Contents

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

2 API Reference 12
   2.1 Classes 12
   2.2 Relationships Between Classes 14
   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 24
      2.4.1 Fields for class: session 24
      2.4.2 RPCs associated with class: session 24
   2.5 Class: auth 34
      2.5.1 Fields for class: auth 34
      2.5.2 RPCs associated with class: auth 34
   2.6 Class: subject 36
      2.6.1 Fields for class: subject 36
      2.6.2 RPCs associated with class: subject 36
   2.7 Class: role 41
      2.7.1 Fields for class: role 41
      2.7.2 RPCs associated with class: role 41
   2.8 Class: task 46
      2.8.1 Fields for class: task 46
      2.8.2 RPCs associated with class: task 47
   2.9 Class: event 57
      2.9.1 Fields for class: event 57
2.9.2 RPCs associated with class: event .................................................. 57
2.10 Class: pool .................................................................................. 60
  2.10.1 Fields for class: pool ................................................................. 60
  2.10.2 RPCs associated with class: pool ........................................... 61
2.11 Class: pool_patch ......................................................................... 96
  2.11.1 Fields for class: pool_patch ...................................................... 96
  2.11.2 RPCs associated with class: pool_patch ............................... 96
2.12 Class: VM ................................................................................... 104
  2.12.1 Fields for class: VM ................................................................. 104
  2.12.2 RPCs associated with class: VM .......................................... 106
2.13 Class: VM_metrics ....................................................................... 175
  2.13.1 Fields for class: VM_metrics ................................................... 175
  2.13.2 RPCs associated with class: VM_metrics .............................. 175
2.14 Class: VM_guest_metrics ............................................................. 182
  2.14.1 Fields for class: VM_guest_metrics ....................................... 182
  2.14.2 RPCs associated with class: VM_guest_metrics .................. 182
2.15 Class: VMPP .................................................................................. 190
  2.15.1 Fields for class: VMPP ............................................................ 190
  2.15.2 RPCs associated with class: VMPP ....................................... 190
2.16 Class: VM_appliance ..................................................................... 209
  2.16.1 Fields for class: VM_appliance .............................................. 209
  2.16.2 RPCs associated with class: VM_appliance .......................... 209
2.17 Class: DR_task ........................................................................... 217
  2.17.1 Fields for class: DR_task ....................................................... 217
  2.17.2 RPCs associated with class: DR_task ................................... 217
2.18 Class: host .................................................................................. 220
  2.18.1 Fields for class: host ............................................................... 220
  2.18.2 RPCs associated with class: host ......................................... 221
2.19 Class: host_crashdump .................................................................. 267
  2.19.1 Fields for class: host_crashdump ......................................... 267
  2.19.2 RPCs associated with class: host_crashdump ...................... 267
2.20 Class: host_patch ........................................................................... 272
  2.20.1 Fields for class: host_patch ................................................... 272
  2.20.2 RPCs associated with class: host_patch ............................... 272
2.21 Class: host_metrics ...................................................................... 279
  2.21.1 Fields for class: host_metrics ............................................... 279
  2.21.2 RPCs associated with class: host_metrics ............................ 279
2.22 Class: host_cpu ............................................................................ 284
  2.22.1 Fields for class: host_cpu ..................................................... 284
  2.22.2 RPCs associated with class: host_cpu ................................. 284
2.23 Class: network ............................................................................ 291
  2.23.1 Fields for class: network ....................................................... 291
  2.23.2 RPCs associated with class: network .................................... 291
2.24 Class: VIF ................................................................................... 303
  2.24.1 Fields for class: VIF ............................................................... 303
  2.24.2 RPCs associated with class: VIF .......................................... 304
2.25 Class: VIF_metrics ................................................................. 322
  2.25.1 Fields for class: VIF_metrics ............................................... 322
  2.25.2 RPCs associated with class: VIF_metrics ............................. 322
2.26 Class: PIF ................................................................................... 327
  2.26.1 Fields for class: PIF ............................................................... 327
2.43.2 RPCs associated with class: message ........................................ 480
2.44 Class: secret .............................................................................. 484
  2.44.1 Fields for class: secret ................................................................. 484
  2.44.2 RPCs associated with class: secret ........................................... 484
2.45 Class: tunnel ............................................................................... 489
  2.45.1 Fields for class: tunnel ................................................................. 489
  2.45.2 RPCs associated with class: tunnel ........................................... 489
2.46 Class: PCI .................................................................................. 495
  2.46.1 Fields for class: PCI ................................................................. 495
  2.46.2 RPCs associated with class: PCI ................................................. 495
2.47 Class: PGPU ............................................................................. 501
  2.47.1 Fields for class: PGPU ................................................................. 501
  2.47.2 RPCs associated with class: PGPU ........................................... 501
2.48 Class: GPU_group ................................................................. 510
  2.48.1 Fields for class: GPU_group ..................................................... 510
  2.48.2 RPCs associated with class: GPU_group ................................... 510
2.49 Class: VGPU ............................................................................ 519
  2.49.1 Fields for class: VGPU ................................................................. 519
  2.49.2 RPCs associated with class: VGPU ........................................... 519
2.50 Class: VGPU_type ................................................................. 525
  2.50.1 Fields for class: VGPU_type ..................................................... 525
  2.50.2 RPCs associated with class: VGPU_type ................................... 525
2.51 Error Handling ................................................................. 532
  2.51.1 Error Codes ..................................................................... 533
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**
  - The field is Read Only. Furthermore, its value is automatically computed at runtime. For example: current CPU load and disk IO throughput.

- **ROins**
  - 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.

- **(t₁, t₂) Map**
  - signifies a mapping from values of type t₁ to values of type t₂.

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:

- \( \text{get}_f(\text{Ref } x) \): takes a \text{Ref} that refers to an object and returns the value of \( f \).

Each field, \( f \), with attribute \( RW \) and whose outermost type is \textit{Set} has the following additional RPCs associated with it:

- an \( \text{add}_f(\text{Ref } x, \ v) \) RPC adds a new element \( v \) to the set\(^1\);
- a \( \text{remove}_f(\text{Ref } x, \ v) \) RPC removes element \( v \) from the set;

Each field, \( f \), with attribute \( RW \) and whose outermost type is \textit{Map} has the following additional RPCs associated with it:

- an \( \text{add}_f(\text{Ref } x, \ k, \ v) \) 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 \( \text{remove}_f(\text{Ref } x, \ k) \) RPC removes the pair with key \( k \) from the mapping stored in \( f \) in object \( x \).

Each field whose outermost type is neither \textit{Set} nor \textit{Map}, but whose attribute is \( RW \) has an RPC accessor associated with it that sets its value:

- For \( RW \) (Read/Write), a \( \text{set}_f(\text{Ref } x, \ v) \) 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 \textit{constructor} RPC named \textit{create} that takes as parameters all fields marked \( RW \) and \( RO \_ins \). The result of this RPC is that a new \textit{persistent} object is created on the server-side with the specified field values.
- Each class has a \textit{get by uuid}(\text{uuid}) RPC that returns the object of that class that has the specified \text{uuid}.
- Each class that has a \textit{name label} field has a \textit{get by name label}(\text{name}) RPC that returns a set of objects of that class that have the specified \text{label}.
- Each class has a \textit{destroy}(\text{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 \textit{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 \textit{VM} class has RPCs for cloning, suspending, starting etc. Such additional RPCs are described explicitly in the API reference.

\(^{1}\)Since sets cannot contain duplicate values this operation has no action in the case that \( v \) was already in the set.
1.3 Wire Protocol for Remote API Calls

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

In our API Reference we specify the signatures of API functions in the following style:

\[(\text{ref}\_\text{vm Set}) \text{ VM.get\_all()}\]

This specifies that the function with name \text{VM.get\_all} takes no parameters and returns a Set of \text{ref\_vms}. These types are mapped onto XML-RPC types in a straight-forward manner:

- Floats, Bools, DateTimes and Strings map directly to the XML-RPC \text{double}, \text{boolean}, \text{dateTime.iso8601}, and \text{string} elements.
- all “ref” types are opaque references, encoded as the XML-RPC’s \text{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 “\text{String}” are mapped to the XML-RPC \text{String} type. The string itself is the OSF DCE UUID presentation format (as output by \text{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 \text{i4} type).
- values of enum types are encoded as strings. For example, a value of \text{destroy} of type \text{on\_normal\_exit}, would be conveyed as:

\[
<\text{value}><\text{string}>\text{destroy}</\text{string}></\text{value}>
\]

- for all our types, \(t\), our type \(t\ \text{Set}\) simply maps to XML-RPC’s \text{Array} type, so for example a value of type \text{String Set} would be transmitted like this:

\[
<\text{array}>
<\text{data}>
<\text{value}><\text{string}>\text{CX8}</\text{string}></\text{value}>
<\text{value}><\text{string}>\text{PSE36}</\text{string}></\text{value}>
<\text{value}><\text{string}>\text{FPU}</\text{string}></\text{value}>
</\text{data}>
</\text{array}>
\]

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

\[
<\text{value}>
<\text{struct}>
<\text{member}>
<\text{name}>Mike</\text{name}>
<\text{value}><\text{double}>2.3</\text{double}></\text{value}>
</\text{member}>
</\text{value}>
\]
1.3 WIRE PROTOCOL FOR REMOTE API CALLS

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:

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

```c
session_id session.login_with_password(string uname, string pwd, string version, string originator)
```

Where `uname` and `password` refer to your username and password while `version` and `originator` are optional and refer to the api version and client name respectively, as defined by the Xen administrator. (The `version` is ignored at present though. A client that doesn’t want to specify it can send `uname, pwd, "", originator`) 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:

```c
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...
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`:

```
>>> 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", "version", "originator")['Value']
```

When serialised, this call looks like the following:

```xml
<?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><param><value><string>version</string></value></param><param><value><string>originator</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']
```

```
['OpaqueRef:1', 'OpaqueRef:2', 'OpaqueRef:3', 'OpaqueRef:4']
```
1.5. EXAMPLE INTERACTIVE SESSION

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:

```python
>>> 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:

```python
>>> 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:

```python
>>> 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:

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

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

<table>
<thead>
<tr>
<th>object.field</th>
<th>object.field</th>
<th>relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM.snapshot_of</td>
<td>VM.snapshots</td>
<td>one-to-many</td>
</tr>
<tr>
<td>VDI.snapshot_of</td>
<td>VDI.snapshots</td>
<td>one-to-many</td>
</tr>
<tr>
<td>VM.parent</td>
<td>VM.children</td>
<td>one-to-many</td>
</tr>
<tr>
<td>task.subtask_of</td>
<td>task.subtasks</td>
<td>one-to-many</td>
</tr>
<tr>
<td>PIF.bond_slave_of</td>
<td>Bond.slaves</td>
<td>one-to-many</td>
</tr>
<tr>
<td>Bond.master</td>
<td>PIF.bond_master_of</td>
<td>one-to-many</td>
</tr>
<tr>
<td>VLAN.tagged_PIF</td>
<td>PIF.VLAN_slave_of</td>
<td>one-to-many</td>
</tr>
<tr>
<td>tunnel.access_PIF</td>
<td>PIF.tunnel_access_PIF_of</td>
<td>one-to-many</td>
</tr>
<tr>
<td>tunnel.transport_PIF</td>
<td>PIF.tunnel_transport_PIF_of</td>
<td>one-to-many</td>
</tr>
<tr>
<td>PBD.host</td>
<td>host.PBDs</td>
<td>one-to-many</td>
</tr>
<tr>
<td>PBD.SR</td>
<td>SR.PBDs</td>
<td>one-to-many</td>
</tr>
<tr>
<td>VBD.VDI</td>
<td>VDI.VBDs</td>
<td>one-to-many</td>
</tr>
<tr>
<td>crashdump.VDI</td>
<td>VDI.crash_dumps</td>
<td>one-to-many</td>
</tr>
<tr>
<td>VBD.VM</td>
<td>VM.VBDs</td>
<td>one-to-many</td>
</tr>
<tr>
<td>crashdump.VM</td>
<td>VM.crash_dumps</td>
<td>one-to-many</td>
</tr>
<tr>
<td>VIF.VM</td>
<td>VM.VIFs</td>
<td>one-to-many</td>
</tr>
<tr>
<td>VIF.network</td>
<td>network.VIFs</td>
<td>one-to-many</td>
</tr>
<tr>
<td>PIF.host</td>
<td>host.PIFs</td>
<td>one-to-many</td>
</tr>
<tr>
<td>PIF.network</td>
<td>network.PIFs</td>
<td>one-to-many</td>
</tr>
<tr>
<td>VDLSR</td>
<td>SR.VDIs</td>
<td>one-to-many</td>
</tr>
<tr>
<td>VTPM.VM</td>
<td>VM.VTPMs</td>
<td>one-to-many</td>
</tr>
<tr>
<td>console.VM</td>
<td>VM.consoles</td>
<td>one-to-many</td>
</tr>
<tr>
<td>VM.resident_on</td>
<td>host.resident_VMs</td>
<td>one-to-many</td>
</tr>
<tr>
<td>host.CPU.host</td>
<td>host.host_CPUs</td>
<td>one-to-many</td>
</tr>
<tr>
<td>host.crashdump.host</td>
<td>host.crashdumps</td>
<td>one-to-many</td>
</tr>
<tr>
<td>host.patch.host</td>
<td>host.patches</td>
<td>one-to-many</td>
</tr>
<tr>
<td>host.patch.pool_pool</td>
<td>pool_patch.pool_patches</td>
<td>one-to-many</td>
</tr>
<tr>
<td>subject.roles</td>
<td>subject.roles</td>
<td>unknown type</td>
</tr>
<tr>
<td>role.subroles</td>
<td>role.subroles</td>
<td>many-to-many</td>
</tr>
<tr>
<td>VM.protection_policy</td>
<td>VMPP.VMs</td>
<td>one-to-many</td>
</tr>
<tr>
<td>VM.appliance</td>
<td>VM_appliance.VMs</td>
<td>one-to-many</td>
</tr>
<tr>
<td>PGPU.GPU_group</td>
<td>GPU_group.PGPUs</td>
<td>one-to-many</td>
</tr>
<tr>
<td>VGPU.GPU_group</td>
<td>GPU_group.VGPUs</td>
<td>one-to-many</td>
</tr>
<tr>
<td>VGPU.type</td>
<td>VGPU.type.VGPUs</td>
<td>one-to-many</td>
</tr>
<tr>
<td>VGPU.VM</td>
<td>VGPU.VM</td>
<td>one-to-many</td>
</tr>
<tr>
<td>VGPU.resident_on</td>
<td>PGPU.resident_VGPUs</td>
<td>one-to-many</td>
</tr>
<tr>
<td>PGPU.supported_VGPU_types</td>
<td>VGPU.type.supported_on_PGPUs</td>
<td>many-to-many</td>
</tr>
<tr>
<td>PGPU.enabled_VGPU_types</td>
<td>VGPU.type.enabled_on_PGPUs</td>
<td>many-to-many</td>
</tr>
<tr>
<td>GPU_group.supported_VGPU_types</td>
<td>VGPU.type.supported_on_GPU_groups</td>
<td>many-to-many</td>
</tr>
<tr>
<td>GPU_group.enabled_VGPU_types</td>
<td>VGPU.type.enabled_on_GPU_groups</td>
<td>many-to-many</td>
</tr>
<tr>
<td>PCI.host</td>
<td>host.PCIs</td>
<td>one-to-many</td>
</tr>
<tr>
<td>PGPU.host</td>
<td>host.PGPUs</td>
<td>one-to-many</td>
</tr>
<tr>
<td>VDI.metadata_of_pool</td>
<td>pool.metadata_VDIs</td>
<td>one-to-many</td>
</tr>
<tr>
<td>SR.introduced_by</td>
<td>DR_task.introduced_SR</td>
<td>one-to-many</td>
</tr>
</tbody>
</table>

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

2.3.1 Primitives

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

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>text strings</td>
</tr>
<tr>
<td>int</td>
<td>64-bit integers</td>
</tr>
<tr>
<td>float</td>
<td>IEEE double-precision floating-point numbers</td>
</tr>
<tr>
<td>bool</td>
<td>boolean</td>
</tr>
<tr>
<td>datetime</td>
<td>date and timestamp</td>
</tr>
</tbody>
</table>

2.3.2 Higher-order types

The following type constructors are used:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>c ref</td>
<td>reference to an object of class c</td>
</tr>
<tr>
<td>t set</td>
<td>a set of elements of type t</td>
</tr>
<tr>
<td>(a → b) map</td>
<td>a table mapping values of type a to values of type b</td>
</tr>
</tbody>
</table>

2.3.3 Enumeration types

The following enumeration types are used:

<table>
<thead>
<tr>
<th>enum event_operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>An object has been created</td>
</tr>
<tr>
<td>del</td>
<td>An object has been deleted</td>
</tr>
<tr>
<td>mod</td>
<td>An object has been modified</td>
</tr>
</tbody>
</table>
### 2.3. TYPES

<table>
<thead>
<tr>
<th>enum vgpu_type_implementation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>passthrough</td>
<td>Pass through an entire physical GPU to a guest</td>
</tr>
<tr>
<td>nvidia</td>
<td>vGPU using NVIDIA hardware</td>
</tr>
<tr>
<td>gvt-g</td>
<td>vGPU using Intel GVT-g</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>enum pgpu_dom0_access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enabled</td>
<td>dom0 can access this device as normal</td>
</tr>
<tr>
<td>disable_on_reboot</td>
<td>On host reboot dom0 will be blocked from accessing this device</td>
</tr>
<tr>
<td>disabled</td>
<td>dom0 cannot access this device</td>
</tr>
<tr>
<td>enable_on_reboot</td>
<td>On host reboot dom0 will be allowed to access this device</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>enum console_protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vt100</td>
<td>VT100 terminal</td>
</tr>
<tr>
<td>rfb</td>
<td>Remote FrameBuffer protocol (as used in VNC)</td>
</tr>
<tr>
<td>rdp</td>
<td>Remote Desktop Protocol</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>enum vbd_operations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>attach</td>
<td>Attempting to attach this VBD to a VM</td>
</tr>
<tr>
<td>eject</td>
<td>Attempting to eject the media from this VBD</td>
</tr>
<tr>
<td>insert</td>
<td>Attempting to insert new media into this VBD</td>
</tr>
<tr>
<td>plug</td>
<td>Attempting to hotplug this VBD</td>
</tr>
<tr>
<td>unplug</td>
<td>Attempting to hot unplug this VBD</td>
</tr>
<tr>
<td>unplug_force</td>
<td>Attempting to forcibly unplug this VBD</td>
</tr>
<tr>
<td>pause</td>
<td>Attempting to pause a block device backend</td>
</tr>
<tr>
<td>unpause</td>
<td>Attempting to unpause a block device backend</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>enum vdi_operations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>scan</td>
<td>Scanning backends for new or deleted VDIs</td>
</tr>
<tr>
<td>clone</td>
<td>Cloning the VDI</td>
</tr>
<tr>
<td>copy</td>
<td>Copying the VDI</td>
</tr>
<tr>
<td>resize</td>
<td>Resizing the VDI</td>
</tr>
<tr>
<td>resize_online</td>
<td>Resizing the VDI which may or may not be online</td>
</tr>
<tr>
<td>snapshot</td>
<td>Snapshotting the VDI</td>
</tr>
<tr>
<td>destroy</td>
<td>Destroying the VDI</td>
</tr>
<tr>
<td>forget</td>
<td>Forget about the VDI</td>
</tr>
<tr>
<td>update</td>
<td>Refreshing the fields of the VDI</td>
</tr>
</tbody>
</table>
2.3. TYPES

| force_unlock | Forcibly unlocking the VDI |
| generate_config | Generating static configuration |
| blocked | Operations on this VDI are temporarily blocked |

