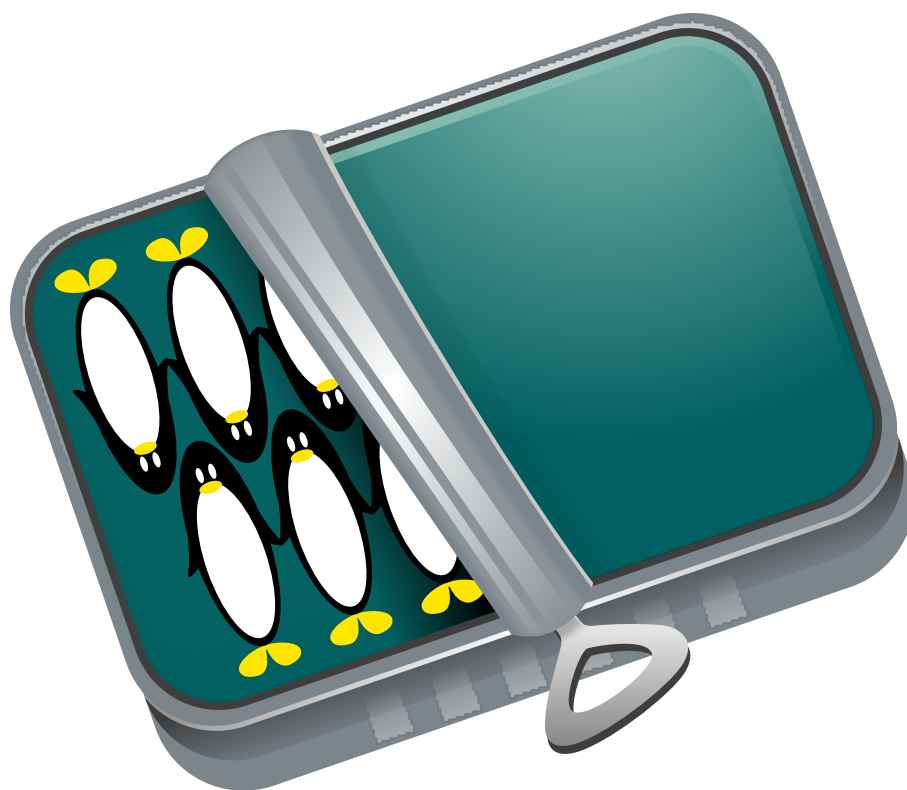


libvirt 0.8.6

Virsh Command Reference



Justin Clift

libvirt 0.8.6 Virsh Command Reference Edition 0

Author

Justin Clift

jclift@redhat.com

Copyright © 2010 Red Hat, Inc.

Copyright © 2009-2010 Red Hat, Inc. and others.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at <http://creativecommons.org/licenses/by-sa/3.0/>. The original authors of this document, and Red Hat, designate the libvirt Project as the "Attribution Party" for purposes of CC-BY-SA. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

All other trademarks are the property of their respective owners.

This book is a work in progress (recently begun), to comprehensively document each command available in **virsh**, the libvirt command shell.

Preface	v
1. Document Conventions	v
1.1. Typographic Conventions	v
1.2. Pull-quote Conventions	vi
1.3. Notes and Warnings	vii
2. We Need Feedback!	vii
1. Command List	1
2. Commands	3
2.1. net-autostart	3
2.2. net-create	5
2.3. net-define	8
2.4. net-destroy	10
2.5. net-dumpxml	12
2.6. net-edit	14
2.7. net-list	16
2.8. net-name	18
2.9. net-start	19
2.10. net-undefine	21
2.11. net-uuid	23
A. Revision History	25
Index	27

Preface

1. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the [Liberation Fonts](https://fedorahosted.org/liberation-fonts/)¹ set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later includes the Liberation Fonts set by default.

1.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

Mono-spaced Bold

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keycaps and key combinations. For example:

To see the contents of the file **my_next_bestselling_novel** in your current working directory, enter the **cat my_next_bestselling_novel** command at the shell prompt and press **Enter** to execute the command.

The above includes a file name, a shell command and a keycap, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from keycaps by the hyphen connecting each part of a key combination. For example:

Press **Enter** to execute the command.

Press **Ctrl+Alt+F2** to switch to the first virtual terminal. Press **Ctrl+Alt+F1** to return to your X-Windows session.

The first paragraph highlights the particular keycap to press. The second highlights two key combinations (each a set of three keycaps with each set pressed simultaneously).

