

NAME

Sys::Virt::DomainCheckpoint – Represent & manage a libvirt guest domain checkpoint

DESCRIPTION

The `Sys::Virt::DomainCheckpoint` module represents a guest domain checkpoint managed by the virtual machine monitor.

METHODS

`my $str = $domchkp->get_name()`

Return the name of the checkpoint

`my $xml = $domchkp->get_xml_description($flags)`

Returns an XML document containing a complete description of the domain checkpoints' configuration. The `$flags` parameter accepts the following constants

`Sys::Virt::DomainCheckpoint::XML_SECURE`

Include security sensitive information in the XML dump, such as passwords.

`Sys::Virt::DomainCheckpoint::XML_SIZE`

Include dynamic per-<disk> size information

`Sys::Virt::DomainCheckpoint::XML_NO_DOMAIN`

Suppress <domain> sub-element

`$domchkp->delete($flags)`

Deletes this checkpoint object & its data. The optional `$flags` parameter controls what should be deleted via the `Sys::Virt::DomainCheckpoint::DELETE_*` constants.

`$parentchkp = $domchkp->get_parent();`

Return the parent of the checkpoint, if any

`my @checkpoints = $domchkp->list_all_children($flags)`

Return a list of all domain checkpoints that are children of this checkpoint. The elements in the returned list are instances of the `Sys::Virt::DomainCheckpoint` class. The `$flags` parameter can be used to filter the list of return domain checkpoints.

CONSTANTS**CHECKPOINT CREATION**

The following constants are useful when creating checkpoints

`Sys::Virt::DomainCheckpoint::CREATE_REDEFINE`

Replace/set the metadata with the checkpoint

`Sys::Virt::DomainCheckpoint::CREATE_QUIESCE`

Quiesce the guest disks while taking the checkpoint

`Sys::Virt::DomainCheckpoint::CREATE_REDEFINE_VALIDATE`

Validate disk data state

CHECKPOINT DELETION

The following constants are useful when deleting checkpoints

`Sys::Virt::DomainCheckpoint::DELETE_CHILDREN`

Recursively delete any child checkpoints

`Sys::Virt::DomainCheckpoint::DELETE_CHILDREN_ONLY`

Only delete the child checkpoints

`Sys::Virt::DomainCheckpoint::DELETE_METADATA_ONLY`

Only delete the checkpoint metadata

CHECKPOINT LIST

The following constants are useful when listing checkpoints

Sys::Virt::DomainCheckpoint::LIST_ROOTS

Only list checkpoints which are root nodes in the tree

Sys::Virt::DomainCheckpoint::LIST_DESCENDANTS

Only list checkpoints which are descendants of the current checkpoint

Sys::Virt::DomainCheckpoint::LIST_LEAVES

Only list leaf nodes in the checkpoint tree

Sys::Virt::DomainCheckpoint::LIST_NO_LEAVES

Only list non-leaf nodes in the checkpoint tree

Sys::Virt::DomainCheckpoint::LIST_TOPOLOGICAL

Sort list in topological order wrt to parent/child relationships.

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>