



# Citrix XenServer Management API

Version: API Revision 2.3

Date: December 9, 2014

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
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
<b>2</b>	<b>API Reference</b>	<b>12</b>
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 . . . . .	23
2.4.1	Fields for class: session . . . . .	23
2.4.2	RPCs associated with class: session . . . . .	23
2.5	Class: auth . . . . .	33
2.5.1	Fields for class: auth . . . . .	33
2.5.2	RPCs associated with class: auth . . . . .	33
2.6	Class: subject . . . . .	35
2.6.1	Fields for class: subject . . . . .	35
2.6.2	RPCs associated with class: subject . . . . .	35
2.7	Class: role . . . . .	40
2.7.1	Fields for class: role . . . . .	40
2.7.2	RPCs associated with class: role . . . . .	40
2.8	Class: task . . . . .	45
2.8.1	Fields for class: task . . . . .	45
2.8.2	RPCs associated with class: task . . . . .	46
2.9	Class: event . . . . .	55
2.9.1	Fields for class: event . . . . .	55
2.9.2	RPCs associated with class: event . . . . .	55
2.10	Class: pool . . . . .	58
2.10.1	Fields for class: pool . . . . .	58

2.10.2	RPCs associated with class: pool . . . . .	59
2.11	Class: pool_patch . . . . .	88
2.11.1	Fields for class: pool_patch . . . . .	88
2.11.2	RPCs associated with class: pool_patch . . . . .	88
2.12	Class: VM . . . . .	96
2.12.1	Fields for class: VM . . . . .	96
2.12.2	RPCs associated with class: VM . . . . .	99
2.13	Class: VM_metrics . . . . .	162
2.13.1	Fields for class: VM_metrics . . . . .	162
2.13.2	RPCs associated with class: VM_metrics . . . . .	162
2.14	Class: VM_guest_metrics . . . . .	169
2.14.1	Fields for class: VM_guest_metrics . . . . .	169
2.14.2	RPCs associated with class: VM_guest_metrics . . . . .	169
2.15	Class: VMPP . . . . .	176
2.15.1	Fields for class: VMPP . . . . .	176
2.15.2	RPCs associated with class: VMPP . . . . .	176
2.16	Class: VM_appliance . . . . .	194
2.16.1	Fields for class: VM_appliance . . . . .	194
2.16.2	RPCs associated with class: VM_appliance . . . . .	194
2.17	Class: DR_task . . . . .	202
2.17.1	Fields for class: DR_task . . . . .	202
2.17.2	RPCs associated with class: DR_task . . . . .	202
2.18	Class: host . . . . .	205
2.18.1	Fields for class: host . . . . .	205
2.18.2	RPCs associated with class: host . . . . .	206
2.19	Class: host_crashdump . . . . .	248
2.19.1	Fields for class: host_crashdump . . . . .	248
2.19.2	RPCs associated with class: host_crashdump . . . . .	248
2.20	Class: host_patch . . . . .	253
2.20.1	Fields for class: host_patch . . . . .	253
2.20.2	RPCs associated with class: host_patch . . . . .	253
2.21	Class: host_metrics . . . . .	260
2.21.1	Fields for class: host_metrics . . . . .	260
2.21.2	RPCs associated with class: host_metrics . . . . .	260
2.22	Class: host_cpu . . . . .	265
2.22.1	Fields for class: host_cpu . . . . .	265
2.22.2	RPCs associated with class: host_cpu . . . . .	265
2.23	Class: network . . . . .	272
2.23.1	Fields for class: network . . . . .	272
2.23.2	RPCs associated with class: network . . . . .	272
2.24	Class: VIF . . . . .	283
2.24.1	Fields for class: VIF . . . . .	283
2.24.2	RPCs associated with class: VIF . . . . .	284
2.25	Class: VIF_metrics . . . . .	299
2.25.1	Fields for class: VIF_metrics . . . . .	299
2.25.2	RPCs associated with class: VIF_metrics . . . . .	299
2.26	Class: PIF . . . . .	303
2.26.1	Fields for class: PIF . . . . .	303
2.26.2	RPCs associated with class: PIF . . . . .	304
2.27	Class: PIF_metrics . . . . .	322
2.27.1	Fields for class: PIF_metrics . . . . .	322
2.27.2	RPCs associated with class: PIF_metrics . . . . .	322
2.28	Class: Bond . . . . .	329
2.28.1	Fields for class: Bond . . . . .	329

2.28.2	RPCs associated with class: Bond	329
2.29	Class: VLAN	336
2.29.1	Fields for class: VLAN	336
2.29.2	RPCs associated with class: VLAN	336
2.30	Class: SM	341
2.30.1	Fields for class: SM	341
2.30.2	RPCs associated with class: SM	341
2.31	Class: SR	349
2.31.1	Fields for class: SR	349
2.31.2	RPCs associated with class: SR	349
2.32	Class: VDI	368
2.32.1	Fields for class: VDI	368
2.32.2	RPCs associated with class: VDI	369
2.33	Class: VBD	396
2.33.1	Fields for class: VBD	396
2.33.2	RPCs associated with class: VBD	396
2.34	Class: VBD_metrics	412
2.34.1	Fields for class: VBD_metrics	412
2.34.2	RPCs associated with class: VBD_metrics	412
2.35	Class: PBD	416
2.35.1	Fields for class: PBD	416
2.35.2	RPCs associated with class: PBD	416
2.36	Class: crashdump	422
2.36.1	Fields for class: crashdump	422
2.36.2	RPCs associated with class: crashdump	422
2.37	Class: VTPM	426
2.37.1	Fields for class: VTPM	426
2.37.2	RPCs associated with class: VTPM	426
2.38	Class: console	429
2.38.1	Fields for class: console	429
2.38.2	RPCs associated with class: console	429
2.39	Class: user	434
2.39.1	Fields for class: user	434
2.39.2	RPCs associated with class: user	434
2.40	Class: data_source	438
2.40.1	Fields for class: data_source	438
2.40.2	RPCs associated with class: data_source	438
2.41	Class: blob	439
2.41.1	Fields for class: blob	439
2.41.2	RPCs associated with class: blob	439
2.42	Class: message	445
2.42.1	Fields for class: message	445
2.42.2	RPCs associated with class: message	445
2.43	Class: secret	449
2.43.1	Fields for class: secret	449
2.43.2	RPCs associated with class: secret	449
2.44	Class: tunnel	454
2.44.1	Fields for class: tunnel	454
2.44.2	RPCs associated with class: tunnel	454
2.45	Class: PCI	460
2.45.1	Fields for class: PCI	460
2.45.2	RPCs associated with class: PCI	460
2.46	Class: PGPU	466
2.46.1	Fields for class: PGPU	466

---

2.46.2	RPCs associated with class: PGPU . . . . .	466
2.47	Class: GPU_group . . . . .	473
2.47.1	Fields for class: GPU_group . . . . .	473
2.47.2	RPCs associated with class: GPU_group . . . . .	473
2.48	Class: VGPU . . . . .	482
2.48.1	Fields for class: VGPU . . . . .	482
2.48.2	RPCs associated with class: VGPU . . . . .	482
2.49	Class: VGPU_type . . . . .	488
2.49.1	Fields for class: VGPU_type . . . . .	488
2.49.2	RPCs associated with class: VGPU_type . . . . .	488
2.50	Error Handling . . . . .	494
2.50.1	Error Codes . . . . .	495

# 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<sub>run</sub>*: the field is Read Only. Furthermore, its value is automatically computed at runtime. For example: current CPU load and disk IO throughput.
- *RO<sub>ins</sub>*: 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<sub>1</sub>, t<sub>2</sub>) Map*: signifies a mapping from values of type *t<sub>1</sub>* to values of type *t<sub>2</sub>*.

Note that there are a number of cases where *Refs* 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 existence 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<sup>1</sup>;
- 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 accessor associated with it that sets its value:

- For *RW* (*Read/Write*), 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<sub>ins</sub>*. 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.

---

<sup>1</sup>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_vms`. These types are mapped onto XML-RPC types in a straight-forward manner:

- Floats, Booleans, 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 subsequent 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 asynchronous 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 omitted 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'
```

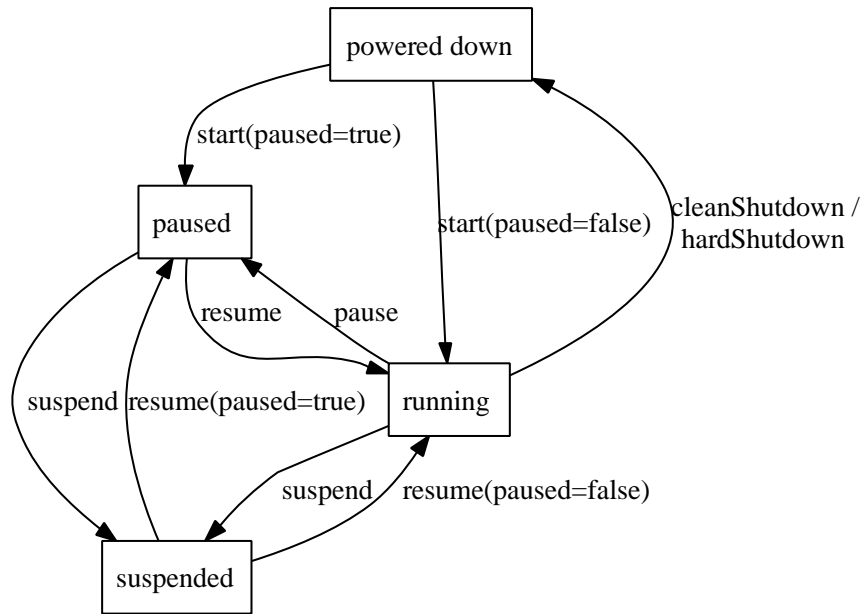


Figure 1.1: VM Lifecycle

## 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
<code>session</code>	A session
<code>auth</code>	Management of remote authentication services
<code>subject</code>	A user or group that can log in xapi
<code>role</code>	A set of permissions associated with a subject
<code>task</code>	A long-running asynchronous task
<code>event</code>	Asynchronous event registration and handling
<code>pool</code>	Pool-wide information
<code>pool_patch</code>	Pool-wide patches
<code>VM</code>	A virtual machine (or 'guest')
<code>VM_metrics</code>	The metrics associated with a VM
<code>VM_guest_metrics</code>	The metrics reported by the guest (as opposed to inferred from outside)
<code>VMPP</code>	VM Protection Policy
<code>VM_appliance</code>	VM appliance
<code>DR_task</code>	DR task
<code>host</code>	A physical host
<code>host_crashdump</code>	Represents a host crash dump
<code>host_patch</code>	Represents a patch stored on a server
<code>host_metrics</code>	The metrics associated with a host
<code>host_cpu</code>	A physical CPU
<code>network</code>	A virtual network
<code>VIF</code>	A virtual network interface
<code>VIF_metrics</code>	The metrics associated with a virtual network device
<code>PIF</code>	A physical network interface (note separate VLANs are represented as several PIFs)
<code>PIF_metrics</code>	The metrics associated with a physical network interface
<code>Bond</code>	
<code>VLAN</code>	A VLAN mux/demux
<code>SM</code>	A storage manager plugin
<code>SR</code>	A storage repository
<code>VDI</code>	A virtual disk image
<code>VBD</code>	A virtual block device
<code>VBD_metrics</code>	The metrics associated with a virtual block device
<code>PBD</code>	The physical block devices through which hosts access SRs
<code>crashdump</code>	A VM crashdump
<code>VTPM</code>	A virtual TPM device
<code>console</code>	A console
<code>user</code>	A user of the system
<code>data_source</code>	Data sources for logging in RRDs
<code>blob</code>	A placeholder for a binary blob
<code>message</code>	An message for the attention of the administrator
<code>secret</code>	A secret
<code>tunnel</code>	A tunnel for network traffic
<code>PCI</code>	A PCI device
<code>PGPU</code>	A physical GPU (pGPU)
<code>GPU_group</code>	A group of compatible GPUs across the resource pool
<code>VGPU</code>	A virtual GPU (vGPU)
<code>VGPU_type</code>	A type of virtual GPU

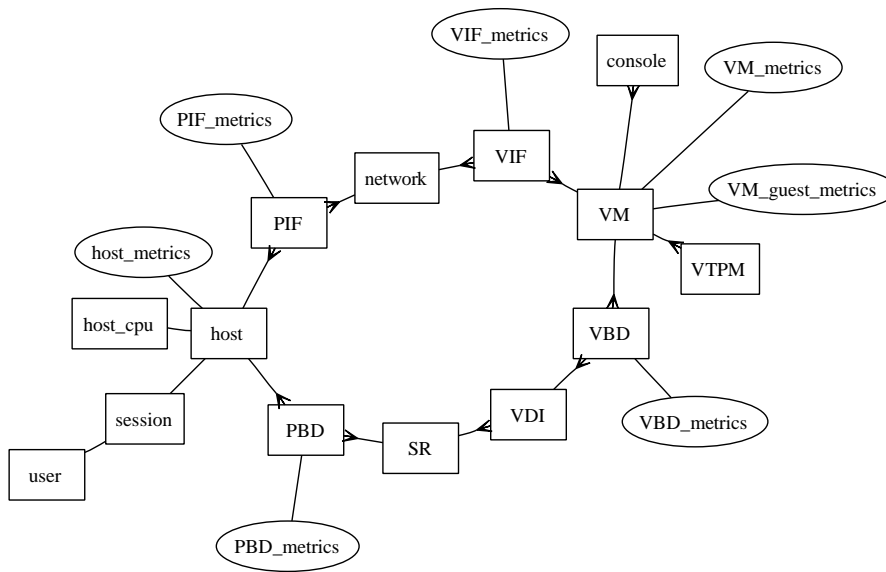
## 2.2 Relationships Between Classes

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



<i>object.field</i>	<i>object.field</i>	<i>relationship</i>
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
VM.appliance	VM_appliance.VMs	one-to-many
PGPU.GPU_group	GPU_group.PGPUs	one-to-many
VGPU.GPU_group	GPU_group.VGPUs	one-to-many
VGPU.type	VGPU_type.VGPUs	one-to-many
VGPU.VM	VM.VGPUs	one-to-many
VGPU.resident_on	PGPU.resident_VGPUs	one-to-many
PGPU.supported_VGPU_types	VGPU_type.supported_on_PGPUs	many-to-many
PGPU.enabled_VGPU_types	VGPU_type.enabled_on_PGPUs	many-to-many
GPU_group.supported_VGPU_types	VGPU_type.supported_on_GPU_groups	many-to-many
GPU_group.enabled_VGPU_types	VGPU_type.enabled_on_GPU_groups	many-to-many
PCI.host	host.PCIs	one-to-many
PGPU.host	host.PGPUs	one-to-many
PCI.attached_VMs	VM.attached_PCIs	many-to-many
VDI.metadata_of_pool	pool.metadata_VDIs	one-to-many
SR.introduced_by	DR_task.introduced_SRs	one-to-many

The following represents bound fields (as specified above) diagrammatically, 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	Cloning a VDI
vdi_snapshot	Snapshotting a VDI
pbd_create	Creating a PBD for this SR

<code>pbddestroy</code>	Destroying one of this SR's PBDs
-------------------------	----------------------------------

<code>enum vif_operations</code>	
<code>attach</code>	Attempting to attach this VIF to a VM
<code>plug</code>	Attempting to hotplug this VIF
<code>unplug</code>	Attempting to hot unplug this VIF

<code>enum network_operations</code>	
<code>attaching</code>	Indicates this network is attaching to a VIF or PIF

<code>enum host_allowed_operations</code>	
<code>provision</code>	Indicates this host is able to provision another VM
<code>evacuate</code>	Indicates this host is evacuating
<code>shutdown</code>	Indicates this host is in the process of shutting itself down
<code>reboot</code>	Indicates this host is in the process of rebooting
<code>power_on</code>	Indicates this host is in the process of being powered on
<code>vm_start</code>	This host is starting a VM
<code>vm_resume</code>	This host is resuming a VM
<code>vm_migrate</code>	This host is the migration target of a VM

<code>enum vm_appliance_operation</code>	
<code>start</code>	Start
<code>clean_shutdown</code>	Clean shutdown
<code>hard_shutdown</code>	Hard shutdown
<code>shutdown</code>	Shutdown

<code>enum vm_power_state</code>	
<code>Halted</code>	VM is offline and not using any resources
<code>Paused</code>	All resources have been allocated but the VM itself is paused and its vCPUs are not running
<code>Running</code>	Running
<code>Suspended</code>	VM state has been saved to disk and it is no longer running. Note that disks remain in-use while

<code>enum after_apply_guidance</code>
--

<code>restartHVM</code>	This patch requires HVM guests to be restarted once applied.
<code>restartPV</code>	This patch requires PV guests to be restarted once applied.
<code>restartHost</code>	This patch requires the host to be restarted once applied.
<code>restartXAPI</code>	This patch requires XAPI to be restarted once applied.

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

enum task_allowed_operations	
<code>cancel</code>	refers to the operation “cancel”
<code>destroy</code>	refers to the operation “destroy”

enum on_normal_exit	
<code>destroy</code>	destroy the VM state
<code>restart</code>	restart the VM

enum on_crash_behaviour	
<code>destroy</code>	destroy the VM state
<code>coredump_and_destroy</code>	record a coredump and then destroy the VM state
<code>restart</code>	restart the VM
<code>coredump_and_restart</code>	record a coredump and then restart the VM
<code>preserve</code>	leave the crashed VM paused
<code>rename_restart</code>	rename the crashed VM and start a new copy

enum vm_operations	
<code>snapshot</code>	refers to the operation “snapshot”
<code>clone</code>	refers to the operation “clone”
<code>copy</code>	refers to the operation “copy”
<code>create_template</code>	refers to the operation “create_template”
<code>revert</code>	refers to the operation “revert”
<code>checkpoint</code>	refers to the operation “checkpoint”
<code>snapshot_with_quiesce</code>	refers to the operation “snapshot_with_quiesce”
<code>provision</code>	refers to the operation “provision”

<code>start</code>	refers to the operation “start”
<code>start_on</code>	refers to the operation “start_on”
<code>pause</code>	refers to the operation “pause”
<code>unpause</code>	refers to the operation “unpause”
<code>clean_shutdown</code>	refers to the operation “clean_shutdown”
<code>clean_reboot</code>	refers to the operation “clean_reboot”
<code>hard_shutdown</code>	refers to the operation “hard_shutdown”
<code>power_state_reset</code>	refers to the operation “power_state_reset”
<code>hard_reboot</code>	refers to the operation “hard_reboot”
<code>suspend</code>	refers to the operation “suspend”
<code>csvm</code>	refers to the operation “csvm”
<code>resume</code>	refers to the operation “resume”
<code>resume_on</code>	refers to the operation “resume_on”
<code>pool_migrate</code>	refers to the operation “pool_migrate”
<code>migrate_send</code>	refers to the operation “migrate_send”
<code>get_boot_record</code>	refers to the operation “get_boot_record”
<code>send_sysrq</code>	refers to the operation “send_sysrq”
<code>send_trigger</code>	refers to the operation “send_trigger”
<code>query_services</code>	refers to the operation “query_services”
<code>shutdown</code>	refers to the operation “shutdown”
<code>changing_memory_live</code>	Changing the memory settings
<code>awaiting_memory_live</code>	Waiting for the memory settings to change
<code>changing_dynamic_range</code>	Changing the memory dynamic range
<code>changing_static_range</code>	Changing the memory static range
<code>changing_memory_limits</code>	Changing the memory limits
<code>changing_shadow_memory</code>	Changing the shadow memory for a halted VM.
<code>changing_shadow_memory_live</code>	Changing the shadow memory for a running VM.
<code>changing_VCPUs</code>	Changing VCPU settings for a halted VM.
<code>changing_VCPUs_live</code>	Changing VCPU settings for a running VM.
<code>assert_operation_valid</code>	
<code>data_source_op</code>	Add, remove, query or list data sources
<code>update_allowed_operations</code>	
<code>make_into_template</code>	Turning this VM into a template
<code>import</code>	importing a VM from a network stream
<code>export</code>	exporting a VM to a network stream
<code>metadata_export</code>	exporting VM metadata to a network stream
<code>reverting</code>	Reverting the VM to a previous snapshotted state
<code>destroy</code>	refers to the act of uninstalling the VM

enum vmpp_backup_frequency	
<code>hourly</code>	Hourly backups
<code>daily</code>	Daily backups
<code>weekly</code>	Weekly backups

enum vmpp_archive_frequency	
<code>never</code>	Never archive
<code>always_after_backup</code>	Archive after backup

<code>daily</code>	Daily archives
<code>weekly</code>	Weekly backups

enum <code>vmpp_archive_target_type</code>	
<code>none</code>	No target config
<code>cifs</code>	CIFS target config
<code>nfs</code>	NFS target config

enum <code>vmpp_backup_type</code>	
<code>snapshot</code>	The backup is a snapshot
<code>checkpoint</code>	The backup is a checkpoint

enum <code>network_default_locking_mode</code>	
<code>unlocked</code>	Treat all VIFs on this network with <code>locking_mode = 'default'</code> as if they have lock
<code>disabled</code>	Treat all VIFs on this network with <code>locking_mode = 'default'</code> as if they have lock

enum <code>vif_locking_mode</code>	
<code>network_default</code>	No specific configuration set - default network policy applies
<code>locked</code>	Only traffic to a specific MAC and a list of IPv4 or IPv6 addresses is permitted
<code>unlocked</code>	All traffic is permitted
<code>disabled</code>	No traffic is permitted

enum <code>ip_configuration_mode</code>	
<code>None</code>	Do not acquire an IP address
<code>DHCP</code>	Acquire an IP address by DHCP
<code>Static</code>	Static IP address configuration

enum <code>ipv6_configuration_mode</code>	
<code>None</code>	Do not acquire an IPv6 address
<code>DHCP</code>	Acquire an IPv6 address by DHCP
<code>Static</code>	Static IPv6 address configuration
<code>Autoconf</code>	Router assigned prefix delegation IPv6 allocation

enum primary_address_type	
IPv4	Primary address is the IPv4 address
IPv6	Primary address is the IPv6 address

enum bond_mode	
balance-slb	Source-level balancing
active-backup	Active/passive bonding: only one NIC is carrying traffic
lacp	Link aggregation control protocol

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 when the VM was last powered off.
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

enum allocation_algorithm	
breadth_first	vGPUs of a given type are allocated evenly across supporting pGPUs.
depth_first	vGPUs of a given type are allocated on supporting pGPUs until they are full.

## 2.4 Class: session

### 2.4.1 Fields for class: session

Name	session		
Description	<i>A session.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	<code>uuid</code>	string	Unique identifier/object reference
<i>RO<sub>run</sub></i>	<code>this_host</code>	host ref	Currently connected host
<i>RO<sub>run</sub></i>	<code>this_user</code>	user ref	Currently connected user
<i>RO<sub>run</sub></i>	<code>last_active</code>	datetime	Timestamp for last time session was active
<i>RO<sub>run</sub></i>	<code>pool</code>	bool	True if this session relates to a intra-pool login, false otherwise
<i>RW</i>	<code>other_config</code>	(string $\rightarrow$ string) Map	additional configuration
<i>RO<sub>run</sub></i>	<code>is_local_superuser</code>	bool	true iff this session was created using local superuser credentials
<i>RO<sub>run</sub></i>	<code>subject</code>	subject ref	references the subject instance that created the session. If a session instance has <code>is_local_superuser</code> set, then the value of this field is undefined.
<i>RO<sub>run</sub></i>	<code>validation_time</code>	datetime	time when session was last validated
<i>RO<sub>run</sub></i>	<code>auth_user_sid</code>	string	the subject identifier of the user that was externally authenticated. If a session instance has <code>is_local_superuser</code> set, then the value of this field is undefined.
<i>RO<sub>run</sub></i>	<code>auth_user_name</code>	string	the subject name of the user that was externally authenticated. If a session instance has <code>is_local_superuser</code> set, then the value of this field is undefined.
<i>RO<sub>ins</sub></i>	<code>rbac_permissions</code>	string Set	list with all RBAC permissions for this session
<i>RO<sub>run</sub></i>	<code>tasks</code>	(task ref) Set	list of tasks created using the current session
<i>RO<sub>ins</sub></i>	<code>parent</code>	session ref	references the parent session that created this session
<i>RO<sub>run</sub></i>	<code>originator</code>	string	a key string provided by a API user to distinguish itself from other users sharing the same login name

### 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, string originator)

**Arguments:**

type	name	description
string	uname	Username for login.
string	pwd	Password for login.
string	version	Client API version.
string	originator	Key string for distinguishing different API users sharing the same login name.

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

**Possible Error Codes:** SESSION\_AUTHENTICATION\_FAILED, HOST\_IS\_SLAVE

**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 privileges 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_originator`

**Overview:**

Get the originator field of the given session.

**Signature:**

