XenEnterprise Management API Draft

Version: API Revision 1.0.9
Date: 7th June 2007
XenSource Confidential

Ewan Mellor: ewan@xensource.com
Richard Sharp: richard.sharp@xensource.com
David Scott: david.scott@xensource.com

Contributors:

Stefan Berger, IBM
Daniel Berrangé, Red Hat
Gareth Bestor, IBM
Hollis Blanchard, IBM
Mike Day, IBM
Jim Fehlig, Novell
Jon Harrop, XenSource
Vincent Hanquez, XenSource
John Levon, Sun Microsystems
Jon Ludlam, XenSource
Alastair Tse, XenSource
Daniel Veillard, Red Hat
Tom Wilkie, University of Cambridge

Contents

Chapter 1

Introduction

This document defines the XenEnterprise Management API—an API for remotely configuring and controlling virtualised guests running on a Xen-enabled cluster.

This document is a preliminary draft provided for the XenSource Technology Partners' review and preliminary implementation.

The API is presented here as a set of Remote Procedure Calls, with a wire format based upon XML-RPC. No specific language bindings are prescribed, although examples will be given in the python programming language.
Although we adopt some terminology from object-oriented programming, future client language bindings may or may not be object oriented. The API reference uses the terminology *classes* and *objects*. For our purposes a *class* is simply a hierarchical namespace; an *object* is an instance of a class with its fields set to specific values. Objects are persistent and exist on the server-side. Clients may obtain opaque references to these server-side objects and then access their fields via get/set RPCs.
For each class we specify a list of fields along with their *types* and *qualifiers*. A qualifier is one of:

- RO_{run}: the field is Read Only. Furthermore, its value is automatically computed at runtime. For example: current CPU load and disk IO throughput.
- RO_{ins}: the field must be manually set when a new object is created, but is then Read Only for the duration of the object's life. For example, the maximum memory addressable by a guest is set before the guest boots.
- RW: the field is Read/Write. For example, the name of a VM.

A full list of types is given in Chapter 2. However, there are three types that require explicit mention:

- $t\ Ref$: signifies a reference to an object of type t.
- $t\ Set$: signifies a set containing values of type t.
- $(t_1, t_2)\ Map$: signifies a mapping from values of type t_1 to values of type t_2.

Note that there are a number of cases where *Ref*s are *doubly linked*—e.g. a VM has a field called `VIFs` of type $(VIF\ Ref)\ Set$; this field lists the network interfaces attached to a particular VM. Similarly, the VIF class has a field called `VM` of type $(VM\ Ref)$ which references the VM to which the interface is connected. These two fields are *bound together*, in the sense that creating a new VIF causes the `VIFs` field of the corresponding VM object to be updated automatically.
The API reference explicitly lists the fields that are bound together in this way. It also contains a diagram that shows relationships between classes. In this diagram an edge signifies the existance of a pair of fields that are bound together, using standard crows-foot notation to signify the type of relationship (e.g. one-many, many-many).

1.1 RPCs associated with fields

Each field, `f`, has an RPC accessor associated with it that returns `f`'s value:

- "`get_f(Ref x)`": takes a `Ref` that refers to an object and returns the value of `f`.

Each field, `f`, with attribute *RW* and whose outermost type is *Set* has the following additional RPCs associated with it:

- an "`add_to_f(Ref x, v)`" RPC adds a new element v to the set[1];
- a "`remove_from_f(Ref x, v)`" RPC removes element `v` from the set;

Each field, `f`, with attribute *RW* and whose outermost type is *Map* has the following additional RPCs associated with it:

- an "`add_to_f(Ref x, k, v)`" RPC adds new pair `(k, v)` to the mapping stored in `f` in object `x`. Adding a new pair for duplicate key, `k`, overwrites any previous mapping for `k`.
- a "`remove_from_f(Ref x, k)`" RPC removes the pair with key `k` from the mapping stored in `f` in object `x`.

Each field whose outermost type is neither *Set* nor *Map*, but whose attribute is *RW* has an RPC acessor associated with it that sets its value:

- For *RW* (*R*ead/*W*rite), a "`set_f(Ref x, v)`" RPC function is also provided. This sets field `f` on object `x` to value `v`.

1.2 RPCs associated with classes

- Each class has a *constructor* RPC named "`create`" that takes as parameters all fields marked *RW* and RO_{ins}. The result of this RPC is that a new *persistent* object is created on the server-side with the specified field values.
- Each class has a `get_by_uuid(uuid)` RPC that returns the object of that class that has the specified `uuid`.
- Each class that has a `name_label` field has a "`get_by_name_label(name)`" RPC that returns a set of objects of that class that have the specified `label`.
- Each class has a "`destroy(Ref x)`" RPC that explicitly deletes the persistent object specified by `x` from the system. This is a non-cascading delete – if the object being removed is referenced by another object then the `destroy` call will fail.

1.2.1 Additional RPCs

As well as the RPCs enumerated above, some classes have additional RPCs associated with them. For example, the `VM` class has RPCs for cloning, suspending, starting etc. Such additional RPCs are described explicitly in the API reference.

[1]Since sets cannot contain duplicate values this operation has no action in the case that `v` was already in the set.

1.3 Wire Protocol for Remote API Calls

API calls are sent over a network to a Xen-enabled host using the XML-RPC protocol. In this Section we describe how the higher-level types used in our API Reference are mapped to primitive XML-RPC types.
In our API Reference we specify the signatures of API functions in the following style:

```
(ref_vm Set)   VM.get_all()
```

This specifies that the function with name `VM.get_all` takes no parameters and returns a Set of `ref_vm`s. These types are mapped onto XML-RPC types in a straight-forward manner:

- Floats, Bools, DateTimes and Strings map directly to the XML-RPC `double`, `boolean`, `dateTime.iso8601`, and `string` elements.
- all "`ref_`" types are opaque references, encoded as the XML-RPC's `String` type. Users of the API should not make assumptions about the concrete form of these strings and should not expect them to remain valid after the client's session with the server has terminated.
- fields named "`uuid`" of type "`String`" are mapped to the XML-RPC `String` type. The string itself is the OSF DCE UUID presentation format (as output by `uuidgen`, etc).
- ints are all assumed to be 64-bit in our API and are encoded as a string of decimal digits (rather than using XML-RPC's built-in 32-bit `i4` type).
- values of enum types are encoded as strings. For example, a value of `destroy` of type `on_normal_exit`, would be conveyed as:

```
<value><string>destroy</string></value>
```

- for all our types, `t`, our type `t Set` simply maps to XML-RPC's `Array` type, so for example a value of type `String Set` would be transmitted like this:

```
<array>
  <data>
    <value><string>CX8</string></value>
    <value><string>PSE36</string></value>
    <value><string>FPU</string></value>
  </data>
</array>
```

- for types `k` and `v`, our type `(k, v) Map` maps onto an XML-RPC struct, with the key as the name of the struct. Note that the `(k, v) Map` type is only valid when `k` is a `String`, `Ref`, or `Int`, and in each case the keys of the maps are stringified as above. For example, the `(String, double) Map` containing a the mappings Mike → 2.3 and John → 1.2 would be represented as:

```
<value>
  <struct>
    <member>
      <name>Mike</name>
      <value><double>2.3</double></value>
    </member>
    <member>
```

```
      <name>John</name>
      <value><double>1.2</double></value>
    </member>
  </struct>
</value>
```

- our `Void` type is transmitted as an empty string.

1.3.1 Note on References vs UUIDs

References are opaque types — encoded as XML-RPC strings on the wire — understood only by the particular server which generated them. Servers are free to choose any concrete representation they find convenient; clients should not make any assumptions or attempt to parse the string contents. References are not guaranteed to be permanent identifiers for objects; clients should not assume that references generated during one session are valid for any future session. References do not allow objects to be compared for equality. Two references to the same object are not guaranteed to be textually identical.
UUIDs are intended to be permanent names for objects. They are guaranteed to be in the OSF DCE UUID presentation format (as output by `uuidgen`. Clients may store UUIDs on disk and use them to lookup objects in subsequent sessions with the server. Clients may also test equality on objects by comparing UUID strings.
The API provides mechanisms for translating between UUIDs and opaque references. Each class that contains a UUID field provides:

- A "`get_by_uuid`" method that takes a UUID, u, and returns an opaque reference to the server-side object that has UUID=u;
- A `get_uuid` function (a regular "field getter" RPC) that takes an opaque reference, r, and returns the UUID of the server-side object that is referenced by r.

1.3.2 Return Values/Status Codes

The return value of an RPC call is an XML-RPC `Struct`.

- The first element of the struct is named `Status`; it contains a string value indicating whether the result of the call was a "`Success`" or a "`Failure`".

If `Status` was set to `Success` then the Struct contains a second element named `Value`:

- The element of the struct named `Value` contains the function's return value.

In the case where `Status` is set to `Failure` then the struct contains a second element named `ErrorDescription`:

- The element of the struct named `ErrorDescription` contains an array of string values. The first element of the array is an error code; the remainder of the array are strings representing error parameters relating to that code.

For example, an XML-RPC return value from the `host.get_resident_VMs` function above may look like this:

```
<struct>
  <member>
    <name>Status</name>
    <value>Success</value>
  </member>
```

```
    <member>
       <name>Value</name>
       <value>
         <array>
            <data>
              <value>81547a35-205c-a551-c577-00b982c5fe00</value>
              <value>61c85a22-05da-b8a2-2e55-06b0847da503</value>
              <value>1d401ec4-3c17-35a6-fc79-cee6bd9811fe</value>
            </data>
         </array>
       </value>
    </member>
 </struct>
```

1.4 Making XML-RPC Calls

1.4.1 Transport Layer

The following transport layers are currently supported:

- HTTP/S for remote administration
- HTTP over Unix domain sockets for local administration

1.4.2 Session Layer

The XML-RPC interface is session-based; before you can make arbitrary RPC calls you must login and initiate a session. For example:

```
session_id    session.login_with_password(string uname, string pwd)
```

Where `uname` and `password` refer to your username and password respectively, as defined by the Xen administrator. The `session_id` returned by `session.login_with_password` is passed to subequent RPC calls as an authentication token.
A session can be terminated with the `session.logout` function:

```
void          session.logout(session_id session)
```

1.4.3 Synchronous and Asynchronous invocation

Each method call (apart from methods on "Session" and "Task" objects and "getters" and "setters" derived from fields) can be made either synchronously or asynchronously. A synchronous RPC call blocks until the return value is received; the return value of a synchronous RPC call is exactly as specified in Section 1.3.2.
Only synchronous API calls are listed explicitly in this document. All asynchronous versions are in the special `Async` namespace. For example, synchronous call `VM.clone(...)` (described in Chapter 2) has an asynchronous counterpart, `Async.VM.clone(...)`, that is non-blocking.
Instead of returning its result directly, an asynchronous RPC call returns a `task-id`; this identifier is subsequently used to track the status of a running asynchronous RPC. Note that an asychronous call may fail immediately, before a `task-id` has even been created—to represent this eventuality, the returned `task-id` is wrapped in an XML-RPC struct with a `Status`, `ErrorDescription` and `Value` fields, exactly as specified in Section 1.3.2.
The `task-id` is provided in the `Value` field if `Status` is set to `Success`.
The RPC call

```
(ref_task Set)   Task.get_all(session_id s)
```

returns a set of all task IDs known to the system. The status (including any returned result and error codes) of these tasks can then be queried by accessing the fields of the Task object in the usual way. Note that, in order to get a consistent snapshot of a task's state, it is advisable to call the "get_record" function.

1.5 Example interactive session

This section describes how an interactive session might look, using the python XML-RPC client library.
First, initialise python and import the library `xmlrpclib`:

```
\$ python2.4
...
>>> import xmlrpclib
```

Create a python object referencing the remote server:

```
>>> xen = xmlrpclib.Server("https://localhost:443")
```

Acquire a session reference by logging in with a username and password (error-handling ommitted for brevity; the session reference is returned under the key `'Value'` in the resulting dictionary)

```
>>> session = xen.session.login_with_password("user", "passwd")['Value']
```

When serialised, this call looks like the following:

```
<?xml version='1.0'?>
<methodCall>
  <methodName>session.login_with_password</methodName>
  <params>
    <param>
      <value><string>user</string></value>
    </param>
    <param>
      <value><string>passwd</string></value>
    </param>
  </params>
</methodCall>
```

Next, the user may acquire a list of all the VMs known to the system: (Note the call takes the session reference as the only parameter)

```
>>> all_vms = xen.VM.get_all(session)['Value']
>>> all_vms
['OpaqueRef:1', 'OpaqueRef:2', 'OpaqueRef:3', 'OpaqueRef:4' ]
```

The VM references here have the form `OpaqueRef:X`, though they may not be that simple in the future, and you should treat them as opaque strings. *Templates* are VMs with the `is_a_template` field set to true. We can find the subset of template VMs using a command like the following:

```
>>> all_templates = filter(lambda x: xen.VM.get_is_a_template(session, x)['Value'], all_vms)
```

Once a reference to a VM has been acquired a lifecycle operation may be invoked:

```
>>> xen.VM.start(session, all_templates[0], False, False)
{'Status': 'Failure', 'ErrorDescription': ['VM_IS_TEMPLATE', 'OpaqueRef:X']}
```

In this case the `start` message has been rejected, because the VM is a template, and so an error response has been returned. These high-level errors are returned as structured data (rather than as XML-RPC faults), allowing them to be internationalised.
Rather than querying fields individually, whole *records* may be returned at once. To retrieve the record of a single object as a python dictionary:

```
>>> record = xen.VM.get_record(session, all_templates[0])['Value']
>>> record['power_state']
'Halted'
>>> record['name_label']
'XenSource P2V Server'
```

To retrieve all the VM records in a single call:

```
>>> records = xen.VM.get_all_records(session)['Value']
>>> records.keys()
['OpaqueRef:1', 'OpaqueRef:2', 'OpaqueRef:3', 'OpaqueRef:4' ]
>>> records['OpaqueRef:1']['name_label']
'RHEL 4.1 Autoinstall Template'
```

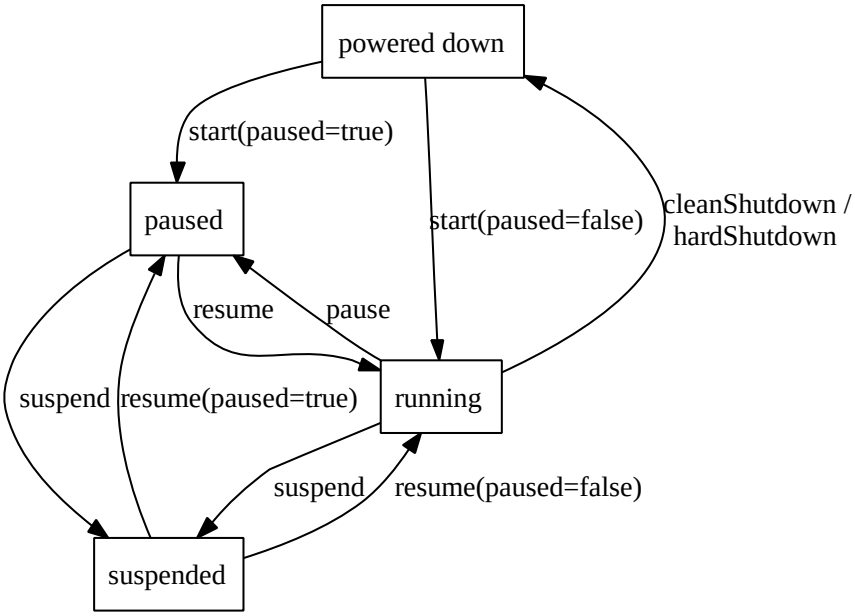


Figure 1.1: VM Lifecycle

1.6 VM Lifecycle

Figure 1.1 shows the states that a VM can be in and the API calls that can be used to move the VM between these states.

1.7 VM boot parameters

The VM class contains a number of fields that control the way in which the VM is booted. With reference to the fields defined in the VM class (see later in this document), this section outlines the boot options available and the mechanisms provided for controlling them.
VM booting is controlled by setting one of the two mutually exclusive groups: "PV", and "HVM". If HVM.boot_policy is the empty string, then paravirtual domain building and booting will be used; otherwise the VM will be loaded as an HVM domain, and booted using an emulated BIOS.
When paravirtual booting is in use, the PV/bootloader field indicates the bootloader to use. It may be "pygrub", in which case the platform's default installation of pygrub will be used, or a full path within the control domain to some other bootloader. The other fields, PV/kernel, PV/ramdisk, PV/args and PV/bootloader_args will be passed to the bootloader unmodified, and interpretation of those fields is then specific to the bootloader itself, including the possibility that the bootloader will ignore some or all of those given values. Finally the paths of all bootable disks are added to the bootloader commandline (a disk is bootable if its VBD has the bootable flag set). There may be zero, one or many bootable disks; the bootloader decides which disk (if any) to boot from.
If the bootloader is pygrub, then the menu.lst is parsed if present in the guest's filesystem, otherwise the specified kernel and ramdisk are used, or an autodetected kernel is used if nothing is specified and autodetection is possible. PV/args is appended to the kernel command line, no matter which mechanism is used for finding the kernel.
If PV/bootloader is empty but PV/kernel is specified, then the kernel and ramdisk values will be treated as paths within the control domain. If both PV/bootloader and PV/kernel are empty, then the behaviour is as if PV/bootloader was specified as "pygrub".
When using HVM booting, HVM/boot_policy and HVM/boot_params specify the boot handling.

Only one policy is currently defined: "BIOS order". In this case, HVM/boot_params should contain one key-value pair "order" = "N" where N is the string that will be passed to QEMU.

Chapter 2

API Reference

2.1 Classes

The following classes are defined:

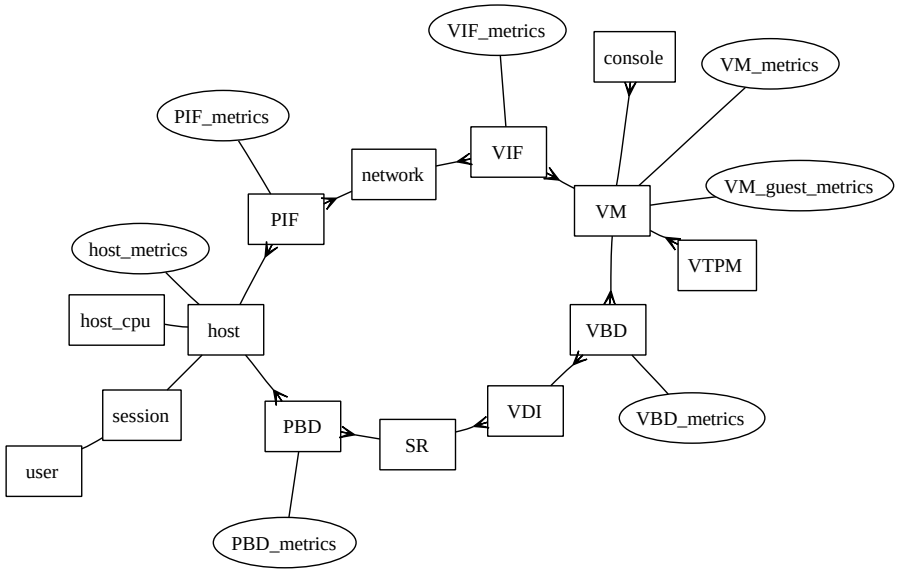
Name	Description
session	A session
task	A long-running asynchronous task
event	Asynchronous event registration and handling
pool	Pool-wide information
VM	A virtual machine (or 'guest')
VM_metrics	The metrics associated with a VM
VM_guest_metrics	The metrics reported by the guest (as opposed to inferred from outside)
host	A physical host
host_crashdump	Represents a host crash dump
host_patch	Represents a patch stored on a server
host_metrics	The metrics associated with a host
host_cpu	A physical CPU
network	A virtual network
VIF	A virtual network interface
VIF_metrics	The metrics associated with a virtual network device
PIF	A physical network interface (note separate VLANs are represented as several PIFs)
PIF_metrics	The metrics associated with a physical network interface
SM	A storage manager plugin
SR	A storage repository
VDI	A virtual disk image
VBD	A virtual block device
VBD_metrics	The metrics associated with a virtual block device
PBD	The physical block devices through which hosts access SRs
crashdump	A VM crashdump
VTPM	A virtual TPM device
console	A console
user	A user of the system
debug	A basic class for testing

2.2 Relationships Between Classes

Fields that are bound together are shown in the following table:

object.field	*object.field*	*relationship*
host.PBDs	PBD.host	many-to-one
SR.PBDs	PBD.SR	many-to-one
VDI.VBDs	VBD.VDI	many-to-one
VDI.crash_dumps	crashdump.VDI	many-to-one
VBD.VM	VM.VBDs	one-to-many
crashdump.VM	VM.crash_dumps	one-to-many
VIF.VM	VM.VIFs	one-to-many
VIF.network	network.VIFs	one-to-many
PIF.host	host.PIFs	one-to-many
PIF.network	network.PIFs	one-to-many
SR.VDIs	VDI.SR	many-to-one
VTPM.VM	VM.VTPMs	one-to-many
console.VM	VM.consoles	one-to-many
host.resident_VMs	VM.resident_on	many-to-one
host.host_CPUs	host_cpu.host	many-to-one
host.crashdumps	host_crashdump.host	many-to-one
host.patches	host_patch.host	many-to-one

The following represents bound fields (as specified above) diagramatically, using crows-foot notation to specify one-to-one, one-to-many or many-to-many relationships:



2.2.1 List of bound fields

2.3 Types

2.3.1 Primitives

The following primitive types are used to specify methods and fields in the API Reference:

Type	Description
String	text strings
Int	64-bit integers
Float	IEEE double-precision floating-point numbers
Bool	boolean
DateTime	date and timestamp
Ref (object name)	reference to an object of class name

2.3.2 Higher order types

The following type constructors are used:

Type	Description
List (t)	an arbitrary-length list of elements of type t
Map (a → b)	a table mapping values of type a to values of type b

2.3.3 Enumeration types

The following enumeration types are used:

enum event_operation	
add	An object has been created
del	An object has been deleted
mod	An object has been modified

enum console_protocol	
vt100	VT100 terminal
rfb	Remote FrameBuffer protocol (as used in VNC)
rdp	Remote Desktop Protocol

enum vbd_operations	
attach	Attempting to attach this VBD to a VM
eject	Attempting to eject the media from this VBD
insert	Attempting to insert new media into this VBD
plug	Attempting to hotplug this VBD
unplug	Attempting to hot unplug this VBD

enum vdi_operations	
scan	Scanning backends for new or deleted VDIs
clone	Cloning the VDI
copy	Copying the VDI
resize	Resizing the VDI
snapshot	Snapshotting the VDI
destroy	Destroying the VDI

force_unlock	Forcibly unlocking the VDI

enum storage_operations	
scan	Scanning backends for new or deleted VDIs
destroy	Destroying the SR
forget	Forgetting about SR
unplug	Unplugging a PBD from this SR
vdi_create	Creating a new VDI
vdi_destroy	Destroying a VDI
vdi_resize	Resizing a VDI
vdi_clone	Cloneing a VDI
vdi_snapshot	Snapshotting a VDI

enum vif_operations	
attach	Attempting to attach this VIF to a VM
plug	Attempting to hotplug this VIF
unplug	Attempting to hot unplug this VIF

enum network_operations	
attaching	Indicates this network is attaching to a VIF or PIF

enum host_allowed_operations	
provision	Indicates this host is able to provision another VM

enum vm_power_state	
Halted	Halted
Paused	Paused
Running	Running
Suspended	Suspended
Unknown	Some other unknown state

enum hello_return
ok

unknown_host
cannot_talk_back

enum task_status_type	
pending	task is in progress
success	task was completed successfully
failure	task has failed
cancelling	task is being cancelled
cancelled	task has been cancelled

enum task_allowed_operations	
cancel	refers to the operation “cancel”

enum vm_operations	
clone	refers to the operation “clone”
copy	refers to the operation “copy”
provision	refers to the operation “provision”
start	refers to the operation “start”
start_on	refers to the operation “start_on”
pause	refers to the operation “pause”
unpause	refers to the operation “unpause”
clean_shutdown	refers to the operation “clean_shutdown”
clean_reboot	refers to the operation “clean_reboot”
hard_shutdown	refers to the operation “hard_shutdown”
power_state_reset	refers to the operation “power_state_reset”
hard_reboot	refers to the operation “hard_reboot”
suspend	refers to the operation “suspend”
resume	refers to the operation “resume”
resume_on	refers to the operation “resume_on”
pool_migrate	refers to the operation “pool_migrate”
migrate	refers to the operation “migrate”
statistics	refers to the operation “statistics”
get_boot_record	refers to the operation “get_boot_record”
send_sysrq	refers to the operation “send_sysrq”
send_trigger	refers to the operation “send_trigger”
changing_memory_live	Changing the memory settings
changing_VCPUs_live	Changing either the VCPUs_number or VCPUs_params
assert_operation_valid	
update_allowed_operations	
make_into_template	Turning this VM into a template
import	importing a VM from a network stream
export	exporting a VM to a network stream
destroy	refers to the act of uninstalling the VM

enum on_normal_exit	
destroy	destroy the VM state
restart	restart the VM

enum on_crash_behaviour	
destroy	destroy the VM state
coredump_and_destroy	record a coredump and then destroy the VM state
restart	restart the VM
coredump_and_restart	record a coredump and then restart the VM
preserve	leave the crashed VM as-is
rename_restart	rename the crashed VM and start a new copy

enum vdi_type	
system	a disk that may be replaced on upgrade
user	a disk that is always preserved on upgrade
ephemeral	a disk that may be reformatted on upgrade
suspend	a disk that stores a suspend image
crashdump	a disk that stores VM crashdump information

enum vbd_mode	
RO	disk is mounted read-only
RW	disk is mounted read-write

enum vbd_type	
CD	VBD will appear to guest as CD
Disk	VBD will appear to guest as disk

2.4 Class: session

2.4.1 Fields for class: session

Name	**session**		
Description	*A session.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RO_{run}	`this_host`	host ref	Currently connected host
RO_{run}	`this_user`	user ref	Currently connected user
RO_{run}	`last_active`	datetime	Timestamp for last time session was active
RO_{run}	`pool`	bool	True if this session relates to a intra-pool login, false otherwise

2.4.2 RPCs associated with class: session

RPC name: login_with_password

Overview: Attempt to authenticate the user, returning a session_id if successful.
Signature:

```
(session ref) login_with_password (string uname, string pwd)
```

Arguments:

type	name	description
`string`	uname	Username for login.
`string`	pwd	Password for login.

