NAME

Sys::Virt::NodeDevice − Represent & manage a libvirt storage pool

DESCRIPTION

The `Sys::Virt::NodeDevice` module represents a storage pool managed by libvirt. There are a variety of storage pool implementations for LVM, Local directories/filesystems, network filesystems, disk partitioning, iSCSI, and SCSI.

METHODS

my `$name` = `$dev`−>**get_name()**

Returns a string with a locally unique name of the device

my `$parentname` = `$dev`−>**get_parent()**

Returns a string with a locally unique name of the parent of the device, or undef if there is no parent

my `$xml` = `$dev`−>**get_xml_description()**

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

`$dev`−>**reattach()**

Rebind the node device to the host OS device drivers.

`$dev`−>**dettach()**

Unbind the node device from the host OS device driver

`$dev`−>**reset()**

Reset the node device. The device must be unbound from the host OS drivers for this to work

`$dev`−>create($flags=0)

Start a node device whose configuration was previously defined using the `define_node_device` method in Sys::Virt. The `$flags` parameter is currently unused and defaults to zero.

`$dev`−>undefine($flags=0)

Delete a node device whose configuration was previously defined using the `define_node_device` method in Sys::Virt. The `$flags` parameter is currently unused and defaults to zero.

`$flag` = `$dev`−>**get_autostart()**;

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

`$dev`−>set_autostart($flag)

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

`$dev`−>**is_active()**

Returns a true value if the node device is currently running

`$dev`−>**is_persistent()**

Returns a true value if the node device has a persistent configuration file defined

`$dev`−>**destroy()**

Destroy the virtual device releasing any OS resources associated with it.

my `@caps` = `$dev`−>**list_capabilities()**

Return a list of all capabilities in the device.

CONSTANTS

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

LIST FILTERING

The following constants are used to filter object lists

Sys::Virt::NodeDevice::LIST_CAP_NET

Include devices with the network capability

Sys::Virt::NodeDevice::LIST_CAP_PCI_DEV
Include devices with the PCI device capability

Sys::Virt::NodeDevice::LIST_CAP_SCSI
Include devices with the SCSI capability

Sys::Virt::NodeDevice::LIST_CAP_SCSI_HOST
Include devices with the SCSI host capability

Sys::Virt::NodeDevice::LIST_CAP_SCSI_TARGET
Include devices with the SCSI target capability

Sys::Virt::NodeDevice::LIST_CAP_STORAGE
Include devices with the storage capability

Sys::Virt::NodeDevice::LIST_CAP_SYSTEM
Include devices with the system capability

Sys::Virt::NodeDevice::LIST_CAP_USB_DEV
Include devices with the USB device capability

Sys::Virt::NodeDevice::LIST_CAP_USB_INTERFACE
Include devices with the USB interface capability

Sys::Virt::NodeDevice::LIST_CAP_FC_HOST
Include devices with the fibre channel host capability

Sys::Virt::NodeDevice::LIST_CAP_VPORTS
Include devices with the NPIV vport capability

Sys::Virt::NodeDevice::LIST_CAP_SCSI_GENERIC
Include devices with the SCSI generic capability

Sys::Virt::NodeDevice::LIST_CAP_DRM
Include devices with the DRM capability

Sys::Virt::NodeDevice::LIST_CAP_MDEV
Include mediated devices

Sys::Virt::NodeDevice::LIST_CAP_MDEV_TYPES
Include devices capable of mediated devices

Sys::Virt::NodeDevice::LIST_CAP_CCW_DEV
Include CCW devices

Sys::Virt::NodeDevice::LIST_CAP_CSS_DEV
Include CSS devices

Sys::Virt::NodeDevice::LIST_CAP_VDPA
Include VDPA devices

Sys::Virt::NodeDevice::LIST_CAP_AP_CARD
Include s390 AP card devices

Sys::Virt::NodeDevice::LIST_CAP_AP_MATRIX
Include s390 AP matrix devices

Sys::Virt::NodeDevice::LIST_CAP_AP_QUEUE
Include s390 AP queue devices

Sys::Virt::NodeDevice::LIST_CAP_VPD
Include PCI devices with VPD

Sys::Virt::NodeDevice::LIST_ACTIVE
Include active devices

Sys::Virt::NodeDevice::LIST_INACTIVE
Include inactive devices

EVENT ID CONSTANTS

Sys::Virt::NodeDevice::EVENT_ID_LIFECYCLE
Node device lifecycle events

Sys::Virt::NodeDevice::EVENT_ID_UPDATE
Node device config update events

LIFECYCLE CHANGE EVENTS

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

Sys::Virt::NodeDevice::EVENT_CREATED
Indicates that a device was created

Sys::Virt::NodeDevice::EVENT_DELETED
Indicates that a device has been deleted

Sys::Virt::NodeDevice::EVENT_DEFINED
Indicates that a device configuration has been created

Sys::Virt::NodeDevice::EVENT_UNDEFINED
Indicates that a device configuration has been deleted

AUTHORS

Daniel P. Berrange <berrange@redhat.com>

COPYRIGHT

LICENSE

SEE ALSO

Sys::Virt, Sys::Virt::Error, `http://libvirt.org`