```
string get_originator (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_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 <code>subject_name</code> , 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 <code>subject_identifier</code> , 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	<b>subject</b>		
Description	<i>A user or group that can log in xapi.</i>		
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 $\rightarrow$ 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	<b>role</b>		
Description	<i>A set of permissions associated with a subject.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	uuid	string	Unique identifier/object reference
<i>RO<sub>ins</sub></i>	name/label	string	a short user-friendly name for the role
<i>RO<sub>ins</sub></i>	name/description	string	what this role is for
<i>RO<sub>ins</sub></i>	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  $\rightarrow$  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	<b>task</b>		
Description	<i>A long-running asynchronous task.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	<b>uuid</b>	string	Unique identifier/object reference
<i>RO<sub>run</sub></i>	<b>name/label</b>	string	a human-readable name
<i>RO<sub>run</sub></i>	<b>name/description</b>	string	a notes field containing human-readable description
<i>RO<sub>run</sub></i>	<b>allowed_operations</b>	(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.
<i>RO<sub>run</sub></i>	<b>current_operations</b>	(string $\rightarrow$ 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.
<i>RO<sub>run</sub></i>	<b>created</b>	datetime	Time task was created
<i>RO<sub>run</sub></i>	<b>finished</b>	datetime	Time task finished (i.e. succeeded or failed). If task-status is pending, then the value of this field has no meaning
<i>RO<sub>run</sub></i>	<b>status</b>	task_status_type	current status of the task
<i>RO<sub>run</sub></i>	<b>session</b>	session ref	the session that created the task
<i>RO<sub>run</sub></i>	<b>resident_on</b>	host ref	the host on which the task is running
<i>RO<sub>run</sub></i>	<b>progress</b>	float	This field contains the estimated fraction of the task which is complete. This field should not be used to determine whether the task is complete - for this the status field of the task should be used.
<i>RO<sub>run</sub></i>	<b>externalpid</b>	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 )
<i>RO<sub>run</sub></i>	<b>stunnelpid</b>	int	If the task has been forwarded, this field records the pid of the stunnel process spawned to manage the forwarding connection
<i>RO<sub>run</sub></i>	<b>forwarded</b>	bool	True if this task has been forwarded to a slave
<i>RO<sub>run</sub></i>	<b>forwarded_to</b>	host ref	The host to which the task has been forwarded
<i>RO<sub>run</sub></i>	<b>type</b>	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.



$RO_{run}$	<b>result</b>	string	if the task has completed successfully, this field contains the result value (either Void or an object reference). Undefined otherwise.
$RO_{run}$	<b>error_info</b>	string Set	if the task has failed, this field contains the set of associated error strings. Undefined otherwise.
$RW$	<b>other_config</b>	(string $\rightarrow$ string) Map	additional configuration
$RO_{run}$	<b>subtask_of</b>	task ref	Ref pointing to the task this is a subtask of.
$RO_{run}$	<b>subtasks</b>	(task ref) Set	List pointing to all the subtasks.

## 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  $\rightarrow$  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
<code>(string → string) Map</code>	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	<b>event</b>		
Description	<i>Asynchronous event registration and handling.</i>		
Quals	Field	Type	Description
<i>RO<sub>ins</sub></i>	id	int	An ID, monotonically increasing, and local to the current session
<i>RO<sub>ins</sub></i>	timestamp	datetime	The time at which the event occurred
<i>RO<sub>ins</sub></i>	class	string	The name of the class of the object that changed
<i>RO<sub>ins</sub></i>	operation	event_operation	The operation that was performed
<i>RO<sub>ins</sub></i>	ref	string	A reference to the object that changed
<i>RO<sub>ins</sub></i>	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 \* as the desired class 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:** from**Overview:**

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

**Signature:**

```
((event record) Set) from (session_id s, string Set classes, string token, float timeout)
```

**Arguments:**

type	name	description
string Set	classes	register for events for the indicated classes
string	token	A token representing the point from which to generate database events. The empty string represents the beginning.
float	timeout	Return after this many seconds if no events match

**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

**RPC name:** inject**Overview:**

Injects an artificial event on the given object and return the corresponding ID.

**Signature:**

```
string inject (session_id s, string class, string ref)
```

**Arguments:**

type	name	description
string	class	class of the object
string	ref	A reference to the object that will be changed.

**Return Type:** string  
the event ID

## 2.10 Class: pool

### 2.10.1 Fields for class: pool

Name	<b>pool</b>		
Description	<i>Pool-wide information.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	<code>uuid</code>	string	Unique identifier/object reference
<i>RW</i>	<code>name_label</code>	string	Short name
<i>RW</i>	<code>name_description</code>	string	Description
<i>RO<sub>run</sub></i>	<code>master</code>	host ref	The host that is pool master
<i>RW</i>	<code>default_SR</code>	SR ref	Default SR for VDIs
<i>RW</i>	<code>suspend_image_SR</code>	SR ref	The SR in which VDIs for suspend images are created
<i>RW</i>	<code>crash_dump_SR</code>	SR ref	The SR in which VDIs for crash dumps are created
<i>RW</i>	<code>other_config</code>	(string → string) Map	additional configuration
<i>RO<sub>run</sub></i>	<code>ha_enabled</code>	bool	true if HA is enabled on the pool, false otherwise
<i>RO<sub>run</sub></i>	<code>ha_configuration</code>	(string → string) Map	The current HA configuration
<i>RO<sub>run</sub></i>	<code>ha_statefiles</code>	string Set	HA statefile VDIs in use
<i>RO<sub>run</sub></i>	<code>ha_host_failures_to_tolerate</code>	int	Number of host failures to tolerate before the Pool is declared to be over-committed
<i>RO<sub>run</sub></i>	<code>ha_plan_exists_for</code>	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.
<i>RW</i>	<code>ha_allow_overcommit</code>	bool	If set to false then operations which would cause the Pool to become over-committed will be blocked.
<i>RO<sub>run</sub></i>	<code>ha_overcommitted</code>	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
<i>RO<sub>run</sub></i>	<code>blobs</code>	(string → blob ref) Map	Binary blobs associated with this pool
<i>RW</i>	<code>tags</code>	string Set	user-specified tags for categorization purposes
<i>RW</i>	<code>gui_config</code>	(string → string) Map	gui-specific configuration for pool
<i>RO<sub>run</sub></i>	<code>wlb_url</code>	string	Url for the configured workload balancing host
<i>RO<sub>run</sub></i>	<code>wlb_username</code>	string	Username for accessing the workload balancing host
<i>RO<sub>run</sub></i>	<code>wlb_password</code>	secret ref	Password for accessing the workload balancing host
<i>RW</i>	<code>wlb_enabled</code>	bool	true if workload balancing is enabled on the pool, false otherwise
<i>RW</i>	<code>wlb_verify_cert</code>	bool	true if communication with the WLB server should enforce SSL certificate verification.

<i>RO<sub>run</sub></i>	<code>redo_log_enabled</code>	bool	true a redo-log is to be used other than when HA is enabled, false otherwise
<i>RO<sub>run</sub></i>	<code>redo_log_vdi</code>	VDI ref	indicates the VDI to use for the redo-log other than when HA is enabled
<i>RO<sub>run</sub></i>	<code>vswitch_controller</code>	string	address of the vswitch controller
<i>RO<sub>run</sub></i>	<code>restrictions</code>	(string $\rightarrow$ string) Map	Pool-wide restrictions currently in effect
<i>RO<sub>run</sub></i>	<code>metadata_VDIs</code>	(VDI ref) Set	The set of currently known metadata VDIs for this pool

### 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 fail)
```

**Arguments:**

type	name	description
(host ref) Set	<code>failed_hosts</code>	The set of hosts to assume have failed
(VM ref) Set	<code>failed_vms</code>	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	<code>self</code>	The pool
int	<code>value</code>	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, bool public)
```

**Arguments:**

type	name	description
pool ref	<code>pool</code>	The pool
string	<code>name</code>	The name associated with the blob
string	<code>mime_type</code>	The mime type for the data. Empty string translates to application/octet-stream
bool	<code>public</code>	True if the blob should be publicly available

**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 xenserver_username, string xenserver_password)
```

**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 username 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\_license\_state

**Overview:**

This call returns the license state for the pool.

**Signature:**

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

**Arguments:**

type	name	description
pool ref	self	Reference to the pool

**Return Type:** (string → string) Map

The pool's license state

**RPC name:** `apply_edition`**Overview:**

Apply an edition to all hosts in the pool.

**Signature:**

```
void apply_edition (session_id s, pool ref self, string edition)
```

**Arguments:**

type	name	description
pool ref	self	Reference to the pool
string	edition	The requested edition

**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  $\rightarrow$  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_metadata_VDIIs`

**Overview:**

Get the `metadata_VDIIs` field of the given pool.

**Signature:**

```
((VDI ref) Set) get_metadata_VDIIs (session_id s, pool ref self)
```

**Arguments:**

type	name	description
pool ref	self	reference to the object

**Return Type:** (VDI ref) Set  
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	<b>pool_patch</b>		
Description	<i>Pool-wide patches.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	uuid	string	Unique identifier/object reference
<i>RO<sub>ins</sub></i>	name/label	string	a human-readable name
<i>RO<sub>ins</sub></i>	name/description	string	a notes field containing human-readable description
<i>RO<sub>ins</sub></i>	version	string	Patch version number
<i>RO<sub>run</sub></i>	filename	string	Filename of the patch
<i>RO<sub>run</sub></i>	size	int	Size of the patch
<i>RO<sub>run</sub></i>	pool_applied	bool	This patch should be applied across the entire pool
<i>RO<sub>run</sub></i>	host_patches	(host_patch ref) Set	This hosts this patch is applied to.
<i>RO<sub>run</sub></i>	after_apply_guidance	(after_apply_guidance) Set	What the client should do after this patch has been applied.
<i>RW</i>	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 the server.

**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:** pool\_clean**Overview:**

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

**Signature:**

```
void pool_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:** clean\_on\_host

**Overview:**

Removes the patch's files from the specified host.

**Signature:**

```
void clean_on_host (session_id s, pool_patch ref self, host ref host)
```

**Arguments:**

type	name	description
pool_patch ref	self	The patch to clean up
host ref	host	The host on which to clean the patch

**Return Type:** void

**RPC name:** get\_all

**Overview:**

Return a list of all the pool\_patches 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\_patches 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
<code>pool_patch ref</code>	<code>self</code>	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
<code>pool_patch ref</code>	<code>self</code>	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
<code>pool_patch ref</code>	<code>self</code>	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
<code>pool_patch ref</code>	<code>self</code>	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
<code>pool_patch ref</code>	<code>self</code>	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
<code>pool_patch ref</code>	<code>self</code>	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
<code>pool_patch ref</code>	<code>self</code>	reference to the object
<code>(string → string) Map</code>	<code>value</code>	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
<code>pool_patch ref</code>	<code>self</code>	reference to the object
<code>string</code>	<code>key</code>	Key to add
<code>string</code>	<code>value</code>	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
<code>pool_patch ref</code>	<code>self</code>	reference to the object
<code>string</code>	<code>key</code>	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	<i>A virtual machine (or 'guest').</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	<code>uuid</code>	string	Unique identifier/object reference
<i>RO<sub>run</sub></i>	<code>allowed_operations</code>	(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.
<i>RO<sub>run</sub></i>	<code>current_operations</code>	(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.
<i>RO<sub>run</sub></i>	<code>power_state</code>	vm_power_state	Current power state of the machine
<i>RW</i>	<code>name/label</code>	string	a human-readable name
<i>RW</i>	<code>name/description</code>	string	a notes field containing human-readable description
<i>RW</i>	<code>user_version</code>	int	a user version number for this machine
<i>RW</i>	<code>is_a_template</code>	bool	true if this is a template. Template VMs can never be started, they are used only for cloning other VMs
<i>RO<sub>run</sub></i>	<code>suspend_VDI</code>	VDI ref	The VDI that a suspend image is stored on. (Only has meaning if VM is currently suspended)
<i>RO<sub>run</sub></i>	<code>resident_on</code>	host ref	the host the VM is currently resident on
<i>RO<sub>run</sub></i>	<code>scheduled_to_be_resident_on</code>	host ref	the host on which the VM is due to be started/resumed/migrated. This acts as a memory reservation indicator
<i>RW</i>	<code>affinity</code>	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.
<i>RO<sub>run</sub></i>	<code>memory/overhead</code>	int	Virtualization memory overhead (bytes).
<i>RO<sub>ins</sub></i>	<code>memory/target</code>	int	Dynamically-set memory target (bytes). The value of this field indicates the current target for memory available to this VM.
<i>RO<sub>ins</sub></i>	<code>memory/static_max</code>	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.
<i>RO<sub>ins</sub></i>	<code>memory/dynamic_max</code>	int	Dynamic maximum (bytes)
<i>RO<sub>ins</sub></i>	<code>memory/dynamic_min</code>	int	Dynamic minimum (bytes)

<i>RO<sub>ins</sub></i>	memory/static_min	int	Statically-set (i.e. absolute) minimum (bytes). The value of this field indicates the least amount of memory this VM can boot with without crashing.
<i>RW</i>	VCPUs/params	(string → string) Map	configuration parameters for the selected VCPU policy
<i>RO<sub>ins</sub></i>	VCPUs/max	int	Max number of VCPUs
<i>RO<sub>ins</sub></i>	VCPUs/at_startup	int	Boot number of VCPUs
<i>RW</i>	actions/after_shutdown	on_normal_exit	action to take after the guest has shutdown itself
<i>RW</i>	actions/after_reboot	on_normal_exit	action to take after the guest has rebooted itself
<i>RW</i>	actions/after_crash	on_crash_behaviour	action to take if the guest crashes
<i>RO<sub>run</sub></i>	consoles	(console ref) Set	virtual console devices
<i>RO<sub>run</sub></i>	VIFs	(VIF ref) Set	virtual network interfaces
<i>RO<sub>run</sub></i>	VBDs	(VBD ref) Set	virtual block devices
<i>RO<sub>run</sub></i>	crash_dumps	(crashdump ref) Set	crash dumps associated with this VM
<i>RO<sub>run</sub></i>	VTPMs	(VTPM ref) Set	virtual TPMs
<i>RW</i>	PV/bootloader	string	name of or path to bootloader
<i>RW</i>	PV/kernel	string	path to the kernel
<i>RW</i>	PV/ramdisk	string	path to the initrd
<i>RW</i>	PV/args	string	kernel command-line arguments
<i>RW</i>	PV/bootloader_args	string	miscellaneous arguments for the bootloader
<i>RW</i>	PV/legacy_args	string	to make Zurich guests boot
<i>RW</i>	HVM/boot_policy	string	HVM boot policy
<i>RW</i>	HVM/boot_params	(string → string) Map	HVM boot params
<i>RO<sub>ins</sub></i>	HVM/shadow_multiplier	float	multiplier applied to the amount of shadow that will be made available to the guest
<i>RW</i>	platform	(string → string) Map	platform-specific configuration
<i>RW</i>	PCI_bus	string	PCI bus path for pass-through devices
<i>RW</i>	other_config	(string → string) Map	additional configuration
<i>RO<sub>run</sub></i>	domid	int	domain ID (if available, -1 otherwise)
<i>RO<sub>run</sub></i>	domarch	string	Domain architecture (if available, null string otherwise)
<i>RO<sub>run</sub></i>	last_boot_CPU_flags	(string → string) Map	describes the CPU flags on which the VM was last booted
<i>RO<sub>run</sub></i>	is_control_domain	bool	true if this is a control domain (domain 0 or a driver domain)
<i>RO<sub>run</sub></i>	metrics	VM_metrics ref	metrics associated with this VM
<i>RO<sub>run</sub></i>	guest_metrics	VM_guest_metrics ref	metrics associated with the running guest
<i>RO<sub>run</sub></i>	last_booted_record	string	marshalled value containing VM record at time of last boot, updated dynamically to reflect the runtime state of the domain
<i>RW</i>	recommendations	string	An XML specification of recommended values and ranges for properties of this VM

<i>RW</i>	<b>xenstore_data</b>	(string $\rightarrow$ string) Map	data to be inserted into the xenstore tree (/local/domain/ <i>i</i> /vm-data) after the VM is created.
<i>RO<sub>ins</sub></i>	<b>ha_always_run</b>	bool	if true then the system will attempt to keep the VM running as much as possible.
<i>RO<sub>ins</sub></i>	<b>ha_restart_priority</b>	string	has 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”; “restart” meaning “this VM should be restarted”; “” meaning “do not try to restart this VM”
<i>RO<sub>run</sub></i>	<b>is_a_snapshot</b>	bool	true if this is a snapshot. Snapshotted VMs can never be started, they are used only for cloning other VMs
<i>RO<sub>run</sub></i>	<b>snapshot_of</b>	VM ref	Ref pointing to the VM this snapshot is of.
<i>RO<sub>run</sub></i>	<b>snapshots</b>	(VM ref) Set	List pointing to all the VM snapshots.
<i>RO<sub>run</sub></i>	<b>snapshot_time</b>	datetime	Date/time when this snapshot was created.
<i>RO<sub>run</sub></i>	<b>transportable_snapshot_id</b>	string	Transportable ID of the snapshot VM
<i>RO<sub>run</sub></i>	<b>blobs</b>	(string $\rightarrow$ blob ref) Map	Binary blobs associated with this VM
<i>RW</i>	<b>tags</b>	string Set	user-specified tags for categorization purposes
<i>RW</i>	<b>blocked_operations</b>	(vm_operations $\rightarrow$ string) Map	List of operations which have been explicitly blocked and an error code
<i>RO<sub>run</sub></i>	<b>snapshot_info</b>	(string $\rightarrow$ string) Map	Human-readable information concerning this snapshot
<i>RO<sub>run</sub></i>	<b>snapshot_metadata</b>	string	Encoded information about the VM’s metadata this is a snapshot of
<i>RO<sub>run</sub></i>	<b>parent</b>	VM ref	Ref pointing to the parent of this VM
<i>RO<sub>run</sub></i>	<b>children</b>	(VM ref) Set	List pointing to all the children of this VM
<i>RO<sub>run</sub></i>	<b>bios_strings</b>	(string $\rightarrow$ string) Map	BIOS strings
<i>RO<sub>ins</sub></i>	<b>protection_policy</b>	VMPP ref	Ref pointing to a protection policy for this VM
<i>RO<sub>ins</sub></i>	<b>is_snapshot_from_vmpp</b>	bool	true if this snapshot was created by the protection policy
<i>RO<sub>ins</sub></i>	<b>appliance</b>	VM_appliance ref	the appliance to which this VM belongs
<i>RO<sub>ins</sub></i>	<b>start_delay</b>	int	The delay to wait before proceeding to the next order in the startup sequence (seconds)
<i>RO<sub>ins</sub></i>	<b>shutdown_delay</b>	int	The delay to wait before proceeding to the next order in the shutdown sequence (seconds)
<i>RO<sub>ins</sub></i>	<b>order</b>	int	The point in the startup or shutdown sequence at which this VM will be started
<i>RO<sub>run</sub></i>	<b>VGPUs</b>	(VGPU ref) Set	Virtual GPUs
<i>RO<sub>run</sub></i>	<b>attached_PCIs</b>	(PCI ref) Set	Currently passed-through PCI devices

<i>RW</i>	<code>suspend_SR</code>	SR ref	The SR on which a suspend image is stored
<i>RO<sub>ins</sub></i>	<code>version</code>	int	The number of times this VM has been recovered
<i>RO<sub>ins</sub></i>	<code>generation_id</code>	string	Generation ID of the VM

### 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\_PLUGIN\_FAILED

**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:**

Checkpoint 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:** shutdown

**Overview:**

Attempts to first clean shutdown a VM and if it should fail then perform a hard shutdown on it.

