

Citrix XenServer Management API

Version: API Revision 1.8
Date: February 28, 2011

Copyright © 2006-2010 Citrix Systems, Inc. All Rights Reserved.

Contents

1 Introduction **3**
- 1.1 RPCs associated with fields . . . 4
- 1.2 RPCs associated with classes . . . 4
  - 1.2.1 Additional RPCs . . . 4
- 1.3 Wire Protocol for Remote API Calls . . . 5
  - 1.3.1 Note on References vs UUIDs . . . 6
  - 1.3.2 Return Values/Status Codes . . . 6
- 1.4 Making XML-RPC Calls . . . 7
  - 1.4.1 Transport Layer . . . 7
  - 1.4.2 Session Layer . . . 7
  - 1.4.3 Synchronous and Asynchronous invocation . . . 7
- 1.5 Example interactive session . . . 8
- 1.6 VM Lifecycle . . . 10
- 1.7 VM boot parameters . . . 10

2 API Reference **12**
- 2.1 Classes . . . 12
- 2.2 Relationships Between Classes . . . 13
  - 2.2.1 List of bound fields . . . 15
- 2.3 Types . . . 15
  - 2.3.1 Primitives . . . 15
  - 2.3.2 Higher order types . . . 15
  - 2.3.3 Enumeration types . . . 15
- 2.4 Class: session . . . 21
  - 2.4.1 Fields for class: session . . . 21
  - 2.4.2 RPCs associated with class: session . . . 21
- 2.5 Class: auth . . . 30
  - 2.5.1 Fields for class: auth . . . 30
  - 2.5.2 RPCs associated with class: auth . . . 30
- 2.6 Class: subject . . . 32
  - 2.6.1 Fields for class: subject . . . 32
  - 2.6.2 RPCs associated with class: subject . . . 32
- 2.7 Class: role . . . 37
  - 2.7.1 Fields for class: role . . . 37
  - 2.7.2 RPCs associated with class: role . . . 37
- 2.8 Class: task . . . 42
  - 2.8.1 Fields for class: task . . . 42
  - 2.8.2 RPCs associated with class: task . . . 43
- 2.9 Class: event . . . 52
  - 2.9.1 Fields for class: event . . . 52
  - 2.9.2 RPCs associated with class: event . . . 52
- 2.10 Class: pool . . . 54
  - 2.10.1 Fields for class: pool . . . 54

2.10.2 RPCs associated with class: pool . . . 55
2.11 Class: pool_patch . . . 83
2.11.1 Fields for class: pool_patch . . . 83
2.11.2 RPCs associated with class: pool_patch . . . 83
2.12 Class: VM . . . 91
2.12.1 Fields for class: VM . . . 91
2.12.2 RPCs associated with class: VM . . . 93
2.13 Class: VM_metrics . . . 150
2.13.1 Fields for class: VM_metrics . . . 150
2.13.2 RPCs associated with class: VM_metrics . . . 150
2.14 Class: VM_guest_metrics . . . 157
2.14.1 Fields for class: VM_guest_metrics . . . 157
2.14.2 RPCs associated with class: VM_guest_metrics . . . 157
2.15 Class: VMPP . . . 164
2.15.1 Fields for class: VMPP . . . 164
2.15.2 RPCs associated with class: VMPP . . . 164
2.16 Class: host . . . 182
2.16.1 Fields for class: host . . . 182
2.16.2 RPCs associated with class: host . . . 183
2.17 Class: host_crashdump . . . 221
2.17.1 Fields for class: host_crashdump . . . 221
2.17.2 RPCs associated with class: host_crashdump . . . 221
2.18 Class: host_patch . . . 226
2.18.1 Fields for class: host_patch . . . 226
2.18.2 RPCs associated with class: host_patch . . . 226
2.19 Class: host_metrics . . . 233
2.19.1 Fields for class: host_metrics . . . 233
2.19.2 RPCs associated with class: host_metrics . . . 233
2.20 Class: host_cpu . . . 238
2.20.1 Fields for class: host_cpu . . . 238
2.20.2 RPCs associated with class: host_cpu . . . 238
2.21 Class: network . . . 245
2.21.1 Fields for class: network . . . 245
2.21.2 RPCs associated with class: network . . . 245
2.22 Class: VIF . . . 255
2.22.1 Fields for class: VIF . . . 255
2.22.2 RPCs associated with class: VIF . . . 255
2.23 Class: VIF_metrics . . . 267
2.23.1 Fields for class: VIF_metrics . . . 267
2.23.2 RPCs associated with class: VIF_metrics . . . 267
2.24 Class: PIF . . . 271
2.24.1 Fields for class: PIF . . . 271
2.24.2 RPCs associated with class: PIF . . . 272
2.25 Class: PIF_metrics . . . 287
2.25.1 Fields for class: PIF_metrics . . . 287
2.25.2 RPCs associated with class: PIF_metrics . . . 287
2.26 Class: Bond . . . 294
2.26.1 Fields for class: Bond . . . 294
2.26.2 RPCs associated with class: Bond . . . 294
2.27 Class: VLAN . . . 299
2.27.1 Fields for class: VLAN . . . 299
2.27.2 RPCs associated with class: VLAN . . . 299
2.28 Class: SM . . . 304
2.28.1 Fields for class: SM . . . 304

2.28.2 RPCs associated with class: SM . . . 304
2.29 Class: SR . . . 311
2.29.1 Fields for class: SR . . . 311
2.29.2 RPCs associated with class: SR . . . 311
2.30 Class: VDI . . . 329
2.30.1 Fields for class: VDI . . . 329
2.30.2 RPCs associated with class: VDI . . . 330
2.31 Class: VBD . . . 353
2.31.1 Fields for class: VBD . . . 353
2.31.2 RPCs associated with class: VBD . . . 353
2.32 Class: VBD_metrics . . . 369
2.32.1 Fields for class: VBD_metrics . . . 369
2.32.2 RPCs associated with class: VBD_metrics . . . 369
2.33 Class: PBD . . . 373
2.33.1 Fields for class: PBD . . . 373
2.33.2 RPCs associated with class: PBD . . . 373
2.34 Class: crashdump . . . 379
2.34.1 Fields for class: crashdump . . . 379
2.34.2 RPCs associated with class: crashdump . . . 379
2.35 Class: VTPM . . . 383
2.35.1 Fields for class: VTPM . . . 383
2.35.2 RPCs associated with class: VTPM . . . 383
2.36 Class: console . . . 386
2.36.1 Fields for class: console . . . 386
2.36.2 RPCs associated with class: console . . . 386
2.37 Class: user . . . 391
2.37.1 Fields for class: user . . . 391
2.37.2 RPCs associated with class: user . . . 391
2.38 Class: data_source . . . 395
2.38.1 Fields for class: data_source . . . 395
2.38.2 RPCs associated with class: data_source . . . 395
2.39 Class: blob . . . 396
2.39.1 Fields for class: blob . . . 396
2.39.2 RPCs associated with class: blob . . . 396
2.40 Class: message . . . 401
2.40.1 Fields for class: message . . . 401
2.40.2 RPCs associated with class: message . . . 401
2.41 Class: secret . . . 405
2.41.1 Fields for class: secret . . . 405
2.41.2 RPCs associated with class: secret . . . 405
2.42 Class: tunnel . . . 410
2.42.1 Fields for class: tunnel . . . 410
2.42.2 RPCs associated with class: tunnel . . . 410
2.43 Error Handling . . . 416
2.43.1 Error Codes . . . 417

Chapter 1

Introduction

This document defines the Citrix XenServer Management API—an API for remotely configuring and controlling virtualised guests running on a XenServer pool.

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'
```

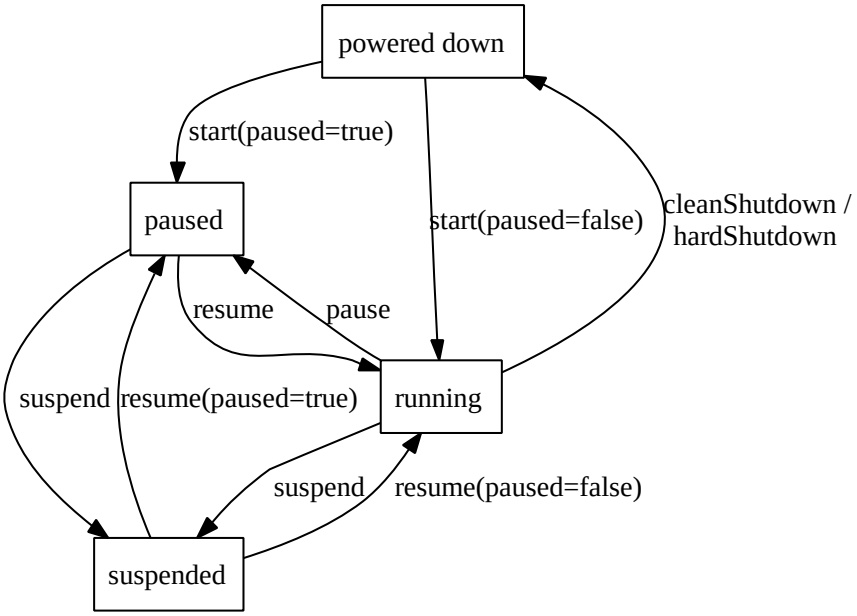


Figure 1.1: VM Lifecycle

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:

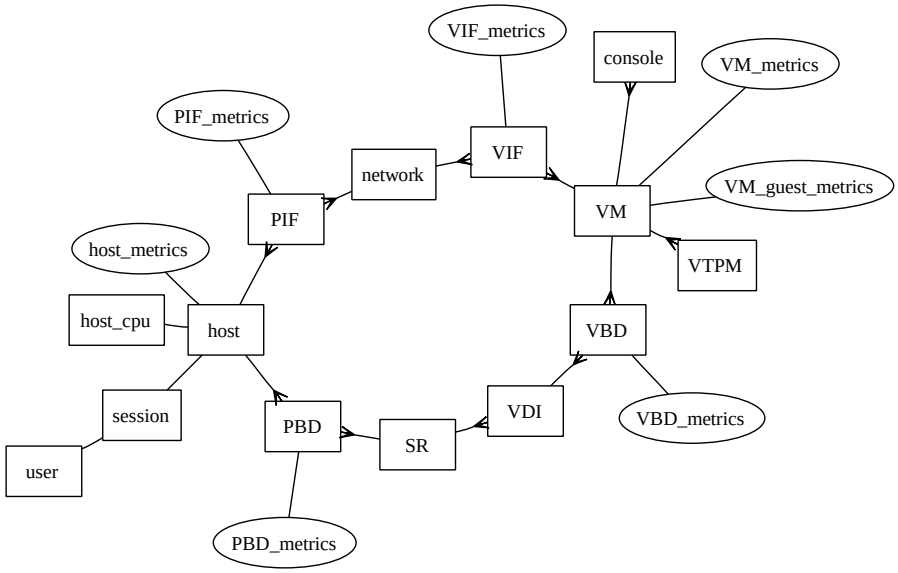
Name	Description
session	A session
auth	Management of remote authentication services
subject	A user or group that can log in xapi
role	A set of permissions associated with a subject
task	A long-running asynchronous task
event	Asynchronous event registration and handling
pool	Pool-wide information
pool_patch	Pool-wide patches
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)
VMPP	VM Protection Policy
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
Bond	
VLAN	A VLAN mux/demux
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
data_source	Data sources for logging in RRDs
blob	A placeholder for a binary blob
message	An message for the attention of the administrator
secret	A secret
tunnel	A tunnel for network traffic

2.2 Relationships Between Classes

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

object.field	*object.field*	*relationship*
VM.snapshot_of	VM.snapshots	one-to-many
VDI.snapshot_of	VDI.snapshots	one-to-many
VM.parent	VM.children	one-to-many
task.subtask_of	task.subtasks	one-to-many
task.session	session.tasks	one-to-many
PIF.bond_slave_of	Bond.slaves	one-to-many
Bond.master	PIF.bond_master_of	one-to-many
VLAN.tagged_PIF	PIF.VLAN_slave_of	one-to-many
tunnel.access_PIF	PIF.tunnel_access_PIF_of	one-to-many
tunnel.transport_PIF	PIF.tunnel_transport_PIF_of	one-to-many
PBD.host	host.PBDs	one-to-many
PBD.SR	SR.PBDs	one-to-many
VBD.VDI	VDI.VBDs	one-to-many
crashdump.VDI	VDI.crash_dumps	one-to-many
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
VDI.SR	SR.VDIs	one-to-many
VTPM.VM	VM.VTPMs	one-to-many
console.VM	VM.consoles	one-to-many
VM.resident_on	host.resident_VMs	one-to-many
host_cpu.host	host.host_CPUs	one-to-many
host_crashdump.host	host.crashdumps	one-to-many
host_patch.host	host.patches	one-to-many
host_patch.pool_patch	pool_patch.host_patches	one-to-many
subject.roles	subject.roles	unknown type
role.subroles	role.subroles	many-to-many
VM.protection_policy	VMPP.VMs	one-to-many

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
`unplug_force`	Attempting to forcibly unplug this VBD
`pause`	Attempting to pause a block device backend
`unpause`	Attempting to unpause a block device backend

enum vdi_operations	
scan	Scanning backends for new or deleted VDIs
clone	Cloning the VDI
copy	Copying the VDI
resize	Resizing the VDI
resize_online	Resizing the VDI which may or may not be online
snapshot	Snapshotting the VDI
destroy	Destroying the VDI
forget	Forget about the VDI
update	Refreshing the fields of the VDI
force_unlock	Forcibly unlocking the VDI
generate_config	Generating static configuration
blocked	Operations on this VDI are temporarily blocked

enum storage_operations	
scan	Scanning backends for new or deleted VDIs
destroy	Destroying the SR
forget	Forgetting about SR
plug	Plugging a PBD into this SR
unplug	Unplugging a PBD from this SR
update	Refresh the fields on the SR
vdi_create	Creating a new VDI
vdi_introduce	Introducing 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
evacuate	Indicates this host is evacuating
shutdown	Indicates this host is in the process of shutting itself down

reboot	Indicates this host is in the process of rebooting
power_on	Indicates this host is in the process of being powered on
vm_start	This host is starting a VM
vm_resume	This host is resuming a VM
vm_migrate	This host is the migration target of a VM

enum vm_power_state	
Halted	VM is offline and not using any resources
Paused	All resources have been allocated but the VM itself is paused and its vCPUs are not running
Running	Running
Suspended	VM state has been saved to disk and it is nolonger running. Note that disks remain in-use while

enum after_apply_guidance	
restartHVM	This patch requires HVM guests to be restarted once applied.
restartPV	This patch requires PV guests to be restarted once applied.
restartHost	This patch requires the host to be restarted once applied.
restartXAPI	This patch requires XAPI to be restarted once applied.

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 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 paused
rename_restart	rename the crashed VM and start a new copy

enum vm_operations	
snapshot	refers to the operation “snapshot”
clone	refers to the operation “clone”
copy	refers to the operation “copy”
create_template	refers to the operation “create_template”
revert	refers to the operation “revert”
checkpoint	refers to the operation “checkpoint”
snapshot_with_quiesce	refers to the operation “snapshot_with_quiesce”
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”
csvm	refers to the operation “csvm”
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”
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
awaiting_memory_live	Waiting for the memory settings to change
changing_dynamic_range	Changing the memory dynamic range
changing_static_range	Changing the memory static range
changing_memory_limits	Changing the memory limits
get_cooperative	Querying the co-operativeness of the VM
changing_shadow_memory	Changing the shadow memory for a halted VM.
changing_shadow_memory_live	Changing the shadow memory for a running VM.
changing_VCPUs	Changing VCPU settings for a halted VM.
changing_VCPUs_live	Changing VCPU settings for a running VM.
assert_operation_valid	
data_source_op	Add, remove, query or list data sources
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
metadata_export	exporting VM metadata to a network stream

reverting	Reverting the VM to a previous snapshotted state
destroy	refers to the act of uninstalling the VM

enum vmpp_backup_frequency	
hourly	Hourly backups
daily	Daily backups
weekly	Weekly backups

enum vmpp_archive_frequency	
never	Never archive
always_after_backup	Archive after backup
daily	Daily archives
weekly	Weekly backups

enum vmpp_archive_target_type	
none	No target config
cifs	CIFS target config
nfs	NFS target config

enum vmpp_backup_type	
snapshot	The backup is a snapshot
checkpoint	The backup is a checkpoint

enum ip_configuration_mode	
None	Do not acquire an IP address
DHCP	Acquire an IP address by DHCP
Static	Static IP address configuration

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
ha_statefile	a disk used for HA storage heartbeating
metadata	a disk used for HA Pool metadata
redo_log	a disk used for a general metadata redo-log

enum on_boot	
reset	When a VM containing this VDI is started, the contents of the VDI are reset to the state they were in wl
persist	Standard behaviour.

enum vbd_mode	
RO	only read-only access will be allowed
RW	read-write access will be allowed

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

enum cls	
VM	VM
Host	Host
SR	SR
Pool	Pool
VMPP	VMPP

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
RW	`other_config`	(string → string) Map	additional configuration
RO_{run}	`is_local_superuser`	bool	true iff this session was created using local superuser credentials
RO_{run}	`subject`	subject ref	references the subject instance that created the session. If a session instance has is_local_superuser set, then the value of this field is undefined.
RO_{run}	`validation_time`	datetime	time when session was last validated
RO_{run}	`auth_user_sid`	string	the subject identifier of the user that was externally authenticated. If a session instance has is_local_superuser set, then the value of this field is undefined.
RO_{run}	`auth_user_name`	string	the subject name of the user that was externally authenticated. If a session instance has is_local_superuser set, then the value of this field is undefined.
RO_{ins}	`rbac_permissions`	string Set	list with all RBAC permissions for this session
RO_{run}	`tasks`	(task ref) Set	list of tasks created using the current session
RO_{ins}	`parent`	session ref	references the parent session that created this session

2.4.2 RPCs associated with class: session

RPC name: login_with_password

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

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

Arguments:

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

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

Possible Error Codes: `SESSION_AUTHENTICATION_FAILED`

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; if your session is authenticated with root priviledges then the old_pwd is validated and the new_pwd is set regardless.
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_local_login_with_password

Overview:
Authenticate locally against a slave in emergency mode. Note the resulting sessions are only good for use on this host.
Signature:

```
(session ref) slave_local_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: local_logout

Overview:
Log out of local session.
Signature:

```
void local_logout (session_id s)
```

Return Type: `void`

RPC name: get_all_subject_identifiers

Overview:
Return a list of all the user subject-identifiers of all existing sessions.
Signature:

```
(string Set) get_all_subject_identifiers (session_id s)
```

Return Type: `string Set`
The list of user subject-identifiers of all existing sessions

RPC name: logout_subject_identifier

Overview:
Log out all sessions associated to a user subject-identifier, except the session associated with the context calling this function.
Signature:

```
void logout_subject_identifier (session_id s, string subject_identifier)
```

Arguments:

type	name	description
`string`	subject_identifier	User subject-identifier of the sessions to be destroyed

Return Type: `void`

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_other_config

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

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

Arguments:

type	name	description
session 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 session.
Signature:

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

Arguments:

type	name	description
session 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 session.
Signature:

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

Arguments:

type	name	description
session 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 session.
If the key is not in that Map, then do nothing.
Signature:

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

Arguments:

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

Return Type: void

RPC name: get_is_local_superuser

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

```
bool get_is_local_superuser (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_subject

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

```
(subject ref) get_subject (session_id s, session ref self)
```

Arguments:

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

Return Type: `subject ref`
value of the field

RPC name: get_validation_time

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

```
datetime get_validation_time (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_auth_user_sid

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

```
string get_auth_user_sid (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_auth_user_name

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

```
string get_auth_user_name (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_rbac_permissions

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

```
(string Set) get_rbac_permissions (session_id s, session ref self)
```

Arguments:

type	name	description
session ref	self	reference to the object

Return Type: `string Set`
value of the field

RPC name: get_tasks

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

```
((task ref) Set) get_tasks (session_id s, session ref self)
```

Arguments:

type	name	description
session ref	self	reference to the object

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

RPC name: get_parent

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

```
(session ref) get_parent (session_id s, session ref self)
```

Arguments:

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

Return Type: `session ref`
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: auth

2.5.1 Fields for class: auth

Class auth has no fields.

2.5.2 RPCs associated with class: auth

RPC name: get_subject_identifier

Overview:
This call queries the external directory service to obtain the subject_identifier as a string from the human-readable subject_name.
Signature:

```
string get_subject_identifier (session_id s, string subject_name)
```

Arguments:

type	name	description
`string`	subject_name	The human-readable subject_name, such as a username or a groupname

Return Type: `string`
the subject_identifier obtained from the external directory service

RPC name: get_subject_information_from_identifier

Overview:
This call queries the external directory service to obtain the user information (e.g. username, organization etc) from the specified subject_identifier.
Signature:

```
((string -> string) Map) get_subject_information_from_identifier (session_id s, string subject_identif
```

Arguments:

type	name	description
`string`	subject_identifier	A string containing the subject_identifier, unique in the external directory service

Return Type: `(string → string) Map`
key-value pairs containing at least a key called subject_name

RPC name: get_group_membership

Overview:
This calls queries the external directory service to obtain the transitively-closed set of groups that the the subject_identifier is member of.
Signature:

```
(string Set) get_group_membership (session_id s, string subject_identifier)
```

Arguments:

type	name	description
`string`	subject_identifier	A string containing the subject_identifier, unique in the external directory service

Return Type: `string Set`
set of subject_identifiers that provides the group membership of subject_identifier passed as argument, it contains, recursively, all groups a subject_identifier is member of.