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **mono-spaced bold**. For example:

File-related classes include **filesystem** for file systems, **file** for files, and **dir** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialog box text; labeled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

Choose **System** → **Preferences** → **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, click the **Left-handed mouse** check box and click

¹ <https://fedorahosted.org/liberation-fonts/>

Close to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).

To insert a special character into a **gedit** file, choose **Applications** → **Accessories** → **Character Map** from the main menu bar. Next, choose **Search** → **Find...** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit** → **Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

Mono-spaced Bold Italic or ***Proportional Bold Italic***

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

To connect to a remote machine using ssh, type **ssh *username@domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh *john@example.com***.

The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount */home***.

To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above — *username*, *domain.name*, *file-system*, *package*, *version* and *release*. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

Publican is a *DocBook* publishing system.

1.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```
books      Desktop  documentation  drafts  mss    photos  stuff  svn
books_tests Desktop1  downloads      images  notes  scripts  svgs
```

Source-code listings are also set in **mono-spaced roman** but add syntax highlighting as follows:

```
package org.jboss.book.jca.ex1;
import javax.naming.InitialContext;
```

```
public class ExClient
{
    public static void main(String args[])
        throws Exception
    {
        InitialContext iniCtx = new InitialContext();
        Object          ref    = iniCtx.lookup("EchoBean");
        EchoHome        home   = (EchoHome) ref;
        Echo             echo   = home.create();

        System.out.println("Created Echo");

        System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
    }
}
```

1.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.



Note

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.



Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled 'Important' will not cause data loss but may cause irritation and frustration.



Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

2. We Need Feedback!

If you find a typographical error in this manual, or if you have thought of a way to make this manual better, we would love to hear from you! Please submit a bug report at <http://libvirt.org/bugs.html>

Command List

These are the commands presently available in `virsh`.

Table 1.1. Virtual Networking

Command	Available from	Description
<i>net-autostart</i>	libvirt 0.2.1	Configure a network to be automatically started at boot
<i>net-create</i>	libvirt 0.2.0	Creates a new transient virtual network from an XML file
<i>net-define</i>	libvirt 0.2.0	Adds a new permanent virtual network from an XML file, without starting it
<i>net-destroy</i>	libvirt 0.2.0	Shuts down a running virtual network
<i>net-dumpxml</i>	libvirt 0.2.0	Displays the XML configuration for a virtual network (to stdout)
<i>net-edit</i>	libvirt 0.4.6	Allows the user to edit the XML configuration of a virtual network, using their preferred editor
<i>net-list</i>	libvirt 0.2.0	Lists the virtual networks libvirt is aware of
<i>net-name</i>	libvirt 0.2.0	When given a network UUID, returns its corresponding network name
<i>net-start</i>	libvirt 0.2.0	Starts a (previously defined) inactive virtual network
<i>net-undefine</i>	libvirt 0.2.0	Removes an inactive virtual network from the libvirt configuration
<i>net-uuid</i>	libvirt 0.2.0	When given a network name, returns its corresponding UUID

Commands

2.1. net-autostart

Used to enable and disable the automatic starting of a given *persistent* virtual network, by the libvirt daemon.

Usage

```
net-autostart --network network-identifier --disable
```

Options

Table 2.1. Options

Name	Required?	Description
<code>--network <i>network-identifier</i></code>	Required	The name or UUID for the virtual network being configured. The word " <code>--network</code> " itself is optional.
<code>--disable</code>	Optional	Disables the automatic starting of the virtual network.

Availability

Available from libvirt 0.2.1 onwards

Platform or Hypervisor specific notes

None yet

Examples

Example 2.1.

```
virsh # net-autostart default --disable
```

Stops the virtual network named "*default*" from automatically starting when the libvirt daemon starts.

Example 2.2.

```
virsh # net-autostart --network default --disable
```

Same as the above example.

Example 2.3.

```
virsh # net-autostart bfb4c69-7d6a-cc9a-904c-09910ce179c0 --disable
```

Stops the virtual network with UUID "*bfb4c69-7d6a-cc9a-904c-09910ce179c0*" from automatically starting when the libvirt daemon starts.

Example 2.4.

```
virsh # net-autostart --network bfb4c69-7d6a-cc9a-904c-09910ce179c0 --disable
```

Same as the above example.

Example 2.5.

```
virsh # net-autostart default
```

Enables the automatic starting of the virtual network named "*default*", by the libvirt daemon when it starts.

Example 2.6.

