NAME

Sys::Virt::Event − An event loop contract

DESCRIPTION

The `Sys::Virt::Event` module represents the contract for integrating libvirt with an event loop. This package is abstract and intended to be subclassed to provide an actual implementation.

METHODS

register_default()

Register the default libvirt event loop implementation

run_default()

Run a single iteration of the default event loop implementation

register($impl)

Register an event loop implementation. The implementation should be a instance of a sub-class of the `Sys::Virt::Event` package.

`$self`−>_run_handle_callback($watch, `$fd`, `$events`, `$cb`, `$opaque`)

A helper method for executing a callback in response to one of more `$events` on the file handle `$fd`. The `$watch` number is the unique idenifier associated with the file descriptor. The `$cb` and `$opaque` parameters are the callback and data registered for the handle.

`$self`−>_run_timeout_callback($timer, `$cb`, `$opaque`)

A helper method for executing a callback in response to the expiry of a timeout identified by `$timer`. The `$cb` and `$opaque` parameters are the callback and data registered for the timeout.

`$self`−>_free_callback_opaque($ff, `$opaque`)

A helper method for freeing the data associated with a callback. The `$ff` and `$opaque` parameters are the callback and data registered for the handle/timeout.

my `$watch` = Sys::Virt::Event::add_handle($fd, `$events`, `$coderef`)

Adds a watch on the file descriptor `$fd` for the events `$events` which is a mask of the FILE HANDLE EVENTS constants listed later. The`$coderef` parameter is a subroutine to in voke when an event is triggered. The subroutine will be passed three parameters, the watch identifier, the file descriptor and the event mask. This method returns the watch identifier which can be used to update or remove the watch

Sys::Virt::Event::update_handle($watch, `$events`)

Update the event mask for the file descriptor watch `$watch` to use the events `$events`.

Sys::Virt::Event::remove_handle($watch)

Remove the event mask for the file descriptor watch `$watch`.

my `$watch` = Sys::Virt::Event::add_timeout($frequency, `$coderef`)

Adds a timeout to trigger with `$frequency` milliseconds interval. The`$coderef` parameter is a subroutine to invoke when an event is triggered. The subroutine will be passed one parameter, the timer identifier. This method returns the watch identifier which can be used to update or remove the watch

Sys::Virt::Event::update_timeout($timer, `$frequency`)

Update the timeout `$timer` to have the frequency `$frequency` milliseconds.

Sys::Virt::Event::remove_timeout($timer)

Remove the timeout `$timer`

CONSTANTS

FILE HANDLE EVENTS

When integrating with an event loop the following constants define the file descriptor events

Sys::Virt::Event::HANDLE_READABLE

The file descriptor has data available for read without blocking

Sys::Virt::Event::HANDLE_WRITABLE
The file descriptor has ability to write data without blocking

Sys::Virt::Event::HANDLE_ERROR
An error occurred on the file descriptor

Sys::Virt::Event::HANDLE_HANGUP
The remote end of the file descriptor closed

AUTHORS

Daniel P. Berrange <berrange@redhat.com>

COPYRIGHT

LICENSE

SEE ALSO

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