2.6 Class: subject

2.6.1 Fields for class: subject

Name	**subject**		
Description	*A user or group that can log in xapi.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	Unique identifier/object reference
RO_{ins}	`subject_identifier`	string	the subject identifier, unique in the external directory service
RO_{ins}	`other_config`	(string → string) Map	additional configuration
RO_{run}	`roles`	(role ref) Set	the roles associated with this subject

2.6.2 RPCs associated with class: subject

RPC name: add_to_roles

Overview:
This call adds a new role to a subject.
Signature:

```
void add_to_roles (session_id s, subject ref self, role ref role)
```

Arguments:

type	name	description
`subject ref`	self	The subject who we want to add the role to
`role ref`	role	The unique role reference

Return Type: `void`

RPC name: remove_from_roles

Overview:
This call removes a role from a subject.
Signature:

```
void remove_from_roles (session_id s, subject ref self, role ref role)
```

Arguments:

type	name	description
`subject ref`	self	The subject from whom we want to remove the role
`role ref`	role	The unique role reference in the subject's roles field

Return Type: `void`

RPC name: get_permissions_name_label

Overview:
This call returns a list of permission names given a subject.
Signature:

```
(string Set) get_permissions_name_label (session_id s, subject ref self)
```

Arguments:

type	name	description
subject ref	self	The subject whose permissions will be retrieved

Return Type: string Set
a list of permission names

RPC name: get_all

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

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

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

RPC name: get_all_records

Overview:
Return a map of subject references to subject records for all subjects known to the system.
Signature:

```
((subject ref -> subject record) Map) get_all_records (session_id s)
```

Return Type: (subject ref → subject record) Map
records of all objects

RPC name: get_uuid

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

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

Arguments:

type	name	description
subject ref	self	reference to the object

Return Type: `string`
value of the field

RPC name: get_subject_identifier

Overview:
Get the subject_identifier field of the given subject.
Signature:

```
string get_subject_identifier (session_id s, subject ref self)
```

Arguments:

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

Return Type: `string`
value of the field

RPC name: get_other_config

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

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

Arguments:

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

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

RPC name: get_roles

Overview:
Get the roles field of the given subject.
Signature:

```
((role ref) Set) get_roles (session_id s, subject ref self)
```

Arguments:

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

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

RPC name: create

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

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

Arguments:

type	name	description
`subject record`	args	All constructor arguments

Return Type: `subject ref`
reference to the newly created object

RPC name: destroy

Overview:
Destroy the specified subject instance.
Signature:

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

Arguments:

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

Return Type: `void`

RPC name: get_by_uuid

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

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

Arguments:

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

Return Type: `subject ref`
reference to the object

RPC name: get_record

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

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

Arguments:

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

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

2.7 Class: role

2.7.1 Fields for class: role

Name	**role**		
Description	*A set of permissions associated with a subject.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	Unique identifier/object reference
RO_{ins}	`name/label`	string	a short user-friendly name for the role
RO_{ins}	`name/description`	string	what this role is for
RO_{ins}	`subroles`	(role ref) Set	a list of pointers to other roles or permissions

2.7.2 RPCs associated with class: role

RPC name: get_permissions

Overview:
This call returns a list of permissions given a role.
Signature:

```
((role ref) Set) get_permissions (session_id s, role ref self)
```

Arguments:

type	name	description
`role ref`	self	a reference to a role

Return Type: `(role ref) Set`
a list of permissions

RPC name: get_permissions_name_label

Overview:
This call returns a list of permission names given a role.
Signature:

```
(string Set) get_permissions_name_label (session_id s, role ref self)
```

Arguments:

type	name	description
`role ref`	self	a reference to a role

Return Type: `string Set`
a list of permission names

RPC name: get_by_permission

Overview:
This call returns a list of roles given a permission.
Signature:

```
((role ref) Set) get_by_permission (session_id s, role ref permission)
```

Arguments:

type	name	description
`role ref`	permission	a reference to a permission

Return Type: `(role ref) Set`
a list of references to roles

RPC name: get_by_permission_name_label

Overview:
This call returns a list of roles given a permission name.
Signature:

```
((role ref) Set) get_by_permission_name_label (session_id s, string label)
```

Arguments:

type	name	description
`string`	label	The short friendly name of the role

Return Type: `(role ref) Set`
a list of references to roles

RPC name: get_all

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

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

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

RPC name: get_all_records

Overview:
Return a map of role references to role records for all roles known to the system.
Signature:

```
((role ref -> role record) Map) get_all_records (session_id s)
```

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

RPC name: get_uuid

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

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

Arguments:

type	name	description
role 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 role.
Signature:

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

Arguments:

type	name	description
role 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 role.
Signature:

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

Arguments:

type	name	description
role ref	self	reference to the object

Return Type: string
value of the field

RPC name: get_subroles

Overview:
Get the subroles field of the given role.
Signature:

```
((role ref) Set) get_subroles (session_id s, role ref self)
```

Arguments:

type	name	description
role ref	self	reference to the object

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

RPC name: get_by_uuid

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

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

Arguments:

type	name	description
string	uuid	UUID of object to return

Return Type: `role ref`
reference to the object

RPC name: get_record

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

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

Arguments:

type	name	description
role ref	self	reference to the object

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

RPC name: get_by_name_label

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

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

Arguments:

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

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

2.8 Class: task

2.8.1 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. This list is advisory only and the server state may have changed by the time this field is read by a client.
RO_{run}	`current_operations`	(string → task_allowed_operations) Map	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
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}	`session`	session ref	the session that created 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 fraction complete (0.-1.). If task has completed (successfully or unsuccessfully) this should be 1.
RO_{run}	`externalpid`	int	If the task has spawned a program, the field record the PID of the process that the task is waiting on. (-1 if no waiting completion of an external program)
RO_{run}	`stunnelpid`	int	If the task has been forwarded, this field records the pid of the stunnel process spawned to manage the forwarding connection
RO_{run}	`forwarded`	bool	True if this task has been forwarded to a slave
RO_{run}	`forwarded_to`	host ref	The host to which the task has been forwarded
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.
RW	`other_config`	(string → string) Map	additional configuration
RO_{run}	`subtask_of`	task ref	Ref pointing to the task this is a substask of.
RO_{run}	`subtasks`	(task ref) Set	List pointing to all the substasks.

2.8.2 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:
Request that a task be cancelled. Note that a task may fail to be cancelled and may complete or fail normally and note that, even when a task does cancel, it might take an arbitrary amount of time.
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

Overview:
Return a map of task references to task records for all tasks known to the system.
Signature:

```
((task ref -> task record) Map) get_all_records (session_id s)
```

Return Type: (task ref → task record) Map
records of all 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_other_config

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

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

Arguments:

type	name	description
task 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 task.
Signature:

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

Arguments:

type	name	description
task 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 task.
Signature:

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

Arguments:

type	name	description
task 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 task. If the key is not in that Map, then do nothing.
Signature:

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

Arguments:

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

Return Type: void

RPC name: get_subtask_of

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

```
(task ref) get_subtask_of (session_id s, task ref self)
```

Arguments:

type	name	description
task ref	self	reference to the object

Return Type: task ref
value of the field

RPC name: get_subtasks

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

```
((task ref) Set) get_subtasks (session_id s, task ref self)
```

Arguments:

type	name	description
task ref	self	reference to the object

Return Type: (task ref) 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

2.9 Class: event

2.9.1 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

2.9.2 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`, `EVENTS_LOST`

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

2.10 Class: pool

2.10.1 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
RO_{run}	`ha_enabled`	bool	true if HA is enabled on the pool, false otherwise
RO_{run}	`ha_configuration`	(string → string) Map	The current HA configuration
RO_{run}	`ha_statefiles`	string Set	HA statefile VDIs in use
RO_{run}	`ha_host_failures_to_tolerate`	int	Number of host failures to tolerate before the Pool is declared to be overcommitted
RO_{run}	`ha_plan_exists_for`	int	Number of future host failures we have managed to find a plan for. Once this reaches zero any future host failures will cause the failure of protected VMs.
RW	`ha_allow_overcommit`	bool	If set to false then operations which would cause the Pool to become overcommitted will be blocked.
RO_{run}	`ha_overcommitted`	bool	True if the Pool is considered to be overcommitted i.e. if there exist insufficient physical resources to tolerate the configured number of host failures
RO_{run}	`blobs`	(string → blob ref) Map	Binary blobs associated with this pool
RW	`tags`	string Set	user-specified tags for categorization purposes
RW	`gui_config`	(string → string) Map	gui-specific configuration for pool
RO_{run}	`wlb_url`	string	Url for the configured workload balancing host
RO_{run}	`wlb_username`	string	Username for accessing the workload balancing host
RO_{run}	`wlb_password`	secret ref	Password for accessing the workload balancing host
RW	`wlb_enabled`	bool	true if workload balancing is enabled on the pool, false otherwise
RW	`wlb_verify_cert`	bool	true if communication with the WLB server should enforce SSL certificate verification.

RO_{run}	redo_log_enabled	bool	true a redo-log is to be used other than when HA is enabled, false otherwise
RO_{run}	redo_log_vdi	VDI ref	indicates the VDI to use for the redo-log other than when HA is enabled
RO_{run}	vswitch_controller	string	address of the vswitch controller
RO_{run}	restrictions	(string → string) Map	Pool-wide restrictions currently in effect

2.10.2 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

Possible Error Codes: JOINING_HOST_CANNOT_CONTAIN_SHARED_SRS

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: emergency_transition_to_master

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

```
void emergency_transition_to_master (session_id s)
```

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 (session_id s, 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: create_VLAN

Overview:
Create PIFs, mapping a network to the same physical interface/VLAN on each host. This call is deprecated: use Pool.create_VLAN_from_PIF instead.
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 create 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: create_VLAN_from_PIF

Overview:
Create a pool-wide VLAN by taking the PIF.
Signature:

```
((PIF ref) Set) create_VLAN_from_PIF (session_id s, PIF ref pif, network ref network, int VLAN)
```

Arguments:

type	name	description
PIF ref	pif	physical interface on any particular host, that identifies the PIF on which to create the (pool-wide) 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: enable_ha

Overview:
Turn on High Availability mode.
Signature:

```
void enable_ha (session_id s, (SR ref) Set heartbeat_srs, (string -> string) Map configuration)
```

Arguments:

type	name	description
(SR ref) Set	heartbeat_srs	Set of SRs to use for storage heartbeating.
(string → string) Map	configuration	Detailed HA configuration to apply

Return Type: void

RPC name: disable_ha

Overview:
Turn off High Availability mode.
Signature:

```
void disable_ha (session_id s)
```

Return Type: void

RPC name: sync_database

Overview:
Forcibly synchronise the database now.
Signature:

```
void sync_database (session_id s)
```

Return Type: void

RPC name: designate_new_master

Overview:
Perform an orderly handover of the role of master to the referenced host.
Signature:

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

Arguments:

type	name	description
host ref	host	The host who should become the new master

Return Type: void

RPC name: ha_prevent_restarts_for

Overview:
When this call returns the VM restart logic will not run for the requested number of seconds. If the argument is zero then the restart thread is immediately unblocked.
Signature:

```
void ha_prevent_restarts_for (session_id s, int seconds)
```

Arguments:

type	name	description
int	seconds	The number of seconds to block the restart thread for

Return Type: void

RPC name: ha_failover_plan_exists

Overview:
Returns true if a VM failover plan exists for up to 'n' host failures.
Signature:

```
bool ha_failover_plan_exists (session_id s, int n)
```

Arguments:

type	name	description
int	n	The number of host failures to plan for

Return Type: bool
true if a failover plan exists for the supplied number of host failures

RPC name: ha_compute_max_host_failures_to_tolerate

Overview:
Returns the maximum number of host failures we could tolerate before we would be unable to restart configured VMs.
Signature:

```
int ha_compute_max_host_failures_to_tolerate (session_id s)
```

Return Type: int
maximum value for ha_host_failures_to_tolerate given current configuration

RPC name: ha_compute_hypothetical_max_host_failures_to_tolerate

Overview:
Returns the maximum number of host failures we could tolerate before we would be unable to restart the provided VMs.
Signature:

```
int ha_compute_hypothetical_max_host_failures_to_tolerate (session_id s, (VM ref -> string) Map config
```

Arguments:

type	name	description
(VM ref → string) Map	configuration	Map of protected VM reference to restart priority

Return Type: `int`
maximum value for ha_host_failures_to_tolerate given provided configuration

RPC name: ha_compute_vm_failover_plan

Overview:
Return a VM failover plan assuming a given subset of hosts fail.
Signature:

```
((VM ref -> (string -> string) Map) Map) ha_compute_vm_failover_plan (session_id s, (host ref) Set fai
```

Arguments:

type	name	description
`(host ref) Set`	failed_hosts	The set of hosts to assume have failed
`(VM ref) Set`	failed_vms	The set of VMs to restart

Return Type: `(VM ref → (string → string) Map) Map`
VM failover plan: a map of VM to host to restart the host on

RPC name: set_ha_host_failures_to_tolerate

Overview:
Set the maximum number of host failures to consider in the HA VM restart planner.
Signature:

```
void set_ha_host_failures_to_tolerate (session_id s, pool ref self, int value)
```

Arguments:

type	name	description
`pool ref`	self	The pool
`int`	value	New number of host failures to consider

Return Type: `void`

RPC name: create_new_blob

Overview:
Create a placeholder for a named binary blob of data that is associated with this pool.
Signature:

```
(blob ref) create_new_blob (session_id s, pool ref pool, string name, string mime_type)
```

Arguments:

type	name	description
`pool ref`	pool	The pool
`string`	name	The name associated with the blob
`string`	mime_type	The mime type for the data. Empty string translates to application/octet-stream

Return Type: `blob ref`
The reference of the blob, needed for populating its data

RPC name: enable_external_auth

Overview:
This call enables external authentication on all the hosts of the pool.
Signature:

```
void enable_external_auth (session_id s, pool ref pool, (string -> string) Map config, string service_
```

Arguments:

type	name	description
`pool ref`	pool	The pool whose external authentication should be enabled
`(string → string) Map`	config	A list of key-values containing the configuration data
`string`	service_name	The name of the service
`string`	auth_type	The type of authentication (e.g. AD for Active Directory)

Return Type: `void`

RPC name: disable_external_auth

Overview:
This call disables external authentication on all the hosts of the pool.
Signature:

```
void disable_external_auth (session_id s, pool ref pool, (string -> string) Map config)
```

Arguments:

type	name	description
`pool ref`	pool	The pool whose external authentication should be disabled
`(string → string) Map`	config	Optional parameters as a list of key-values containing the configuration data

Return Type: `void`

RPC name: detect_nonhomogeneous_external_auth

Overview:
This call asynchronously detects if the external authentication configuration in any slave is different from that in the master and raises appropriate alerts.
Signature:

```
void detect_nonhomogeneous_external_auth (session_id s, pool ref pool)
```

Arguments:

type	name	description
`pool ref`	pool	The pool where to detect non-homogeneous external authentication configuration

Return Type: `void`

RPC name: initialize_wlb

Overview:
Initializes workload balancing monitoring on this pool with the specified wlb server.
Signature:

```
void initialize_wlb (session_id s, string wlb_url, string wlb_username, string wlb_password, string xe
```

Arguments:

type	name	description
`string`	wlb_url	The ip address and port to use when accessing the wlb server
`string`	wlb_username	The username used to authenticate with the wlb server
`string`	wlb_password	The password used to authenticate with the wlb server
`string`	xenserver_username	The usernamed used by the wlb server to authenticate with the xenserver
`string`	xenserver_password	The password used by the wlb server to authenticate with the xenserver

Return Type: `void`

RPC name: deconfigure_wlb

Overview:
Permanently deconfigures workload balancing monitoring on this pool.
Signature:

```
void deconfigure_wlb (session_id s)
```

Return Type: `void`

RPC name: send_wlb_configuration

Overview:
Sets the pool optimization criteria for the workload balancing server.
Signature:

```
void send_wlb_configuration (session_id s, (string -> string) Map config)
```

Arguments:

type	name	description
(string → string) Map	config	The configuration to use in optimizing this pool

Return Type: void

RPC name: retrieve_wlb_configuration

Overview:
Retrieves the pool optimization criteria from the workload balancing server.
Signature:

```
((string -> string) Map) retrieve_wlb_configuration (session_id s)
```

Return Type: (string → string) Map
The configuration used in optimizing this pool

RPC name: retrieve_wlb_recommendations

Overview:
Retrieves vm migrate recommendations for the pool from the workload balancing server.
Signature:

```
((VM ref -> string Set) Map) retrieve_wlb_recommendations (session_id s)
```

Return Type: (VM ref → string Set) Map
The list of vm migration recommendations

RPC name: send_test_post

Overview:
Send the given body to the given host and port, using HTTPS, and print the response. This is used for debugging the SSL layer.
Signature:

```
string send_test_post (session_id s, string host, int port, string body)
```

Arguments:

type	name	description
string	host	
int	port	
string	body	

Return Type: string
The response

RPC name: certificate_install

Overview:
Install an SSL certificate pool-wide.
Signature:

```
void certificate_install (session_id s, string name, string cert)
```

Arguments:

type	name	description
`string`	name	A name to give the certificate
`string`	cert	The certificate

Return Type: `void`

RPC name: certificate_uninstall

Overview:
Remove an SSL certificate.
Signature:

```
void certificate_uninstall (session_id s, string name)
```

Arguments:

type	name	description
`string`	name	The certificate name

Return Type: `void`

RPC name: certificate_list

Overview:
List all installed SSL certificates.
Signature:

```
(string Set) certificate_list (session_id s)
```

Return Type: `string Set`
All installed certificates

RPC name: crl_install

Overview:
Install an SSL certificate revocation list, pool-wide.
Signature:

```
void crl_install (session_id s, string name, string cert)
```

Arguments:

type	name	description
`string`	name	A name to give the CRL
`string`	cert	The CRL

Return Type: `void`

RPC name: crl_uninstall

Overview:
Remove an SSL certificate revocation list.
Signature:

```
void crl_uninstall (session_id s, string name)
```

Arguments:

type	name	description
`string`	name	The CRL name

Return Type: `void`

RPC name: crl_list

Overview:
List all installed SSL certificate revocation lists.
Signature:

```
(string Set) crl_list (session_id s)
```

Return Type: `string Set`
All installed CRLs

RPC name: certificate_sync

Overview:
Sync SSL certificates from master to slaves.
Signature:

```
void certificate_sync (session_id s)
```

Return Type: `void`

RPC name: enable_redo_log

Overview:
Enable the redo log on the given SR and start using it, unless HA is enabled.
Signature:

```
void enable_redo_log (session_id s, SR ref sr)
```

Arguments:

type	name	description
SR ref	sr	SR to hold the redo log.

Return Type: void

RPC name: disable_redo_log

Overview:
Disable the redo log if in use, unless HA is enabled.
Signature:

```
void disable_redo_log (session_id s)
```

Return Type: void

RPC name: set_vswitch_controller

Overview:
Set the IP address of the vswitch controller.
Signature:

```
void set_vswitch_controller (session_id s, string address)
```

Arguments:

type	name	description
string	address	IP address of the vswitch controller.

Return Type: void

RPC name: test_archive_target

Overview:
This call tests if a location is valid.
Signature:

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

Arguments:

type	name	description
pool ref	self	Reference to the pool
(string → string) Map	config	Location config settings to test

Return Type: `string`
An XMLRPC result

RPC name: enable_local_storage_caching

Overview:
This call attempts to enable pool-wide local storage caching.
Signature:

```
void enable_local_storage_caching (session_id s, pool ref self)
```

Arguments:

type	name	description
`pool ref`	self	Reference to the pool

Return Type: `void`

RPC name: disable_local_storage_caching

Overview:
This call disables pool-wide local storage caching.
Signature:

```
void disable_local_storage_caching (session_id s, pool ref self)
```

Arguments:

type	name	description
`pool ref`	self	Reference to the pool

Return Type: `void`

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

Overview:
Return a map of pool references to pool records for all pools known to the system.
Signature:

```
((pool ref -> pool record) Map) get_all_records (session_id s)
```

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

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

```
bool get_ha_enabled (session_id s, pool ref self)
```

Arguments:

type	name	description
pool ref	self	reference to the object

Return Type: bool
value of the field

RPC name: get_ha_configuration

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

```
((string -> string) Map) get_ha_configuration (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: get_ha_statefiles

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

```
(string Set) get_ha_statefiles (session_id s, pool ref self)
```

Arguments:

type	name	description
pool ref	self	reference to the object

Return Type: string Set
value of the field

RPC name: get_ha_host_failures_to_tolerate

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

```
int get_ha_host_failures_to_tolerate (session_id s, pool ref self)
```

Arguments:

type	name	description
pool ref	self	reference to the object

Return Type: int
value of the field

RPC name: get_ha_plan_exists_for

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

```
int get_ha_plan_exists_for (session_id s, pool ref self)
```

Arguments:

type	name	description
pool ref	self	reference to the object

Return Type: int
value of the field

RPC name: get_ha_allow_overcommit

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

```
bool get_ha_allow_overcommit (session_id s, pool ref self)
```

Arguments:

type	name	description
pool ref	self	reference to the object

Return Type: bool
value of the field

RPC name: set_ha_allow_overcommit

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

```
void set_ha_allow_overcommit (session_id s, pool ref self, bool value)
```

Arguments:

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

Return Type: void

RPC name: get_ha_overcommitted

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

```
bool get_ha_overcommitted (session_id s, pool ref self)
```

Arguments:

type	name	description
pool ref	self	reference to the object

Return Type: bool
value of the field

RPC name: get_blobs

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

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

Arguments:

type	name	description
pool ref	self	reference to the object

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

RPC name: get_tags

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

```
(string Set) get_tags (session_id s, pool ref self)
```

Arguments:

type	name	description
pool ref	self	reference to the object

Return Type: string Set
value of the field

RPC name: set_tags

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

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

Arguments:

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

Return Type: void

RPC name: add_tags

Overview:
Add the given value to the tags field of the given pool. If the value is already in that Set, then do nothing.
Signature:

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

Arguments:

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

Return Type: `void`

RPC name: remove_tags

Overview:
Remove the given value from the tags field of the given pool. If the value is not in that Set, then do nothing.
Signature:

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

Arguments:

type	name	description
pool ref	self	reference to the object
string	value	Value to remove

Return Type: `void`

RPC name: get_gui_config

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

```
((string -> string) Map) get_gui_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_gui_config

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

```
void set_gui_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_gui_config

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

```
void add_to_gui_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_gui_config

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

```
void remove_from_gui_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_wlb_url

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

```
string get_wlb_url (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_wlb_username

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

```
string get_wlb_username (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_wlb_enabled

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

```
bool get_wlb_enabled (session_id s, pool ref self)
```

Arguments:

type	name	description
pool ref	self	reference to the object

Return Type: bool
value of the field

RPC name: set_wlb_enabled

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

```
void set_wlb_enabled (session_id s, pool ref self, bool value)
```

Arguments:

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

Return Type: void

RPC name: get_wlb_verify_cert

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

```
bool get_wlb_verify_cert (session_id s, pool ref self)
```

Arguments:

type	name	description
pool ref	self	reference to the object

Return Type: bool
value of the field

RPC name: set_wlb_verify_cert

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

```
void set_wlb_verify_cert (session_id s, pool ref self, bool value)
```

Arguments:

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

Return Type: void

RPC name: get_redo_log_enabled

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

```
bool get_redo_log_enabled (session_id s, pool ref self)
```

Arguments:

type	name	description
pool ref	self	reference to the object

Return Type: bool
value of the field

RPC name: get_redo_log_vdi

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

```
(VDI ref) get_redo_log_vdi (session_id s, pool ref self)
```

Arguments:

type	name	description
pool ref	self	reference to the object

Return Type: VDI ref
value of the field

RPC name: get_vswitch_controller

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

```
string get_vswitch_controller (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_restrictions

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

```
((string -> string) Map) get_restrictions (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: 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

2.11 Class: pool_patch

2.11.1 Fields for class: pool_patch

Name	**pool_patch**		
Description	*Pool-wide patches.*		
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_{run}	`filename`	string	Filename of the patch
RO_{run}	`size`	int	Size of the patch
RO_{run}	`pool_applied`	bool	This patch should be applied across the entire pool
RO_{run}	`host_patches`	(host_patch ref) Set	This hosts this patch is applied to.
RO_{run}	`after_apply_guidance`	(after_apply_guidance) Set	What the client should do after this patch has been applied.
RW	`other_config`	(string → string) Map	additional configuration

2.11.2 RPCs associated with class: pool_patch

RPC name: apply

Overview:
Apply the selected patch to a host and return its output.
Signature:

```
string apply (session_id s, pool_patch ref self, host ref host)
```

Arguments:

type	name	description
`pool_patch ref`	self	The patch to apply
`host ref`	host	The host to apply the patch too

Return Type: `string`
the output of the patch application process

RPC name: pool_apply

Overview:
Apply the selected patch to all hosts in the pool and return a map of host_ref -¿ patch output.
Signature:

```
void pool_apply (session_id s, pool_patch ref self)
```

Arguments:

type	name	description
`pool_patch ref`	self	The patch to apply

Return Type: `void`

RPC name: precheck

Overview:
Execute the precheck stage of the selected patch on a host and return its output.
Signature:

```
string precheck (session_id s, pool_patch ref self, host ref host)
```

Arguments:

type	name	description
pool_patch ref	self	The patch whose prechecks will be run
host ref	host	The host to run the prechecks on

Return Type: `string`
the output of the patch prechecks

RPC name: clean

Overview:
Removes the patch's files from all hosts in the pool, but does not remove the database entries.
Signature:

```
void clean (session_id s, pool_patch ref self)
```

Arguments:

type	name	description
pool_patch ref	self	The patch to clean up

Return Type: `void`

RPC name: destroy

Overview:
Removes the patch's files from all hosts in the pool, and removes the database entries. Only works on unapplied patches.
Signature:

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

Arguments:

type	name	description
pool_patch ref	self	The patch to destroy

Return Type: `void`

RPC name: get_all

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

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

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

RPC name: get_all_records

Overview:
Return a map of pool_patch references to pool_patch records for all pool_patchs known to the system.
Signature:

```
((pool_patch ref -> pool_patch record) Map) get_all_records (session_id s)
```

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

RPC name: get_uuid

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

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

Arguments:

type	name	description
`pool_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 pool_patch.
Signature:

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

Arguments:

type	name	description
`pool_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 pool_patch.
Signature:

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

Arguments:

type	name	description
`pool_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 pool_patch.
Signature:

```
string get_version (session_id s, pool_patch ref self)
```

Arguments:

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

Return Type: `string`
value of the field

RPC name: get_size

Overview:
Get the size field of the given pool_patch.
Signature:

```
int get_size (session_id s, pool_patch ref self)
```

Arguments:

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

Return Type: `int`
value of the field

RPC name: get_pool_applied

Overview:
Get the pool_applied field of the given pool_patch.
Signature:

```
bool get_pool_applied (session_id s, pool_patch ref self)
```

Arguments:

type	name	description
pool_patch ref	self	reference to the object

Return Type: `bool`
value of the field

RPC name: get_host_patches

Overview:
Get the host_patches field of the given pool_patch.
Signature:

```
((host_patch ref) Set) get_host_patches (session_id s, pool_patch ref self)
```

Arguments:

type	name	description
pool_patch ref	self	reference to the object

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

RPC name: get_after_apply_guidance

Overview:
Get the after_apply_guidance field of the given pool_patch.
Signature:

```
((after_apply_guidance) Set) get_after_apply_guidance (session_id s, pool_patch ref self)
```

Arguments:

type	name	description
pool_patch ref	self	reference to the object

Return Type: `(after_apply_guidance) Set`
value of the field

RPC name: get_other_config

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

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

Arguments:

type	name	description
pool_patch 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_patch.
Signature:

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

Arguments:

type	name	description
pool_patch 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_patch.
Signature:

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

Arguments:

type	name	description
pool_patch 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_patch. If the key is not in that Map, then do nothing.
Signature:

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

Arguments:

type	name	description
pool_patch 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_patch instance with the specified UUID.
Signature:

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

Arguments:

type	name	description
string	uuid	UUID of object to return

Return Type: `pool_patch ref`
reference to the object

RPC name: get_record

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

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

Arguments:

type	name	description
pool_patch ref	self	reference to the object

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

RPC name: get_by_name_label

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

```
((pool_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: `(pool_patch ref) Set`
references to objects with matching names

2.12 Class: VM

2.12.1 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. This list is advisory only and the server state may have changed by the time this field is read by a client.
RO_{run}	`current_operations`	(string → vm_operations) Map	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
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
RO_{run}	`scheduled_to_be_resident_on`	host ref	the host on which the VM is due to be started/resumed/migrated. This acts as a memory reservation indicator
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.
RO_{run}	`memory/overhead`	int	Virtualization memory overhead (bytes).
RO_{ins}	`memory/target`	int	Dynamically-set memory target (bytes). The value of this field indicates the current target for memory available to this VM.
RO_{ins}	`memory/static_max`	int	Statically-set (i.e. absolute) maximum (bytes). The value of this field at VM start time acts as a hard limit of the amount of memory a guest can use. New values only take effect on reboot.
RO_{ins}	`memory/dynamic_max`	int	Dynamic maximum (bytes)
RO_{ins}	`memory/dynamic_min`	int	Dynamic minimum (bytes)

RO_{ins}	memory/static_min	int	Statically-set (i.e. absolute) mininum (bytes). The value of this field indicates the least amount of memory this VM can boot with without crashing.
RW	VCPUs/params	(string → string) Map	configuration parameters for the selected VCPU policy
RO_{ins}	VCPUs/max	int	Max number of VCPUs
RO_{ins}	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
RO_{ins}	HVM/shadow_multiplier	float	multiplier applied to the amount of shadow that will be made available to the guest
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
RO_{run}	last_booted_record	string	marshalled value containing VM record at time of last boot, updated dynamically to reflect the runtime state of the domain
RW	recommendations	string	An XML specification of recommended values and ranges for properties of this VM

RW	xenstore_data	(string → string) Map	data to be inserted into the xenstore tree (/local/domain/¡domid¿/vm-data) after the VM is created.
RO_{ins}	ha_always_run	bool	if true then the system will attempt to keep the VM running as much as possible.
RO_{ins}	ha_restart_priority	string	Only defined if ha_always_run is set possible values: "best-effort" meaning "try to restart this VM if possible but don't consider the Pool to be overcommitted if this is not possible"; and a numerical restart priority (e.g. 1, 2, 3,...)
RO_{run}	is_a_snapshot	bool	true if this is a snapshot. Snapshotted VMs can never be started, they are used only for cloning other VMs
RO_{run}	snapshot_of	VM ref	Ref pointing to the VM this snapshot is of.
RO_{run}	snapshots	(VM ref) Set	List pointing to all the VM snapshots.
RO_{run}	snapshot_time	datetime	Date/time when this snapshot was created.
RO_{run}	transportable_snapshot_id	string	Transportable ID of the snapshot VM
RO_{run}	blobs	(string → blob ref) Map	Binary blobs associated with this VM
RW	tags	string Set	user-specified tags for categorization purposes
RW	blocked_operations	(vm_operations → string) Map	List of operations which have been explicitly blocked and an error code
RO_{run}	snapshot_info	(string → string) Map	Human-readable information concerning this snapshot
RO_{run}	snapshot_metadata	string	Encoded information about the VM's metadata this is a snapshot of
RO_{run}	parent	VM ref	Ref pointing to the parent of this VM
RO_{run}	children	(VM ref) Set	List pointing to all the children of this VM
RO_{run}	bios_strings	(string → string) Map	BIOS strings
RO_{ins}	protection_policy	VMPP ref	Ref pointing to a protection policy for this VM
RO_{ins}	is_snapshot_from_vmpp	bool	true if this snapshot was created by the protection policy

2.12.2 RPCs associated with class: VM

RPC name: snapshot

Overview:
Snapshots the specified VM, making a new VM. Snapshot automatically exploits the capabilities of the underlying storage repository in which the VM's disk images are stored (e.g. Copy on Write).

Signature:

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

Arguments:

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

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

Possible Error Codes: VM_BAD_POWER_STATE, SR_FULL, OPERATION_NOT_ALLOWED

RPC name: snapshot_with_quiesce

Overview:
Snapshots the specified VM with quiesce, making a new VM. Snapshot automatically exploits the capabilities of the underlying storage repository in which the VM's disk images are stored (e.g. Copy on Write).
Signature:

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

Arguments:

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

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

Possible Error Codes: VM_BAD_POWER_STATE, SR_FULL, OPERATION_NOT_ALLOWED, VM_SNAPSHOT_WITH_QUIESCE_FAILED, VM_SNAPSHOT_WITH_QUIESCE_TIMEOUT, VM_SNAPSHOT_WITH_QUIESCE_PLUGIN_DEOS_NOT_RESPOND, VM_SNAPSHOT_WITH_QUIESCE_N

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 reference 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 reference of the newly created VM.

Possible Error Codes: VM_BAD_POWER_STATE, SR_FULL, OPERATION_NOT_ALLOWED

RPC name: revert

Overview:
Reverts the specified VM to a previous state.
Signature:

```
void revert (session_id s, VM ref snapshot)
```

Arguments:

type	name	description
VM ref	snapshot	The snapshotted state that we revert to

Return Type: void

Possible Error Codes: VM_BAD_POWER_STATE, OPERATION_NOT_ALLOWED, SR_FULL, VM_REVERT_FAILED

RPC name: checkpoint

Overview:
Checkpoints the specified VM, making a new VM. Checkpoint automatically exploits the capabilities of the underlying storage repository in which the VM's disk images are stored (e.g. Copy on Write) and saves the memory image as well.
Signature:

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

Arguments:

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

Return Type: `VM ref`
The reference of the newly created VM.

Possible Error Codes: `VM_BAD_POWER_STATE`, `SR_FULL`, `OPERATION_NOT_ALLOWED`, `VM_CHECKPOINT_SUSPEND_FAILED`, `VM_CHECKPOINT_RESUME_FAILED`

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`, `NO_HOSTS_AVAILABLE`, `LICENCE_RESTRICTION`

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). 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). 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.
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 and immediately restart the VM.
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: 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, VM_MISSING_PV_DRIVERS

RPC name: set_VCPUs_number_live

Overview:
Set the number of VCPUs for a running VM.
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_ha_restart_priority

Overview:
Set the value of the ha_restart_priority field.
Signature:

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

Arguments:

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

Return Type: void

RPC name: set_ha_always_run

Overview:
Set the value of the ha_always_run.
Signature:

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

Arguments:

type	name	description
VM ref	self	The VM
bool	value	The value

Return Type: void

RPC name: compute_memory_overhead

Overview:
Computes the virtualization memory overhead of a VM.
Signature:

```
int compute_memory_overhead (session_id s, VM ref vm)
```

Arguments:

type	name	description
VM ref	vm	The VM for which to compute the memory overhead

Return Type: int
the virtualization memory overhead of the VM.

RPC name: set_memory_dynamic_max

Overview:
Set the value of the memory_dynamic_max field.
Signature:

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

Arguments:

type	name	description
VM ref	self	The VM to modify
int	value	The new value of memory_dynamic_max

Return Type: void

RPC name: set_memory_dynamic_min

Overview:
Set the value of the memory_dynamic_min field.
Signature:

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

Arguments:

type	name	description
VM ref	self	The VM to modify
int	value	The new value of memory_dynamic_min

Return Type: void

RPC name: set_memory_dynamic_range

Overview:
Set the minimum and maximum amounts of physical memory the VM is allowed to use.
Signature:

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

Arguments:

type	name	description
VM ref	self	The VM
int	min	The new minimum value
int	max	The new maximum value

Return Type: void

RPC name: set_memory_static_max

Overview:
Set the value of the memory_static_max field.
Signature:

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

Arguments:

type	name	description
VM ref	self	The VM to modify
int	value	The new value of memory_static_max

Return Type: void

Possible Error Codes: HA_OPERATION_WOULD_BREAK_FAILOVER_PLAN

RPC name: set_memory_static_min

Overview:
Set the value of the memory_static_min field.
Signature:

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

Arguments:

type	name	description
VM ref	self	The VM to modify
int	value	The new value of memory_static_min

Return Type: void

RPC name: set_memory_static_range

Overview:
Set the static (ie boot-time) range of virtual memory that the VM is allowed to use.
Signature:

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

Arguments:

type	name	description
VM ref	self	The VM
int	min	The new minimum value
int	max	The new maximum value

Return Type: void

RPC name: set_memory_limits

Overview:
Set the memory limits of this VM.
Signature:

```
void set_memory_limits (session_id s, VM ref self, int static_min, int static_max, int dynamic_min, in
```

Arguments:

type	name	description
VM ref	self	The VM
int	static_min	The new value of memory_static_min.
int	static_max	The new value of memory_static_max.
int	dynamic_min	The new value of memory_dynamic_min.
int	dynamic_max	The new value of memory_dynamic_max.

Return Type: void

RPC name: set_memory_target_live

Overview: This message is deprecated Set the memory target for a running VM.
Signature:

```
void set_memory_target_live (session_id s, VM ref self, int target)
```

Arguments:

type	name	description
VM ref	self	The VM
int	target	The target in bytes

Return Type: void

RPC name: wait_memory_target_live

Overview: This message is deprecated Wait for a running VM to reach its current memory target.
Signature:

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

Arguments:

type	name	description
VM ref	self	The VM

Return Type: void

RPC name: get_cooperative

Overview:
Return true if the VM is currently 'co-operative' i.e. is expected to reach a balloon target and actually has done.
Signature:

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

Arguments:

type	name	description
VM ref	self	The VM

Return Type: `bool`
true if the VM is currently 'co-operative'; false otherwise

RPC name: set_HVM_shadow_multiplier

Overview:
Set the shadow memory multiplier on a halted VM.
Signature:

```
void set_HVM_shadow_multiplier (session_id s, VM ref self, float value)
```

Arguments:

type	name	description
`VM ref`	self	The VM
`float`	value	The new shadow memory multiplier to set

Return Type: `void`

RPC name: set_shadow_multiplier_live

Overview:
Set the shadow memory multiplier on a running VM.
Signature:

```
void set_shadow_multiplier_live (session_id s, VM ref self, float multiplier)
```

Arguments:

type	name	description
`VM ref`	self	The VM
`float`	multiplier	The new shadow memory multiplier to set

Return Type: `void`

RPC name: set_VCPUs_max

Overview:
Set the maximum number of VCPUs for a halted VM.
Signature:

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

Arguments:

type	name	description
`VM ref`	self	The VM
`int`	value	The new maximum number of VCPUs

Return Type: `void`

RPC name: set_VCPUs_at_startup

Overview:
Set the number of startup VCPUs for a halted VM.
Signature:

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

Arguments:

type	name	description
VM ref	self	The VM
int	value	The new maximum number of VCPUs

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: maximise_memory

Overview:
Returns the maximum amount of guest memory which will fit, together with overheads, in the supplied amount of physical memory. If 'exact' is true then an exact calculation is performed using the VM's current settings. If 'exact' is false then a more conservative approximation is used.
Signature:

```
int maximise_memory (session_id s, VM ref self, int total, bool approximate)
```

Arguments:

type	name	description
VM ref	self	The VM
int	total	Total amount of physical RAM to fit within
bool	approximate	If false the limit is calculated with the guest's current exact configuration. Otherwise a more approximate calculation is performed

Return Type: int
The maximum possible static-max

RPC name: get_boot_record

Overview:
Returns a record describing the VM's dynamic state, initialised when the VM boots and updated to reflect runtime configuration changes e.g. CPU hotplug.
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: get_data_sources

Overview:
.
Signature:

```
((data_source record) Set) get_data_sources (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	The VM to interrogate

Return Type: (data_source record) Set
A set of data sources

RPC name: record_data_source

Overview:
Start recording the specified data source.
Signature:

```
void record_data_source (session_id s, VM ref self, string data_source)
```

Arguments:

type	name	description
VM ref	self	The VM
string	data_source	The data source to record

Return Type: void

RPC name: query_data_source

Overview:
Query the latest value of the specified data source.
Signature:

```
float query_data_source (session_id s, VM ref self, string data_source)
```

Arguments:

type	name	description
VM ref	self	The VM
string	data_source	The data source to query

Return Type: float
The latest value, averaged over the last 5 seconds

RPC name: forget_data_source_archives

Overview:
Forget the recorded statistics related to the specified data source.
Signature:

```
void forget_data_source_archives (session_id s, VM ref self, string data_source)
```

Arguments:

type	name	description
VM ref	self	The VM
string	data_source	The data source whose archives are to be forgotten

Return Type: void

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

Possible Error Codes: HOST_NOT_ENOUGH_FREE_MEMORY, VM_REQUIRES_SR

RPC name: create_new_blob

Overview:
Create a placeholder for a named binary blob of data that is associated with this VM.
Signature:

```
(blob ref) create_new_blob (session_id s, VM ref vm, string name, string mime_type)
```

Arguments:

type	name	description
VM ref	vm	The VM
string	name	The name associated with the blob
string	mime_type	The mime type for the data. Empty string translates to application/octet-stream

Return Type: blob ref
The reference of the blob, needed for populating its data

RPC name: assert_agile

Overview:
Returns an error if the VM is not considered agile e.g. because it is tied to a resource local to a host.
Signature:

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

Arguments:

type	name	description
VM ref	self	The VM

Return Type: void

RPC name: retrieve_wlb_recommendations

Overview:
Returns mapping of hosts to ratings, indicating the suitability of starting the VM at that location according to wlb. Rating is replaced with an error if the VM cannot boot there.
Signature:

```
((host ref -> string Set) Map) retrieve_wlb_recommendations (session_id s, VM ref vm)
```

Arguments:

type	name	description
VM ref	vm	The VM

Return Type: (host ref → string Set) Map
The potential hosts and their corresponding recommendations or errors

RPC name: copy_bios_strings

Overview:
Copy the BIOS strings from the given host to this VM.
Signature:

```
void copy_bios_strings (session_id s, VM ref vm, host ref host)
```

Arguments:

type	name	description
VM ref	vm	The VM to modify
host ref	host	The host to copy the BIOS strings from

Return Type: `void`

RPC name: set_protection_policy

Overview:
Set the value of the protection_policy field.
Signature:

```
void set_protection_policy (session_id s, VM ref self, VMPP ref value)
```

Arguments:

type	name	description
VM ref	self	The VM
VMPP ref	value	The value

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

Overview:
Return a map of VM references to VM records for all VMs known to the system.
Signature:

```
((VM ref -> VM record) Map) get_all_records (session_id s)
```

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

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

```
int get_memory_overhead (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_memory_target

Overview: This message is deprecated Get the memory/target field of the given VM.
Signature:

```
int get_memory_target (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_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: 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: 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: 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: 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: 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: 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_HVM_shadow_multiplier

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

```
float get_HVM_shadow_multiplier (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: float
value of the field

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_last_booted_record

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

```
string get_last_booted_record (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_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: get_xenstore_data

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

```
((string -> string) Map) get_xenstore_data (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_xenstore_data

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

```
void set_xenstore_data (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_xenstore_data

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

```
void add_to_xenstore_data (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_xenstore_data

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

```
void remove_from_xenstore_data (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_ha_always_run

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

```
bool get_ha_always_run (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_ha_restart_priority

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

```
string get_ha_restart_priority (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_is_a_snapshot

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

```
bool get_is_a_snapshot (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_snapshot_of

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

```
(VM ref) get_snapshot_of (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: `VM ref`
value of the field

RPC name: get_snapshots

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

```
((VM ref) Set) get_snapshots (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

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

RPC name: get_snapshot_time

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

```
datetime get_snapshot_time (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: `datetime`
value of the field

RPC name: get_transportable_snapshot_id

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

```
string get_transportable_snapshot_id (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_blobs

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

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

Arguments:

type	name	description
VM ref	self	reference to the object

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

RPC name: get_tags

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

```
(string Set) get_tags (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: string Set
value of the field

RPC name: set_tags

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

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

Arguments:

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

Return Type: void

RPC name: add_tags

Overview:
Add the given value to the tags field of the given VM. If the value is already in that Set, then do nothing.
Signature:

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

Arguments:

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

Return Type: void

RPC name: remove_tags

Overview:
Remove the given value from the tags field of the given VM. If the value is not in that Set, then do nothing.
Signature:

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

Arguments:

type	name	description
VM ref	self	reference to the object
string	value	Value to remove

Return Type: void

RPC name: get_blocked_operations

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

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

Arguments:

type	name	description
VM ref	self	reference to the object

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

RPC name: set_blocked_operations

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

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

Arguments:

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

Return Type: void

RPC name: add_to_blocked_operations

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

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

Arguments:

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

Return Type: void

RPC name: remove_from_blocked_operations

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

```
void remove_from_blocked_operations (session_id s, VM ref self, vm_operations key)
```

Arguments:

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

Return Type: void

RPC name: get_snapshot_info

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

```
((string -> string) Map) get_snapshot_info (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_snapshot_metadata

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

```
string get_snapshot_metadata (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_parent

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

```
(VM ref) get_parent (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: VM ref
value of the field

RPC name: get_children

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

```
((VM ref) Set) get_children (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

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

RPC name: get_bios_strings

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

```
((string -> string) Map) get_bios_strings (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_protection_policy

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

```
(VMPP ref) get_protection_policy (session_id s, VM ref self)
```

Arguments:

type	name	description
VM ref	self	reference to the object

Return Type: `VMPP ref`
value of the field

RPC name: get_is_snapshot_from_vmpp

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

```
bool get_is_snapshot_from_vmpp (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: 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

2.13 Class: VM_metrics

2.13.1 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
RW	`other_config`	(string → string) Map	additional configuration

2.13.2 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

Overview:
Return a map of VM_metrics references to VM_metrics records for all VM_metrics instances known to the system.
Signature:

```
((VM_metrics ref -> VM_metrics record) Map) get_all_records (session_id s)
```

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

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

```
((string -> string) Map) get_other_config (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: set_other_config

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

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

Arguments:

type	name	description
VM_metrics 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_metrics.
Signature:

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

Arguments:

type	name	description
VM_metrics 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_metrics.
If the key is not in that Map, then do nothing.
Signature:

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

Arguments:

type	name	description
VM_metrics 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 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

2.14 Class: VM_guest_metrics

2.14.1 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
RW	`other_config`	(string → string) Map	additional configuration
RO_{run}	`live`	bool	True if the guest is sending heartbeat messages via the guest agent

2.14.2 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

Overview:
Return a map of VM_guest_metrics references to VM_guest_metrics records for all VM_guest_metrics instances known to the system.
Signature:

```
((VM_guest_metrics ref -> VM_guest_metrics record) Map) get_all_records (session_id s)
```

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

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

```
((string -> string) Map) get_other_config (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: set_other_config

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

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

Arguments:

type	name	description
VM_guest_metrics 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_guest_metrics.
Signature:

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

Arguments:

type	name	description
VM_guest_metrics 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_guest_metrics. If the key is not in that Map, then do nothing.
Signature:

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

Arguments:

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

Return Type: void

RPC name: get_live

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

```
bool get_live (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_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

2.15 Class: VMPP

2.15.1 Fields for class: VMPP

Name	**VMPP**		
Description	*VM Protection Policy.*		
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
RW	`is_policy_enabled`	bool	enable or disable this policy
RW	`backup_type`	vmpp_backup_type	type of the backup sub-policy
RO_{ins}	`backup_retention_value`	int	maximum number of backups that should be stored at any time
RO_{ins}	`backup_frequency`	vmpp_backup_frequency	frequency of the backup schedule
RO_{ins}	`backup_schedule`	(string → string) Map	schedule of the backup containing 'hour', 'min', 'days'. Date/time-related information is in XenServer Local Timezone
RO_{run}	`is_backup_running`	bool	true if this protection policy's backup is running
RO_{run}	`backup_last_run_time`	datetime	time of the last backup
RO_{ins}	`archive_target_type`	vmpp_archive_target_type	type of the archive target config
RO_{ins}	`archive_target_config`	(string → string) Map	configuration for the archive, including its 'location', 'username', 'password'
RO_{ins}	`archive_frequency`	vmpp_archive_frequency	frequency of the archive schedule
RO_{ins}	`archive_schedule`	(string → string) Map	schedule of the archive containing 'hour', 'min', 'days'. Date/time-related information is in XenServer Local Timezone
RO_{run}	`is_archive_running`	bool	true if this protection policy's archive is running
RO_{run}	`archive_last_run_time`	datetime	time of the last archive
RO_{run}	`VMs`	(VM ref) Set	all VMs attached to this protection policy
RO_{ins}	`is_alarm_enabled`	bool	true if alarm is enabled for this policy
RO_{ins}	`alarm_config`	(string → string) Map	configuration for the alarm
RO_{run}	`recent_alerts`	string Set	recent alerts

2.15.2 RPCs associated with class: VMPP

RPC name: protect_now

Overview:
This call executes the protection policy immediately.
Signature:

```
string protect_now (session_id s, VMPP ref vmpp)
```

Arguments:

type	name	description
`VMPP ref`	vmpp	The protection policy to execute

Return Type: `string`
An XMLRPC result

RPC name: archive_now

Overview:
This call archives the snapshot provided as a parameter.
Signature:

```
string archive_now (session_id s, VM ref snapshot)
```

Arguments:

type	name	description
`VM ref`	snapshot	The snapshot to archive

Return Type: `string`
An XMLRPC result

RPC name: get_alerts

Overview:
This call fetches a history of alerts for a given protection policy.
Signature:

```
(string Set) get_alerts (session_id s, VMPP ref vmpp, int hours_from_now)
```

Arguments:

type	name	description
`VMPP ref`	vmpp	The protection policy
`int`	hours_from_now	how many hours in the past the oldest record to fetch is

Return Type: `string Set`
A list of alerts encoded in xml

RPC name: set_backup_retention_value

Overview:
.
Signature:

```
void set_backup_retention_value (session_id s, VMPP ref self, int value)
```

Arguments:

type	name	description
`VMPP ref`	self	The protection policy
`int`	value	the value to set

Return Type: `void`

RPC name: set_backup_frequency

Overview:
Set the value of the backup_frequency field.
Signature:

```
void set_backup_frequency (session_id s, VMPP ref self, vmpp_backup_frequency value)
```

Arguments:

type	name	description
VMPP ref	self	The protection policy
vmpp_backup_frequency	value	the backup frequency

Return Type: void

RPC name: set_backup_schedule

Overview:
.
Signature:

```
void set_backup_schedule (session_id s, VMPP ref self, (string -> string) Map value)
```

Arguments:

type	name	description
VMPP ref	self	The protection policy
(string → string) Map	value	the value to set

Return Type: void

RPC name: set_archive_frequency

Overview:
Set the value of the archive_frequency field.
Signature:

```
void set_archive_frequency (session_id s, VMPP ref self, vmpp_archive_frequency value)
```

Arguments:

type	name	description
VMPP ref	self	The protection policy
vmpp_archive_frequency	value	the archive frequency

Return Type: void

RPC name: set_archive_schedule

Overview:

.