```
virsh # net-autostart --network default
```

Same as the above example.

Example in context

Starting with an XML file we've already created, using the [required XML format](#)¹:

```
<network>
  <name>examplenetwork</name>
  <bridge name="virbr100" />
  <forward mode="route" />
  <ip address="10.10.120.1" netmask="255.255.255.0" />
</network>
```

```
# ls -al /root/examplenetwork.xml
-rw-r--r--. 1 root root 162 Nov  7 16:43 /root/examplenetwork.xml
```

We start `virsh` interactively, then define a *persistent* virtual network:

```
# virsh
Welcome to virsh, the virtualization interactive terminal.

Type: 'help' for help with commands
      'quit' to quit
```

```
virsh # net-list
Name           State      Autostart
-----
```

¹ <http://libvirt.org/formatnetwork.html>

```
default          active    yes
```

```
virsh # net-define /root/examplenetwork.xml
Network examplenetwork defined from /root/examplenetwork.xml
```

Newly defined virtual networks aren't set to automatically be started, as can be seen here:

```
virsh # net-list --all
Name           State      Autostart
-----
default        active    yes
examplenetwork  inactive no      <-- this is the important bit
```

We enable automatic starting for it:

```
virsh # net-autostart examplenetwork
Network examplenetwork marked as autostarted
```

Checking, to make sure:

```
virsh # net-list --all
Name           State      Autostart
-----
default        active    yes
examplenetwork  inactive yes      <-- this is the important bit
```

From now on, whenever the libvirt daemon is started, it will automatically start this virtual network too (unless it's already running).

If at some point we want to turn off automatic starting of the virtual network, we use the `--disable` option to the command:

```
# net-autostart --disable examplenetwork
Network examplenetwork unmarked as autostarted
```

```
virsh # net-list --all
Name           State      Autostart
-----
default        active    yes
examplenetwork  inactive no      <-- this is the important bit
```

See also

- [net-define](#) - Adds a new **persistent** virtual network to libvirt, without starting it, using settings from an XML file.
- [net-list](#) - Displays a list of the virtual networks libvirt is aware of.

2.2. net-create

Creates a running, *transient* virtual network, using settings from an XML file.

Usage

```
net-create --file file-name
```

Options

Table 2.2. Options

Name	Required?	Description
<code>--file <i>file-name</i></code>	Required	The full path (and file name) to an XML file containing the network settings required ² . The word " <code>--file</code> " itself is optional.

Availability

Available from libvirt 0.2.0 onwards

Platform or Hypervisor specific notes

None yet

Examples

Example 2.7.

```
virsh # net-create /root/examplenetwork.xml
```

Creates a new, transient, virtual network using the settings from `/root/examplenetwork.xml`.

Example 2.8.

```
virsh # net-create --file /root/examplenetwork.xml
```

Same as the above example.

Example in context

Starting with an XML file we've already created, using the [required XML format](#)³:

```
<network>
  <name>examplenetwork</name>
  <bridge name="virbr100" />
  <forward mode="route" />
  <ip address="10.10.120.1" netmask="255.255.255.0" />
</network>
```

```
# ls -al /root/examplenetwork.xml
-rw-r--r--. 1 root root 162 Nov  7 16:43 /root/examplenetwork.xml
```

³ <http://libvirt.org/formatnetwork.html>

We start virsh interactively, then create the *transient* virtual network:

```
# virsh
Welcome to virsh, the virtualization interactive terminal.

Type: 'help' for help with commands
      'quit' to quit
```

```
virsh # net-list
Name          State      Autostart
-----
default       active     yes
```

```
virsh # net-create /root/examplenetwork.xml
Network examplenetwork created from /root/examplenetwork.xml
```

Created. Now we confirm:

```
virsh # net-list
Name          State      Autostart
-----
default       active     yes
examplenetwork active no
```

We check the details of the created network from virsh. This shows us the generated UUID, and anything else that may be in effect (ie Spanning Tree Protocol).

```
virsh # net-dumpxml examplenetwork
<network>
  <name>examplenetwork</name>
  <uuid>97ce3914-231e-4026-0a78-822e1e2e7226</uuid>
  <forward mode='route' />
  <bridge name='virbr100' stp='on' delay='0' />
  <ip address='10.10.120.1' netmask='255.255.255.0'>
  </ip>
</network>
```

Then, after exiting virsh, we check how it appears to the host Linux OS:

```
# ifconfig virbr100
virbr100 Link encap:Ethernet HWaddr 02:95:C3:06:A5:BF
          inet addr:10.10.120.1 Bcast:10.10.120.255 Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b) TX bytes:2653 (2.5 KiB)
```

See also

- [net-define](#) - An alternative to [net-create](#). Use this when you want a **persistent** virtual network, that will last through reboots and shutdowns, rather than a transient one created by [net-create](#).
- [net-destroy](#) - Shuts down a running virtual network, as started with [net-create](#) or [net-start](#).

- [XML Network Format Reference page](#)⁴ - Gives the details of the XML needed by *net-create*.

2.3. net-define

Adds a new *persistent* virtual network to libvirt, without starting it, using settings from an XML file.

Unless you enable automatic starting of it, by the libvirt daemon when it starts, you will need to manually start this virtual network as needed, using *net-start*.

To enable automatic starting of this virtual network, use the virsh *net-autostart* command.

Usage

```
net-define --file file-name
```

Options

Table 2.3. Options

Name	Required?	Description
<code>--file file-name</code>	Required	The full path (and file name) to an XML file containing the network settings required ⁵ . The word " <code>--file</code> " itself is optional.

Availability

Available from libvirt 0.2.0 onwards

Platform or Hypervisor specific notes

None yet

Examples

Example 2.9.

```
virsh # net-define /root/exampnetwork.xml
```

Creates a new, persistent, virtual network using the settings from the XML file `/root/exampnetwork.xml`.

Example 2.10.

```
virsh # net-define --file /root/exampnetwork.xml
```

Same as the above example.

Example in context

Starting with an XML file we've already created, using the [required XML format](#)⁶:

⁶ <http://libvirt.org/formatnetwork.html>


```
<network>
  <name>examplenetwork</name>
  <bridge name="virbr100" />
  <forward mode="route" />
  <ip address="10.10.120.1" netmask="255.255.255.0" />
</network>
```

```
# ls -al /root/examplenetwork.xml
-rw-r--r--. 1 root root 162 Nov  7 16:43 /root/examplenetwork.xml
```

We start virsh interactively, then create the *transient* virtual network:

```
# virsh
Welcome to virsh, the virtualization interactive terminal.

Type: 'help' for help with commands
      'quit' to quit
```

```
virsh # net-list
Name                State      Autostart
-----
default             active     yes
```

```
virsh # net-define /root/examplenetwork.xml
Network examplenetwork defined from /root/examplenetwork.xml
```

Defined. Now we confirm:

```
virsh # net-list --all
Name                State      Autostart
-----
default             active     yes
examplenetwork    inactive no
```

Newly defined virtual networks aren't automatically started, so we manually start it now:

```
virsh # net-start examplenetwork
Network examplenetwork started
```

```
virsh # net-list
Name                State      Autostart
-----
default             active     yes
examplenetwork    active    no
```

We check the details of the started network from virsh. This shows us the generated UUID, and anything else that may be in effect (ie Spanning Tree Protocol).

```
virsh # net-dumpxml examplenetwork
<network>
  <name>examplenetwork</name>
  <uuid>97ce3914-231e-4026-0a78-822e1e2e7226</uuid>
  <forward mode='route' />
  <bridge name='virbr100' stp='on' delay='0' />
  <ip address='10.10.120.1' netmask='255.255.255.0'>
  </ip>
</network>
```

If the virtualisation server is running Linux, we can check how it appears to the host OS:

```
# ifconfig virbr100
virbr100 Link encap:Ethernet HWaddr A6:45:97:AE:8E:08
inet addr:10.10.120.1 Bcast:10.10.120.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:11 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:0 (0.0 b) TX bytes:2653 (2.5 KiB)
```

See also

- [net-autostart](#) - Used to enable and disable the automatic starting of a virtual network.
- [net-create](#) - An alternative to [net-define](#). Use this when you want a **transient** virtual network that will disappear when the host is rebooted or shutdown, rather than a **persistent** one created using [net-define](#).
- [net-destroy](#) - Shuts down a running virtual network, as started with [net-create](#) or [net-start](#).
- [net-start](#) - Manually starts a virtual network that isn't running.
- [XML Network Format Reference page](#)⁷ - Gives the details of the XML needed by [net-define](#).

2.4. net-destroy

Shuts down a virtual network, releasing any resources in use by it.

Usage

```
net-destroy --network network-identifier
```

Options

Table 2.4. Options

Name	Required?	Description
<code>--network <i>network-identifier</i></code>	Required	The name or UUID of the network to be shut down.. The word " <code>--network</code> " itself is optional.

Availability

Available from libvirt 0.2.0 onwards

Platform or Hypervisor specific notes

Linux

If the virtualisation host is running Linux, the name the operating system uses for the network interface can be found using the [net-dumpxml](#) virsh command.

Look for the name value of the **bridge** line. **virbr100** in this instance:

```
virsh # net-dumpxml examplenetwork
<network>
  <name>examplenetwork</name>
  <uuid>b7005dec-be1a-fe9a-338a-0cb1301dfcfd</uuid>
  <forward mode='route' />
  <bridge name='virbr100' stp='on' delay='0' />
  <ip address='10.10.120.1' netmask='255.255.255.0'>
  </ip>
</network>
```

Using **ifconfig**, or a similar tool such as **ip**, the **virbr100** interface will be seen on the host when the virtual network is running:

```
# ifconfig virbr100
virbr100 Link encap:Ethernet HWaddr D2:43:D9:47:FA:AA
          inet addr:10.10.120.1 Bcast:10.10.120.255 Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b) TX bytes:1553 (1.5 KiB)
```

After shutting down a virtual network with the **net-destroy** command, the Linux OS will no longer show this interface:

```
# ifconfig virbr100
virbr100: error fetching interface information: Device not found
```

Examples

Example 2.11.

```
virsh # net-destroy mynetwork
```

Shuts down the virtual network named "mynetwork".

Example 2.12.

```
virsh # net-destroy --network mynetwork
```

Same as the above example.

Example 2.13.

```
virsh # net-destroy bfb4c69-7d6a-cc9a-904c-09910ce179c0
```

Shuts down the virtual network that has a UUID of "*bfb4c69-7d6a-cc9a-904c-09910ce179c0*".

Example 2.14.

```
virsh # net-destroy --network bfb4c69-7d6a-cc9a-904c-09910ce179c0
```

Same as the above example.

Example in context

Starting with a virtual network named *examplenetwork*, already running on a virtualisation host server:

```
virsh # net-list
Name          State      Autostart
-----
default       active    yes
examplenetwork active   yes
```

The network is shut down by simply using the **net-destroy** command on it:

```
# net-destroy examplenetwork
Network examplenetwork destroyed
```

The *net-list* command now shows it as inactive:

```
virsh # net-list --all
Name          State      Autostart
-----
default       active    yes
examplenetwork inactive yes
```

See also

- [net-create](#) - Creates a running, **transient** virtual network, using settings from an XML file.
- [net-list](#) - Displays a list of the virtual networks libvirt is aware of.
- [net-start](#) - Manually starts a virtual network that isn't running.

2.5. net-dumpxml

Outputs the XML configuration for a virtual network, to stdout.

Usage

```
net-dumpxml --network network-identifier
```

Options

Table 2.5. Options

Name	Required?	Description
<code>--network network-identifier</code>	Required	The name or UUID of the network whose XML configuration is to be displayed. The word " <code>--network</code> " itself is optional.

Availability

Available from libvirt 0.2.0 onwards

Platform or Hypervisor specific notes

None yet

Examples

Example 2.15.

```
virsh # net-dumpxml mynetwork
```

Outputs the XML configuration for the virtual network named "mynetwork".

Example 2.16.

```
virsh # net-dumpxml --network mynetwork
```

Same as the above example.

Example 2.17.

```
virsh # net-dumpxml bfb4c69-7d6a-cc9a-904c-09910ce179c0
```

Outputs the XML configuration for the virtual network that has a UUID of "bfb4c69-7d6a-cc9a-904c-09910ce179c0".

Example 2.18.

```
virsh # net-dumpxml --network bfb4c69-7d6a-cc9a-904c-09910ce179c0
```

Same as the above example.

Example in context

Starting with a few virtual networks already defined:

```
virsh # net-list --all
Name           State      Autostart
-----
default        active    yes
examplenetwork active    no
```

We use **net-dumpxml** to look at the XML configuration for "examplenetwork":

```
virsh # net-dumpxml examplenetwork
<network>
  <name>examplenetwork</name>
  <uuid>b7005dec-be1a-fe9a-338a-0cb1301dfcfd</uuid>
  <forward mode='route' />
  <bridge name='virbr100' stp='on' delay='0' />
  <ip address='10.10.120.1' netmask='255.255.255.0'>
  </ip>
</network>
```

Done.

See also

- [net-list](#) - Displays a list of the virtual networks libvirt is aware of.

2.6. net-edit

Allows the user to edit the XML configuration of a virtual network, using their preferred editor.

net-edit launches the command (or script) is defined in the users *\$EDITOR* environment variable, passing it a temporary copy of the XML configuration for the virtual network.

When the user exits the editor, **net-edit** checks if the temporary file was changed.

If it was, then **net-edit** validates it to ensure it's error free. If no errors are found, **net-edit** then overwrites the existing saved virtual network configuration using it.

Usage

```
net-edit --network network-identifier
```

Options

Table 2.6. Options

Name	Required?	Description
<code>--network network-identifier</code>	Required	The name or UUID of the virtual network whose XML configuration is to be edited. The word " <code>--network</code> " itself is optional.

Availability

Available from libvirt 0.4.6 onwards

Platform or Hypervisor specific notes

None yet

Examples

Example 2.19.

