NAME

virt-xml − Edit libvirt XML using command line options.

SYNOPSIS

virt−xml DOMAIN XML−ACTION XML−OPTION [OUTPUT−OPTION] [MISC−OPTIONS] ...

DESCRIPTION

virt−xml is a command line tool for editing libvirt XML using explicit command line options. See the EXAMPLES section at the end of this document to jump right in.

Each **virt−xml** invocation requires 3 things: name of an existing domain to alter (or XML passed on stdin), an action to on the XML, and an XML change to make. actions are one of:

- **−−add−device**: Append a new device definition to the XML
- **−−remove−device**: Remove an existing device definition
- **−−edit**: Edit an existing XML block
- **−−build−xml**: Just build the requested XML block and print it. No domain or input are required here, but it's recommended to provide them, so virt−xml can fill in optimal defaults.

An XML change is one instance of any of the XML options provided by virt−xml, for example −−disk or −−boot.

virt−xml only allows one action and XML pair per invocation. If you need to make multiple edits, invoke the command multiple times.

OPTIONS

−c −−connect URI

Connect to a non−default hypervisor. See virt−install(1) for details

domain

domain is the name, UUID, or ID of the existing VM. This can be omitted if using −−build−xml, or if XML is passed on stdin.

When a domain is specified, the default output action is −−define, even if the VM is running. To update the running VM configuration, add the −−update option (but not all options/devices support updating the running VM configuration).

If XML is passed on stdin, the default output is −−print−xml.

XML ACTIONS

−−edit [EDIT−OPTIONS]

Edit the specified XML block. EDIT−OPTIONS tell **virt−xml** which block to edit. The type of XML that we are editing is decided by XML option that is passed to **virt−xml** . So if −−disk is passed, EDIT−OPTIONS select which <disk> block to edit.

Certain XML options only ever map to a single XML block, like −−cpu, −−security, −−boot, −−clock, and a few others. In those cases, **virt−xml** will not complain if a corresponding XML block does not already exist, it will create it for you.

Most XML options support a special value 'clearxml=yes'. When combined with −−edit, it will completely blank out the XML block being edited before applying the requested changes. This allows completely rebuilding an XML block. See EXAMPLES for some usage.

EDIT−OPTIONS examples:

•

−−edit −−edit without any options implies 'edit the first block'. So '−−edit −−disk DISK−OPTIONS' means 'edit the first <disk>'.

For the single XML block options mentioned above, plain '−−edit' without any options is what you always want to use.

- **−−edit #**
  Select the specified XML block number. So '−−edit 2 −−disk DISK−OPTS' means 'edit the second <disk>'. This option only really applies for device XML.

- **−−edit all**
  Modify every XML block of the XML option type. So '−−edit all −−disk DISK−OPTS' means 'edit ever <disk> block'. This option only really applies for device XML.

- **−−edit DEVICE−OPTIONS**
  Modify every XML block that matches the passed device options. The device options are in the same format as would be passed to the XML option.

So *−−edit path=/tmp/foo −−disk DISK−OPTS* means 'edit every <disk> with path /tmp/foo'. This option only really applies for device XML.

−−add−device
Append the specified XML options to the XML <devices> list. Example: '−−add−device −−disk DISK−OPTIONS' will create a new <disk> block and add it to the XML.

This option will error if specified with a non−device XML option (see −−edit section for a partial list).

−−remove−device
Remove the specified device from the XML. The device to remove is chosen by the XML option, which takes arguments in the same format as −−edit. Examples:

- **−−remove−device −−disk 2**
  Remove the second disk device

- **−−remove−device −−network all**
  Remove all network devices

- **−−remove−device −−sound pcspk**
  Remove all sound devices with model='pcspk'

This option will error if specified with a non−device XML option (see −−edit isection for a partial list).

−−build−xml
Just build the specified XML, and print it to stdout. No input domain or input XML is required. Example: '−−build−xml −−disk DISK−OPTIONS' will just print the new <disk> device.