<table>
<thead>
<tr>
<th>enum storage_operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>scan</td>
</tr>
<tr>
<td>destroy</td>
</tr>
<tr>
<td>forget</td>
</tr>
<tr>
<td>plug</td>
</tr>
<tr>
<td>unplug</td>
</tr>
<tr>
<td>update</td>
</tr>
<tr>
<td>vdi_create</td>
</tr>
<tr>
<td>vdi_introduce</td>
</tr>
<tr>
<td>vdi_destroy</td>
</tr>
<tr>
<td>vdi_resize</td>
</tr>
<tr>
<td>vdi_clone</td>
</tr>
<tr>
<td>vdi_snapshot</td>
</tr>
<tr>
<td>pbd_create</td>
</tr>
<tr>
<td>pbdDestroy</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>enum vif_operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>attach</td>
</tr>
<tr>
<td>plug</td>
</tr>
<tr>
<td>unplug</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>enum network_operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>attaching</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>enum host_allowed_operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>provision</td>
</tr>
<tr>
<td>evacuate</td>
</tr>
<tr>
<td>shutdown</td>
</tr>
<tr>
<td>reboot</td>
</tr>
<tr>
<td>power_on</td>
</tr>
<tr>
<td>vm_start</td>
</tr>
<tr>
<td>vm_resume</td>
</tr>
<tr>
<td>vm_migrate</td>
</tr>
</tbody>
</table>
### 2.3. TYPES

#### Chapter 2. API Reference

```markdown
**enum vm_appliance_operation**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>Start</td>
</tr>
<tr>
<td>clean_shutdown</td>
<td>Clean shutdown</td>
</tr>
<tr>
<td>hard_shutdown</td>
<td>Hard shutdown</td>
</tr>
<tr>
<td>shutdown</td>
<td>Shutdown</td>
</tr>
</tbody>
</table>
```

```markdown
**enum tristate_type**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>Known to be true</td>
</tr>
<tr>
<td>no</td>
<td>Known to be false</td>
</tr>
<tr>
<td>unspecified</td>
<td>Unknown or unspecified</td>
</tr>
</tbody>
</table>
```

```markdown
**enum vm_power_state**

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Halted</td>
<td>VM is offline and not using any resources</td>
</tr>
<tr>
<td>Paused</td>
<td>All resources have been allocated but the VM itself is paused and its vCPUs are not running</td>
</tr>
<tr>
<td>Running</td>
<td>Running</td>
</tr>
<tr>
<td>Suspended</td>
<td>VM state has been saved to disk and it is no longer running. Note that disks remain in-use while the VM is suspended.</td>
</tr>
</tbody>
</table>
```

```markdown
**enum after_apply_guidance**

- **restartHVM**: This patch requires HVM guests to be restarted once applied.
- **restartPV**: This patch requires PV guests to be restarted once applied.
- **restartHost**: This patch requires the host to be restarted once applied.
- **restartXAPI**: This patch requires XAPI to be restarted once applied.
```

```markdown
**enum pool_allowed_operations**

- **ha_enable**: Indicates this pool is in the process of enabling HA
- **ha_disable**: Indicates this pool is in the process of disabling HA
```

```markdown
**enum task_status_type**

- **pending**: task is in progress
- **success**: task was completed successfully
- **failure**: task has failed
- **cancelling**: task is being cancelled
- **cancelled**: task has been cancelled
```
### 2.3. TYPES

#### enum task:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cancel</td>
<td>refers to the operation “cancel”</td>
</tr>
<tr>
<td>destroy</td>
<td>refers to the operation “destroy”</td>
</tr>
</tbody>
</table>

#### enum on_normal_exit:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>destroy</td>
<td>destroy the VM state</td>
</tr>
<tr>
<td>restart</td>
<td>restart the VM</td>
</tr>
</tbody>
</table>

#### enum on_crash_behaviour:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>destroy</td>
<td>destroy the VM state</td>
</tr>
<tr>
<td>coredump_and_destroy</td>
<td>record a coredump and then destroy the VM state</td>
</tr>
<tr>
<td>restart</td>
<td>restart the VM</td>
</tr>
<tr>
<td>coredump_and_restart</td>
<td>record a coredump and then restart the VM</td>
</tr>
<tr>
<td>preserve</td>
<td>leave the crashed VM paused</td>
</tr>
<tr>
<td>rename_restart</td>
<td>rename the crashed VM and start a new copy</td>
</tr>
</tbody>
</table>

#### enum vm_operations:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>snapshot</td>
<td>refers to the operation “snapshot”</td>
</tr>
<tr>
<td>clone</td>
<td>refers to the operation “clone”</td>
</tr>
<tr>
<td>copy</td>
<td>refers to the operation “copy”</td>
</tr>
<tr>
<td>create_template</td>
<td>refers to the operation “create_template”</td>
</tr>
<tr>
<td>revert</td>
<td>refers to the operation “revert”</td>
</tr>
<tr>
<td>checkpoint</td>
<td>refers to the operation “checkpoint”</td>
</tr>
<tr>
<td>snapshot_with_quiesce</td>
<td>refers to the operation “snapshot_with_quiesce”</td>
</tr>
<tr>
<td>provision</td>
<td>refers to the operation “provision”</td>
</tr>
<tr>
<td>start</td>
<td>refers to the operation “start”</td>
</tr>
<tr>
<td>start_on</td>
<td>refers to the operation “start_on”</td>
</tr>
<tr>
<td>pause</td>
<td>refers to the operation “pause”</td>
</tr>
<tr>
<td>unpause</td>
<td>refers to the operation “unpause”</td>
</tr>
<tr>
<td>clean_shutdown</td>
<td>refers to the operation “clean_shutdown”</td>
</tr>
<tr>
<td>clean_reboot</td>
<td>refers to the operation “clean_reboot”</td>
</tr>
<tr>
<td>hard_shutdown</td>
<td>refers to the operation “hard_shutdown”</td>
</tr>
<tr>
<td>power_state_reset</td>
<td>refers to the operation “power_state_reset”</td>
</tr>
<tr>
<td>hard_reboot</td>
<td>refers to the operation “hard_reboot”</td>
</tr>
<tr>
<td>suspend</td>
<td>refers to the operation “suspend”</td>
</tr>
<tr>
<td>cvm</td>
<td>refers to the operation “cvm”</td>
</tr>
<tr>
<td>resume</td>
<td>refers to the operation “resume”</td>
</tr>
<tr>
<td>resume_on</td>
<td>refers to the operation “resume_on”</td>
</tr>
<tr>
<td>pool_migrate</td>
<td>refers to the operation “pool_migrate”</td>
</tr>
<tr>
<td>migrate_send</td>
<td>refers to the operation “migrate_send”</td>
</tr>
</tbody>
</table>
get_boot_record refers to the operation “get_boot_record”
send_sysrq refers to the operation “send_sysrq”
send_trigger refers to the operation “send_trigger”
query_services refers to the operation “query_services”
shutdown refers to the operation “shutdown”
call_plugin refers to the operation “call_plugin”
changing_memory_live refers to the operation “changing_memory_live”
awaiting_memory_live refers to the operation “awaiting_memory_live”
changing_dynamic_range refers to the operation “changing_dynamic_range”
changing_static_range refers to the operation “changing_static_range”
changing_memory_limits refers to the operation “changing_memory_limits”
changing_shadow_memory refers to the operation “changing_shadow_memory”
changing_shadow_memory_live refers to the operation “changing_shadow_memory_live”
changing_VCPUs refers to the operation “changing_VCPUs”
changing_VCPUs_live refers to the operation “changing_VCPUs_live”
assert_operation_valid refers to the operation “assert_operation_valid”
data_source_op refers to the operation “data_source_op”
update_allowed_operations refers to the operation “update_allowed_operations”
make_into_template refers to the operation “make_into_template”
import refers to the operation “import”
export refers to the operation “export”
metadata_export refers to the operation “metadata_export”
reverting refers to the operation “reverting”
destroy refers to the operation “destroy”

Add, remove, query or list data sources

Turning this VM into a template
importing a VM from a network stream
exporting a VM to a network stream
exporting VM metadata to a network stream
Reverting the VM to a previous snapshotted state
refers to the act of uninstalling the VM

<table>
<thead>
<tr>
<th>enum vmpp_backup_frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>hourly</td>
</tr>
<tr>
<td>daily</td>
</tr>
<tr>
<td>weekly</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>enum vmpp_archive_frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>never</td>
</tr>
<tr>
<td>always_after_backup</td>
</tr>
<tr>
<td>daily</td>
</tr>
<tr>
<td>weekly</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>enum vmpp_archive_target_type</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
</tr>
<tr>
<td>cifs</td>
</tr>
<tr>
<td>nfs</td>
</tr>
</tbody>
</table>
### enum vmpp_backup_type

<table>
<thead>
<tr>
<th>backup type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>snapshot</td>
<td>The backup is a snapshot</td>
</tr>
<tr>
<td>checkpoint</td>
<td>The backup is a checkpoint</td>
</tr>
</tbody>
</table>

### enum host_display

<table>
<thead>
<tr>
<th>display mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enabled</td>
<td>This host is outputting its console to a physical display device</td>
</tr>
<tr>
<td>disable_on_reboot</td>
<td>The host will stop outputting its console to a physical display device on next boot</td>
</tr>
<tr>
<td>disabled</td>
<td>This host is not outputting its console to a physical display device</td>
</tr>
<tr>
<td>enable_on_reboot</td>
<td>The host will start outputting its console to a physical display device on next boot</td>
</tr>
</tbody>
</table>

