] group which specifies options for all guests that don't have an explicit group.

For each guest, the initial fullscreen monitor configuration can be specified by using the monitor-mapping key. This configuration only takes effect when the -f/--full-screen option is specified.

The value of this key is a list of mappings between a guest display and a client monitor. Each mapping is separated by a semicolon character, and the mappings have the format <GUEST-DISPLAY-ID>:<CLIENT-MONITOR-ID>.

For example, to map guest displays 1 and 2 to client monitors 2 and 3 for the guest with a UUID of e4591275-d9d3-4a44-a18b-ef2fbc8ac3e2, use:

[e4591275-d9d3-4a44-a18b-ef2fbc8ac3e2]

monitor-mapping=1:2;2:3

The monitor-mapping must contain ids of all displays from 1 to the last desired display id, e.g. "monitor-mapping=3:3" is invalid because mappings for displays 1 and 2 are not specified.

EXAMPLES

To connect to the guest called 'demo' running under Xen

```
virt-viewer demo
```

To use GUI for connecting to a guest running under QEMU

```
virt-viewer --connect qemu:///system
```

To connect to the guest with ID 7 running under QEMU

```
virt-viewer --connect qemu:///system 7
```

To wait for the guest with UUID 66ab33c0-6919-a3f7-e659-16c82d248521 to startup and then connect, also reconnecting upon restart of VM

```
virt-viewer --reconnect --wait 66ab33c0-6919-a3f7-e659-16c82d248521
```

To connect to a remote console using TLS

```
virt-viewer --connect xen://example.org/ demo
```

To connect to a remote host using SSH, lookup the guest config and then make a tunnelled connection of the console

```
virt-viewer --connect qemu+ssh://root@example.org/system demo
```

When using a SSH tunnel to connect to a SPICE console, it's recommended to have ssh-agent running to avoid getting multiple authentication prompts.

To connect to a remote host using SSH, lookup the guest config and then make a direct non-tunnelled connection of the console

```
virt-viewer --direct --connect xen+ssh://root@example.org/ demo
```

AUTHOR

Written by Daniel P. Berrange, based on the GTK-VNC example program gvncviewer.

BUGS

Report bugs to https://gitlab.com/virt-viewer/virt-viewer/-/issues

COPYRIGHT

is free software. You may redistribute copies of it under the terms of

the GNU General Public License

"https://www.gnu.org/licenses/gpl-2.0.html". There is NO WARRANTY, to

the extent permitted by law.

SEE ALSO

virsh(1), "virt-manager(1)", "spice-client(1)", the project website

"http://gitlab.com/virt-viewer/virt-viewer"