**Signature:**

```
void 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

**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:** This message is deprecated 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, int dynamic_max)
```

**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:** This message is deprecated 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: migrate\_send****Overview:**

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

**Signature:**

```
void migrate_send (session_id s, VM ref vm, (string -> string) Map dest, bool live, (VDI ref -> SR ref
```

**Arguments:**

type	name	description
VM ref	vm	The VM
(string → string) Map	dest	The result of a Host.migrate_receive call.
bool	live	Live migration
(VDI ref → SR ref) Map	vdi_map	Map of source VDI to destination SR
(VIF ref → network ref) Map	vif_map	Map of source VIF to destination network
(string → string) Map	options	Other parameters

**Return Type:** void

**Possible Error Codes:** VM\_BAD\_POWER\_STATE

**RPC name: assert\_can\_migrate****Overview:**

Assert whether a VM can be migrated to the specified destination.

**Signature:**

```
void assert_can_migrate (session_id s, VM ref vm, (string -> string) Map dest, bool live, (VDI ref -> SR ref
```

**Arguments:**

type	name	description
VM ref	vm	The VM
(string → string) Map	dest	The result of a VM.migrate_receive call.
bool	live	Live migration
(VDI ref → SR ref) Map	vdi_map	Map of source VDI to destination SR
(VIF ref → network ref) Map	vif_map	Map of source VIF to destination network
(string → string) Map	options	Other parameters

**Return Type:** void

**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`, `VM_HOST_INCOMPATIBLE_VERSION`

**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, bool public)
```

**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
bool	public	True if the blob should be publicly available

**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  $\rightarrow$  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:** `set_start_delay`

**Overview:**

Set this VM's start delay in seconds.

**Signature:**

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

**Arguments:**

type	name	description
VM ref	self	The VM
int	value	This VM's start delay in seconds

**Return Type:** void

**RPC name:** `set_shutdown_delay`

**Overview:**

Set this VM's shutdown delay in seconds.

**Signature:**

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

**Arguments:**

type	name	description
VM ref	self	The VM
int	value	This VM's shutdown delay in seconds

**Return Type:** void

**RPC name:** `set_order`

**Overview:**

Set this VM's boot order.

**Signature:**

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

**Arguments:**

type	name	description
VM ref	self	The VM
int	value	This VM's boot order

**Return Type:** void

**RPC name:** set\_suspend\_VDI**Overview:**

Set this VM's suspend VDI, which must be identical to its current one.

**Signature:**

```
void set_suspend_VDI (session_id s, VM ref self, VDI ref value)
```

**Arguments:**

type	name	description
VM ref	self	The VM
VDI ref	value	The suspend VDI uuid

**Return Type:** void

**RPC name:** assert\_can\_be\_recovered**Overview:**

Assert whether all SRs required to recover this VM are available.

**Signature:**

```
void assert_can_be_recovered (session_id s, VM ref self, session ref session_to)
```

**Arguments:**

type	name	description
VM ref	self	The VM to recover
session ref	session_to	The session to which the VM is to be recovered.

**Return Type:** void

**Possible Error Codes:** VM\_IS\_PART\_OF\_AN\_APPLIANCE, VM\_REQUIRES\_SR

**RPC name:** get\_SRs\_required\_for\_recovery**Overview:**

List all the SR's that are required for the VM to be recovered.

**Signature:**

```
((SR ref) Set) get_SRs_required_for_recovery (session_id s, VM ref self, session ref session_to)
```

**Arguments:**

type	name	description
VM ref	self	The VM for which the SRs have to be recovered
session ref	session_to	The session to which the SRs of the VM have to be recovered.

**Return Type:** (SR ref) Set

refs for SRs required to recover the VM

**RPC name:** recover

**Overview:**

Recover the VM.

**Signature:**

```
void recover (session_id s, VM ref self, session ref session_to, bool force)
```

**Arguments:**

type	name	description
VM ref	self	The VM to recover
session ref	session_to	The session to which the VM is to be recovered.
bool	force	Whether the VM should replace newer versions of itself.

**Return Type:** void

**RPC name:** import\_convert

**Overview:**

Import using a conversion service.

**Signature:**

```
void import_convert (session_id s, string type, string username, string password, SR ref sr, (string -
```

**Arguments:**

type	name	description
string	type	Type of the conversion
string	username	Admin username on the host
string	password	Password on the host
SR ref	sr	The destination SR
(string → string) Map	remote_config	Remote configuration options

**Return Type:** void

**RPC name:** set\_appliance

**Overview:**

Assign this VM to an appliance.

**Signature:**

```
void set_appliance (session_id s, VM ref self, VM_appliance ref value)
```

**Arguments:**

type	name	description
VM ref	self	The VM to assign to an appliance.
VM_appliance ref	value	The appliance to which this VM should be assigned.

**Return Type:** void

**RPC name:** `query_services`

**Overview:**

Query the system services advertised by this VM and register them. This can only be applied to a system domain.

**Signature:**

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

**Arguments:**

type	name	description
VM ref	self	The VM

**Return Type:** `(string → string) Map`

map of service type to name

**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
<code>(string → string) Map</code>	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:** This message is deprecated 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:** This message is deprecated Set the PCIbus 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 $\rightarrow$ 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:** This message is deprecated 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  $\rightarrow$  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:** This message is deprecated 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:** This message is deprecated 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:** get\_appliance**Overview:**

Get the appliance field of the given VM.

**Signature:**

```
(VM_appliance ref) get_appliance (session_id s, VM ref self)
```

**Arguments:**

type	name	description
VM ref	self	reference to the object

**Return Type:** VM\_appliance ref

value of the field

**RPC name:** get\_start\_delay**Overview:**

Get the start\_delay field of the given VM.

**Signature:**

```
int get_start_delay (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\_shutdown\_delay**Overview:**

Get the shutdown\_delay field of the given VM.

**Signature:**

```
int get_shutdown_delay (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\_order

**Overview:**  
Get the order field of the given VM.  
**Signature:**

```
int get_order (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\_VGPUs

**Overview:**  
Get the VGPUs field of the given VM.  
**Signature:**

```
((VGPU ref) Set) get_VGPUs (session_id s, VM ref self)
```

**Arguments:**

type	name	description
VM ref	self	reference to the object

**Return Type:** (VGPU ref) Set  
value of the field

**RPC name:** get\_attached\_PCIs

**Overview:**  
Get the attached\_PCIs field of the given VM.  
**Signature:**

```
((PCI ref) Set) get_attached_PCIs (session_id s, VM ref self)
```

**Arguments:**

type	name	description
VM ref	self	reference to the object

**Return Type:** (PCI ref) Set  
value of the field

**RPC name:** `get_suspend_SR`

**Overview:**

Get the `suspend_SR` field of the given VM.

**Signature:**

```
(SR ref) get_suspend_SR (session_id s, VM ref self)
```

**Arguments:**

type	name	description
VM ref	self	reference to the object

**Return Type:** SR ref

value of the field

**RPC name:** `set_suspend_SR`

**Overview:**

Set the `suspend_SR` field of the given VM.

**Signature:**

```
void set_suspend_SR (session_id s, VM ref self, SR ref value)
```

**Arguments:**

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

**Return Type:** void

**RPC name:** `get_version`

**Overview:**

Get the `version` field of the given VM.

**Signature:**

```
int get_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:** `get_generation_id`

**Overview:**

Get the `generation_id` field of the given VM.

**Signature:**

```
string get_generation_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:** `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	<i>The metrics associated with a VM.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	uuid	string	Unique identifier/object reference
<i>RO<sub>run</sub></i>	memory/actual	int	Guest's actual memory (bytes)
<i>RO<sub>run</sub></i>	VCPUs/number	int	Current number of VCPUs
<i>RO<sub>run</sub></i>	VCPUs/utilisation	(int → float) Map	Utilisation for all of guest's current VCPUs
<i>RO<sub>run</sub></i>	VCPUs/CPU	(int → int) Map	VCPU to PCPU map
<i>RO<sub>run</sub></i>	VCPUs/params	(string → string) Map	The live equivalent to VM.VCPUs_params
<i>RO<sub>run</sub></i>	VCPUs/flags	(int → string Set) Map	CPU flags (blocked,online,running)
<i>RO<sub>run</sub></i>	state	string Set	The state of the guest, eg blocked, dying etc
<i>RO<sub>run</sub></i>	start_time	datetime	Time at which this VM was last booted
<i>RO<sub>run</sub></i>	install_time	datetime	Time at which the VM was installed
<i>RO<sub>run</sub></i>	last_updated	datetime	Time at which this information was last updated
<i>RW</i>	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:**  $(\text{int} \rightarrow \text{float}) \text{ Map}$   
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:**  $(\text{int} \rightarrow \text{int}) \text{ Map}$   
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:**  $(\text{string} \rightarrow \text{string}) \text{ Map}$   
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
<code>VM_metrics ref</code>	<code>self</code>	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
<code>VM_metrics ref</code>	<code>self</code>	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
<code>VM_metrics ref</code>	<code>self</code>	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
<code>VM_metrics ref</code>	<code>self</code>	reference to the object
<code>(string → string) Map</code>	<code>value</code>	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
<code>VM_metrics ref</code>	<code>self</code>	reference to the object
<code>string</code>	<code>key</code>	Key to add
<code>string</code>	<code>value</code>	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
<code>VM_metrics ref</code>	<code>self</code>	reference to the object
<code>string</code>	<code>key</code>	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
<code>VM_metrics ref</code>	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	<i>The metrics reported by the guest (as opposed to inferred from outside).</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	uuid	string	Unique identifier/object reference
<i>RO<sub>run</sub></i>	os_version	(string → string) Map	version of the OS
<i>RO<sub>run</sub></i>	PV_drivers_version	(string → string) Map	version of the PV drivers
<i>RO<sub>run</sub></i>	PV_drivers_up_to_date	bool	true if the PV drivers appear to be up to date
<i>RO<sub>run</sub></i>	memory	(string → string) Map	free/used/total memory
<i>RO<sub>run</sub></i>	disks	(string → string) Map	disk configuration/free space
<i>RO<sub>run</sub></i>	networks	(string → string) Map	network configuration
<i>RO<sub>run</sub></i>	other	(string → string) Map	anything else
<i>RO<sub>run</sub></i>	last_updated	datetime	Time at which this information was last updated
<i>RW</i>	other_config	(string → string) Map	additional configuration
<i>RO<sub>run</sub></i>	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
<code>VM_guest_metrics ref</code>	<code>self</code>	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
<code>VM_guest_metrics ref</code>	<code>self</code>	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
<code>VM_guest_metrics ref</code>	<code>self</code>	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
<code>VM_guest_metrics ref</code>	<code>self</code>	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
<code>VM_guest_metrics ref</code>	<code>self</code>	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
<code>VM_guest_metrics ref</code>	<code>self</code>	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
<code>VM_guest_metrics ref</code>	<code>self</code>	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
<code>VM_guest_metrics ref</code>	<code>self</code>	reference to the object
<code>(string → string) Map</code>	<code>value</code>	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
<code>VM_guest_metrics ref</code>	<code>self</code>	reference to the object
<code>string</code>	<code>key</code>	Key to add
<code>string</code>	<code>value</code>	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
<code>VM_guest_metrics ref</code>	<code>self</code>	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
<i>RO<sub>run</sub></i>	<code>uuid</code>	string	Unique identifier/object reference
<i>RW</i>	<code>name/label</code>	string	a human-readable name
<i>RW</i>	<code>name/description</code>	string	a notes field containing human-readable description
<i>RW</i>	<code>is_policy_enabled</code>	bool	enable or disable this policy
<i>RW</i>	<code>backup_type</code>	vmpp_backup_type	type of the backup sub-policy
<i>RO<sub>ins</sub></i>	<code>backup_retention_value</code>	int	maximum number of backups that should be stored at any time
<i>RO<sub>ins</sub></i>	<code>backup_frequency</code>	vmpp_backup_frequency	frequency of the backup schedule
<i>RO<sub>ins</sub></i>	<code>backup_schedule</code>	(string → string) Map	schedule of the backup containing 'hour', 'min', 'days'. Date/time-related information is in XenServer Local Timezone
<i>RO<sub>run</sub></i>	<code>is_backup_running</code>	bool	true if this protection policy's backup is running
<i>RO<sub>run</sub></i>	<code>backup_last_run_time</code>	datetime	time of the last backup
<i>RO<sub>ins</sub></i>	<code>archive_target_type</code>	vmpp_archive_target_type	type of the archive target config
<i>RO<sub>ins</sub></i>	<code>archive_target_config</code>	(string → string) Map	configuration for the archive, including its 'location', 'username', 'password'
<i>RO<sub>ins</sub></i>	<code>archive_frequency</code>	vmpp_archive_frequency	frequency of the archive schedule
<i>RO<sub>ins</sub></i>	<code>archive_schedule</code>	(string → string) Map	schedule of the archive containing 'hour', 'min', 'days'. Date/time-related information is in XenServer Local Timezone
<i>RO<sub>run</sub></i>	<code>is_archive_running</code>	bool	true if this protection policy's archive is running
<i>RO<sub>run</sub></i>	<code>archive_last_run_time</code>	datetime	time of the last archive
<i>RO<sub>run</sub></i>	<code>VMs</code>	(VM ref) Set	all VMs attached to this protection policy
<i>RO<sub>ins</sub></i>	<code>is_alarm_enabled</code>	bool	true if alarm is enabled for this policy
<i>RO<sub>ins</sub></i>	<code>alarm_config</code>	(string → string) Map	configuration for the alarm
<i>RO<sub>run</sub></i>	<code>recent_alerts</code>	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  $\rightarrow$  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: VM\_appliance

### 2.16.1 Fields for class: VM\_appliance

Name	VM_appliance		
Description	VM appliance.		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	<code>uuid</code>	string	Unique identifier/object reference
<i>RW</i>	<code>name/label</code>	string	a human-readable name
<i>RW</i>	<code>name/description</code>	string	a notes field containing human-readable description
<i>RO<sub>run</sub></i>	<code>allowed_operations</code>	(vm_appliance_operation) 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.
<i>RO<sub>run</sub></i>	<code>current_operations</code>	(string → vm_appliance_operation) Map	links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.
<i>RO<sub>run</sub></i>	<code>VMs</code>	(VM ref) Set	all VMs in this appliance

### 2.16.2 RPCs associated with class: VM\_appliance

**RPC name:** start

**Overview:**

Start all VMs in the appliance.

**Signature:**

```
void start (session_id s, VM_appliance ref self, bool paused)
```

**Arguments:**

type	name	description
VM_appliance ref	self	The VM appliance
bool	paused	Instantiate all VMs belonging to this appliance in paused state if set to true.

**Return Type:** void

**Possible Error Codes:** OPERATION\_PARTIALLY\_FAILED

**RPC name:** clean\_shutdown

**Overview:**

Perform a clean shutdown of all the VMs in the appliance.

**Signature:**

```
void clean_shutdown (session_id s, VM_appliance ref self)
```

**Arguments:**

type	name	description
VM_appliance ref	self	The VM appliance

**Return Type:** void

**Possible Error Codes:** OPERATION\_PARTIALLY\_FAILED

**RPC name:** hard\_shutdown

**Overview:**

Perform a hard shutdown of all the VMs in the appliance.

**Signature:**

```
void hard_shutdown (session_id s, VM_appliance ref self)
```

**Arguments:**

type	name	description
VM_appliance ref	self	The VM appliance

**Return Type:** void

**Possible Error Codes:** OPERATION\_PARTIALLY\_FAILED

**RPC name:** shutdown

**Overview:**

For each VM in the appliance, try to shut it down cleanly. If this fails, perform a hard shutdown of the VM.

**Signature:**

```
void shutdown (session_id s, VM_appliance ref self)
```

**Arguments:**

type	name	description
VM_appliance ref	self	The VM appliance

**Return Type:** void

**Possible Error Codes:** OPERATION\_PARTIALLY\_FAILED

**RPC name:** assert\_can\_be\_recovered

**Overview:**

Assert whether all SRs required to recover this VM appliance are available.

**Signature:**

```
void assert_can_be_recovered (session_id s, VM_appliance ref self, session ref session_to)
```

**Arguments:**

type	name	description
VM_appliance ref	self	The VM appliance to recover
session ref	session_to	The session to which the VM appliance is to be recovered.

**Return Type:** void

**Possible Error Codes:** VM\_REQUIRES\_SR

**RPC name:** `get_SRs_required_for_recovery`

**Overview:**

Get the list of SRs required by the VM appliance to recover.

**Signature:**

```
((SR ref) Set) get_SRs_required_for_recovery (session_id s, VM_appliance ref self, session ref session_to)
```

**Arguments:**

type	name	description
VM_appliance ref	self	The VM appliance for which the required list of SRs has to be recovered.
session ref	session_to	The session to which the list of SRs have to be recovered .

**Return Type:** (SR ref) Set

refs for SRs required to recover the VM

**RPC name:** `recover`

**Overview:**

Recover the VM appliance.

**Signature:**

```
void recover (session_id s, VM_appliance ref self, session ref session_to, bool force)
```

**Arguments:**

type	name	description
VM_appliance ref	self	The VM appliance to recover
session ref	session_to	The session to which the VM appliance is to be recovered.
bool	force	Whether the VMs should replace newer versions of themselves.

**Return Type:** void

**Possible Error Codes:** VM\_REQUIRES\_SR

**RPC name:** `get_all`

**Overview:**

Return a list of all the VM\_appliances known to the system.

**Signature:**

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

**Return Type:** (VM\_appliance ref) Set

references to all objects

**RPC name:** `get_all_records`**Overview:**

Return a map of VM\_appliance references to VM\_appliance records for all VM\_appliances known to the system.

**Signature:**

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

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

**RPC name:** `get_uuid`**Overview:**

Get the uuid field of the given VM\_appliance.

**Signature:**

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

**Arguments:**

type	name	description
VM_appliance 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 VM\_appliance.

**Signature:**

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

**Arguments:**

type	name	description
VM_appliance 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\_appliance.

**Signature:**

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



**Arguments:**

type	name	description
VM_appliance 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\_appliance.

**Signature:**

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

**Arguments:**

type	name	description
VM_appliance 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\_appliance.

**Signature:**

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

**Arguments:**

type	name	description
VM_appliance 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 VM\_appliance.

**Signature:**

```
((vm_appliance_operation) Set) get_allowed_operations (session_id s, VM_appliance ref self)
```

**Arguments:**

type	name	description
VM_appliance ref	self	reference to the object

**Return Type:** (vm\_appliance\_operation) Set  
value of the field

**RPC name:** get\_current\_operations

**Overview:**

Get the current\_operations field of the given VM\_appliance.

**Signature:**

```
((string -> vm_appliance_operation) Map) get_current_operations (session_id s, VM_appliance ref self)
```

**Arguments:**

type	name	description
VM_appliance ref	self	reference to the object

**Return Type:** (string → vm\_appliance\_operation) Map  
value of the field

**RPC name:** get\_VMs

**Overview:**

Get the VMs field of the given VM\_appliance.

**Signature:**

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

**Arguments:**

type	name	description
VM_appliance ref	self	reference to the object

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

**RPC name:** create

**Overview:**

Create a new VM\_appliance instance, and return its handle.

**Signature:**

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

**Arguments:**

type	name	description
VM_appliance record	args	All constructor arguments

**Return Type:** VM\_appliance ref  
reference to the newly created object

**RPC name:** destroy**Overview:**

Destroy the specified VM\_appliance instance.

**Signature:**

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

**Arguments:**

type	name	description
VM_appliance ref	self	reference to the object

**Return Type:** void

**RPC name:** get\_by\_uuid**Overview:**

Get a reference to the VM\_appliance instance with the specified UUID.

**Signature:**

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

**Arguments:**

type	name	description
string	uuid	UUID of object to return

**Return Type:** VM\_appliance ref

reference to the object

**RPC name:** get\_record**Overview:**

Get a record containing the current state of the given VM\_appliance.

**Signature:**

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

**Arguments:**

type	name	description
VM_appliance ref	self	reference to the object

**Return Type:** VM\_appliance record

all fields from the object

**RPC name:** get\_by\_name\_label**Overview:**

Get all the VM\_appliance instances with the given label.