Signature:

```
void set_archive_schedule (session_id s, VMPP ref self, (string -> string) Map value)
```

Arguments:

type	name	description
VMPP ref	self	The protection policy
(string → string) Map	value	the value to set

Return Type: void

RPC name: set_archive_target_type

Overview:
Set the value of the archive_target_config_type field.
Signature:

```
void set_archive_target_type (session_id s, VMPP ref self, vmpp_archive_target_type value)
```

Arguments:

type	name	description
VMPP ref	self	The protection policy
vmpp_archive_target_type	value	the archive target config type

Return Type: void

RPC name: set_archive_target_config

Overview:

.

Signature:

```
void set_archive_target_config (session_id s, VMPP ref self, (string -> string) Map value)
```

Arguments:

type	name	description
VMPP ref	self	The protection policy
(string → string) Map	value	the value to set

Return Type: void

RPC name: set_is_alarm_enabled

Overview:
Set the value of the is_alarm_enabled field.
Signature:

```
void set_is_alarm_enabled (session_id s, VMPP ref self, bool value)
```

Arguments:

type	name	description
VMPP ref	self	The protection policy
bool	value	true if alarm is enabled for this policy

Return Type: void

RPC name: set_alarm_config

Overview:
.
Signature:

```
void set_alarm_config (session_id s, VMPP ref self, (string -> string) Map value)
```

Arguments:

type	name	description
VMPP ref	self	The protection policy
(string → string) Map	value	the value to set

Return Type: void

RPC name: add_to_backup_schedule

Overview:
.
Signature:

```
void add_to_backup_schedule (session_id s, VMPP ref self, string key, string value)
```

Arguments:

type	name	description
VMPP ref	self	The protection policy
string	key	the key to add
string	value	the value to add

Return Type: void

RPC name: add_to_archive_target_config

Overview:

.

Signature:

```
void add_to_archive_target_config (session_id s, VMPP ref self, string key, string value)
```

Arguments:

type	name	description
VMPP ref	self	The protection policy
string	key	the key to add
string	value	the value to add

Return Type: void

RPC name: add_to_archive_schedule

Overview:

.

Signature:

```
void add_to_archive_schedule (session_id s, VMPP ref self, string key, string value)
```

Arguments:

type	name	description
VMPP ref	self	The protection policy
string	key	the key to add
string	value	the value to add

Return Type: void

RPC name: add_to_alarm_config

Overview:

.

Signature:

```
void add_to_alarm_config (session_id s, VMPP ref self, string key, string value)
```

Arguments:

type	name	description
VMPP ref	self	The protection policy
string	key	the key to add
string	value	the value to add

Return Type: void

RPC name: remove_from_backup_schedule

Overview:

.

Signature:

```
void remove_from_backup_schedule (session_id s, VMPP ref self, string key)
```

Arguments:

type	name	description
VMPP ref	self	The protection policy
string	key	the key to remove

Return Type: void

RPC name: remove_from_archive_target_config

Overview:

.

Signature:

```
void remove_from_archive_target_config (session_id s, VMPP ref self, string key)
```

Arguments:

type	name	description
VMPP ref	self	The protection policy
string	key	the key to remove

Return Type: void

RPC name: remove_from_archive_schedule

Overview:

.

Signature:

```
void remove_from_archive_schedule (session_id s, VMPP ref self, string key)
```

Arguments:

type	name	description
VMPP ref	self	The protection policy
string	key	the key to remove

Return Type: void

RPC name: remove_from_alarm_config

Overview:

.

Signature:

```
void remove_from_alarm_config (session_id s, VMPP ref self, string key)
```

Arguments:

type	name	description
VMPP ref	self	The protection policy
string	key	the key to remove

Return Type: void

RPC name: set_backup_last_run_time

Overview:

.

Signature:

```
void set_backup_last_run_time (session_id s, VMPP ref self, datetime value)
```

Arguments:

type	name	description
VMPP ref	self	The protection policy
datetime	value	the value to set

Return Type: void

RPC name: set_archive_last_run_time

Overview:

.

Signature:

```
void set_archive_last_run_time (session_id s, VMPP ref self, datetime value)
```

Arguments:

type	name	description
VMPP ref	self	The protection policy
datetime	value	the value to set

Return Type: void

RPC name: get_all

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

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

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

RPC name: get_all_records

Overview:
Return a map of VMPP references to VMPP records for all VMPPs known to the system.
Signature:

```
((VMPP ref -> VMPP record) Map) get_all_records (session_id s)
```

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

RPC name: get_uuid

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

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

Arguments:

type	name	description
`VMPP 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 VMPP.
Signature:

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

Arguments:

type	name	description
`VMPP 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 VMPP.
Signature:

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

Arguments:

type	name	description
VMPP 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 VMPP.
Signature:

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

Arguments:

type	name	description
VMPP 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 VMPP.
Signature:

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

Arguments:

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

Return Type: void

RPC name: get_is_policy_enabled

Overview:
Get the is_policy_enabled field of the given VMPP.
Signature:

```
bool get_is_policy_enabled (session_id s, VMPP ref self)
```

Arguments:

type	name	description
VMPP ref	self	reference to the object

Return Type: bool
value of the field

RPC name: set_is_policy_enabled

Overview:
Set the is_policy_enabled field of the given VMPP.
Signature:

```
void set_is_policy_enabled (session_id s, VMPP ref self, bool value)
```

Arguments:

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

Return Type: void

RPC name: get_backup_type

Overview:
Get the backup_type field of the given VMPP.
Signature:

```
(vmpp_backup_type) get_backup_type (session_id s, VMPP ref self)
```

Arguments:

type	name	description
VMPP ref	self	reference to the object

Return Type: vmpp_backup_type
value of the field

RPC name: set_backup_type

Overview:
Set the backup_type field of the given VMPP.
Signature:

```
void set_backup_type (session_id s, VMPP ref self, vmpp_backup_type value)
```

Arguments:

type	name	description
VMPP ref	self	reference to the object
vmpp_backup_type	value	New value to set

Return Type: void

RPC name: get_backup_retention_value

Overview:
Get the backup_retention_value field of the given VMPP.
Signature:

```
int get_backup_retention_value (session_id s, VMPP ref self)
```

Arguments:

type	name	description
VMPP ref	self	reference to the object

Return Type: int
value of the field

RPC name: get_backup_frequency

Overview:
Get the backup_frequency field of the given VMPP.
Signature:

```
(vmpp_backup_frequency) get_backup_frequency (session_id s, VMPP ref self)
```

Arguments:

type	name	description
VMPP ref	self	reference to the object

Return Type: vmpp_backup_frequency
value of the field

RPC name: get_backup_schedule

Overview:
Get the backup_schedule field of the given VMPP.
Signature:

```
((string -> string) Map) get_backup_schedule (session_id s, VMPP ref self)
```

Arguments:

type	name	description
VMPP ref	self	reference to the object

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

RPC name: get_is_backup_running

Overview:
Get the is_backup_running field of the given VMPP.
Signature:

```
bool get_is_backup_running (session_id s, VMPP ref self)
```

Arguments:

type	name	description
VMPP ref	self	reference to the object

Return Type: bool
value of the field

RPC name: get_backup_last_run_time

Overview:
Get the backup_last_run_time field of the given VMPP.
Signature:

```
datetime get_backup_last_run_time (session_id s, VMPP ref self)
```

Arguments:

type	name	description
VMPP ref	self	reference to the object

Return Type: datetime
value of the field

RPC name: get_archive_target_type

Overview:
Get the archive_target_type field of the given VMPP.
Signature:

```
(vmpp_archive_target_type) get_archive_target_type (session_id s, VMPP ref self)
```

Arguments:

type	name	description
VMPP ref	self	reference to the object

Return Type: vmpp_archive_target_type
value of the field

RPC name: get_archive_target_config

Overview:
Get the archive_target_config field of the given VMPP.
Signature:

```
((string -> string) Map) get_archive_target_config (session_id s, VMPP ref self)
```

Arguments:

type	name	description
VMPP ref	self	reference to the object

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

RPC name: get_archive_frequency

Overview:
Get the archive_frequency field of the given VMPP.
Signature:

```
(vmpp_archive_frequency) get_archive_frequency (session_id s, VMPP ref self)
```

Arguments:

type	name	description
VMPP ref	self	reference to the object

Return Type: vmpp_archive_frequency
value of the field

RPC name: get_archive_schedule

Overview:
Get the archive_schedule field of the given VMPP.
Signature:

```
((string -> string) Map) get_archive_schedule (session_id s, VMPP ref self)
```

Arguments:

type	name	description
VMPP ref	self	reference to the object

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

RPC name: get_is_archive_running

Overview:
Get the is_archive_running field of the given VMPP.
Signature:

```
bool get_is_archive_running (session_id s, VMPP ref self)
```

Arguments:

type	name	description
VMPP ref	self	reference to the object

Return Type: bool
value of the field

RPC name: get_archive_last_run_time

Overview:
Get the archive_last_run_time field of the given VMPP.
Signature:

```
datetime get_archive_last_run_time (session_id s, VMPP ref self)
```

Arguments:

type	name	description
VMPP ref	self	reference to the object

Return Type: datetime
value of the field

RPC name: get_VMs

Overview:
Get the VMs field of the given VMPP.
Signature:

```
((VM ref) Set) get_VMs (session_id s, VMPP ref self)
```

Arguments:

type	name	description
VMPP ref	self	reference to the object

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

RPC name: get_is_alarm_enabled

Overview:
Get the is_alarm_enabled field of the given VMPP.
Signature:

```
bool get_is_alarm_enabled (session_id s, VMPP ref self)
```

Arguments:

type	name	description
VMPP ref	self	reference to the object

Return Type: `bool`
value of the field

RPC name: get_alarm_config

Overview:
Get the alarm_config field of the given VMPP.
Signature:

```
((string -> string) Map) get_alarm_config (session_id s, VMPP ref self)
```

Arguments:

type	name	description
VMPP ref	self	reference to the object

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

RPC name: get_recent_alerts

Overview:
Get the recent_alerts field of the given VMPP.
Signature:

```
(string Set) get_recent_alerts (session_id s, VMPP ref self)
```

Arguments:

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

Return Type: `string Set`
value of the field

RPC name: create

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

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

Arguments:

type	name	description
`VMPP record`	args	All constructor arguments

Return Type: `VMPP ref`
reference to the newly created object

RPC name: destroy

Overview:
Destroy the specified VMPP instance.
Signature:

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

Arguments:

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

Return Type: `void`

RPC name: get_by_uuid

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

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

Arguments:

type	name	description
string	uuid	UUID of object to return

Return Type: `VMPP ref`
reference to the object

RPC name: get_record

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

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

Arguments:

type	name	description
VMPP ref	self	reference to the object

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

RPC name: get_by_name_label

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

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

Arguments:

type	name	description
string	label	label of object to return

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

2.16 Class: host

2.16.1 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}	memory/overhead	int	Virtualization memor (bytes).
RO_{run}	allowed_operations	(host_allowed_operations) Set	list of the operations al state. This list is advis the server state may hav the time this field is rea
RO_{run}	current_operations	(string → host_allowed_operations) Map	links each of the runnin this object (by referenc rent_operation enum wh the nature of the task.
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
RO_{run}	cpu_info	(string → string) Map	Details about the physi this host
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	State of the current lice
RO_{run}	boot_free_mem	int	Free memory on host at
RO_{run}	ha_statefiles	string Set	The set of statefiles ac this host
RO_{run}	ha_network_peers	string Set	The set of hosts visible work from this host
RO_{run}	blobs	(string → blob ref) Map	Binary blobs associate host
RW	tags	string Set	user-specified tags for c purposes
RO_{run}	external_auth_type	string	type of external authe vice configured; empty figured.
RO_{run}	external_auth_service_name	string	name of external authe vice configured; empty i ured.
RO_{run}	external_auth_configuration	(string → string) Map	configuration specific to thentication service
RO_{run}	edition	string	XenServer edition
RW	license_server	(string → string) Map	Contact information o server
RO_{run}	bios_strings	(string → string) Map	BIOS strings
RO_{run}	power_on_mode	string	The power on mode
RO_{run}	power_on_config	(string → string) Map	The power on config
RO_{run}	local_cache_sr	SR ref	The SR that is used as

2.16.2 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 Citrix 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

Possible Error Codes: LICENSE_PROCESSING_ERROR

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: power_on

Overview:
Attempt to power-on the host (if the capability exists).
Signature:

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

Arguments:

type	name	description
host ref	host	The Host to power on

Return Type: void

RPC name: emergency_ha_disable

Overview:
This call disables HA on the local host. This should only be used with extreme care.
Signature:

```
void emergency_ha_disable (session_id s)
```

Return Type: void

RPC name: get_data_sources

Overview:
.
Signature:

```
((data_source record) Set) get_data_sources (session_id s, host ref host)
```

Arguments:

type	name	description
host ref	host	The host to interrogate

Return Type: (data_source record) Set
A set of data sources

RPC name: record_data_source

Overview:
Start recording the specified data source.
Signature:

```
void record_data_source (session_id s, host ref host, string data_source)
```

Arguments:

type	name	description
host ref	host	The host
string	data_source	The data source to record

Return Type: void

RPC name: query_data_source

Overview:
Query the latest value of the specified data source.
Signature:

```
float query_data_source (session_id s, host ref host, string data_source)
```

Arguments:

type	name	description
host ref	host	The host
string	data_source	The data source to query

Return Type: float
The latest value, averaged over the last 5 seconds

RPC name: forget_data_source_archives

Overview:
Forget the recorded statistics related to the specified data source.
Signature:

```
void forget_data_source_archives (session_id s, host ref host, string data_source)
```

Arguments:

type	name	description
host ref	host	The host
string	data_source	The data source whose archives are to be forgotten

Return Type: void

RPC name: assert_can_evacuate

Overview:
Check this host can be evacuated.
Signature:

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

Arguments:

type	name	description
host ref	host	The host to evacuate

Return Type: void

RPC name: get_vms_which_prevent_evacuation

Overview:
Return a set of VMs which prevent the host being evacuated, with per-VM error codes.
Signature:

```
((VM ref -> string Set) Map) get_vms_which_prevent_evacuation (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	The host to query

Return Type: (VM ref → string Set) Map
VMs which block evacuation together with reasons

RPC name: get_uncooperative_resident_VMs

Overview:
Return a set of VMs which are not co-operating with the host's memory control system.
Signature:

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

Arguments:

type	name	description
host ref	self	The host to query

Return Type: (VM ref) Set
VMs which are not co-operating

RPC name: evacuate

Overview:
Migrate all VMs off of this host, where possible.
Signature:

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

Arguments:

type	name	description
host ref	host	The host to evacuate

Return Type: void

RPC name: syslog_reconfigure

Overview:
Re-configure syslog logging.
Signature:

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

Arguments:

type	name	description
host ref	host	Tell the host to reread its Host.logging parameters and reconfigure itself accordingly

Return Type: void

RPC name: management_reconfigure

Overview:
Reconfigure the management network interface.
Signature:

```
void management_reconfigure (session_id s, PIF ref pif)
```

Arguments:

type	name	description
PIF ref	pif	reference to a PIF object corresponding to the management interface

Return Type: void

RPC name: local_management_reconfigure

Overview:
Reconfigure the management network interface. Should only be used if Host.management_reconfigure is impossible because the network configuration is broken.
Signature:

```
void local_management_reconfigure (session_id s, string interface)
```

Arguments:

type	name	description
`string`	interface	name of the interface to use as a management interface

Return Type: `void`

RPC name: management_disable

Overview:
Disable the management network interface.
Signature:

```
void management_disable (session_id s)
```

Return Type: `void`

RPC name: get_system_status_capabilities

Overview:
.
Signature:

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

Arguments:

type	name	description
`host ref`	host	The host to interrogate

Return Type: `string`
An XML fragment containing the system status capabilities.

RPC name: restart_agent

Overview:
Restarts the agent after a 10 second pause. WARNING: this is a dangerous operation. Any operations in progress will be aborted, and unrecoverable data loss may occur. The caller is responsible for ensuring that there are no operations in progress when this method is called.
Signature:

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

Arguments:

type	name	description
`host ref`	host	The Host on which you want to restart the agent

Return Type: `void`

RPC name: shutdown_agent

Overview:
Shuts the agent down after a 10 second pause. WARNING: this is a dangerous operation. Any operations in progress will be aborted, and unrecoverable data loss may occur. The caller is responsible for ensuring that there are no operations in progress when this method is called.
Signature:

```
void shutdown_agent (session_id s)
```

Return Type: `void`

RPC name: set_hostname_live

Overview:
Sets the host name to the specified string. Both the API and lower-level system hostname are changed immediately.
Signature:

```
void set_hostname_live (session_id s, host ref host, string hostname)
```

Arguments:

type	name	description
`host ref`	host	The host whose host name to set
`string`	hostname	The new host name

Return Type: `void`

Possible Error Codes: `HOST_NAME_INVALID`

RPC name: compute_free_memory

Overview:
Computes the amount of free memory on the host.
Signature:

```
int compute_free_memory (session_id s, host ref host)
```

Arguments:

type	name	description
`host ref`	host	The host to send the request to

Return Type: `int`
the amount of free memory on the host.

RPC name: compute_memory_overhead

Overview:
Computes the virtualization memory overhead of a host.
Signature:

```
int compute_memory_overhead (session_id s, host ref host)
```

Arguments:

type	name	description
host ref	host	The host for which to compute the memory overhead

Return Type: int
the virtualization memory overhead of the host.

RPC name: sync_data

Overview:
This causes the synchronisation of the non-database data (messages, RRDs and so on) stored on the master to be synchronised with the host.
Signature:

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

Arguments:

type	name	description
host ref	host	The host to whom the data should be sent

Return Type: void

RPC name: backup_rrds

Overview:
This causes the RRDs to be backed up to the master.
Signature:

```
void backup_rrds (session_id s, host ref host, float delay)
```

Arguments:

type	name	description
host ref	host	Schedule a backup of the RRDs of this host
float	delay	Delay in seconds from when the call is received to perform the backup

Return Type: void

RPC name: create_new_blob

Overview:
Create a placeholder for a named binary blob of data that is associated with this host.
Signature:

```
(blob ref) create_new_blob (session_id s, host ref host, string name, string mime_type)
```

Arguments:

type	name	description
`host ref`	host	The host
`string`	name	The name associated with the blob
`string`	mime_type	The mime type for the data. Empty string translates to application/octet-stream

Return Type: `blob ref`
The reference of the blob, needed for populating its data

RPC name: call_plugin

Overview:
Call a XenAPI plugin on this host.
Signature:

```
string call_plugin (session_id s, host ref host, string plugin, string fn, (string -> string) Map args
```

Arguments:

type	name	description
`host ref`	host	The host
`string`	plugin	The name of the plugin
`string`	fn	The name of the function within the plugin
`(string → string) Map`	args	Arguments for the function

Return Type: `string`
Result from the plugin

RPC name: get_servertime

Overview:
This call queries the host's clock for the current time.
Signature:

```
datetime get_servertime (session_id s, host ref host)
```

Arguments:

type	name	description
`host ref`	host	The host whose clock should be queried

Return Type: `datetime`
The current time

RPC name: get_server_localtime

Overview:
This call queries the host's clock for the current time in the host's local timezone.
Signature:

```
datetime get_server_localtime (session_id s, host ref host)
```

Arguments:

type	name	description
host ref	host	The host whose clock should be queried

Return Type: datetime
The current local time

RPC name: enable_external_auth

Overview:
This call enables external authentication on a host.
Signature:

```
void enable_external_auth (session_id s, host ref host, (string -> string) Map config, string service_
```

Arguments:

type	name	description
host ref	host	The host whose external authentication should be enabled
(string → string) Map	config	A list of key-values containing the configuration data
string	service_name	The name of the service
string	auth_type	The type of authentication (e.g. AD for Active Directory)

Return Type: void

RPC name: disable_external_auth

Overview:
This call disables external authentication on the local host.
Signature:

```
void disable_external_auth (session_id s, host ref host, (string -> string) Map config)
```

Arguments:

type	name	description
host ref	host	The host whose external authentication should be disabled
(string → string) Map	config	Optional parameters as a list of key-values containing the configuration data

Return Type: void

RPC name: retrieve_wlb_evacuate_recommendations

Overview:
Retrieves recommended host migrations to perform when evacuating the host from the wlb server. If a VM cannot be migrated from the host the reason is listed instead of a recommendation.
Signature:

```
((VM ref -> string Set) Map) retrieve_wlb_evacuate_recommendations (session_id s, host ref self)
```

Arguments:

type	name	description
host ref	self	The host to query

Return Type: `(VM ref → string Set) Map`
VMs and the reasons why they would block evacuation, or their target host recommended by the wlb server

RPC name: get_server_certificate

Overview:
Get the installed server SSL certificate.
Signature:

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

Arguments:

type	name	description
host ref	host	The host

Return Type: `string`
The installed server SSL certificate, in PEM form.

RPC name: apply_edition

Overview:
Change to another edition, or reactivate the current edition after a license has expired. This may be subject to the successful checkout of an appropriate license.
Signature:

```
void apply_edition (session_id s, host ref host, string edition)
```

Arguments:

type	name	description
host ref	host	The host
string	edition	The requested edition

Return Type: `void`

RPC name: refresh_pack_info

Overview:
Refresh the list of installed Supplemental Packs.
Signature:

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

Arguments:

type	name	description
host ref	host	The Host to modify

Return Type: void

RPC name: set_power_on_mode

Overview:
Set the power-on-mode, host, user and password .
Signature:

```
void set_power_on_mode (session_id s, host ref self, string power_on_mode, (string -> string) Map powe
```

Arguments:

type	name	description
host ref	self	The host
string	power_on_mode	power-on-mode can be empty,iLO,wake-on-lan, DRAC or other
(string → string) Map	power_on_config	Power on config

Return Type: void

RPC name: set_cpu_features

Overview:
Set the CPU features to be used after a reboot, if the given features string is valid.
Signature:

```
void set_cpu_features (session_id s, host ref host, string features)
```

Arguments:

type	name	description
host ref	host	The host
string	features	The features string (32 hexadecimal digits)

Return Type: void

RPC name: reset_cpu_features

Overview:
Remove the feature mask, such that after a reboot all features of the CPU are enabled.
Signature:

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

Arguments:

type	name	description
host ref	host	The host

Return Type: void

RPC name: enable_local_storage_caching

Overview:
Enable the use of a local SR for caching purposes.
Signature:

```
void enable_local_storage_caching (session_id s, host ref host, SR ref sr)
```

Arguments:

type	name	description
host ref	host	The host
SR ref	sr	The SR to use as a local cache

Return Type: void

RPC name: disable_local_storage_caching

Overview:
Disable the use of a local SR for caching purposes.
Signature:

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

Arguments:

type	name	description
host ref	host	The host

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

Overview:
Return a map of host references to host records for all hosts known to the system.
Signature:

```
((host ref -> host record) Map) get_all_records (session_id s)
```

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

Overview:
Get the memory/overhead field of the given host.
Signature:

```
int get_memory_overhead (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_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_cpu_info

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

```
((string -> string) Map) get_cpu_info (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_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_ha_statefiles

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

```
(string Set) get_ha_statefiles (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_ha_network_peers

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

```
(string Set) get_ha_network_peers (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_blobs

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

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

Arguments:

type	name	description
host ref	self	reference to the object

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

RPC name: get_tags

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

```
(string Set) get_tags (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: set_tags

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

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

Arguments:

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

Return Type: void

RPC name: add_tags

Overview:
Add the given value to the tags field of the given host. If the value is already in that Set, then do nothing.
Signature:

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

Arguments:

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

Return Type: void

RPC name: remove_tags

Overview:
Remove the given value from the tags field of the given host. If the value is not in that Set, then do nothing.
Signature:

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

Arguments:

type	name	description
host ref	self	reference to the object
string	value	Value to remove

Return Type: void

RPC name: get_external_auth_type

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

```
string get_external_auth_type (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_external_auth_service_name

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

```
string get_external_auth_service_name (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_external_auth_configuration

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

```
((string -> string) Map) get_external_auth_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_edition

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

```
string get_edition (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_license_server

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

```
((string -> string) Map) get_license_server (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_license_server

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

```
void set_license_server (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_license_server

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

```
void add_to_license_server (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_license_server

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

```
void remove_from_license_server (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_bios_strings

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

```
((string -> string) Map) get_bios_strings (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_power_on_mode

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

```
string get_power_on_mode (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_power_on_config

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

```
((string -> string) Map) get_power_on_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: get_local_cache_sr

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

```
(SR ref) get_local_cache_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: 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

2.17 Class: host_crashdump

2.17.1 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
RO_{ins}	`filename`	string	filename of crash dir
RW	`other_config`	(string → string) Map	additional configuration

2.17.2 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

Overview:
Return a map of host_crashdump references to host_crashdump records for all host_crashdumps known to the system.
Signature:

```
((host_crashdump ref -> host_crashdump record) Map) get_all_records (session_id s)
```

Return Type: `(host_crashdump ref → host_crashdump record) Map`
records of all 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_other_config

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

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

Arguments:

type	name	description
host_crashdump 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_crashdump.
Signature:

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

Arguments:

type	name	description
host_crashdump 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_crashdump.
Signature:

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

Arguments:

type	name	description
host_crashdump 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_crashdump.
If the key is not in that Map, then do nothing.
Signature:

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

Arguments:

type	name	description
host_crashdump 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 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

2.18 Class: host_patch

2.18.1 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}	`filename`	string	Filename of the patch
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
RO_{ins}	`pool_patch`	pool_patch ref	The patch applied
RW	`other_config`	(string → string) Map	additional configuration

2.18.2 RPCs associated with class: host_patch

RPC name: destroy

Overview: This message is deprecated 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: This message is deprecated 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

Overview:
Return a map of host_patch references to host_patch records for all host_patchs known to the system.
Signature:

```
((host_patch ref -> host_patch record) Map) get_all_records (session_id s)
```

Return Type: `(host_patch ref → host_patch record) Map`
records of all 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_pool_patch

Overview:
Get the pool_patch field of the given host_patch.
Signature:

```
(pool_patch ref) get_pool_patch (session_id s, host_patch ref self)
```

Arguments:

type	name	description
host_patch ref	self	reference to the object

Return Type: `pool_patch ref`
value of the field

RPC name: get_other_config

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

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

Arguments:

type	name	description
host_patch 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_patch.
Signature:

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

Arguments:

type	name	description
host_patch 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_patch.
Signature:

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

Arguments:

type	name	description
host_patch 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_patch.
If the key is not in that Map, then do nothing.
Signature:

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

Arguments:

type	name	description
host_patch 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 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

2.19 Class: host_metrics

2.19.1 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	Total host memory (bytes)
RO_{run}	`memory/free`	int	Free host 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
RW	`other_config`	(string → string) Map	additional configuration

2.19.2 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

Overview:
Return a map of host_metrics references to host_metrics records for all host_metrics instances known to the system.
Signature:

```
((host_metrics ref -> host_metrics record) Map) get_all_records (session_id s)
```

Return Type: `(host_metrics ref → host_metrics record) Map`
records of all 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: This message is deprecated 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_other_config

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

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

Arguments:

type	name	description
host_metrics 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_metrics.
Signature:

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

Arguments:

type	name	description
host_metrics 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_metrics.
Signature:

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

Arguments:

type	name	description
host_metrics 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_metrics.
If the key is not in that Map, then do nothing.
Signature:

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

Arguments:

type	name	description
host_metrics 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 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

2.20 Class: host_cpu

2.20.1 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
RW	`other_config`	(string → string) Map	additional configuration

2.20.2 RPCs associated with class: host_cpu

RPC name: get_all

Overview: This message is deprecated 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

Overview:
Return a map of host_cpu references to host_cpu records for all host_cpus known to the system.
Signature:

```
((host_cpu ref -> host_cpu record) Map) get_all_records (session_id s)
```

Return Type: `(host_cpu ref` → `host_cpu record) Map`
records of all 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_other_config

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

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

Arguments:

type	name	description
host_cpu 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_cpu.
Signature:

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

Arguments:

type	name	description
host_cpu 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_cpu.
Signature:

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

Arguments:

type	name	description
host_cpu 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_cpu. If the key is not in that Map, then do nothing.
Signature:

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

Arguments:

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

Return Type: void

RPC name: get_by_uuid

Overview: This message is deprecated 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: This message is deprecated 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

2.21 Class: network

2.21.1 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. This list is advisory only and the server state may have changed by the time this field is read by a client.
RO_{run}	`current_operations`	(string → network_operations) Map	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
RO_{run}	`VIFs`	(VIF ref) Set	list of connected vifs
RO_{run}	`PIFs`	(PIF ref) Set	list of connected pifs
RW	`MTU`	int	MTU in octets
RW	`other_config`	(string → string) Map	additional configuration
RO_{run}	`bridge`	string	name of the bridge corresponding to this network on the local host
RO_{run}	`blobs`	(string → blob ref) Map	Binary blobs associated with this network
RW	`tags`	string Set	user-specified tags for categorization purposes

2.21.2 RPCs associated with class: network

RPC name: create_new_blob

Overview:
Create a placeholder for a named binary blob of data that is associated with this pool.
Signature:

```
(blob ref) create_new_blob (session_id s, network ref network, string name, string mime_type)
```

Arguments:

type	name	description
`network ref`	network	The network
`string`	name	The name associated with the blob
`string`	mime_type	The mime type for the data. Empty string translates to application/octet-stream

Return Type: `blob ref`
The reference of the blob, needed for populating its data

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

Overview:
Return a map of network references to network records for all networks known to the system.
Signature:

```
((network ref -> network record) Map) get_all_records (session_id s)
```

Return Type: `(network ref → network record) Map`
records of all 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_MTU

Overview:
Get the MTU field of the given network.
Signature:

```
int get_MTU (session_id s, network ref self)
```

Arguments:

type	name	description
network 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 network.
Signature:

```
void set_MTU (session_id s, network ref self, int value)
```

Arguments:

type	name	description
network 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 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: get_blobs

Overview:
Get the blobs field of the given network.
Signature:

```
((string -> blob ref) Map) get_blobs (session_id s, network ref self)
```

Arguments:

type	name	description
network ref	self	reference to the object

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

RPC name: get_tags

Overview:
Get the tags field of the given network.
Signature:

```
(string Set) get_tags (session_id s, network ref self)
```

Arguments:

type	name	description
network ref	self	reference to the object

Return Type: string Set
value of the field

RPC name: set_tags

Overview:
Set the tags field of the given network.
Signature:

```
void set_tags (session_id s, network ref self, string Set value)
```

Arguments:

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

Return Type: void

RPC name: add_tags

Overview:
Add the given value to the tags field of the given network. If the value is already in that Set, then do nothing.
Signature:

```
void add_tags (session_id s, network ref self, string value)
```

Arguments:

type	name	description
network ref	self	reference to the object
string	value	New value to add

Return Type: void

RPC name: remove_tags

Overview:
Remove the given value from the tags field of the given network. If the value is not in that Set, then do nothing.
Signature:

```
void remove_tags (session_id s, network ref self, string value)
```

Arguments:

type	name	description
network ref	self	reference to the object
string	value	Value to remove

Return Type: void

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

2.22 Class: VIF

2.22.1 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. This list is advisory only and the server state may have changed by the time this field is read by a client.
RO_{run}	`current_operations`	(string → vif_operations) Map	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
RO_{ins}	`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
RO_{ins}	`MAC`	string	ethernet MAC address of virtual interface, as exposed to guest
RO_{ins}	`MTU`	int	MTU in octets
RO_{run}	`reserved`	bool	true if the VIF is reserved pending a reboot/migrate
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
RO_{run}	`MAC_autogenerated`	bool	true if the MAC was autogenerated; false indicates it was set manually

2.22.2 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

Overview:
Return a map of VIF references to VIF records for all VIFs known to the system.
Signature:

```
((VIF ref -> VIF record) Map) get_all_records (session_id s)
```

Return Type: `(VIF ref → VIF record) Map`
records of all 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: 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: 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: 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: get_MAC_autogenerated

Overview:
Get the MAC_autogenerated field of the given VIF.
Signature:

```
bool get_MAC_autogenerated (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: 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

2.23 Class: VIF_metrics

2.23.1 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
RW	`other_config`	(string → string) Map	additional configuration

2.23.2 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

Overview:
Return a map of VIF_metrics references to VIF_metrics records for all VIF_metrics instances known to the system.
Signature:

```
((VIF_metrics ref -> VIF_metrics record) Map) get_all_records (session_id s)
```

Return Type: `(VIF_metrics ref` → `VIF_metrics record) Map`
records of all 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_other_config

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

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

Arguments:

type	name	description
VIF_metrics 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_metrics.
Signature:

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

Arguments:

type	name	description
VIF_metrics 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_metrics.
Signature:

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

Arguments:

type	name	description
VIF_metrics 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_metrics. If the key is not in that Map, then do nothing.
Signature:

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

Arguments:

type	name	description
VIF_metrics 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 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

2.24 Class: PIF

2.24.1 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
RO_{ins}	`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
RO_{ins}	`MAC`	string	ethernet MAC address of physical interface
RO_{ins}	`MTU`	int	MTU in octets
RO_{ins}	`VLAN`	int	VLAN tag for all traffic passing through this interface
RW	`device_name`	string	actual dom0 device name
RO_{run}	`metrics`	PIF_metrics ref	metrics associated with this PIF
RO_{run}	`physical`	bool	true if this represents a physical network interface
RO_{run}	`currently_attached`	bool	true if this interface is online
RO_{run}	`ip_configuration_mode`	ip_configuration_mode	Sets if and how this interface gets an IP address
RO_{run}	`IP`	string	IP address
RO_{run}	`netmask`	string	IP netmask
RO_{run}	`gateway`	string	IP gateway
RO_{run}	`DNS`	string	IP address of DNS servers to use
RO_{run}	`bond_slave_of`	Bond ref	Indicates which bond this interface is part of
RO_{run}	`bond_master_of`	(Bond ref) Set	Indicates this PIF represents the results of a bond
RO_{run}	`VLAN_master_of`	VLAN ref	Indicates wich VLAN this interface receives untagged traffic from
RO_{run}	`VLAN_slave_of`	(VLAN ref) Set	Indicates which VLANs this interface transmits tagged traffic to
RO_{run}	`management`	bool	Indicates whether the control software is listening for connections on this interface
RW	`other_config`	(string → string) Map	Additional configuration
RW	`disallow_unplug`	bool	Prevent this PIF from being unplugged; set this to notify the management tool-stack that the PIF has a special use and should not be unplugged under any circumstances (e.g. because you're running storage traffic over it)
RO_{run}	`tunnel_access_PIF_of`	(tunnel ref) Set	Indicates to which tunnel this PIF gives access

RO_{run}	tunnel_transport_PIF_of	(tunnel ref) Set	Indicates to which tunnel this PIF provides transport

2.24.2 RPCs associated with class: PIF

RPC name: create_VLAN

Overview: This message is deprecated Create a VLAN interface from an existing physical interface. This call is deprecated: use VLAN.create instead.
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 create 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: This message is deprecated Destroy the PIF object (provided it is a VLAN interface). This call is deprecated: use VLAN.destroy or Bond.destroy instead.
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: reconfigure_ip

Overview:
Reconfigure the IP address settings for this interface.
Signature:

```
void reconfigure_ip (session_id s, PIF ref self, ip_configuration_mode mode, string IP, string netmask
```

Arguments:

type	name	description
PIF ref	self	the PIF object to reconfigure
ip_configuration_mode	mode	whether to use dynamic/static/no-assignment
string	IP	the new IP address
string	netmask	the new netmask
string	gateway	the new gateway
string	DNS	the new DNS settings

Return Type: void

RPC name: scan

Overview:
Scan for physical interfaces on a host and create PIF objects to represent them.
Signature:

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

Arguments:

type	name	description
host ref	host	The host on which to scan

Return Type: void

RPC name: scan_bios

Overview:
Scan for physical interfaces on a host and create PIF objects to represent them. Use BIOS-based device names.
Signature:

```
((PIF ref) Set) scan_bios (session_id s, host ref host)
```

Arguments:

type	name	description
host ref	host	The host on which to scan

Return Type: (PIF ref) Set
List of newly created PIFs

RPC name: introduce

Overview:
Create a PIF object matching a particular network interface.
Signature:

```
(PIF ref) introduce (session_id s, host ref host, string MAC, string device)
```

Arguments:

type	name	description
host ref	host	The host on which the interface exists
string	MAC	The MAC address of the interface
string	device	The device name to use for the interface

Return Type: PIF ref
The reference of the created PIF object

RPC name: forget

Overview:
Destroy the PIF object matching a particular network interface.
Signature:

```
void forget (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_TUNNEL_STILL_EXISTS

RPC name: unplug

Overview:
Attempt to bring down a physical interface.
Signature:

```
void unplug (session_id s, PIF ref self)
```

Arguments:

type	name	description
PIF ref	self	the PIF object to unplug

Return Type: void

RPC name: plug

Overview:
Attempt to bring up a physical interface.
Signature:

```
void plug (session_id s, PIF ref self)
```

Arguments:

type	name	description
PIF ref	self	the PIF object to plug

Return Type: `void`

Possible Error Codes: `TRANSPORT_PIF_NOT_CONFIGURED`

RPC name: db_introduce

Overview:
Create a new PIF record in the database only.
Signature:

```
(PIF ref) db_introduce (session_id s, string device, network ref network, host ref host, string MAC, i
```

Arguments:

type	name	description
string	device	
network ref	network	
host ref	host	
string	MAC	
int	MTU	
int	VLAN	
bool	physical	
ip_configuration_mode	ip_configuration_mode	
string	IP	
string	netmask	
string	gateway	
string	DNS	
Bond ref	bond_slave_of	
VLAN ref	VLAN_master_of	
bool	management	
(string → string) Map	other_config	
bool	disallow_unplug	

Return Type: `PIF ref`
The ref of the newly created PIF record.

RPC name: db_forget

Overview:
Destroy a PIF database record.
Signature:

```
void db_forget (session_id s, PIF ref self)
```

Arguments:

type	name	description
PIF ref	self	The ref of the PIF whose database record should be destroyed

Return Type: `void`

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

Overview:
Return a map of PIF references to PIF records for all PIFs known to the system.
Signature:

```
((PIF ref -> PIF record) Map) get_all_records (session_id s)
```

Return Type: `(PIF ref → PIF record) Map`
records of all 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: 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: 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: 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: 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_physical

Overview:
Get the physical field of the given PIF.
Signature:

```
bool get_physical (session_id s, PIF ref self)
```

Arguments:

type	name	description
PIF ref	self	reference to the object

Return Type: bool
value of the field

RPC name: get_currently_attached

Overview:
Get the currently_attached field of the given PIF.
Signature:

```
bool get_currently_attached (session_id s, PIF ref self)
```

Arguments:

type	name	description
PIF ref	self	reference to the object

Return Type: bool
value of the field

RPC name: get_ip_configuration_mode

Overview:
Get the ip_configuration_mode field of the given PIF.
Signature:

```
(ip_configuration_mode) get_ip_configuration_mode (session_id s, PIF ref self)
```

Arguments:

type	name	description
PIF ref	self	reference to the object

Return Type: ip_configuration_mode
value of the field

RPC name: get_IP

Overview:
Get the IP field of the given PIF.
Signature:

```
string get_IP (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_netmask

Overview:
Get the netmask field of the given PIF.
Signature:

```
string get_netmask (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_gateway

Overview:
Get the gateway field of the given PIF.
Signature:

```
string get_gateway (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_DNS

Overview:
Get the DNS field of the given PIF.
Signature:

```
string get_DNS (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_bond_slave_of

Overview:
Get the bond_slave_of field of the given PIF.
Signature:

```
(Bond ref) get_bond_slave_of (session_id s, PIF ref self)
```

Arguments:

type	name	description
PIF ref	self	reference to the object

Return Type: `Bond ref`
value of the field

RPC name: get_bond_master_of

Overview:
Get the bond_master_of field of the given PIF.
Signature:

```
((Bond ref) Set) get_bond_master_of (session_id s, PIF ref self)
```

Arguments:

type	name	description
PIF ref	self	reference to the object

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

RPC name: get_VLAN_master_of

Overview:
Get the VLAN_master_of field of the given PIF.
Signature:

```
(VLAN ref) get_VLAN_master_of (session_id s, PIF ref self)
```

Arguments:

type	name	description
PIF ref	self	reference to the object

Return Type: `VLAN ref`
value of the field

RPC name: get_VLAN_slave_of

Overview:
Get the VLAN_slave_of field of the given PIF.
Signature:

```
((VLAN ref) Set) get_VLAN_slave_of (session_id s, PIF ref self)
```

Arguments:

type	name	description
PIF ref	self	reference to the object

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

RPC name: get_management

Overview:
Get the management field of the given PIF.
Signature:

```
bool get_management (session_id s, PIF ref self)
```

Arguments:

type	name	description
PIF 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 PIF.
Signature:

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

Arguments:

type	name	description
PIF 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 PIF.
Signature:

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

Arguments:

type	name	description
PIF 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 PIF.
Signature:

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

Arguments:

type	name	description
PIF 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 PIF. If the key is not in that Map, then do nothing.
Signature:

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

Arguments:

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

Return Type: `void`

RPC name: get_disallow_unplug

Overview:
Get the disallow_unplug field of the given PIF.
Signature:

```
bool get_disallow_unplug (session_id s, PIF ref self)
```

Arguments:

type	name	description
PIF ref	self	reference to the object

Return Type: `bool`
value of the field

RPC name: set_disallow_unplug

Overview:
Set the disallow_unplug field of the given PIF.
Signature:

```
void set_disallow_unplug (session_id s, PIF ref self, bool value)
```

Arguments:

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

Return Type: `void`

RPC name: get_tunnel_access_PIF_of

Overview:
Get the tunnel_access_PIF_of field of the given PIF.
Signature:

```
((tunnel ref) Set) get_tunnel_access_PIF_of (session_id s, PIF ref self)
```

Arguments:

type	name	description
PIF ref	self	reference to the object

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

RPC name: get_tunnel_transport_PIF_of

Overview:
Get the tunnel_transport_PIF_of field of the given PIF.
Signature:

```
((tunnel ref) Set) get_tunnel_transport_PIF_of (session_id s, PIF ref self)
```

Arguments:

type	name	description
PIF ref	self	reference to the object

Return Type: `(tunnel ref) Set`
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

2.25 Class: PIF_metrics

2.25.1 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
RW	`other_config`	(string → string) Map	additional configuration

2.25.2 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

Overview:
Return a map of PIF_metrics references to PIF_metrics records for all PIF_metrics instances known to the system.
Signature:

```
((PIF_metrics ref -> PIF_metrics record) Map) get_all_records (session_id s)
```

Return Type: `(PIF_metrics ref → PIF_metrics record) Map`
records of all 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_other_config

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

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

Arguments:

type	name	description
PIF_metrics 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 PIF_metrics.
Signature:

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

Arguments:

type	name	description
PIF_metrics 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 PIF_metrics.
Signature:

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

Arguments:

type	name	description
PIF_metrics 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 PIF_metrics. If the key is not in that Map, then do nothing.
Signature:

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

Arguments:

type	name	description
PIF_metrics 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 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

2.26 Class: Bond

2.26.1 Fields for class: Bond

Name	**Bond**		
Description	·		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	Unique identifier/object reference
RO_{ins}	`master`	PIF ref	The bonded interface
RO_{run}	`slaves`	(PIF ref) Set	The interfaces which are part of this bond
RW	`other_config`	(string → string) Map	additional configuration

2.26.2 RPCs associated with class: Bond

RPC name: create

Overview:
Create an interface bond.
Signature:

```
(Bond ref) create (session_id s, network ref network, (PIF ref) Set members, string MAC)
```

Arguments:

type	name	description
`network ref`	network	Network to add the bonded PIF to
`(PIF ref) Set`	members	PIFs to add to this bond
`string`	MAC	The MAC address to use on the bond itself. If this parameter is the empty string then the bond will inherit its MAC address from the first of the specified 'members'

Return Type: `Bond ref`
The reference of the created Bond object

RPC name: destroy

Overview:
Destroy an interface bond.
Signature:

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

Arguments:

type	name	description
`Bond ref`	self	Bond to destroy

Return Type: `void`

RPC name: get_all

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

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

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

RPC name: get_all_records

Overview:
Return a map of Bond references to Bond records for all Bonds known to the system.
Signature:

```
((Bond ref -> Bond record) Map) get_all_records (session_id s)
```

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

RPC name: get_uuid

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

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

Arguments:

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

Return Type: `string`
value of the field

RPC name: get_master

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

```
(PIF ref) get_master (session_id s, Bond ref self)
```

Arguments:

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

Return Type: `PIF ref`

value of the field

RPC name: get_slaves

Overview:
Get the slaves field of the given Bond.
Signature:

```
((PIF ref) Set) get_slaves (session_id s, Bond ref self)
```

Arguments:

type	name	description
`Bond 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 Bond.
Signature:

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

Arguments:

type	name	description
`Bond 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 Bond.
Signature:

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

Arguments:

type	name	description
`Bond 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 Bond.
Signature:

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

Arguments:

type	name	description
Bond 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 Bond. If the key is not in that Map, then do nothing.
Signature:

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

Arguments:

type	name	description
Bond 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 Bond instance with the specified UUID.
Signature:

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

Arguments:

type	name	description
string	uuid	UUID of object to return

Return Type: `Bond ref`
reference to the object

RPC name: get_record

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

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

Arguments:

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

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

2.27 Class: VLAN

2.27.1 Fields for class: VLAN

Name	**VLAN**		
Description	*A VLAN mux/demux.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	Unique identifier/object reference
RO_{ins}	`tagged_PIF`	PIF ref	interface on which traffic is tagged
RO_{run}	`untagged_PIF`	PIF ref	interface on which traffic is untagged
RO_{ins}	`tag`	int	VLAN tag in use
RW	`other_config`	(string → string) Map	additional configuration

2.27.2 RPCs associated with class: VLAN

RPC name: create

Overview:
Create a VLAN mux/demuxer.
Signature:

```
(VLAN ref) create (session_id s, PIF ref tagged_PIF, int tag, network ref network)
```

Arguments:

type	name	description
`PIF ref`	tagged_PIF	PIF which receives the tagged traffic
`int`	tag	VLAN tag to use
`network ref`	network	Network to receive the untagged traffic

Return Type: `VLAN ref`
The reference of the created VLAN object

RPC name: destroy

Overview:
Destroy a VLAN mux/demuxer.
Signature:

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

Arguments:

type	name	description
`VLAN ref`	self	VLAN mux/demuxer to destroy

Return Type: `void`

RPC name: get_all

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

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

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

RPC name: get_all_records

Overview:
Return a map of VLAN references to VLAN records for all VLANs known to the system.
Signature:

```
((VLAN ref -> VLAN record) Map) get_all_records (session_id s)
```

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

RPC name: get_uuid

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

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

Arguments:

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

Return Type: `string`
value of the field

RPC name: get_tagged_PIF

Overview:
Get the tagged_PIF field of the given VLAN.
Signature:

```
(PIF ref) get_tagged_PIF (session_id s, VLAN ref self)
```

Arguments:

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

Return Type: `PIF ref`
value of the field

RPC name: get_untagged_PIF

Overview:
Get the untagged_PIF field of the given VLAN.
Signature:

```
(PIF ref) get_untagged_PIF (session_id s, VLAN ref self)
```

Arguments:

type	name	description
VLAN ref	self	reference to the object

Return Type: `PIF ref`
value of the field

RPC name: get_tag

Overview:
Get the tag field of the given VLAN.
Signature:

```
int get_tag (session_id s, VLAN ref self)
```

Arguments:

type	name	description
VLAN ref	self	reference to the object

Return Type: `int`
value of the field

RPC name: get_other_config

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

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

Arguments:

type	name	description
VLAN 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 VLAN.
Signature:

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

Arguments:

type	name	description
`VLAN 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 VLAN.
Signature:

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

Arguments:

type	name	description
`VLAN 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 VLAN. If the key is not in that Map, then do nothing.
Signature:

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

Arguments:

type	name	description
`VLAN 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 VLAN instance with the specified UUID.
Signature:

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

Arguments:

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

Return Type: `VLAN ref`
reference to the object

RPC name: get_record

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

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

Arguments:

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

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

2.28 Class: SM

2.28.1 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
RO_{run}	`capabilities`	string Set	capabilities of the SM plugin
RW	`other_config`	(string → string) Map	additional configuration
RO_{run}	`driver_filename`	string	filename of the storage driver

2.28.2 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

Overview:
Return a map of SM references to SM records for all SMs known to the system.
Signature:

```
((SM ref -> SM record) Map) get_all_records (session_id s)
```

Return Type: `(SM ref → SM record) Map`
records of all 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_capabilities

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

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

Arguments:

type	name	description
SM ref	self	reference to the object

Return Type: `string Set`
value of the field

RPC name: get_other_config

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

```
((string -> string) Map) get_other_config (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: set_other_config

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

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

Arguments:

type	name	description
SM 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 SM.
Signature:

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

Arguments:

type	name	description
SM 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 SM. If the key is not in that Map, then do nothing.
Signature:

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

Arguments:

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

Return Type: void

RPC name: get_driver_filename

Overview:
Get the driver_filename field of the given SM.
Signature:

```
string get_driver_filename (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_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

2.29 Class: SR

2.29.1 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. This list is advisory only and the server state may have changed by the time this field is read by a client.
RO_{run}	`current_operations`	(string → storage_operations) Map	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
RO_{run}	`VDIs`	(VDI ref) Set	all virtual disks known to this storage repository
RO_{run}	`PBDs`	(PBD ref) Set	describes how particular hosts can see this storage repository
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)
RO_{run}	`shared`	bool	true if this SR is (capable of being) shared between multiple hosts
RW	`other_config`	(string → string) Map	additional configuration
RW	`tags`	string Set	user-specified tags for categorization purposes
RO_{run}	`default_vdi_visibility`	bool	
RW	`sm_config`	(string → string) Map	SM dependent data
RO_{run}	`blobs`	(string → blob ref) Map	Binary blobs associated with this SR
RO_{run}	`local_cache_enabled`	bool	True if this SR is assigned to be the local cache for its host

2.29.2 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
(string → string) Map	sm_config	Storage backend specific configuration options

Return Type: SR ref
The reference of the newly created Storage Repository.