### enum network_default_locking_mode

<table>
<thead>
<tr>
<th>locking mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>unlocked</td>
<td>Treat all VIFs on this network with locking_mode = 'default' as if they have locking</td>
</tr>
<tr>
<td>disabled</td>
<td>Treat all VIFs on this network with locking_mode = 'default' as if they have locking</td>
</tr>
</tbody>
</table>

### enum vif_locking_mode

<table>
<thead>
<tr>
<th>locking mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network_default</td>
<td>No specific configuration set - default network policy applies</td>
</tr>
<tr>
<td>locked</td>
<td>Only traffic to a specific MAC and a list of IPv4 or IPv6 addresses is permitted</td>
</tr>
<tr>
<td>unlocked</td>
<td>All traffic is permitted</td>
</tr>
<tr>
<td>disabled</td>
<td>No traffic is permitted</td>
</tr>
</tbody>
</table>

### enum vif_ipv4_configuration_mode

<table>
<thead>
<tr>
<th>configuration mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Follow the default IPv4 configuration of the guest (this is guest-dependent)</td>
</tr>
<tr>
<td>Static</td>
<td>Static IPv4 address configuration</td>
</tr>
</tbody>
</table>

### enum vif_ipv6_configuration_mode

<table>
<thead>
<tr>
<th>configuration mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Follow the default IPv6 configuration of the guest (this is guest-dependent)</td>
</tr>
<tr>
<td>Static</td>
<td>Static IPv6 address configuration</td>
</tr>
</tbody>
</table>

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

| None       | Do not acquire an IPv6 address |
| DHCP       | Acquire an IPv6 address by DHCP |
| Static     | Static IPv6 address configuration |
| Autoconf   | Router assigned prefix delegation IPv6 allocation |

| IPv4       | Primary address is the IPv4 address |
| IPv6       | Primary address is the IPv6 address |

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

| 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 |
| rrd           | a disk that stores SR-level RRDs |

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

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO</td>
<td>only read-only access will be allowed</td>
</tr>
<tr>
<td>RW</td>
<td>read-write access will be allowed</td>
</tr>
</tbody>
</table>

### enum vbd_type

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD</td>
<td>VBD will appear to guest as CD</td>
</tr>
<tr>
<td>Disk</td>
<td>VBD will appear to guest as disk</td>
</tr>
<tr>
<td>Floppy</td>
<td>VBD will appear as a floppy</td>
</tr>
</tbody>
</table>

###_enum cls

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM</td>
<td>VM</td>
</tr>
<tr>
<td>Host</td>
<td>Host</td>
</tr>
<tr>
<td>SR</td>
<td>SR</td>
</tr>
<tr>
<td>Pool</td>
<td>Pool</td>
</tr>
<tr>
<td>VMPP</td>
<td>VMPP</td>
</tr>
</tbody>
</table>

### enum allocation_algorithm

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>breadth_first</td>
<td>vGPUs of a given type are allocated evenly across supporting pGPUs.</td>
</tr>
<tr>
<td>depth_first</td>
<td>vGPUs of a given type are allocated on supporting pGPUs until they are full.</td>
</tr>
</tbody>
</table>
2.4 Class: session

2.4.1 Fields for class: session

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

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:
### 2.4. CLASS: SESSION

#### Overview:

The API reference for the `session` class.

#### Fields:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uname</td>
<td>Username for login.</td>
</tr>
<tr>
<td>string</td>
<td>pwd</td>
<td>Password for login.</td>
</tr>
<tr>
<td>string</td>
<td>version</td>
<td>Client API version.</td>
</tr>
<tr>
<td>string</td>
<td>originator</td>
<td>Key string for distinguishing different API users sharing the same login name.</td>
</tr>
</tbody>
</table>

**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:**

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

**Return Type:** `void`

#### RPC name: change_password

**Overview:**
Change the account password; if your session is authenticated with root priviledges then the old pwd is validated and the new pwd is set regardless.

**Signature:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>old_pwd</td>
<td>Old password for account</td>
</tr>
<tr>
<td>string</td>
<td>new_pwd</td>
<td>New password for account</td>
</tr>
</tbody>
</table>

**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:**

```c
(session ref) slave_local_login_with_password (string uname, string pwd)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uname</td>
<td>Username for login.</td>
</tr>
<tr>
<td>string</td>
<td>pwd</td>
<td>Password for login.</td>
</tr>
</tbody>
</table>

Return Type: session ref
ID of newly created session

RPC name: create_from_db_file

Overview:

Signature:

\[(\text{session ref}) \ \text{create\_from\_db\_file} (\text{session\_id} \ s, \ \text{string} \ \text{filename})\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>filename</td>
<td>Database dump filename.</td>
</tr>
</tbody>
</table>

Return Type: session ref
ID of newly created session

RPC name: local_logout

Overview:
Log out of local session.

Signature:

\[\text{void} \ \text{local\_logout} (\text{session\_id} \ s)\]

Return Type: void

RPC name: get_all_subject_identifiers

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

Signature:

\[(\text{string set}) \ \text{get\_all\_subject\_identifiers} (\text{session\_id} \ s)\]

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:

```c
void logout\_subject\_identifier (session\_id s, string subject\_identifier)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>subject_identifier</td>
<td>User subject-identifier of the sessions to be destroyed</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get\_uuid

Overview:
Get the uuid field of the given session.

Signature:

```c
string get\_uuid (session\_id s, session ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string

value of the field

RPC name: get\_this\_host

Overview:
Get the this\_host field of the given session.

Signature:

```c
(host ref) get\_this\_host (session\_id s, session ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: host ref

value of the field
RPC name: get\_this\_user

Overview:
Get the this\_user field of the given session.

Signature:

\[(\text{user ref}) \text{get\_this\_user (session\_id s, session ref self)}\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: user ref
value of the field

RPC name: get\_last\_active

Overview:
Get the last\_active field of the given session.

Signature:

\[(\text{datetime get\_last\_active (session\_id s, session ref self)}\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: datetime
value of the field

RPC name: get\_pool

Overview:
Get the pool field of the given session.

Signature:

\[(\text{bool get\_pool (session\_id s, session ref self)}\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: bool
value of the field
RPC name: get\_other\_config

Overview:
Get the other\_config field of the given session.