**Signature:**

```
((VM_appliance 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\_appliance ref) Set  
references to objects with matching names

## 2.17 Class: DR\_task

### 2.17.1 Fields for class: DR\_task

Name	<b>DR_task</b>		
Description	<i>DR task.</i>		
Quals	Field	Type	Description
$RO_{run}$	uuid	string	Unique identifier/object reference
$RO_{run}$	introduced_SRs	(SR ref) Set	All SRs introduced by this appliance

### 2.17.2 RPCs associated with class: DR\_task

**RPC name:** create

**Overview:**

Create a disaster recovery task which will query the supplied list of devices.

**Signature:**

(DR\_task ref) create (session\_id s, string type, (string -> string) Map device\_config, string Set whitelist)

**Arguments:**

type	name	description
string	type	The SR driver type of the SRs to introduce
(string → string) Map	device_config	The device configuration of the SRs to introduce
string Set	whitelist	The devices to use for disaster recovery

**Return Type:** DR\_task ref

The reference to the created task

**RPC name:** destroy

**Overview:**

Destroy the disaster recovery task, detaching and forgetting any SRs introduced which are no longer required.

**Signature:**

void destroy (session\_id s, DR\_task ref self)

**Arguments:**

type	name	description
DR_task ref	self	The disaster recovery task to destroy

**Return Type:** void

**RPC name:** get\_all

**Overview:**

Return a list of all the DR\_tasks known to the system.

**Signature:**

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

**Return Type:** (DR\_task ref) Set  
references to all objects

**RPC name:** get\_all\_records

**Overview:**

Return a map of DR\_task references to DR\_task records for all DR\_tasks known to the system.

**Signature:**

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

**Return Type:** (DR\_task ref  $\rightarrow$  DR\_task record) Map  
records of all objects

**RPC name:** get\_uuid

**Overview:**

Get the uuid field of the given DR\_task.

**Signature:**

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

**Arguments:**

type	name	description
DR_task ref	self	reference to the object

**Return Type:** string  
value of the field

**RPC name:** get\_introduced\_SRs

**Overview:**

Get the introduced\_SRs field of the given DR\_task.

**Signature:**

```
((SR ref) Set) get_introduced_SRs (session_id s, DR_task ref self)
```

**Arguments:**

type	name	description
DR_task ref	self	reference to the object

**Return Type:** (SR ref) Set  
value of the field

**RPC name:** `get_by_uuid`

**Overview:**

Get a reference to the `DR_task` instance with the specified UUID.

**Signature:**

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

**Arguments:**

type	name	description
string	uuid	UUID of object to return

**Return Type:** `DR_task ref`

reference to the object

**RPC name:** `get_record`

**Overview:**

Get a record containing the current state of the given `DR_task`.

**Signature:**

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

**Arguments:**

type	name	description
<code>DR_task ref</code>	self	reference to the object

**Return Type:** `DR_task record`

all fields from the object

## 2.18 Class: host

### 2.18.1 Fields for class: host

Name	host		
Description	<i>A physical host.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	uuid	string	Unique identifier/object
<i>RW</i>	name/label	string	a human-readable name
<i>RW</i>	name/description	string	a notes field contain readable description
<i>RO<sub>run</sub></i>	memory/overhead	int	Virtualization memor (bytes).
<i>RO<sub>run</sub></i>	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
<i>RO<sub>run</sub></i>	current_operations	(string → host_allowed_operations) Map	links each of the running this object (by referenc rent_operation enum wh the nature of the task.
<i>RO<sub>run</sub></i>	API_version/major	int	major version number
<i>RO<sub>run</sub></i>	API_version/minor	int	minor version number
<i>RO<sub>run</sub></i>	API_version/vendor	string	identification of vendor
<i>RO<sub>run</sub></i>	API_version/vendor_implementation	(string → string) Map	details of vendor implem
<i>RO<sub>run</sub></i>	enabled	bool	True if the host is curre
<i>RO<sub>ins</sub></i>	software_version	(string → string) Map	version strings
<i>RW</i>	other_config	(string → string) Map	additional configuration
<i>RO<sub>ins</sub></i>	capabilities	string Set	Xen capabilities
<i>RO<sub>run</sub></i>	cpu_configuration	(string → string) Map	The CPU configur this host. May c such as “nr_nodes ets_per_node”, “cores-pe “threads_per_core”
<i>RO<sub>run</sub></i>	sched_policy	string	Scheduler policy current this host
<i>RO<sub>run</sub></i>	supported_bootloaders	string Set	a list of the bootloaders the machine
<i>RO<sub>run</sub></i>	resident_VMs	(VM ref) Set	list of VMs currently res
<i>RW</i>	logging	(string → string) Map	logging configuration
<i>RO<sub>run</sub></i>	PIFs	(PIF ref) Set	physical network interfa
<i>RW</i>	suspend_image_sr	SR ref	The SR in which VDIS images are created
<i>RW</i>	crash_dump_sr	SR ref	The SR in which VD dumps are created
<i>RO<sub>run</sub></i>	crashdumps	(host_crashdump ref) Set	Set of host crash dumps
<i>RO<sub>run</sub></i>	patches	(host_patch ref) Set	Set of host patches
<i>RO<sub>run</sub></i>	PBDs	(PBD ref) Set	physical blockdevices
<i>RO<sub>run</sub></i>	host_CPUs	(host_cpu ref) Set	The physical CPUs on t
<i>RO<sub>run</sub></i>	cpu_info	(string → string) Map	Details about the physi this host
<i>RW</i>	hostname	string	The hostname of this ho



<i>RW</i>	<b>address</b>	string	The address by which the host can be contacted from any other host in the pool
<i>RO<sub>run</sub></i>	<b>metrics</b>	host_metrics ref	metrics associated with the host
<i>RO<sub>run</sub></i>	<b>license_params</b>	(string → string) Map	State of the current license
<i>RO<sub>run</sub></i>	<b>boot_free_mem</b>	int	Free memory on host at boot
<i>RO<sub>run</sub></i>	<b>ha_statefiles</b>	string Set	The set of statefiles associated with this host
<i>RO<sub>run</sub></i>	<b>ha_network_peers</b>	string Set	The set of hosts visible to this host
<i>RO<sub>run</sub></i>	<b>blobs</b>	(string → blob ref) Map	Binary blobs associated with this host
<i>RW</i>	<b>tags</b>	string Set	user-specified tags for classification purposes
<i>RO<sub>run</sub></i>	<b>external_auth_type</b>	string	type of external authentication service configured; empty if not configured.
<i>RO<sub>run</sub></i>	<b>external_auth_service_name</b>	string	name of external authentication service configured; empty if not configured.
<i>RO<sub>run</sub></i>	<b>external_auth_configuration</b>	(string → string) Map	configuration specific to the authentication service
<i>RO<sub>run</sub></i>	<b>edition</b>	string	XenServer edition
<i>RW</i>	<b>license_server</b>	(string → string) Map	Contact information of the license server
<i>RO<sub>run</sub></i>	<b>bios_strings</b>	(string → string) Map	BIOS strings
<i>RO<sub>run</sub></i>	<b>power_on_mode</b>	string	The power on mode
<i>RO<sub>run</sub></i>	<b>power_on_config</b>	(string → string) Map	The power on configuration
<i>RO<sub>ins</sub></i>	<b>local_cache_sr</b>	SR ref	The SR that is used as local cache
<i>RO<sub>run</sub></i>	<b>chipset_info</b>	(string → string) Map	Information about chipsets
<i>RO<sub>run</sub></i>	<b>PCIs</b>	(PCI ref) Set	List of PCI devices in the host
<i>RO<sub>run</sub></i>	<b>PGPUs</b>	(PGPU ref) Set	List of physical GPUs in the host
<i>RW</i>	<b>guest_VCPUs_params</b>	(string → string) Map	VCPUs params to apply to dependent guests

### 2.18.2 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  $\rightarrow$  string Set) Map  
VMs which block evacuation together with reasons

**RPC name:** get\_uncooperative\_resident\_VMs

**Overview:** **This message is deprecated** 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_management_interface`**Overview:**

Returns the management interface for the specified host.

**Signature:**

```
(PIF ref) get_management_interface (session_id s, host ref host)
```

**Arguments:**

type	name	description
host ref	host	Which host's management interface is required

**Return Type:** PIF ref

The management interface for the host

**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, bool public)
```

**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
bool	public	True if the blob should be publicly available

**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_servvertime`

##### Overview:

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

##### Signature:

```
datetime get_servvertime (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, bool force)
```

#### Arguments:

type	name	description
host ref	host	The host
string	edition	The requested edition
bool	force	Update the license params even if the apply call fails

**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 power_on_config)
```

#### 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: migrate\_receive****Overview:**

Prepare to receive a VM, returning a token which can be passed to VM.migrate.

**Signature:**

```
((string -> string) Map) migrate_receive (session_id s, host ref host, network ref network, (string ->
```

**Arguments:**

type	name	description
host ref	host	The target host
network ref	network	The network through which migration traffic should be received.
(string → string) Map	options	Extra configuration operations

**Return Type:** (string → string) Map

A value which should be passed to VM.migrate

**RPC name: declare\_dead****Overview:**

Declare that a host is dead. This is a dangerous operation, and should only be called if the administrator is absolutely sure the host is definitely dead.

**Signature:**

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

**Arguments:**

type	name	description
host ref	host	The Host to declare is dead

**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
<code>(string → string) Map</code>	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
<code>(string → string) Map</code>	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
<code>(string → string) Map</code>	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_chipset_info`

**Overview:**

Get the `chipset_info` field of the given host.

**Signature:**

```
((string -> string) Map) get_chipset_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_PCIs`

**Overview:**

Get the `PCIs` field of the given host.

**Signature:**

```
((PCI ref) Set) get_PCIs (session_id s, host ref self)
```

**Arguments:**

type	name	description
host ref	self	reference to the object

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

**RPC name:** `get_PGPUs`

**Overview:**

Get the `PGPUs` field of the given host.

**Signature:**

```
((PGPU ref) Set) get_PGPUs (session_id s, host ref self)
```

**Arguments:**

type	name	description
host ref	self	reference to the object

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

**RPC name:** `get_guest_VCPUs_params`

**Overview:**

Get the `guest_VCPUs_params` field of the given host.

**Signature:**

```
((string -> string) Map) get_guest_VCPUs_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:** `set_guest_VCPUs_params`

**Overview:**

Set the `guest_VCPUs_params` field of the given host.

**Signature:**

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

**Arguments:**

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

**Return Type:** `void`

**RPC name:** `add_to_guest_VCPUs_params`

**Overview:**

Add the given key-value pair to the `guest_VCPUs_params` field of the given host.

**Signature:**

```
void add_to_guest_VCPUs_params (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_guest_VCPUs_params`

**Overview:**

Remove the given key and its corresponding value from the `guest_VCPUs_params` field of the given host. If the key is not in that Map, then do nothing.

**Signature:**

```
void remove_from_guest_VCPUs_params (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_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

## 2.19 Class: host\_crashdump

### 2.19.1 Fields for class: host\_crashdump

Name	<b>host_crashdump</b>		
Description	<i>Represents a host crash dump.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	<b>uuid</b>	string	Unique identifier/object reference
<i>RO<sub>ins</sub></i>	<b>host</b>	host ref	Host the crashdump relates to
<i>RO<sub>run</sub></i>	<b>timestamp</b>	datetime	Time the crash happened
<i>RO<sub>run</sub></i>	<b>size</b>	int	Size of the crashdump
<i>RO<sub>ins</sub></i>	<b>filename</b>	string	filename of crash dir
<i>RW</i>	<b>other_config</b>	(string → string) Map	additional configuration

### 2.19.2 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  $\rightarrow$  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
<code>host_crashdump ref</code>	self	reference to the object

**Return Type:** `host_crashdump record`  
all fields from the object

## 2.20 Class: host\_patch

### 2.20.1 Fields for class: host\_patch

Name	<b>host_patch</b>		
Description	<i>Represents a patch stored on a server.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	uuid	string	Unique identifier/object reference
<i>RO<sub>ins</sub></i>	name/label	string	a human-readable name
<i>RO<sub>ins</sub></i>	name/description	string	a notes field containing human-readable description
<i>RO<sub>ins</sub></i>	version	string	Patch version number
<i>RO<sub>ins</sub></i>	host	host ref	Host the patch relates to
<i>RO<sub>run</sub></i>	filename	string	Filename of the patch
<i>RO<sub>run</sub></i>	applied	bool	True if the patch has been applied
<i>RO<sub>run</sub></i>	timestamp_applied	datetime	Time the patch was applied
<i>RO<sub>run</sub></i>	size	int	Size of the patch
<i>RO<sub>ins</sub></i>	pool_patch	pool_patch ref	The patch applied
<i>RW</i>	other_config	(string → string) Map	additional configuration

### 2.20.2 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_patches` 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_patches` 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
<code>host_patch ref</code>	<code>self</code>	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
<code>host_patch ref</code>	<code>self</code>	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
<code>(string → string) Map</code>	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

## 2.21 Class: host\_metrics

### 2.21.1 Fields for class: host\_metrics

Name	<b>host_metrics</b>		
Description	<i>The metrics associated with a host.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	uuid	string	Unique identifier/object reference
<i>RO<sub>run</sub></i>	memory/total	int	Total host memory (bytes)
<i>RO<sub>run</sub></i>	memory/free	int	Free host memory (bytes)
<i>RO<sub>run</sub></i>	live	bool	Pool master thinks this host is live
<i>RO<sub>run</sub></i>	last_updated	datetime	Time at which this information was last updated
<i>RW</i>	other_config	(string → string) Map	additional configuration

### 2.21.2 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
<code>host_metrics ref</code>	<code>self</code>	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
<code>host_metrics ref</code>	<code>self</code>	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
<code>host_metrics ref</code>	<code>self</code>	reference to the object
<code>(string → string) Map</code>	<code>value</code>	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
<code>host_metrics ref</code>	<code>self</code>	reference to the object
<code>string</code>	<code>key</code>	Key to add
<code>string</code>	<code>value</code>	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
<code>host_metrics ref</code>	<code>self</code>	reference to the object
<code>string</code>	<code>key</code>	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
<code>string</code>	<code>uuid</code>	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
<code>host_metrics ref</code>	<code>self</code>	reference to the object

**Return Type:** `host_metrics record`

all fields from the object

## 2.22 Class: host\_cpu

### 2.22.1 Fields for class: host\_cpu

Name	host_cpu		
Description	<i>A physical CPU.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	<code>uuid</code>	string	Unique identifier/object reference
<i>RO<sub>run</sub></i>	<code>host</code>	host ref	the host the CPU is in
<i>RO<sub>run</sub></i>	<code>number</code>	int	the number of the physical CPU within the host
<i>RO<sub>run</sub></i>	<code>vendor</code>	string	the vendor of the physical CPU
<i>RO<sub>run</sub></i>	<code>speed</code>	int	the speed of the physical CPU
<i>RO<sub>run</sub></i>	<code>modelname</code>	string	the model name of the physical CPU
<i>RO<sub>run</sub></i>	<code>family</code>	int	the family (number) of the physical CPU
<i>RO<sub>run</sub></i>	<code>model</code>	int	the model number of the physical CPU
<i>RO<sub>run</sub></i>	<code>stepping</code>	string	the stepping of the physical CPU
<i>RO<sub>run</sub></i>	<code>flags</code>	string	the flags of the physical CPU (a decoded version of the features field)
<i>RO<sub>run</sub></i>	<code>features</code>	string	the physical CPU feature bitmap
<i>RO<sub>run</sub></i>	<code>utilisation</code>	float	the current CPU utilisation
<i>RW</i>	<code>other_config</code>	(string → string) Map	additional configuration

### 2.22.2 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
<code>(string → string) Map</code>	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

## 2.23 Class: network

### 2.23.1 Fields for class: network

Name	<b>network</b>		
Description	<i>A virtual network.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	uuid	string	Unique identifier/object reference
<i>RW</i>	name/label	string	a human-readable name
<i>RW</i>	name/description	string	a notes field containing human-readable description
<i>RO<sub>run</sub></i>	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.
<i>RO<sub>run</sub></i>	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.
<i>RO<sub>run</sub></i>	VIFs	(VIF ref) Set	list of connected vifs
<i>RO<sub>run</sub></i>	PIFs	(PIF ref) Set	list of connected pifs
<i>RW</i>	MTU	int	MTU in octets
<i>RW</i>	other_config	(string → string) Map	additional configuration
<i>RO<sub>run</sub></i>	bridge	string	name of the bridge corresponding to this network on the local host
<i>RO<sub>run</sub></i>	blobs	(string → blob ref) Map	Binary blobs associated with this network
<i>RW</i>	tags	string Set	user-specified tags for categorization purposes
<i>RO<sub>run</sub></i>	default_locking_mode	network_default_locking_mode	The network will use this value to determine the behaviour of all VIFs where locking_mode = default
<i>RO<sub>run</sub></i>	assigned_ips	(VIF ref → string) Map	The IP addresses assigned to VIFs on networks that have active xapi-managed DHCP

### 2.23.2 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, bool public)

##### 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
bool	public	True if the blob should be publicly available

**Return Type:** blob ref

The reference of the blob, needed for populating its data

**RPC name:** set\_default\_locking\_mode**Overview:**

Set the default locking mode for VIFs attached to this network.

**Signature:**

```
void set_default_locking_mode (session_id s, network ref network, network_default_locking_mode value)
```

**Arguments:**

type	name	description
network ref	network	The network
network_default_locking_mode	value	The default locking mode for VIFs attached to this network.

**Return Type:** void**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: get\_default\_locking\_mode****Overview:**

Get the default\_locking\_mode field of the given network.

**Signature:**

```
(network_default_locking_mode) get_default_locking_mode (session_id s, network ref self)
```

**Arguments:**

type	name	description
network ref	self	reference to the object

**Return Type:** network\_default\_locking\_mode  
value of the field

**RPC name:** `get_assigned_ips`

**Overview:**

Get the `assigned_ips` field of the given network.

**Signature:**

```
((VIF ref -> string) Map) get_assigned_ips (session_id s, network ref self)
```

**Arguments:**

type	name	description
network ref	self	reference to the object

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

**RPC name:** `create`

**Overview:**

Create a new network instance, and return its handle.

**Signature:**

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

**Arguments:**

type	name	description
network record	args	All constructor arguments

**Return Type:** `network ref`  
reference to the newly created object

**RPC name:** `destroy`

**Overview:**

Destroy the specified network instance.

**Signature:**

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

**Arguments:**

type	name	description
network ref	self	reference to the object

**Return Type:** `void`

**RPC name:** `get_by_uuid`

**Overview:**

Get a reference to the network instance with the specified UUID.

**Signature:**

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

**Arguments:**

type	name	description
string	uuid	UUID of object to return

**Return Type:** network ref  
reference to the object

**RPC name:** get\_record

**Overview:**

Get a record containing the current state of the given network.

**Signature:**

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

**Arguments:**

type	name	description
network ref	self	reference to the object

**Return Type:** network record  
all fields from the object

**RPC name:** get\_by\_name\_label

**Overview:**

Get all the network instances with the given label.

**Signature:**

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

**Arguments:**

type	name	description
string	label	label of object to return

**Return Type:** (network ref) Set  
references to objects with matching names

## 2.24 Class: VIF

### 2.24.1 Fields for class: VIF

Name	VIF		
Description	<i>A virtual network interface.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	<code>uuid</code>	string	Unique identifier/object reference
<i>RO<sub>run</sub></i>	<code>allowed_operations</code>	(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.
<i>RO<sub>run</sub></i>	<code>current_operations</code>	(string $\rightarrow$ 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.
<i>RO<sub>ins</sub></i>	<code>device</code>	string	order in which VIF backends are created by xapi
<i>RO<sub>ins</sub></i>	<code>network</code>	network ref	virtual network to which this vif is connected
<i>RO<sub>ins</sub></i>	<code>VM</code>	VM ref	virtual machine to which this vif is connected
<i>RO<sub>ins</sub></i>	<code>MAC</code>	string	ethernet MAC address of virtual interface, as exposed to guest
<i>RO<sub>ins</sub></i>	<code>MTU</code>	int	MTU in octets
<i>RO<sub>run</sub></i>	<code>reserved</code>	bool	true if the VIF is reserved pending a reboot/migrate
<i>RW</i>	<code>other_config</code>	(string $\rightarrow$ string) Map	additional configuration
<i>RO<sub>run</sub></i>	<code>currently_attached</code>	bool	is the device currently attached (erased on reboot)
<i>RO<sub>run</sub></i>	<code>status_code</code>	int	error/success code associated with last attach-operation (erased on reboot)
<i>RO<sub>run</sub></i>	<code>status_detail</code>	string	error/success information associated with last attach-operation status (erased on reboot)
<i>RO<sub>run</sub></i>	<code>runtime_properties</code>	(string $\rightarrow$ string) Map	Device runtime properties
<i>RW</i>	<code>qos/algorithm.type</code>	string	QoS algorithm to use
<i>RW</i>	<code>qos/algorithm.params</code>	(string $\rightarrow$ string) Map	parameters for chosen QoS algorithm
<i>RO<sub>run</sub></i>	<code>qos/supported_algorithms</code>	string Set	supported QoS algorithms for this VIF
<i>RO<sub>run</sub></i>	<code>metrics</code>	VIF_metrics ref	metrics associated with this VIF
<i>RO<sub>run</sub></i>	<code>MAC_autogenerated</code>	bool	true if the MAC was autogenerated; false indicates it was set manually
<i>RO<sub>ins</sub></i>	<code>locking_mode</code>	vif_locking_mode	current locking mode of the VIF
<i>RO<sub>ins</sub></i>	<code>ipv4_allowed</code>	string Set	A list of IPv4 addresses which can be used to filter traffic passing through this VIF
<i>RO<sub>ins</sub></i>	<code>ipv6_allowed</code>	string Set	A list of IPv6 addresses which can be used to filter traffic passing through this VIF

**2.24.2 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:** unplug\_force**Overview:**

Forcibly unplug the specified VIF.

**Signature:**

```
void unplug_force (session_id s, VIF ref self)
```

**Arguments:**

type	name	description
VIF ref	self	The VIF to forcibly unplug

**Return Type:** void**RPC name:** set\_locking\_mode**Overview:**

Set the locking mode for this VIF.

**Signature:**

```
void set_locking_mode (session_id s, VIF ref self, vif_locking_mode value)
```

**Arguments:**

type	name	description
VIF ref	self	The VIF whose locking mode will be set
vif_locking_mode	value	The new locking mode for the VIF

**Return Type:** void**RPC name:** set\_ipv4\_allowed**Overview:**

Set the IPv4 addresses to which traffic on this VIF can be restricted.

**Signature:**

```
void set_ipv4_allowed (session_id s, VIF ref self, string Set value)
```

**Arguments:**

type	name	description
VIF ref	self	The VIF which the IP addresses will be associated with
string Set	value	The IP addresses which will be associated with the VIF

**Return Type:** void**RPC name:** add\_ipv4\_allowed**Overview:**

Associates an IPv4 address with this VIF.

**Signature:**

```
void add_ipv4_allowed (session_id s, VIF ref self, string value)
```

**Arguments:**

type	name	description
VIF ref	self	The VIF which the IP address will be associated with
string	value	The IP address which will be associated with the VIF

**Return Type:** void**RPC name:** remove\_ipv4\_allowed**Overview:**

Removes an IPv4 address from this VIF.

**Signature:**

```
void remove_ipv4_allowed (session_id s, VIF ref self, string value)
```



**Arguments:**

type	name	description
VIF ref	self	The VIF from which the IP address will be removed
string	value	The IP address which will be removed from the VIF

**Return Type:** void**RPC name:** set\_ipv6\_allowed**Overview:**

Set the IPv6 addresses to which traffic on this VIF can be restricted.

**Signature:**

```
void set_ipv6_allowed (session_id s, VIF ref self, string Set value)
```

**Arguments:**

type	name	description
VIF ref	self	The VIF which the IP addresses will be associated with
string Set	value	The IP addresses which will be associated with the VIF

**Return Type:** void**RPC name:** add\_ipv6\_allowed**Overview:**

Associates an IPv6 address with this VIF.

**Signature:**

```
void add_ipv6_allowed (session_id s, VIF ref self, string value)
```

**Arguments:**

type	name	description
VIF ref	self	The VIF which the IP address will be associated with
string	value	The IP address which will be associated with the VIF

**Return Type:** void**RPC name:** remove\_ipv6\_allowed**Overview:**

Removes an IPv6 address from this VIF.

**Signature:**

```
void remove_ipv6_allowed (session_id s, VIF ref self, string value)
```

**Arguments:**

type	name	description
VIF ref	self	The VIF from which the IP address will be removed
string	value	The IP address which will be removed from the VIF

**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  $\rightarrow$  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
<code>(string → string) Map</code>	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:** `get_locking_mode`

**Overview:**

Get the `locking_mode` field of the given VIF.

**Signature:**

```
(vif_locking_mode) get_locking_mode (session_id s, VIF ref self)
```

**Arguments:**

type	name	description
VIF ref	self	reference to the object

**Return Type:** `vif_locking_mode`  
value of the field

**RPC name:** `get_ipv4_allowed`

**Overview:**

Get the `ipv4_allowed` field of the given VIF.

**Signature:**

```
(string Set) get_ipv4_allowed (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_ipv6_allowed`

**Overview:**

Get the `ipv6_allowed` field of the given VIF.

