

NAME

Sys::Virt::Network – Represent & manage a libvirt virtual network

DESCRIPTION

The `Sys::Virt::Network` module represents a virtual network managed by the virtual machine monitor.

METHODS

`my $uuid = $net->get_uuid()`

Returns a 16 byte long string containing the raw globally unique identifier (UUID) for the network.

`my $uuid = $net->get_uuid_string()`

Returns a printable string representation of the raw UUID, in the format 'XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX'.

`my $name = $net->get_name()`

Returns a string with a locally unique name of the network

`$net->is_active()`

Returns a true value if the network is currently running

`$net->is_persistent()`

Returns a true value if the network has a persistent configuration file defined

`my $xml = $net->get_xml_description()`

Returns an XML document containing a complete description of the network's configuration

`$net->create()`

Start a network whose configuration was previously defined using the `define_network` method in `Sys::Virt`.

`$net->undefine()`

Remove the configuration associated with a network previously defined with the `define_network` method in `Sys::Virt`. If the network is running, you probably want to use the `shutdown` or `destroy` methods instead.

`$net->destroy()`

Immediately terminate the machine, and remove it from the virtual machine monitor. The `$net` handle is invalid after this call completes and should not be used again.

`$net->update($command, $section, $parentIndex, $xml, $flags=0)`

Update the network configuration with `$xml`. The `$section` parameter, which must be one of the XML SECTION CONSTANTS listed later, indicates what schema is used in `$xml`. The `$command` parameter determines what action is taken. Finally, the `$flags` parameter can be used to control which config is affected.

`$net->get_bridge_name()`

Return the name of the bridge device associated with the virtual network

`$flag = $net->get_autostart();`

Return a true value if the virtual network is configured to automatically start upon boot. Return false, otherwise

`$net->set_autostart($flag)`

Set the state of the autostart flag, which determines whether the virtual network will automatically start upon boot of the host OS.

`@leases = $net->get_dhcp_leases($mac=undef, $flags=0)`

Get a list of all active DHCP leases. If `$mac` is undefined then leases for all VMs are returned, otherwise only leases for the matching MAC address are returned. The `$flags` parameter is currently unused and defaults to zero.

The elements in the returned array are hash references with the following fields

`iface`
Network interface name

`expirytime`
Seconds since the epoch until the lease expires

`type`
One of the Sys::Virt IP address type constants

`mac`
The MAC address of the lease

`iaid`
The IAID of the client

`ipaddr`
The IP address of the lease

`prefix`
The IP address prefix

`hostname`
The optional hostname associated with the lease

`clientid`
The client ID or DUID

`@port = $net->list_all_ports($flags=0)`
List all ports associated with the network. The return array elements are instances of the Sys::Virt::NetworkPort class.

`$port = $net->create_port($xml, $flags=0)`
Create a new network port from the given \$xml description. The \$flags parameter can optionally taken one or more of the network port creation constants. The returned \$port object is an instance of the Sys::Virt::NetworkPort class.

`$port = $net->get_port_by_uuid($uuid);`
Lookup a network port from a raw or printable UUID. The returned \$port object is an instance of the Sys::Virt::NetworkPort class.

CONSTANTS

This section documents constants that are used with various APIs described above

NETWORK CREATION CONSTANTS

When creating networks zero or more of the following constants may be used

Sys::Virt::Network::CREATE_VALIDATE
Validate the XML document against the XML schema

LIST FILTERING

The following constants are used to filter object lists

Sys::Virt::Network::LIST_ACTIVE
Include networks which are active

Sys::Virt::Network::LIST_INACTIVE
Include networks which are not active

Sys::Virt::Network::LIST_AUTOSTART
Include networks which are set to autostart

Sys::Virt::Network::LIST_NO_AUTOSTART
Include networks which are not set to autostart

Sys::Virt::Network::LIST_PERSISTENT
Include networks which are persistent

Sys::Virt::Network::LIST_TRANSIENT
Include networks which are transient

NETWORK DEFINE

The following constants can be used to control the behaviour of network define operations

Sys::Virt::Network::DEFINE_VALIDATE
Validate the XML document against the XML schema

XML CONSTANTS

The following constants are used when querying XML

Sys::Virt::Network::XML_INACTIVE
Request the inactive XML, instead of the current possibly live config.

XML SECTION CONSTANTS

The following constants are used to refer to sections of the XML document

Sys::Virt::Network::SECTION_BRIDGE
The bridge device element

Sys::Virt::Network::SECTION_DNS_HOST
The DNS host record section

Sys::Virt::Network::SECTION_DNS_SRV
The DNS SRV record section

Sys::Virt::Network::SECTION_DNS_TXT
The DNS TXT record section

Sys::Virt::Network::SECTION_DOMAIN
The domain name section

Sys::Virt::Network::SECTION_FORWARD
The forward device section

Sys::Virt::Network::SECTION_FORWARD_INTERFACE
The forward interface section

Sys::Virt::Network::SECTION_FORWARD_PF
The forward physical function section

Sys::Virt::Network::SECTION_IP
The IP address section

Sys::Virt::Network::SECTION_IP_DHCP_HOST
The IP address DHCP host section

Sys::Virt::Network::SECTION_IP_DHCP_RANGE
The IP address DHCP range section

Sys::Virt::Network::SECTION_PORTGROUP
The port group section

Sys::Virt::Network::SECTION_NONE
The top level domain element

XML UPDATE FLAGS

Sys::Virt::Network::UPDATE_AFFECT_CURRENT
Affect whatever the current object state is

Sys::Virt::Network::UPDATE_AFFECT_CONFIG
Always update the config file

Sys::Virt::Network::UPDATE_AFFECT_LIVE

Always update the live config

XML UPDATE COMMANDS

Sys::Virt::Network::UPDATE_COMMAND_NONE

No update

Sys::Virt::Network::UPDATE_COMMAND_DELETE

Remove the matching entry

Sys::Virt::Network::UPDATE_COMMAND_MODIFY

Modify the matching entry

Sys::Virt::Network::UPDATE_COMMAND_ADD_FIRST

Insert the matching entry at the start

Sys::Virt::Network::UPDATE_COMMAND_ADD_LAST

Insert the matching entry at the end

EVENT ID CONSTANTS

Sys::Virt::Network::EVENT_ID_LIFECYCLE

Network lifecycle events

LIFECYCLE CHANGE EVENTS

The following constants allow network lifecycle change events to be interpreted. The events contain both a state change, and a reason though the reason is currently unused.

Sys::Virt::Network::EVENT_DEFINED

Indicates that a persistent configuration has been defined for the network.

Sys::Virt::Network::EVENT_STARTED

The network has started running

Sys::Virt::Network::EVENT_STOPPED

The network has stopped running

Sys::Virt::Network::EVENT_UNDEFINED

The persistent configuration has gone away

AUTHORS

Daniel P. Berrange <berrange@redhat.com>

COPYRIGHT

Copyright (C) 2006 Red Hat Copyright (C) 2006–2007 Daniel P. Berrange

LICENSE

This program is free software; you can redistribute it and/or modify it under the terms of either 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), or, the Artistic License, as specified in the Perl README file.

SEE ALSO

Sys::Virt, Sys::Virt::Error, <http://libvirt.org>