Signature:

```
((string \rightarrow string) map) get\_other\_config (session\_id s, session ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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 \rightarrow string) map value)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string \rightarrow string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: subject ref
value of the field
RPC name: get_validation_time

Overview:
Get the validation_time field of the given session.

Signature:

\[
\text{datetime get_validation_time (session_id s, session ref self)}
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

\[
\text{string get_auth_user_sid (session_id s, session ref self)}
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

\[
\text{string get_auth_user_name (session_id s, session ref self)}
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: session ref
value of the field
RPC name: get_originator

Overview:
Get the originator field of the given session.

Signature:

\[
\text{string get_originator (session_id s, session ref self)}
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

\[
\text{(session ref) get_by_uuid (session_id s, string uuid)}
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

\[
\text{(session record) get_record (session_id s, session ref self)}
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>subject_name</td>
<td>The human-readable subject name, such as a username or a groupname</td>
</tr>
</tbody>
</table>

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_identifier)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>subject_identifier</td>
<td>A string containing the subject_identifier, unique in the external directory service</td>
</tr>
</tbody>
</table>

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

RPC name: get_group_membership

Overview:
This call 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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>subject_identifier</td>
<td>A string containing the subject_identifier, unique in the external directory service</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject</td>
<td>subject</td>
<td></td>
<td>A user or group that can log in xapi.</td>
</tr>
<tr>
<td></td>
<td>run uuid</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td></td>
<td>ins subject</td>
<td>string</td>
<td>the subject identifier, unique in the external directory service</td>
</tr>
<tr>
<td></td>
<td>ins other_config</td>
<td>(string → string) map</td>
<td>additional configuration</td>
</tr>
<tr>
<td></td>
<td>run roles</td>
<td>role ref set</td>
<td>the roles associated with this subject</td>
</tr>
</tbody>
</table>

2.6.2 RPCs associated with class: subject

RPC name: add_to_roles

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

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject ref</td>
<td>self</td>
<td>The subject who we want to add the role to</td>
</tr>
<tr>
<td>role ref</td>
<td>role</td>
<td>The unique role reference</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: remove_from_roles

Overview:
This call removes a role from a subject.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject ref</td>
<td>self</td>
<td>The subject from whom we want to remove the role</td>
</tr>
<tr>
<td>role ref</td>
<td>role</td>
<td>The unique role reference in the subject’s roles field</td>
</tr>
</tbody>
</table>

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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject ref</td>
<td>self</td>
<td>The subject whose permissions will be retrieved</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject record</td>
<td>args</td>
<td>All constructor arguments</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: subject record
all fields from the object
2.7 Class: role

2.7.1 Fields for class: role

<table>
<thead>
<tr>
<th>Name</th>
<th>role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>A set of permissions associated with a subject.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO_run</td>
<td>uuid</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>RO_ins</td>
<td>name_label</td>
<td>string</td>
<td>a short user-friendly name for the role</td>
</tr>
<tr>
<td>RO_ins</td>
<td>name_description</td>
<td>string</td>
<td>what this role is for</td>
</tr>
<tr>
<td>RO_ins</td>
<td>subroles</td>
<td>role ref set</td>
<td>a list of pointers to other roles or permissions</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>role ref</td>
<td>self</td>
<td>a reference to a role</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>role ref</td>
<td>self</td>
<td>a reference to a role</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>role ref</td>
<td>permission</td>
<td>a reference to a permission</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>label</td>
<td>The short friendly name of the role</td>
</tr>
</tbody>
</table>

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 \rightarrow 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:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>role ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

**RPC name: get_name_label**

**Overview:**
Get the name/label field of the given role.

**Signature:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>role ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

**RPC name: get_name_description**

**Overview:**
Get the name/description field of the given role.

**Signature:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>role ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>role ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>role ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:**

\[(\text{role ref set}) \text{get\_by\_name\_label} (\text{session\_id s, string label})\]

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>label</td>
<td>label of object to return</td>
</tr>
</tbody>
</table>

**Return Type:** \text{role ref set}

references to objects with matching names
## 2.8 Class: task

### 2.8.1 Fields for class: task

<table>
<thead>
<tr>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO</td>
<td>uuid</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>RO</td>
<td>name_label</td>
<td>string</td>
<td>a human-readable name</td>
</tr>
<tr>
<td>RO</td>
<td>name_description</td>
<td>string</td>
<td>a notes field containing human-readable description</td>
</tr>
<tr>
<td>RO</td>
<td>allowed_operations</td>
<td>task_allowed_operations set</td>
<td>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.</td>
</tr>
<tr>
<td>RO</td>
<td>current_operations</td>
<td>(string → task_allowed_operations) map</td>
<td>links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.</td>
</tr>
<tr>
<td>RO</td>
<td>created</td>
<td>datetime</td>
<td>Time task was created</td>
</tr>
<tr>
<td>RO</td>
<td>finished</td>
<td>datetime</td>
<td>Time task finished (i.e. succeeded or failed). If task-status is pending, then the value of this field has no meaning current status of the task</td>
</tr>
<tr>
<td>RO</td>
<td>status</td>
<td>task_status_type</td>
<td>the host on which the task is running</td>
</tr>
<tr>
<td>RO</td>
<td>resident_on</td>
<td>host ref</td>
<td>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.</td>
</tr>
<tr>
<td>RO</td>
<td>progress</td>
<td>float</td>
<td></td>
</tr>
<tr>
<td>RO</td>
<td>type</td>
<td>string</td>
<td>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.</td>
</tr>
<tr>
<td>RO</td>
<td>result</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>RO</td>
<td>error_info</td>
<td>string set</td>
<td>if the task has failed, this field contains the set of associated error strings. Undefined otherwise.</td>
</tr>
<tr>
<td>RW</td>
<td>other_config</td>
<td>(string → string) map</td>
<td>additional configuration</td>
</tr>
<tr>
<td>RO</td>
<td>subtask_of</td>
<td>task ref</td>
<td>Ref pointing to the task this is a subtask of.</td>
</tr>
<tr>
<td>RO</td>
<td>subtasks</td>
<td>task ref set</td>
<td>List pointing to all the subtasks.</td>
</tr>
<tr>
<td>RO</td>
<td>backtrace</td>
<td>string</td>
<td>Function call trace for debugging.</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>label</td>
<td>short label for the new task</td>
</tr>
<tr>
<td>string</td>
<td>description</td>
<td>longer description for the new task</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task ref</td>
<td>self</td>
<td>Reference to the task object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task ref</td>
<td>task</td>
<td>The task</td>
</tr>
</tbody>
</table>

Return Type: void

Possible Error Codes: OPERATION_NOT_ALLOWED
RPC name: get_all

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

(task ref set) get_all (session_id s)

Return Type: task ref set
references to all objects

RPC name: get_all_records

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

((task ref -> task record) map) get_all_records (session_id s)

Return Type: (task ref -> task record) map
records of all objects

RPC name: get_uuid

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

string get_uuid (session_id s, task ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get_name_description

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

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get_allowed_operations

Overview:
Get the allowed operations field of the given task.

Signature:

```plaintext
(task_allowed_operations set) get_allowed_operations (session_id s, task ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```plaintext
((string -> task_allowed_operations) map) get_current_operations (session_id s, task ref self)
```
2.8. CLASS: TASK

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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)
**RPC name:** get_resident_on

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

**Signature:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**Return Type:** host ref

value of the field

**RPC name:** get_progress

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

**Signature:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**Return Type:** float

value of the field

**RPC name:** get_type

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

**Signature:**

```plaintext
string get_type (session_id s, task ref self)
```
2.8. CLASS: TASK

CHAPTER 2. API REFERENCE

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get_result

Overview:
Get the result field of the given task.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get_error_info

Overview:
Get the error_info field of the given task.

Signature:

```plaintext
(string set) get_error_info (session_id s, task ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string set
value of the field

RPC name: get_other_config

Overview:
Get the other_config field of the given task.

Signature:

```plaintext
((string -> string) map) get_other_config (session_id s, task ref self)
```
2.8. CLASS: TASK

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

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

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: task ref set
value of the field

RPC name: get_backtrace
Overview:
Get the backtrace field of the given task.
Signature:

string get_backtrace (session_id s, task ref self)
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>task ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>label</td>
<td>label of object to return</td>
</tr>
</tbody>
</table>

Return Type: task ref set

references to objects with matching names
2.9 Class: event

2.9.1 Fields for class: event

<table>
<thead>
<tr>
<th>Name</th>
<th>event</th>
<th>Description</th>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Asynchronous event registration and handling.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RO_ins</td>
<td>id</td>
<td>int</td>
<td></td>
<td></td>
<td></td>
<td>An ID, monotonically increasing, and local to the current session</td>
</tr>
<tr>
<td>RO_ins</td>
<td>timestamp</td>
<td>datetime</td>
<td></td>
<td></td>
<td></td>
<td>The time at which the event occurred</td>
</tr>
<tr>
<td>RO_ins</td>
<td>class</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td>The name of the class of the object that changed</td>
</tr>
<tr>
<td>RO_ins</td>
<td>operation</td>
<td>event_operation</td>
<td></td>
<td></td>
<td></td>
<td>The operation that was performed</td>
</tr>
<tr>
<td>RO_ins</td>
<td>ref</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td>A reference to the object that changed</td>
</tr>
<tr>
<td>RO_ins</td>
<td>obj_uuid</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td>The uuid of the object that changed</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string set</td>
<td>classes</td>
<td>register for events for the indicated classes</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: unregister

Overview:
Unregisters this session with the event system.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string set</td>
<td>classes</td>
<td>remove this session’s registration for the indicated classes</td>
</tr>
</tbody>
</table>

Return Type: void
RPC name: next

Overview:
Blocking call which returns a (possibly empty) batch of events. This method is only recommended for legacy use. New development should use event.from which supercedes this method.

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 new token and a (possibly empty) batch of events. The returned token can be used in subsequent calls to this function.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string set</td>
<td>classes</td>
<td>register for events for the indicated classes</td>
</tr>
<tr>
<td>string</td>
<td>token</td>
<td>A token representing the point from which to generate database events. The empty string represents the beginning.</td>
</tr>
<tr>
<td>float</td>
<td>timeout</td>
<td>Return after this many seconds if no events match</td>
</tr>
</tbody>
</table>

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

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>class</td>
<td>class of the object</td>
</tr>
<tr>
<td>string</td>
<td>ref</td>
<td>A reference to the object that will be changed.</td>
</tr>
</tbody>
</table>

Return Type: string
the event ID
## 2.10 Class: pool

### 2.10.1 Fields for class: pool

<table>
<thead>
<tr>
<th>Name</th>
<th>pool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Quals</strong></td>
<td><strong>Field</strong></td>
<td><strong>Type</strong></td>
</tr>
<tr>
<td>RO run</td>
<td>uuid</td>
<td>string</td>
</tr>
<tr>
<td>RW</td>
<td>name_label</td>
<td>string</td>
</tr>
<tr>
<td>RW</td>
<td>name_description</td>
<td>string</td>
</tr>
<tr>
<td>RO run</td>
<td>master</td>
<td>host ref</td>
</tr>
<tr>
<td>RW</td>
<td>default_SR</td>
<td>SR ref</td>
</tr>
<tr>
<td>RW</td>
<td>suspend_image_SR</td>
<td>SR ref</td>
</tr>
<tr>
<td>RW</td>
<td>crash_dump_SR</td>
<td>SR ref</td>
</tr>
<tr>
<td>RW</td>
<td>other_config</td>
<td>(string → string) map</td>
</tr>
<tr>
<td>RO run</td>
<td>ha_enabled</td>
<td>bool</td>
</tr>
<tr>
<td>RO run</td>
<td>ha_configuration</td>
<td>(string → string) map</td>
</tr>
<tr>
<td>RO run</td>
<td>ha_statefiles</td>
<td>string set</td>
</tr>
<tr>
<td>RO run</td>
<td>ha_host_failures_to_tolerate</td>
<td>int</td>
</tr>
<tr>
<td>RO run</td>
<td>ha_plan_exists_for</td>
<td>int</td>
</tr>
<tr>
<td>RW</td>
<td>ha_allow_overcommit</td>
<td>bool</td>
</tr>
<tr>
<td>RO run</td>
<td>ha_overcommitted</td>
<td>bool</td>
</tr>
<tr>
<td>RO run</td>
<td>blobs</td>
<td>(string → blob ref) map</td>
</tr>
<tr>
<td>RW</td>
<td>tags</td>
<td>string set</td>
</tr>
<tr>
<td>RW</td>
<td>gui_config</td>
<td>(string → string) map</td>
</tr>
<tr>
<td>RW</td>
<td>health_check_config</td>
<td>(string → string) map</td>
</tr>
<tr>
<td>RO run</td>
<td>wlb_url</td>
<td>string</td>
</tr>
<tr>
<td>RO run</td>
<td>wlb_username</td>
<td>string</td>
</tr>
<tr>
<td>RW</td>
<td>wlb_enabled</td>
<td>bool</td>
</tr>
<tr>
<td>RW</td>
<td>wlb_verify_cert</td>
<td>bool</td>
</tr>
</tbody>
</table>