**Signature:**

```
(string Set) get_ipv6_allowed (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:** 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

## 2.25 Class: VIF\_metrics

### 2.25.1 Fields for class: VIF\_metrics

Name	VIF_metrics		
Description	<i>The metrics associated with a virtual network device.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	<code>uuid</code>	string	Unique identifier/object reference
<i>RO<sub>run</sub></i>	<code>io/read_kbs</code>	float	Read bandwidth (KiB/s)
<i>RO<sub>run</sub></i>	<code>io/write_kbs</code>	float	Write bandwidth (KiB/s)
<i>RO<sub>run</sub></i>	<code>last_updated</code>	datetime	Time at which this information was last updated
<i>RW</i>	<code>other_config</code>	(string → string) Map	additional configuration

### 2.25.2 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
<code>VIF_metrics ref</code>	<code>self</code>	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
<code>VIF_metrics ref</code>	<code>self</code>	reference to the object
<code>(string → string) Map</code>	<code>value</code>	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
<code>VIF_metrics ref</code>	<code>self</code>	reference to the object
<code>string</code>	<code>key</code>	Key to add
<code>string</code>	<code>value</code>	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

## 2.26 Class: PIF

### 2.26.1 Fields for class: PIF

Name	<b>PIF</b>		
Description	<i>A physical network interface (note separate VLANs are represented as several PIFs).</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	<code>uuid</code>	string	Unique identifier/object reference
<i>RO<sub>ins</sub></i>	<code>device</code>	string	machine-readable name of the interface (e.g. eth0)
<i>RO<sub>ins</sub></i>	<code>network</code>	network ref	virtual network to which this pif is connected
<i>RO<sub>ins</sub></i>	<code>host</code>	host ref	physical machine to which this pif is connected
<i>RO<sub>ins</sub></i>	<code>MAC</code>	string	ethernet MAC address of physical interface
<i>RO<sub>ins</sub></i>	<code>MTU</code>	int	MTU in octets
<i>RO<sub>ins</sub></i>	<code>VLAN</code>	int	VLAN tag for all traffic passing through this interface
<i>RW</i>	<code>device_name</code>	string	actual dom0 device name
<i>RO<sub>run</sub></i>	<code>metrics</code>	PIF_metrics ref	metrics associated with this PIF
<i>RO<sub>run</sub></i>	<code>physical</code>	bool	true if this represents a physical network interface
<i>RO<sub>run</sub></i>	<code>currently_attached</code>	bool	true if this interface is online
<i>RO<sub>run</sub></i>	<code>ip_configuration_mode</code>	ip_configuration_mode	Sets if and how this interface gets an IP address
<i>RO<sub>run</sub></i>	<code>IP</code>	string	IP address
<i>RO<sub>run</sub></i>	<code>netmask</code>	string	IP netmask
<i>RO<sub>run</sub></i>	<code>gateway</code>	string	IP gateway
<i>RO<sub>run</sub></i>	<code>DNS</code>	string	IP address of DNS servers to use
<i>RO<sub>run</sub></i>	<code>bond_slave_of</code>	Bond ref	Indicates which bond this interface is part of
<i>RO<sub>run</sub></i>	<code>bond_master_of</code>	(Bond ref) Set	Indicates this PIF represents the results of a bond
<i>RO<sub>run</sub></i>	<code>VLAN_master_of</code>	VLAN ref	Indicates which VLAN this interface receives untagged traffic from
<i>RO<sub>run</sub></i>	<code>VLAN_slave_of</code>	(VLAN ref) Set	Indicates which VLANs this interface transmits tagged traffic to
<i>RO<sub>run</sub></i>	<code>management</code>	bool	Indicates whether the control software is listening for connections on this interface
<i>RW</i>	<code>other_config</code>	(string → string) Map	Additional configuration
<i>RW</i>	<code>disallow_unplug</code>	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)
<i>RO<sub>run</sub></i>	<code>tunnel_access_PIF_of</code>	(tunnel ref) Set	Indicates to which tunnel this PIF gives access
<i>RO<sub>run</sub></i>	<code>tunnel_transport_PIF_of</code>	(tunnel ref) Set	Indicates to which tunnel this PIF provides transport

<i>RO<sub>run</sub></i>	ipv6_configuration_mode	ipv6_configuration_mode	Sets if and how this interface gets an IPv6 address
<i>RO<sub>run</sub></i>	IPv6	string Set	IPv6 address
<i>RO<sub>run</sub></i>	ipv6_gateway	string	IPv6 gateway
<i>RO<sub>run</sub></i>	primary_address_type	primary_address_type	Which protocol should define the primary address of this interface
<i>RO<sub>ins</sub></i>	managed	bool	Indicates whether the interface is managed by xapi. If it is not, then xapi will not configure the interface, the commands PIF.plugin/unplug/reconfigure_ip(v6) can not be used, nor can the interface be bonded or have VLANs based on top through xapi.
<i>RO<sub>run</sub></i>	properties	(string → string) Map	Additional configuration properties for the interface.

### 2.26.2 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:** reconfigure\_ipv6

**Overview:**

Reconfigure the IPv6 address settings for this interface.

**Signature:**

```
void reconfigure_ipv6 (session_id s, PIF ref self, ipv6_configuration_mode mode, string IPv6, string gateway)
```

**Arguments:**

type	name	description
PIF ref	self	the PIF object to reconfigure
ipv6_configuration_mode	mode	whether to use dynamic/static/no-assignment
string	IPv6	the new IPv6 address (in  addr / prefix length  format)
string	gateway	the new gateway
string	DNS	the new DNS settings

**Return Type:** void

**RPC name:** set\_primary\_address\_type

**Overview:**

Change the primary address type used by this PIF.

**Signature:**

```
void set_primary_address_type (session_id s, PIF ref self, primary_address_type primary_address_type)
```

**Arguments:**

type	name	description
PIF ref	self	the PIF object to reconfigure
primary_address_type	primary_address_type	Whether to prefer IPv4 or IPv6 connections

**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:** 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, bool managed)
```

**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
bool	managed	Indicates whether the interface is managed by xapi (defaults to “true”)

**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	
ipv6_configuration_mode	ipv6_configuration_mode	
string Set	IPv6	
string	ipv6_gateway	
primary_address_type	primary_address_type	
bool	managed	
(string → string) Map	properties	

**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:** set\_property

**Overview:**

Set the value of a property of the PIF.

**Signature:**

```
void set_property (session_id s, PIF ref self, string name, string value)
```

**Arguments:**

type	name	description
PIF ref	self	The PIF
string	name	The property name
string	value	The property value

**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  $\rightarrow$  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
<code>(string → string) Map</code>	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_ipv6_configuration_mode`

**Overview:**

Get the `ipv6_configuration_mode` field of the given PIF.

**Signature:**

```
(ipv6_configuration_mode) get_ipv6_configuration_mode (session_id s, PIF ref self)
```

**Arguments:**

type	name	description
PIF ref	self	reference to the object

**Return Type:** `ipv6_configuration_mode`  
value of the field

**RPC name:** `get_IPv6`

**Overview:**

Get the IPv6 field of the given PIF.

**Signature:**

```
(string Set) get_IPv6 (session_id s, PIF ref self)
```

**Arguments:**

type	name	description
PIF ref	self	reference to the object

**Return Type:** `string Set`  
value of the field

**RPC name:** `get_ipv6_gateway`

**Overview:**

Get the `ipv6_gateway` field of the given PIF.

**Signature:**

```
string get_ipv6_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_primary_address_type`

**Overview:**

Get the `primary_address_type` field of the given PIF.

**Signature:**

```
(primary_address_type) get_primary_address_type (session_id s, PIF ref self)
```

**Arguments:**

type	name	description
PIF ref	self	reference to the object

**Return Type:** `primary_address_type`  
value of the field

**RPC name:** `get_managed`

**Overview:**

Get the `managed` field of the given PIF.

**Signature:**

```
bool get_managed (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_properties`

**Overview:**

Get the `properties` field of the given PIF.

**Signature:**

```
((string -> string) Map) get_properties (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:** `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

## 2.27 Class: PIF\_metrics

### 2.27.1 Fields for class: PIF\_metrics

Name	PIF_metrics		
Description	<i>The metrics associated with a physical network interface.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	<code>uuid</code>	string	Unique identifier/object reference
<i>RO<sub>run</sub></i>	<code>io/read_kbs</code>	float	Read bandwidth (KiB/s)
<i>RO<sub>run</sub></i>	<code>io/write_kbs</code>	float	Write bandwidth (KiB/s)
<i>RO<sub>run</sub></i>	<code>carrier</code>	bool	Report if the PIF got a carrier or not
<i>RO<sub>run</sub></i>	<code>vendor_id</code>	string	Report vendor ID
<i>RO<sub>run</sub></i>	<code>vendor_name</code>	string	Report vendor name
<i>RO<sub>run</sub></i>	<code>device_id</code>	string	Report device ID
<i>RO<sub>run</sub></i>	<code>device_name</code>	string	Report device name
<i>RO<sub>run</sub></i>	<code>speed</code>	int	Speed of the link (if available)
<i>RO<sub>run</sub></i>	<code>duplex</code>	bool	Full duplex capability of the link (if available)
<i>RO<sub>run</sub></i>	<code>pci_bus_path</code>	string	PCI bus path of the pif (if available)
<i>RO<sub>run</sub></i>	<code>last_updated</code>	datetime	Time at which this information was last updated
<i>RW</i>	<code>other_config</code>	(string → string) Map	additional configuration

### 2.27.2 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
<code>PIF_metrics ref</code>	<code>self</code>	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
<code>PIF_metrics ref</code>	<code>self</code>	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
<code>PIF_metrics ref</code>	<code>self</code>	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
<code>PIF_metrics ref</code>	<code>self</code>	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
<code>PIF_metrics ref</code>	<code>self</code>	reference to the object
<code>(string → string) Map</code>	<code>value</code>	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
<code>PIF_metrics ref</code>	<code>self</code>	reference to the object
<code>string</code>	<code>key</code>	Key to add
<code>string</code>	<code>value</code>	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
<code>PIF_metrics ref</code>	<code>self</code>	reference to the object
<code>string</code>	<code>key</code>	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
<code>string</code>	<code>uuid</code>	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
<code>PIF_metrics ref</code>	<code>self</code>	reference to the object

**Return Type:** `PIF_metrics record`  
all fields from the object

## 2.28 Class: Bond

### 2.28.1 Fields for class: Bond

Name	Bond		
Description	.		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	<code>uuid</code>	string	Unique identifier/object reference
<i>RO<sub>ins</sub></i>	<code>master</code>	PIF ref	The bonded interface
<i>RO<sub>run</sub></i>	<code>slaves</code>	(PIF ref) Set	The interfaces which are part of this bond
<i>RW</i>	<code>other_config</code>	(string $\rightarrow$ string) Map	additional configuration
<i>RO<sub>run</sub></i>	<code>primary_slave</code>	PIF ref	The PIF of which the IP configuration and MAC were copied to the bond, and which will receive all configuration/VLANs/VIFs on the bond if the bond is destroyed
<i>RO<sub>run</sub></i>	<code>mode</code>	bond_mode	The algorithm used to distribute traffic among the bonded NICs
<i>RO<sub>run</sub></i>	<code>properties</code>	(string $\rightarrow$ string) Map	Additional configuration properties specific to the bond mode.
<i>RO<sub>run</sub></i>	<code>links_up</code>	int	Number of links up in this bond

### 2.28.2 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, bond\_mode mode)

**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 primary slave.
bond_mode	mode	Bonding mode to use for the new bond
(string $\rightarrow$ string) Map	properties	Additional configuration parameters specific to the bond mode

**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:** set\_mode

**Overview:**

Change the bond mode.

**Signature:**

```
void set_mode (session_id s, Bond ref self, bond_mode value)
```

**Arguments:**

type	name	description
Bond ref	self	The bond
bond_mode	value	The new bond mode

**Return Type:** void

**RPC name:** set\_property

**Overview:**

Set the value of a property of the bond.

**Signature:**

```
void set_property (session_id s, Bond ref self, string name, string value)
```

**Arguments:**

type	name	description
Bond ref	self	The bond
string	name	The property name
string	value	The property value

**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\_primary\_slave

**Overview:**

Get the primary\_slave field of the given Bond.

**Signature:**

```
(PIF ref) get_primary_slave (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\_mode

**Overview:**

Get the mode field of the given Bond.

**Signature:**

```
(bond_mode) get_mode (session_id s, Bond ref self)
```

**Arguments:**

type	name	description
Bond ref	self	reference to the object

**Return Type:** bond\_mode

value of the field



**RPC name:** `get_properties`

**Overview:**

Get the properties field of the given Bond.

**Signature:**

```
((string -> string) Map) get_properties (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:** `get_links_up`

**Overview:**

Get the links\_up field of the given Bond.

**Signature:**

```
int get_links_up (session_id s, Bond ref self)
```

**Arguments:**

type	name	description
Bond ref	self	reference to the object

**Return Type:** `int`  
value of the field

**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

## 2.29 Class: VLAN

### 2.29.1 Fields for class: VLAN

Name	<b>VLAN</b>		
Description	<i>A VLAN mux/demux.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	<i>uuid</i>	string	Unique identifier/object reference
<i>RO<sub>ins</sub></i>	<i>tagged_PIF</i>	PIF ref	interface on which traffic is tagged
<i>RO<sub>run</sub></i>	<i>untagged_PIF</i>	PIF ref	interface on which traffic is untagged
<i>RO<sub>ins</sub></i>	<i>tag</i>	int	VLAN tag in use
<i>RW</i>	<i>other_config</i>	(string → string) Map	additional configuration

### 2.29.2 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  $\rightarrow$  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

## 2.30 Class: SM

### 2.30.1 Fields for class: SM

Name	SM		
Description	<i>A storage manager plugin.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	uuid	string	Unique identifier/object reference
<i>RO<sub>run</sub></i>	name/label	string	a human-readable name
<i>RO<sub>run</sub></i>	name/description	string	a notes field containing human-readable description
<i>RO<sub>run</sub></i>	type	string	SR.type
<i>RO<sub>run</sub></i>	vendor	string	Vendor who created this plugin
<i>RO<sub>run</sub></i>	copyright	string	Entity which owns the copyright of this plugin
<i>RO<sub>run</sub></i>	version	string	Version of the plugin
<i>RO<sub>run</sub></i>	required_api_version	string	Minimum SM API version required on the server
<i>RO<sub>run</sub></i>	configuration	(string → string) Map	names and descriptions of device config keys
<i>RO<sub>run</sub></i>	capabilities	string Set	capabilities of the SM plugin
<i>RO<sub>run</sub></i>	features	(string → int) Map	capabilities of the SM plugin, with capability version numbers
<i>RW</i>	other_config	(string → string) Map	additional configuration
<i>RO<sub>run</sub></i>	driver_filename	string	filename of the storage driver

### 2.30.2 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:** This message is deprecated 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_features`

**Overview:**

Get the features field of the given SM.

**Signature:**

```
((string -> int) Map) get_features (session_id s, SM ref self)
```

**Arguments:**

type	name	description
SM ref	self	reference to the object

**Return Type:** `(string → int) Map`

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

## 2.31 Class: SR

### 2.31.1 Fields for class: SR

Name	SR		
Description	<i>A storage repository.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	uuid	string	Unique identifier/object reference
<i>RO<sub>ins</sub></i>	name/label	string	a human-readable name
<i>RO<sub>ins</sub></i>	name/description	string	a notes field containing human-readable description
<i>RO<sub>run</sub></i>	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.
<i>RO<sub>run</sub></i>	current_operations	(string $\rightarrow$ 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.
<i>RO<sub>run</sub></i>	VDIs	(VDI ref) Set	all virtual disks known to this storage repository
<i>RO<sub>run</sub></i>	PBDs	(PBD ref) Set	describes how particular hosts can see this storage repository
<i>RO<sub>run</sub></i>	virtual_allocation	int	sum of virtual_sizes of all VDIs in this storage repository (in bytes)
<i>RO<sub>run</sub></i>	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
<i>RO<sub>ins</sub></i>	physical_size	int	total physical size of the repository (in bytes)
<i>RO<sub>ins</sub></i>	type	string	type of the storage repository
<i>RO<sub>ins</sub></i>	content_type	string	the type of the SR's content, if required (e.g. ISOs)
<i>RO<sub>run</sub></i>	shared	bool	true if this SR is (capable of being) shared between multiple hosts
<i>RW</i>	other_config	(string $\rightarrow$ string) Map	additional configuration
<i>RW</i>	tags	string Set	user-specified tags for categorization purposes
<i>RO<sub>run</sub></i>	default_vdi_visibility	bool	SM dependent data
<i>RW</i>	sm_config	(string $\rightarrow$ string) Map	
<i>RO<sub>run</sub></i>	blobs	(string $\rightarrow$ blob ref) Map	
<i>RO<sub>run</sub></i>	local_cache_enabled	bool	
<i>RO<sub>run</sub></i>	introduced_by	DR_task ref	The disaster recovery task which introduced this SR

### 2.31.2 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:** set\_name\_label**Overview:**

Set the name label of the SR.

**Signature:**

```
void set_name_label (session_id s, SR ref sr, string value)
```

**Arguments:**

type	name	description
SR ref	sr	The SR
string	value	The name label for the SR

**Return Type:** void

**RPC name:** set\_name\_description

**Overview:**

Set the name description of the SR.

**Signature:**

```
void set_name_description (session_id s, SR ref sr, string value)
```

**Arguments:**

type	name	description
SR ref	sr	The SR
string	value	The name description for the SR

**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, bool public)
```

**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
bool	public	True if the blob should be publicly available

**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: assert\_supports\_database\_replication****Overview:**

Returns successfully if the given SR supports database replication. Otherwise returns an error to explain why not.

**Signature:**

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

**Arguments:**

type	name	description
SR ref	sr	The SR to query

**Return Type:** void

**RPC name: enable\_database\_replication****Overview:**

.

**Signature:**

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

**Arguments:**

type	name	description
SR ref	sr	The SR to which metadata should be replicated

**Return Type:** void

**RPC name: disable\_database\_replication****Overview:**

.

**Signature:**

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

**Arguments:**

type	name	description
SR ref	sr	The SR to which metadata should be no longer replicated

**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  $\rightarrow$  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:** `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:** `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\_VDI

**Overview:**

Get the VDIs field of the given SR.