```
virsh # net-edit mynetwork
```

Edits the XML configuration for the virtual network named "mynetwork".

Example 2.20.

```
virsh # net-edit --network mynetwork
```

Same as the above example.

Example 2.21.

```
virsh # net-edit bfb4c69-7d6a-cc9a-904c-09910ce179c0
```

Edits the XML configuration for the virtual network having UUID "bfb4c69-7d6a-cc9a-904c-09910ce179c0".

Example 2.22.

```
virsh # net-edit --network bfb4c69-7d6a-cc9a-904c-09910ce179c0
```

Same as the above example.

Example in context

Starting with a few virtual networks already defined:

```
virsh # net-list --all
Name                State      Autostart
-----
default             active    yes
examp1enetwork      active    no
```

We use `net-dumpxml` to view the XML configuration for "examp1enetwork":

```
virsh # net-dumpxml examp1enetwork
<network>
  <name>examp1enetwork</name>
  <uuid>b7005dec-be1a-fe9a-338a-0cb1301dfcfd</uuid>
  <forward mode='route' />
  <bridge name='virbr100' stp='on' delay='0' />
  <ip address='10.10.120.1' netmask='255.255.0'>
  </ip>
</network>
```

We want to change one of the values, for example, the Spanning Tree Protocol delay of 0. Let's say we want it to be 30 (seconds) instead.

Using **net-edit**, we launch an editor on the XML fragment. (**vi** is the editor shown):

```
virsh # net-edit examplenetwork
```

The editor window appears, and we make the change directly:

```
<network>
  <name>examplenetwork</name>
  <uuid>b7005dec-be1a-fe9a-338a-0cb1301dfcfd</uuid>
  <forward mode='route' />
  <bridge name='virbr100' stp='on' delay='30' />      <-- 0 changed to 30 here
  <ip address='10.10.120.1' netmask='255.255.255.0'>
  </ip>
</network>
~
~
~
~
~
~
"/tmp/virsh2Uz6L" 8L, 238C
```

Then save the (temporary) file and exit the editor. **net-edit** automatically copies the temporary XML to the saved configuration, if no errors in it were detected.

```
Network examplenetwork XML configuration edited.
```

The next time the "*examplenetwork*" virtual network is started, it will use the new value.

See also

- [net-dumpxml](#) - Outputs the XML configuration for a virtual network, to stdout.
- [net-list](#) - Displays a list of the virtual networks libvirt is aware of.

2.7. net-list

Lists the virtual networks libvirt is aware of, along with basic status and autostart information

Used without parameters, **net-list** displays information for only *active* virtual networks.

Usage

```
net-list --all --inactive
```

Options

Table 2.7. Options

Name	Required?	Description
<code>--all</code>	Optional	Instructs net-list to display both <i>active</i> and <i>inactive</i> virtual networks

Name	Required?	Description
<code>--inactive</code>	Optional	Instructs net-list to only display <i>inactive</i> virtual networks.

Availability

Available from libvirt 0.2.0 onwards

Platform or Hypervisor specific notes

None yet

Examples

Example 2.23.

```
virsh # net-list
```

Displays the *active* libvirt virtual networks.

Example 2.24.

```
virsh # net-list --all
```

Displays all virtual networks libvirt knows of, both *active* and *inactive*.

Example 2.25.

```
virsh # net-list --inactive
```

Displays only the *inactive* libvirt virtual networks.

Example in context

Displaying all of the libvirt virtual networks on a host:

```
virsh # net-list --all
Name                State      Autostart
-----
default             active     yes      <-- this is a virtual network
examplenetwork      inactive  no       <-- this is a virtual network
```

See also

- [net-autostart](#) - Used to enable and disable the automatic starting of a virtual network.
- [net-destroy](#) - Shuts down a running virtual network, as started with [net-create](#) or [net-start](#).
- [net-start](#) - Manually starts a virtual network that isn't running.

2.8. net-name

When given a virtual network UUID, returns its corresponding virtual network name

Usage

```
net-name --network network-UUID
```

Options

Table 2.8. Options

Name	Required?	Description
<code>--network <i>network-UUID</i></code>	Required	The UUID of the virtual network you want the name for. The word " <code>--network</code> " itself is optional.

Availability

Available from libvirt 0.2.0 onwards

Platform or Hypervisor specific notes

None yet

Examples

Example 2.26.