Possible Error Codes: SR_UNKNOWN_DRIVER

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
(string → string) Map	sm_config	Storage backend specific configuration options

Return Type: SR ref
The reference of the newly introduced Storage Repository.

RPC name: make

Overview: This message is deprecated Create a new Storage Repository on disk. This call is deprecated: use SR.create instead.
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)
(string → string) Map	sm_config	Storage backend specific configuration options

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

Possible Error Codes: SR_HAS_PBD

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: update

Overview:
Refresh the fields on the SR object.
Signature:

```
void update (session_id s, SR ref sr)
```

Arguments:

type	name	description
SR ref	sr	The SR whose fields should be refreshed

Return Type: `void`

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 list of VDIs associated with 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: probe

Overview:
Perform a backend-specific scan, using the given device_config. If the device_config is complete, then this will return a list of the SRs present of this type on the device, if any. If the device_config is partial, then a backend-specific scan will be performed, returning results that will guide the user in improving the device_config.
Signature:

```
string probe (session_id s, host ref host, (string -> string) Map device_config, string type, (string
```

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
`string`	type	The type of the SR; used to specify the SR backend driver to use
`(string → string) Map`	sm_config	Storage backend specific configuration options

Return Type: `string`
An XML fragment containing the scan results. These are specific to the scan being performed, and the backend.

RPC name: set_shared

Overview:
Sets the shared flag on the SR.
Signature:

```
void set_shared (session_id s, SR ref sr, bool value)
```

Arguments:

type	name	description
`SR ref`	sr	The SR
`bool`	value	True if the SR is shared

Return Type: `void`

RPC name: create_new_blob

Overview:
Create a placeholder for a named binary blob of data that is associated with this SR.
Signature:

```
(blob ref) create_new_blob (session_id s, SR ref sr, string name, string mime_type)
```

Arguments:

type	name	description
`SR ref`	sr	The SR
`string`	name	The name associated with the blob
`string`	mime_type	The mime type for the data. Empty string translates to application/octet-stream

Return Type: `blob ref`
The reference of the blob, needed for populating its data

RPC name: set_physical_size

Overview:
Sets the SR's physical_size field.
Signature:

```
void set_physical_size (session_id s, SR ref self, int value)
```

Arguments:

type	name	description
SR ref	self	The SR to modify
int	value	The new value of the SR's physical_size

Return Type: void

RPC name: set_virtual_allocation

Overview:
Sets the SR's virtual_allocation field.
Signature:

```
void set_virtual_allocation (session_id s, SR ref self, int value)
```

Arguments:

type	name	description
SR ref	self	The SR to modify
int	value	The new value of the SR's virtual_allocation

Return Type: void

RPC name: set_physical_utilisation

Overview:
Sets the SR's physical_utilisation field.
Signature:

```
void set_physical_utilisation (session_id s, SR ref self, int value)
```

Arguments:

type	name	description
SR ref	self	The SR to modify
int	value	The new value of the SR's physical utilisation

Return Type: void

RPC name: assert_can_host_ha_statefile

Overview:
Returns successfully if the given SR can host an HA statefile. Otherwise returns an error to explain why not.
Signature:

```
void assert_can_host_ha_statefile (session_id s, SR ref sr)
```

Arguments:

type	name	description
SR ref	sr	The SR to query

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

Overview:
Return a map of SR references to SR records for all SRs known to the system.
Signature:

```
((SR ref -> SR record) Map) get_all_records (session_id s)
```

Return Type: `(SR ref → SR record) Map`
records of all 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: 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_tags

Overview:
Get the tags field of the given SR.
Signature:

```
(string Set) get_tags (session_id s, SR ref self)
```

Arguments:

type	name	description
SR ref	self	reference to the object

Return Type: `string Set`
value of the field

RPC name: set_tags

Overview:
Set the tags field of the given SR.
Signature:

```
void set_tags (session_id s, SR ref self, string Set value)
```

Arguments:

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

Return Type: `void`

RPC name: add_tags

Overview:
Add the given value to the tags field of the given SR. If the value is already in that Set, then do nothing.
Signature:

```
void add_tags (session_id s, SR ref self, string value)
```

Arguments:

type	name	description
SR ref	self	reference to the object
string	value	New value to add

Return Type: `void`

RPC name: remove_tags

Overview:
Remove the given value from the tags field of the given SR. If the value is not in that Set, then do nothing.
Signature:

```
void remove_tags (session_id s, SR ref self, string value)
```

Arguments:

type	name	description
SR ref	self	reference to the object
string	value	Value to remove

Return Type: `void`

RPC name: get_sm_config

Overview:
Get the sm_config field of the given SR.
Signature:

```
((string -> string) Map) get_sm_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_sm_config

Overview:
Set the sm_config field of the given SR.
Signature:

```
void set_sm_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_sm_config

Overview:
Add the given key-value pair to the sm_config field of the given SR.
Signature:

```
void add_to_sm_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_sm_config

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

```
void remove_from_sm_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_blobs

Overview:
Get the blobs field of the given SR.
Signature:

```
((string -> blob ref) Map) get_blobs (session_id s, SR ref self)
```

Arguments:

type	name	description
SR ref	self	reference to the object

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

RPC name: get_local_cache_enabled

Overview:
Get the local_cache_enabled field of the given SR.
Signature:

```
bool get_local_cache_enabled (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: 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

2.30 Class: VDI

2.30.1 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. This list is advisory only and the server state may have changed by the time this field is read by a client.
RO_{run}	`current_operations`	(string → vdi_operations) Map	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
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
RO_{ins}	`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
RO_{ins}	`sharable`	bool	true if this disk may be shared
RO_{ins}	`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}	`location`	string	location information
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
RW	`xenstore_data`	(string → string) Map	data to be inserted into the xenstore tree (/local/domain/0/backend/vbd/¡domid¿/¡device id¿/sm-data) after the VDI is attached. This is generally set by the SM backends on vdi_attach.
RW	`sm_config`	(string → string) Map	SM dependent data
RO_{run}	`is_a_snapshot`	bool	true if this is a snapshot.
RO_{run}	`snapshot_of`	VDI ref	Ref pointing to the VDI this snapshot is of.

RO_{run}	snapshots	(VDI ref) Set	List pointing to all the VDIs snapshots.
RO_{run}	snapshot_time	datetime	Date/time when this snapshot was created.
RW	tags	string Set	user-specified tags for categorization purposes
RO_{run}	allow_caching	bool	true if this VDI is to be cached in the local cache SR
RO_{run}	on_boot	on_boot	The behaviour of this VDI on a VM boot

2.30.2 RPCs associated with class: VDI

RPC name: snapshot

Overview:
Take a read-only snapshot of the VDI, returning a reference to the snapshot. If any driver_params are specified then these are passed through to the storage-specific substrate driver that takes the snapshot. NB the snapshot lives in the same Storage Repository as its parent.
Signature:

```
(VDI ref) snapshot (session_id s, VDI ref vdi, (string -> string) Map driver_params)
```

Arguments:

type	name	description
VDI ref	vdi	The VDI to snapshot
(string → string) Map	driver_params	Optional parameters that can be passed through to backend driver in order to specify storage-type-specific snapshot options

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

RPC name: clone

Overview:
Take an exact copy of the VDI and return a reference to the new disk. If any driver_params are specified then these are passed through to the storage-specific substrate driver that implements the clone operation. NB the clone lives in the same Storage Repository as its parent.
Signature:

```
(VDI ref) clone (session_id s, VDI ref vdi, (string -> string) Map driver_params)
```

Arguments:

type	name	description
VDI ref	vdi	The VDI to clone
(string → string) Map	driver_params	Optional parameters that are passed through to the backend driver in order to specify storage-type-specific clone options

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

RPC name: resize

Overview:
Resize the VDI.
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: resize_online

Overview:
Resize the VDI which may or may not be attached to running guests.
Signature:

```
void resize_online (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
(string → string) Map	xenstore_data	Data to insert into xenstore
(string → string) Map	sm_config	Storage-specific config

Return Type: `VDI ref`
The ref of the newly created VDI record.

Possible Error Codes: `SR_OPERATION_NOT_SUPPORTED`

RPC name: db_introduce

Overview:
Create a new VDI record in the database only.
Signature:

```
(VDI ref) db_introduce (session_id s, string uuid, string name_label, string name_description, SR ref S
```

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
`(string → string) Map`	xenstore_data	Data to insert into xenstore
`(string → string) Map`	sm_config	Storage-specific config

Return Type: `VDI ref`
The ref of the newly created VDI record.

RPC name: db_forget

Overview:
Removes a VDI record from the database.
Signature:

```
void db_forget (session_id s, VDI ref vdi)
```

Arguments:

type	name	description
`VDI ref`	vdi	The VDI to forget about

Return Type: `void`

RPC name: update

Overview:
Ask the storage backend to refresh the fields in the VDI object.
Signature:

```
void update (session_id s, VDI ref vdi)
```

Arguments:

type	name	description
VDI ref	vdi	The VDI whose stats (eg size) should be updated

Return Type: `void`

Possible Error Codes: `SR_OPERATION_NOT_SUPPORTED`

RPC name: copy

Overview:
Make a fresh VDI in the specified SR and copy the supplied VDI's data to the new disk.
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 reference of the newly created VDI.

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: forget

Overview:
Removes a VDI record from the database.
Signature:

```
void forget (session_id s, VDI ref vdi)
```

Arguments:

type	name	description
VDI ref	vdi	The VDI to forget about

Return Type: void

RPC name: set_sharable

Overview:
Sets the VDI's sharable field.
Signature:

```
void set_sharable (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 sharable field

Return Type: void

RPC name: set_read_only

Overview:
Sets the VDI's read_only field.
Signature:

```
void set_read_only (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 read_only field

Return Type: void

RPC name: set_missing

Overview:
Sets the VDI's missing field.
Signature:

```
void set_missing (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 missing field

Return Type: `void`

RPC name: set_virtual_size

Overview:
Sets the VDI's virtual_size field.
Signature:

```
void set_virtual_size (session_id s, VDI ref self, int value)
```

Arguments:

type	name	description
`VDI ref`	self	The VDI to modify
`int`	value	The new value of the VDI's virtual size

Return Type: `void`

RPC name: set_physical_utilisation

Overview:
Sets the VDI's physical_utilisation field.
Signature:

```
void set_physical_utilisation (session_id s, VDI ref self, int value)
```

Arguments:

type	name	description
`VDI ref`	self	The VDI to modify
`int`	value	The new value of the VDI's physical utilisation

Return Type: `void`

RPC name: set_on_boot

Overview:
Set the value of the on_boot parameter. This value can only be changed when the VDI is not attached to a running VM.
Signature:

```
void set_on_boot (session_id s, VDI ref self, on_boot value)
```

Arguments:

type	name	description
`VDI ref`	self	The VDI to modify
`on_boot`	value	The value to set

Return Type: `void`

RPC name: set_allow_caching

Overview:
Set the value of the allow_caching parameter. This value can only be changed when the VDI is not attached to a running VM. The caching behaviour is only affected by this flag for VHD-based VDIs that have one parent and no child VHDs. Moreover, caching only takes place when the host running the VM containing this VDI has a nominated SR for local caching.
Signature:

```
void set_allow_caching (session_id s, VDI ref self, bool value)
```

Arguments:

type	name	description
VDI ref	self	The VDI to modify
bool	value	The value to set

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

Overview:
Return a map of VDI references to VDI records for all VDIs known to the system.
Signature:

```
((VDI ref -> VDI record) Map) get_all_records (session_id s)
```

Return Type: `(VDI ref → VDI record) Map`
records of all 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: 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: 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: 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_location

Overview:
Get the location field of the given VDI.
Signature:

```
string get_location (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_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: get_xenstore_data

Overview:
Get the xenstore_data field of the given VDI.
Signature:

```
((string -> string) Map) get_xenstore_data (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_xenstore_data

Overview:
Set the xenstore_data field of the given VDI.
Signature:

```
void set_xenstore_data (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_xenstore_data

Overview:
Add the given key-value pair to the xenstore_data field of the given VDI.
Signature:

```
void add_to_xenstore_data (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_xenstore_data

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

```
void remove_from_xenstore_data (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_sm_config

Overview:
Get the sm_config field of the given VDI.
Signature:

```
((string -> string) Map) get_sm_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_sm_config

Overview:
Set the sm_config field of the given VDI.
Signature:

```
void set_sm_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_sm_config

Overview:
Add the given key-value pair to the sm_config field of the given VDI.
Signature:

```
void add_to_sm_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_sm_config

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

```
void remove_from_sm_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_is_a_snapshot

Overview:
Get the is_a_snapshot field of the given VDI.
Signature:

```
bool get_is_a_snapshot (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_snapshot_of

Overview:
Get the snapshot_of field of the given VDI.
Signature:

```
(VDI ref) get_snapshot_of (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: get_snapshots

Overview:
Get the snapshots field of the given VDI.
Signature:

```
((VDI ref) Set) get_snapshots (session_id s, VDI ref self)
```

Arguments:

type	name	description
VDI ref	self	reference to the object

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

RPC name: get_snapshot_time

Overview:
Get the snapshot_time field of the given VDI.
Signature:

```
datetime get_snapshot_time (session_id s, VDI ref self)
```

Arguments:

type	name	description
VDI ref	self	reference to the object

Return Type: `datetime`
value of the field

RPC name: get_tags

Overview:
Get the tags field of the given VDI.
Signature:

```
(string Set) get_tags (session_id s, VDI ref self)
```

Arguments:

type	name	description
VDI ref	self	reference to the object

Return Type: `string Set`
value of the field

RPC name: set_tags

Overview:
Set the tags field of the given VDI.
Signature:

```
void set_tags (session_id s, VDI ref self, string Set value)
```

Arguments:

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

Return Type: void

RPC name: add_tags

Overview:
Add the given value to the tags field of the given VDI. If the value is already in that Set, then do nothing.
Signature:

```
void add_tags (session_id s, VDI ref self, string value)
```

Arguments:

type	name	description
VDI ref	self	reference to the object
string	value	New value to add

Return Type: void

RPC name: remove_tags

Overview:
Remove the given value from the tags field of the given VDI. If the value is not in that Set, then do nothing.
Signature:

```
void remove_tags (session_id s, VDI ref self, string value)
```

Arguments:

type	name	description
VDI ref	self	reference to the object
string	value	Value to remove

Return Type: void

RPC name: get_allow_caching

Overview:
Get the allow_caching field of the given VDI.
Signature:

```
bool get_allow_caching (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_on_boot

Overview:
Get the on_boot field of the given VDI.
Signature:

```
(on_boot) get_on_boot (session_id s, VDI ref self)
```

Arguments:

type	name	description
VDI ref	self	reference to the object

Return Type: on_boot
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

2.31 Class: VBD

2.31.1 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. This list is advisory only and the server state may have changed by the time this field is read by a client.
RO_{run}	`current_operations`	(string → vbd_operations) Map	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
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)
RW	`unpluggable`	bool	true if this VBD will support hot-unplug
RO_{run}	`storage_lock`	bool	true if a storage level lock was acquired
RO_{ins}	`empty`	bool	if true this represents an empty drive
RO_{run}	`reserved`	bool	true if the VBD is reserved pending a reboot/migrate
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

2.31.2 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

Possible Error Codes: DEVICE_DETACH_REJECTED, DEVICE_ALREADY_DETACHED

RPC name: unplug_force

Overview:
Forcibly unplug the specified VBD.
Signature:

```
void unplug_force (session_id s, VBD ref self)
```

Arguments:

type	name	description
VBD ref	self	The VBD to forcibly 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

Overview:
Return a map of VBD references to VBD records for all VBDs known to the system.
Signature:

```
((VBD ref -> VBD record) Map) get_all_records (session_id s)
```

Return Type: `(VBD ref → VBD record) Map`
records of all 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_unpluggable

Overview:
Get the unpluggable field of the given VBD.
Signature:

```
bool get_unpluggable (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_unpluggable

Overview:
Set the unpluggable field of the given VBD.
Signature:

```
void set_unpluggable (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_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

2.32 Class: VBD_metrics

2.32.1 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
RW	`other_config`	(string → string) Map	additional configuration

2.32.2 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

Overview:
Return a map of VBD_metrics references to VBD_metrics records for all VBD_metrics instances known to the system.
Signature:

```
((VBD_metrics ref -> VBD_metrics record) Map) get_all_records (session_id s)
```

Return Type: `(VBD_metrics ref` → `VBD_metrics record) Map`
records of all 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_other_config

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

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

Arguments:

type	name	description
VBD_metrics 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_metrics.
Signature:

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

Arguments:

type	name	description
VBD_metrics 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_metrics.
Signature:

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

Arguments:

type	name	description
VBD_metrics 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_metrics. If the key is not in that Map, then do nothing.
Signature:

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

Arguments:

type	name	description
VBD_metrics 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 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

2.33 Class: PBD

2.33.1 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?
RW	`other_config`	(string → string) Map	additional configuration

2.33.2 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: set_device_config

Overview:
Sets the PBD's device_config field.
Signature:

```
void set_device_config (session_id s, PBD ref self, (string -> string) Map value)
```

Arguments:

type	name	description
PBD ref	self	The PBD to modify
(string → string) Map	value	The new value of the PBD's device_config

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

Overview:
Return a map of PBD references to PBD records for all PBDs known to the system.
Signature:

```
((PBD ref -> PBD record) Map) get_all_records (session_id s)
```

Return Type: (PBD ref → PBD record) Map
records of all 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: get_other_config

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

```
((string -> string) Map) get_other_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: set_other_config

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

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

Arguments:

type	name	description
PBD 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 PBD.
Signature:

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

Arguments:

type	name	description
PBD 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 PBD. If the key is not in that Map, then do nothing.
Signature:

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

Arguments:

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

Return Type: void

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

2.34 Class: crashdump

2.34.1 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
RW	`other_config`	(string → string) Map	additional configuration

2.34.2 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

Overview:
Return a map of crashdump references to crashdump records for all crashdumps known to the system.
Signature:

```
((crashdump ref -> crashdump record) Map) get_all_records (session_id s)
```

Return Type: `(crashdump ref → crashdump record) Map`
records of all 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_other_config

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

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

Arguments:

type	name	description
`crashdump 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 crashdump.
Signature:

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

Arguments:

type	name	description
`crashdump 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 crashdump.
Signature:

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

Arguments:

type	name	description
`crashdump 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 crashdump. If the key is not in that Map, then do nothing.
Signature:

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

Arguments:

type	name	description
`crashdump 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 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

2.35 Class: VTPM

2.35.1 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

2.35.2 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

2.36 Class: console

2.36.1 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
RW	`port`	int	port in dom0 on which the console server is listening

2.36.2 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

Overview:
Return a map of console references to console records for all consoles known to the system.
Signature:

```
((console ref -> console record) Map) get_all_records (session_id s)
```

Return Type: `(console ref → console record) Map`
records of all 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

2.37 Class: user

2.37.1 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
RW	`other_config`	(string → string) Map	additional configuration

2.37.2 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: get_other_config

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

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

Arguments:

type	name	description
`user 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 user.
Signature:

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

Arguments:

type	name	description
`user 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 user.
Signature:

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

Arguments:

type	name	description
`user 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 user. If the key is not in that Map, then do nothing.
Signature:

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

Arguments:

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

Return Type: `void`

RPC name: create

Overview: This message is deprecated 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: This message is deprecated 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: This message is deprecated 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: This message is deprecated 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

2.38 Class: data_source

2.38.1 Fields for class: data_source

Name	**data_source**		
Description	*Data sources for logging in RRDs.*		
Quals	Field	Type	Description
RO_{run}	`name/label`	string	a human-readable name
RO_{run}	`name/description`	string	a notes field containg human-readable description
RO_{run}	`enabled`	bool	true if the data source is being logged
RO_{run}	`standard`	bool	true if the data source is enabled by default. Non-default data sources cannot be disabled
RO_{run}	`units`	string	the units of the value
RO_{run}	`min`	float	the minimum value of the data source
RO_{run}	`max`	float	the maximum value of the data source
RO_{run}	`value`	float	current value of the data source

2.38.2 RPCs associated with class: data_source

Class data_source has no additional RPCs associated with it.