**Signature:**

```
((VDI ref) Set) get_VDI (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_introduced_by`

**Overview:**

Get the `introduced_by` field of the given SR.

**Signature:**

```
(DR_task ref) get_introduced_by (session_id s, SR ref self)
```

**Arguments:**

type	name	description
SR ref	self	reference to the object

**Return Type:** `DR_task ref`

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

## 2.32 Class: VDI

### 2.32.1 Fields for class: VDI

Name	VDI		
Description	<i>A virtual disk image.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	uuid	string	Unique identifier/object reference
<i>RO<sub>ins</sub></i>	name/label	string	a human-readable name
<i>RO<sub>ins</sub></i>	name/description	string	a notes field containing human-readable description
<i>RO<sub>run</sub></i>	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.
<i>RO<sub>run</sub></i>	current_operations	(string $\rightarrow$ 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.
<i>RO<sub>ins</sub></i>	SR	SR ref	storage repository in which the VDI resides
<i>RO<sub>run</sub></i>	VBDs	(VBD ref) Set	list of vbds that refer to this disk
<i>RO<sub>run</sub></i>	crash_dumps	(crashdump ref) Set	list of crash dumps that refer to this disk
<i>RO<sub>ins</sub></i>	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
<i>RO<sub>run</sub></i>	physical_utilisation	int	amount of physical space that the disk image is currently taking up on the storage repository (in bytes)
<i>RO<sub>ins</sub></i>	type	vdi_type	type of the VDI
<i>RO<sub>ins</sub></i>	sharable	bool	true if this disk may be shared
<i>RO<sub>ins</sub></i>	read_only	bool	true if this disk may ONLY be mounted read-only
<i>RW</i>	other_config	(string $\rightarrow$ string) Map	additional configuration
<i>RO<sub>run</sub></i>	storage_lock	bool	true if this disk is locked at the storage level
<i>RO<sub>run</sub></i>	location	string	location information
<i>RO<sub>run</sub></i>	managed	bool	
<i>RO<sub>run</sub></i>	missing	bool	true if SR scan operation reported this VDI as not present on disk
<i>RO<sub>run</sub></i>	parent	VDI ref	References the parent disk, if this VDI is part of a chain
<i>RW</i>	xenstore_data	(string $\rightarrow$ string) Map	data to be inserted into the xenstore tree (/local/domain/0/backend/vbd/ <i>id<sub>i</sub></i> /id <sub>i</sub> /device/ <i>id<sub>i</sub></i> /sm-data) after the VDI is attached. This is generally set by the SM backends on vdi_attach.
<i>RW</i>	sm_config	(string $\rightarrow$ string) Map	SM dependent data
<i>RO<sub>run</sub></i>	is_a_snapshot	bool	true if this is a snapshot.
<i>RO<sub>run</sub></i>	snapshot_of	VDI ref	Ref pointing to the VDI this snapshot is of.

<i>RO<sub>run</sub></i>	<code>snapshots</code>	(VDI ref) Set	List pointing to all the VDIs snapshots.
<i>RO<sub>run</sub></i>	<code>snapshot_time</code>	datetime	Date/time when this snapshot was created.
<i>RW</i>	<code>tags</code>	string Set	user-specified tags for categorization purposes
<i>RO<sub>run</sub></i>	<code>allow_caching</code>	bool	true if this VDI is to be cached in the local cache SR
<i>RO<sub>run</sub></i>	<code>on_boot</code>	on_boot	The behaviour of this VDI on a VM boot
<i>RO<sub>run</sub></i>	<code>metadata_of_pool</code>	pool ref	The pool whose metadata is contained in this VDI
<i>RO<sub>run</sub></i>	<code>metadata_latest</code>	bool	Whether this VDI contains the latest known accessible metadata for the pool

### 2.32.2 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
bool	managed	Storage-specific config
int	virtual_size	Storage-specific config
int	physical_utilisation	Storage-specific config
pool ref	metadata_of_pool	Storage-specific config
bool	is_a_snapshot	Storage-specific config
datetime	snapshot_time	Storage-specific config
VDI ref	snapshot_of	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
bool	managed	Storage-specific config
int	virtual_size	Storage-specific config
int	physical_utilisation	Storage-specific config
pool ref	metadata_of_pool	Storage-specific config
bool	is_a_snapshot	Storage-specific config
datetime	snapshot_time	Storage-specific config
VDI ref	snapshot_of	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:**

Copy either a full VDI or the block differences between two VDIs into either a fresh VDI or an existing VDI.

**Signature:**

```
(VDI ref) copy (session_id s, VDI ref vdi, SR ref sr, VDI ref base_vdi, VDI ref into_vdi)
```

**Arguments:**

type	name	description
VDI ref	vdi	The VDI to copy
SR ref	sr	The destination SR (only required if the destination VDI is not specified)
VDI ref	base_vdi	The base VDI (only required if copying only changed blocks, by default all blocks will be copied)
VDI ref	into_vdi	The destination VDI to copy blocks into (if omitted then a destination SR must be provided and a fresh VDI will be created)

**Return Type:** VDI ref

The reference of the VDI where the blocks were written.

**Possible Error Codes:** VDI\_READONLY, VDI\_TOO\_SMALL, VDI\_NOT\_SPARSE

**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\_is\_a\_snapshot

**Overview:**

Sets whether this VDI is a snapshot.

**Signature:**

```
void set_is_a_snapshot (session_id s, VDI ref self, bool value)
```

**Arguments:**

type	name	description
VDI ref	self	The VDI to modify
bool	value	The new value indicating whether this VDI is a snapshot

**Return Type:** void

**RPC name: set\_snapshot\_of****Overview:**

Sets the VDI of which this VDI is a snapshot.

**Signature:**

```
void set_snapshot_of (session_id s, VDI ref self, VDI ref value)
```

**Arguments:**

type	name	description
VDI ref	self	The VDI to modify
VDI ref	value	The VDI of which this VDI is a snapshot

**Return Type:** void

**RPC name: set\_snapshot\_time****Overview:**

Sets the snapshot time of this VDI.

**Signature:**

```
void set_snapshot_time (session_id s, VDI ref self, datetime value)
```

**Arguments:**

type	name	description
VDI ref	self	The VDI to modify
datetime	value	The snapshot time of this VDI.

**Return Type:** void

**RPC name: set\_metadata\_of\_pool****Overview:**

Records the pool whose metadata is contained by this VDI.

**Signature:**

```
void set_metadata_of_pool (session_id s, VDI ref self, pool ref value)
```

**Arguments:**

type	name	description
VDI ref	self	The VDI to modify
pool ref	value	The pool whose metadata is contained by this VDI

**Return Type:** void

**RPC name:** set\_name\_label**Overview:**

Set the name label of the VDI. This can only happen when then its SR is currently attached.

**Signature:**

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

**Arguments:**

type	name	description
VDI ref	self	The VDI to modify
string	value	The name lable for the VDI

**Return Type:** void

**RPC name:** set\_name\_description**Overview:**

Set the name description of the VDI. This can only happen when its SR is currently attached.

**Signature:**

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

**Arguments:**

type	name	description
VDI ref	self	The VDI to modify
string	value	The name description for the VDI

**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: open\_database****Overview:**

Load the metadata found on the supplied VDI and return a session reference which can be used in XenAPI calls to query its contents.

**Signature:**

```
(session ref) open_database (session_id s, VDI ref self)
```

**Arguments:**

type	name	description
VDI ref	self	The VDI which contains the database to open

**Return Type:** session ref

A session which can be used to query the database

**RPC name: read\_database\_pool\_uuid****Overview:**

Check the VDI cache for the pool UUID of the database on this VDI.

**Signature:**

```
string read_database_pool_uuid (session_id s, VDI ref self)
```

**Arguments:**

type	name	description
VDI ref	self	The metadata VDI to look up in the cache.

**Return Type:** string

The cached pool UUID of the database on the VDI.

**RPC name:** `pool_migrate`

**Overview:**

Migrate a VDI, which may be attached to a running guest, to a different SR. The destination SR must be visible to the guest.

**Signature:**

```
(VDI ref) pool_migrate (session_id s, VDI ref vdi, SR ref sr, (string -> string) Map options)
```

**Arguments:**

type	name	description
VDI ref	vdi	The VDI to migrate
SR ref	sr	The destination SR
(string → string) Map	options	Other parameters

**Return Type:** VDI ref

The new reference of the migrated VDI.

**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:** 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:** 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
<code>(string → string) Map</code>	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
<code>(string → string) Map</code>	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
<code>(string → string) Map</code>	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:** `get_metadata_of_pool`

**Overview:**

Get the `metadata_of_pool` field of the given VDI.

**Signature:**

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

**Arguments:**

type	name	description
VDI ref	self	reference to the object

**Return Type:** `pool ref`

value of the field

**RPC name:** `get_metadata_latest`

**Overview:**

Get the `metadata_latest` field of the given VDI.

**Signature:**

```
bool get_metadata_latest (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:** `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

## 2.33 Class: VBD

### 2.33.1 Fields for class: VBD

Name	VBD		
Description	<i>A virtual block device.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	uuid	string	Unique identifier/object reference
<i>RO<sub>run</sub></i>	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.
<i>RO<sub>run</sub></i>	current_operations	(string $\rightarrow$ 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.
<i>RO<sub>ins</sub></i>	VM	VM ref	the virtual machine
<i>RO<sub>ins</sub></i>	VDI	VDI ref	the virtual disk
<i>RO<sub>run</sub></i>	device	string	device seen by the guest e.g. hda1
<i>RW</i>	userdevice	string	user-friendly device name e.g. 0,1,2,etc.
<i>RW</i>	bootable	bool	true if this VBD is bootable
<i>RW</i>	mode	vbd_mode	the mode the VBD should be mounted with
<i>RW</i>	type	vbd_type	how the VBD will appear to the guest (e.g. disk or CD)
<i>RW</i>	unpluggable	bool	true if this VBD will support hot-unplug
<i>RO<sub>run</sub></i>	storage_lock	bool	true if a storage level lock was acquired
<i>RO<sub>ins</sub></i>	empty	bool	if true this represents an empty drive
<i>RO<sub>run</sub></i>	reserved	bool	true if the VBD is reserved pending a reboot/migrate
<i>RW</i>	other_config	(string $\rightarrow$ string) Map	additional configuration
<i>RO<sub>run</sub></i>	currently_attached	bool	is the device currently attached (erased on reboot)
<i>RO<sub>run</sub></i>	status_code	int	error/success code associated with last attach-operation (erased on reboot)
<i>RO<sub>run</sub></i>	status_detail	string	error/success information associated with last attach-operation status (erased on reboot)
<i>RO<sub>run</sub></i>	runtime_properties	(string $\rightarrow$ string) Map	Device runtime properties
<i>RW</i>	qos/algorithm.type	string	QoS algorithm to use
<i>RW</i>	qos/algorithm.params	(string $\rightarrow$ string) Map	parameters for chosen QoS algorithm
<i>RO<sub>run</sub></i>	qos/supported.algorithms	string Set	supported QoS algorithms for this VBD
<i>RO<sub>run</sub></i>	metrics	VBD_metrics ref	metrics associated with this VBD

### 2.33.2 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  $\rightarrow$  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
<code>(string → string) Map</code>	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

## 2.34 Class: VBD\_metrics

### 2.34.1 Fields for class: VBD\_metrics

Name	VBD_metrics		
Description	<i>The metrics associated with a virtual block device.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	<code>uuid</code>	string	Unique identifier/object reference
<i>RO<sub>run</sub></i>	<code>io/read_kbs</code>	float	Read bandwidth (KiB/s)
<i>RO<sub>run</sub></i>	<code>io/write_kbs</code>	float	Write bandwidth (KiB/s)
<i>RO<sub>run</sub></i>	<code>last_updated</code>	datetime	Time at which this information was last updated
<i>RW</i>	<code>other_config</code>	(string → string) Map	additional configuration

### 2.34.2 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
<code>VBD_metrics ref</code>	<code>self</code>	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
<code>VBD_metrics ref</code>	<code>self</code>	reference to the object
<code>(string → string) Map</code>	<code>value</code>	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
<code>VBD_metrics ref</code>	<code>self</code>	reference to the object
<code>string</code>	<code>key</code>	Key to add
<code>string</code>	<code>value</code>	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
<code>VBD_metrics ref</code>	<code>self</code>	reference to the object
<code>string</code>	<code>key</code>	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
<code>string</code>	<code>uuid</code>	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
<code>VBD_metrics ref</code>	<code>self</code>	reference to the object

**Return Type:** `VBD_metrics record`  
all fields from the object

## 2.35 Class: PBD

### 2.35.1 Fields for class: PBD

Name	PBD		
Description	<i>The physical block devices through which hosts access SRs.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	uuid	string	Unique identifier/object reference
<i>RO<sub>ins</sub></i>	host	host ref	physical machine on which the pbd is available
<i>RO<sub>ins</sub></i>	SR	SR ref	the storage repository that the pbd realises
<i>RO<sub>ins</sub></i>	device_config	(string → string) Map	a config string to string map that is provided to the host's SR-backend-driver
<i>RO<sub>run</sub></i>	currently_attached	bool	is the SR currently attached on this host?
<i>RW</i>	other_config	(string → string) Map	additional configuration

### 2.35.2 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 no longer 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 <code>device_config</code>

**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
<code>(string → string) Map</code>	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



## 2.36 Class: crashdump

### 2.36.1 Fields for class: crashdump

Name	<b>crashdump</b>		
Description	<i>A VM crashdump.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	uuid	string	Unique identifier/object reference
<i>RO<sub>ins</sub></i>	VM	VM ref	the virtual machine
<i>RO<sub>ins</sub></i>	VDI	VDI ref	the virtual disk
<i>RW</i>	other_config	(string → string) Map	additional configuration

### 2.36.2 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
<code>(string → string) Map</code>	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

## 2.37 Class: VTPM

### 2.37.1 Fields for class: VTPM

Name	<b>VTPM</b>		
Description	<i>A virtual TPM device.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	uuid	string	Unique identifier/object reference
<i>RO<sub>ins</sub></i>	VM	VM ref	the virtual machine
<i>RO<sub>ins</sub></i>	backend	VM ref	the domain where the backend is located

### 2.37.2 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

## 2.38 Class: console

### 2.38.1 Fields for class: console

Name	<b>console</b>		
Description	<i>A console.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	<code>uuid</code>	string	Unique identifier/object reference
<i>RO<sub>run</sub></i>	<code>protocol</code>	console_protocol	the protocol used by this console
<i>RO<sub>run</sub></i>	<code>location</code>	string	URI for the console service
<i>RO<sub>run</sub></i>	<code>VM</code>	VM ref	VM to which this console is attached
<i>RW</i>	<code>other_config</code>	(string → string) Map	additional configuration
<i>RW</i>	<code>port</code>	int	port in dom0 on which the console server is listening

### 2.38.2 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
<code>(string → string) Map</code>	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

## 2.39 Class: user

### 2.39.1 Fields for class: user

Name	<b>user</b>		
Description	<i>A user of the system.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	<b>uuid</b>	string	Unique identifier/object reference
<i>RO<sub>ins</sub></i>	<b>short_name</b>	string	short name (e.g. userid)
<i>RW</i>	<b>fullname</b>	string	full name
<i>RW</i>	<b>other_config</b>	(string → string) Map	additional configuration

### 2.39.2 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



## 2.40 Class: data\_source

### 2.40.1 Fields for class: data\_source

Name	<b>data_source</b>		
Description	<i>Data sources for logging in RRDs.</i>		
Quals	Field	Type	Description
<i>RO_run</i>	<b>name/label</b>	string	a human-readable name
<i>RO_run</i>	<b>name/description</b>	string	a notes field containing human-readable description
<i>RO_run</i>	<b>enabled</b>	bool	true if the data source is being logged
<i>RO_run</i>	<b>standard</b>	bool	true if the data source is enabled by default. Non-default data sources cannot be disabled
<i>RO_run</i>	<b>units</b>	string	the units of the value
<i>RO_run</i>	<b>min</b>	float	the minimum value of the data source
<i>RO_run</i>	<b>max</b>	float	the maximum value of the data source
<i>RO_run</i>	<b>value</b>	float	current value of the data source

### 2.40.2 RPCs associated with class: data\_source

Class `data_source` has no additional RPCs associated with it.

## 2.41 Class: blob

### 2.41.1 Fields for class: blob

Name	<b>blob</b>		
Description	<i>A placeholder for a binary blob.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	<b>uuid</b>	string	Unique identifier/object reference
<i>RW</i>	<b>name/label</b>	string	a human-readable name
<i>RW</i>	<b>name/description</b>	string	a notes field containing human-readable description
<i>RO<sub>run</sub></i>	<b>size</b>	int	Size of the binary data, in bytes
<i>RW</i>	<b>public</b>	bool	True if the blob is publicly accessible
<i>RO<sub>ins</sub></i>	<b>last_updated</b>	datetime	Time at which the data in the blob was last updated
<i>RO<sub>ins</sub></i>	<b>mime_type</b>	string	The mime type associated with this object. Defaults to 'application/octet-stream' if the empty string is supplied

### 2.41.2 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, bool public)
```

**Arguments:**

type	name	description
string	mime_type	The mime-type of the blob. Defaults to 'application/octet-stream' if the empty string is supplied
bool	public	True if the blob should be publicly available

**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_public`

**Overview:**

Get the public field of the given blob.

**Signature:**

```
bool get_public (session_id s, blob ref self)
```

**Arguments:**

type	name	description
blob ref	self	reference to the object

**Return Type:** `bool`

value of the field

**RPC name:** `set_public`

**Overview:**

Set the public field of the given blob.

**Signature:**

```
void set_public (session_id s, blob ref self, bool value)
```

**Arguments:**

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

**Return Type:** `void`

**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

## 2.42 Class: message

### 2.42.1 Fields for class: message

Name	<b>message</b>		
Description	<i>An message for the attention of the administrator.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	<b>uuid</b>	string	Unique identifier/object reference
<i>RO<sub>run</sub></i>	<b>name</b>	string	The name of the message
<i>RO<sub>run</sub></i>	<b>priority</b>	int	The message priority, 0 being low priority
<i>RO<sub>run</sub></i>	<b>cls</b>	cls	The class of the object this message is associated with
<i>RO<sub>run</sub></i>	<b>obj_uuid</b>	string	The uuid of the object this message is associated with
<i>RO<sub>run</sub></i>	<b>timestamp</b>	datetime	The time at which the message was created
<i>RO<sub>run</sub></i>	<b>body</b>	string	The body of the message

### 2.42.2 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  $\rightarrow$  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  $\rightarrow$  message record) Map  
The messages

## 2.43 Class: secret

### 2.43.1 Fields for class: secret

Name	<b>secret</b>		
Description	<i>A secret.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	<b>uuid</b>	string	Unique identifier/object reference
<i>RW</i>	<b>value</b>	string	the secret
<i>RW</i>	<b>other_config</b>	(string → string) Map	other_config

### 2.43.2 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



## 2.44 Class: tunnel

### 2.44.1 Fields for class: tunnel

Name	<b>tunnel</b>		
Description	<i>A tunnel for network traffic.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	<code>uuid</code>	string	Unique identifier/object reference
<i>RO<sub>ins</sub></i>	<code>access_PIF</code>	PIF ref	The interface through which the tunnel is accessed
<i>RO<sub>ins</sub></i>	<code>transport_PIF</code>	PIF ref	The interface used by the tunnel
<i>RW</i>	<code>status</code>	(string → string) Map	Status information about the tunnel
<i>RW</i>	<code>other_config</code>	(string → string) Map	Additional configuration

### 2.44.2 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
<code>(string → string) Map</code>	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

## 2.45 Class: PCI

### 2.45.1 Fields for class: PCI

Name	PCI		
Description	<i>A PCI device.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	<code>uuid</code>	string	Unique identifier/object reference
<i>RO<sub>ins</sub></i>	<code>class_id</code>	string	PCI class ID
<i>RO<sub>ins</sub></i>	<code>class_name</code>	string	PCI class name
<i>RO<sub>ins</sub></i>	<code>vendor_id</code>	string	Vendor ID
<i>RO<sub>ins</sub></i>	<code>vendor_name</code>	string	Vendor name
<i>RO<sub>ins</sub></i>	<code>device_id</code>	string	Device ID
<i>RO<sub>ins</sub></i>	<code>device_name</code>	string	Device name
<i>RO<sub>ins</sub></i>	<code>host</code>	host ref	Physical machine that owns the PCI device
<i>RO<sub>ins</sub></i>	<code>pci_id</code>	string	PCI ID of the physical device
<i>RO<sub>run</sub></i>	<code>functions</code>	int	Number of physical + virtual PCI functions
<i>RO<sub>run</sub></i>	<code>attached_VMs</code>	(VM ref) Set	VMs that currently have a function of this PCI device passed-through to them
<i>RO<sub>run</sub></i>	<code>dependencies</code>	(PCI ref) Set	List of dependent PCI devices
<i>RW</i>	<code>other_config</code>	(string $\rightarrow$ string) Map	Additional configuration
<i>RO<sub>ins</sub></i>	<code>subsystem_vendor_id</code>	string	Subsystem vendor ID
<i>RO<sub>ins</sub></i>	<code>subsystem_vendor_name</code>	string	Subsystem vendor name
<i>RO<sub>ins</sub></i>	<code>subsystem_device_id</code>	string	Subsystem device ID
<i>RO<sub>ins</sub></i>	<code>subsystem_device_name</code>	string	Subsystem device name

### 2.45.2 RPCs associated with class: PCI

**RPC name:** `get_all`

**Overview:**

Return a list of all the PCIs known to the system.

**Signature:**

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

**Return Type:** (PCI ref) Set

references to all objects

**RPC name:** `get_all_records`

**Overview:**

Return a map of PCI references to PCI records for all PCIs known to the system.

**Signature:**

```
((PCI ref  $\rightarrow$  PCI record) Map) get_all_records (session_id s)
```

**Return Type:** (PCI ref  $\rightarrow$  PCI record) Map

records of all objects

**RPC name:** `get_uuid`

**Overview:**

Get the `uuid` field of the given PCI.

**Signature:**

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

**Arguments:**

type	name	description
PCI ref	self	reference to the object

**Return Type:** `string`

value of the field

**RPC name:** `get_class_name`

**Overview:**

Get the `class_name` field of the given PCI.

**Signature:**

```
string get_class_name (session_id s, PCI ref self)
```

**Arguments:**

type	name	description
PCI 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 PCI.

**Signature:**

```
string get_vendor_name (session_id s, PCI ref self)
```

**Arguments:**

type	name	description
PCI 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 PCI.

**Signature:**

```
string get_device_name (session_id s, PCI ref self)
```

**Arguments:**

type	name	description
PCI 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 PCI.

**Signature:**

```
(host ref) get_host (session_id s, PCI ref self)
```

**Arguments:**

type	name	description
PCI ref	self	reference to the object

**Return Type:** `host ref`

value of the field

**RPC name:** `get_pci_id`

**Overview:**

Get the `pci_id` field of the given PCI.

**Signature:**