If set to false then operations which cause the Pool to become overcommitted will be blocked.
### 2.10. CLASS: POOL

#### 2.10.2 RPCs associated with class: pool

**RPC name:** join

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

**Signature:**
```java
void join (session_id s, string master_address, string master_username, string master_password)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>master_address</td>
<td>The hostname of the master of the pool to join</td>
</tr>
<tr>
<td>string</td>
<td>master_username</td>
<td>The username of the master (for initial authentication)</td>
</tr>
<tr>
<td>string</td>
<td>master_password</td>
<td>The password for the master (for initial authentication)</td>
</tr>
</tbody>
</table>

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>master_address</td>
<td>The hostname of the master of the pool to join</td>
</tr>
<tr>
<td>string</td>
<td>master_username</td>
<td>The username of the master (for initial authentication)</td>
</tr>
<tr>
<td>string</td>
<td>master_password</td>
<td>The password for the master (for initial authentication)</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host to eject</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>master_address</td>
<td>The hostname of the master</td>
</tr>
</tbody>
</table>

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 successfully 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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>device</td>
<td>physical interface on which to create the VLAN interface</td>
</tr>
<tr>
<td>network ref</td>
<td>network</td>
<td>network to which this interface should be connected</td>
</tr>
<tr>
<td>int</td>
<td>VLAN</td>
<td>VLAN tag for the new interface</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>pif</td>
<td>physical interface on any particular host, that identifies the PIF on which to create the (pool-wide) VLAN interface</td>
</tr>
<tr>
<td>network ref</td>
<td>network</td>
<td>network to which this interface should be connected</td>
</tr>
<tr>
<td>int</td>
<td>VLAN</td>
<td>VLAN tag for the new interface</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref set</td>
<td>heartbeat_srs</td>
<td>Set of SRs to use for storage heartbeating</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>configuration</td>
<td>Detailed HA configuration to apply</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host who should become the new master</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>seconds</td>
<td>The number of seconds to block the restart thread for</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>n</td>
<td>The number of host failures to plan for</td>
</tr>
</tbody>
</table>

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:

```c
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:

```c
int ha_compute_hypothetical_max_host_failures_to_tolerate (session_id s, (VM ref -> string) map configuration)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(VM ref -&gt; string) map</td>
<td>configuration</td>
<td>Map of protected VM reference to restart priority</td>
</tr>
</tbody>
</table>

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:

```c
((VM ref -> (string -> string) map) map) ha_compute_vm_failover_plan (session_id s, host ref set failed_hosts, VM ref set failed_vms)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref set</td>
<td>failed_hosts</td>
<td>The set of hosts to assume have failed</td>
</tr>
<tr>
<td>VM ref set</td>
<td>failed_vms</td>
<td>The set of VMs to restart</td>
</tr>
</tbody>
</table>

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:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>The pool</td>
</tr>
<tr>
<td>int</td>
<td>value</td>
<td>New number of host failures to consider</td>
</tr>
</tbody>
</table>

**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:**

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

**Arguments:**

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

**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:**

```c
void enable_external_auth (session_id s, pool ref pool, (string -> string) map config, string service_name)
```

**Arguments:**
<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>pool</td>
<td>The pool whose external authentication should be enabled</td>
</tr>
<tr>
<td>(string → string)</td>
<td>config</td>
<td>A list of key-values containing the configuration data</td>
</tr>
<tr>
<td>string</td>
<td>service_name</td>
<td>The name of the service</td>
</tr>
<tr>
<td>string</td>
<td>auth_type</td>
<td>The type of authentication (e.g. AD for Active Directory)</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>pool</td>
<td>The pool whose external authentication should be disabled</td>
</tr>
<tr>
<td>(string → string)</td>
<td>config</td>
<td>Optional parameters as a list of key-values containing the configuration data</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>pool</td>
<td>The pool where to detect non-homogeneous external authentication configuration</td>
</tr>
</tbody>
</table>

Return Type: void
RPC name: initialize_wlb

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

Signature:

```c
void initialize_wlb (session_id s, string wlb_url, string wlb_username, string wlb_password, string xenserver_username, string xenserver_password);
```

Arguments:

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>wlb_url</td>
<td>The ip address and port to use when accessing the wlb server</td>
</tr>
<tr>
<td>string</td>
<td>wlb_username</td>
<td>The username used to authenticate with the wlb server</td>
</tr>
<tr>
<td>string</td>
<td>wlb_password</td>
<td>The password used to authenticate with the wlb server</td>
</tr>
<tr>
<td>string</td>
<td>xenserver_username</td>
<td>The username used by the wlb server to authenticate with the xenserver</td>
</tr>
<tr>
<td>string</td>
<td>xenserver_password</td>
<td>The password used by the wlb server to authenticate with the xenserver</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: deconfigure_wlb

Overview:
Permanently deconfigures workload balancing monitoring on this pool.

Signature:

```c
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:

```c
void send_wlb_configuration (session_id s, (string -> string) map config);
```

Arguments:

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(string -&gt; string) map</td>
<td>config</td>
<td>The configuration to use in optimizing this pool</td>
</tr>
</tbody>
</table>

Return Type: void
RPC name: retrieve_wlb_configuration

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

Signature:

\[((\text{string} \rightarrow \text{string}) \text{map}) \text{ retrieve}_wlb\_configuration \text{ (session\_id s)}\]

Return Type: (\text{string} \rightarrow \text{string}) \text{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:

\[((\text{VM ref} \rightarrow \text{string set}) \text{map}) \text{ retrieve}_wlb\_recommendations \text{ (session\_id s)}\]

Return Type: (\text{VM ref} \rightarrow \text{string set}) \text{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:

\text{string} \text{send}\_test\_post \text{(session\_id s, string host, int port, string body)}

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>host</td>
<td></td>
</tr>
<tr>
<td>int</td>
<td>port</td>
<td></td>
</tr>
<tr>
<td>string</td>
<td>body</td>
<td></td>
</tr>
</tbody>
</table>

Return Type: string
The response

RPC name: certificate_install

Overview:
Install an SSL certificate pool-wide.

Signature:

\text{void certificate\_install (session\_id s, string name, string cert)}
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>name</td>
<td>A name to give the certificate</td>
</tr>
<tr>
<td>string</td>
<td>cert</td>
<td>The certificate</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: certificate_uninstall

Overview:
Remove an SSL certificate.

Signature:

void certificate_uninstall (session_id s, string name)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>name</td>
<td>The certificate name</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>name</td>
<td>A name to give the CRL</td>
</tr>
<tr>
<td>string</td>
<td>cert</td>
<td>The CRL</td>
</tr>
</tbody>
</table>

Return Type: void
RPC name: crl_uninstall
Overview:
Remove an SSL certificate revocation list.
Signature:
void crl_uninstall (session_id s, string name)
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>name</td>
<td>The CRL name</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
</table>
| SR ref  | sr   | SR to hold the redo log,
### 2.10. CLASS: POOL

**Return Type:** void

**RPC name:** disable_redo_log

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

**Signature:**

```c
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:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>address</td>
<td>IP address of the vswitch controller.</td>
</tr>
</tbody>
</table>

**Return Type:** void

**RPC name:** test_archive_target

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

**Signature:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>Reference to the pool</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>config</td>
<td>Location config settings to test</td>
</tr>
</tbody>
</table>

**Return Type:** string

An XMLRPC result
RPC name: enable_local_storage_caching

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

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>Reference to the pool</td>
</tr>
</tbody>
</table>

Return Type: void

---

RPC name: disable_local_storage_caching

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

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>Reference to the pool</td>
</tr>
</tbody>
</table>

Return Type: void

---

RPC name: get_license_state

Overview:
This call returns the license state for the pool.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>Reference to the pool</td>
</tr>
</tbody>
</table>

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:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>Reference to the pool</td>
</tr>
<tr>
<td>string</td>
<td>edition</td>
<td>The requested edition</td>
</tr>
</tbody>
</table>

**Return Type:** void

**RPC name: enable_ssl_legacy**

**Overview:**
Sets ssl_legacy true on each host, pool-master last. See Host.ssl_legacy and Host.set_ssl_legacy.

**Signature:**

```c
void enable_ssl_legacy (session_id s, pool ref self)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>(ignored)</td>
</tr>
</tbody>
</table>

**Return Type:** void

**RPC name: disable_ssl_legacy**

**Overview:**
Sets ssl_legacy true on each host, pool-master last. See Host.ssl_legacy and Host.set_ssl_legacy.

**Signature:**

```c
void disable_ssl_legacy (session_id s, pool ref self)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>(ignored)</td>
</tr>
</tbody>
</table>
RPC name: has_extension

Overview:
Return true if the extension is available on the pool.

Signature:

bool has_extension (session_id s, pool ref self, string name)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>The pool</td>
</tr>
<tr>
<td>string</td>
<td>name</td>
<td>The name of the API call</td>
</tr>
</tbody>
</table>

Return Type: bool
True if the extension exists, false otherwise

RPC name: add_to_guest_agent_config

Overview:
Add a key-value pair to the pool-wide guest agent configuration.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>The pool</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>The key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>The value to add</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: remove_from_guest_agent_config

Overview:
Remove a key-value pair from the pool-wide guest agent configuration.

Signature:

void remove_from_guest_agent_config (session_id s, pool ref self, string key)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>The pool</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>The key to remove</td>
</tr>
</tbody>
</table>

Return Type: void
RPC name: get_all

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

Signature:
(pool ref set) get_all (session_id s)

Return Type: pool ref set
references to all objects

RPC name: get_all_records

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

Signature:
((pool ref -> pool record) map) get_all_records (session_id s)

Return Type: (pool ref -> pool record) map
records of all objects

RPC name: get_uuid

Overview:
Get the uuid field of the given pool.