Return Type: `session ref`
ID of newly created session

RPC name: logout

Overview: Log out of a session.
Signature:

```
void logout (session_id s)
```

Return Type: `void`

RPC name: change_password

Overview: Change the account password.
Signature:

```
void change_password (session_id s, string old_pwd, string new_pwd)
```

Arguments:

type	name	description
string	old_pwd	Old password for account
string	new_pwd	New password for account

Return Type: void

RPC name: slave_login

Overview: Attempt to authenticate to the pool master by presenting the slave's host ref and pool secret.
Signature:

```
(session ref) slave_login (host ref host, string psecret)
```

Arguments:

type	name	description
host ref	host	Host id of slave
string	psecret	Pool secret

Return Type: session ref
ID of newly created session

RPC name: get_uuid

Overview: Get the uuid field of the given session.
Signature:

```
string get_uuid (session_id s, session ref self)
```

Arguments:

type	name	description
session ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_this_host

Overview: Get the this_host field of the given session.
Signature:

```
(host ref) get_this_host (session_id s, session ref self)
```

Arguments:

type	name	description
session ref	self	reference to the object

Return Type: host ref
value of the field

RPC name: get_this_user

Overview: Get the this_user field of the given session.
Signature:

```
(user ref) get_this_user (session_id s, session ref self)
```

Arguments:

type	name	description
session ref	self	reference to the object

Return Type: user ref
value of the field

RPC name: get_last_active

Overview: Get the last_active field of the given session.
Signature:

```
datetime get_last_active (session_id s, session ref self)
```

Arguments:

type	name	description
session ref	self	reference to the object

Return Type: datetime
value of the field

RPC name: get_pool

Overview: Get the pool field of the given session.
Signature:

```
bool get_pool (session_id s, session ref self)
```

Arguments:

type	name	description
session ref	self	reference to the object

Return Type: bool
value of the field

RPC name: get_by_uuid

Overview: Get a reference to the session instance with the specified UUID.
Signature:

```
(session ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
`string`	uuid	UUID of object to return

Return Type: `session ref`
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given session.
Signature:

```
(session record) get_record (session_id s, session ref self)
```

Arguments:

type	name	description
`session ref`	self	reference to the object

Return Type: `session record`
all fields from the object

2.5 Class: task

2.5.1 Fields for class: task

Name	**task**		
Description	*A long-running asynchronous task.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RO_{run}	`name/label`	string	a human-readable name
RO_{run}	`name/description`	string	a notes field containg human-readable description
RO_{run}	`allowed_operations`	(task_allowed_operations) Set	list of the operations allowed in this state
RO_{run}	`current_operations`	(string → task_allowed_operations) Map	Map of task reference to current operation enumeration
RO_{run}	`created`	datetime	Time task was created
RO_{run}	`finished`	datetime	Time task finished (i.e. succeeded or failed). If task-status is pending, then the value of this field has no meaning
RO_{run}	`status`	task_status_type	current status of the task
RO_{run}	`resident_on`	host ref	the host on which the task is running
RO_{run}	`progress`	float	if the task is still pending, this field contains the estimated percentage complete (0.-1.). If task has completed (successfully or unsuccessfully) this should be 1.
RO_{run}	`type`	string	if the task has completed successfully, this field contains the type of the encoded result (i.e. name of the class whose reference is in the result field). Undefined otherwise.
RO_{run}	`result`	string	if the task has completed successfully, this field contains the result value (either Void or an object reference). Undefined otherwise.
RO_{run}	`error_info`	string Set	if the task has failed, this field contains the set of associated error strings. Undefined otherwise.

2.5.2 RPCs associated with class: task

RPC name: create

Overview: Create a new task object which must be manually destroyed.
Signature:

```
(task ref) create (session_id s, string label, string description)
```

Arguments:

type	name	description
`string`	label	short label for the new task
`string`	description	longer description for the new task

Return Type: `task ref`

The reference of the created task object

RPC name: destroy

Overview: Destroy the task object.
Signature:

```
void destroy (session_id s, task ref self)
```

Arguments:

type	name	description
task ref	self	Reference to the task object

Return Type: void

RPC name: cancel

Overview: Cancel this task. If task.allowed_operations does not contain Cancel, then this will fail with OPERATION_NOT_ALLOWED. The task will show the status 'cancelling', and you should continue to check its status until it shows 'cancelled'. There is no guarantee as to the time within which this task will be cancelled.
Signature:

```
void cancel (session_id s, task ref task)
```

Arguments:

type	name	description
task ref	task	The task

Return Type: void

Possible Error Codes: OPERATION_NOT_ALLOWED

RPC name: get_all

Overview: Return a list of all the tasks known to the system.
Signature:

```
((task ref) Set) get_all (session_id s)
```

Return Type: (task ref) Set
references to all objects

RPC name: get_all_records_where

Overview: Return a list of all the tasks known to the system.
Signature:

```
((task ref -> task record) Map) get_all_records_where (session_id s, string expr)
```

Arguments:

type	name	description
`string`	expr	expression representing records to fetch

Return Type: `(task ref → task record) Map`
records of all matching objects

RPC name: get_uuid

Overview: Get the uuid field of the given task.
Signature:

```
string get_uuid (session_id s, task ref self)
```

Arguments:

type	name	description
`task ref`	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_name_label

Overview: Get the name/label field of the given task.
Signature:

```
string get_name_label (session_id s, task ref self)
```

Arguments:

type	name	description
`task ref`	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_name_description

Overview: Get the name/description field of the given task.
Signature:

```
string get_name_description (session_id s, task ref self)
```

Arguments:

type	name	description
`task ref`	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_allowed_operations

Overview: Get the allowed_operations field of the given task.
Signature:

```
((task_allowed_operations) Set) get_allowed_operations (session_id s, task ref self)
```

Arguments:

type	name	description
task ref	self	reference to the object

Return Type: (task_allowed_operations) Set
value of the field

RPC name: get_current_operations

Overview: Get the current_operations field of the given task.
Signature:

```
((string -> task_allowed_operations) Map) get_current_operations (session_id s, task ref self)
```

Arguments:

type	name	description
task ref	self	reference to the object

Return Type: (string → task_allowed_operations) Map
value of the field

RPC name: get_created

Overview: Get the created field of the given task.
Signature:

```
datetime get_created (session_id s, task ref self)
```

Arguments:

type	name	description
task ref	self	reference to the object

Return Type: datetime
value of the field

RPC name: get_finished

Overview: Get the finished field of the given task.
Signature:

```
datetime get_finished (session_id s, task ref self)
```

Arguments:

type	name	description
`task ref`	self	reference to the object

Return Type: `datetime`
value of the field

RPC name: get_status

Overview: Get the status field of the given task.
Signature:

```
(task_status_type) get_status (session_id s, task ref self)
```

Arguments:

type	name	description
`task ref`	self	reference to the object

Return Type: `task_status_type`
value of the field

RPC name: get_resident_on

Overview: Get the resident_on field of the given task.
Signature:

```
(host ref) get_resident_on (session_id s, task ref self)
```

Arguments:

type	name	description
`task ref`	self	reference to the object

Return Type: `host ref`
value of the field

RPC name: get_progress

Overview: Get the progress field of the given task.
Signature:

```
float get_progress (session_id s, task ref self)
```

Arguments:

type	name	description
`task ref`	self	reference to the object

Return Type: `float`
value of the field

RPC name: get_type

Overview: Get the type field of the given task.
Signature:

```
string get_type (session_id s, task ref self)
```

Arguments:

type	name	description
task ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_result

Overview: Get the result field of the given task.
Signature:

```
string get_result (session_id s, task ref self)
```

Arguments:

type	name	description
task ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_error_info

Overview: Get the error_info field of the given task.
Signature:

```
(string Set) get_error_info (session_id s, task ref self)
```

Arguments:

type	name	description
task ref	self	reference to the object

Return Type: string Set
value of the field

RPC name: get_by_uuid

Overview: Get a reference to the task instance with the specified UUID.
Signature:

```
(task ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
`string`	uuid	UUID of object to return

Return Type: `task ref`
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given task.
Signature:

```
(task record) get_record (session_id s, task ref self)
```

Arguments:

type	name	description
`task ref`	self	reference to the object

Return Type: `task record`
all fields from the object

RPC name: get_by_name_label

Overview: Get all the task instances with the given label.
Signature:

```
((task ref) Set) get_by_name_label (session_id s, string label)
```

Arguments:

type	name	description
`string`	label	label of object to return

Return Type: `(task ref) Set`
references to objects with matching names

2.6 Class: event

2.6.1 Fields for class: event

Name	**event**		
Description	*Asynchronous event registration and handling.*		
Quals	Field	Type	Description
RO_{ins}	`id`	int	An ID, monotonically increasing, and local to the current session
RO_{ins}	`timestamp`	datetime	The time at which the event occurred
RO_{ins}	`class`	string	The name of the class of the object that changed
RO_{ins}	`operation`	event_operation	The operation that was performed
RO_{ins}	`ref`	string	A reference to the object that changed
RO_{ins}	`obj_uuid`	string	The uuid of the object that changed

2.6.2 RPCs associated with class: event

RPC name: register

Overview: Registers this session with the event system. Specifying the empty list will register for all classes.
Signature:

```
void register (session_id s, string Set classes)
```

Arguments:

type	name	description
`string Set`	classes	register for events for the indicated classes

Return Type: `void`

RPC name: unregister

Overview: Unregisters this session with the event system.
Signature:

```
void unregister (session_id s, string Set classes)
```

Arguments:

type	name	description
`string Set`	classes	remove this session's registration for the indicated classes

Return Type: `void`

RPC name: next

Overview: Blocking call which returns a (possibly empty) batch of events.
Signature:

```
((event record) Set) next (session_id s)
```

Return Type: `(event record) Set`
the batch of events

Possible Error Codes: `SESSION_NOT_REGISTERED`

RPC name: get_current_id

Overview: Return the ID of the next event to be generated by the system.
Signature:

```
int get_current_id (session_id s)
```

Return Type: `int`
the event ID

2.7 Class: pool

2.7.1 Fields for class: pool

Name	**pool**		
Description	*Pool-wide information.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RW	`name_label`	string	Short name
RW	`name_description`	string	Description
RO_{run}	`master`	host ref	The host that is pool master
RW	`default_SR`	SR ref	Default SR for VDIs
RW	`suspend_image_SR`	SR ref	The SR in which VDIs for suspend images are created
RW	`crash_dump_SR`	SR ref	The SR in which VDIs for crash dumps are created
RW	`other_config`	(string → string) Map	additional configuration

2.7.2 RPCs associated with class: pool

RPC name: join

Overview: Instruct host to join a new pool.
Signature:

```
void join (session_id s, string master_address, string master_username, string master_password)
```

Arguments:

type	name	description
`string`	master_address	The hostname of the master of the pool to join
`string`	master_username	The username of the master (for initial authentication)
`string`	master_password	The password for the master (for initial authentication)

Return Type: `void`

RPC name: join_force

Overview: Instruct host to join a new pool.
Signature:

```
void join_force (session_id s, string master_address, string master_username, string master_password)
```

Arguments:

type	name	description
`string`	master_address	The hostname of the master of the pool to join
`string`	master_username	The username of the master (for initial authentication)
`string`	master_password	The password for the master (for initial authentication)

Return Type: `void`

RPC name: eject

Overview: Instruct a pool master to eject a host from the pool.
Signature:

```
void eject (session_id s, host ref host)
```

Arguments:

type	name	description
host ref	host	The host to eject

Return Type: void

RPC name: initial_auth

Overview: Internal use only.
Signature:

```
string initial_auth (session_id s)
```

Return Type: string

RPC name: emergency_transition_to_master

Overview: Instruct host that's currently a slave to transition to being master.
Signature:

```
void emergency_transition_to_master ()
```

Return Type: void

RPC name: emergency_reset_master

Overview: Instruct a slave already in a pool that the master has changed.
Signature:

```
void emergency_reset_master (string master_address)
```

Arguments:

type	name	description
string	master_address	The hostname of the master

Return Type: void

RPC name: recover_slaves

Overview: Instruct a pool master, M, to try and contact its slaves and, if slaves are in emergency mode, reset their master address to M.
Signature:

```
((host ref) Set) recover_slaves (session_id s)
```

Return Type: `(host ref) Set`
list of hosts whose master address were succesfully reset

RPC name: hello

Overview: Internal use only.
Signature:

```
(hello_return) hello (session_id s, string host_uuid)
```

Arguments:

type	name	description
`string`	host_uuid	

Return Type: `hello_return`

RPC name: is_slave

Overview: Internal use only.
Signature:

```
bool is_slave (host ref host)
```

Arguments:

type	name	description
`host ref`	host	

Return Type: `bool`
returns false if pinged host is master [indicating critical error condition]; true if pinged host is slave

RPC name: create_VLAN

Overview: Create PIFs, mapping a network to the same physical interface/VLAN on each host.
Signature:

```
((PIF ref) Set) create_VLAN (session_id s, string device, network ref network, int VLAN)
```

Arguments:

type	name	description
string	device	physical interface on which to crate the VLAN interface
network ref	network	network to which this interface should be connected
int	VLAN	VLAN tag for the new interface

Return Type: (PIF ref) Set
The references of the created PIF objects

Possible Error Codes: VLAN_TAG_INVALID

RPC name: slave_network_report

Overview: Internal use only.
Signature:

```
((PIF ref) Set) slave_network_report (session_id s, (string -> string) Map phydevs, (string -> string)
```

Arguments:

type	name	description
(string → string) Map	phydevs	(device,bridge) pairs of physical NICs on slave
(string → string) Map	dev_to_mac	(device,mac) pairs of physical NICs on slave
(string → int) Map	dev_to_mtu	(device,mtu) pairs of physical NICs on slave
host ref	slave_host	the host that the PIFs will be attached to when created

Return Type: (PIF ref) Set
refs for pifs corresponding to device list

RPC name: get_all

Overview: Return a list of all the pools known to the system.
Signature:

```
((pool ref) Set) get_all (session_id s)
```

Return Type: (pool ref) Set
references to all objects

RPC name: get_all_records_where

Overview: Return a list of all the pools known to the system.
Signature:

```
((pool ref -> pool record) Map) get_all_records_where (session_id s, string expr)
```

Arguments:

type	name	description
string	expr	expression representing records to fetch

Return Type: `(pool ref → pool record) Map`
records of all matching objects

RPC name: get_uuid

Overview: Get the uuid field of the given pool.
Signature:

```
string get_uuid (session_id s, pool ref self)
```

Arguments:

type	name	description
`pool ref`	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_name_label

Overview: Get the name_label field of the given pool.
Signature:

```
string get_name_label (session_id s, pool ref self)
```

Arguments:

type	name	description
`pool ref`	self	reference to the object

Return Type: `string`
value of the field

RPC name: set_name_label

Overview: Set the name_label field of the given pool.
Signature:

```
void set_name_label (session_id s, pool ref self, string value)
```

Arguments:

type	name	description
`pool ref`	self	reference to the object
`string`	value	New value to set

Return Type: `void`

RPC name: get_name_description

Overview: Get the name_description field of the given pool.
Signature:

```
string get_name_description (session_id s, pool ref self)
```

Arguments:

type	name	description
pool ref	self	reference to the object

Return Type: string
value of the field

RPC name: set_name_description

Overview: Set the name_description field of the given pool.
Signature:

```
void set_name_description (session_id s, pool ref self, string value)
```

Arguments:

type	name	description
pool ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_master

Overview: Get the master field of the given pool.
Signature:

```
(host ref) get_master (session_id s, pool ref self)
```

Arguments:

type	name	description
pool ref	self	reference to the object

Return Type: host ref
value of the field

RPC name: get_default_SR

Overview: Get the default_SR field of the given pool.
Signature:

```
(SR ref) get_default_SR (session_id s, pool ref self)
```

Arguments:

type	name	description
pool ref	self	reference to the object

Return Type: SR ref
value of the field

RPC name: set_default_SR

Overview: Set the default_SR field of the given pool.
Signature:

```
void set_default_SR (session_id s, pool ref self, SR ref value)
```

Arguments:

type	name	description
pool ref	self	reference to the object
SR ref	value	New value to set

Return Type: void

RPC name: get_suspend_image_SR

Overview: Get the suspend_image_SR field of the given pool.
Signature:

```
(SR ref) get_suspend_image_SR (session_id s, pool ref self)
```

Arguments:

type	name	description
pool ref	self	reference to the object

Return Type: SR ref
value of the field

RPC name: set_suspend_image_SR

Overview: Set the suspend_image_SR field of the given pool.
Signature:

```
void set_suspend_image_SR (session_id s, pool ref self, SR ref value)
```

Arguments:

type	name	description
pool ref	self	reference to the object
SR ref	value	New value to set

Return Type: void

RPC name: get_crash_dump_SR

Overview: Get the crash_dump_SR field of the given pool.
Signature:

```
(SR ref) get_crash_dump_SR (session_id s, pool ref self)
```

Arguments:

type	name	description
pool ref	self	reference to the object

Return Type: SR ref
value of the field

RPC name: set_crash_dump_SR

Overview: Set the crash_dump_SR field of the given pool.
Signature:

```
void set_crash_dump_SR (session_id s, pool ref self, SR ref value)
```

Arguments:

type	name	description
pool ref	self	reference to the object
SR ref	value	New value to set

Return Type: void

RPC name: get_other_config

Overview: Get the other_config field of the given pool.
Signature:

```
((string -> string) Map) get_other_config (session_id s, pool ref self)
```

Arguments:

type	name	description
pool ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: set_other_config

Overview: Set the other_config field of the given pool.
Signature:

```
void set_other_config (session_id s, pool ref self, (string -> string) Map value)
```

Arguments:

type	name	description
pool ref	self	reference to the object
(string → string) Map	value	New value to set

Return Type: void

RPC name: add_to_other_config

Overview: Add the given key-value pair to the other_config field of the given pool.
Signature:

```
void add_to_other_config (session_id s, pool ref self, string key, string value)
```

Arguments:

type	name	description
pool ref	self	reference to the object
string	key	Key to add
string	value	Value to add

Return Type: void

RPC name: remove_from_other_config

Overview: Remove the given key and its corresponding value from the other_config field of the given pool. If the key is not in that Map, then do nothing.
Signature:

```
void remove_from_other_config (session_id s, pool ref self, string key)
```

Arguments:

type	name	description
pool ref	self	reference to the object
string	key	Key to remove

Return Type: void

RPC name: get_by_uuid

Overview: Get a reference to the pool instance with the specified UUID.
Signature:

```
(pool ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
string	uuid	UUID of object to return

Return Type: pool ref

reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given pool.
Signature:

```
(pool record) get_record (session_id s, pool ref self)
```

Arguments:

type	name	description
pool ref	self	reference to the object

Return Type: `pool record`
all fields from the object

2.8 Class: VM

2.8.1 Fields for class: VM

Name	**VM**		
Description	*A virtual machine (or 'guest').*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RO_{run}	`allowed_operations`	(vm_operations) Set	list of the operations allowed in this state
RO_{run}	`current_operations`	(string → vm_operations) Map	Map of task reference to current operation enumeration
RO_{run}	`power_state`	vm_power_state	Current power state of the machine
RW	`name/label`	string	a human-readable name
RW	`name/description`	string	a notes field containg human-readable description
RW	`user_version`	int	a user version number for this machine
RW	`is_a_template`	bool	true if this is a template. Template VMs can never be started, they are used only for cloning other VMs
RO_{run}	`suspend_VDI`	VDI ref	The VDI that a suspend image is stored on. (Only has meaning if VM is currently suspended)
RO_{run}	`resident_on`	host ref	the host the VM is currently resident on
RW	`affinity`	host ref	a host which the VM has some affinity for (or NULL). This is used as a hint to the start call when it decides where to run the VM. Implementations are free to ignore this field.
RW	`memory/static_max`	int	Statically-set (i.e. absolute) maximum (bytes)
RW	`memory/dynamic_max`	int	Dynamic maximum (bytes)
RW	`memory/dynamic_min`	int	Dynamic minimum (bytes)
RW	`memory/static_min`	int	Statically-set (i.e. absolute) mininum (bytes)
RW	`VCPUs/params`	(string → string) Map	configuration parameters for the selected VCPU policy
RW	`VCPUs/max`	int	Max number of VCPUs
RW	`VCPUs/at_startup`	int	Boot number of VCPUs
RW	`actions/after_shutdown`	on_normal_exit	action to take after the guest has shutdown itself
RW	`actions/after_reboot`	on_normal_exit	action to take after the guest has rebooted itself
RW	`actions/after_crash`	on_crash_behaviour	action to take if the guest crashes
RO_{run}	`consoles`	(console ref) Set	virtual console devices
RO_{run}	`VIFs`	(VIF ref) Set	virtual network interfaces
RO_{run}	`VBDs`	(VBD ref) Set	virtual block devices
RO_{run}	`crash_dumps`	(crashdump ref) Set	crash dumps associated with this VM
RO_{run}	`VTPMs`	(VTPM ref) Set	virtual TPMs
RW	`PV/bootloader`	string	name of or path to bootloader
RW	`PV/kernel`	string	path to the kernel
RW	`PV/ramdisk`	string	path to the initrd

RW	PV/args	string	kernel command-line arguments
RW	PV/bootloader_args	string	miscellaneous arguments for the bootloader
RW	PV/legacy_args	string	to make Zurich guests boot
RW	HVM/boot_policy	string	HVM boot policy
RW	HVM/boot_params	(string → string) Map	HVM boot params
RW	platform	(string → string) Map	platform-specific configuration
RW	PCI_bus	string	PCI bus path for pass-through devices
RW	other_config	(string → string) Map	additional configuration
RO_{run}	domid	int	domain ID (if available, -1 otherwise)
RO_{run}	domarch	string	Domain architecture (if available, null string otherwise)
RO_{run}	last_boot_CPU_flags	(string → string) Map	describes the CPU flags on which the VM was last booted
RO_{run}	is_control_domain	bool	true if this is a control domain (domain 0 or a driver domain)
RO_{run}	metrics	VM_metrics ref	metrics associated with this VM
RO_{run}	guest_metrics	VM_guest_metrics ref	metrics associated with the running guest
RW	recommendations	string	An XML specification of recommended values and ranges for properties of this VM

2.8.2 RPCs associated with class: VM

RPC name: clone

Overview: Clones the specified VM, making a new VM. Clone automatically exploits the capabilities of the underlying storage repository in which the VM's disk images are stored (e.g. Copy on Write). This function can only be called when the VM is in the Halted State.
Signature:

```
(VM ref) clone (session_id s, VM ref vm, string new_name)
```

Arguments:

type	name	description
VM ref	vm	The VM to be cloned
string	new_name	The name of the cloned VM

Return Type: VM ref
The ID of the newly created VM.

Possible Error Codes: VM_BAD_POWER_STATE, SR_FULL, OPERATION_NOT_ALLOWED

RPC name: copy

Overview: Copied the specified VM, making a new VM. Unlike clone, copy does not exploits the capabilities of the underlying storage repository in which the VM's disk images are stored. Instead, copy guarantees that the disk images of the newly created VM will be 'full disks' - i.e. not part of a CoW chain. This function can only be called when the VM is in the Halted State.
Signature:

```
(VM ref) copy (session_id s, VM ref vm, string new_name, SR ref sr)
```

Arguments:

type	name	description
VM ref	vm	The VM to be copied
string	new_name	The name of the copied VM
SR ref	sr	An SR to copy all the VM's disks into (if an invalid reference then it uses the existing SRs)

Return Type: VM ref
The ID of the newly created VM.

Possible Error Codes: VM_BAD_POWER_STATE, SR_FULL, OPERATION_NOT_ALLOWED

RPC name: provision

Overview: Inspects the disk configuration contained within the VM's other_config, creates VDIs and VBDs and then executes any applicable post-install script.
Signature:

```
void provision (session_id s, VM ref vm)
```

Arguments:

type	name	description
VM ref	vm	The VM to be provisioned

Return Type: void

Possible Error Codes: VM_BAD_POWER_STATE, SR_FULL, OPERATION_NOT_ALLOWED

RPC name: start

Overview: Start the specified VM. This function can only be called with the VM is in the Halted State.
Signature:

```
void start (session_id s, VM ref vm, bool start_paused, bool force)
```

Arguments:

type	name	description
VM ref	vm	The VM to start
bool	start_paused	Instantiate VM in paused state if set to true.
bool	force	Attempt to force the VM to start. If this flag is false then the VM may fail pre-boot safety checks (e.g. if the CPU the VM last booted on looks substantially different to the current one)

Return Type: void

Possible Error Codes: VM_BAD_POWER_STATE, VM_HVM_REQUIRED, VM_IS_TEMPLATE, OTHER_OPERATION_IN_PROGRESS, OPERATION_NOT_ALLOWED, BOOTLOADER_FAILED, UNKNOWN_BOOTLOADER

RPC name: start_on

Overview: Start the specified VM on a particular host. This function can only be called with the VM is in the Halted State.
Signature:

```
void start_on (session_id s, VM ref vm, host ref host, bool start_paused, bool force)
```

Arguments:

type	name	description
VM ref	vm	The VM to start
host ref	host	The Host on which to start the VM
bool	start_paused	Instantiate VM in paused state if set to true.
bool	force	Attempt to force the VM to start. If this flag is false then the VM may fail pre-boot safety checks (e.g. if the CPU the VM last booted on looks substantially different to the current one)

Return Type: void

Possible Error Codes: VM_BAD_POWER_STATE, VM_IS_TEMPLATE, OTHER_OPERATION_IN_PROGRESS, OPERATION_NOT_ALLOWED, BOOTLOADER_FAILED, UNKNOWN_BOOTLOADER

RPC name: pause

Overview: Pause the specified VM. This can only be called when the specified VM is in the Running state.
Signature:

```
void pause (session_id s, VM ref vm)
```

Arguments:

type	name	description
VM ref	vm	The VM to pause

Return Type: void

Possible Error Codes: VM_BAD_POWER_STATE, OTHER_OPERATION_IN_PROGRESS, OPERATION_NOT_ALLOWED, VM_IS_TEMPLATE

RPC name: unpause

Overview: Resume the specified VM. This can only be called when the specified VM is in the Paused state.
Signature:

```
void unpause (session_id s, VM ref vm)
```

Arguments:

type	name	description
VM ref	vm	The VM to unpause

Return Type: void

Possible Error Codes: VM_BAD_POWER_STATE, OPERATION_NOT_ALLOWED, VM_IS_TEMPLATE

RPC name: clean_shutdown

Overview: Attempt to cleanly shutdown the specified VM. (Note: this may not be supported—e.g. if a guest agent is not installed).
Once shutdown has been completed perform poweroff action specified in guest configuration.
This can only be called when the specified VM is in the Running state.
Signature:

```
void clean_shutdown (session_id s, VM ref vm)
```

Arguments:

type	name	description
VM ref	vm	The VM to shutdown

Return Type: void

Possible Error Codes: VM_BAD_POWER_STATE, OTHER_OPERATION_IN_PROGRESS, OPERATION_NOT_ALLOWED, VM_IS_TEMPLATE

RPC name: clean_reboot

Overview: Attempt to cleanly shutdown the specified VM (Note: this may not be supported—e.g. if a guest agent is not installed).
Once shutdown has been completed perform reboot action specified in guest configuration.
This can only be called when the specified VM is in the Running state.
Signature:

```
void clean_reboot (session_id s, VM ref vm)
```

Arguments:

type	name	description
VM ref	vm	The VM to shutdown

Return Type: void

Possible Error Codes: VM_BAD_POWER_STATE, OTHER_OPERATION_IN_PROGRESS, OPERATION_NOT_ALLOWED, VM_IS_TEMPLATE

RPC name: hard_shutdown

Overview: Stop executing the specified VM without attempting a clean shutdown. Then perform poweroff action specified in VM configuration.
Signature:

```
void hard_shutdown (session_id s, VM ref vm)
```

Arguments:

type	name	description
VM ref	vm	The VM to destroy

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OTHER_OPERATION_IN_PROGRESS`, `OPERATION_NOT_ALLOWED`, `VM_IS_TEMPLATE`

RPC name: power_state_reset

Overview: Reset the power-state of the VM to halted in the database only. (Used to recover from slave failures in pooling scenarios by resetting the power-states of VMs running on dead slaves to halted.) This is a potentially dangerous operation; use with care.
Signature:

```
void power_state_reset (session_id s, VM ref vm)
```

Arguments:

type	name	description
`VM ref`	vm	The VM to reset

Return Type: `void`

RPC name: hard_reboot

Overview: Stop executing the specified VM without attempting a clean shutdown. Then perform reboot action specified in VM configuration.
Signature:

```
void hard_reboot (session_id s, VM ref vm)
```

Arguments:

type	name	description
`VM ref`	vm	The VM to reboot

Return Type: `void`

Possible Error Codes: `VM_BAD_POWER_STATE`, `OTHER_OPERATION_IN_PROGRESS`, `OPERATION_NOT_ALLOWED`, `VM_IS_TEMPLATE`

RPC name: suspend

Overview: Suspend the specified VM to disk. This can only be called when the specified VM is in the Running state.
Signature:

```
void suspend (session_id s, VM ref vm)
```

Arguments:

type	name	description
`VM ref`	vm	The VM to suspend

Return Type: `void`

Possible Error Codes: VM_BAD_POWER_STATE, OTHER_OPERATION_IN_PROGRESS, OPERATION_NOT_ALLOWED, VM_IS_TEMPLATE

RPC name: resume

Overview: Awaken the specified VM and resume it. This can only be called when the specified VM is in the Suspended state.
Signature:

```
void resume (session_id s, VM ref vm, bool start_paused, bool force)
```

Arguments:

type	name	description
VM ref	vm	The VM to resume
bool	start_paused	Resume VM in paused state if set to true.
bool	force	Attempt to force the VM to resume. If this flag is false then the VM may fail pre-resume safety checks (e.g. if the CPU the VM was running on looks substantially different to the current one)

Return Type: void

Possible Error Codes: VM_BAD_POWER_STATE, OPERATION_NOT_ALLOWED, VM_IS_TEMPLATE

RPC name: resume_on

Overview: Awaken the specified VM and resume it on a particular Host. This can only be called when the specified VM is in the Suspended state.
Signature:

```
void resume_on (session_id s, VM ref vm, host ref host, bool start_paused, bool force)
```

Arguments:

type	name	description
VM ref	vm	The VM to resume
host ref	host	The Host on which to resume the VM
bool	start_paused	Resume VM in paused state if set to true.
bool	force	Attempt to force the VM to resume. If this flag is false then the VM may fail pre-resume safety checks (e.g. if the CPU the VM was running on looks substantially different to the current one)

Return Type: void

Possible Error Codes: VM_BAD_POWER_STATE, OPERATION_NOT_ALLOWED, VM_IS_TEMPLATE

RPC name: statistics

Overview: Give statistics against this VM. This can only be called when the specified VM is in the Running state.
Signature:

```
string statistics (session_id s, VM ref vm, string resource, int timescale, int nrecord)
```

Arguments:

type	name	description
VM ref	vm	The VM to interrogate
string	resource	Type of resource to query.
int	timescale	Time scale in second of data
int	nrecord	Number of record to get back

Return Type: string
resource string

Possible Error Codes: VM_BAD_POWER_STATE, VM_IS_TEMPLATE, OPERATION_NOT_ALLOWED

RPC name: pool_migrate

Overview: Migrate a VM to another Host. This can only be called when the specified VM is in the Running state.
Signature:

```
void pool_migrate (session_id s, VM ref vm, host ref host, (string -> string) Map options)
```

Arguments:

type	name	description
VM ref	vm	The VM to migrate
host ref	host	The target host
(string → string) Map	options	Extra configuration operations

Return Type: void

Possible Error Codes: VM_BAD_POWER_STATE, OTHER_OPERATION_IN_PROGRESS, VM_IS_TEMPLATE, OPERATION_NOT_ALLOWED, VM_MIGRATE_FAILED

RPC name: set_VCPUs_number_live

Overview: Set this VM's VCPUs/at_startup value, and set the same value on the VM, if running.
Signature:

```
void set_VCPUs_number_live (session_id s, VM ref self, int nvcpu)
```

Arguments:

type	name	description
VM ref	self	The VM
int	nvcpu	The number of VCPUs

Return Type: void

RPC name: add_to_VCPUs_params_live

Overview: Add the given key-value pair to VM.VCPUs_params, and apply that value on the running VM.
Signature:

```
void add_to_VCPUs_params_live (session_id s, VM ref self, string key, string value)
```

Arguments:

type	name	description
VM ref	self	The VM
string	key	The key
string	value	The value

Return Type: void

RPC name: set_memory_dynamic_max_live

Overview: Set memory_dynamic_max in database and on running VM.
Signature:

```
void set_memory_dynamic_max_live (session_id s, VM ref self, int max)
```

Arguments:

type	name	description
VM ref	self	The VM
int	max	The memory_dynamic_max value

Return Type: void

RPC name: set_memory_dynamic_min_live

Overview: Set memory_dynamic_min in database and on running VM.
Signature:

```
void set_memory_dynamic_min_live (session_id s, VM ref self, int min)
```

Arguments:

type	name	description
VM ref	self	The VM
int	min	The memory_dynamic_min value

Return Type: void

RPC name: send_sysrq

Overview: Send the given key as a sysrq to this VM. The key is specified as a single character (a String of length 1). This can only be called when the specified VM is in the Running state.
Signature:

```
void send_sysrq (session_id s, VM ref vm, string key)
```

Arguments:

type	name	description
VM ref	vm	The VM
string	key	The key to send

Return Type: void

Possible Error Codes: VM_BAD_POWER_STATE

RPC name: send_trigger

Overview: Send the named trigger to this VM. This can only be called when the specified VM is in the Running state.
Signature:

```
void send_trigger (session_id s, VM ref vm, string trigger)
```

Arguments:

type	name	description
VM ref	vm	The VM
string	trigger	The trigger to send

Return Type: void

Possible Error Codes: VM_BAD_POWER_STATE

RPC name: migrate

Overview: Migrate the VM to another host. This can only be called when the specified VM is in the Running state.
Signature:

```
void migrate (session_id s, VM ref vm, string dest, bool live, (string -> string) Map options)
```

Arguments:

type	name	description
VM ref	vm	The VM
string	dest	The destination host
bool	live	Live migration
(string → string) Map	options	Other parameters

Return Type: void

Possible Error Codes: VM_BAD_POWER_STATE

RPC name: get_boot_record

Overview: Returns a record describing the VM's state when it last booted.
Signature:

```
(VM record) get_boot_record (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	The VM whose boot-time state to return

Return Type: VM record
A record describing the VM

RPC name: assert_operation_valid

Overview: Check to see whether this operation is acceptable in the current state of the system, raising an error if the operation is invalid for some reason.
Signature:

```
void assert_operation_valid (session_id s, VM ref self, vm_operations op)
```

Arguments:

type	name	description
VM ref	self	reference to the object
vm_operations	op	proposed operation

Return Type: void

RPC name: update_allowed_operations

Overview: Recomputes the list of acceptable operations.
Signature:

```
void update_allowed_operations (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: void

RPC name: get_allowed_VBD_devices

Overview: Returns a list of the allowed values that a VBD device field can take.
Signature:

```
(string Set) get_allowed_VBD_devices (session_id s, VM ref vm)
```

Arguments:

type	name	description
VM ref	vm	The VM to query

Return Type: string Set

The allowed values

RPC name: get_allowed_VIF_devices

Overview: Returns a list of the allowed values that a VIF device field can take.
Signature:

```
(string Set) get_allowed_VIF_devices (session_id s, VM ref vm)
```

Arguments:

type	name	description
VM ref	vm	The VM to query

Return Type: `string Set`
The allowed values

RPC name: get_possible_hosts

Overview: Return the list of hosts on which this VM may run.
Signature:

```
((host ref) Set) get_possible_hosts (session_id s, VM ref vm)
```

Arguments:

type	name	description
VM ref	vm	The VM

Return Type: `(host ref) Set`
The possible hosts

RPC name: assert_can_boot_here

Overview: Returns an error if the VM could not boot on this host for some reason.
Signature:

```
void assert_can_boot_here (session_id s, VM ref self, host ref host)
```

Arguments:

type	name	description
VM ref	self	The VM
host ref	host	The host

Return Type: `void`

RPC name: get_all

Overview: Return a list of all the VMs known to the system.
Signature:

```
((VM ref) Set) get_all (session_id s)
```

Return Type: `(VM ref) Set`
references to all objects

RPC name: get_all_records_where

Overview: Return a list of all the VMs known to the system.
Signature:

```
((VM ref -> VM record) Map) get_all_records_where (session_id s, string expr)
```

Arguments:

type	name	description
`string`	expr	expression representing records to fetch

Return Type: `(VM ref → VM record) Map`
records of all matching objects

RPC name: get_uuid

Overview: Get the uuid field of the given VM.
Signature:

```
string get_uuid (session_id s, VM ref self)
```

Arguments:

type	name	description
`VM ref`	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_allowed_operations

Overview: Get the allowed_operations field of the given VM.
Signature:

```
((vm_operations) Set) get_allowed_operations (session_id s, VM ref self)
```

Arguments:

type	name	description
`VM ref`	self	reference to the object

Return Type: `(vm_operations) Set`

value of the field

RPC name: get_current_operations

Overview: Get the current_operations field of the given VM.
Signature:

```
((string -> vm_operations) Map) get_current_operations (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: (string → vm_operations) Map
value of the field

RPC name: get_power_state

Overview: Get the power_state field of the given VM.
Signature:

```
(vm_power_state) get_power_state (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: vm_power_state
value of the field

RPC name: get_name_label

Overview: Get the name/label field of the given VM.
Signature:

```
string get_name_label (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: string
value of the field

RPC name: set_name_label

Overview: Set the name/label field of the given VM.
Signature:

```
void set_name_label (session_id s, VM ref self, string value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_name_description

Overview: Get the name/description field of the given VM.
Signature:

```
string get_name_description (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: string
value of the field

RPC name: set_name_description

Overview: Set the name/description field of the given VM.
Signature:

```
void set_name_description (session_id s, VM ref self, string value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_user_version

Overview: Get the user_version field of the given VM.
Signature:

```
int get_user_version (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: int
value of the field

RPC name: set_user_version

Overview: Set the user_version field of the given VM.
Signature:

```
void set_user_version (session_id s, VM ref self, int value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
int	value	New value to set

Return Type: void

RPC name: get_is_a_template

Overview: Get the is_a_template field of the given VM.
Signature:

```
bool get_is_a_template (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: bool
value of the field

RPC name: set_is_a_template

Overview: Set the is_a_template field of the given VM.
Signature:

```
void set_is_a_template (session_id s, VM ref self, bool value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
bool	value	New value to set

Return Type: void

RPC name: get_suspend_VDI

Overview: Get the suspend_VDI field of the given VM.
Signature:

```
(VDI ref) get_suspend_VDI (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: VDI ref
value of the field

RPC name: get_resident_on

Overview: Get the resident_on field of the given VM.
Signature:

```
(host ref) get_resident_on (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: host ref
value of the field

RPC name: get_affinity

Overview: Get the affinity field of the given VM.
Signature:

```
(host ref) get_affinity (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: host ref
value of the field

RPC name: set_affinity

Overview: Set the affinity field of the given VM.
Signature:

```
void set_affinity (session_id s, VM ref self, host ref value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
host ref	value	New value to set

Return Type: void

RPC name: get_memory_static_max

Overview: Get the memory/static_max field of the given VM.
Signature:

```
int get_memory_static_max (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: int
value of the field

RPC name: set_memory_static_max

Overview: Set the memory/static_max field of the given VM.
Signature:

```
void set_memory_static_max (session_id s, VM ref self, int value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
int	value	New value to set

Return Type: void

RPC name: get_memory_dynamic_max

Overview: Get the memory/dynamic_max field of the given VM.
Signature:

```
int get_memory_dynamic_max (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: int
value of the field

RPC name: set_memory_dynamic_max

Overview: Set the memory/dynamic_max field of the given VM.
Signature:

```
void set_memory_dynamic_max (session_id s, VM ref self, int value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
int	value	New value to set

Return Type: void

RPC name: get_memory_dynamic_min

Overview: Get the memory/dynamic_min field of the given VM.
Signature:

```
int get_memory_dynamic_min (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: int
value of the field

RPC name: set_memory_dynamic_min

Overview: Set the memory/dynamic_min field of the given VM.
Signature:

```
void set_memory_dynamic_min (session_id s, VM ref self, int value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
int	value	New value to set

Return Type: void

RPC name: get_memory_static_min

Overview: Get the memory/static_min field of the given VM.
Signature:

```
int get_memory_static_min (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: int
value of the field

RPC name: set_memory_static_min

Overview: Set the memory/static_min field of the given VM.
Signature:

```
void set_memory_static_min (session_id s, VM ref self, int value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
int	value	New value to set

Return Type: void

RPC name: get_VCPUs_params

Overview: Get the VCPUs/params field of the given VM.
Signature:

```
((string -> string) Map) get_VCPUs_params (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: set_VCPUs_params

Overview: Set the VCPUs/params field of the given VM.
Signature:

```
void set_VCPUs_params (session_id s, VM ref self, (string -> string) Map value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
(string → string) Map	value	New value to set

Return Type: void

RPC name: add_to_VCPUs_params

Overview: Add the given key-value pair to the VCPUs/params field of the given VM.
Signature:

```
void add_to_VCPUs_params (session_id s, VM ref self, string key, string value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
string	key	Key to add
string	value	Value to add

Return Type: void

RPC name: remove_from_VCPUs_params

Overview: Remove the given key and its corresponding value from the VCPUs/params field of the given VM. If the key is not in that Map, then do nothing.
Signature:

```
void remove_from_VCPUs_params (session_id s, VM ref self, string key)
```

Arguments:

type	name	description
VM ref	self	reference to the object
string	key	Key to remove

Return Type: void

RPC name: get_VCPUs_max

Overview: Get the VCPUs/max field of the given VM.
Signature:

```
int get_VCPUs_max (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: int
value of the field

RPC name: set_VCPUs_max

Overview: Set the VCPUs/max field of the given VM.
Signature:

```
void set_VCPUs_max (session_id s, VM ref self, int value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
int	value	New value to set

Return Type: void

RPC name: get_VCPUs_at_startup

Overview: Get the VCPUs/at_startup field of the given VM.
Signature:

```
int get_VCPUs_at_startup (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: int
value of the field

RPC name: set_VCPUs_at_startup

Overview: Set the VCPUs/at_startup field of the given VM.
Signature:

```
void set_VCPUs_at_startup (session_id s, VM ref self, int value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
int	value	New value to set

Return Type: void

RPC name: get_actions_after_shutdown

Overview: Get the actions/after_shutdown field of the given VM.
Signature:

```
(on_normal_exit) get_actions_after_shutdown (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: on_normal_exit
value of the field

RPC name: set_actions_after_shutdown

Overview: Set the actions/after_shutdown field of the given VM.
Signature:

```
void set_actions_after_shutdown (session_id s, VM ref self, on_normal_exit value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
on_normal_exit	value	New value to set

Return Type: void

RPC name: get_actions_after_reboot

Overview: Get the actions/after_reboot field of the given VM.
Signature:

```
(on_normal_exit) get_actions_after_reboot (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: on_normal_exit
value of the field

RPC name: set_actions_after_reboot

Overview: Set the actions/after_reboot field of the given VM.
Signature:

```
void set_actions_after_reboot (session_id s, VM ref self, on_normal_exit value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
on_normal_exit	value	New value to set

Return Type: void

RPC name: get_actions_after_crash

Overview: Get the actions/after_crash field of the given VM.
Signature:

```
(on_crash_behaviour) get_actions_after_crash (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: on_crash_behaviour
value of the field

RPC name: set_actions_after_crash

Overview: Set the actions/after_crash field of the given VM.
Signature:

```
void set_actions_after_crash (session_id s, VM ref self, on_crash_behaviour value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
on_crash_behaviour	value	New value to set

Return Type: void

RPC name: get_consoles

Overview: Get the consoles field of the given VM.
Signature:

```
((console ref) Set) get_consoles (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: (console ref) Set
value of the field

RPC name: get_VIFs

Overview: Get the VIFs field of the given VM.
Signature:

```
((VIF ref) Set) get_VIFs (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: (VIF ref) Set
value of the field

RPC name: get_VBDs

Overview: Get the VBDs field of the given VM.
Signature:

```
((VBD ref) Set) get_VBDs (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: `(VBD ref) Set`
value of the field

RPC name: get_crash_dumps

Overview: Get the crash_dumps field of the given VM.
Signature:

```
((crashdump ref) Set) get_crash_dumps (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: `(crashdump ref) Set`
value of the field

RPC name: get_VTPMs

Overview: Get the VTPMs field of the given VM.
Signature:

```
((VTPM ref) Set) get_VTPMs (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: `(VTPM ref) Set`
value of the field

RPC name: get_PV_bootloader

Overview: Get the PV/bootloader field of the given VM.
Signature:

```
string get_PV_bootloader (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: string
value of the field

RPC name: set_PV_bootloader

Overview: Set the PV/bootloader field of the given VM.
Signature:

```
void set_PV_bootloader (session_id s, VM ref self, string value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_PV_kernel

Overview: Get the PV/kernel field of the given VM.
Signature:

```
string get_PV_kernel (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: string
value of the field

RPC name: set_PV_kernel

Overview: Set the PV/kernel field of the given VM.
Signature:

```
void set_PV_kernel (session_id s, VM ref self, string value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_PV_ramdisk

Overview: Get the PV/ramdisk field of the given VM.
Signature:

```
string get_PV_ramdisk (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: string
value of the field

RPC name: set_PV_ramdisk

Overview: Set the PV/ramdisk field of the given VM.
Signature:

```
void set_PV_ramdisk (session_id s, VM ref self, string value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_PV_args

Overview: Get the PV/args field of the given VM.
Signature:

```
string get_PV_args (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: string
value of the field

RPC name: set_PV_args

Overview: Set the PV/args field of the given VM.
Signature:

```
void set_PV_args (session_id s, VM ref self, string value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_PV_bootloader_args

Overview: Get the PV/bootloader_args field of the given VM.
Signature:

```
string get_PV_bootloader_args (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: string
value of the field

RPC name: set_PV_bootloader_args

Overview: Set the PV/bootloader_args field of the given VM.
Signature:

```
void set_PV_bootloader_args (session_id s, VM ref self, string value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_PV_legacy_args

Overview: Get the PV/legacy_args field of the given VM.
Signature:

```
string get_PV_legacy_args (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: string
value of the field

RPC name: set_PV_legacy_args

Overview: Set the PV/legacy_args field of the given VM.
Signature:

```
void set_PV_legacy_args (session_id s, VM ref self, string value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_HVM_boot_policy

Overview: Get the HVM/boot_policy field of the given VM.
Signature:

```
string get_HVM_boot_policy (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: string
value of the field

RPC name: set_HVM_boot_policy

Overview: Set the HVM/boot_policy field of the given VM.
Signature:

```
void set_HVM_boot_policy (session_id s, VM ref self, string value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_HVM_boot_params

Overview: Get the HVM/boot_params field of the given VM.
Signature:

```
((string -> string) Map) get_HVM_boot_params (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: set_HVM_boot_params

Overview: Set the HVM/boot_params field of the given VM.
Signature:

```
void set_HVM_boot_params (session_id s, VM ref self, (string -> string) Map value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
(string → string) Map	value	New value to set

Return Type: void

RPC name: add_to_HVM_boot_params

Overview: Add the given key-value pair to the HVM/boot_params field of the given VM.
Signature:

```
void add_to_HVM_boot_params (session_id s, VM ref self, string key, string value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
string	key	Key to add
string	value	Value to add

Return Type: void

RPC name: remove_from_HVM_boot_params

Overview: Remove the given key and its corresponding value from the HVM/boot_params field of the given VM. If the key is not in that Map, then do nothing.
Signature:

```
void remove_from_HVM_boot_params (session_id s, VM ref self, string key)
```

Arguments:

type	name	description
VM ref	self	reference to the object
string	key	Key to remove

Return Type: `void`

RPC name: get_platform

Overview: Get the platform field of the given VM.
Signature:

```
((string -> string) Map) get_platform (session_id s, VM ref self)
```

Arguments:

type	name	description
`VM ref`	self	reference to the object

Return Type: `(string → string) Map`
value of the field

RPC name: set_platform

Overview: Set the platform field of the given VM.
Signature:

```
void set_platform (session_id s, VM ref self, (string -> string) Map value)
```

Arguments:

type	name	description
`VM ref`	self	reference to the object
`(string → string) Map`	value	New value to set

Return Type: `void`

RPC name: add_to_platform

Overview: Add the given key-value pair to the platform field of the given VM.
Signature:

```
void add_to_platform (session_id s, VM ref self, string key, string value)
```

Arguments:

type	name	description
`VM ref`	self	reference to the object
`string`	key	Key to add
`string`	value	Value to add

Return Type: `void`

RPC name: remove_from_platform

Overview: Remove the given key and its corresponding value from the platform field of the given VM. If the key is not in that Map, then do nothing.
Signature:

```
void remove_from_platform (session_id s, VM ref self, string key)
```

Arguments:

type	name	description
VM ref	self	reference to the object
string	key	Key to remove

Return Type: void

RPC name: get_PCI_bus

Overview: Get the PCI_bus field of the given VM.
Signature:

```
string get_PCI_bus (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: string
value of the field

RPC name: set_PCI_bus

Overview: Set the PCI_bus field of the given VM.
Signature:

```
void set_PCI_bus (session_id s, VM ref self, string value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_other_config

Overview: Get the other_config field of the given VM.
Signature:

```
((string -> string) Map) get_other_config (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: set_other_config

Overview: Set the other_config field of the given VM.
Signature:

```
void set_other_config (session_id s, VM ref self, (string -> string) Map value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
(string → string) Map	value	New value to set

Return Type: void

RPC name: add_to_other_config

Overview: Add the given key-value pair to the other_config field of the given VM.
Signature:

```
void add_to_other_config (session_id s, VM ref self, string key, string value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
string	key	Key to add
string	value	Value to add

Return Type: void

RPC name: remove_from_other_config

Overview: Remove the given key and its corresponding value from the other_config field of the given VM. If the key is not in that Map, then do nothing.
Signature:

```
void remove_from_other_config (session_id s, VM ref self, string key)
```

Arguments:

type	name	description
VM ref	self	reference to the object
string	key	Key to remove

Return Type: `void`

RPC name: get_domid

Overview: Get the domid field of the given VM.
Signature:

```
int get_domid (session_id s, VM ref self)
```

Arguments:

type	name	description
`VM ref`	self	reference to the object

Return Type: `int`
value of the field

RPC name: get_domarch

Overview: Get the domarch field of the given VM.
Signature:

```
string get_domarch (session_id s, VM ref self)
```

Arguments:

type	name	description
`VM ref`	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_last_boot_CPU_flags

Overview: Get the last_boot_CPU_flags field of the given VM.
Signature:

```
((string -> string) Map) get_last_boot_CPU_flags (session_id s, VM ref self)
```

Arguments:

type	name	description
`VM ref`	self	reference to the object

Return Type: `(string → string) Map`
value of the field

RPC name: get_is_control_domain

Overview: Get the is_control_domain field of the given VM.
Signature:

```
bool get_is_control_domain (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: bool
value of the field

RPC name: get_metrics

Overview: Get the metrics field of the given VM.
Signature:

```
(VM_metrics ref) get_metrics (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: VM_metrics ref
value of the field

RPC name: get_guest_metrics

Overview: Get the guest_metrics field of the given VM.
Signature:

```
(VM_guest_metrics ref) get_guest_metrics (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: VM_guest_metrics ref
value of the field

RPC name: get_recommendations

Overview: Get the recommendations field of the given VM.
Signature:

```
string get_recommendations (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: string
value of the field

RPC name: set_recommendations

Overview: Set the recommendations field of the given VM.
Signature:

```
void set_recommendations (session_id s, VM ref self, string value)
```

Arguments:

type	name	description
VM ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: create

Overview: Create a new VM instance, and return its handle.
Signature:

```
(VM ref) create (session_id s, VM record args)
```

Arguments:

type	name	description
VM record	args	All constructor arguments

Return Type: VM ref
reference to the newly created object

RPC name: destroy

Overview: Destroy the specified VM. The VM is completely removed from the system. This function can only be called when the VM is in the Halted State.
Signature:

```
void destroy (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: void

RPC name: get_by_uuid

Overview: Get a reference to the VM instance with the specified UUID.
Signature:

```
(VM ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
string	uuid	UUID of object to return

Return Type: `VM ref`
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given VM.
Signature:

```
(VM record) get_record (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: `VM record`
all fields from the object

RPC name: get_by_name_label

Overview: Get all the VM instances with the given label.
Signature:

```
((VM ref) Set) get_by_name_label (session_id s, string label)
```

Arguments:

type	name	description
string	label	label of object to return

Return Type: `(VM ref) Set`
references to objects with matching names

2.9 Class: VM_metrics

2.9.1 Fields for class: VM_metrics

Name	**VM_metrics**		
Description	*The metrics associated with a VM.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RO_{run}	`memory/actual`	int	Guest's actual memory (bytes)
RO_{run}	`VCPUs/number`	int	Current number of VCPUs
RO_{run}	`VCPUs/utilisation`	(int → float) Map	Utilisation for all of guest's current VCPUs
RO_{run}	`VCPUs/CPU`	(int → int) Map	VCPU to PCPU map
RO_{run}	`VCPUs/params`	(string → string) Map	The live equivalent to VM.VCPUs_params
RO_{run}	`VCPUs/flags`	(int → string Set) Map	CPU flags (blocked,online,running)
RO_{run}	`state`	string Set	The state of the guest, eg blocked, dying etc
RO_{run}	`start_time`	datetime	Time at which this VM was last booted
RO_{run}	`install_time`	datetime	Time at which the VM was installed
RO_{run}	`last_updated`	datetime	Time at which this information was last updated

2.9.2 RPCs associated with class: VM_metrics

RPC name: get_all

Overview: Return a list of all the VM_metrics instances known to the system.
Signature:

```
((VM_metrics ref) Set) get_all (session_id s)
```

Return Type: `(VM_metrics ref) Set`
references to all objects

RPC name: get_all_records_where

Overview: Return a list of all the VM_metrics instances known to the system.
Signature:

```
((VM_metrics ref -> VM_metrics record) Map) get_all_records_where (session_id s, string expr)
```

Arguments:

type	name	description
`string`	expr	expression representing records to fetch

Return Type: `(VM_metrics ref` → `VM_metrics record) Map`
records of all matching objects

RPC name: get_uuid

Overview: Get the uuid field of the given VM_metrics.
Signature:

```
string get_uuid (session_id s, VM_metrics ref self)
```

Arguments:

type	name	description
VM_metrics ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_memory_actual

Overview: Get the memory/actual field of the given VM_metrics.
Signature:

```
int get_memory_actual (session_id s, VM_metrics ref self)
```

Arguments:

type	name	description
VM_metrics ref	self	reference to the object

Return Type: int
value of the field

RPC name: get_VCPUs_number

Overview: Get the VCPUs/number field of the given VM_metrics.
Signature:

```
int get_VCPUs_number (session_id s, VM_metrics ref self)
```

Arguments:

type	name	description
VM_metrics ref	self	reference to the object

Return Type: int
value of the field

RPC name: get_VCPUs_utilisation

Overview: Get the VCPUs/utilisation field of the given VM_metrics.
Signature:

```
((int -> float) Map) get_VCPUs_utilisation (session_id s, VM_metrics ref self)
```