```
string get_pci_id (session_id s, PCI ref self)
```

**Arguments:**

type	name	description
PCI ref	self	reference to the object

**Return Type:** `string`

value of the field

**RPC name:** `get_dependencies`

**Overview:**

Get the dependencies field of the given PCI.

**Signature:**

```
((PCI ref) Set) get_dependencies (session_id s, PCI ref self)
```

**Arguments:**

type	name	description
PCI ref	self	reference to the object

**Return Type:** (PCI ref) Set  
value of the field

**RPC name:** `get_other_config`

**Overview:**

Get the other\_config field of the given PCI.

**Signature:**

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

**Arguments:**

type	name	description
PCI 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 PCI.

**Signature:**

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

**Arguments:**

type	name	description
PCI 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 PCI.

**Signature:**

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

**Arguments:**

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

**Signature:**

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

**Arguments:**

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

**Return Type:** void

**RPC name:** get\_subsystem\_vendor\_name

**Overview:**

Get the subsystem\_vendor\_name field of the given PCI.

**Signature:**

```
string get_subsystem_vendor_name (session_id s, PCI ref self)
```

**Arguments:**

type	name	description
PCI ref	self	reference to the object

**Return Type:** string  
value of the field

**RPC name:** `get_subsystem_device_name`

**Overview:**

Get the `subsystem_device_name` field of the given PCI.

**Signature:**

```
string get_subsystem_device_name (session_id s, PCI ref self)
```

**Arguments:**

type	name	description
PCI ref	self	reference to the object

**Return Type:** `string`

value of the field

**RPC name:** `get_by_uuid`

**Overview:**

Get a reference to the PCI instance with the specified UUID.

**Signature:**

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

**Arguments:**

type	name	description
string	uuid	UUID of object to return

**Return Type:** `PCI ref`

reference to the object

**RPC name:** `get_record`

**Overview:**

Get a record containing the current state of the given PCI.

**Signature:**

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

**Arguments:**

type	name	description
PCI ref	self	reference to the object

**Return Type:** `PCI record`

all fields from the object

## 2.46 Class: PGPU

### 2.46.1 Fields for class: PGPU

Name	PGPU		
Description	<i>A physical GPU (pGPU).</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	uuid	string	Unique identifier/object reference
<i>RO<sub>ins</sub></i>	PCI	PCI ref	Link to underlying PCI device
<i>RO<sub>ins</sub></i>	GPU_group	GPU_group ref	GPU group the pGPU is contained in
<i>RO<sub>run</sub></i>	host	host ref	Host that own the GPU
<i>RW</i>	other_config	(string → string) Map	Additional configuration
<i>RO<sub>run</sub></i>	supported_VGPU_types	(VGPU_type ref) Set	List of VGPU types supported by the underlying hardware
<i>RO<sub>run</sub></i>	enabled_VGPU_types	(VGPU_type ref) Set	List of VGPU types which have been enabled for this PGPU
<i>RO<sub>run</sub></i>	resident_VGPUs	(VGPU ref) Set	List of VGPUs running on this PGPU
<i>RO<sub>ins</sub></i>	size	int	Abstract size of this PGPU
<i>RO<sub>run</sub></i>	supported_VGPU_max_capacities	(VGPU_type ref → int) Map	A map relating each VGPU type supported on this GPU to the maximum number of VGPUs of that type which can run simultaneously on this GPU

### 2.46.2 RPCs associated with class: PGPU

RPC name: add\_enabled\_VGPU\_types

Overview:

.

Signature:

```
void add_enabled_VGPU_types (session_id s, PGPU ref self, VGPU_type ref value)
```

Arguments:

type	name	description
PGPU ref	self	The PGPU to which we are adding an enabled VGPU type
VGPU_type ref	value	The VGPU type to enable

Return Type: void

RPC name: remove\_enabled\_VGPU\_types

Overview:

.

Signature:

```
void remove_enabled_VGPU_types (session_id s, PGPU ref self, VGPU_type ref value)
```

Arguments:

type	name	description
PGPU ref	self	The PGPU from which we are removing an enabled VGPU type
VGPU_type ref	value	The VGPU type to disable

**Return Type:** void

**RPC name:** set\_enabled\_VGPU\_types

**Overview:**

.

**Signature:**

```
void set_enabled_VGPU_types (session_id s, PGPU ref self, (VGPU_type ref) Set value)
```

**Arguments:**

type	name	description
PGPU ref	self	The PGPU on which we are enabling a set of VGPU types
(VGPU_type ref) Set	value	The VGPU types to enable

**Return Type:** void

**RPC name:** set\_GPU\_group

**Overview:**

.

**Signature:**

```
void set_GPU_group (session_id s, PGPU ref self, GPU_group ref value)
```

**Arguments:**

type	name	description
PGPU ref	self	The PGPU to move to a new group
GPU_group ref	value	The group to which the PGPU will be moved

**Return Type:** void

**RPC name:** get\_remaining\_capacity

**Overview:**

.

**Signature:**

```
int get_remaining_capacity (session_id s, PGPU ref self, VGPU_type ref vgpu_type)
```

**Arguments:**

type	name	description
PGPU ref	self	The PGPU to query
VGPU_type ref	vgpu_type	The VGPU type for which we want to find the number of VGPU types which can still be started on this PGPU

**Return Type:** int

The number of VGPUs of the specified type which can still be started on this PGPU

**RPC name:** `get_all`

**Overview:**

Return a list of all the PGPUs known to the system.

**Signature:**

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

**Return Type:** (PGPU ref) Set

references to all objects

**RPC name:** `get_all_records`

**Overview:**

Return a map of PGPU references to PGPU records for all PGPUs known to the system.

**Signature:**

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

**Return Type:** (PGPU ref  $\rightarrow$  PGPU record) Map

records of all objects

**RPC name:** `get_uuid`

**Overview:**

Get the uuid field of the given PGPU.

**Signature:**

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

**Arguments:**

type	name	description
PGPU ref	self	reference to the object

**Return Type:** string

value of the field

**RPC name:** `get_PCI`

**Overview:**

Get the PCI field of the given PGPU.

**Signature:**

```
(PCI ref) get_PCI (session_id s, PGPU ref self)
```

**Arguments:**

type	name	description
PGPU ref	self	reference to the object

**Return Type:** PCI ref

value of the field

**RPC name:** get\_GPU\_group**Overview:**

Get the GPU\_group field of the given PGPU.

**Signature:**

```
(GPU_group ref) get_GPU_group (session_id s, PGPU ref self)
```

**Arguments:**

type	name	description
PGPU ref	self	reference to the object

**Return Type:** GPU\_group ref

value of the field

**RPC name:** get\_host**Overview:**

Get the host field of the given PGPU.

**Signature:**

```
(host ref) get_host (session_id s, PGPU ref self)
```

**Arguments:**

type	name	description
PGPU ref	self	reference to the object

**Return Type:** host ref

value of the field

**RPC name:** get\_other\_config**Overview:**

Get the other\_config field of the given PGPU.

**Signature:**

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

**Arguments:**

type	name	description
PGPU ref	self	reference to the object



**Return Type:**  $(\text{string} \rightarrow \text{string})$  Map  
value of the field

**RPC name:** `set_other_config`

**Overview:**

Set the `other_config` field of the given PGPU.

**Signature:**

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

**Arguments:**

type	name	description
PGPU ref	self	reference to the object
$(\text{string} \rightarrow \text{string})$ 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 PGPU.

**Signature:**

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

**Arguments:**

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

**Signature:**

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

**Arguments:**

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

**Return Type:** void

**RPC name:** `get_supported_VGPU_types`

**Overview:**

Get the `supported_VGPU_types` field of the given PGPU.

**Signature:**

```
((VGPU_type ref) Set) get_supported_VGPU_types (session_id s, PGPU ref self)
```

**Arguments:**

type	name	description
PGPU ref	self	reference to the object

**Return Type:** (VGPU\_type ref) Set  
value of the field

**RPC name:** `get_enabled_VGPU_types`

**Overview:**

Get the `enabled_VGPU_types` field of the given PGPU.

**Signature:**

```
((VGPU_type ref) Set) get_enabled_VGPU_types (session_id s, PGPU ref self)
```

**Arguments:**

type	name	description
PGPU ref	self	reference to the object

**Return Type:** (VGPU\_type ref) Set  
value of the field

**RPC name:** `get_resident_VGPUs`

**Overview:**

Get the `resident_VGPUs` field of the given PGPU.

**Signature:**

```
((VGPU ref) Set) get_resident_VGPUs (session_id s, PGPU ref self)
```

**Arguments:**

type	name	description
PGPU ref	self	reference to the object

**Return Type:** (VGPU ref) Set  
value of the field

**RPC name:** `get_supported_VGPU_max_capacities`

**Overview:**

Get the `supported_VGPU_max_capacities` field of the given PGPU.

**Signature:**

```
((VGPU_type ref -> int) Map) get_supported_VGPU_max_capacities (session_id s, PGPU ref self)
```

**Arguments:**

type	name	description
PGPU ref	self	reference to the object

**Return Type:** `(VGPU_type ref → int) Map`  
value of the field

**RPC name:** `get_by_uuid`

**Overview:**

Get a reference to the PGPU instance with the specified UUID.

**Signature:**

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

**Arguments:**

type	name	description
string	uuid	UUID of object to return

**Return Type:** `PGPU ref`  
reference to the object

**RPC name:** `get_record`

**Overview:**

Get a record containing the current state of the given PGPU.

**Signature:**

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

**Arguments:**

type	name	description
PGPU ref	self	reference to the object

**Return Type:** `PGPU record`  
all fields from the object

## 2.47 Class: GPU\_group

### 2.47.1 Fields for class: GPU\_group

Name	<b>GPU_group</b>		
Description	<i>A group of compatible GPUs across the resource pool.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	uuid	string	Unique identifier/object reference
<i>RW</i>	name/label	string	a human-readable name
<i>RW</i>	name/description	string	a notes field containing human-readable description
<i>RO<sub>run</sub></i>	PGPUs	(PGPU ref) Set	List of pGPUs in the group
<i>RO<sub>run</sub></i>	VGPUs	(VGPU ref) Set	List of vGPUs using the group
<i>RO<sub>run</sub></i>	GPU_types	string Set	List of GPU types (vendor+device ID) that can be in this group
<i>RW</i>	other_config	(string $\rightarrow$ string) Map	Additional configuration
<i>RW</i>	allocation_algorithm	allocation_algorithm	Current allocation of vGPUs to pGPUs for this group
<i>RO<sub>run</sub></i>	supported_VGPU_types	(VGPU_type ref) Set	vGPU types supported on at least one of the pGPUs in this group
<i>RO<sub>run</sub></i>	enabled_VGPU_types	(VGPU_type ref) Set	vGPU types supported on at least one of the pGPUs in this group

### 2.47.2 RPCs associated with class: GPU\_group

RPC name: create

Overview:

.

Signature:

(GPU\_group ref) create (session\_id s, string name\_label, string name\_description, (string  $\rightarrow$  string) M

Arguments:

type	name	description
string	name_label	
string	name_description	
(string $\rightarrow$ string) Map	other_config	

Return Type: GPU\_group ref

RPC name: destroy

Overview:

.

Signature:

void destroy (session\_id s, GPU\_group ref self)

Arguments:

type	name	description
GPU_group ref	self	The vGPU to destroy

**Return Type:** void

**RPC name:** get\_remaining\_capacity

**Overview:**

.

**Signature:**

```
int get_remaining_capacity (session_id s, GPU_group ref self, VGPU_type ref vgpu_type)
```

**Arguments:**

type	name	description
GPU_group ref	self	The GPU group to query
VGPU_type ref	vgpu_type	The VGPU_type for which the remaining capacity will be calculated

**Return Type:** int

The number of VGPUs of the given type which can still be started on the PGPUs in the group

**RPC name:** get\_all

**Overview:**

Return a list of all the GPU\_groups known to the system.

**Signature:**

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

**Return Type:** (GPU\_group ref) Set

references to all objects

**RPC name:** get\_all\_records

**Overview:**

Return a map of GPU\_group references to GPU\_group records for all GPU\_groups known to the system.

**Signature:**

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

**Return Type:** (GPU\_group ref  $\rightarrow$  GPU\_group record) Map

records of all objects

**RPC name:** `get_uuid`

**Overview:**

Get the uuid field of the given GPU\_group.

**Signature:**

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

**Arguments:**

type	name	description
GPU_group 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 GPU\_group.

**Signature:**

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

**Arguments:**

type	name	description
GPU_group 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 GPU\_group.

**Signature:**

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

**Arguments:**

type	name	description
GPU_group 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 GPU\_group.

**Signature:**

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

**Arguments:**

type	name	description
GPU_group 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 GPU\_group.

**Signature:**

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

**Arguments:**

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

**Return Type:** void

**RPC name:** `get_PGPUs`

**Overview:**

Get the PGPUs field of the given GPU\_group.

**Signature:**

```
((PGPU ref) Set) get_PGPUs (session_id s, GPU_group ref self)
```

**Arguments:**

type	name	description
GPU_group ref	self	reference to the object

**Return Type:** (PGPU ref) Set

value of the field

**RPC name:** `get_VGPUs`

**Overview:**

Get the VGPUs field of the given GPU\_group.

**Signature:**

```
((VGPU ref) Set) get_VGPUs (session_id s, GPU_group ref self)
```

**Arguments:**

type	name	description
GPU_group ref	self	reference to the object

**Return Type:** (VGPU ref) Set  
value of the field

**RPC name:** `get_GPU_types`

**Overview:**

Get the GPU\_types field of the given GPU\_group.

**Signature:**

```
(string Set) get_GPU_types (session_id s, GPU_group ref self)
```

**Arguments:**

type	name	description
GPU_group 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 GPU\_group.

**Signature:**

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

**Arguments:**

type	name	description
GPU_group 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 `GPU_group`.

**Signature:**

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

**Arguments:**

type	name	description
<code>GPU_group ref</code>	<code>self</code>	reference to the object
<code>(string → string) Map</code>	<code>value</code>	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 `GPU_group`.

**Signature:**

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

**Arguments:**

type	name	description
<code>GPU_group ref</code>	<code>self</code>	reference to the object
<code>string</code>	<code>key</code>	Key to add
<code>string</code>	<code>value</code>	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 `GPU_group`.  
If the key is not in that Map, then do nothing.

**Signature:**

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

**Arguments:**

type	name	description
<code>GPU_group ref</code>	<code>self</code>	reference to the object
<code>string</code>	<code>key</code>	Key to remove

**Return Type:** `void`

**RPC name:** `get_allocation_algorithm`

**Overview:**

Get the `allocation_algorithm` field of the given `GPU_group`.

**Signature:**

```
(allocation_algorithm) get_allocation_algorithm (session_id s, GPU_group ref self)
```

**Arguments:**

type	name	description
GPU_group ref	self	reference to the object

**Return Type:** `allocation_algorithm`  
value of the field

**RPC name:** `set_allocation_algorithm`

**Overview:**

Set the `allocation_algorithm` field of the given `GPU_group`.

**Signature:**

```
void set_allocation_algorithm (session_id s, GPU_group ref self, allocation_algorithm value)
```

**Arguments:**

type	name	description
GPU_group ref	self	reference to the object
allocation_algorithm	value	New value to set

**Return Type:** `void`

**RPC name:** `get_supported_VGPU_types`

**Overview:**

Get the `supported_VGPU_types` field of the given `GPU_group`.

**Signature:**

```
((VGPU_type ref) Set) get_supported_VGPU_types (session_id s, GPU_group ref self)
```

**Arguments:**

type	name	description
GPU_group ref	self	reference to the object

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

**RPC name:** `get_enabled_VGPU_types`

**Overview:**

Get the `enabled_VGPU_types` field of the given `GPU_group`.

**Signature:**

```
((VGPU_type ref) Set) get_enabled_VGPU_types (session_id s, GPU_group ref self)
```

**Arguments:**

type	name	description
GPU_group ref	self	reference to the object

**Return Type:** (VGPU\_type ref) Set  
value of the field

**RPC name:** `get_by_uuid`

**Overview:**

Get a reference to the `GPU_group` instance with the specified UUID.

**Signature:**

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

**Arguments:**

type	name	description
string	uuid	UUID of object to return

**Return Type:** GPU\_group ref  
reference to the object

**RPC name:** `get_record`

**Overview:**

Get a record containing the current state of the given `GPU_group`.

**Signature:**

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

**Arguments:**

type	name	description
GPU_group ref	self	reference to the object

**Return Type:** GPU\_group record  
all fields from the object

**RPC name:** `get_by_name_label`

**Overview:**

Get all the GPU\_group instances with the given label.

**Signature:**

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

**Arguments:**

type	name	description
string	label	label of object to return

**Return Type:** (GPU\_group ref) Set  
references to objects with matching names

## 2.48 Class: VGPU

### 2.48.1 Fields for class: VGPU

Name	<b>VGPU</b>		
Description	<i>A virtual GPU (vGPU).</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	uuid	string	Unique identifier/object reference
<i>RO<sub>run</sub></i>	VM	VM ref	VM that owns the vGPU
<i>RO<sub>run</sub></i>	GPU_group	GPU_group ref	GPU group used by the vGPU
<i>RO<sub>run</sub></i>	device	string	Order in which the devices are plugged into the VM
<i>RO<sub>run</sub></i>	currently_attached	bool	Reflects whether the virtual device is currently connected to a physical device
<i>RW</i>	other_config	(string → string) Map	Additional configuration
<i>RO<sub>run</sub></i>	type	VGPU_type ref	Preset type for this VGPU
<i>RO<sub>run</sub></i>	resident_on	PGPU ref	The PGPU on which this VGPU is running

### 2.48.2 RPCs associated with class: VGPU

**RPC name:** create

**Overview:**

.

**Signature:**

(VGPU ref) create (session\_id s, VM ref VM, GPU\_group ref GPU\_group, string device, (string → string)

**Arguments:**

type	name	description
VM ref	VM	
GPU_group ref	GPU_group	
string	device	
(string → string) Map	other_config	
VGPU_type ref	type	

**Return Type:** VGPU ref

reference to the newly created object

**RPC name:** destroy

**Overview:**

.

**Signature:**

void destroy (session\_id s, VGPU ref self)

**Arguments:**

type	name	description
VGPU ref	self	The vGPU to destroy

**Return Type:** void

**RPC name:** get\_all

**Overview:**

Return a list of all the VGPU s known to the system.

**Signature:**

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

**Return Type:** (VGPU ref) Set

references to all objects

**RPC name:** get\_all\_records

**Overview:**

Return a map of VGPU references to VGPU records for all VGPU s known to the system.

**Signature:**

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

**Return Type:** (VGPU ref  $\rightarrow$  VGPU record) Map

records of all objects

**RPC name:** get\_uuid

**Overview:**

Get the uuid field of the given VGPU.

**Signature:**

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

**Arguments:**

type	name	description
VGPU 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 VGPU.

**Signature:**

```
(VM ref) get_VM (session_id s, VGPU ref self)
```

**Arguments:**

type	name	description
VGPU ref	self	reference to the object

**Return Type:** VM ref

value of the field

**RPC name:** get\_GPU\_group**Overview:**

Get the GPU\_group field of the given VGPU.

**Signature:**

```
(GPU_group ref) get_GPU_group (session_id s, VGPU ref self)
```

**Arguments:**

type	name	description
VGPU ref	self	reference to the object

**Return Type:** GPU\_group ref

value of the field

**RPC name:** get\_device**Overview:**

Get the device field of the given VGPU.

**Signature:**

```
string get_device (session_id s, VGPU ref self)
```

**Arguments:**

type	name	description
VGPU ref	self	reference to the object

**Return Type:** string

value of the field

**RPC name:** get\_currently\_attached**Overview:**

Get the currently\_attached field of the given VGPU.

**Signature:**

```
bool get_currently_attached (session_id s, VGPU ref self)
```

**Arguments:**

type	name	description
VGPU 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 VGPU.

**Signature:**

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

**Arguments:**

type	name	description
VGPU 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 VGPU.

**Signature:**

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

**Arguments:**

type	name	description
VGPU 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 VGPU.

**Signature:**

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

**Arguments:**

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

**Signature:**

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

**Arguments:**

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

**Return Type:** void

**RPC name:** `get_type`**Overview:**

Get the type field of the given VGPU.

**Signature:**

```
(VGPU_type ref) get_type (session_id s, VGPU ref self)
```

**Arguments:**

type	name	description
VGPU ref	self	reference to the object

**Return Type:** VGPU\_type ref  
value of the field

**RPC name:** `get_resident_on`**Overview:**

Get the resident\_on field of the given VGPU.

**Signature:**