However if the generated XML is targeted for a specific domain, it's recommended to pass it to virt−xml, so the tool can set optimal defaults.

This option will error if specified with an XML option that does not map cleanly to a specific XML block, like −−vcpus or −−memory.

OUTPUT OPTIONS

These options decide what action to take after altering the XML. In the common case these do not need to be specified, as 'XML actions' will imply a default output action, described in detail above. These are only needed if you want to modify the default output.

−−update

If the specified domain is running, attempt to alter the running VM configuration. If combined with −−edit, this is an update operation. If combined with −−add−device, this is a device hotplug. If combined with −−remove−device, this is a device hotunplug.

Keep in mind, most XML properties and devices do not support live update operations, so don't expect it to succeed in all cases.

By default this also implies **−−define**.

−−define

Define the requested XML change. This is typically the default if no output option is specified, but if a −−print option is specified, −−define is required to force the change.

−−no−define

Explicitly do not define the XML. For example if you only want to alter the runtime state of a VM, combine this with **−−update**.

−−start

Start the VM after performing the requested changes. If combined with −−no−define, this will create transient VM boot with the requested changes.

−−print−diff

Print the generated XML change in unified diff format. If only this output option is specified, all other output options are disabled and no persistent change is made.

−−print−xml

Print the generated XML in its entirety. If only this output option is specified, all other output options are disabled and no persistent change is made.

−−confirm

Before defining or updating the domain, show the generated XML diff and interactively request confirmation.

GUEST OS OPTIONS

−−os−variant, −−osinfo OS_VARIANT

Optimize the guest configuration for a specific operating system (ex. 'fedora29', 'rhel7', 'win10'). While not required, specifying this options is HIGHLY RECOMMENDED, as it can greatly increase performance by specifying virtio among other guest tweaks.

If the guest has been installed using virt−manager version 2.0.0 or newer, providing this information should not be necessary, as the OS variant will have been stored in the guest configuration during installation and virt−xml will retrieve it from there automatically.

Use the command **virt−xml −−osinfo list** to get the list of the accepted OS variants. See **osinfo−query os** for even more output.

See virt−install(1) documentation for more details about **−−os−variant/−−osinfo**

XML OPTIONS

- **−−disk**

- **−−network**
- **−−graphics**
- **−−metadata**
- **−−memory**
- **−−vcpus**
- **−−cpu**
- **−−iothreads**
- **−−seclabel**
- **−−keywrap**
- **−−cputune**
- **−−numatune**
- **−−memtune**
- **−−blkiotune**
- **−−memorybacking**
- **−−features**
- **−−clock**
- **−−pm**
- **−−events**
- **−−resources**
- **−−sysinfo**
- **−−xml**
- **−−qemu−commandline**
- **−−launchSecurity**
- **−−boot**
- **−−idmap**
- **−−controller**
- **−−input**
- **−−serial**
- **−−parallel**
- **−−channel**
- **−−console**
- **−−hostdev**
- **−−filesystem**
- **−−sound**
- **−−audio**
- **−−watchdog**
- **−−video**
- **−−smartcard**

- **−−redirdev**
- **−−memballoon**
- **−−tpm**
- **−−rng**
- **−−panic**
- **−−shmem**
- **−−memdev**

These options alter the XML for a single class of XML elements. More complete documentation is found in virt−install(1).

Generally these options map pretty straightforwardly to the libvirt XML, documented at *https://libvirt.org/formatdomain.html*

Option strings are in the format of: −−option opt=val,opt2=val2,... example: −−disk path=/tmp/foo,shareable=on. Properties can be used with '−−option opt=,', so to clear a disks cache setting you could use '−−disk cache=,'

For any option, use −−option=? to see a list of all available sub options, example: −−disk=? or −−boot=?