2.39 Class: blob

2.39.1 Fields for class: blob

Name	**blob**		
Description	*A placeholder for a binary blob.*		
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}	`size`	int	Size of the binary data, in bytes
RO_{ins}	`last_updated`	datetime	Time at which the data in the blob was last updated
RO_{ins}	`mime_type`	string	The mime type associated with this object. Defaults to 'application/octet-stream' if the empty string is supplied

2.39.2 RPCs associated with class: blob

RPC name: create

Overview:
Create a placeholder for a binary blob.
Signature:

```
(blob ref) create (session_id s, string mime_type)
```

Arguments:

type	name	description
`string`	mime_type	The mime-type of the blob. Defaults to 'application/octet-stream' if the empty string is supplied

Return Type: `blob ref`
The reference to the created blob

RPC name: destroy

Overview:
.
Signature:

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

Arguments:

type	name	description
`blob ref`	self	The reference of the blob to destroy

Return Type: `void`

RPC name: get_all

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

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

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

RPC name: get_all_records

Overview:
Return a map of blob references to blob records for all blobs known to the system.
Signature:

```
((blob ref -> blob record) Map) get_all_records (session_id s)
```

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

RPC name: get_uuid

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

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

Arguments:

type	name	description
`blob 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 blob.
Signature:

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

Arguments:

type	name	description
`blob 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 blob.
Signature:

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

Arguments:

type	name	description
blob 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 blob.
Signature:

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

Arguments:

type	name	description
blob 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 blob.
Signature:

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

Arguments:

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

Return Type: void

RPC name: get_size

Overview:
Get the size field of the given blob.
Signature:

```
int get_size (session_id s, blob ref self)
```

Arguments:

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

Return Type: `int`
value of the field

RPC name: get_last_updated

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

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

Arguments:

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

Return Type: `datetime`
value of the field

RPC name: get_mime_type

Overview:
Get the mime_type field of the given blob.
Signature:

```
string get_mime_type (session_id s, blob ref self)
```

Arguments:

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

Return Type: `string`
value of the field

RPC name: get_by_uuid

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

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

Arguments:

type	name	description
string	uuid	UUID of object to return

Return Type: `blob ref`
reference to the object

RPC name: get_record

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

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

Arguments:

type	name	description
blob ref	self	reference to the object

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

RPC name: get_by_name_label

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

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

Arguments:

type	name	description
string	label	label of object to return

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

2.40 Class: message

2.40.1 Fields for class: message

Name	**message**		
Description	*An message for the attention of the administrator.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	Unique identifier/object reference
RO_{run}	`name`	string	The name of the message
RO_{run}	`priority`	int	The message priority, 0 being low priority
RO_{run}	`cls`	cls	The class of the object this message is associated with
RO_{run}	`obj_uuid`	string	The uuid of the object this message is associated with
RO_{run}	`timestamp`	datetime	The time at which the message was created
RO_{run}	`body`	string	The body of the message

2.40.2 RPCs associated with class: message

RPC name: create

Overview:

.

Signature:

```
(message ref) create (session_id s, string name, int priority, cls cls, string obj_uuid, string body)
```

Arguments:

type	name	description
`string`	name	The name of the message
`int`	priority	The priority of the message
`cls`	cls	The class of object this message is associated with
`string`	obj_uuid	The uuid of the object this message is associated with
`string`	body	The body of the message

Return Type: `message ref`
The reference of the created message

RPC name: destroy

Overview:

.

Signature:

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

Arguments:

type	name	description
`message ref`	self	The reference of the message to destroy

Return Type: `void`

RPC name: get

Overview:

.

Signature:

```
((message ref -> message record) Map) get (session_id s, cls cls, string obj_uuid, datetime since)
```

Arguments:

type	name	description
`cls`	cls	The class of object
`string`	obj_uuid	The uuid of the object
`datetime`	since	The cutoff time

Return Type: `(message ref → message record) Map`
The relevant messages

RPC name: get_all

Overview:

.

Signature:

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

Return Type: `(message ref) Set`
The references to the messages

RPC name: get_since

Overview:

.

Signature:

```
((message ref -> message record) Map) get_since (session_id s, datetime since)
```

Arguments:

type	name	description
`datetime`	since	The cutoff time

Return Type: `(message ref → message record) Map`
The relevant messages

RPC name: get_record

Overview:

.

Signature:

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

Arguments:

type	name	description
`message ref`	self	The reference to the message

Return Type: `message record`
The message record

RPC name: get_by_uuid

Overview:

.

Signature:

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

Arguments:

type	name	description
`string`	uuid	The uuid of the message

Return Type: `message ref`
The message reference

RPC name: get_all_records

Overview:

.

Signature:

```
((message ref -> message record) Map) get_all_records (session_id s)
```

Return Type: `(message ref → message record) Map`
The messages

RPC name: get_all_records_where

Overview:

.

Signature:

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

Arguments:

type	name	description
`string`	expr	The expression to match (not currently used)

Return Type: `(message ref → message record) Map`
The messages

2.41 Class: secret

2.41.1 Fields for class: secret

Name	**secret**		
Description	*A secret.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	Unique identifier/object reference
RW	`value`	string	the secret
RW	`other_config`	(string → string) Map	other_config

2.41.2 RPCs associated with class: secret

RPC name: get_all

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

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

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

RPC name: get_all_records

Overview:
Return a map of secret references to secret records for all secrets known to the system.
Signature:

```
((secret ref -> secret record) Map) get_all_records (session_id s)
```

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

RPC name: get_uuid

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

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

Arguments:

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

Return Type: `string`
value of the field

RPC name: get_value

Overview:
Get the value field of the given secret.
Signature:

```
string get_value (session_id s, secret ref self)
```

Arguments:

type	name	description
secret ref	self	reference to the object

Return Type: string
value of the field

RPC name: set_value

Overview:
Set the value field of the given secret.
Signature:

```
void set_value (session_id s, secret ref self, string value)
```

Arguments:

type	name	description
secret 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 secret.
Signature:

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

Arguments:

type	name	description
secret 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 secret.
Signature:

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

Arguments:

type	name	description
secret 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 secret.
Signature:

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

Arguments:

type	name	description
secret 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 secret. If the key is not in that Map, then do nothing.
Signature:

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

Arguments:

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

Return Type: void

RPC name: create

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

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

Arguments:

type	name	description
`secret record`	args	All constructor arguments

Return Type: `secret ref`
reference to the newly created object

RPC name: destroy

Overview:
Destroy the specified secret instance.
Signature:

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

Arguments:

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

Return Type: `void`

RPC name: get_by_uuid

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

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

Arguments:

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

Return Type: `secret ref`
reference to the object

RPC name: get_record

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

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

Arguments:

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

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

2.42 Class: tunnel

2.42.1 Fields for class: tunnel

Name	**tunnel**		
Description	*A tunnel for network traffic.*		
Quals	Field	Type	Description
RO_{run}	`uuid`	string	Unique identifier/object reference
RO_{ins}	`access_PIF`	PIF ref	The interface through which the tunnel is accessed
RO_{ins}	`transport_PIF`	PIF ref	The interface used by the tunnel
RW	`status`	(string → string) Map	Status information about the tunnel
RW	`other_config`	(string → string) Map	Additional configuration

2.42.2 RPCs associated with class: tunnel

RPC name: create

Overview:
Create a tunnel.
Signature:

```
(tunnel ref) create (session_id s, PIF ref transport_PIF, network ref network)
```

Arguments:

type	name	description
`PIF ref`	transport_PIF	PIF which receives the tagged traffic
`network ref`	network	Network to receive the tunnelled traffic

Return Type: `tunnel ref`
The reference of the created tunnel object

Possible Error Codes: `OPENVSWITCH_NOT_ACTIVE`, `TRANSPORT_PIF_NOT_CONFIGURED`, `IS_TUNNEL_ACCESS_PIF`

RPC name: destroy

Overview:
Destroy a tunnel.
Signature:

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

Arguments:

type	name	description
`tunnel ref`	self	tunnel to destroy

Return Type: `void`

RPC name: get_all

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

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

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

RPC name: get_all_records

Overview:
Return a map of tunnel references to tunnel records for all tunnels known to the system.
Signature:

```
((tunnel ref -> tunnel record) Map) get_all_records (session_id s)
```

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

RPC name: get_uuid

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

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

Arguments:

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

Return Type: `string`
value of the field

RPC name: get_access_PIF

Overview:
Get the access_PIF field of the given tunnel.
Signature:

```
(PIF ref) get_access_PIF (session_id s, tunnel ref self)
```

Arguments:

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

Return Type: `PIF ref`

value of the field

RPC name: get_transport_PIF

Overview:
Get the transport_PIF field of the given tunnel.
Signature:

```
(PIF ref) get_transport_PIF (session_id s, tunnel ref self)
```

Arguments:

type	name	description
tunnel ref	self	reference to the object

Return Type: `PIF ref`
value of the field

RPC name: get_status

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

```
((string -> string) Map) get_status (session_id s, tunnel ref self)
```

Arguments:

type	name	description
tunnel ref	self	reference to the object

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

RPC name: set_status

Overview:
Set the status field of the given tunnel.
Signature:

```
void set_status (session_id s, tunnel ref self, (string -> string) Map value)
```

Arguments:

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

Return Type: `void`

RPC name: add_to_status

Overview:
Add the given key-value pair to the status field of the given tunnel.
Signature:

```
void add_to_status (session_id s, tunnel ref self, string key, string value)
```

Arguments:

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

Return Type: `void`

RPC name: remove_from_status

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

```
void remove_from_status (session_id s, tunnel ref self, string key)
```

Arguments:

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

Return Type: `void`

RPC name: get_other_config

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

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

Arguments:

type	name	description
tunnel 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 tunnel.
Signature:

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

Arguments:

type	name	description
`tunnel 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 tunnel.
Signature:

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

Arguments:

type	name	description
`tunnel 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 tunnel.
If the key is not in that Map, then do nothing.
Signature:

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

Arguments:

type	name	description
`tunnel 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 tunnel instance with the specified UUID.
Signature:

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

Arguments:

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

Return Type: `tunnel ref`
reference to the object

RPC name: get_record

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

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

Arguments:

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

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

2.43 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.

2.43.1 Error Codes

ACTIVATION_WHILE_NOT_FREE

An activation key can only be applied when the edition is set to 'free'.

No parameters.

AUTH_ALREADY_ENABLED

External authentication for this host is already enabled.

Signature:

```
AUTH_ALREADY_ENABLED(current auth_type, current service_name)
```

AUTH_DISABLE_FAILED

The host failed to disable external authentication.

Signature:

```
AUTH_DISABLE_FAILED(message)
```

AUTH_DISABLE_FAILED_PERMISSION_DENIED

The host failed to disable external authentication.

Signature:

```
AUTH_DISABLE_FAILED_PERMISSION_DENIED(message)
```

AUTH_DISABLE_FAILED_WRONG_CREDENTIALS

The host failed to disable external authentication.

Signature:

```
AUTH_DISABLE_FAILED_WRONG_CREDENTIALS(message)
```

AUTH_ENABLE_FAILED

The host failed to enable external authentication.

Signature:

```
AUTH_ENABLE_FAILED(message)
```

AUTH_ENABLE_FAILED_DOMAIN_LOOKUP_FAILED

The host failed to enable external authentication.

Signature:

```
AUTH_ENABLE_FAILED_DOMAIN_LOOKUP_FAILED(message)
```

AUTH_ENABLE_FAILED_PERMISSION_DENIED

The host failed to enable external authentication.

Signature:

```
AUTH_ENABLE_FAILED_PERMISSION_DENIED(message)
```

AUTH_ENABLE_FAILED_UNAVAILABLE

The host failed to enable external authentication.

Signature:

```
AUTH_ENABLE_FAILED_UNAVAILABLE(message)
```

AUTH_ENABLE_FAILED_WRONG_CREDENTIALS

The host failed to enable external authentication.

Signature:

```
AUTH_ENABLE_FAILED_WRONG_CREDENTIALS(message)
```

AUTH_IS_DISABLED

External authentication is disabled, unable to resolve subject name.

No parameters.

AUTH_SERVICE_ERROR

Error querying the external directory service.

Signature:

```
AUTH_SERVICE_ERROR(message)
```

AUTH_UNKNOWN_TYPE

Unknown type of external authentication.

Signature:

```
AUTH_UNKNOWN_TYPE(type)
```

BACKUP_SCRIPT_FAILED

The backup could not be performed because the backup script failed.

Signature:

```
BACKUP_SCRIPT_FAILED(log)
```

BOOTLOADER_FAILED

The bootloader returned an error

Signature:

```
BOOTLOADER_FAILED(vm, msg)
```

CANNOT_CONTACT_HOST

Cannot forward messages because the host cannot be contacted. The host may be switched off or there may be network connectivity problems.

Signature:

```
CANNOT_CONTACT_HOST(host)
```

CANNOT_CREATE_STATE_FILE

An HA statefile could not be created, perhaps because no SR with the appropriate capability was found.

No parameters.

CANNOT_DESTROY_SYSTEM_NETWORK

You tried to destroy a system network: these cannot be destroyed.

Signature:

```
CANNOT_DESTROY_SYSTEM_NETWORK(network)
```

CANNOT_ENABLE_REDO_LOG

Could not enable redo log.

Signature:

```
CANNOT_ENABLE_REDO_LOG(reason)
```

CANNOT_EVACUATE_HOST

This host cannot be evacuated.

Signature:

```
CANNOT_EVACUATE_HOST(errors)
```

CANNOT_FETCH_PATCH

The requested update could to be obtained from the master.

Signature:

```
CANNOT_FETCH_PATCH(uuid)
```

CANNOT_FIND_OEM_BACKUP_PARTITION

The backup partition to stream the updat to cannot be found

No parameters.

CANNOT_FIND_PATCH

The requested update could not be found. This can occur when you designate a new master or xe patch-clean. Please upload the update again

No parameters.

CANNOT_FIND_STATE_PARTITION

This operation could not be performed because the state partition could not be found

No parameters.

CANNOT_PLUG_VIF

Cannot plug VIF

Signature:

```
CANNOT_PLUG_VIF(VIF)
```

CANNOT_RESET_CONTROL_DOMAIN

The power-state of a control domain cannot be reset.

Signature:

```
CANNOT_RESET_CONTROL_DOMAIN(vm)
```

CERTIFICATE_ALREADY_EXISTS

A certificate already exists with the specified name.

Signature:

```
CERTIFICATE_ALREADY_EXISTS(name)
```

CERTIFICATE_CORRUPT

The specified certificate is corrupt or unreadable.

Signature:

```
CERTIFICATE_CORRUPT(name)
```

CERTIFICATE_DOES_NOT_EXIST

The specified certificate does not exist.

Signature:

```
CERTIFICATE_DOES_NOT_EXIST(name)
```

CERTIFICATE_LIBRARY_CORRUPT

The certificate library is corrupt or unreadable.

No parameters.

CERTIFICATE_NAME_INVALID

The specified certificate name is invalid.

Signature:

```
CERTIFICATE_NAME_INVALID(name)
```

CHANGE_PASSWORD_REJECTED

The system rejected the password change request; perhaps the new password was too short?

Signature:

```
CHANGE_PASSWORD_REJECTED(msg)
```

CPU_FEATURE_MASKING_NOT_SUPPORTED

The CPU does not support masking of features.

Signature:

```
CPU_FEATURE_MASKING_NOT_SUPPORTED(details)
```

CRL_ALREADY_EXISTS

A CRL already exists with the specified name.

Signature:

```
CRL_ALREADY_EXISTS(name)
```

CRL_CORRUPT

The specified CRL is corrupt or unreadable.

Signature:

```
CRL_CORRUPT(name)
```

CRL_DOES_NOT_EXIST

The specified CRL does not exist.

Signature:

```
CRL_DOES_NOT_EXIST(name)
```

CRL_NAME_INVALID

The specified CRL name is invalid.

Signature:

```
CRL_NAME_INVALID(name)
```

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)
```

DEVICE_NOT_ATTACHED

The operation could not be performed because the VBD was not connected to the VM.

Signature:

```
DEVICE_NOT_ATTACHED(VBD)
```

DOMAIN_BUILDER_ERROR

An internal error generated by the domain builder.

Signature:

```
DOMAIN_BUILDER_ERROR(function, code, message)
```

DOMAIN_EXISTS

The operation could not be performed because a domain still exists for the specified VM.

Signature:

```
DOMAIN_EXISTS(vm, domid)
```

DUPLICATE_PIF_DEVICE_NAME

A PIF with this specified device name already exists.

Signature:

```
DUPLICATE_PIF_DEVICE_NAME(device)
```

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.

FEATURE_RESTRICTED

The use of this feature is restricted.

No parameters.

FIELD_TYPE_ERROR

The value specified is of the wrong type

Signature:

```
FIELD_TYPE_ERROR(field)
```

HANDLE_INVALID

You gave an invalid object reference. 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)
```

HA_ABORT_NEW_MASTER

This host cannot accept the proposed new master setting at this time.

Signature:

```
HA_ABORT_NEW_MASTER(reason)
```

HA_CONSTRAINT_VIOLATION_NETWORK_NOT_SHARED

This operation cannot be performed because the referenced network is not properly shared. The network must either be entirely virtual or must be physically present via a currently_attached PIF on every host.

Signature:

```
HA_CONSTRAINT_VIOLATION_NETWORK_NOT_SHARED(network)
```

HA_CONSTRAINT_VIOLATION_SR_NOT_SHARED

This operation cannot be performed because the referenced SR is not properly shared. The SR must both be marked as shared and a currently_attached PBD must exist for each host.

Signature:

```
HA_CONSTRAINT_VIOLATION_SR_NOT_SHARED(SR)
```

HA_FAILED_TO_FORM_LIVESET

HA could not be enabled on the Pool because a liveset could not be formed: check storage and network heartbeat paths.

No parameters.

HA_HEARTBEAT_DAEMON_STARTUP_FAILED

The host could not join the liveset because the HA daemon failed to start.

No parameters.

HA_HOST_CANNOT_ACCESS_STATEFILE

The host could not join the liveset because the HA daemon could not access the heartbeat disk.

No parameters.

HA_HOST_CANNOT_SEE_PEERS

The operation failed because the HA software on the specified host could not see a subset of other hosts. Check your network connectivity.

Signature:

```
HA_HOST_CANNOT_SEE_PEERS(host, all, subset)
```

HA_HOST_IS_ARMED

The operation could not be performed while the host is still armed; it must be disarmed first

Signature:

```
HA_HOST_IS_ARMED(host)
```

HA_IS_ENABLED

The operation could not be performed because HA is enabled on the Pool

No parameters.

HA_LOST_STATEFILE

This host lost access to the HA statefile.

No parameters.

HA_NOT_ENABLED

The operation could not be performed because HA is not enabled on the Pool

No parameters.

HA_NOT_INSTALLED

The operation could not be performed because the HA software is not installed on this host.

Signature:

```
HA_NOT_INSTALLED(host)
```

HA_NO_PLAN

Cannot find a plan for placement of VMs as there are no other hosts available.

No parameters.

HA_OPERATION_WOULD_BREAK_FAILOVER_PLAN

This operation cannot be performed because it would invalidate VM failover planning such that the system would be unable to guarantee to restart protected VMs after a Host failure.

No parameters.

HA_POOL_IS_ENABLED_BUT_HOST_IS_DISABLED

This host cannot join the pool because the pool has HA enabled but this host has HA disabled.

No parameters.

HA_SHOULD_BE_FENCED

Host cannot rejoin pool because it should have fenced (it is not in the master's partition)

Signature:

```
HA_SHOULD_BE_FENCED(host)
```

HA_TOO_FEW_HOSTS

HA can only be enabled for 2 hosts or more. Note that 2 hosts requires a pre-configured quorum tiebreak script.

No parameters.

HOSTS_NOT_HOMOGENEOUS

The hosts in this pool are not homogeneous.

Signature:

```
HOSTS_NOT_HOMOGENEOUS(reason)
```

HOST_BROKEN

This host failed in the middle of an automatic failover operation and needs to retry the failover action

No parameters.

HOST_CANNOT_ATTACH_NETWORK

Host cannot attach network (in the case of NIC bonding, this may be because attaching the network on this host would require other networks [that are currently active] to be taken down).

Signature:

```
HOST_CANNOT_ATTACH_NETWORK(host, network)
```

HOST_CANNOT_DESTROY_SELF

The pool master host cannot be removed.

Signature:

```
HOST_CANNOT_DESTROY_SELF(host)
```

HOST_CANNOT_READ_METRICS

The metrics of this host could not be read.

No parameters.

HOST_CD_DRIVE_EMPTY

The host CDROM drive does not contain a valid CD

No parameters.

HOST_DISABLED

The specified host is disabled.

Signature:

```
HOST_DISABLED(host)
```

HOST_DISABLED_UNTIL_REBOOT

The specified host is disabled and cannot be re-enabled until after it has rebooted.

Signature:

```
HOST_DISABLED_UNTIL_REBOOT(host)
```

HOST_HAS_NO_MANAGEMENT_IP

The host failed to acquire an IP address on its management interface and therefore cannot contact the master.

No parameters.

HOST_HAS_RESIDENT_VMS

This host can not be forgotten because there are some user VMs still running

Signature:

```
HOST_HAS_RESIDENT_VMS(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 by (at least) the object of type and ref echoed below.

Signature:

```
HOST_IN_USE(host, type, ref)
```

HOST_IS_LIVE

This operation cannot be completed as the host is still live.

Signature:

```
HOST_IS_LIVE(host)
```

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_ITS_OWN_SLAVE

The host is its own slave. Please use pool-emergency-transition-to-master or pool-emergency-reset-master.

No parameters.

HOST_MASTER_CANNOT_TALK_BACK

The master reports that it cannot talk back to the slave on the supplied management IP address.

Signature:

```
HOST_MASTER_CANNOT_TALK_BACK(ip)
```

HOST_NAME_INVALID

The host name is invalid.

Signature:

```
HOST_NAME_INVALID(reason)
```

HOST_NOT_DISABLED

This operation cannot be performed because the host is not disabled. Please disable the host and then try again.

No parameters.

HOST_NOT_ENOUGH_FREE_MEMORY

Not enough host memory is available to perform this operation

Signature:

```
HOST_NOT_ENOUGH_FREE_MEMORY(needed, available)
```

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)
```

HOST_POWER_ON_MODE_DISABLED

This operation cannot be completed as the host power on mode is disabled.

No parameters.

HOST_STILL_BOOTING

The host is still booting.

No parameters.

HOST_UNKNOWN_TO_MASTER

The master says the host is not known to it. Perhaps the Host was deleted from the master's database? Perhaps the slave is pointing to the wrong master?

Signature:

```
HOST_UNKNOWN_TO_MASTER(host)
```

IMPORT_ERROR

The VM could not be imported.

Signature:

```
IMPORT_ERROR(msg)
```

IMPORT_ERROR_ATTACHED_DISKS_NOT_FOUND

The VM could not be imported because attached disks could not be found.

No parameters.

IMPORT_ERROR_CANNOT_HANDLE_CHUNKED

Cannot import VM using chunked encoding.

No parameters.

IMPORT_ERROR_FAILED_TO_FIND_OBJECT

The VM could not be imported because a required object could not be found.

Signature:

```
IMPORT_ERROR_FAILED_TO_FIND_OBJECT(id)
```

IMPORT_ERROR_PREMATURE_EOF

The VM could not be imported; the end of the file was reached prematurely.

No parameters.

IMPORT_ERROR_SOME_CHECKSUMS_FAILED

Some data checksums were incorrect; the VM may be corrupt.

No parameters.

IMPORT_ERROR_UNEXPECTED_FILE

The VM could not be imported because the XVA file is invalid: an unexpected file was encountered.

Signature:

```
IMPORT_ERROR_UNEXPECTED_FILE(filename_expected, filename_found)
```

IMPORT_INCOMPATIBLE_VERSION

The import failed because this export has been created by a different (incompatible) product version

No parameters.

INTERFACE_HAS_NO_IP

The specified interface cannot be used because it has no IP address

Signature:

```
INTERFACE_HAS_NO_IP(interface)
```

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_EDITION

The edition you supplied is invalid.

Signature:

```
INVALID_EDITION(edition)
```

INVALID_FEATURE_STRING

The given feature string is not valid.

Signature:

```
INVALID_FEATURE_STRING(details)
```

INVALID_IP_ADDRESS_SPECIFIED

A required parameter contained an invalid IP address

Signature:

```
INVALID_IP_ADDRESS_SPECIFIED(parameter)
```

INVALID_PATCH

The uploaded patch file is invalid

No parameters.

INVALID_PATCH_WITH_LOG

The uploaded patch file is invalid. See attached log for more details.

Signature:

```
INVALID_PATCH_WITH_LOG(log)
```

INVALID_VALUE

The value given is invalid

Signature:

```
INVALID_VALUE(field, value)
```

IS_TUNNEL_ACCESS_PIF

You tried to create a VLAN or tunnel on top of a tunnel access PIF - use the underlying transport PIF instead.

Signature:

```
IS_TUNNEL_ACCESS_PIF(PIF)
```

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_RUNNING_VMS

The host joining the pool cannot have any running 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 while joining it in the pool.