Signature:
string get_uuid (session_id s, pool ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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)
```
2.10. CLASS: POOL

CHAPTER 2. API REFERENCE

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>SR ref</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>SR ref</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>SR ref</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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)
2.10. CLASS: POOL

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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)
```

86
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

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

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>bool</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string set
value of the field

RPC name: set_tags

Overview:
Set the tags field of the given pool.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string set</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to add</td>
</tr>
</tbody>
</table>

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:

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

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_health_check_config

Overview:
Get the health_check_config field of the given pool.

Signature:

```
((string -> string) map) get_health_check_config (session_id s, pool ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: (string → string) map

value of the field

RPC name: set_health_check_config

Overview:
Set the health_check_config field of the given pool.

Signature:

```
void set_health_check_config (session_id s, pool ref self, (string -> string) map value)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: add_to_health_check_config

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

Signature:
void add_to_health_check_config (session_id s, pool ref self, string key, string value)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: remove_from_health_check_config

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

Signature:
void remove_from_health_check_config (session_id s, pool ref self, string key)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get_wlb_username

Overview:
Get the wlb_username field of the given pool.

Signature:

```c
string get_wlb_username (session_id s, pool ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get_wlb_enabled

Overview:
Get the wlb_enabled field of the given pool.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: bool
value of the field

RPC name: set_wlb_enabled

Overview:
Set the wlb_enabled field of the given pool.

Signature:

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

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>bool</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_wlb_verify_cert

Overview:
Get the wlb_verify_cert field of the given pool.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>bool</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_redo_log_enabled

Overview:
Get the redo_log_enabled field of the given pool.

Signature:

```cpp
bool get_redo_log_enabled (session_id s, pool ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: \((\text{string} \rightarrow \text{string}) \text{ map}\)
value of the field

RPC name: get_metadata_VDIs

Overview:
Get the metadata_VDIs field of the given pool.

Signature:

\((\text{VDI ref set}) \text{ get_metadata_VDIs (session_id s, pool ref self)}\)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: \(\text{VDI ref set}\)
value of the field

RPC name: get_ha_cluster_stack

Overview:
Get the ha_cluster_stack field of the given pool.

Signature:

\(\text{string get_ha_cluster_stack (session_id s, pool ref self)}\)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: \(\text{string}\)
value of the field

RPC name: get_allowed_operations

Overview:
Get the allowed_operations field of the given pool.

Signature:

\((\text{pool_allowed_operations set}) \text{ get_allowed_operations (session_id s, pool ref self)}\)
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: `pool.allowed_operations` set
value of the field

RPC name: `get.current_operations`

Overview:
Get the current_operations field of the given pool.

Signature:
```plaintext
((string -> pool.allowed_operations) map) get_current_operations (session_id s, pool ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: `(string -> pool_allowed_operations) map`
value of the field

RPC name: `get.guest_agent_config`

Overview:
Get the guest_agent_config field of the given pool.

Signature:
```plaintext
((string -> string) map) get_guest_agent_config (session_id s, pool ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: `(string -> string) map`
value of the field

RPC name: `get.cpu_info`

Overview:
Get the cpu_info field of the given pool.

Signature:
```plaintext
((string -> string) map) get_cpu_info (session_id s, pool ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: \((\text{string} \rightarrow \text{string})\) map
value of the field

RPC name: get_by_uoid

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

Signature:

\((\text{pool ref}) \text{get_by_uoid} (\text{session_id s, string uuid})\)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

Return Type: \(\text{pool ref}\)
reference to the object

RPC name: get_record

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

Signature:

\((\text{pool record}) \text{get_record} (\text{session_id s, pool ref self})\)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: \(\text{pool record}\)
all fields from the object
2.11 Class: pool_patch

2.11.1 Fields for class: pool_patch

<table>
<thead>
<tr>
<th>Name</th>
<th>pool_patch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Pool-wide patches.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO</td>
<td>run</td>
<td>uuid</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>RO</td>
<td>ins</td>
<td>name_label</td>
<td>a human-readable name</td>
</tr>
<tr>
<td>RO</td>
<td>ins</td>
<td>name_description</td>
<td>a notes field containing human-readable description</td>
</tr>
<tr>
<td>RO</td>
<td>run</td>
<td>version</td>
<td>Patch version number</td>
</tr>
<tr>
<td>RO</td>
<td>run</td>
<td>size</td>
<td>Size of the patch</td>
</tr>
<tr>
<td>RO</td>
<td>run</td>
<td>pool_applied</td>
<td>This patch should be applied across the entire pool</td>
</tr>
<tr>
<td>RO</td>
<td>run</td>
<td>host_patches</td>
<td>This hosts this patch is applied to.</td>
</tr>
<tr>
<td>RO</td>
<td>run</td>
<td>after_apply_guidance</td>
<td>What the client should do after this patch has been applied.</td>
</tr>
<tr>
<td>RW</td>
<td>other_config</td>
<td>(string → string) map</td>
<td>additional configuration</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool_patch ref</td>
<td>self</td>
<td>The patch to apply</td>
</tr>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host to apply the patch too</td>
</tr>
</tbody>
</table>

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_patch ref -→ patch output.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool_patch ref</td>
<td>self</td>
<td>The patch to apply</td>
</tr>
</tbody>
</table>
RPC name: precheck
Overview:
Execute the precheck stage of the selected patch on a host and return its output.
Signature:

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

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool_patch ref</td>
<td>self</td>
<td>The patch whose prechecks will be run</td>
</tr>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host to run the prechecks on</td>
</tr>
</tbody>
</table>

Return Type: string
the output of the patch prechecks

RPC name: clean
Overview:
Removes the patch’s files from the server.
Signature:

```plaintext
void clean (session_id s, pool_patch ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool_patch ref</td>
<td>self</td>
<td>The patch to clean up</td>
</tr>
</tbody>
</table>