```
virsh # net-name b7005dec-be1a-fe9a-338a-0cb1301dfcfd
```

Returns the name of the virtual network having a UUID of "b7005dec-be1a-fe9a-338a-0cb1301dfcfd".

Example 2.27.

```
virsh # net-name --network b7005dec-be1a-fe9a-338a-0cb1301dfcfd
```

Same as the above example.

Example in context

Given a virtual network UUID, we can determine which virtual network it belongs to:

```
virsh # net-name b7005dec-be1a-fe9a-338a-0cb1301dfcfd  
examplenetwork
```

We can confirm by using the `net-dumpxml` command on the returned network name:

```
virsh # net-dumpxml examplenetwork  
<network>  
  <name>examplenetwork</name>          <-- the name is here
```

```
<uuid>b7005dec-be1a-fe9a-338a-0cb1301dfcfd</uuid> <-- the UUID is here
<forward mode='route' />
<bridge name='virbr100' stp='on' delay='1' />
<ip address='10.10.120.1' netmask='255.255.255.0'>
</ip>
</network>
```

Using **net-name** is more efficient than dumping the XML for the virtual network and manually extracting the *name* value.

See also

- [net-dumpxml](#) - Outputs the XML configuration for a virtual network, to stdout.

2.9. net-start

Starts an inactive, previously defined, virtual network.

Usage

```
net-start --network network-identifier
```

Options

Table 2.9. Options

Name	Required?	Description
<code>--network network-identifier</code>	Required	The name or UUID of the virtual network to start. The word " <code>--network</code> " itself is optional.

Availability

Available from libvirt 0.2.0 onwards

Platform or Hypervisor specific notes

None yet

Examples

Example 2.28.

```
virsh # net-start examplenetwork
```

Starts the virtual network named "examplenetwork".

Example 2.29.

```
virsh # net-start --network examplenetwork
```

Same as the above example.

Example 2.30.

```
virsh # net-start b7005dec-be1a-fe9a-338a-0cb1301dfcfd
```

Starts the virtual network that has a UUID of "b7005dec-be1a-fe9a-338a-0cb1301dfcfd".

Example 2.31.

```
virsh # net-start --network b7005dec-be1a-fe9a-338a-0cb1301dfcfd
```

Same as the above example.

Example in context

Starting with an XML file we've already created, using the [required XML format](#)⁸:

```
<network>
  <name>examplenetwork</name>
  <bridge name="virbr100" />
  <forward mode="route" />
  <ip address="10.10.120.1" netmask="255.255.255.0" />
</network>
```

```
# ls -al /root/examplenetwork.xml
-rw-r--r--. 1 root root 162 Nov  7 16:43 /root/examplenetwork.xml
```

We start virsh interactively, then define a *persistent* virtual network using the XML file:

```
# virsh
Welcome to virsh, the virtualization interactive terminal.

Type: 'help' for help with commands
      'quit' to quit
```

```
virsh # net-list
Name           State      Autostart
-----
default        active     yes
```

```
virsh # net-define /root/examplenetwork.xml
Network examplenetwork defined from /root/examplenetwork.xml
```

Defined. Now we confirm:

```
virsh # net-list --all
Name           State      Autostart
-----
default        active     yes
```

⁸ <http://libvirt.org/formatnetwork.html>

```
examplenetwork    inactive    no    <-- new persistent networks start out inactive
```

Newly defined virtual networks aren't automatically started, so we manually start it now:

```
virsh # net-start examplenetwork    <-- this is net-start in action
Network examplenetwork started
```

```
virsh # net-list
Name                State      Autostart
-----
default             active     yes
examplenetwork      active    no        <-- the persistent network is now running
(active)
```

We check the details of the started network from virsh, using [net-dumpxml](#). This shows us the name of the bridge network interface.