Arguments:

type	name	description
VM_metrics ref	self	reference to the object

Return Type: (int → float) Map
value of the field

RPC name: get_VCPUs_CPU

Overview: Get the VCPUs/CPU field of the given VM_metrics.
Signature:

```
((int -> int) Map) get_VCPUs_CPU (session_id s, VM_metrics ref self)
```

Arguments:

type	name	description
VM_metrics ref	self	reference to the object

Return Type: (int → int) Map
value of the field

RPC name: get_VCPUs_params

Overview: Get the VCPUs/params field of the given VM_metrics.
Signature:

```
((string -> string) Map) get_VCPUs_params (session_id s, VM_metrics ref self)
```

Arguments:

type	name	description
VM_metrics ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: get_VCPUs_flags

Overview: Get the VCPUs/flags field of the given VM_metrics.
Signature:

```
((int -> string Set) Map) get_VCPUs_flags (session_id s, VM_metrics ref self)
```

Arguments:

type	name	description
VM_metrics ref	self	reference to the object

Return Type: (int → string Set) Map
value of the field

RPC name: get_state

Overview: Get the state field of the given VM_metrics.
Signature:

```
(string Set) get_state (session_id s, VM_metrics ref self)
```

Arguments:

type	name	description
VM_metrics ref	self	reference to the object

Return Type: string Set
value of the field

RPC name: get_start_time

Overview: Get the start_time field of the given VM_metrics.
Signature:

```
datetime get_start_time (session_id s, VM_metrics ref self)
```

Arguments:

type	name	description
VM_metrics ref	self	reference to the object

Return Type: datetime
value of the field

RPC name: get_install_time

Overview: Get the install_time field of the given VM_metrics.
Signature:

```
datetime get_install_time (session_id s, VM_metrics ref self)
```

Arguments:

type	name	description
VM_metrics ref	self	reference to the object

Return Type: datetime
value of the field

RPC name: get_last_updated

Overview: Get the last_updated field of the given VM_metrics.
Signature:

```
datetime get_last_updated (session_id s, VM_metrics ref self)
```

Arguments:

type	name	description
VM_metrics ref	self	reference to the object

Return Type: datetime
value of the field

RPC name: get_by_uuid

Overview: Get a reference to the VM_metrics instance with the specified UUID.
Signature:

```
(VM_metrics ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
string	uuid	UUID of object to return

Return Type: VM_metrics ref
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given VM_metrics.
Signature:

```
(VM_metrics record) get_record (session_id s, VM_metrics ref self)
```

Arguments:

type	name	description
VM_metrics ref	self	reference to the object

Return Type: VM_metrics record
all fields from the object

2.10 Class: VM_guest_metrics

2.10.1 Fields for class: VM_guest_metrics

Name	**VM_guest_metrics**		
Description	*The metrics reported by the guest (as opposed to inferred from outside).*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RO_{run}	`os_version`	(string → string) Map	version of the OS
RO_{run}	`PV_drivers_version`	(string → string) Map	version of the PV drivers
RO_{run}	`PV_drivers_up_to_date`	bool	true if the PV drivers appear to be up to date
RO_{run}	`memory`	(string → string) Map	free/used/total memory
RO_{run}	`disks`	(string → string) Map	disk configuration/free space
RO_{run}	`networks`	(string → string) Map	network configuration
RO_{run}	`other`	(string → string) Map	anything else
RO_{run}	`last_updated`	datetime	Time at which this information was last updated

2.10.2 RPCs associated with class: VM_guest_metrics

RPC name: get_all

Overview: Return a list of all the VM_guest_metrics instances known to the system.
Signature:

```
((VM_guest_metrics ref) Set) get_all (session_id s)
```

Return Type: `(VM_guest_metrics ref) Set`
references to all objects

RPC name: get_all_records_where

Overview: Return a list of all the VM_guest_metrics instances known to the system.
Signature:

```
((VM_guest_metrics ref -> VM_guest_metrics record) Map) get_all_records_where (session_id s, string ex
```

Arguments:

type	name	description
`string`	expr	expression representing records to fetch

Return Type: `(VM_guest_metrics ref → VM_guest_metrics record) Map`
records of all matching objects

RPC name: get_uuid

Overview: Get the uuid field of the given VM_guest_metrics.
Signature:

```
string get_uuid (session_id s, VM_guest_metrics ref self)
```

Arguments:

type	name	description
VM_guest_metrics ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_os_version

Overview: Get the os_version field of the given VM_guest_metrics.
Signature:

```
((string -> string) Map) get_os_version (session_id s, VM_guest_metrics ref self)
```

Arguments:

type	name	description
VM_guest_metrics ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: get_PV_drivers_version

Overview: Get the PV_drivers_version field of the given VM_guest_metrics.
Signature:

```
((string -> string) Map) get_PV_drivers_version (session_id s, VM_guest_metrics ref self)
```

Arguments:

type	name	description
VM_guest_metrics ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: get_PV_drivers_up_to_date

Overview: Get the PV_drivers_up_to_date field of the given VM_guest_metrics.
Signature:

```
bool get_PV_drivers_up_to_date (session_id s, VM_guest_metrics ref self)
```

Arguments:

type	name	description
VM_guest_metrics ref	self	reference to the object

Return Type: bool
value of the field

RPC name: get_memory

Overview: Get the memory field of the given VM_guest_metrics.
Signature:

```
((string -> string) Map) get_memory (session_id s, VM_guest_metrics ref self)
```

Arguments:

type	name	description
VM_guest_metrics ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: get_disks

Overview: Get the disks field of the given VM_guest_metrics.
Signature:

```
((string -> string) Map) get_disks (session_id s, VM_guest_metrics ref self)
```

Arguments:

type	name	description
VM_guest_metrics ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: get_networks

Overview: Get the networks field of the given VM_guest_metrics.
Signature:

```
((string -> string) Map) get_networks (session_id s, VM_guest_metrics ref self)
```

Arguments:

type	name	description
VM_guest_metrics ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: get_other

Overview: Get the other field of the given VM_guest_metrics.
Signature:

```
((string -> string) Map) get_other (session_id s, VM_guest_metrics ref self)
```

Arguments:

type	name	description
VM_guest_metrics ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: get_last_updated

Overview: Get the last_updated field of the given VM_guest_metrics.
Signature:

```
datetime get_last_updated (session_id s, VM_guest_metrics ref self)
```

Arguments:

type	name	description
VM_guest_metrics ref	self	reference to the object

Return Type: datetime
value of the field

RPC name: get_by_uuid

Overview: Get a reference to the VM_guest_metrics instance with the specified UUID.
Signature:

```
(VM_guest_metrics ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
string	uuid	UUID of object to return

Return Type: VM_guest_metrics ref
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given VM_guest_metrics.
Signature:

```
(VM_guest_metrics record) get_record (session_id s, VM_guest_metrics ref self)
```

Arguments:

type	name	description
VM_guest_metrics ref	self	reference to the object

Return Type: VM_guest_metrics record
all fields from the object

2.11 Class: host

2.11.1 Fields for class: host

Name	**host**		
Description	*A physical host.*		
Quals	Field	Type	Description
RO_{run}	uuid	string	unique identifier/object
RW	name/label	string	a human-readable name
RW	name/description	string	a notes field conta readable description
RO_{run}	allowed_operations	(host_allowed_operations) Set	list of the operations al state
RO_{run}	current_operations	(string → host_allowed_operations) Map	Map of task reference t eration enumeration
RO_{run}	API_version/major	int	major version number
RO_{run}	API_version/minor	int	minor version number
RO_{run}	API_version/vendor	string	identification of vendor
RO_{run}	API_version/vendor_implementation	(string → string) Map	details of vendor implen
RO_{run}	enabled	bool	True if the host is curre
RO_{ins}	software_version	(string → string) Map	version strings
RW	other_config	(string → string) Map	additional configuration
RO_{ins}	capabilities	string Set	Xen capabilities
RO_{run}	cpu_configuration	(string → string) Map	The CPU configu this host. May c such as "nr_nodes ets_per_node", "cores_pe "threads_per_core"
RO_{run}	sched_policy	string	Scheduler policy current this host
RO_{run}	supported_bootloaders	string Set	a list of the bootloaders the machine
RO_{run}	resident_VMs	(VM ref) Set	list of VMs currently res
RW	logging	(string → string) Map	logging configuration
RO_{run}	PIFs	(PIF ref) Set	physical network interfa
RW	suspend_image_sr	SR ref	The SR in which VDIs images are created
RW	crash_dump_sr	SR ref	The SR in which VD dumps are created
RO_{run}	crashdumps	(host_crashdump ref) Set	Set of host crash dumps
RO_{run}	patches	(host_patch ref) Set	Set of host patches
RO_{run}	PBDs	(PBD ref) Set	physical blockdevices
RO_{run}	host_CPUs	(host_cpu ref) Set	The physical CPUs on t
RW	hostname	string	The hostname of this ho
RW	address	string	The address by which t be contacted from any the pool
RO_{run}	metrics	host_metrics ref	metrics associated with
RO_{run}	license_params	(string → string) Map	The key/value pairs rea cense file

2.11.2 RPCs associated with class: host

RPC name: disable

Overview: Puts the host into a state in which no new VMs can be started. Currently active VMs on the host continue to execute.
Signature:

```
void disable (session_id s, host ref host)
```

Arguments:

type	name	description
host ref	host	The Host to disable

Return Type: void

RPC name: enable

Overview: Puts the host into a state in which new VMs can be started.
Signature:

```
void enable (session_id s, host ref host)
```

Arguments:

type	name	description
host ref	host	The Host to enable

Return Type: void

RPC name: shutdown

Overview: Shutdown the host. (This function can only be called if there are no currently running VMs on the host and it is disabled.).
Signature:

```
void shutdown (session_id s, host ref host)
```

Arguments:

type	name	description
host ref	host	The Host to shutdown

Return Type: void

RPC name: reboot

Overview: Reboot the host. (This function can only be called if there are no currently running VMs on the host and it is disabled.).
Signature:

```
void reboot (session_id s, host ref host)
```

Arguments:

type	name	description
`host ref`	host	The Host to reboot

Return Type: `void`

RPC name: dmesg

Overview: Get the host xen dmesg.
Signature:

```
string dmesg (session_id s, host ref host)
```

Arguments:

type	name	description
`host ref`	host	The Host to query

Return Type: `string`
dmesg string

RPC name: dmesg_clear

Overview: Get the host xen dmesg, and clear the buffer.
Signature:

```
string dmesg_clear (session_id s, host ref host)
```

Arguments:

type	name	description
`host ref`	host	The Host to query

Return Type: `string`
dmesg string

RPC name: get_log

Overview: Get the host's log file.
Signature:

```
string get_log (session_id s, host ref host)
```

Arguments:

type	name	description
`host ref`	host	The Host to query

Return Type: `string`
The contents of the host's primary log file

RPC name: send_debug_keys

Overview: Inject the given string as debugging keys into Xen.
Signature:

```
void send_debug_keys (session_id s, host ref host, string keys)
```

Arguments:

type	name	description
host ref	host	The host
string	keys	The keys to send

Return Type: void

RPC name: bugreport_upload

Overview: Run xen-bugtool –yestoall and upload the output to XenSource support.
Signature:

```
void bugreport_upload (session_id s, host ref host, string url, (string -> string) Map options)
```

Arguments:

type	name	description
host ref	host	The host on which to run xen-bugtool
string	url	The URL to upload to
(string → string) Map	options	Extra configuration operations

Return Type: void

RPC name: list_methods

Overview: List all supported methods.
Signature:

```
(string Set) list_methods (session_id s)
```

Return Type: string Set
The name of every supported method.

RPC name: license_apply

Overview: Apply a new license to a host.
Signature:

```
void license_apply (session_id s, host ref host, string contents)
```

Arguments:

type	name	description
host ref	host	The host to upload the license to
string	contents	The contents of the license file, base64 encoded

Return Type: `void`

RPC name: create

Overview: Create a new host record.
Signature:

```
(host ref) create (session_id s, string uuid, string name_label, string name_description, string hostn
```

Arguments:

type	name	description
`string`	uuid	unique identifier/object reference
`string`	name_label	The name of the new storage repository
`string`	name_description	The description of the new storage repository
`string`	hostname	Hostname
`string`	address	An address by which this host can be contacted by other members in its pool

Return Type: `host ref`
Reference to the newly created host object.

RPC name: destroy

Overview: Destroy specified host record in database.
Signature:

```
void destroy (session_id s, host ref self)
```

Arguments:

type	name	description
`host ref`	self	The host record to remove

Return Type: `void`

RPC name: get_all

Overview: Return a list of all the hosts known to the system.
Signature:

```
((host ref) Set) get_all (session_id s)
```

Return Type: `(host ref) Set`
references to all objects

RPC name: get_all_records_where

Overview: Return a list of all the hosts known to the system.
Signature:

```
((host ref -> host record) Map) get_all_records_where (session_id s, string expr)
```

Arguments:

type	name	description
string	expr	expression representing records to fetch

Return Type: `(host ref → host record) Map`
records of all matching objects

RPC name: get_uuid

Overview: Get the uuid field of the given host.
Signature:

```
string get_uuid (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_name_label

Overview: Get the name/label field of the given host.
Signature:

```
string get_name_label (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: `string`
value of the field

RPC name: set_name_label

Overview: Set the name/label field of the given host.
Signature:

```
void set_name_label (session_id s, host ref self, string value)
```

Arguments:

type	name	description
host ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_name_description

Overview: Get the name/description field of the given host.
Signature:

```
string get_name_description (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: string
value of the field

RPC name: set_name_description

Overview: Set the name/description field of the given host.
Signature:

```
void set_name_description (session_id s, host ref self, string value)
```

Arguments:

type	name	description
host ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_allowed_operations

Overview: Get the allowed_operations field of the given host.
Signature:

```
((host_allowed_operations) Set) get_allowed_operations (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: (host_allowed_operations) Set
value of the field

RPC name: get_current_operations

Overview: Get the current_operations field of the given host.
Signature:

```
((string -> host_allowed_operations) Map) get_current_operations (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: (string → host_allowed_operations) Map
value of the field

RPC name: get_API_version_major

Overview: Get the API_version/major field of the given host.
Signature:

```
int get_API_version_major (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: int
value of the field

RPC name: get_API_version_minor

Overview: Get the API_version/minor field of the given host.
Signature:

```
int get_API_version_minor (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: int
value of the field

RPC name: get_API_version_vendor

Overview: Get the API_version/vendor field of the given host.
Signature:

```
string get_API_version_vendor (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_API_version_vendor_implementation

Overview: Get the API_version/vendor_implementation field of the given host.
Signature:

```
((string -> string) Map) get_API_version_vendor_implementation (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: get_enabled

Overview: Get the enabled field of the given host.
Signature:

```
bool get_enabled (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: bool
value of the field

RPC name: get_software_version

Overview: Get the software_version field of the given host.
Signature:

```
((string -> string) Map) get_software_version (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: get_other_config

Overview: Get the other_config field of the given host.
Signature:

```
((string -> string) Map) get_other_config (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: set_other_config

Overview: Set the other_config field of the given host.
Signature:

```
void set_other_config (session_id s, host ref self, (string -> string) Map value)
```

Arguments:

type	name	description
host ref	self	reference to the object
(string → string) Map	value	New value to set

Return Type: void

RPC name: add_to_other_config

Overview: Add the given key-value pair to the other_config field of the given host.
Signature:

```
void add_to_other_config (session_id s, host ref self, string key, string value)
```

Arguments:

type	name	description
host ref	self	reference to the object
string	key	Key to add
string	value	Value to add

Return Type: void

RPC name: remove_from_other_config

Overview: Remove the given key and its corresponding value from the other_config field of the given host. If the key is not in that Map, then do nothing.
Signature:

```
void remove_from_other_config (session_id s, host ref self, string key)
```

Arguments:

type	name	description
host ref	self	reference to the object
string	key	Key to remove

Return Type: void

RPC name: get_capabilities

Overview: Get the capabilities field of the given host.
Signature:

```
(string Set) get_capabilities (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: string Set
value of the field

RPC name: get_cpu_configuration

Overview: Get the cpu_configuration field of the given host.
Signature:

```
((string -> string) Map) get_cpu_configuration (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: get_sched_policy

Overview: Get the sched_policy field of the given host.
Signature:

```
string get_sched_policy (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_supported_bootloaders

Overview: Get the supported_bootloaders field of the given host.
Signature:

```
(string Set) get_supported_bootloaders (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: string Set
value of the field

RPC name: get_resident_VMs

Overview: Get the resident_VMs field of the given host.
Signature:

```
((VM ref) Set) get_resident_VMs (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: (VM ref) Set
value of the field

RPC name: get_logging

Overview: Get the logging field of the given host.
Signature:

```
((string -> string) Map) get_logging (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: set_logging

Overview: Set the logging field of the given host.
Signature:

```
void set_logging (session_id s, host ref self, (string -> string) Map value)
```

Arguments:

type	name	description
`host ref`	self	reference to the object
`(string → string) Map`	value	New value to set

Return Type: `void`

RPC name: add_to_logging

Overview: Add the given key-value pair to the logging field of the given host.
Signature:

```
void add_to_logging (session_id s, host ref self, string key, string value)
```

Arguments:

type	name	description
`host ref`	self	reference to the object
`string`	key	Key to add
`string`	value	Value to add

Return Type: `void`

RPC name: remove_from_logging

Overview: Remove the given key and its corresponding value from the logging field of the given host. If the key is not in that Map, then do nothing.
Signature:

```
void remove_from_logging (session_id s, host ref self, string key)
```

Arguments:

type	name	description
`host ref`	self	reference to the object
`string`	key	Key to remove

Return Type: `void`

RPC name: get_PIFs

Overview: Get the PIFs field of the given host.
Signature:

```
((PIF ref) Set) get_PIFs (session_id s, host ref self)
```

Arguments:

type	name	description
`host ref`	self	reference to the object

Return Type: `(PIF ref) Set`

value of the field

RPC name: get_suspend_image_sr

Overview: Get the suspend_image_sr field of the given host.
Signature:

```
(SR ref) get_suspend_image_sr (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: SR ref
value of the field

RPC name: set_suspend_image_sr

Overview: Set the suspend_image_sr field of the given host.
Signature:

```
void set_suspend_image_sr (session_id s, host ref self, SR ref value)
```

Arguments:

type	name	description
host ref	self	reference to the object
SR ref	value	New value to set

Return Type: void

RPC name: get_crash_dump_sr

Overview: Get the crash_dump_sr field of the given host.
Signature:

```
(SR ref) get_crash_dump_sr (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: SR ref
value of the field

RPC name: set_crash_dump_sr

Overview: Set the crash_dump_sr field of the given host.
Signature:

```
void set_crash_dump_sr (session_id s, host ref self, SR ref value)
```

Arguments:

type	name	description
host ref	self	reference to the object
SR ref	value	New value to set

Return Type: void

RPC name: get_crashdumps

Overview: Get the crashdumps field of the given host.
Signature:

```
((host_crashdump ref) Set) get_crashdumps (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: (host_crashdump ref) Set
value of the field

RPC name: get_patches

Overview: Get the patches field of the given host.
Signature:

```
((host_patch ref) Set) get_patches (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: (host_patch ref) Set
value of the field

RPC name: get_PBDs

Overview: Get the PBDs field of the given host.
Signature:

```
((PBD ref) Set) get_PBDs (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: (PBD ref) Set
value of the field

RPC name: get_host_CPUs

Overview: Get the host_CPUs field of the given host.
Signature:

```
((host_cpu ref) Set) get_host_CPUs (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: (host_cpu ref) Set
value of the field

RPC name: get_hostname

Overview: Get the hostname field of the given host.
Signature:

```
string get_hostname (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: string
value of the field

RPC name: set_hostname

Overview: Set the hostname field of the given host.
Signature:

```
void set_hostname (session_id s, host ref self, string value)
```

Arguments:

type	name	description
host ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_address

Overview: Get the address field of the given host.
Signature:

```
string get_address (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: string
value of the field

RPC name: set_address

Overview: Set the address field of the given host.
Signature:

```
void set_address (session_id s, host ref self, string value)
```

Arguments:

type	name	description
host ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_metrics

Overview: Get the metrics field of the given host.
Signature:

```
(host_metrics ref) get_metrics (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	reference to the object

Return Type: host_metrics ref
value of the field

RPC name: get_license_params

Overview: Get the license_params field of the given host.
Signature:

```
((string -> string) Map) get_license_params (session_id s, host ref self)
```

Arguments:

type	name	description
`host ref`	self	reference to the object

Return Type: `(string → string) Map`
value of the field

RPC name: get_by_uuid

Overview: Get a reference to the host instance with the specified UUID.
Signature:

```
(host ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
`string`	uuid	UUID of object to return

Return Type: `host ref`
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given host.
Signature:

```
(host record) get_record (session_id s, host ref self)
```

Arguments:

type	name	description
`host ref`	self	reference to the object

Return Type: `host record`
all fields from the object

RPC name: get_by_name_label

Overview: Get all the host instances with the given label.
Signature:

```
((host ref) Set) get_by_name_label (session_id s, string label)
```

Arguments:

type	name	description
`string`	label	label of object to return

Return Type: `(host ref) Set`
references to objects with matching names

2.12 Class: host_crashdump

2.12.1 Fields for class: host_crashdump

Name	**host_crashdump**		
Description	*Represents a host crash dump.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RO_{ins}	`host`	host ref	Host the crashdump relates to
RO_{run}	`timestamp`	datetime	Time the crash happened
RO_{run}	`size`	int	Size of the crashdump

2.12.2 RPCs associated with class: host_crashdump

RPC name: destroy

Overview: Destroy specified host crash dump, removing it from the disk.
Signature:

```
void destroy (session_id s, host_crashdump ref self)
```

Arguments:

type	name	description
`host_crashdump ref`	self	The host crashdump to destroy

Return Type: `void`

RPC name: upload

Overview: Upload the specified host crash dump to a specified URL.
Signature:

```
void upload (session_id s, host_crashdump ref self, string url, (string -> string) Map options)
```

Arguments:

type	name	description
`host_crashdump ref`	self	The host crashdump to upload
`string`	url	The URL to upload to
`(string → string) Map`	options	Extra configuration operations

Return Type: `void`

RPC name: get_all

Overview: Return a list of all the host_crashdumps known to the system.
Signature:

```
((host_crashdump ref) Set) get_all (session_id s)
```

Return Type: `(host_crashdump ref) Set`
references to all objects

RPC name: get_all_records_where

Overview: Return a list of all the host_crashdumps known to the system.
Signature:

```
((host_crashdump ref -> host_crashdump record) Map) get_all_records_where (session_id s, string expr)
```

Arguments:

type	name	description
`string`	expr	expression representing records to fetch

Return Type: `(host_crashdump ref → host_crashdump record) Map`
records of all matching objects

RPC name: get_uuid

Overview: Get the uuid field of the given host_crashdump.
Signature:

```
string get_uuid (session_id s, host_crashdump ref self)
```

Arguments:

type	name	description
`host_crashdump ref`	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_host

Overview: Get the host field of the given host_crashdump.
Signature:

```
(host ref) get_host (session_id s, host_crashdump ref self)
```

Arguments:

type	name	description
`host_crashdump ref`	self	reference to the object

Return Type: `host ref`
value of the field

RPC name: get_timestamp

Overview: Get the timestamp field of the given host_crashdump.
Signature:

```
datetime get_timestamp (session_id s, host_crashdump ref self)
```

Arguments:

type	name	description
host_crashdump ref	self	reference to the object

Return Type: datetime
value of the field

RPC name: get_size

Overview: Get the size field of the given host_crashdump.
Signature:

```
int get_size (session_id s, host_crashdump ref self)
```

Arguments:

type	name	description
host_crashdump ref	self	reference to the object

Return Type: int
value of the field

RPC name: get_by_uuid

Overview: Get a reference to the host_crashdump instance with the specified UUID.
Signature:

```
(host_crashdump ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
string	uuid	UUID of object to return

Return Type: host_crashdump ref
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given host_crashdump.
Signature:

```
(host_crashdump record) get_record (session_id s, host_crashdump ref self)
```

Arguments:

type	name	description
`host_crashdump ref`	self	reference to the object

Return Type: `host_crashdump record`
all fields from the object

2.13 Class: host_patch

2.13.1 Fields for class: host_patch

Name	**host_patch**		
Description	*Represents a patch stored on a server.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RO_{ins}	`name/label`	string	a human-readable name
RO_{ins}	`name/description`	string	a notes field containg human-readable description
RO_{ins}	`version`	string	Patch version number
RO_{ins}	`host`	host ref	Host the patch relates to
RO_{run}	`applied`	bool	True if the patch has been applied
RO_{run}	`timestamp_applied`	datetime	Time the patch was applied
RO_{run}	`size`	int	Size of the patch

2.13.2 RPCs associated with class: host_patch

RPC name: destroy

Overview: Destroy the specified host patch, removing it from the disk. This does NOT reverse the patch.
Signature:

```
void destroy (session_id s, host_patch ref self)
```

Arguments:

type	name	description
`host_patch ref`	self	The patch to destroy

Return Type: `void`

RPC name: apply

Overview: Apply the selected patch and return its output.
Signature:

```
string apply (session_id s, host_patch ref self)
```

Arguments:

type	name	description
`host_patch ref`	self	The patch to apply

Return Type: `string`
the output of the patch application process

RPC name: get_all

Overview: Return a list of all the host_patchs known to the system.
Signature:

```
((host_patch ref) Set) get_all (session_id s)
```

Return Type: `(host_patch ref) Set`
references to all objects

RPC name: get_all_records_where

Overview: Return a list of all the host_patchs known to the system.
Signature:

```
((host_patch ref -> host_patch record) Map) get_all_records_where (session_id s, string expr)
```

Arguments:

type	name	description
`string`	expr	expression representing records to fetch

Return Type: `(host_patch ref → host_patch record) Map`
records of all matching objects

RPC name: get_uuid

Overview: Get the uuid field of the given host_patch.
Signature:

```
string get_uuid (session_id s, host_patch ref self)
```

Arguments:

type	name	description
`host_patch ref`	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_name_label

Overview: Get the name/label field of the given host_patch.
Signature:

```
string get_name_label (session_id s, host_patch ref self)
```

Arguments:

type	name	description
`host_patch ref`	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_name_description

Overview: Get the name/description field of the given host_patch.
Signature:

```
string get_name_description (session_id s, host_patch ref self)
```

Arguments:

type	name	description
host_patch ref	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_version

Overview: Get the version field of the given host_patch.
Signature:

```
string get_version (session_id s, host_patch ref self)
```

Arguments:

type	name	description
host_patch ref	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_host

Overview: Get the host field of the given host_patch.
Signature:

```
(host ref) get_host (session_id s, host_patch ref self)
```

Arguments:

type	name	description
host_patch ref	self	reference to the object

Return Type: `host ref`
value of the field

RPC name: get_applied

Overview: Get the applied field of the given host_patch.
Signature:

```
bool get_applied (session_id s, host_patch ref self)
```

Arguments:

type	name	description
`host_patch ref`	self	reference to the object

Return Type: `bool`
value of the field

RPC name: get_timestamp_applied

Overview: Get the timestamp_applied field of the given host_patch.
Signature:

```
datetime get_timestamp_applied (session_id s, host_patch ref self)
```

Arguments:

type	name	description
`host_patch ref`	self	reference to the object

Return Type: `datetime`
value of the field

RPC name: get_size

Overview: Get the size field of the given host_patch.
Signature:

```
int get_size (session_id s, host_patch ref self)
```

Arguments:

type	name	description
`host_patch ref`	self	reference to the object

Return Type: `int`
value of the field

RPC name: get_by_uuid

Overview: Get a reference to the host_patch instance with the specified UUID.
Signature:

```
(host_patch ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
`string`	uuid	UUID of object to return

Return Type: `host_patch ref`
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given host_patch.
Signature:

```
(host_patch record) get_record (session_id s, host_patch ref self)
```

Arguments:

type	name	description
host_patch ref	self	reference to the object

Return Type: `host_patch record`
all fields from the object

RPC name: get_by_name_label

Overview: Get all the host_patch instances with the given label.
Signature:

```
((host_patch ref) Set) get_by_name_label (session_id s, string label)
```

Arguments:

type	name	description
string	label	label of object to return

Return Type: `(host_patch ref) Set`
references to objects with matching names

2.14 Class: host_metrics

2.14.1 Fields for class: host_metrics

Name	**host_metrics**		
Description	*The metrics associated with a host.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RO_{run}	`memory/total`	int	Host's total memory (bytes)
RO_{run}	`memory/free`	int	Host's free memory (bytes)
RO_{run}	`live`	bool	Pool master thinks this host is live
RO_{run}	`last_updated`	datetime	Time at which this information was last updated

2.14.2 RPCs associated with class: host_metrics

RPC name: get_all

Overview: Return a list of all the host_metrics instances known to the system.
Signature:

```
((host_metrics ref) Set) get_all (session_id s)
```

Return Type: `(host_metrics ref) Set`
references to all objects

RPC name: get_all_records_where

Overview: Return a list of all the host_metrics instances known to the system.
Signature:

```
((host_metrics ref -> host_metrics record) Map) get_all_records_where (session_id s, string expr)
```

Arguments:

type	name	description
`string`	expr	expression representing records to fetch

Return Type: `(host_metrics ref → host_metrics record) Map`
records of all matching objects

RPC name: get_uuid

Overview: Get the uuid field of the given host_metrics.
Signature:

```
string get_uuid (session_id s, host_metrics ref self)
```

Arguments:

type	name	description
`host_metrics ref`	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_memory_total

Overview: Get the memory/total field of the given host_metrics.
Signature:

```
int get_memory_total (session_id s, host_metrics ref self)
```

Arguments:

type	name	description
`host_metrics ref`	self	reference to the object

Return Type: `int`
value of the field

RPC name: get_memory_free

Overview: Get the memory/free field of the given host_metrics.
Signature:

```
int get_memory_free (session_id s, host_metrics ref self)
```

Arguments:

type	name	description
`host_metrics ref`	self	reference to the object

Return Type: `int`
value of the field

RPC name: get_live

Overview: Get the live field of the given host_metrics.
Signature:

```
bool get_live (session_id s, host_metrics ref self)
```

Arguments:

type	name	description
`host_metrics ref`	self	reference to the object

Return Type: `bool`
value of the field

RPC name: get_last_updated

Overview: Get the last_updated field of the given host_metrics.
Signature:

```
datetime get_last_updated (session_id s, host_metrics ref self)
```

Arguments:

type	name	description
host_metrics ref	self	reference to the object

Return Type: datetime
value of the field

RPC name: get_by_uuid

Overview: Get a reference to the host_metrics instance with the specified UUID.
Signature:

```
(host_metrics ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
string	uuid	UUID of object to return

Return Type: host_metrics ref
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given host_metrics.
Signature:

```
(host_metrics record) get_record (session_id s, host_metrics ref self)
```

Arguments:

type	name	description
host_metrics ref	self	reference to the object

Return Type: host_metrics record
all fields from the object

2.15 Class: host_cpu

2.15.1 Fields for class: host_cpu

Name	**host_cpu**		
Description	*A physical CPU.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RO_{run}	`host`	host ref	the host the CPU is in
RO_{run}	`number`	int	the number of the physical CPU within the host
RO_{run}	`vendor`	string	the vendor of the physical CPU
RO_{run}	`speed`	int	the speed of the physical CPU
RO_{run}	`modelname`	string	the model name of the physical CPU
RO_{run}	`family`	int	the family (number) of the physical CPU
RO_{run}	`model`	int	the model number of the physical CPU
RO_{run}	`stepping`	string	the stepping of the physical CPU
RO_{run}	`flags`	string	the flags of the physical CPU (a decoded version of the features field)
RO_{run}	`features`	string	the physical CPU feature bitmap
RO_{run}	`utilisation`	float	the current CPU utilisation

2.15.2 RPCs associated with class: host_cpu

RPC name: get_all

Overview: Return a list of all the host_cpus known to the system.
Signature:

```
((host_cpu ref) Set) get_all (session_id s)
```

Return Type: `(host_cpu ref) Set`
references to all objects

RPC name: get_all_records_where

Overview: Return a list of all the host_cpus known to the system.
Signature:

```
((host_cpu ref -> host_cpu record) Map) get_all_records_where (session_id s, string expr)
```

Arguments:

type	name	description
`string`	expr	expression representing records to fetch

Return Type: `(host_cpu ref → host_cpu record) Map`
records of all matching objects

RPC name: get_uuid

Overview: Get the uuid field of the given host_cpu.
Signature:

```
string get_uuid (session_id s, host_cpu ref self)
```

Arguments:

type	name	description
host_cpu ref	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_host

Overview: Get the host field of the given host_cpu.
Signature:

```
(host ref) get_host (session_id s, host_cpu ref self)
```

Arguments:

type	name	description
host_cpu ref	self	reference to the object

Return Type: `host ref`
value of the field

RPC name: get_number

Overview: Get the number field of the given host_cpu.
Signature:

```
int get_number (session_id s, host_cpu ref self)
```

Arguments:

type	name	description
host_cpu ref	self	reference to the object

Return Type: `int`
value of the field

RPC name: get_vendor

Overview: Get the vendor field of the given host_cpu.
Signature:

```
string get_vendor (session_id s, host_cpu ref self)
```

Arguments:

type	name	description
`host_cpu ref`	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_speed

Overview: Get the speed field of the given host_cpu.
Signature:

```
int get_speed (session_id s, host_cpu ref self)
```

Arguments:

type	name	description
`host_cpu ref`	self	reference to the object

Return Type: `int`
value of the field

RPC name: get_modelname

Overview: Get the modelname field of the given host_cpu.
Signature:

```
string get_modelname (session_id s, host_cpu ref self)
```

Arguments:

type	name	description
`host_cpu ref`	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_family

Overview: Get the family field of the given host_cpu.
Signature:

```
int get_family (session_id s, host_cpu ref self)
```

Arguments:

type	name	description
`host_cpu ref`	self	reference to the object

Return Type: `int`
value of the field

RPC name: get_model

Overview: Get the model field of the given host_cpu.
Signature:

```
int get_model (session_id s, host_cpu ref self)
```

Arguments:

type	name	description
host_cpu ref	self	reference to the object

Return Type: int
value of the field

RPC name: get_stepping

Overview: Get the stepping field of the given host_cpu.
Signature:

```
string get_stepping (session_id s, host_cpu ref self)
```

Arguments:

type	name	description
host_cpu ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_flags

Overview: Get the flags field of the given host_cpu.
Signature:

```
string get_flags (session_id s, host_cpu ref self)
```

Arguments:

type	name	description
host_cpu ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_features

Overview: Get the features field of the given host_cpu.
Signature:

```
string get_features (session_id s, host_cpu ref self)
```

Arguments:

type	name	description
`host_cpu ref`	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_utilisation

Overview: Get the utilisation field of the given host_cpu.
Signature:

```
float get_utilisation (session_id s, host_cpu ref self)
```

Arguments:

type	name	description
`host_cpu ref`	self	reference to the object

Return Type: `float`
value of the field

RPC name: get_by_uuid

Overview: Get a reference to the host_cpu instance with the specified UUID.
Signature:

```
(host_cpu ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
`string`	uuid	UUID of object to return

Return Type: `host_cpu ref`
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given host_cpu.
Signature:

```
(host_cpu record) get_record (session_id s, host_cpu ref self)
```

Arguments:

type	name	description
`host_cpu ref`	self	reference to the object

Return Type: `host_cpu record`
all fields from the object

2.16 Class: network

2.16.1 Fields for class: network

Name	**network**		
Description	*A virtual network.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RW	`name/label`	string	a human-readable name
RW	`name/description`	string	a notes field containg human-readable description
RO_{run}	`allowed_operations`	(network_operations) Set	list of the operations allowed in this state
RO_{run}	`current_operations`	(string → network_operations) Map	Map of task reference to current operation enumeration
RO_{run}	`VIFs`	(VIF ref) Set	list of connected vifs
RO_{run}	`PIFs`	(PIF ref) Set	list of connected pifs
RW	`other_config`	(string → string) Map	additional configuration
RO_{run}	`bridge`	string	name of the bridge corresponding to this network on the local host

2.16.2 RPCs associated with class: network

RPC name: get_all

Overview: Return a list of all the networks known to the system.
Signature:

```
((network ref) Set) get_all (session_id s)
```

Return Type: `(network ref) Set`
references to all objects

RPC name: get_all_records_where

Overview: Return a list of all the networks known to the system.
Signature:

```
((network ref -> network record) Map) get_all_records_where (session_id s, string expr)
```

Arguments:

type	name	description
`string`	expr	expression representing records to fetch

Return Type: `(network ref → network record) Map`
records of all matching objects

RPC name: get_uuid

Overview: Get the uuid field of the given network.
Signature:

```
string get_uuid (session_id s, network ref self)
```

Arguments:

type	name	description
network ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_name_label

Overview: Get the name/label field of the given network.
Signature:

```
string get_name_label (session_id s, network ref self)
```

Arguments:

type	name	description
network ref	self	reference to the object

Return Type: string
value of the field

RPC name: set_name_label

Overview: Set the name/label field of the given network.
Signature:

```
void set_name_label (session_id s, network ref self, string value)
```

Arguments:

type	name	description
network ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_name_description

Overview: Get the name/description field of the given network.
Signature:

```
string get_name_description (session_id s, network ref self)
```

Arguments:

type	name	description
network ref	self	reference to the object

Return Type: string

value of the field

RPC name: set_name_description

Overview: Set the name/description field of the given network.
Signature:

```
void set_name_description (session_id s, network ref self, string value)
```

Arguments:

type	name	description
network ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_allowed_operations

Overview: Get the allowed_operations field of the given network.
Signature:

```
((network_operations) Set) get_allowed_operations (session_id s, network ref self)
```

Arguments:

type	name	description
network ref	self	reference to the object

Return Type: (network_operations) Set
value of the field

RPC name: get_current_operations

Overview: Get the current_operations field of the given network.
Signature:

```
((string -> network_operations) Map) get_current_operations (session_id s, network ref self)
```

Arguments:

type	name	description
network ref	self	reference to the object

Return Type: (string → network_operations) Map
value of the field

RPC name: get_VIFs

Overview: Get the VIFs field of the given network.
Signature:

```
((VIF ref) Set) get_VIFs (session_id s, network ref self)
```

Arguments:

type	name	description
network ref	self	reference to the object

Return Type: (VIF ref) Set
value of the field

RPC name: get_PIFs

Overview: Get the PIFs field of the given network.
Signature:

```
((PIF ref) Set) get_PIFs (session_id s, network ref self)
```

Arguments:

type	name	description
network ref	self	reference to the object

Return Type: (PIF ref) Set
value of the field

RPC name: get_other_config

Overview: Get the other_config field of the given network.
Signature:

```
((string -> string) Map) get_other_config (session_id s, network ref self)
```

Arguments:

type	name	description
network ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: set_other_config

Overview: Set the other_config field of the given network.
Signature:

```
void set_other_config (session_id s, network ref self, (string -> string) Map value)
```

Arguments:

type	name	description
network ref	self	reference to the object
(string → string) Map	value	New value to set

Return Type: void

RPC name: add_to_other_config

Overview: Add the given key-value pair to the other_config field of the given network.
Signature:

```
void add_to_other_config (session_id s, network ref self, string key, string value)
```

Arguments:

type	name	description
network ref	self	reference to the object
string	key	Key to add
string	value	Value to add

Return Type: void

RPC name: remove_from_other_config

Overview: Remove the given key and its corresponding value from the other_config field of the given network. If the key is not in that Map, then do nothing.
Signature:

```
void remove_from_other_config (session_id s, network ref self, string key)
```

Arguments:

type	name	description
network ref	self	reference to the object
string	key	Key to remove

Return Type: void

RPC name: get_bridge

Overview: Get the bridge field of the given network.
Signature:

```
string get_bridge (session_id s, network ref self)
```

Arguments:

type	name	description
network ref	self	reference to the object

Return Type: string

value of the field

RPC name: create

Overview: Create a new network instance, and return its handle.
Signature:

```
(network ref) create (session_id s, network record args)
```

Arguments:

type	name	description
network record	args	All constructor arguments

Return Type: network ref
reference to the newly created object

RPC name: destroy

Overview: Destroy the specified network instance.
Signature:

```
void destroy (session_id s, network ref self)
```

Arguments:

type	name	description
network ref	self	reference to the object

Return Type: void

RPC name: get_by_uuid

Overview: Get a reference to the network instance with the specified UUID.
Signature:

```
(network ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
string	uuid	UUID of object to return

Return Type: network ref
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given network.
Signature:

```
(network record) get_record (session_id s, network ref self)
```

Arguments:

type	name	description
network ref	self	reference to the object

Return Type: network record
all fields from the object

RPC name: get_by_name_label

Overview: Get all the network instances with the given label.
Signature:

```
((network ref) Set) get_by_name_label (session_id s, string label)
```

Arguments:

type	name	description
string	label	label of object to return

Return Type: (network ref) Set
references to objects with matching names

2.17 Class: VIF

2.17.1 Fields for class: VIF

Name	**VIF**		
Description	*A virtual network interface.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RO_{run}	`allowed_operations`	(vif_operations) Set	list of the operations allowed in this state
RO_{run}	`current_operations`	(string → vif_operations) Map	Map of task reference to current operation enumeration
RW	`device`	string	order in which VIF backends are created by xapi
RO_{ins}	`network`	network ref	virtual network to which this vif is connected
RO_{ins}	`VM`	VM ref	virtual machine to which this vif is connected
RW	`MAC`	string	ethernet MAC address of virtual interface, as exposed to guest
RW	`MTU`	int	MTU in octets
RW	`other_config`	(string → string) Map	additional configuration
RO_{run}	`currently_attached`	bool	is the device currently attached (erased on reboot)
RO_{run}	`status_code`	int	error/success code associated with last attach-operation (erased on reboot)
RO_{run}	`status_detail`	string	error/success information associated with last attach-operation status (erased on reboot)
RO_{run}	`runtime_properties`	(string → string) Map	Device runtime properties
RW	`qos/algorithm_type`	string	QoS algorithm to use
RW	`qos/algorithm_params`	(string → string) Map	parameters for chosen QoS algorithm
RO_{run}	`qos/supported_algorithms`	string Set	supported QoS algorithms for this VIF
RO_{run}	`metrics`	VIF_metrics ref	metrics associated with this VIF

2.17.2 RPCs associated with class: VIF

RPC name: plug

Overview: Hotplug the specified VIF, dynamically attaching it to the running VM.
Signature:

```
void plug (session_id s, VIF ref self)
```

Arguments:

type	name	description
`VIF ref`	self	The VIF to hotplug

Return Type: `void`

RPC name: unplug

Overview: Hot-unplug the specified VIF, dynamically unattaching it from the running VM.
Signature:

```
void unplug (session_id s, VIF ref self)
```

Arguments:

type	name	description
VIF ref	self	The VIF to hot-unplug

Return Type: `void`

RPC name: get_all

Overview: Return a list of all the VIFs known to the system.
Signature:

```
((VIF ref) Set) get_all (session_id s)
```

Return Type: `(VIF ref) Set`
references to all objects

RPC name: get_all_records_where

Overview: Return a list of all the VIFs known to the system.
Signature:

```
((VIF ref -> VIF record) Map) get_all_records_where (session_id s, string expr)
```

Arguments:

type	name	description
string	expr	expression representing records to fetch

Return Type: `(VIF ref → VIF record) Map`
records of all matching objects

RPC name: get_uuid

Overview: Get the uuid field of the given VIF.
Signature:

```
string get_uuid (session_id s, VIF ref self)
```

Arguments:

type	name	description
VIF ref	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_allowed_operations

Overview: Get the allowed_operations field of the given VIF.
Signature:

```
((vif_operations) Set) get_allowed_operations (session_id s, VIF ref self)
```

Arguments:

type	name	description
VIF ref	self	reference to the object

Return Type: `(vif_operations) Set`
value of the field

RPC name: get_current_operations

Overview: Get the current_operations field of the given VIF.
Signature:

```
((string -> vif_operations) Map) get_current_operations (session_id s, VIF ref self)
```

Arguments:

type	name	description
VIF ref	self	reference to the object

Return Type: `(string → vif_operations) Map`
value of the field

RPC name: get_device

Overview: Get the device field of the given VIF.
Signature:

```
string get_device (session_id s, VIF ref self)
```

Arguments:

type	name	description
VIF ref	self	reference to the object

Return Type: `string`
value of the field

RPC name: set_device

Overview: Set the device field of the given VIF.
Signature:

```
void set_device (session_id s, VIF ref self, string value)
```

Arguments:

type	name	description
VIF ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_network

Overview: Get the network field of the given VIF.
Signature:

```
(network ref) get_network (session_id s, VIF ref self)
```

Arguments:

type	name	description
VIF ref	self	reference to the object

Return Type: network ref
value of the field

RPC name: get_VM

Overview: Get the VM field of the given VIF.
Signature:

```
(VM ref) get_VM (session_id s, VIF ref self)
```

Arguments:

type	name	description
VIF ref	self	reference to the object

Return Type: VM ref
value of the field

RPC name: get_MAC

Overview: Get the MAC field of the given VIF.
Signature:

```
string get_MAC (session_id s, VIF ref self)
```

Arguments:

type	name	description
VIF ref	self	reference to the object

Return Type: string
value of the field

RPC name: set_MAC

Overview: Set the MAC field of the given VIF.
Signature:

```
void set_MAC (session_id s, VIF ref self, string value)
```

Arguments:

type	name	description
VIF ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_MTU

Overview: Get the MTU field of the given VIF.
Signature:

```
int get_MTU (session_id s, VIF ref self)
```

Arguments:

type	name	description
VIF ref	self	reference to the object

Return Type: int
value of the field

RPC name: set_MTU

Overview: Set the MTU field of the given VIF.
Signature:

```
void set_MTU (session_id s, VIF ref self, int value)
```

Arguments:

type	name	description
VIF ref	self	reference to the object
int	value	New value to set

Return Type: void

RPC name: get_other_config

Overview: Get the other_config field of the given VIF.
Signature:

```
((string -> string) Map) get_other_config (session_id s, VIF ref self)
```

Arguments:

type	name	description
VIF ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: set_other_config

Overview: Set the other_config field of the given VIF.
Signature:

```
void set_other_config (session_id s, VIF ref self, (string -> string) Map value)
```

Arguments:

type	name	description
VIF ref	self	reference to the object
(string → string) Map	value	New value to set

Return Type: void

RPC name: add_to_other_config

Overview: Add the given key-value pair to the other_config field of the given VIF.
Signature:

```
void add_to_other_config (session_id s, VIF ref self, string key, string value)
```

Arguments:

type	name	description
VIF ref	self	reference to the object
string	key	Key to add
string	value	Value to add

Return Type: void

RPC name: remove_from_other_config

Overview: Remove the given key and its corresponding value from the other_config field of the given VIF. If the key is not in that Map, then do nothing.
Signature:

```
void remove_from_other_config (session_id s, VIF ref self, string key)
```

Arguments:

type	name	description
VIF ref	self	reference to the object
string	key	Key to remove

Return Type: `void`

RPC name: get_currently_attached

Overview: Get the currently_attached field of the given VIF.
Signature:

```
bool get_currently_attached (session_id s, VIF ref self)
```

Arguments:

type	name	description
`VIF ref`	self	reference to the object

Return Type: `bool`
value of the field

RPC name: get_status_code

Overview: Get the status_code field of the given VIF.
Signature:

```
int get_status_code (session_id s, VIF ref self)
```

Arguments:

type	name	description
`VIF ref`	self	reference to the object

Return Type: `int`
value of the field

RPC name: get_status_detail

Overview: Get the status_detail field of the given VIF.
Signature:

```
string get_status_detail (session_id s, VIF ref self)
```

Arguments:

type	name	description
`VIF ref`	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_runtime_properties

Overview: Get the runtime_properties field of the given VIF.
Signature:

```
((string -> string) Map) get_runtime_properties (session_id s, VIF ref self)
```

Arguments:

type	name	description
VIF ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: get_qos_algorithm_type

Overview: Get the qos/algorithm_type field of the given VIF.
Signature:

```
string get_qos_algorithm_type (session_id s, VIF ref self)
```

Arguments:

type	name	description
VIF ref	self	reference to the object

Return Type: string
value of the field

RPC name: set_qos_algorithm_type

Overview: Set the qos/algorithm_type field of the given VIF.
Signature:

```
void set_qos_algorithm_type (session_id s, VIF ref self, string value)
```

Arguments:

type	name	description
VIF ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_qos_algorithm_params

Overview: Get the qos/algorithm_params field of the given VIF.
Signature:

```
((string -> string) Map) get_qos_algorithm_params (session_id s, VIF ref self)
```

Arguments:

type	name	description
VIF ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: set_qos_algorithm_params

Overview: Set the qos/algorithm_params field of the given VIF.
Signature:

```
void set_qos_algorithm_params (session_id s, VIF ref self, (string -> string) Map value)
```

Arguments:

type	name	description
VIF ref	self	reference to the object
(string → string) Map	value	New value to set

Return Type: void

RPC name: add_to_qos_algorithm_params

Overview: Add the given key-value pair to the qos/algorithm_params field of the given VIF.
Signature:

```
void add_to_qos_algorithm_params (session_id s, VIF ref self, string key, string value)
```

Arguments:

type	name	description
VIF ref	self	reference to the object
string	key	Key to add
string	value	Value to add

Return Type: void

RPC name: remove_from_qos_algorithm_params

Overview: Remove the given key and its corresponding value from the qos/algorithm_params field of the given VIF. If the key is not in that Map, then do nothing.
Signature:

```
void remove_from_qos_algorithm_params (session_id s, VIF ref self, string key)
```

Arguments:

type	name	description
VIF ref	self	reference to the object
string	key	Key to remove

Return Type: `void`

RPC name: get_qos_supported_algorithms

Overview: Get the qos/supported_algorithms field of the given VIF.
Signature:

```
(string Set) get_qos_supported_algorithms (session_id s, VIF ref self)
```

Arguments:

type	name	description
`VIF ref`	self	reference to the object

Return Type: `string Set`
value of the field

RPC name: get_metrics

Overview: Get the metrics field of the given VIF.
Signature:

```
(VIF_metrics ref) get_metrics (session_id s, VIF ref self)
```

Arguments:

type	name	description
`VIF ref`	self	reference to the object

Return Type: `VIF_metrics ref`
value of the field

RPC name: create

Overview: Create a new VIF instance, and return its handle.
Signature:

```
(VIF ref) create (session_id s, VIF record args)
```

Arguments:

type	name	description
`VIF record`	args	All constructor arguments

Return Type: `VIF ref`
reference to the newly created object

RPC name: destroy

Overview: Destroy the specified VIF instance.
Signature:

```
void destroy (session_id s, VIF ref self)
```

Arguments:

type	name	description
VIF ref	self	reference to the object

Return Type: `void`

RPC name: get_by_uuid

Overview: Get a reference to the VIF instance with the specified UUID.
Signature:

```
(VIF ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
string	uuid	UUID of object to return

Return Type: `VIF ref`
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given VIF.
Signature:

```
(VIF record) get_record (session_id s, VIF ref self)
```

Arguments:

type	name	description
VIF ref	self	reference to the object

Return Type: `VIF record`
all fields from the object

2.18 Class: VIF_metrics

2.18.1 Fields for class: VIF_metrics

Name	**VIF_metrics**		
Description	*The metrics associated with a virtual network device.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RO_{run}	`io/read_kbs`	float	Read bandwidth (KiB/s)
RO_{run}	`io/write_kbs`	float	Write bandwidth (KiB/s)
RO_{run}	`last_updated`	datetime	Time at which this information was last updated

2.18.2 RPCs associated with class: VIF_metrics

RPC name: get_all

Overview: Return a list of all the VIF_metrics instances known to the system.
Signature:

```
((VIF_metrics ref) Set) get_all (session_id s)
```

Return Type: `(VIF_metrics ref) Set`
references to all objects

RPC name: get_all_records_where

Overview: Return a list of all the VIF_metrics instances known to the system.
Signature:

```
((VIF_metrics ref -> VIF_metrics record) Map) get_all_records_where (session_id s, string expr)
```

Arguments:

type	name	description
`string`	expr	expression representing records to fetch

Return Type: `(VIF_metrics ref → VIF_metrics record) Map`
records of all matching objects

RPC name: get_uuid

Overview: Get the uuid field of the given VIF_metrics.
Signature:

```
string get_uuid (session_id s, VIF_metrics ref self)
```

Arguments:

type	name	description
`VIF_metrics ref`	self	reference to the object

Return Type: `string`

value of the field

RPC name: get_io_read_kbs

Overview: Get the io/read_kbs field of the given VIF_metrics.
Signature:

```
float get_io_read_kbs (session_id s, VIF_metrics ref self)
```

Arguments:

type	name	description
VIF_metrics ref	self	reference to the object

Return Type: `float`
value of the field

RPC name: get_io_write_kbs

Overview: Get the io/write_kbs field of the given VIF_metrics.
Signature:

```
float get_io_write_kbs (session_id s, VIF_metrics ref self)
```

Arguments:

type	name	description
VIF_metrics ref	self	reference to the object

Return Type: `float`
value of the field

RPC name: get_last_updated

Overview: Get the last_updated field of the given VIF_metrics.
Signature:

```
datetime get_last_updated (session_id s, VIF_metrics ref self)
```

Arguments:

type	name	description
VIF_metrics ref	self	reference to the object

Return Type: `datetime`
value of the field

RPC name: get_by_uuid

Overview: Get a reference to the VIF_metrics instance with the specified UUID.
Signature:

```
(VIF_metrics ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
string	uuid	UUID of object to return

Return Type: VIF_metrics ref
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given VIF_metrics.
Signature:

```
(VIF_metrics record) get_record (session_id s, VIF_metrics ref self)
```

Arguments:

type	name	description
VIF_metrics ref	self	reference to the object

Return Type: VIF_metrics record
all fields from the object

2.19 Class: PIF

2.19.1 Fields for class: PIF

Name	**PIF**		
Description	*A physical network interface (note separate VLANs are represented as several PIFs).*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RW	`device`	string	machine-readable name of the interface (e.g. eth0)
RO_{ins}	`network`	network ref	virtual network to which this pif is connected
RO_{ins}	`host`	host ref	physical machine to which this pif is connected
RW	`MAC`	string	ethernet MAC address of physical interface
RW	`MTU`	int	MTU in octets
RW	`VLAN`	int	VLAN tag for all traffic passing through this interface
RO_{run}	`metrics`	PIF_metrics ref	metrics associated with this PIF

2.19.2 RPCs associated with class: PIF

RPC name: create_VLAN

Overview: Create a VLAN interface from an existing physical interface.
Signature:

```
(PIF ref) create_VLAN (session_id s, string device, network ref network, host ref host, int VLAN)
```

Arguments:

type	name	description
`string`	device	physical interface on which to crate the VLAN interface
`network ref`	network	network to which this interface should be connected
`host ref`	host	physical machine to which this PIF is connected
`int`	VLAN	VLAN tag for the new interface

Return Type: `PIF ref`
The reference of the created PIF object

Possible Error Codes: `VLAN_TAG_INVALID`

RPC name: destroy

Overview: Destroy the interface (provided it is a synthetic interface like a VLAN; fail if it is a physical interface).
Signature:

```
void destroy (session_id s, PIF ref self)
```

Arguments:

type	name	description
PIF ref	self	the PIF object to destroy

Return Type: void

Possible Error Codes: PIF_IS_PHYSICAL

RPC name: get_all

Overview: Return a list of all the PIFs known to the system.
Signature:

```
((PIF ref) Set) get_all (session_id s)
```

Return Type: (PIF ref) Set
references to all objects

RPC name: get_all_records_where

Overview: Return a list of all the PIFs known to the system.
Signature:

```
((PIF ref -> PIF record) Map) get_all_records_where (session_id s, string expr)
```

Arguments:

type	name	description
string	expr	expression representing records to fetch

Return Type: (PIF ref → PIF record) Map
records of all matching objects

RPC name: get_uuid

Overview: Get the uuid field of the given PIF.
Signature:

```
string get_uuid (session_id s, PIF ref self)
```

Arguments:

type	name	description
PIF ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_device

Overview: Get the device field of the given PIF.
Signature:

```
string get_device (session_id s, PIF ref self)
```

Arguments:

type	name	description
PIF ref	self	reference to the object

Return Type: `string`
value of the field

RPC name: set_device

Overview: Set the device field of the given PIF.
Signature:

```
void set_device (session_id s, PIF ref self, string value)
```

Arguments:

type	name	description
PIF ref	self	reference to the object
string	value	New value to set

Return Type: `void`

RPC name: get_network

Overview: Get the network field of the given PIF.
Signature:

```
(network ref) get_network (session_id s, PIF ref self)
```

Arguments:

type	name	description
PIF ref	self	reference to the object

Return Type: `network ref`
value of the field

RPC name: get_host

Overview: Get the host field of the given PIF.
Signature:

```
(host ref) get_host (session_id s, PIF ref self)
```

Arguments:

type	name	description
`PIF ref`	self	reference to the object

Return Type: `host ref`
value of the field

RPC name: get_MAC

Overview: Get the MAC field of the given PIF.
Signature:

```
string get_MAC (session_id s, PIF ref self)
```

Arguments:

type	name	description
`PIF ref`	self	reference to the object

Return Type: `string`
value of the field

RPC name: set_MAC

Overview: Set the MAC field of the given PIF.
Signature:

```
void set_MAC (session_id s, PIF ref self, string value)
```

Arguments:

type	name	description
`PIF ref`	self	reference to the object
`string`	value	New value to set

Return Type: `void`

RPC name: get_MTU

Overview: Get the MTU field of the given PIF.
Signature:

```
int get_MTU (session_id s, PIF ref self)
```

Arguments:

type	name	description
`PIF ref`	self	reference to the object

Return Type: `int`
value of the field

RPC name: set_MTU

Overview: Set the MTU field of the given PIF.
Signature:

```
void set_MTU (session_id s, PIF ref self, int value)
```

Arguments:

type	name	description
PIF ref	self	reference to the object
int	value	New value to set

Return Type: `void`

RPC name: get_VLAN

Overview: Get the VLAN field of the given PIF.
Signature:

```
int get_VLAN (session_id s, PIF ref self)
```

Arguments:

type	name	description
PIF ref	self	reference to the object

Return Type: `int`
value of the field

RPC name: set_VLAN

Overview: Set the VLAN field of the given PIF.
Signature:

```
void set_VLAN (session_id s, PIF ref self, int value)
```

Arguments:

type	name	description
PIF ref	self	reference to the object
int	value	New value to set

Return Type: `void`

RPC name: get_metrics

Overview: Get the metrics field of the given PIF.
Signature:

```
(PIF_metrics ref) get_metrics (session_id s, PIF ref self)
```

Arguments:

type	name	description
`PIF ref`	self	reference to the object

Return Type: `PIF_metrics ref`
value of the field

RPC name: get_by_uuid

Overview: Get a reference to the PIF instance with the specified UUID.
Signature:

```
(PIF ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
`string`	uuid	UUID of object to return

Return Type: `PIF ref`
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given PIF.
Signature:

```
(PIF record) get_record (session_id s, PIF ref self)
```

Arguments:

type	name	description
`PIF ref`	self	reference to the object

Return Type: `PIF record`
all fields from the object

2.20 Class: PIF_metrics

2.20.1 Fields for class: PIF_metrics

Name	**PIF_metrics**		
Description	*The metrics associated with a physical network interface.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RO_{run}	`io/read_kbs`	float	Read bandwidth (KiB/s)
RO_{run}	`io/write_kbs`	float	Write bandwidth (KiB/s)
RO_{run}	`carrier`	bool	Report if the PIF got a carrier or not
RO_{run}	`vendor_id`	string	Report vendor ID
RO_{run}	`vendor_name`	string	Report vendor name
RO_{run}	`device_id`	string	Report device ID
RO_{run}	`device_name`	string	Report device name
RO_{run}	`speed`	int	Speed of the link (if available)
RO_{run}	`duplex`	bool	Full duplex capability of the link (if available)
RO_{run}	`pci_bus_path`	string	PCI bus path of the pif (if available)
RO_{run}	`last_updated`	datetime	Time at which this information was last updated

2.20.2 RPCs associated with class: PIF_metrics

RPC name: get_all

Overview: Return a list of all the PIF_metrics instances known to the system.
Signature:

```
((PIF_metrics ref) Set) get_all (session_id s)
```

Return Type: `(PIF_metrics ref) Set`
references to all objects

RPC name: get_all_records_where

Overview: Return a list of all the PIF_metrics instances known to the system.
Signature:

```
((PIF_metrics ref -> PIF_metrics record) Map) get_all_records_where (session_id s, string expr)
```

Arguments:

type	name	description
`string`	expr	expression representing records to fetch

Return Type: `(PIF_metrics ref → PIF_metrics record) Map`
records of all matching objects

RPC name: get_uuid

Overview: Get the uuid field of the given PIF_metrics.
Signature:

```
string get_uuid (session_id s, PIF_metrics ref self)
```

Arguments:

type	name	description
PIF_metrics ref	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_io_read_kbs

Overview: Get the io/read_kbs field of the given PIF_metrics.
Signature:

```
float get_io_read_kbs (session_id s, PIF_metrics ref self)
```

Arguments:

type	name	description
PIF_metrics ref	self	reference to the object

Return Type: `float`
value of the field

RPC name: get_io_write_kbs

Overview: Get the io/write_kbs field of the given PIF_metrics.
Signature:

```
float get_io_write_kbs (session_id s, PIF_metrics ref self)
```

Arguments:

type	name	description
PIF_metrics ref	self	reference to the object

Return Type: `float`
value of the field

RPC name: get_carrier

Overview: Get the carrier field of the given PIF_metrics.
Signature:

```
bool get_carrier (session_id s, PIF_metrics ref self)
```

Arguments:

type	name	description
PIF_metrics ref	self	reference to the object

Return Type: bool
value of the field

RPC name: get_vendor_id

Overview: Get the vendor_id field of the given PIF_metrics.
Signature:

```
string get_vendor_id (session_id s, PIF_metrics ref self)
```

Arguments:

type	name	description
PIF_metrics ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_vendor_name

Overview: Get the vendor_name field of the given PIF_metrics.
Signature:

```
string get_vendor_name (session_id s, PIF_metrics ref self)
```

Arguments:

type	name	description
PIF_metrics ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_device_id

Overview: Get the device_id field of the given PIF_metrics.
Signature:

```
string get_device_id (session_id s, PIF_metrics ref self)
```

Arguments:

type	name	description
PIF_metrics ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_device_name

Overview: Get the device_name field of the given PIF_metrics.
Signature:

```
string get_device_name (session_id s, PIF_metrics ref self)
```

Arguments:

type	name	description
PIF_metrics ref	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_speed

Overview: Get the speed field of the given PIF_metrics.
Signature:

```
int get_speed (session_id s, PIF_metrics ref self)
```

Arguments:

type	name	description
PIF_metrics ref	self	reference to the object

Return Type: `int`
value of the field

RPC name: get_duplex

Overview: Get the duplex field of the given PIF_metrics.
Signature:

```
bool get_duplex (session_id s, PIF_metrics ref self)
```

Arguments:

type	name	description
PIF_metrics ref	self	reference to the object

Return Type: `bool`
value of the field

RPC name: get_pci_bus_path

Overview: Get the pci_bus_path field of the given PIF_metrics.
Signature:

```
string get_pci_bus_path (session_id s, PIF_metrics ref self)
```

Arguments:

type	name	description
PIF_metrics ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_last_updated

Overview: Get the last_updated field of the given PIF_metrics.
Signature:

```
datetime get_last_updated (session_id s, PIF_metrics ref self)
```

Arguments:

type	name	description
PIF_metrics ref	self	reference to the object

Return Type: datetime
value of the field

RPC name: get_by_uuid

Overview: Get a reference to the PIF_metrics instance with the specified UUID.
Signature:

```
(PIF_metrics ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
string	uuid	UUID of object to return

Return Type: PIF_metrics ref
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given PIF_metrics.
Signature:

```
(PIF_metrics record) get_record (session_id s, PIF_metrics ref self)
```

Arguments:

type	name	description
PIF_metrics ref	self	reference to the object

Return Type: PIF_metrics record
all fields from the object

2.21 Class: SM

2.21.1 Fields for class: SM

Name	**SM**		
Description	*A storage manager plugin.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RO_{run}	`name/label`	string	a human-readable name
RO_{run}	`name/description`	string	a notes field containg human-readable description
RO_{run}	`type`	string	SR.type
RO_{run}	`vendor`	string	Vendor who created this plugin
RO_{run}	`copyright`	string	Entity which owns the copyright of this plugin
RO_{run}	`version`	string	Version of the plugin
RO_{run}	`required_api_version`	string	Minimum SM API version required on the server
RO_{run}	`configuration`	(string → string) Map	names and descriptions of device config keys

2.21.2 RPCs associated with class: SM

RPC name: get_all

Overview: Return a list of all the SMs known to the system.
Signature:

```
((SM ref) Set) get_all (session_id s)
```

Return Type: `(SM ref) Set`
references to all objects

RPC name: get_all_records_where

Overview: Return a list of all the SMs known to the system.
Signature:

```
((SM ref -> SM record) Map) get_all_records_where (session_id s, string expr)
```

Arguments:

type	name	description
`string`	expr	expression representing records to fetch

Return Type: `(SM ref → SM record) Map`
records of all matching objects

RPC name: get_uuid

Overview: Get the uuid field of the given SM.
Signature:

```
string get_uuid (session_id s, SM ref self)
```

Arguments:

type	name	description
SM ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_name_label

Overview: Get the name/label field of the given SM.
Signature:

```
string get_name_label (session_id s, SM ref self)
```

Arguments:

type	name	description
SM ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_name_description

Overview: Get the name/description field of the given SM.
Signature:

```
string get_name_description (session_id s, SM ref self)
```

Arguments:

type	name	description
SM ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_type

Overview: Get the type field of the given SM.
Signature:

```
string get_type (session_id s, SM ref self)
```

Arguments:

type	name	description
SM ref	self	reference to the object

Return Type: string

value of the field

RPC name: get_vendor

Overview: Get the vendor field of the given SM.
Signature:

```
string get_vendor (session_id s, SM ref self)
```

Arguments:

type	name	description
SM ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_copyright

Overview: Get the copyright field of the given SM.
Signature:

```
string get_copyright (session_id s, SM ref self)
```

Arguments:

type	name	description
SM ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_version

Overview: Get the version field of the given SM.
Signature:

```
string get_version (session_id s, SM ref self)
```

Arguments:

type	name	description
SM ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_required_api_version

Overview: Get the required_api_version field of the given SM.
Signature:

```
string get_required_api_version (session_id s, SM ref self)
```

Arguments:

type	name	description
SM ref	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_configuration

Overview: Get the configuration field of the given SM.
Signature:

```
((string -> string) Map) get_configuration (session_id s, SM ref self)
```

Arguments:

type	name	description
SM ref	self	reference to the object

Return Type: `(string → string) Map`
value of the field

RPC name: get_by_uuid

Overview: Get a reference to the SM instance with the specified UUID.
Signature:

```
(SM ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
string	uuid	UUID of object to return

Return Type: `SM ref`
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given SM.
Signature:

```
(SM record) get_record (session_id s, SM ref self)
```

Arguments:

type	name	description
`SM ref`	self	reference to the object

Return Type: `SM record`
all fields from the object

RPC name: get_by_name_label

Overview: Get all the SM instances with the given label.
Signature:

```
((SM ref) Set) get_by_name_label (session_id s, string label)
```

Arguments:

type	name	description
`string`	label	label of object to return

Return Type: `(SM ref) Set`
references to objects with matching names

2.22 Class: SR

2.22.1 Fields for class: SR

Name	**SR**		
Description	*A storage repository.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RW	`name/label`	string	a human-readable name
RW	`name/description`	string	a notes field containg human-readable description
RO_{run}	`allowed_operations`	(storage_operations) Set	list of the operations allowed in this state
RO_{run}	`current_operations`	(string → storage_operations) Map	Map of task reference to current operation enumeration
RO_{run}	`VDIs`	(VDI ref) Set	managed virtual disks
RO_{run}	`PBDs`	(PBD ref) Set	physical blockdevices
RO_{run}	`virtual_allocation`	int	sum of virtual_sizes of all VDIs in this storage repository (in bytes)
RO_{run}	`physical_utilisation`	int	physical space currently utilised on this storage repository (in bytes). Note that for sparse disk formats, physical_utilisation may be less than virtual_allocation
RO_{ins}	`physical_size`	int	total physical size of the repository (in bytes)
RO_{ins}	`type`	string	type of the storage repository
RO_{ins}	`content_type`	string	the type of the SR's content, if required (e.g. ISOs)
RW	`shared`	bool	true if this SR is (capable of being) shared between multiple hosts
RW	`other_config`	(string → string) Map	additional configuration

2.22.2 RPCs associated with class: SR

RPC name: create

Overview: Create a new Storage Repository and introduce it into the managed system, creating both SR record and PBD record to attach it to current host (with specified device_config parameters).

Signature:

```
(SR ref) create (session_id s, host ref host, (string -> string) Map device_config, int physical_size,
```

Arguments:

type	name	description
host ref	host	The host to create/make the SR on
(string → string) Map	device_config	The device config string that will be passed to backend SR driver
int	physical_size	The physical size of the new storage repository
string	name_label	The name of the new storage repository
string	name_description	The description of the new storage repository
string	type	The type of the SR; used to specify the SR backend driver to use
string	content_type	The type of the new SRs content, if required (e.g. ISOs)
bool	shared	True if the SR (is capable of) being shared by multiple hosts

Return Type: SR ref
The ID of the newly created Storage Repository.

RPC name: introduce

Overview: Introduce a new Storage Repository into the managed system.
Signature:

```
(SR ref) introduce (session_id s, string uuid, string name_label, string name_description, string type
```

Arguments:

type	name	description
string	uuid	The uuid assigned to the introduced SR
string	name_label	The name of the new storage repository
string	name_description	The description of the new storage repository
string	type	The type of the SR; used to specify the SR backend driver to use
string	content_type	The type of the new SRs content, if required (e.g. ISOs)
bool	shared	True if the SR (is capable of) being shared by multiple hosts

Return Type: SR ref
The ID of the newly introduced Storage Repository.

RPC name: make

Overview: Create a new Storage Repository on disk.
Signature:

```
string make (session_id s, host ref host, (string -> string) Map device_config, int physical_size, str
```

Arguments:

type	name	description
host ref	host	The host to create/make the SR on
(string → string) Map	device_config	The device config string that will be passed to backend SR driver
int	physical_size	The physical size of the new storage repository
string	name_label	The name of the new storage repository
string	name_description	The description of the new storage repository
string	type	The type of the SR; used to specify the SR backend driver to use
string	content_type	The type of the new SRs content, if required (e.g. ISOs)

Return Type: string
The uuid of the newly created Storage Repository.

RPC name: destroy

Overview: Destroy specified SR, removing SR-record from database and remove SR from disk. (In order to affect this operation the appropriate device_config is read from the specified SR's PBD on current host).
Signature:

```
void destroy (session_id s, SR ref sr)
```

Arguments:

type	name	description
SR ref	sr	The SR to destroy

Return Type: void

RPC name: forget

Overview: Removing specified SR-record from database, without attempting to remove SR from disk.
Signature:

```
void forget (session_id s, SR ref sr)
```

Arguments:

type	name	description
SR ref	sr	The SR to destroy

Return Type: void

Possible Error Codes: SR_HAS_PBD

RPC name: get_supported_types

Overview: Return a set of all the SR types supported by the system.
Signature:

```
(string Set) get_supported_types (session_id s)
```

Return Type: `string Set`
the supported SR types

RPC name: scan

Overview: Refreshes the VDIs contained within an SR.
Signature:

```
void scan (session_id s, SR ref sr)
```

Arguments:

type	name	description
`SR ref`	sr	The SR to scan

Return Type: `void`

RPC name: get_all

Overview: Return a list of all the SRs known to the system.
Signature:

```
((SR ref) Set) get_all (session_id s)
```

Return Type: `(SR ref) Set`
references to all objects

RPC name: get_all_records_where

Overview: Return a list of all the SRs known to the system.
Signature:

```
((SR ref -> SR record) Map) get_all_records_where (session_id s, string expr)
```

Arguments:

type	name	description
`string`	expr	expression representing records to fetch

Return Type: `(SR ref → SR record) Map`
records of all matching objects

RPC name: get_uuid

Overview: Get the uuid field of the given SR.
Signature:

```
string get_uuid (session_id s, SR ref self)
```

Arguments:

type	name	description
SR ref	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_name_label

Overview: Get the name/label field of the given SR.
Signature:

```
string get_name_label (session_id s, SR ref self)
```

Arguments:

type	name	description
SR ref	self	reference to the object

Return Type: `string`
value of the field

RPC name: set_name_label

Overview: Set the name/label field of the given SR.
Signature:

```
void set_name_label (session_id s, SR ref self, string value)
```

Arguments:

type	name	description
SR ref	self	reference to the object
string	value	New value to set

Return Type: `void`

RPC name: get_name_description

Overview: Get the name/description field of the given SR.
Signature:

```
string get_name_description (session_id s, SR ref self)
```

Arguments:

type	name	description
SR ref	self	reference to the object

Return Type: string
value of the field

RPC name: set_name_description

Overview: Set the name/description field of the given SR.
Signature:

```
void set_name_description (session_id s, SR ref self, string value)
```

Arguments:

type	name	description
SR ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_allowed_operations

Overview: Get the allowed_operations field of the given SR.
Signature:

```
((storage_operations) Set) get_allowed_operations (session_id s, SR ref self)
```

Arguments:

type	name	description
SR ref	self	reference to the object

Return Type: (storage_operations) Set
value of the field

RPC name: get_current_operations

Overview: Get the current_operations field of the given SR.
Signature:

```
((string -> storage_operations) Map) get_current_operations (session_id s, SR ref self)
```

Arguments:

type	name	description
SR ref	self	reference to the object

Return Type: (string → storage_operations) Map
value of the field

RPC name: get_VDIs

Overview: Get the VDIs field of the given SR.
Signature:

```
((VDI ref) Set) get_VDIs (session_id s, SR ref self)
```

Arguments:

type	name	description
SR ref	self	reference to the object

Return Type: `(VDI ref) Set`
value of the field

RPC name: get_PBDs

Overview: Get the PBDs field of the given SR.
Signature:

```
((PBD ref) Set) get_PBDs (session_id s, SR ref self)
```

Arguments:

type	name	description
SR ref	self	reference to the object

Return Type: `(PBD ref) Set`
value of the field

RPC name: get_virtual_allocation

Overview: Get the virtual_allocation field of the given SR.
Signature:

```
int get_virtual_allocation (session_id s, SR ref self)
```

Arguments:

type	name	description
SR ref	self	reference to the object

Return Type: `int`
value of the field

RPC name: get_physical_utilisation

Overview: Get the physical_utilisation field of the given SR.
Signature:

```
int get_physical_utilisation (session_id s, SR ref self)
```

Arguments:

type	name	description
SR ref	self	reference to the object

Return Type: int
value of the field

RPC name: get_physical_size

Overview: Get the physical_size field of the given SR.
Signature:

```
int get_physical_size (session_id s, SR ref self)
```

Arguments:

type	name	description
SR ref	self	reference to the object

Return Type: int
value of the field

RPC name: get_type

Overview: Get the type field of the given SR.
Signature:

```
string get_type (session_id s, SR ref self)
```

Arguments:

type	name	description
SR ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_content_type

Overview: Get the content_type field of the given SR.
Signature:

```
string get_content_type (session_id s, SR ref self)
```

Arguments:

type	name	description
SR ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_shared

Overview: Get the shared field of the given SR.
Signature:

```
bool get_shared (session_id s, SR ref self)
```

Arguments:

type	name	description
SR ref	self	reference to the object

Return Type: `bool`
value of the field

RPC name: set_shared

Overview: Set the shared field of the given SR.
Signature:

```
void set_shared (session_id s, SR ref self, bool value)
```

Arguments:

type	name	description
SR ref	self	reference to the object
bool	value	New value to set

Return Type: `void`

RPC name: get_other_config

Overview: Get the other_config field of the given SR.
Signature:

```
((string -> string) Map) get_other_config (session_id s, SR ref self)
```

Arguments:

type	name	description
SR ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: set_other_config

Overview: Set the other_config field of the given SR.
Signature:

```
void set_other_config (session_id s, SR ref self, (string -> string) Map value)
```

Arguments:

type	name	description
SR ref	self	reference to the object
(string → string) Map	value	New value to set

Return Type: void

RPC name: add_to_other_config

Overview: Add the given key-value pair to the other_config field of the given SR.
Signature:

```
void add_to_other_config (session_id s, SR ref self, string key, string value)
```

Arguments:

type	name	description
SR ref	self	reference to the object
string	key	Key to add
string	value	Value to add

Return Type: void

RPC name: remove_from_other_config

Overview: Remove the given key and its corresponding value from the other_config field of the given SR. If the key is not in that Map, then do nothing.
Signature:

```
void remove_from_other_config (session_id s, SR ref self, string key)
```

Arguments:

type	name	description
SR ref	self	reference to the object
string	key	Key to remove

Return Type: void

RPC name: get_by_uuid

Overview: Get a reference to the SR instance with the specified UUID.
Signature:

```
(SR ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
string	uuid	UUID of object to return

Return Type: SR ref

reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given SR.
Signature:

```
(SR record) get_record (session_id s, SR ref self)
```

Arguments:

type	name	description
SR ref	self	reference to the object

Return Type: `SR record`
all fields from the object

RPC name: get_by_name_label

Overview: Get all the SR instances with the given label.
Signature:

```
((SR ref) Set) get_by_name_label (session_id s, string label)
```

Arguments:

type	name	description
string	label	label of object to return

Return Type: `(SR ref) Set`
references to objects with matching names

2.23 Class: VDI

2.23.1 Fields for class: VDI

Name	**VDI**		
Description	*A virtual disk image.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RW	`name/label`	string	a human-readable name
RW	`name/description`	string	a notes field containg human-readable description
RO_{run}	`allowed_operations`	(vdi_operations) Set	list of the operations allowed in this state
RO_{run}	`current_operations`	(string → vdi_operations) Map	Map of task reference to current operation enumeration
RO_{ins}	`SR`	SR ref	storage repository in which the VDI resides
RO_{run}	`VBDs`	(VBD ref) Set	list of vbds that refer to this disk
RO_{run}	`crash_dumps`	(crashdump ref) Set	list of crash dumps that refer to this disk
RW	`virtual_size`	int	size of disk as presented to the guest (in bytes). Note that, depending on storage backend type, requested size may not be respected exactly
RO_{run}	`physical_utilisation`	int	amount of physical space that the disk image is currently taking up on the storage repository (in bytes)
RO_{ins}	`type`	vdi_type	type of the VDI
RW	`sharable`	bool	true if this disk may be shared
RW	`read_only`	bool	true if this disk may ONLY be mounted read-only
RW	`other_config`	(string → string) Map	additional configuration
RO_{run}	`storage_lock`	bool	true if this disk is locked at the storage level
RO_{run}	`managed`	bool	
RO_{run}	`missing`	bool	true if SR scan operation reported this VDI as not present on disk
RO_{run}	`parent`	VDI ref	References the parent disk, if this VDI is part of a chain

2.23.2 RPCs associated with class: VDI

RPC name: snapshot

Overview: Take a read-only snapshot of the VDI, returning a reference this; the snapshot lives in the same Storage Repository as its parent.

Signature:

```
(VDI ref) snapshot (session_id s, VDI ref vdi)
```

Arguments:

type	name	description
`VDI ref`	vdi	The VDI to snapshot

Return Type: `VDI ref`

The ID of the newly created VDI.

RPC name: clone

Overview: Take an exact copy of the VDI, returning a reference this; the clone lives in the same Storage Repository as its parent.
Signature:

```
(VDI ref) clone (session_id s, VDI ref vdi)
```

Arguments:

type	name	description
VDI ref	vdi	The VDI to clone

Return Type: VDI ref
The ID of the newly created VDI.

RPC name: resize

Overview: Resize the vdi to the size.
Signature:

```
void resize (session_id s, VDI ref vdi, int size)
```

Arguments:

type	name	description
VDI ref	vdi	The VDI to resize
int	size	The new size of the VDI

Return Type: void

RPC name: introduce

Overview: Create a new VDI record in the database only.
Signature:

```
(VDI ref) introduce (session_id s, string uuid, string name_label, string name_description, SR ref SR,
```

Arguments:

type	name	description
string	uuid	The uuid of the disk to introduce
string	name_label	The name of the disk record
string	name_description	The description of the disk record
SR ref	SR	The SR that the VDI is in
vdi_type	type	The type of the VDI
bool	sharable	true if this disk may be shared
bool	read_only	true if this disk may ONLY be mounted read-only
(string → string) Map	other_config	additional configuration
string	location	location information

Return Type: `VDI ref`
The ref of the newly created VDI record.

RPC name: copy

Overview: Copy a VDI to a specified SR.
Signature:

```
(VDI ref) copy (session_id s, VDI ref vdi, SR ref sr)
```

Arguments:

type	name	description
`VDI ref`	vdi	The VDI to copy
`SR ref`	sr	The destination SR

Return Type: `VDI ref`
The ref of the newly created VDI record.

RPC name: force_unlock

Overview: Steals the lock on this VDI and leaves it unlocked. This function is extremely dangerous.
Signature:

```
void force_unlock (session_id s, VDI ref vdi)
```

Arguments:

type	name	description
`VDI ref`	vdi	The VDI to forcibly unlock

Return Type: `void`

RPC name: set_managed

Overview: Sets the VDI's managed field.
Signature:

```
void set_managed (session_id s, VDI ref self, bool value)
```

Arguments:

type	name	description
`VDI ref`	self	The VDI to modify
`bool`	value	The new value of the VDI's managed field

Return Type: `void`

RPC name: get_all

Overview: Return a list of all the VDIs known to the system.
Signature:

```
((VDI ref) Set) get_all (session_id s)
```

Return Type: `(VDI ref) Set`
references to all objects

RPC name: get_all_records_where

Overview: Return a list of all the VDIs known to the system.
Signature:

```
((VDI ref -> VDI record) Map) get_all_records_where (session_id s, string expr)
```

Arguments:

type	name	description
`string`	expr	expression representing records to fetch

Return Type: `(VDI ref → VDI record) Map`
records of all matching objects

RPC name: get_uuid

Overview: Get the uuid field of the given VDI.
Signature:

```
string get_uuid (session_id s, VDI ref self)
```

Arguments:

type	name	description
`VDI ref`	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_name_label

Overview: Get the name/label field of the given VDI.
Signature:

```
string get_name_label (session_id s, VDI ref self)
```

Arguments:

type	name	description
`VDI ref`	self	reference to the object

Return Type: `string`

value of the field

RPC name: set_name_label

Overview: Set the name/label field of the given VDI.
Signature:

```
void set_name_label (session_id s, VDI ref self, string value)
```

Arguments:

type	name	description
VDI ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_name_description

Overview: Get the name/description field of the given VDI.
Signature:

```
string get_name_description (session_id s, VDI ref self)
```

Arguments:

type	name	description
VDI ref	self	reference to the object

Return Type: string
value of the field

RPC name: set_name_description

Overview: Set the name/description field of the given VDI.
Signature:

```
void set_name_description (session_id s, VDI ref self, string value)
```

Arguments:

type	name	description
VDI ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_allowed_operations

Overview: Get the allowed_operations field of the given VDI.
Signature:

```
((vdi_operations) Set) get_allowed_operations (session_id s, VDI ref self)
```

Arguments:

type	name	description
VDI ref	self	reference to the object

Return Type: `(vdi_operations) Set`
value of the field

RPC name: get_current_operations

Overview: Get the current_operations field of the given VDI.
Signature:

```
((string -> vdi_operations) Map) get_current_operations (session_id s, VDI ref self)
```

Arguments:

type	name	description
VDI ref	self	reference to the object

Return Type: `(string → vdi_operations) Map`
value of the field

RPC name: get_SR

Overview: Get the SR field of the given VDI.
Signature:

```
(SR ref) get_SR (session_id s, VDI ref self)
```

Arguments:

type	name	description
VDI ref	self	reference to the object

Return Type: `SR ref`
value of the field

RPC name: get_VBDs

Overview: Get the VBDs field of the given VDI.
Signature:

```
((VBD ref) Set) get_VBDs (session_id s, VDI ref self)
```

Arguments:

type	name	description
VDI ref	self	reference to the object

Return Type: (VBD ref) Set
value of the field

RPC name: get_crash_dumps

Overview: Get the crash_dumps field of the given VDI.
Signature:

```
((crashdump ref) Set) get_crash_dumps (session_id s, VDI ref self)
```

Arguments:

type	name	description
VDI ref	self	reference to the object

Return Type: (crashdump ref) Set
value of the field

RPC name: get_virtual_size

Overview: Get the virtual_size field of the given VDI.
Signature:

```
int get_virtual_size (session_id s, VDI ref self)
```

Arguments:

type	name	description
VDI ref	self	reference to the object

Return Type: int
value of the field

RPC name: set_virtual_size

Overview: Set the virtual_size field of the given VDI.
Signature:

```
void set_virtual_size (session_id s, VDI ref self, int value)
```

Arguments:

type	name	description
VDI ref	self	reference to the object
int	value	New value to set

Return Type: void

RPC name: get_physical_utilisation

Overview: Get the physical_utilisation field of the given VDI.
Signature:

```
int get_physical_utilisation (session_id s, VDI ref self)
```

Arguments:

type	name	description
VDI ref	self	reference to the object

Return Type: int
value of the field

RPC name: get_type

Overview: Get the type field of the given VDI.
Signature:

```
(vdi_type) get_type (session_id s, VDI ref self)
```

Arguments:

type	name	description
VDI ref	self	reference to the object

Return Type: vdi_type
value of the field

RPC name: get_sharable

Overview: Get the sharable field of the given VDI.
Signature:

```
bool get_sharable (session_id s, VDI ref self)
```

Arguments:

type	name	description
VDI ref	self	reference to the object

Return Type: bool
value of the field

RPC name: set_sharable

Overview: Set the sharable field of the given VDI.
Signature:

```
void set_sharable (session_id s, VDI ref self, bool value)
```

Arguments:

type	name	description
VDI ref	self	reference to the object
bool	value	New value to set

Return Type: void

RPC name: get_read_only

Overview: Get the read_only field of the given VDI.
Signature:

```
bool get_read_only (session_id s, VDI ref self)
```

Arguments:

type	name	description
VDI ref	self	reference to the object

Return Type: bool
value of the field

RPC name: set_read_only

Overview: Set the read_only field of the given VDI.
Signature:

```
void set_read_only (session_id s, VDI ref self, bool value)
```

Arguments:

type	name	description
VDI ref	self	reference to the object
bool	value	New value to set

Return Type: void

RPC name: get_other_config

Overview: Get the other_config field of the given VDI.
Signature:

```
((string -> string) Map) get_other_config (session_id s, VDI ref self)
```

Arguments:

type	name	description
VDI ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: set_other_config

Overview: Set the other_config field of the given VDI.
Signature:

```
void set_other_config (session_id s, VDI ref self, (string -> string) Map value)
```

Arguments:

type	name	description
VDI ref	self	reference to the object
(string → string) Map	value	New value to set

Return Type: void

RPC name: add_to_other_config

Overview: Add the given key-value pair to the other_config field of the given VDI.
Signature:

```
void add_to_other_config (session_id s, VDI ref self, string key, string value)
```

Arguments:

type	name	description
VDI ref	self	reference to the object
string	key	Key to add
string	value	Value to add

Return Type: void

RPC name: remove_from_other_config

Overview: Remove the given key and its corresponding value from the other_config field of the given VDI. If the key is not in that Map, then do nothing.
Signature:

```
void remove_from_other_config (session_id s, VDI ref self, string key)
```

Arguments:

type	name	description
VDI ref	self	reference to the object
string	key	Key to remove

Return Type: void

RPC name: get_storage_lock

Overview: Get the storage_lock field of the given VDI.
Signature:

```
bool get_storage_lock (session_id s, VDI ref self)
```

Arguments:

type	name	description
VDI ref	self	reference to the object

Return Type: bool
value of the field

RPC name: get_managed

Overview: Get the managed field of the given VDI.
Signature:

```
bool get_managed (session_id s, VDI ref self)
```

Arguments:

type	name	description
VDI ref	self	reference to the object

Return Type: bool
value of the field

RPC name: get_missing

Overview: Get the missing field of the given VDI.
Signature:

```
bool get_missing (session_id s, VDI ref self)
```

Arguments:

type	name	description
VDI ref	self	reference to the object

Return Type: bool
value of the field

RPC name: get_parent

Overview: Get the parent field of the given VDI.
Signature:

```
(VDI ref) get_parent (session_id s, VDI ref self)
```

Arguments:

type	name	description
VDI ref	self	reference to the object

Return Type: VDI ref
value of the field

RPC name: create

Overview: Create a new VDI instance, and return its handle.
Signature:

```
(VDI ref) create (session_id s, VDI record args)
```

Arguments:

type	name	description
VDI record	args	All constructor arguments

Return Type: `VDI ref`
reference to the newly created object

RPC name: destroy

Overview: Destroy the specified VDI instance.
Signature:

```
void destroy (session_id s, VDI ref self)
```

Arguments:

type	name	description
VDI ref	self	reference to the object

Return Type: `void`

RPC name: get_by_uuid

Overview: Get a reference to the VDI instance with the specified UUID.
Signature:

```
(VDI ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
string	uuid	UUID of object to return

Return Type: `VDI ref`
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given VDI.
Signature:

```
(VDI record) get_record (session_id s, VDI ref self)
```

Arguments:

type	name	description
`VDI ref`	self	reference to the object

Return Type: `VDI record`
all fields from the object

RPC name: get_by_name_label

Overview: Get all the VDI instances with the given label.
Signature:

```
((VDI ref) Set) get_by_name_label (session_id s, string label)
```

Arguments:

type	name	description
`string`	label	label of object to return

Return Type: `(VDI ref) Set`
references to objects with matching names

2.24 Class: VBD

2.24.1 Fields for class: VBD

Name	**VBD**		
Description	*A virtual block device.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RO_{run}	`allowed_operations`	(vbd_operations) Set	list of the operations allowed in this state
RO_{run}	`current_operations`	(string → vbd_operations) Map	Map of task reference to current operation enumeration
RO_{ins}	`VM`	VM ref	the virtual machine
RO_{ins}	`VDI`	VDI ref	the virtual disk
RO_{run}	`device`	string	device seen by the guest e.g. hda1
RW	`userdevice`	string	user-friendly device name e.g. 0,1,2,etc.
RW	`bootable`	bool	true if this VBD is bootable
RW	`mode`	vbd_mode	the mode the VBD should be mounted with
RW	`type`	vbd_type	how the VBD will appear to the guest (e.g. disk or CD)
RO_{run}	`storage_lock`	bool	true if a storage level lock was acquired
RO_{ins}	`empty`	bool	if true this represents an empty drive
RW	`other_config`	(string → string) Map	additional configuration
RO_{run}	`currently_attached`	bool	is the device currently attached (erased on reboot)
RO_{run}	`status_code`	int	error/success code associated with last attach-operation (erased on reboot)
RO_{run}	`status_detail`	string	error/success information associated with last attach-operation status (erased on reboot)
RO_{run}	`runtime_properties`	(string → string) Map	Device runtime properties
RW	`qos/algorithm_type`	string	QoS algorithm to use
RW	`qos/algorithm_params`	(string → string) Map	parameters for chosen QoS algorithm
RO_{run}	`qos/supported_algorithms`	string Set	supported QoS algorithms for this VBD
RO_{run}	`metrics`	VBD_metrics ref	metrics associated with this VBD

2.24.2 RPCs associated with class: VBD

RPC name: eject

Overview: Remove the media from the device and leave it empty.
Signature:

```
void eject (session_id s, VBD ref vbd)
```

Arguments:

type	name	description
`VBD ref`	vbd	The vbd representing the CDROM-like device

Return Type: `void`

Possible Error Codes: VBD_NOT_REMOVABLE_MEDIA, VBD_IS_EMPTY

RPC name: insert

Overview: Insert new media into the device.
Signature:

```
void insert (session_id s, VBD ref vbd, VDI ref vdi)
```

Arguments:

type	name	description
VBD ref	vbd	The vbd representing the CDROM-like device
VDI ref	vdi	The new VDI to 'insert'

Return Type: void

Possible Error Codes: VBD_NOT_REMOVABLE_MEDIA, VBD_NOT_EMPTY

RPC name: plug

Overview: Hotplug the specified VBD, dynamically attaching it to the running VM.
Signature:

```
void plug (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	The VBD to hotplug

Return Type: void

RPC name: unplug

Overview: Hot-unplug the specified VBD, dynamically unattaching it from the running VM.
Signature:

```
void unplug (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	The VBD to hot-unplug

Return Type: void

RPC name: assert_attachable

Overview: Throws an error if this VBD could not be attached to this VM if the VM were running. Intended for debugging.
Signature:

```
void assert_attachable (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	The VBD to query

Return Type: `void`

RPC name: get_all

Overview: Return a list of all the VBDs known to the system.
Signature:

```
((VBD ref) Set) get_all (session_id s)
```

Return Type: `(VBD ref) Set`
references to all objects

RPC name: get_all_records_where

Overview: Return a list of all the VBDs known to the system.
Signature:

```
((VBD ref -> VBD record) Map) get_all_records_where (session_id s, string expr)
```

Arguments:

type	name	description
string	expr	expression representing records to fetch

Return Type: `(VBD ref → VBD record) Map`
records of all matching objects

RPC name: get_uuid

Overview: Get the uuid field of the given VBD.
Signature:

```
string get_uuid (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	reference to the object

Return Type: `string`

value of the field

RPC name: get_allowed_operations

Overview: Get the allowed_operations field of the given VBD.
Signature:

```
((vbd_operations) Set) get_allowed_operations (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	reference to the object

Return Type: `(vbd_operations) Set`
value of the field

RPC name: get_current_operations

Overview: Get the current_operations field of the given VBD.
Signature:

```
((string -> vbd_operations) Map) get_current_operations (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	reference to the object

Return Type: `(string → vbd_operations) Map`
value of the field

RPC name: get_VM

Overview: Get the VM field of the given VBD.
Signature:

```
(VM ref) get_VM (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	reference to the object

Return Type: `VM ref`
value of the field

RPC name: get_VDI

Overview: Get the VDI field of the given VBD.
Signature:

```
(VDI ref) get_VDI (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	reference to the object

Return Type: VDI ref
value of the field

RPC name: get_device

Overview: Get the device field of the given VBD.
Signature:

```
string get_device (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_userdevice

Overview: Get the userdevice field of the given VBD.
Signature:

```
string get_userdevice (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	reference to the object

Return Type: string
value of the field

RPC name: set_userdevice

Overview: Set the userdevice field of the given VBD.
Signature:

```
void set_userdevice (session_id s, VBD ref self, string value)
```

Arguments:

type	name	description
VBD ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_bootable

Overview: Get the bootable field of the given VBD.
Signature:

```
bool get_bootable (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	reference to the object

Return Type: bool
value of the field

RPC name: set_bootable

Overview: Set the bootable field of the given VBD.
Signature:

```
void set_bootable (session_id s, VBD ref self, bool value)
```

Arguments:

type	name	description
VBD ref	self	reference to the object
bool	value	New value to set

Return Type: void

RPC name: get_mode

Overview: Get the mode field of the given VBD.
Signature:

```
(vbd_mode) get_mode (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	reference to the object

Return Type: vbd_mode
value of the field

RPC name: set_mode

Overview: Set the mode field of the given VBD.
Signature:

```
void set_mode (session_id s, VBD ref self, vbd_mode value)
```

Arguments:

type	name	description
VBD ref	self	reference to the object
vbd_mode	value	New value to set

Return Type: void

RPC name: get_type

Overview: Get the type field of the given VBD.
Signature:

```
(vbd_type) get_type (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	reference to the object

Return Type: vbd_type
value of the field

RPC name: set_type

Overview: Set the type field of the given VBD.
Signature:

```
void set_type (session_id s, VBD ref self, vbd_type value)
```

Arguments:

type	name	description
VBD ref	self	reference to the object
vbd_type	value	New value to set

Return Type: void

RPC name: get_storage_lock

Overview: Get the storage_lock field of the given VBD.
Signature:

```
bool get_storage_lock (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	reference to the object

Return Type: bool
value of the field

RPC name: get_empty

Overview: Get the empty field of the given VBD.
Signature:

```
bool get_empty (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	reference to the object

Return Type: bool
value of the field

RPC name: get_other_config

Overview: Get the other_config field of the given VBD.
Signature:

```
((string -> string) Map) get_other_config (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: set_other_config

Overview: Set the other_config field of the given VBD.
Signature:

```
void set_other_config (session_id s, VBD ref self, (string -> string) Map value)
```

Arguments:

type	name	description
VBD ref	self	reference to the object
(string → string) Map	value	New value to set

Return Type: void

RPC name: add_to_other_config

Overview: Add the given key-value pair to the other_config field of the given VBD.
Signature:

```
void add_to_other_config (session_id s, VBD ref self, string key, string value)
```

Arguments:

type	name	description
VBD ref	self	reference to the object
string	key	Key to add
string	value	Value to add

Return Type: void

RPC name: remove_from_other_config

Overview: Remove the given key and its corresponding value from the other_config field of the given VBD. If the key is not in that Map, then do nothing.
Signature:

```
void remove_from_other_config (session_id s, VBD ref self, string key)
```

Arguments:

type	name	description
VBD ref	self	reference to the object
string	key	Key to remove

Return Type: void

RPC name: get_currently_attached

Overview: Get the currently_attached field of the given VBD.
Signature:

```
bool get_currently_attached (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	reference to the object

Return Type: bool
value of the field

RPC name: get_status_code

Overview: Get the status_code field of the given VBD.
Signature:

```
int get_status_code (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	reference to the object

Return Type: int
value of the field

RPC name: get_status_detail

Overview: Get the status_detail field of the given VBD.
Signature:

```
string get_status_detail (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_runtime_properties

Overview: Get the runtime_properties field of the given VBD.
Signature:

```
((string -> string) Map) get_runtime_properties (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: get_qos_algorithm_type

Overview: Get the qos/algorithm_type field of the given VBD.
Signature:

```
string get_qos_algorithm_type (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	reference to the object

Return Type: string
value of the field

RPC name: set_qos_algorithm_type

Overview: Set the qos/algorithm_type field of the given VBD.
Signature:

```
void set_qos_algorithm_type (session_id s, VBD ref self, string value)
```

Arguments:

type	name	description
VBD ref	self	reference to the object
string	value	New value to set

Return Type: void

RPC name: get_qos_algorithm_params

Overview: Get the qos/algorithm_params field of the given VBD.
Signature:

```
((string -> string) Map) get_qos_algorithm_params (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: set_qos_algorithm_params

Overview: Set the qos/algorithm_params field of the given VBD.
Signature:

```
void set_qos_algorithm_params (session_id s, VBD ref self, (string -> string) Map value)
```

Arguments:

type	name	description
VBD ref	self	reference to the object
(string → string) Map	value	New value to set

Return Type: void

RPC name: add_to_qos_algorithm_params

Overview: Add the given key-value pair to the qos/algorithm_params field of the given VBD.
Signature:

```
void add_to_qos_algorithm_params (session_id s, VBD ref self, string key, string value)
```

Arguments:

type	name	description
VBD ref	self	reference to the object
string	key	Key to add
string	value	Value to add

Return Type: void

RPC name: remove_from_qos_algorithm_params

Overview: Remove the given key and its corresponding value from the qos/algorithm_params field of the given VBD. If the key is not in that Map, then do nothing.
Signature:

```
void remove_from_qos_algorithm_params (session_id s, VBD ref self, string key)
```

Arguments:

type	name	description
VBD ref	self	reference to the object
string	key	Key to remove

Return Type: void

RPC name: get_qos_supported_algorithms

Overview: Get the qos/supported_algorithms field of the given VBD.
Signature:

```
(string Set) get_qos_supported_algorithms (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	reference to the object

Return Type: string Set
value of the field

RPC name: get_metrics

Overview: Get the metrics field of the given VBD.
Signature:

```
(VBD_metrics ref) get_metrics (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	reference to the object

Return Type: VBD_metrics ref

value of the field

RPC name: create

Overview: Create a new VBD instance, and return its handle.
Signature:

```
(VBD ref) create (session_id s, VBD record args)
```

Arguments:

type	name	description
VBD record	args	All constructor arguments

Return Type: VBD ref
reference to the newly created object

RPC name: destroy

Overview: Destroy the specified VBD instance.
Signature:

```
void destroy (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	reference to the object

Return Type: void

RPC name: get_by_uuid

Overview: Get a reference to the VBD instance with the specified UUID.
Signature:

```
(VBD ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
string	uuid	UUID of object to return

Return Type: VBD ref
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given VBD.
Signature:

```
(VBD record) get_record (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	reference to the object

Return Type: `VBD record`
all fields from the object

2.25 Class: VBD_metrics

2.25.1 Fields for class: VBD_metrics

Name	**VBD_metrics**		
Description	*The metrics associated with a virtual block device.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RO_{run}	`io/read_kbs`	float	Read bandwidth (KiB/s)
RO_{run}	`io/write_kbs`	float	Write bandwidth (KiB/s)
RO_{run}	`last_updated`	datetime	Time at which this information was last updated

2.25.2 RPCs associated with class: VBD_metrics

RPC name: get_all

Overview: Return a list of all the VBD_metrics instances known to the system.
Signature:

```
((VBD_metrics ref) Set) get_all (session_id s)
```

Return Type: `(VBD_metrics ref) Set`
references to all objects

RPC name: get_all_records_where

Overview: Return a list of all the VBD_metrics instances known to the system.
Signature:

```
((VBD_metrics ref -> VBD_metrics record) Map) get_all_records_where (session_id s, string expr)
```

Arguments:

type	name	description
`string`	expr	expression representing records to fetch

Return Type: `(VBD_metrics ref` → `VBD_metrics record) Map`
records of all matching objects

RPC name: get_uuid

Overview: Get the uuid field of the given VBD_metrics.
Signature:

```
string get_uuid (session_id s, VBD_metrics ref self)
```

Arguments:

type	name	description
`VBD_metrics ref`	self	reference to the object

Return Type: `string`

value of the field

RPC name: get_io_read_kbs

Overview: Get the io/read_kbs field of the given VBD_metrics.
Signature:

```
float get_io_read_kbs (session_id s, VBD_metrics ref self)
```

Arguments:

type	name	description
VBD_metrics ref	self	reference to the object

Return Type: `float`
value of the field

RPC name: get_io_write_kbs

Overview: Get the io/write_kbs field of the given VBD_metrics.
Signature:

```
float get_io_write_kbs (session_id s, VBD_metrics ref self)
```

Arguments:

type	name	description
VBD_metrics ref	self	reference to the object

Return Type: `float`
value of the field

RPC name: get_last_updated

Overview: Get the last_updated field of the given VBD_metrics.
Signature:

```
datetime get_last_updated (session_id s, VBD_metrics ref self)
```

Arguments:

type	name	description
VBD_metrics ref	self	reference to the object

Return Type: `datetime`
value of the field

RPC name: get_by_uuid

Overview: Get a reference to the VBD_metrics instance with the specified UUID.
Signature:

```
(VBD_metrics ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
string	uuid	UUID of object to return

Return Type: `VBD_metrics ref`
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given VBD_metrics.
Signature:

```
(VBD_metrics record) get_record (session_id s, VBD_metrics ref self)
```

Arguments:

type	name	description
VBD_metrics ref	self	reference to the object

Return Type: `VBD_metrics record`
all fields from the object

2.26 Class: PBD

2.26.1 Fields for class: PBD

Name	**PBD**		
Description	*The physical block devices through which hosts access SRs.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RO_{ins}	`host`	host ref	physical machine on which the pbd is available
RO_{ins}	`SR`	SR ref	the storage repository that the pbd realises
RO_{ins}	`device_config`	(string → string) Map	a config string to string map that is provided to the host's SR-backend-driver
RO_{run}	`currently_attached`	bool	is the SR currently attached on this host?

2.26.2 RPCs associated with class: PBD

RPC name: plug

Overview: Activate the specified PBD, causing the referenced SR to be attached and scanned.
Signature:

```
void plug (session_id s, PBD ref self)
```

Arguments:

type	name	description
`PBD ref`	self	The PBD to activate

Return Type: `void`

Possible Error Codes: `SR_UNKNOWN_DRIVER`

RPC name: unplug

Overview: Deactivate the specified PBD, causing the referenced SR to be detached and nolonger scanned.
Signature:

```
void unplug (session_id s, PBD ref self)
```

Arguments:

type	name	description
`PBD ref`	self	The PBD to deactivate

Return Type: `void`

RPC name: get_all

Overview: Return a list of all the PBDs known to the system.
Signature:

```
((PBD ref) Set) get_all (session_id s)
```

Return Type: `(PBD ref) Set`
references to all objects

RPC name: get_all_records_where

Overview: Return a list of all the PBDs known to the system.
Signature:

```
((PBD ref -> PBD record) Map) get_all_records_where (session_id s, string expr)
```

Arguments:

type	name	description
`string`	expr	expression representing records to fetch

Return Type: `(PBD ref → PBD record) Map`
records of all matching objects

RPC name: get_uuid

Overview: Get the uuid field of the given PBD.
Signature:

```
string get_uuid (session_id s, PBD ref self)
```

Arguments:

type	name	description
`PBD ref`	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_host

Overview: Get the host field of the given PBD.
Signature:

```
(host ref) get_host (session_id s, PBD ref self)
```

Arguments:

type	name	description
`PBD ref`	self	reference to the object

Return Type: `host ref`

value of the field

RPC name: get_SR

Overview: Get the SR field of the given PBD.
Signature:

```
(SR ref) get_SR (session_id s, PBD ref self)
```

Arguments:

type	name	description
PBD ref	self	reference to the object

Return Type: SR ref
value of the field

RPC name: get_device_config

Overview: Get the device_config field of the given PBD.
Signature:

```
((string -> string) Map) get_device_config (session_id s, PBD ref self)
```

Arguments:

type	name	description
PBD ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: get_currently_attached

Overview: Get the currently_attached field of the given PBD.
Signature:

```
bool get_currently_attached (session_id s, PBD ref self)
```

Arguments:

type	name	description
PBD ref	self	reference to the object

Return Type: bool
value of the field

RPC name: create

Overview: Create a new PBD instance, and return its handle.
Signature:

```
(PBD ref) create (session_id s, PBD record args)
```

Arguments:

type	name	description
PBD record	args	All constructor arguments

Return Type: PBD ref
reference to the newly created object

RPC name: destroy

Overview: Destroy the specified PBD instance.
Signature:

```
void destroy (session_id s, PBD ref self)
```

Arguments:

type	name	description
PBD ref	self	reference to the object

Return Type: void

RPC name: get_by_uuid

Overview: Get a reference to the PBD instance with the specified UUID.
Signature:

```
(PBD ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
string	uuid	UUID of object to return

Return Type: PBD ref
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given PBD.
Signature:

```
(PBD record) get_record (session_id s, PBD ref self)
```

Arguments:

type	name	description
`PBD ref`	self	reference to the object

Return Type: `PBD record`
all fields from the object

2.27 Class: crashdump

2.27.1 Fields for class: crashdump

Name	**crashdump**		
Description	*A VM crashdump.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RO_{ins}	`VM`	VM ref	the virtual machine
RO_{ins}	`VDI`	VDI ref	the virtual disk

2.27.2 RPCs associated with class: crashdump

RPC name: destroy

Overview: Destroy the specified crashdump.
Signature:

```
void destroy (session_id s, crashdump ref self)
```

Arguments:

type	name	description
`crashdump ref`	self	The crashdump to destroy

Return Type: `void`

RPC name: get_all

Overview: Return a list of all the crashdumps known to the system.
Signature:

```
((crashdump ref) Set) get_all (session_id s)
```

Return Type: `(crashdump ref) Set`
references to all objects

RPC name: get_all_records_where

Overview: Return a list of all the crashdumps known to the system.
Signature:

```
((crashdump ref -> crashdump record) Map) get_all_records_where (session_id s, string expr)
```

Arguments:

type	name	description
`string`	expr	expression representing records to fetch

Return Type: `(crashdump ref` → `crashdump record) Map`
records of all matching objects

RPC name: get_uuid

Overview: Get the uuid field of the given crashdump.
Signature:

```
string get_uuid (session_id s, crashdump ref self)
```

Arguments:

type	name	description
crashdump ref	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_VM

Overview: Get the VM field of the given crashdump.
Signature:

```
(VM ref) get_VM (session_id s, crashdump ref self)
```

Arguments:

type	name	description
crashdump ref	self	reference to the object

Return Type: `VM ref`
value of the field

RPC name: get_VDI

Overview: Get the VDI field of the given crashdump.
Signature:

```
(VDI ref) get_VDI (session_id s, crashdump ref self)
```

Arguments:

type	name	description
crashdump ref	self	reference to the object

Return Type: `VDI ref`
value of the field

RPC name: get_by_uuid

Overview: Get a reference to the crashdump instance with the specified UUID.
Signature:

```
(crashdump ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
`string`	uuid	UUID of object to return

Return Type: `crashdump ref`
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given crashdump.
Signature:

```
(crashdump record) get_record (session_id s, crashdump ref self)
```

Arguments:

type	name	description
`crashdump ref`	self	reference to the object

Return Type: `crashdump record`
all fields from the object

2.28 Class: VTPM

2.28.1 Fields for class: VTPM

Name	**VTPM**		
Description	*A virtual TPM device.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RO_{ins}	`VM`	VM ref	the virtual machine
RO_{ins}	`backend`	VM ref	the domain where the backend is located

2.28.2 RPCs associated with class: VTPM

RPC name: get_uuid

Overview: Get the uuid field of the given VTPM.
Signature:

```
string get_uuid (session_id s, VTPM ref self)
```

Arguments:

type	name	description
`VTPM ref`	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_VM

Overview: Get the VM field of the given VTPM.
Signature:

```
(VM ref) get_VM (session_id s, VTPM ref self)
```

Arguments:

type	name	description
`VTPM ref`	self	reference to the object

Return Type: `VM ref`
value of the field

RPC name: get_backend

Overview: Get the backend field of the given VTPM.
Signature:

```
(VM ref) get_backend (session_id s, VTPM ref self)
```

Arguments:

type	name	description
`VTPM ref`	self	reference to the object

Return Type: `VM ref`
value of the field

RPC name: create

Overview: Create a new VTPM instance, and return its handle.
Signature:

```
(VTPM ref) create (session_id s, VTPM record args)
```

Arguments:

type	name	description
`VTPM record`	args	All constructor arguments

Return Type: `VTPM ref`
reference to the newly created object

RPC name: destroy

Overview: Destroy the specified VTPM instance.
Signature:

```
void destroy (session_id s, VTPM ref self)
```

Arguments:

type	name	description
`VTPM ref`	self	reference to the object

Return Type: `void`

RPC name: get_by_uuid

Overview: Get a reference to the VTPM instance with the specified UUID.
Signature:

```
(VTPM ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
`string`	uuid	UUID of object to return

Return Type: `VTPM ref`
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given VTPM.
Signature:

```
(VTPM record) get_record (session_id s, VTPM ref self)
```

Arguments:

type	name	description
`VTPM ref`	self	reference to the object

Return Type: `VTPM record`
all fields from the object

2.29 Class: console

2.29.1 Fields for class: console

Name	**console**		
Description	*A console.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RO_{run}	`protocol`	console_protocol	the protocol used by this console
RO_{run}	`location`	string	URI for the console service
RO_{run}	`VM`	VM ref	VM to which this console is attached
RW	`other_config`	(string → string) Map	additional configuration

2.29.2 RPCs associated with class: console

RPC name: get_all

Overview: Return a list of all the consoles known to the system.
Signature:

```
((console ref) Set) get_all (session_id s)
```

Return Type: `(console ref) Set`
references to all objects

RPC name: get_all_records_where

Overview: Return a list of all the consoles known to the system.
Signature:

```
((console ref -> console record) Map) get_all_records_where (session_id s, string expr)
```

Arguments:

type	name	description
`string`	expr	expression representing records to fetch

Return Type: `(console ref → console record) Map`
records of all matching objects

RPC name: get_uuid

Overview: Get the uuid field of the given console.
Signature:

```
string get_uuid (session_id s, console ref self)
```

Arguments:

type	name	description
`console ref`	self	reference to the object

Return Type: `string`

value of the field

RPC name: get_protocol

Overview: Get the protocol field of the given console.
Signature:

```
(console_protocol) get_protocol (session_id s, console ref self)
```

Arguments:

type	name	description
console ref	self	reference to the object

Return Type: console_protocol
value of the field

RPC name: get_location

Overview: Get the location field of the given console.
Signature:

```
string get_location (session_id s, console ref self)
```

Arguments:

type	name	description
console ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_VM

Overview: Get the VM field of the given console.
Signature:

```
(VM ref) get_VM (session_id s, console ref self)
```

Arguments:

type	name	description
console ref	self	reference to the object

Return Type: VM ref
value of the field

RPC name: get_other_config

Overview: Get the other_config field of the given console.
Signature:

```
((string -> string) Map) get_other_config (session_id s, console ref self)
```

Arguments:

type	name	description
console ref	self	reference to the object

Return Type: (string → string) Map
value of the field

RPC name: set_other_config

Overview: Set the other_config field of the given console.
Signature:

```
void set_other_config (session_id s, console ref self, (string -> string) Map value)
```

Arguments:

type	name	description
console ref	self	reference to the object
(string → string) Map	value	New value to set

Return Type: void

RPC name: add_to_other_config

Overview: Add the given key-value pair to the other_config field of the given console.
Signature:

```
void add_to_other_config (session_id s, console ref self, string key, string value)
```

Arguments:

type	name	description
console ref	self	reference to the object
string	key	Key to add
string	value	Value to add

Return Type: void

RPC name: remove_from_other_config

Overview: Remove the given key and its corresponding value from the other_config field of the given console. If the key is not in that Map, then do nothing.
Signature:

```
void remove_from_other_config (session_id s, console ref self, string key)
```

Arguments:

type	name	description
`console ref`	self	reference to the object
`string`	key	Key to remove

Return Type: `void`

RPC name: create

Overview: Create a new console instance, and return its handle.
Signature:

```
(console ref) create (session_id s, console record args)
```

Arguments:

type	name	description
`console record`	args	All constructor arguments

Return Type: `console ref`
reference to the newly created object

RPC name: destroy

Overview: Destroy the specified console instance.
Signature:

```
void destroy (session_id s, console ref self)
```

Arguments:

type	name	description
`console ref`	self	reference to the object

Return Type: `void`

RPC name: get_by_uuid

Overview: Get a reference to the console instance with the specified UUID.
Signature:

```
(console ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
`string`	uuid	UUID of object to return

Return Type: `console ref`
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given console.
Signature:

```
(console record) get_record (session_id s, console ref self)
```

Arguments:

type	name	description
`console ref`	self	reference to the object

Return Type: `console record`
all fields from the object

2.30 Class: user

2.30.1 Fields for class: user

Name	**user**		
Description	*A user of the system.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	unique identifier/object reference
RO_{ins}	`short_name`	string	short name (e.g. userid)
RW	`fullname`	string	full name

2.30.2 RPCs associated with class: user

RPC name: get_uuid

Overview: Get the uuid field of the given user.
Signature:

```
string get_uuid (session_id s, user ref self)
```

Arguments:

type	name	description
`user ref`	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_short_name

Overview: Get the short_name field of the given user.
Signature:

```
string get_short_name (session_id s, user ref self)
```

Arguments:

type	name	description
`user ref`	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_fullname

Overview: Get the fullname field of the given user.
Signature:

```
string get_fullname (session_id s, user ref self)
```

Arguments:

type	name	description
`user ref`	self	reference to the object

Return Type: `string`
value of the field

RPC name: set_fullname

Overview: Set the fullname field of the given user.
Signature:

```
void set_fullname (session_id s, user ref self, string value)
```

Arguments:

type	name	description
`user ref`	self	reference to the object
`string`	value	New value to set

Return Type: `void`

RPC name: create

Overview: Create a new user instance, and return its handle.
Signature:

```
(user ref) create (session_id s, user record args)
```

Arguments:

type	name	description
`user record`	args	All constructor arguments

Return Type: `user ref`
reference to the newly created object

RPC name: destroy

Overview: Destroy the specified user instance.
Signature:

```
void destroy (session_id s, user ref self)
```

Arguments:

type	name	description
`user ref`	self	reference to the object

Return Type: `void`

RPC name: get_by_uuid

Overview: Get a reference to the user instance with the specified UUID.
Signature:

```
(user ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
`string`	uuid	UUID of object to return

Return Type: `user ref`
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given user.
Signature:

```
(user record) get_record (session_id s, user ref self)
```

Arguments:

type	name	description
`user ref`	self	reference to the object

Return Type: `user record`
all fields from the object

2.31 Class: debug

2.31.1 Fields for class: debug

Class debug has no fields.

2.31.2 RPCs associated with class: debug

RPC name: return_failure

Overview: Return an API 'successful' failure.
Signature:

```
void return_failure (session_id s)
```

Return Type: `void`

RPC name: get_all

Overview: Return a list of all the debugs known to the system.
Signature:

```
((debug ref) Set) get_all (session_id s)
```

Return Type: `(debug ref) Set`
references to all objects

RPC name: get_all_records_where

Overview: Return a list of all the debugs known to the system.
Signature:

```
((debug ref -> debug record) Map) get_all_records_where (session_id s, string expr)
```

Arguments:

type	name	description
`string`	expr	expression representing records to fetch

Return Type: `(debug ref → debug record) Map`
records of all matching objects

RPC name: create

Overview: Create a new debug instance, and return its handle.
Signature:

```
(debug ref) create (session_id s, debug record args)
```

Arguments:

type	name	description
`debug record`	args	All constructor arguments

Return Type: `debug ref`
reference to the newly created object

RPC name: destroy

Overview: Destroy the specified debug instance.
Signature:

```
void destroy (session_id s, debug ref self)
```

Arguments:

type	name	description
`debug ref`	self	reference to the object

Return Type: `void`

RPC name: get_by_uuid

Overview: Get a reference to the debug instance with the specified UUID.
Signature:

```
(debug ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

type	name	description
`string`	uuid	UUID of object to return

Return Type: `debug ref`
reference to the object

RPC name: get_record

Overview: Get a record containing the current state of the given debug.
Signature:

```
(debug record) get_record (session_id s, debug ref self)
```

Arguments:

type	name	description
`debug ref`	self	reference to the object

Return Type: `debug record`
all fields from the object

2.32 Error Handling

When a low-level transport error occurs, or a request is malformed at the HTTP or XML-RPC level, the server may send an XML-RPC Fault response, or the client may simulate the same. The client must be prepared to handle these errors, though they may be treated as fatal. On the wire, these are transmitted in a form similar to this:

```
<methodResponse>
  <fault>
    <value>
      <struct>
        <member>
            <name>faultCode</name>
            <value><int>-1</int></value>
          </member>
          <member>
            <name>faultString</name>
            <value><string>Malformed request</string></value>
        </member>
      </struct>
    </value>
  </fault>
</methodResponse>
```

All other failures are reported with a more structured error response, to allow better automatic response to failures, proper internationalisation of any error message, and easier debugging. On the wire, these are transmitted like this:

```
<struct>
  <member>
    <name>Status</name>
    <value>Failure</value>
  </member>
  <member>
    <name>ErrorDescription</name>
    <value>
      <array>
        <data>
          <value>MAP_DUPLICATE_KEY</value>
          <value>Customer</value>
          <value>eSpeil Inc.</value>
          <value>eSpeil Incorporated</value>
        </data>
      </array>
    </value>
  </member>
</struct>
```

Note that `ErrorDescription` value is an array of string values. The first element of the array is an error code; the remainder of the array are strings representing error parameters relating to that code. In this case, the client has attempted to add the mapping `Customer` → `eSpiel Incorporated` to a Map, but it already contains the mapping `Customer` → `eSpiel Inc.`, and so the request has failed.
Each possible error code is documented in the following section.

2.32.1 Error Codes

BOOTLOADER_FAILED

The bootloader returned an error

Signature:

```
BOOTLOADER_FAILED(vm, msg)
```

CHANGE_PASSWORD_REJECTED

The system rejected the password change request; perhaps the new password was too short?

Signature:

```
CHANGE_PASSWORD_REJECTED(msg)
```

DB_UNIQUENESS_CONSTRAINT_VIOLATION

You attempted an operation which would have resulted in duplicate keys in the database

Signature:

```
DB_UNIQUENESS_CONSTRAINT_VIOLATION(table, field, value)
```

DEFAULT_SR_NOT_FOUND

The default SR reference does not point to a valid SR

Signature:

```
DEFAULT_SR_NOT_FOUND(sr)
```

DEVICE_ALREADY_ATTACHED

The device is already attached to a VM

Signature:

```
DEVICE_ALREADY_ATTACHED(device)
```

DEVICE_ALREADY_DETACHED

The device is not currently attached

Signature:

```
DEVICE_ALREADY_DETACHED(device)
```

DEVICE_ALREADY_EXISTS

A device with the name given already exists on the selected VM

Signature:

```
DEVICE_ALREADY_EXISTS(device)
```

DEVICE_ATTACH_TIMEOUT

A timeout happened while attempting to attach a device to a VM

Signature:

```
DEVICE_ATTACH_TIMEOUT(type, ref)
```

DEVICE_DETACH_REJECTED

The VM rejected the attempt to detach the device

Signature:

```
DEVICE_DETACH_REJECTED(type, ref, msg)
```

DEVICE_DETACH_TIMEOUT

A timeout happened while attempting to detach a device from a VM

Signature:

```
DEVICE_DETACH_TIMEOUT(type, ref)
```

DUPLICATE_VM

Cannot restore this VM because it would create a duplicate

Signature:

```
DUPLICATE_VM(vm)
```

EVENTS_LOST

Some events have been lost from the queue and cannot be retrieved.

No parameters.

FIELD_TYPE_ERROR

One of the fields you suppiled was of the wrong type.

No parameters.

HANDLE_INVALID

You gave an invalid handle. The object may have recently been deleted. The class parameter gives the type of reference given, and the handle parameter echoes the bad value given.

Signature:

```
HANDLE_INVALID(class, handle)
```

HOSTS_NOT_HOMOGENEOUS

The hosts in this pool are not homogeneous.

No parameters.

HOST_CANNOT_DESTROY_SELF

This host cannot destroy itself.

No parameters.

HOST_CANNOT_READ_METRICS

The metrics of this host could not be read.

No parameters.

HOST_DISABLED

The specified host is disabled.

Signature:

```
HOST_DISABLED(host)
```

HOST_IN_EMERGENCY_MODE

Cannot perform operation as the host is running in emergency mode.

No parameters.

HOST_IN_USE

This operation cannot be completed as the host is in use.

No parameters.

HOST_IS_SLAVE

You cannot make regular API calls directly on a slave. Please pass API calls via the master host.

Signature:

```
HOST_IS_SLAVE(Master IP address)
```

HOST_NOT_DISABLED

This host cannot be enabled as it is not disabled.

No parameters.

HOST_NOT_ENOUGH_FREE_MEMORY

Not enough host memory is available to perform this operation

No parameters.

HOST_NOT_LIVE

This operation cannot be completed as the host is not live.

No parameters.

HOST_OFFLINE

You attempted an operation which involves a host which could not be contacted.

Signature:

```
HOST_OFFLINE(host)
```

IMPORT_ERROR

The VM could not be imported; is the file corrupt?

Signature:

```
IMPORT_ERROR(msg)
```

IMPORT_INCOMPATIBLE_VERSION

The import failed because this export has been created by a different (incompatible) product verion

No parameters.

INTERNAL_ERROR

The server failed to handle your request, due to an internal error. The given message may give details useful for debugging the problem.

Signature:

```
INTERNAL_ERROR(message)
```

INVALID_DEVICE

The device name is invalid

Signature:

```
INVALID_DEVICE(device)
```

INVALID_PATCH

The uploaded patch file is invalid

No parameters.

INVALID_VALUE

The value given is invalid

Signature:

```
INVALID_VALUE(field, value)
```

INVALID_VALUE

The value specified is of the wrong type

Signature:

```
INVALID_VALUE(field)
```

JOINING_HOST_CANNOT_BE_MASTER_OF_OTHER_HOSTS

The host joining the pool cannot already be a master of another pool.

No parameters.

JOINING_HOST_CANNOT_CONTAIN_SHARED_SRS

The host joining the pool cannot contain any shared storage.

No parameters.

JOINING_HOST_CANNOT_HAVE_RUNNING_OR_SUSPENDED_VMS

The host joining the pool cannot have any running or suspended VMs.

No parameters.

JOINING_HOST_CANNOT_HAVE_VMS_WITH_CURRENT_OPERATIONS

The host joining the pool cannot have any VMs with active tasks.

No parameters.

JOINING_HOST_CONNECTION_FAILED

There was an error connecting to the host whilst joining it the the pool.

No parameters.

LICENCE_RESTRICTION

This operation is not allow under your license. Please contact your support representative.

No parameters.

LICENSE_EXPIRED

Your license has expired. Please contact you support representative.

No parameters.

LICENSE_PROCESSING_ERROR

There was an error processing your license. Please contact your support representative

No parameters.

MAP_DUPLICATE_KEY

You tried to add a key-value pair to a map, but that key is already there. The key, current value, and the new value that you tried to set are all echoed.

Signature:

```
MAP_DUPLICATE_KEY(key, current value, new value)
```

MESSAGE_METHOD_UNKNOWN

You tried to call a method that does not exist. The method name that you used is echoed.

Signature:

```
MESSAGE_METHOD_UNKNOWN(method)
```

MESSAGE_PARAMETER_COUNT_MISMATCH

You tried to call a method with the incorrect number of parameters. The fully-qualified method name that you used, and the number of received and expected parameters are returned.

Signature:

```
MESSAGE_PARAMETER_COUNT_MISMATCH(method, expected, received)
```

NETWORK_ALREADY_CONNECTED

You tried to create a PIF, but the network you tried to attach it to is already attached to some other PIF, and so the creation failed.

Signature:

```
NETWORK_ALREADY_CONNECTED(network, connected PIF)
```

NETWORK_CONTAINS_PIF

The network contains active PIFs and cannot be deleted

Signature:

```
NETWORK_CONTAINS_PIF(pifs)
```

NETWORK_CONTAINS_VIF

The network contains active VIFs and cannot be deleted

Signature:

```
NETWORK_CONTAINS_VIF(vifs)
```

NOT_IMPLEMENTED

The function is not implemented

Signature:

```
NOT_IMPLEMENTED(function)
```

NOT_IN_EMERGENCY_MODE

This pool is not in emergency mode.

No parameters.

NO_HOSTS_AVAILABLE

There were no hosts availible to complete the specified operation.

No parameters.

OBJECT_NOLONGER_EXISTS

The specified object no longer exists.

No parameters.

OPERATION_NOT_ALLOWED

You attempted an operation that was not allowed.

Signature:

```
OPERATION_NOT_ALLOWED(reason)
```

OTHER_OPERATION_IN_PROGRESS

Another operation involving the object is currently in progress

Signature:

```
OTHER_OPERATION_IN_PROGRESS(class, object)
```

PATCH_ALREADY_APPLIED

This patch has already been applied

Signature:

```
PATCH_ALREADY_APPLIED(patch)
```

PIF_DEVICE_NOT_FOUND

The specifed device was not found.

No parameters.

PIF_IS_PHYSICAL

You tried to destroy a PIF, but it represents an aspect of the physical host configuration, and so cannot be destroyed. The parameter echoes the PIF handle you gave.

Signature:

```
PIF_IS_PHYSICAL(PIF)
```

PIF_VLAN_EXISTS

You tried to create a PIF, but it already exists

Signature:

```
PIF_VLAN_EXISTS(PIF)
```

RESTORE_INCOMPATIBLE_VERSION

The restore could not be performed because this backup has been created by a different (incompatible) product verion

No parameters.

SESSION_AUTHENTICATION_FAILED

The credentials given by the user are incorrect, so access has been denied, and you have not been issued a session handle.

No parameters.

SESSION_INVALID

You gave an invalid session handle. It may have been invalidated by a server restart, or timed out. You should get a new session handle, using one of the session.login_ calls. This error does not invalidate the current connection. The handle parameter echoes the bad value given.

Signature:

```
SESSION_INVALID(handle)
```

SESSION_NOT_REGISTERED

This session is not registered to receive events. You must call event.register before event.next. The session handle you are using is echoed.

Signature:

```
SESSION_NOT_REGISTERED(handle)
```

SR_ATTACH_FAILED

Attaching this SR failed.

Signature:

```
SR_ATTACH_FAILED(sr)
```

SR_BACKEND_FAILURE

There was an SR backend failure.

Signature:

```
SR_BACKEND_FAILURE(sr)
```

SR_DEVICE_IN_USE

The SR operation cannot be performed because a device underlying the SR is in use by the host.

No parameters.

SR_FULL

The SR is full. Requested new size exceeds the maximum size

Signature:

```
SR_FULL(requested, maximum)
```

SR_HAS_NO_PBDS

The SR has no attached PBDs

Signature:

```
SR_HAS_NO_PBDS(sr)
```

SR_HAS_PBD

The SR is still connected to a host via a PBD. It cannot be destroyed.

Signature:

```
SR_HAS_PBD(sr)
```

SR_NOT_EMPTY

The SR operation cannot be performed because the SR is not empty.

No parameters.

SR_NOT_SHARABLE

The PBD could not be plugged because the SR is in use by another host and is not marked as sharable.

Signature:

```
SR_NOT_SHARABLE(sr, host)
```

SR_OPERATION_NOT_SUPPORTED

The SR backend does not support the operation (check the SR's allowed operations)

Signature:

```
SR_OPERATION_NOT_SUPPORTED(sr)
```

SR_UNKNOWN_DRIVER

The SR could not be connected because the driver was not recognised.

Signature:

```
SR_UNKNOWN_DRIVER(driver)
```

SR_UUID_EXISTS

An SR with that uuid already exists.

Signature:

```
SR_UUID_EXISTS(uuid)
```

SR_VDI_LOCKING_FAILED

The operation could not proceed because necessary VDIs were already locked at the storage level.

No parameters.

TASK_CANCELLED

The request was asynchronously cancelled.

Signature:

```
TASK_CANCELLED(task)
```

TOO_MANY_PENDING_TASKS

The request was rejected because there are too many pending tasks on the server.

No parameters.

UNKNOWN_BOOTLOADER

The requested bootloader is unknown

Signature:

```
UNKNOWN_BOOTLOADER(vm, bootloader)
```

UUID_INVALID

The uuid you supplied was invalid.

No parameters.

VALUE_NOT_SUPPORTED

You attempted to set a value that is not supported by this implementation. The fully-qualified field name and the value that you tried to set are returned. Also returned is a developer-only diagnostic reason.

Signature:

```
VALUE_NOT_SUPPORTED(field, value, reason)
```

VBD_CDS_MUST_BE_READONLY

Read/write CDs are not supported

No parameters.

VBD_IS_EMPTY

Operation could not be performed because the drive is empty

Signature:

```
VBD_IS_EMPTY(vbd)
```

VBD_NOT_EMPTY

Operation could not be performed because the drive is not empty

Signature:

```
VBD_NOT_EMPTY(vbd)
```

VBD_NOT_REMOVABLE_MEDIA

Media could not be ejected because it is not removable

Signature:

```
VBD_NOT_REMOVABLE_MEDIA(vbd)
```

VDI_IN_USE

This operation cannot be performed because this VDI is in use by some other operation

Signature:

```
VDI_IN_USE(vdi, operation)
```

VDI_IS_NOT_ISO

This operation can only be performed on CD VDIs (iso files or CDROM drives)

Signature:

```
VDI_IS_NOT_ISO(vdi, type)
```

VDI_NOT_AVAILABLE

This operation cannot be performed because this VDI could not be properly attached to the VM.

Signature:

```
VDI_NOT_AVAILABLE(vdi)
```

VDI_READONLY

The operation required write access but this VDI is read-only

Signature:

```
VDI_READONLY(vdi)
```

VLAN_TAG_INVALID

You tried to create a VLAN, but the tag you gave was invalid – it must be between 0 and 4095. The parameter echoes the VLAN tag you gave.

Signature:

```
VLAN_TAG_INVALID(VLAN)
```

VM_BAD_POWER_STATE

You attempted an operation on a VM that was not in an appropriate power state at the time; for example, you attempted to start a VM that was already running. The parameters returned are the VM's handle, and the expected and actual VM state at the time of the call.

Signature:

```
VM_BAD_POWER_STATE(vm, expected, actual)
```

VM_CANNOT_DELETE_DEFAULT_TEMPLATE

You cannot delete the specified default template.

Signature:

```
VM_CANNOT_DELETE_DEFAULT_TEMPLATE(vm)
```

VM_FAILED_SHUTDOWN_ACKNOWLEDGMENT

VM didn't acknowledge the need to shutdown.

No parameters.

VM_HVM_REQUIRED

HVM is required for this operation

Signature:

```
VM_HVM_REQUIRED(vm)
```

VM_IS_TEMPLATE

The operation attempted is not valid for a template VM

Signature:

```
VM_IS_TEMPLATE(vm)
```

VM_MEMORY_SIZE_TOO_LOW

The specified VM has too little memory to be started.

Signature:

```
VM_MEMORY_SIZE_TOO_LOW(vm)
```

VM_MIGRATE_FAILED

An error occurred during the migration process.

Signature:

```
VM_MIGRATE_FAILED(vm, source, destination, msg)
```

VM_MISSING_PV_DRIVERS

You attempted an operation on a VM which requires PV drivers to be installed but the drivers were not detected.

Signature:

```
VM_MISSING_PV_DRIVERS(vm)
```

VM_NO_CRASHDUMP_SR

This VM does not have a crashdump SR specified.

No parameters.

VM_NO_SUSPEND_SR

This VM does not have a suspend SR specified.

No parameters.

VM_NO_VCPUS

You need at least 1 VCPU to start a VM

No parameters.

VM_OLD_PV_DRIVERS

You attempted an operation on a VM which requires a more recent version of the PV drivers. Please upgrade your PV drivers.

Signature:

```
VM_OLD_PV_DRIVERS(vm, major, minor)
```

VM_REQUIRES_NETWORK

You attempted to run a VM on a host which doesn't have a PIF on a Network needed by the VM. The VM has at least one VIF attached to the Network.

Signature:

```
VM_REQUIRES_NETWORK(vm, network)
```

VM_REQUIRES_SR

You attempted to run a VM on a host which doesn't have access to an SR needed by the VM. The VM has at least one VBD attached to a VDI in the SR.

Signature:

```
VM_REQUIRES_SR(vm, sr)
```

VM_REQUIRES_VDI

VM cannot be started because it requires a VDI which cannot be attached

Signature:

```
VM_REQUIRES_VDI(vm, vdi)
```

VM_SHUTDOWN_TIMEOUT

VM failed to shutdown before the timeout expired

Signature:

```
VM_SHUTDOWN_TIMEOUT(vm, timeout)
```

VM_TOO_MANY_VCPUS

Too many VCPUs to start this VM

No parameters.

VM_UNSAFE_BOOT

You attempted an operation on a VM that was judged to be unsafe by the server. This can happen if the VM would run on a CPU that has a potentially incompatable set of feature flags to those the VM requires. If you want to override this warning then use the 'force' option.

Signature:

```
VM_UNSAFE_BOOT(vm)
```

XMLRPC_UNMARSHAL_FAILURE

The server failed to unmarshal the XMLRPC message; it was expecting one element and received something else.

Signature:

```
XMLRPC_UNMARSHAL_FAILURE(expected, received)
```