```
(PGPU ref) get_resident_on (session_id s, VGPU ref self)
```

**Arguments:**

type	name	description
VGPU ref	self	reference to the object

**Return Type:** PGPU ref  
value of the field

**RPC name:** `get_by_uuid`

**Overview:**

Get a reference to the VGPU instance with the specified UUID.

**Signature:**

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

**Arguments:**

type	name	description
string	uuid	UUID of object to return

**Return Type:** VGPU ref

reference to the object

**RPC name:** `get_record`

**Overview:**

Get a record containing the current state of the given VGPU.

**Signature:**

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

**Arguments:**

type	name	description
VGPU ref	self	reference to the object

**Return Type:** VGPU record

all fields from the object

## 2.49 Class: VGPU\_type

### 2.49.1 Fields for class: VGPU\_type

Name	VGPU_type		
Description	<i>A type of virtual GPU.</i>		
Quals	Field	Type	Description
<i>RO<sub>run</sub></i>	<code>uuid</code>	string	Unique identifier/object reference
<i>RO<sub>ins</sub></i>	<code>vendor_name</code>	string	Name of VGPU vendor
<i>RO<sub>ins</sub></i>	<code>model_name</code>	string	Model name associated with the VGPU type
<i>RO<sub>ins</sub></i>	<code>framebuffer_size</code>	int	Framebuffer size of the VGPU type, in bytes
<i>RO<sub>ins</sub></i>	<code>max_heads</code>	int	Maximum number of displays supported by the VGPU type
<i>RO<sub>ins</sub></i>	<code>max_resolution_x</code>	int	Maximum resolution (width) supported by the VGPU type
<i>RO<sub>ins</sub></i>	<code>max_resolution_y</code>	int	Maximum resolution (height) supported by the VGPU type
<i>RO<sub>ins</sub></i>	<code>size</code>	int	Abstract size for tracking PGPU utilisation
<i>RO<sub>run</sub></i>	<code>supported_on_PGPUs</code>	(PGPU ref) Set	List of PGPUs that support this VGPU type
<i>RO<sub>run</sub></i>	<code>enabled_on_PGPUs</code>	(PGPU ref) Set	List of PGPUs that have this VGPU type enabled
<i>RO<sub>run</sub></i>	<code>VGPUs</code>	(VGPU ref) Set	List of VGPUs of this type
<i>RO<sub>ins</sub></i>	<code>internal_config</code>	(string → string) Map	Extra configuration information for internal use.
<i>RO<sub>run</sub></i>	<code>supported_on_GPU_groups</code>	(GPU_group ref) Set	List of GPU groups in which at least one PGPU supports this VGPU type
<i>RO<sub>run</sub></i>	<code>enabled_on_GPU_groups</code>	(GPU_group ref) Set	List of GPU groups in which at least one have this VGPU type enabled

### 2.49.2 RPCs associated with class: VGPU\_type

**RPC name:** `get_all`

**Overview:**

Return a list of all the VGPU\_types known to the system.

**Signature:**

((VGPU\_type ref) Set) `get_all` (session\_id s)

**Return Type:** (VGPU\_type ref) Set

references to all objects

**RPC name:** `get_all_records`

**Overview:**

Return a map of VGPU\_type references to VGPU\_type records for all VGPU\_types known to the system.

**Signature:**

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

**Return Type:** (VGPU\_type ref  $\rightarrow$  VGPU\_type record) Map  
records of all objects

**RPC name:** get\_uuid

**Overview:**

Get the uuid field of the given VGPU\_type.

**Signature:**

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

**Arguments:**

type	name	description
VGPU_type 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 VGPU\_type.

**Signature:**

```
string get_vendor_name (session_id s, VGPU_type ref self)
```

**Arguments:**

type	name	description
VGPU_type ref	self	reference to the object

**Return Type:** string  
value of the field

**RPC name:** get\_model\_name

**Overview:**

Get the model\_name field of the given VGPU\_type.

**Signature:**

```
string get_model_name (session_id s, VGPU_type ref self)
```

**Arguments:**

type	name	description
VGPU_type ref	self	reference to the object

**Return Type:** string  
value of the field

**RPC name:** `get_framebuffer_size`

**Overview:**

Get the `framebuffer_size` field of the given `VGPU_type`.

**Signature:**

```
int get_framebuffer_size (session_id s, VGPU_type ref self)
```

**Arguments:**

type	name	description
VGPU_type ref	self	reference to the object

**Return Type:** `int`

value of the field

**RPC name:** `get_max_heads`

**Overview:**

Get the `max_heads` field of the given `VGPU_type`.

**Signature:**

```
int get_max_heads (session_id s, VGPU_type ref self)
```

**Arguments:**

type	name	description
VGPU_type ref	self	reference to the object

**Return Type:** `int`

value of the field

**RPC name:** `get_max_resolution_x`

**Overview:**

Get the `max_resolution_x` field of the given `VGPU_type`.

**Signature:**

```
int get_max_resolution_x (session_id s, VGPU_type ref self)
```

**Arguments:**

type	name	description
VGPU_type ref	self	reference to the object

**Return Type:** `int`

value of the field

**RPC name:** `get_max_resolution_y`

**Overview:**

Get the `max_resolution_y` field of the given `VGPU_type`.

**Signature:**

```
int get_max_resolution_y (session_id s, VGPU_type ref self)
```

**Arguments:**

type	name	description
VGPU_type ref	self	reference to the object

**Return Type:** `int`

value of the field

**RPC name:** `get_supported_on_PGPUs`

**Overview:**

Get the `supported_on_PGPUs` field of the given `VGPU_type`.

**Signature:**

```
((PGPU ref) Set) get_supported_on_PGPUs (session_id s, VGPU_type ref self)
```

**Arguments:**

type	name	description
VGPU_type ref	self	reference to the object

**Return Type:** `((PGPU ref) Set)`

value of the field

**RPC name:** `get_enabled_on_PGPUs`

**Overview:**

Get the `enabled_on_PGPUs` field of the given `VGPU_type`.

**Signature:**

```
((PGPU ref) Set) get_enabled_on_PGPUs (session_id s, VGPU_type ref self)
```

**Arguments:**

type	name	description
VGPU_type ref	self	reference to the object

**Return Type:** `((PGPU ref) Set)`

value of the field

**RPC name:** `get_VGPUs`

**Overview:**

Get the `VGPUs` field of the given `VGPU_type`.

**Signature:**

```
((VGPU ref) Set) get_VGPUs (session_id s, VGPU_type ref self)
```

**Arguments:**

type	name	description
VGPU_type ref	self	reference to the object

**Return Type:** (VGPU ref) Set  
value of the field

**RPC name:** `get_supported_on_GPU_groups`

**Overview:**

Get the `supported_on_GPU_groups` field of the given `VGPU_type`.

**Signature:**

```
((GPU_group ref) Set) get_supported_on_GPU_groups (session_id s, VGPU_type ref self)
```

**Arguments:**

type	name	description
VGPU_type ref	self	reference to the object

**Return Type:** (GPU\_group ref) Set  
value of the field

**RPC name:** `get_enabled_on_GPU_groups`

**Overview:**

Get the `enabled_on_GPU_groups` field of the given `VGPU_type`.

**Signature:**

```
((GPU_group ref) Set) get_enabled_on_GPU_groups (session_id s, VGPU_type ref self)
```

**Arguments:**

type	name	description
VGPU_type ref	self	reference to the object

**Return Type:** (GPU\_group ref) Set  
value of the field

**RPC name:** `get_by_uuid`

**Overview:**

Get a reference to the `VGPU_type` instance with the specified UUID.

**Signature:**

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

**Arguments:**

type	name	description
string	uuid	UUID of object to return

**Return Type:** `VGPU_type ref`

reference to the object

**RPC name:** `get_record`

**Overview:**

Get a record containing the current state of the given `VGPU_type`.

**Signature:**

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

**Arguments:**

type	name	description
<code>VGPU_type ref</code>	self	reference to the object

**Return Type:** `VGPU_type record`

all fields from the object



## 2.50 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` → `eSpeil Incorporated` to a Map, but it already contains the mapping `Customer` → `eSpeil Inc.`, and so the request has failed.

Each possible error code is documented in the following section.

### 2.50.1 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*)

---

**BRIDGE\_NOT\_AVAILABLE**

Could not find bridge required by VM.

**Signature:**

BRIDGE\_NOT\_AVAILABLE(*bridge*)

---

**CANNOT\_ADD\_TUNNEL\_TO\_BOND\_SLAVE**

This PIF is a bond slave and cannot have a tunnel on it.

**Signature:**

CANNOT\_ADD\_TUNNEL\_TO\_BOND\_SLAVE(*PIF*)

---

**CANNOT\_ADD\_VLAN\_TO\_BOND\_SLAVE**

This PIF is a bond slave and cannot have a VLAN on it.

**Signature:**

CANNOT\_ADD\_VLAN\_TO\_BOND\_SLAVE(*PIF*)

---

**CANNOT\_CHANGE\_PIF\_PROPERTIES**

This properties of this PIF cannot be changed. Only the properties of non-bonded physical PIFs, or bond masters can be changed.

**Signature:**

`CANNOT_CHANGE_PIF_PROPERTIES(PIF)`

---

**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\_DISASTER\_RECOVERY\_TASK**

The disaster recovery task could not be cleanly destroyed.

**Signature:**

`CANNOT_DESTROY_DISASTER_RECOVERY_TASK(reason)`

---

**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\_BOND\_SLAVE**

This PIF is a bond slave and cannot be plugged.

**Signature:**

`CANNOT_PLUG_BOND_SLAVE(PIF)`

---

**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)`

---

**COULD\_NOT\_FIND\_NETWORK\_INTERFACE\_WITH\_SPECIFIED\_DEVICE\_NAME\_AND\_MAC\_ADDRESS**

Could not find a network interface with the specified device name and MAC address.

**Signature:**

`COULD_NOT_FIND_NETWORK_INTERFACE_WITH_SPECIFIED_DEVICE_NAME_AND_MAC_ADDRESS(device, mac)`

---

**COULD\_NOT\_IMPORT\_DATABASE**

An error occurred while attempting to import a database from a metadata VDI

**Signature:**

`COULD_NOT_IMPORT_DATABASE(reason)`

---

**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)
```

---

**DISK\_VBD\_MUST\_BE\_READWRITE\_FOR\_HVM**

All VBDs of type 'disk' must be read/write for HVM guests

**Signature:**

```
DISK_VBD_MUST_BE_READWRITE_FOR_HVM(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.

---

**EVENT\_FROM\_TOKEN\_PARSE\_FAILURE**

The event.from token could not be parsed. Valid values include: ”, and a value returned from a previous event.from call.

**Signature:**

EVENT\_FROM\_TOKEN\_PARSE\_FAILURE(token)

---

**EVENT\_SUBSCRIPTION\_PARSE\_FAILURE**

The server failed to parse your event subscription. Valid values include: \*, class-name, class-name/object-reference.

**Signature:**

EVENT\_SUBSCRIPTION\_PARSE\_FAILURE(subscription)

---

**FEATURE\_REQUIRES\_HVM**

The VM is set up to use a feature that requires it to boot as HVM.

**Signature:**

FEATURE\_REQUIRES\_HVM(details)

---

**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)

---

**GPU\_GROUP\_CONTAINS\_NO\_PGPUS**

The GPU group does not contain any PGPUs.

**Signature:**

GPU\_GROUP\_CONTAINS\_NO\_PGPUS(gpu\_group)

---

**GPU\_GROUP\_CONTAINS\_PGPU**

The GPU group contains active PGPUs and cannot be deleted.

**Signature:**

GPU\_GROUP\_CONTAINS\_PGPU(pgpus)

---

**GPU\_GROUP\_CONTAINS\_VGPU**

The GPU group contains active VGPU's and cannot be deleted.

**Signature:**

GPU\_GROUP\_CONTAINS\_VGPU(vgpus)

---

**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\_CANNOT\_CHANGE\_BOND\_STATUS\_OF\_MGMT\_IFACE**

This operation cannot be performed because creating or deleting a bond involving the management interface is not allowed while HA is on. In order to do that, disable HA, create or delete the bond then re-enable HA.

No parameters.

---

**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\_LIVASET**

HA could not be enabled on the Pool because a livenesset could not be formed: check storage and network heartbeat paths.

No parameters.

---

**HA\_HEARTBEAT\_DAEMON\_STARTUP\_FAILED**

The host could not join the livenesset because the HA daemon failed to start.

No parameters.

---

**HA\_HOST\_CANNOT\_ACCESS\_STATEFILE**

The host could not join the livenesset 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\_COMPATIBLE**

The hosts in this pool are not compatible.

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\_EVACUATE\_IN\_PROGRESS**

This host is being evacuated.

**Signature:**

HOST\_EVACUATE\_IN\_PROGRESS(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 toolstack is still initialising. Please wait.

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)`

---

**ILLEGAL\_VBD\_DEVICE**

The specified VBD device is not recognized: please use a non-negative integer

**Signature:**

ILLEGAL\_VBD\_DEVICE(vbd, device)

---

**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.

---

**INCOMPATIBLE\_PIF\_PROPERTIES**

These PIFs can not be bonded, because their properties are different.

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 its 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\_HOST\_POOL\_MISMATCH**

Host and pool have incompatible licenses (editions).

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)

---

**MESSAGE\_REMOVED**

This function is no longer available.

No parameters.

---

**MIRROR\_FAILED**

The VDI mirroring cannot be performed

**Signature:**

MIRROR\_FAILED(vdi)

---

**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.

---

**NOT\_SYSTEM\_DOMAIN**

The given VM is not registered as a system domain. This operation can only be performed on a registered system domain.

**Signature:**

`NOT_SYSTEM_DOMAIN(vm)`

---

**NO\_HOSTS\_AVAILABLE**

There were no hosts available to complete the specified operation.

No parameters.

---

**NO\_MORE\_REDO\_LOGS\_ALLOWED**

The upper limit of active redo log instances was reached.

No parameters.

---

**OBJECT\_NO\_LONGER\_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)
```

---

**OPERATION\_PARTIALLY\_FAILED**

Some VMs belonging to the appliance threw an exception while carrying out the specified operation

**Signature:**

```
OPERATION_PARTIALLY_FAILED(operation)
```

---

**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\_ISO\_MOUNTED**

XenServer Tools ISO must be ejected from all running VMs.

**Signature:**

PATCH\_PRECHECK\_FAILED\_ISO\_MOUNTED(patch)

---

**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\_BUILD**

The patch precheck stage failed: the server is of an incorrect build.

**Signature:**

```
PATCH_PRECHECK_FAILED_WRONG_SERVER_BUILD(patch, found_build, required_build)
```

---

**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)
```

---

**PGPU\_INSUFFICIENT\_CAPACITY\_FOR\_VGPU**

There is insufficient capacity on this PGPU to run the VGPU.

**Signature:**

```
PGPU_INSUFFICIENT_CAPACITY_FOR_VGPU(pgpu, vgpu_type)
```

---

**PGPU\_IN\_USE\_BY\_VM**

This PGPU is currently in use by running VMs.

**Signature:**

PGPU\_IN\_USE\_BY\_VM(VMs)

---

**PGPU\_NOT\_COMPATIBLE\_WITH\_GPU\_GROUP**

PGPU type not compatible with destination group.

**Signature:**

PGPU\_NOT\_COMPATIBLE\_WITH\_GPU\_GROUP(type, group\_types)

---

**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\_HAS\_NO\_V6\_NETWORK\_CONFIGURATION**

PIF has no IPv6 configuration (mode curently set to 'none')

No parameters.

---

**PIF\_INCOMPATIBLE\_PRIMARY\_ADDRESS\_TYPE**

The primary address types are not compatible

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\_UNMANAGED**

The operation you requested cannot be performed because the specified PIF is not managed by xapi.

**Signature:**

PIF\_UNMANAGED(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.

---

**SM\_PLUGIN\_COMMUNICATION\_FAILURE**

The SM plugin did not respond to a query.

**Signature:**

SM\_PLUGIN\_COMMUNICATION\_FAILURE(sm)

---

**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 or forgotten.

**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\_IS\_CACHE\_SR**

The SR is currently being used as a local cache SR.

**Signature:**

SR\_IS\_CACHE\_SR(host)

---

**SR\_NOT\_ATTACHED**

The SR is not attached.

**Signature:**

SR\_NOT\_ATTACHED(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.

---

**TOO\_MANY\_STORAGE\_MIGRATES**

You reached the maximal number of concurrently migrating VMs.

**Signature:**

`TOO_MANY_STORAGE_MIGRATES(number)`

---

**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\_CONTAINS\_METADATA\_OF\_THIS\_POOL**

The VDI could not be opened for metadata recovery as it contains the current pool's metadata.

**Signature:**

VDI\_CONTAINS\_METADATA\_OF\_THIS\_POOL(vdi, pool)

---

**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\_NEEDS\_VM\_FOR\_MIGRATE**

You attempted to migrate a VDI which is not attached to a running VM.

**Signature:**

VDI\_NEEDS\_VM\_FOR\_MIGRATE(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\_IN\_MAP**

This VDI was not mapped to a destination SR in VM.migrate\_send operation

**Signature:**

VDI\_NOT\_IN\_MAP(vdi)

---

**VDI\_NOT\_MANAGED**

This operation cannot be performed because the system does not manage this VDI

**Signature:**

VDI\_NOT\_MANAGED(vdi)

---

**VDI\_NOT\_SPARSE**

The VDI is not stored using a sparse format. It is not possible to query and manipulate only the changed blocks (or 'block differences' or 'disk deltas') between two VDIs. Please select a VDI which uses a sparse-aware technology such as VHD.

**Signature:**

VDI\_NOT\_SPARSE(vdi)

---

**VDI\_READONLY**

The operation required write access but this VDI is read-only

**Signature:**

VDI\_READONLY(vdi)

---

**VDI\_TOO\_SMALL**

The VDI is too small. Please resize it to at least the minimum size.

**Signature:**

VDI\_TOO\_SMALL(vdi, minimum size)

---

**VGPU\_TYPE\_NOT\_COMPATIBLE\_WITH\_RUNNING\_TYPE**

VGPU type is not compatible with one or more of the VGPU types currently running on this PGPU

**Signature:**

VGPU\_TYPE\_NOT\_COMPATIBLE\_WITH\_RUNNING\_TYPE(pgpu, type, running\_type)

---

**VGPU\_TYPE\_NOT\_ENABLED**

VGPU type is not one of the PGPU's enabled types.

**Signature:**

VGPU\_TYPE\_NOT\_ENABLED(type, enabled\_types)

---

**VGPU\_TYPE\_NOT\_SUPPORTED**

VGPU type is not one of the PGPU's supported types.

**Signature:**

VGPU\_TYPE\_NOT\_SUPPORTED(*type*, *supported\_types*)

---

**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 one 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\_ASSIGNED\_TO\_PROTECTION\_POLICY**

This VM is assigned to a protection policy.

**Signature:**

VM\_ASSIGNED\_TO\_PROTECTION\_POLICY(*vm*, *vmpp*)

---



**VM\_ATTACHED\_TO\_MORE\_THAN\_ONE\_VDI\_WITH\_TIMEOFFSET\_MARKED\_AS\_RESET\_ON\_BOOT**

You attempted to start a VM that's attached to more than one VDI with a timeoffset marked as reset-on-boot.

**Signature:**

VM\_ATTACHED\_TO\_MORE\_THAN\_ONE\_VDI\_WITH\_TIMEOFFSET\_MARKED\_AS\_RESET\_ON\_BOOT(vm)

---

**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 occurred while restoring the memory image of the specified virtual machine

**Signature:**

VM\_CHECKPOINT\_RESUME\_FAILED(vm)

---

**VM\_CHECKPOINT\_SUSPEND\_FAILED**

An error occurred 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\_HAS\_CHECKPOINT**

You attempted to migrate a VM which has a checkpoint.

**Signature:**

VM\_HAS\_CHECKPOINT(vm)

---

**VM\_HAS\_PCI\_ATTACHED**

This operation could not be performed, because the VM has one or more PCI devices passed through.

**Signature:**

VM\_HAS\_PCI\_ATTACHED(vm)

---

**VM\_HAS\_TOO\_MANY\_SNAPSHOTS**

You attempted to migrate a VM with more than one snapshot.

**Signature:**

VM\_HAS\_TOO\_MANY\_SNAPSHOTS(vm)

---

**VM\_HAS\_VGPU**

This operation could not be performed, because the VM has one or more virtual GPUs.

**Signature:**

VM\_HAS\_VGPU(vm)

---

**VM\_HOST\_INCOMPATIBLE\_VERSION**

This VM operation cannot be performed on an older-versioned host during an upgrade.

**Signature:**

VM\_HOST\_INCOMPATIBLE\_VERSION(host, vm)

---

**VM\_HVM\_REQUIRED**

HVM is required for this operation

**Signature:**

VM\_HVM\_REQUIRED(vm)

---

**VM\_INCOMPATIBLE\_WITH\_THIS\_HOST**

The VM is incompatible with the CPU features of this host.

**Signature:**

VM\_INCOMPATIBLE\_WITH\_THIS\_HOST(vm, host, reason)

---

**VM\_IS\_PART\_OF\_AN\_APPLIANCE**

This operation is not allowed as the VM is part of an appliance.

**Signature:**

VM\_IS\_PART\_OF\_AN\_APPLIANCE(vm, appliance)

---

**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\_LACKS\_FEATURE\_SHUTDOWN**

You attempted an operation which needs the cooperative shutdown feature on a VM which lacks it.

**Signature:**

VM\_LACKS\_FEATURE\_SHUTDOWN(vm)

---

**VM\_LACKS\_FEATURE\_SUSPEND**

You attempted an operation which needs the VM cooperative suspend feature on a VM which lacks it.

**Signature:**

VM\_LACKS\_FEATURE\_SUSPEND(vm)

---

**VM\_LACKS\_FEATURE\_VCPU\_HOTPLUG**

You attempted an operation which needs the VM hotplug-vcpu feature on a VM which lacks it.

**Signature:**

VM\_LACKS\_FEATURE\_VCPU\_HOTPLUG(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\_GPU**

You attempted to run a VM on a host which doesn't have a pGPU available in the GPU group needed by the VM. The VM has a vGPU attached to this GPU group.

**Signature:**

VM\_REQUIRES\_GPU(vm, GPU\_group)

---

**VM\_REQUIRES\_IOMMU**

You attempted to run a VM on a host which doesn't have I/O virtualization (IOMMU/VT-d) enabled, which is needed by the VM.

**Signature:**

VM\_REQUIRES\_IOMMU(host)

---

**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\_REQUIRES\_VGPU**

You attempted to run a VM on a host on which the vGPU required by the VM cannot be allocated on any pGPUs in the GPU\_group needed by the VM.

**Signature:**

```
VM_REQUIRES_VGPU(vm, GPU_group, vGPU_type)
```

---

**VM\_REVERT\_FAILED**

An error occurred 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\_TO\_IMPORT\_IS\_NOT\_NEWER\_VERSION**

The VM cannot be imported unforced because it is either the same version or an older version of an existing VM.

**Signature:**

VM\_TO\_IMPORT\_IS\_NOT\_NEWER\_VERSION(vm, existing\_version, version\_to\_import)

---

**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 requester 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)

---