```
virsh # net-dumpxml examplenetwork
<network>
  <name>examplenetwork</name>
  <uuid>b7005dec-be1a-fe9a-338a-0cb1301dfcfd</uuid>
  <forward mode='route' />
  <bridge name='virbr100' stp='on' delay='0' />    <-- the "virbr100" here
  <ip address='10.10.120.1' netmask='255.255.255.0'>
  </ip>
</network>
```

If the virtualisation server is running Linux, we can check how the bridge interface appears to the host OS:

```
# ifconfig virbr100
virbr100  Link encap:Ethernet  HWaddr A6:45:97:AE:8E:08
          inet addr:10.10.120.1  Bcast:10.10.120.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:2653 (2.5 KiB)
```

See also

- [net-define](#) - Adds a new **persistent** virtual network to libvirt, without starting it, using settings from an XML file.
- [net-dumpxml](#) - Outputs the XML configuration for a virtual network, to stdout.
- [net-list](#) - Displays a list of the virtual networks libvirt is aware of.

2.10. net-undefine

Removes an inactive virtual network from the libvirt configuration

Usage

```
net-undefine --network network-identifier
```

Options

Table 2.10. Options

Name	Required?	Description
<code>--network network-identifier</code>	Required	The name or UUID of the virtual network to remove. The word " <code>--network</code> " itself is optional.

Availability

Available from libvirt 0.2.0 onwards

Platform or Hypervisor specific notes

None yet

Examples

Example 2.32.

```
virsh # net-undefine examplenetwork
```

Undefines the virtual network named "*examplenetwork*".

Example 2.33.

```
virsh # net-undefine --network examplenetwork
```

Same as the above example.

Example 2.34.

```
virsh # net-undefine b7005dec-be1a-fe9a-338a-0cb1301dfcfd
```

Undefines the virtual network having a UUID of "*b7005dec-be1a-fe9a-338a-0cb1301dfcfd*".

Example 2.35.

```
virsh # net-undefine --network b7005dec-be1a-fe9a-338a-0cb1301dfcfd
```

Same as the above example.

Example in context

Starting with a virtual network named *examplenetwork*, already running on a virtualisation host server:

```
virsh # net-list
Name           State      Autostart
-----
default        active    yes
examplenetwork active    yes
```

The virtual network is running (active), so we need to shut it down before removing it. We use the `net-destroy` command to shut it down:

```
# net-destroy examplenetwork
Network examplenetwork destroyed
```

Then remove it using `net-undefine`:

```
virsh # net-undefine examplenetwork      <-- this is net-undefine in action
Network examplenetwork has been undefined
```

Done. The `net-list` command no longer shows it listed:

```
virsh # net-list --all
Name           State      Autostart
-----
default        active    yes
```

See also

- `net-define` - Adds a new **persistent** virtual network to libvirt, without starting it, using settings from an XML file.
- `net-destroy` - Shuts down a running virtual network, as started with `net-create` or `net-start`.
- `net-list` - Displays a list of the virtual networks libvirt is aware of.

2.11. net-uuid

When given a network name, returns its corresponding UUID

Usage

```
net-uuid --network network-name
```

Options

Table 2.11. Options

Name	Required?	Description
<code>--network network-name</code>	Required	The name of the virtual network you want the UUID for. The word " <code>--network</code> " itself is optional.

Availability

Available from libvirt 0.2.0 onwards

Chapter 2. Commands

Platform or Hypervisor specific notes

None yet

Examples

Example 2.36.

```
virsh # net-uuid mynetwork
```

Returns the UUID of the virtual network named "mynetwork".

Example 2.37.

```
virsh # net-uuid --network mynetwork
```

Same as the above example.

Example in context

Given a virtual network name, we can get its UUID:

```
virsh # net-uuid examplenetwork
bfbc4c69-7d6a-cc9a-904c-09910ce179c0
```

We can confirm by using the [net-dumpxml](#) command on the returned network UUID:

```
virsh # net-dumpxml bfbc4c69-7d6a-cc9a-904c-09910ce179c0
<network>
  <name>examplenetwork</name>          <-- the name is here
  <uuid>b7005dec-be1a-fe9a-338a-0cb1301dfcfd</uuid> <-- the UUID is here
  <forward mode='route' />
  <bridge name='virbr100' stp='on' delay='1' />
  <ip address='10.10.120.1' netmask='255.255.255.0'>
  </ip>
</network>
```

Using **net-uuid** is more efficient than dumping the XML for the virtual network and manually extracting the *uuid* value.

See also

- [net-dumpxml](#) - Outputs the XML configuration for a virtual network, to stdout
- [net-list](#) - Lists the virtual networks livirt is aware of

Appendix A. Revision History

Revision 0-0 Wed Nov 10 2010

Justin Clift jclift@redhat.com

Initial content added, covering the Virtual Networking commands.

Index

F

feedback1

contact information for this brand, vii