No parameters.

JOINING_HOST_SERVICE_FAILED

There was an error connecting to the host. the service contacted didn't reply properly.

No parameters.

LICENCE_RESTRICTION

This operation is not allowed under your license. Please contact your support representative.

No parameters.

LICENSE_CANNOT_DOWNGRADE_WHILE_IN_POOL

Cannot downgrade license while in pool. Please disband the pool first, then downgrade licenses on hosts separately.

No parameters.

LICENSE_CHECKOUT_ERROR

The license for the edition you requested is not available.

Signature:

```
LICENSE_CHECKOUT_ERROR(reason)
```

LICENSE_DOES_NOT_SUPPORT_POOLING

This host cannot join a pool because it's license does not support pooling

No parameters.

LICENSE_DOES_NOT_SUPPORT_XHA

XHA cannot be enabled because this host's license does not allow it

No parameters.

LICENSE_EXPIRED

Your license has expired. Please contact your support representative.

No parameters.

LICENSE_FILE_DEPRECATED

This license file is no longer accepted. Please upgrade to the new licensing system.

No parameters.

LICENSE_PROCESSING_ERROR

There was an error processing your license. Please contact your support representative.

No parameters.

LOCATION_NOT_UNIQUE

A VDI with the specified location already exists within the SR

Signature:

```
LOCATION_NOT_UNIQUE(SR, location)
```

MAC_DOES_NOT_EXIST

The MAC address specified doesn't exist on this host.

Signature:

```
MAC_DOES_NOT_EXIST(MAC)
```

MAC_INVALID

The MAC address specified is not valid.

Signature:

```
MAC_INVALID(MAC)
```

MAC_STILL_EXISTS

The MAC address specified still exists on this host.

Signature:

```
MAC_STILL_EXISTS(MAC)
```

MAP_DUPLICATE_KEY

You tried to add a key-value pair to a map, but that key is already there.

Signature:

```
MAP_DUPLICATE_KEY(type, param_name, uuid, key)
```

MESSAGE_DEPRECATED

This message has been deprecated.

No parameters.

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)
```

MISSING_CONNECTION_DETAILS

The license-server connection details (address or port) were missing or incomplete.

No parameters.

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_ALLOWED_ON_OEM_EDITION

This command is not allowed on the OEM edition.

Signature:

```
NOT_ALLOWED_ON_OEM_EDITION(command)
```

NOT_IMPLEMENTED

The function is not implemented

Signature:

```
NOT_IMPLEMENTED(function)
```

NOT_IN_EMERGENCY_MODE

This pool is not in emergency mode.

No parameters.

NOT_SUPPORTED_DURING_UPGRADE

This operation is not supported during an upgrade.

No parameters.

NO_HOSTS_AVAILABLE

There were no hosts available to complete the specified operation.

No parameters.

OBJECT_NOLONGER_EXISTS

The specified object no longer exists.

No parameters.

ONLY_ALLOWED_ON_OEM_EDITION

This command is only allowed on the OEM edition.

Signature:

```
ONLY_ALLOWED_ON_OEM_EDITION(command)
```

OPENVSWITCH_NOT_ACTIVE

This operation needs the OpenVSwitch networking backend to be enabled on all hosts in the pool.

No parameters.

OPERATION_BLOCKED

You attempted an operation that was explicitly blocked (see the blocked_operations field of the given object).

Signature:

```
OPERATION_BLOCKED(ref, code)
```

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)
```

OUT_OF_SPACE

There is not enough space to upload the update

Signature:

```
OUT_OF_SPACE(location)
```

PATCH_ALREADY_APPLIED

This patch has already been applied

Signature:

```
PATCH_ALREADY_APPLIED(patch)
```

PATCH_ALREADY_EXISTS

The uploaded patch file already exists

Signature:

```
PATCH_ALREADY_EXISTS(uuid)
```

PATCH_APPLY_FAILED

The patch apply failed. Please see attached output.

Signature:

```
PATCH_APPLY_FAILED(output)
```

PATCH_IS_APPLIED

The specified patch is applied and cannot be destroyed.

No parameters.

PATCH_PRECHECK_FAILED_PREREQUISITE_MISSING

The patch precheck stage failed: prerequisite patches are missing.

Signature:

```
PATCH_PRECHECK_FAILED_PREREQUISITE_MISSING(patch, prerequisite_patch_uuid_list)
```

PATCH_PRECHECK_FAILED_UNKNOWN_ERROR

The patch precheck stage failed with an unknown error. See attached info for more details.

Signature:

```
PATCH_PRECHECK_FAILED_UNKNOWN_ERROR(patch, info)
```

PATCH_PRECHECK_FAILED_VM_RUNNING

The patch precheck stage failed: there are one or more VMs still running on the server. All VMs must be suspended before the patch can be applied.

Signature:

```
PATCH_PRECHECK_FAILED_VM_RUNNING(patch)
```

PATCH_PRECHECK_FAILED_WRONG_SERVER_VERSION

The patch precheck stage failed: the server is of an incorrect version.

Signature:

```
PATCH_PRECHECK_FAILED_WRONG_SERVER_VERSION(patch, found_version, required_version)
```

PBD_EXISTS

A PBD already exists connecting the SR to the host

Signature:

```
PBD_EXISTS(sr, host, pbd)
```

PERMISSION_DENIED

Caller not allowed to perform this operation.

Signature:

```
PERMISSION_DENIED(message)
```

PIF_ALREADY_BONDED

This operation cannot be performed because the pif is bonded.

Signature:

```
PIF_ALREADY_BONDED(PIF)
```

PIF_BOND_NEEDS_MORE_MEMBERS

A bond must consist of at least two member interfaces

No parameters.

PIF_CANNOT_BOND_CROSS_HOST

You cannot bond interfaces across different hosts.

No parameters.

PIF_CONFIGURATION_ERROR

An unknown error occurred while attempting to configure an interface.

Signature:

```
PIF_CONFIGURATION_ERROR(PIF, msg)
```

PIF_DEVICE_NOT_FOUND

The specified device was not found.

No parameters.

PIF_DOES_NOT_ALLOW_UNPLUG

The operation you requested cannot be performed because the specified PIF does not allow unplug.

Signature:

```
PIF_DOES_NOT_ALLOW_UNPLUG(PIF)
```

PIF_HAS_NO_NETWORK_CONFIGURATION

PIF has no IP configuration (mode curently set to 'none')

No parameters.

PIF_IS_MANAGEMENT_INTERFACE

The operation you requested cannot be performed because the specified PIF is the management interface.

Signature:

```
PIF_IS_MANAGEMENT_INTERFACE(PIF)
```

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_IS_VLAN

You tried to create a VLAN on top of another VLAN - use the underlying physical PIF/bond instead

Signature:

```
PIF_IS_VLAN(PIF)
```

PIF_TUNNEL_STILL_EXISTS

Operation cannot proceed while a tunnel exists on this interface.

Signature:

```
PIF_TUNNEL_STILL_EXISTS(PIF)
```

PIF_VLAN_EXISTS

You tried to create a PIF, but it already exists.

Signature:

```
PIF_VLAN_EXISTS(PIF)
```

PIF_VLAN_STILL_EXISTS

Operation cannot proceed while a VLAN exists on this interface.

Signature:

```
PIF_VLAN_STILL_EXISTS(PIF)
```

POOL_AUTH_ALREADY_ENABLED

External authentication in this pool is already enabled for at least one host.

Signature:

```
POOL_AUTH_ALREADY_ENABLED(host)
```

POOL_AUTH_DISABLE_FAILED

The pool failed to disable the external authentication of at least one host.

Signature:

```
POOL_AUTH_DISABLE_FAILED(host, message)
```

POOL_AUTH_DISABLE_FAILED_PERMISSION_DENIED

The pool failed to disable the external authentication of at least one host.

Signature:

```
POOL_AUTH_DISABLE_FAILED_PERMISSION_DENIED(host, message)
```

POOL_AUTH_DISABLE_FAILED_WRONG_CREDENTIALS

The pool failed to disable the external authentication of at least one host.

Signature:

```
POOL_AUTH_DISABLE_FAILED_WRONG_CREDENTIALS(host, message)
```

POOL_AUTH_ENABLE_FAILED

The pool failed to enable external authentication.

Signature:

```
POOL_AUTH_ENABLE_FAILED(host, message)
```

POOL_AUTH_ENABLE_FAILED_DOMAIN_LOOKUP_FAILED

The pool failed to enable external authentication.

Signature:

```
POOL_AUTH_ENABLE_FAILED_DOMAIN_LOOKUP_FAILED(host, message)
```

POOL_AUTH_ENABLE_FAILED_DUPLICATE_HOSTNAME

The pool failed to enable external authentication.

Signature:

```
POOL_AUTH_ENABLE_FAILED_DUPLICATE_HOSTNAME(host, message)
```

POOL_AUTH_ENABLE_FAILED_INVALID_OU

The pool failed to enable external authentication.

Signature:

```
POOL_AUTH_ENABLE_FAILED_INVALID_OU(host, message)
```

POOL_AUTH_ENABLE_FAILED_PERMISSION_DENIED

The pool failed to enable external authentication.

Signature:

```
POOL_AUTH_ENABLE_FAILED_PERMISSION_DENIED(host, message)
```

POOL_AUTH_ENABLE_FAILED_WRONG_CREDENTIALS

The pool failed to enable external authentication.

Signature:

```
POOL_AUTH_ENABLE_FAILED_WRONG_CREDENTIALS(host, message)
```

POOL_JOINING_EXTERNAL_AUTH_MISMATCH

Cannot join pool whose external authentication configuration is different.

No parameters.

POOL_JOINING_HOST_MUST_HAVE_PHYSICAL_MANAGEMENT_NIC

The host joining the pool must have a physical management NIC (i.e. the management NIC must not be on a VLAN or bonded PIF).

No parameters.

POOL_JOINING_HOST_MUST_HAVE_SAME_PRODUCT_VERSION

The host joining the pool must have the same product version as the pool master.

No parameters.

PROVISION_FAILED_OUT_OF_SPACE

The provision call failed because it ran out of space.

No parameters.

PROVISION_ONLY_ALLOWED_ON_TEMPLATE

The provision call can only be invoked on templates, not regular VMs.

No parameters.

RBAC_PERMISSION_DENIED

RBAC permission denied.

Signature:

```
RBAC_PERMISSION_DENIED(permission, message)
```

REDO_LOG_IS_ENABLED

The operation could not be performed because a redo log is enabled on the Pool.

No parameters.

RESTORE_INCOMPATIBLE_VERSION

The restore could not be performed because this backup has been created by a different (incompatible) product version

No parameters.

RESTORE_SCRIPT_FAILED

The restore could not be performed because the restore script failed. Is the file corrupt?

Signature:

```
RESTORE_SCRIPT_FAILED(log)
```

RESTORE_TARGET_MGMT_IF_NOT_IN_BACKUP

The restore could not be performed because the host's current management interface is not in the backup. The interfaces mentioned in the backup are:

No parameters.

RESTORE_TARGET_MISSING_DEVICE

The restore could not be performed because a network interface is missing

Signature:

```
RESTORE_TARGET_MISSING_DEVICE(device)
```

ROLE_ALREADY_EXISTS

Role already exists.

No parameters.

ROLE_NOT_FOUND

Role cannot be found.

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 reference. 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)
```

SLAVE_REQUIRES_MANAGEMENT_INTERFACE

The management interface on a slave cannot be disabled because the slave would enter emergency mode.

No parameters.

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(status, stdout, stderr)
```

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_MULTIPLE_PBDS

The SR.shared flag cannot be set to false while the SR remains connected to multiple hosts

Signature:

```
SR_HAS_MULTIPLE_PBDS(PBD)
```

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_INDESTRUCTIBLE

The SR could not be destroyed, as the 'indestructible' flag was set on it.

Signature:

```
SR_INDESTRUCTIBLE(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_REQUIRES_UPGRADE

The operation cannot be performed until the SR has been upgraded

Signature:

```
SR_REQUIRES_UPGRADE(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.

SSL_VERIFY_ERROR

The remote system's SSL certificate failed to verify against our certificate library.

Signature:

```
SSL_VERIFY_ERROR(reason)
```

SUBJECT_ALREADY_EXISTS

Subject already exists.

No parameters.

SUBJECT_CANNOT_BE_RESOLVED

Subject cannot be resolved by the external directory service.

No parameters.

SYSTEM_STATUS_MUST_USE_TAR_ON_OEM

You must use tar output to retrieve system status from an OEM host.

No parameters.

SYSTEM_STATUS_RETRIEVAL_FAILED

Retrieving system status from the host failed. A diagnostic reason suitable for support organisations is also returned.

Signature:

```
SYSTEM_STATUS_RETRIEVAL_FAILED(reason)
```

TASK_CANCELLED

The request was asynchronously cancelled.

Signature:

```
TASK_CANCELLED(task)
```

TOO_BUSY

The request was rejected because the server is too busy.

No parameters.

TOO_MANY_PENDING_TASKS

The request was rejected because there are too many pending tasks on the server.

No parameters.

TRANSPORT_PIF_NOT_CONFIGURED

The tunnel transport PIF has no IP configuration set.

Signature:

```
TRANSPORT_PIF_NOT_CONFIGURED(PIF)
```

UNKNOWN_BOOTLOADER

The requested bootloader is unknown

Signature:

```
UNKNOWN_BOOTLOADER(vm, bootloader)
```

USER_IS_NOT_LOCAL_SUPERUSER

Only the local superuser can execute this operation

Signature:

```
USER_IS_NOT_LOCAL_SUPERUSER(msg)
```

UUID_INVALID

The uuid you supplied was invalid.

Signature:

```
UUID_INVALID(type, uuid)
```

V6D_FAILURE

There was a problem with the license daemon (v6d). Is it running?

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)
```

VBD_NOT_UNPLUGGABLE

Drive could not be hot-unplugged because it is not marked as unpluggable

Signature:

```
VBD_NOT_UNPLUGGABLE(vbd)
```

VBD_TRAY_LOCKED

This VM has locked the DVD drive tray, so the disk cannot be ejected

Signature:

```
VBD_TRAY_LOCKED(vbd)
```

VDI_INCOMPATIBLE_TYPE

This operation cannot be performed because the specified VDI is of an incompatible type (eg: an HA statefile cannot be attached to a guest)

Signature:

```
VDI_INCOMPATIBLE_TYPE(vdi, type)
```

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_A_PHYSICAL_DEVICE

The operation cannot be performed on physical device

Signature:

```
VDI_IS_A_PHYSICAL_DEVICE(vdi)
```

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_LOCATION_MISSING

This operation cannot be performed because the specified VDI could not be found in the specified SR

Signature:

```
VDI_LOCATION_MISSING(sr, location)
```

VDI_MISSING

This operation cannot be performed because the specified VDI could not be found on the storage substrate

Signature:

```
VDI_MISSING(sr, vdi)
```

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_NOT_MANAGED

This operation cannot be performed because the system does not manage this VDI

Signature:

```
VDI_NOT_MANAGED(vdi)
```

VDI_READONLY

The operation required write access but this VDI is read-only

Signature:

```
VDI_READONLY(vdi)
```

VIF_IN_USE

Network has active VIFs

Signature:

```
VIF_IN_USE(network, VIF)
```

VLAN_TAG_INVALID

You tried to create a VLAN, but the tag you gave was invalid – it must be between 0 and 4094. The parameter echoes the VLAN tag you gave.

Signature:

```
VLAN_TAG_INVALID(VLAN)
```

VMPP_ARCHIVE_MORE_FREQUENT_THAN_BACKUP

Archive more frequent than backup.

No parameters.

VMPP_HAS_VM

There is at least on VM assigned to this protection policy.

No parameters.

VMS_FAILED_TO_COOPERATE

The given VMs failed to release memory when instructed to do so

No parameters.

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_BIOS_STRINGS_ALREADY_SET

The BIOS strings for this VM have already been set and cannot be changed anymore.

No parameters.

VM_CANNOT_DELETE_DEFAULT_TEMPLATE

You cannot delete the specified default template.

Signature:

```
VM_CANNOT_DELETE_DEFAULT_TEMPLATE(vm)
```

VM_CHECKPOINT_RESUME_FAILED

An error occured while restoring the memory image of the specified virtual machine

Signature:

```
VM_CHECKPOINT_RESUME_FAILED(vm)
```

VM_CHECKPOINT_SUSPEND_FAILED

An error occured while saving the memory image of the specified virtual machine

Signature:

```
VM_CHECKPOINT_SUSPEND_FAILED(vm)
```

VM_CRASHED

The VM crashed

Signature:

```
VM_CRASHED(vm)
```

VM_DUPLICATE_VBD_DEVICE

The specified VM has a duplicate VBD device and cannot be started.

Signature:

```
VM_DUPLICATE_VBD_DEVICE(vm, vbd, device)
```

VM_FAILED_SHUTDOWN_ACKNOWLEDGMENT

VM didn't acknowledge the need to shutdown.

No parameters.

VM_HALTED

The VM unexpectedly halted

Signature:

```
VM_HALTED(vm)
```

VM_HVM_REQUIRED

HVM is required for this operation

Signature:

```
VM_HVM_REQUIRED(vm)
```

VM_IS_PROTECTED

This operation cannot be performed because the specified VM is protected by xHA

Signature:

```
VM_IS_PROTECTED(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_NOT_RESIDENT_HERE

The specified VM is not currently resident on the specified host.

Signature:

```
VM_NOT_RESIDENT_HERE(vm, host)
```

VM_NO_CRASHDUMP_SR

This VM does not have a crashdump SR specified.

Signature:

```
VM_NO_CRASHDUMP_SR(vm)
```

VM_NO_SUSPEND_SR

This VM does not have a suspend SR specified.

Signature:

```
VM_NO_SUSPEND_SR(vm)
```

VM_NO_VCPUS

You need at least 1 VCPU to start a VM

Signature:

```
VM_NO_VCPUS(vm)
```

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_REBOOTED

The VM unexpectedly rebooted

Signature:

```
VM_REBOOTED(vm)
```

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_REVERT_FAILED

An error occured while reverting the specified virtual machine to the specified snapshot

Signature:

```
VM_REVERT_FAILED(vm, snapshot)
```

VM_SHUTDOWN_TIMEOUT

VM failed to shutdown before the timeout expired

Signature:

```
VM_SHUTDOWN_TIMEOUT(vm, timeout)
```

VM_SNAPSHOT_WITH_QUIESCE_FAILED

The quiesced-snapshot operation failed for an unexpected reason

Signature:

```
VM_SNAPSHOT_WITH_QUIESCE_FAILED(vm)
```

VM_SNAPSHOT_WITH_QUIESCE_NOT_SUPPORTED

The VSS plug-in is not installed on this virtual machine

Signature:

```
VM_SNAPSHOT_WITH_QUIESCE_NOT_SUPPORTED(vm, error)
```

VM_SNAPSHOT_WITH_QUIESCE_PLUGIN_DEOS_NOT_RESPOND

The VSS plug-in cannot be contacted

Signature:

```
VM_SNAPSHOT_WITH_QUIESCE_PLUGIN_DEOS_NOT_RESPOND(vm)
```

VM_SNAPSHOT_WITH_QUIESCE_TIMEOUT

The VSS plug-in has timed out

Signature:

```
VM_SNAPSHOT_WITH_QUIESCE_TIMEOUT(vm)
```

VM_TOO_MANY_VCPUS

Too many VCPUs to start this VM

Signature:

```
VM_TOO_MANY_VCPUS(vm)
```

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 incompatible 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)
```

WLB_AUTHENTICATION_FAILED

The WLB server rejected our configured authentication details.

No parameters.

WLB_CONNECTION_REFUSED

The WLB server refused a connection to XenServer.

No parameters.

WLB_CONNECTION_RESET

The connection to the WLB server was reset.

No parameters.

WLB_DISABLED

This pool has wlb-enabled set to false.

No parameters.

WLB_INTERNAL_ERROR

The WLB server reported an internal error.

No parameters.

WLB_MALFORMED_REQUEST

The WLB server rejected XenServer's request as malformed.

No parameters.

WLB_MALFORMED_RESPONSE

The WLB server said something that XenServer wasn't expecting or didn't understand. The method called on the WLB server, a diagnostic reason, and the response from WLB are returned.

Signature:

```
WLB_MALFORMED_RESPONSE(method, reason, response)
```

WLB_NOT_INITIALIZED

No WLB connection is configured.

No parameters.

WLB_TIMEOUT

The communication with the WLB server timed out.

Signature:

```
WLB_TIMEOUT(configured_timeout)
```

WLB_UNKNOWN_HOST

The configured WLB server name failed to resolve in DNS.

No parameters.

WLB_URL_INVALID

The WLB URL is invalid. Ensure it is in format: ¡ipaddress¿:¡port¿. The configured/given URL is returned.

Signature:

```
WLB_URL_INVALID(url)
```

WLB_XENSERVER_AUTHENTICATION_FAILED

The WLB server reported that XenServer rejected its configured authentication details.

No parameters.

WLB_XENSERVER_CONNECTION_REFUSED

The WLB server reported that XenServer refused it a connection (even though we're connecting perfectly fine in the other direction).

No parameters.

WLB_XENSERVER_MALFORMED_RESPONSE

The WLB server reported that XenServer said something to it that WLB wasn't expecting or didn't understand.

No parameters.

WLB_XENSERVER_TIMEOUT

The WLB server reported that communication with XenServer timed out.

No parameters.

WLB_XENSERVER_UNKNOWN_HOST

The WLB server reported that its configured server name for this XenServer instance failed to resolve in DNS.

No parameters.

XAPI_HOOK_FAILED

3rd party xapi hook failed

Signature:

```
XAPI_HOOK_FAILED(hook_name, reason, stdout, exit_code)
```

XENAPI_MISSING_PLUGIN

The requested plugin could not be found.

Signature:

```
XENAPI_MISSING_PLUGIN(name)
```

XENAPI_PLUGIN_FAILURE

There was a failure communicating with the plugin.

Signature:

```
XENAPI_PLUGIN_FAILURE(status, stdout, stderr)
```

XEN_VSS_REQ_ERROR_ADDING_VOLUME_TO_SNAPSET_FAILED

Some volumes to be snapshot could not be added to the VSS snapshot set

Signature:

```
XEN_VSS_REQ_ERROR_ADDING_VOLUME_TO_SNAPSET_FAILED(vm, error_code)
```

XEN_VSS_REQ_ERROR_CREATING_SNAPSHOT

An attempt to create the snapshots failed

Signature:

```
XEN_VSS_REQ_ERROR_CREATING_SNAPSHOT(vm, error_code)
```

XEN_VSS_REQ_ERROR_CREATING_SNAPSHOT_XML_STRING

Could not create the XML string generated by the transportable snapshot

Signature:

```
XEN_VSS_REQ_ERROR_CREATING_SNAPSHOT_XML_STRING(vm, error_code)
```

XEN_VSS_REQ_ERROR_INIT_FAILED

Initialization of the VSS requestor failed

Signature:

```
XEN_VSS_REQ_ERROR_INIT_FAILED(vm, error_code)
```

XEN_VSS_REQ_ERROR_NO_VOLUMES_SUPPORTED

Could not find any volumes supported by the Citrix XenServer Vss Provider

Signature:

```
XEN_VSS_REQ_ERROR_NO_VOLUMES_SUPPORTED(vm, error_code)
```

XEN_VSS_REQ_ERROR_PREPARING_WRITERS

An attempt to prepare VSS writers for the snapshot failed

Signature:

```
XEN_VSS_REQ_ERROR_PREPARING_WRITERS(vm, error_code)
```

XEN_VSS_REQ_ERROR_PROV_NOT_LOADED

The Citrix XenServer Vss Provider is not loaded

Signature:

```
XEN_VSS_REQ_ERROR_PROV_NOT_LOADED(vm, error_code)
```

XEN_VSS_REQ_ERROR_START_SNAPSHOT_SET_FAILED

An attempt to start a new VSS snapshot failed

Signature:

```
XEN_VSS_REQ_ERROR_START_SNAPSHOT_SET_FAILED(vm, error_code)
```

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)
```