−−help output also lists a few general examples. See the EXAMPLES section below for some common examples.

virt−xml specifically has some operations that don't really apply to virt−install Examples:

−−boot refresh−machine−type=yes
Refresh the XML **<os><type machine=X></os>** value to the latest one that qemu provides. For example, if your VM has a machine type value **pc−q35−4.0**, this will reset the value to **q35**, and works similarly with other versioned machine types. Occasionally this is necessary to get enable qemu bug fixes, or when qemu deprecates and removes old machine type values.

MISCELLANEOUS OPTIONS

−h, −−help
Show the help message and exit

−−version
Show program's version number and exit

−q, −−quiet
Avoid verbose output.

−d, −−debug
Print debugging information

EXAMPLES

See a list of all suboptions that −−disk and −−network take

```
# virt−xml −−disk=? −−network=?
```

Change the <description> of domain 'EXAMPLE':

```
# virt−xml EXAMPLE −−edit −−metadata description="my new description"
```

Enable the boot device menu for domain 'EXAMPLE':

```
# virt−xml EXAMPLE −−edit −−boot menu=on
```

Clear the previous <cpu> definition of domain 'winxp', change it to 'host−model', but interactively confirm the diff before saving:

```
# virt−xml winxp −−edit −−cpu host−model,clearxml=yes −−confirm
```

Change the second sound card to model=ich6 on 'fedora19', but only output the diff:

```
# virt−xml fedora19 −−edit 2 −−sound model=ich6 −−print−diff
```

Update the every graphics device password to 'foo' of the running VM 'rhel6':

```
# virt−xml rhel6 −−edit all −−graphics password=foo −−update
```

Remove the disk path from disk device hdc:

```
# virt−xml rhel6 −−edit target=hdc −−disk path=
```

Change all disk devices of type 'disk' to use cache=none, using XML from stdin, printing the new XML to stdout.

```
# cat <xmlfile> | virt−xml −−edit device=disk −−disk cache=none
```

Change disk 'hda' IO to native and use startup policy as 'optional'.

```
# virt−xml fedora20 −−edit target=hda \
           −−disk io=native,startup_policy=optional
```

Change all host devices to use driver_name=vfio for VM 'fedora20' on the remote connection

```
# virt−xml −−connect qemu+ssh://remotehost/system \
           fedora20 −−edit all −−hostdev driver_name=vfio
```

Hotplug host USB device 001.003 to running domain 'fedora19':

```
# virt−xml fedora19 −−update −−add−device −−hostdev 001.003
```

Add a spicevmc channel to the domain 'winxp', that will be available after the next VM shutdown.

```
# virt−xml winxp −−add−device −−channel spicevmc
```

Create a 10G qcow2 disk image and attach it to 'fedora18' for the next VM startup:

```
# virt−xml fedora18 −−add−device \
  −−disk /var/lib/libvirt/images/newimage.qcow2,format=qcow2,size=10
```

Same as above, but ensure the disk is attached to the most appropriate bus for the guest OS by providing information about it on the command line:

```
# virt−xml fedora18 −−osinfo fedora18 −−add−device \
  −−disk /var/lib/libvirt/images/newimage.qcow2,format=qcow2,size=10
```

Hotunplug the disk vdb from the running domain 'rhel7':

```
# virt−xml rhel7 −−update −−remove−device −−disk target=vdb
```

Remove all graphics devices from the VM 'rhel7' after the next shutdown:

```
# virt-xml rhel7 --remove-device --graphics all
```

Generate XML for a virtio console device and print it to stdout:

```
# virt-xml --build-xml --console pty,target_type=virtio
```

Add qemu command line passthrough:

```
# virt-xml f25 --edit --confirm --qemu-commandline="-device FOO"
```

Use boot device 'network' for a single transient boot:

```
# virt-xml myvm --no-define --start --edit --boot network
```

CAVEATS

Virtualization hosts supported by libvirt may not permit all changes that might seem possible. Some edits made to a VM's definition may be ignored. For instance, QEMU does not allow the removal of certain devices once they've been defined.

BUGS

Please see *https://virt-manager.org/bugs*

COPYRIGHT

SEE ALSO

virt-install(1), the project website *https://virt-manager.org*