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:

```plaintext
void pool_clean (session_id s, pool_patch ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool_patch ref</td>
<td>self</td>
<td>The patch to clean up</td>
</tr>
</tbody>
</table>

Return Type: void
2.11. CLASS: POOL_PATCH

CHAPTER 2. API REFERENCE

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool_patch ref</td>
<td>self</td>
<td>The patch to destroy</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool_patch ref</td>
<td>self</td>
<td>The patch to clean up</td>
</tr>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host on which to clean the patch</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string \rightarrow string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>label</td>
<td>label of object to return</td>
</tr>
</tbody>
</table>

Return Type: pool_patch ref set
references to objects with matching names
2.12 Class: VM

2.12.1 Fields for class: VM

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM</td>
<td></td>
<td>A virtual machine (or 'guest').</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO</td>
<td>uuid</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>RO</td>
<td>allowed_operations</td>
<td>vm_operations set</td>
<td>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.</td>
</tr>
<tr>
<td>RO</td>
<td>current_operations</td>
<td>(string → vm_operations) map</td>
<td>links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.</td>
</tr>
<tr>
<td>RO</td>
<td>power_state</td>
<td>vm_power_state</td>
<td>Current power state of the machine</td>
</tr>
<tr>
<td>RW</td>
<td>name_label</td>
<td>string</td>
<td>a human-readable name</td>
</tr>
<tr>
<td>RW</td>
<td>name_description</td>
<td>string</td>
<td>a notes field containing human-readable description</td>
</tr>
<tr>
<td>RW</td>
<td>user_version</td>
<td>int</td>
<td>Creators of VMs and templates may store version information here.</td>
</tr>
<tr>
<td>RW</td>
<td>is_a_template</td>
<td>bool</td>
<td>true if this is a template. Template VMs can never be started, they are used only for cloning other VMs</td>
</tr>
<tr>
<td>RO</td>
<td>suspend_VDI</td>
<td>VDI ref</td>
<td>The VDI that a suspend image is stored on. (Only has meaning if VM is currently suspended)</td>
</tr>
<tr>
<td>RO</td>
<td>resident_on</td>
<td>host ref</td>
<td>the host the VM is currently resident on 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.</td>
</tr>
<tr>
<td>RW</td>
<td>affinity</td>
<td>host ref</td>
<td></td>
</tr>
<tr>
<td>RO</td>
<td>memory_overhead</td>
<td>int</td>
<td>Virtualization memory overhead (bytes).</td>
</tr>
<tr>
<td>RO</td>
<td>memory_target</td>
<td>int</td>
<td>Dynamically-set memory target (bytes). The value of this field indicates the current target for memory available to this VM. 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.</td>
</tr>
<tr>
<td>RO</td>
<td>memory_static_max</td>
<td>int</td>
<td>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. Configuration parameters for the selected VCPU policy</td>
</tr>
<tr>
<td>RW</td>
<td>VCPUs_params</td>
<td>(string → string) map</td>
<td></td>
</tr>
</tbody>
</table>

107
### 2.12. CLASS: VM

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO</td>
<td>VCPUs_max</td>
<td>Max number of VCPUs</td>
</tr>
<tr>
<td>RO</td>
<td>VCPUs_at_startup</td>
<td>Boot number of VCPUs</td>
</tr>
<tr>
<td>RW</td>
<td>actions_after_shutdown</td>
<td>action to take after the guest has shutdown itself</td>
</tr>
<tr>
<td>RW</td>
<td>actions_after_reboot</td>
<td>action to take after the guest has rebooted itself</td>
</tr>
<tr>
<td>RO</td>
<td>consoles</td>
<td>virtual console devices</td>
</tr>
<tr>
<td>RO</td>
<td>VIFs</td>
<td>virtual network interfaces</td>
</tr>
<tr>
<td>RO</td>
<td>VBDs</td>
<td>virtual block devices</td>
</tr>
<tr>
<td>RO</td>
<td>crash_dumps</td>
<td>crash dumps associated with this VM</td>
</tr>
<tr>
<td>RO</td>
<td>VTPMs</td>
<td>virtual TPMS</td>
</tr>
<tr>
<td>RW</td>
<td>PV_bootloader</td>
<td>name of or path to bootloader</td>
</tr>
<tr>
<td>RW</td>
<td>PV_kernel</td>
<td>path to the kernel</td>
</tr>
<tr>
<td>RW</td>
<td>PV_ramdisk</td>
<td>path to the initrd</td>
</tr>
<tr>
<td>RW</td>
<td>PV_args</td>
<td>kernel command-line arguments</td>
</tr>
<tr>
<td>RW</td>
<td>PV_bootloader_args</td>
<td>miscellaneous arguments for the bootloader</td>
</tr>
<tr>
<td>RW</td>
<td>PV_legacy_args</td>
<td>to make Zurich guests boot</td>
</tr>
<tr>
<td>RW</td>
<td>HVM_boot_policy</td>
<td>HVM boot policy</td>
</tr>
<tr>
<td>RW</td>
<td>HVM_boot_params</td>
<td>HVM boot params</td>
</tr>
<tr>
<td>RO</td>
<td>HVM_shadow_multiplier</td>
<td>multiplier applied to the amount of shadow that will be made available to</td>
</tr>
<tr>
<td>RW</td>
<td>platform</td>
<td>platform-specific configuration</td>
</tr>
<tr>
<td>RW</td>
<td>PCI_bus</td>
<td>PCI bus path for pass-through devices</td>
</tr>
<tr>
<td>RW</td>
<td>other_config</td>
<td>additional configuration</td>
</tr>
<tr>
<td>RO</td>
<td>domid</td>
<td>domain ID (if available, -1 otherwise)</td>
</tr>
<tr>
<td>RO</td>
<td>domarch</td>
<td>Domain architecture (if available, null string otherwise)</td>
</tr>
<tr>
<td>RO</td>
<td>last_boot_CPU_flags</td>
<td>describes the CPU flags on which the VM was last booted</td>
</tr>
<tr>
<td>RO</td>
<td>is_control_domain</td>
<td>true if this is a control domain (domain 0 or a driver domain)</td>
</tr>
<tr>
<td>RO</td>
<td>metrics</td>
<td>metrics associated with this VM</td>
</tr>
<tr>
<td>RO</td>
<td>guest_metrics</td>
<td>metrics associated with the running guest</td>
</tr>
<tr>
<td>RO</td>
<td>last_booted_record</td>
<td>marshallled value containing VM record at time of last boot, updated</td>
</tr>
<tr>
<td>RW</td>
<td>recommendations</td>
<td>An XML specification of recommended values and ranges for properties of</td>
</tr>
<tr>
<td>RW</td>
<td>xenstore_data</td>
<td>this VM data to be inserted into the xenstore tree (/local/domain/¡domid¿/vm-data) after the VM is created.</td>
</tr>
<tr>
<td>RO</td>
<td>ha_always_run</td>
<td>if true then the system will attempt to keep the VM running as much as</td>
</tr>
<tr>
<td>RO</td>
<td>ha_restart_priority</td>
<td>has possible values: “best-effort” meaning “try to restart this VM if</td>
</tr>
</tbody>
</table>

108
### 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 VMs that cannot be started.
storage repository in which the VM's disk images are stored (e.g. Copy on Write).

**Signature:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM to be snapshotted</td>
</tr>
<tr>
<td>string</td>
<td>new_name</td>
<td>The name of the snapshotted VM</td>
</tr>
</tbody>
</table>

**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:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM to be snapshotted</td>
</tr>
<tr>
<td>string</td>
<td>new_name</td>
<td>The name of the snapshotted VM</td>
</tr>
</tbody>
</table>

**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_NOT

**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:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM to be cloned</td>
</tr>
<tr>
<td>string</td>
<td>new_name</td>
<td>The name of the cloned VM</td>
</tr>
</tbody>
</table>
Return Type: VM ref
The reference of the newly created VM.

Possible Error Codes: VM.BAD_POWER_STATE, SR_FULL, OPERATION_NOT_ALLOWED, LICENCE_RESTRICTION

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM to be copied</td>
</tr>
<tr>
<td>string</td>
<td>new_name</td>
<td>The name of the copied VM</td>
</tr>
<tr>
<td>SR ref</td>
<td>sr</td>
<td>An SR to copy all the VM’s disks into (if an invalid reference then it uses the existing SRs)</td>
</tr>
</tbody>
</table>

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

Possible Error Codes: VM.BAD_POWER_STATE, SR_FULL, OPERATION_NOT_ALLOWED, LICENCE_RESTRICTION

RPC name: revert

Overview:
Reverts the specified VM to a previous state.

Signature:
void revert (session_id s, VM ref snapshot)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>snapshot</td>
<td>The snapshotted state that we revert to</td>
</tr>
</tbody>
</table>

Return Type: void
Possible Error Codes: VM.BAD_POWER_STATE, OPERATION_NOT_ALLOWED, SR_FULL, VM.REVERT_FAILED

RPC name: checkpoint

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

**Signature:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM to be checkpointed</td>
</tr>
<tr>
<td>string</td>
<td>new_name</td>
<td>The name of the checkpointed VM</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM to be provisioned</td>
</tr>
</tbody>
</table>

**Return Type:** void

**Possible Error Codes:** VM_BAD_POWER_STATE, SR_FULL, OPERATION_NOT_ALLOWED, LICENCE_RESTRICTION

**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 startPaused, bool force)
```

**Arguments:**
<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM to start</td>
</tr>
<tr>
<td>bool</td>
<td>start paused</td>
<td>Instantiate VM in paused state if set to true</td>
</tr>
<tr>
<td>bool</td>
<td>force</td>
<td>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)</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM to start</td>
</tr>
<tr>
<td>host ref</td>
<td>host</td>
<td>The Host on which to start the VM</td>
</tr>
<tr>
<td>bool</td>
<td>start paused</td>
<td>Instantiate VM in paused state if set to true</td>
</tr>
<tr>
<td>bool</td>
<td>force</td>
<td>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)</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM to pause</td>
</tr>
</tbody>
</table>
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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM to unpause</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM to shutdown</td>
</tr>
</tbody>
</table>

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:

```c
void shutdown (session_id s, VM ref vm)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM to shutdown</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM to shutdown</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM to destroy</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM to reset</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: hard_reboot

Overview:
Stop executing the specified VM without attempting a clean shutdown and immediately restart the VM.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM to reboot</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM to suspend</td>
</tr>
</tbody>
</table>

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:

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

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

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:

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

Arguments:

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

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.

Signature:

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

Arguments:
### type name description

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM to migrate</td>
</tr>
<tr>
<td>host ref</td>
<td>host</td>
<td>The target host</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>options</td>
<td>Extra configuration operations</td>
</tr>
</tbody>
</table>

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:**

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

**Arguments:**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
<tr>
<td>int</td>
<td>nvcpu</td>
<td>The number of VCPUs</td>
</tr>
</tbody>
</table>

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:**

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

**Arguments:**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>The key</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>The value</td>
</tr>
</tbody>
</table>

Return Type: void

---

### RPC name: set_ha_restart_priority

**Overview:**
Set the value of the ha.restart.priority field.

**Signature:**

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

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>The value</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: set_ha_always_run

Overview: This message is deprecated Set the value of the ha_always_run.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
<tr>
<td>bool</td>
<td>value</td>
<td>The value</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: compute_memory_overhead

Overview:
Computes the virtualization memory overhead of a VM.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM for which to compute the memory overhead</td>
</tr>
</tbody>
</table>

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:

```c
void set_memory_dynamic_max (session_id s, VM ref self, int value)
```
### Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM to modify</td>
</tr>
<tr>
<td>int</td>
<td>value</td>
<td>The new value of memory_dynamic_min</td>
</tr>
</tbody>
</table>

**Return Type:** void

### RPC name: set\_memory\_dynamic\_min

**Overview:**
Set the value of the memory\_dynamic\_min field.

**Signature:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM to modify</td>
</tr>
<tr>
<td>int</td>
<td>value</td>
<td>The new value of memory_dynamic_min</td>
</tr>
</tbody>
</table>

**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:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
<tr>
<td>int</td>
<td>min</td>
<td>The new minimum value</td>
</tr>
<tr>
<td>int</td>
<td>max</td>
<td>The new maximum value</td>
</tr>
</tbody>
</table>

**Return Type:** void

### RPC name: set\_memory\_static\_max

**Overview:**
Set the value of the memory\_static\_max field.

**Signature:**

```c
void set_memory_static_max (session_id s, VM ref self, int value)
```
2.12. CLASS: VM

CHAPTER 2. API REFERENCE

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM to modify</td>
</tr>
<tr>
<td>int</td>
<td>value</td>
<td>The new value of memory.static_max</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM to modify</td>
</tr>
<tr>
<td>int</td>
<td>value</td>
<td>The new value of memory.static_min</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
<tr>
<td>int</td>
<td>min</td>
<td>The new minimum value</td>
</tr>
<tr>
<td>int</td>
<td>max</td>
<td>The new maximum value</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
<tr>
<td>int</td>
<td>static_min</td>
<td>The new value of memory_static_min.</td>
</tr>
<tr>
<td>int</td>
<td>static_max</td>
<td>The new value of memory_static_max.</td>
</tr>
<tr>
<td>int</td>
<td>dynamic_min</td>
<td>The new value of memory_dynamic_min.</td>
</tr>
<tr>
<td>int</td>
<td>dynamic_max</td>
<td>The new value of memory_dynamic_max.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
<tr>
<td>int</td>
<td>target</td>
<td>The target in bytes</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
</tbody>
</table>

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)
2.12. CLASS: VM

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
</tbody>
</table>

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

RPC name: set_HVM_shadow_multipler

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
<tr>
<td>float</td>
<td>value</td>
<td>The new shadow memory multiplier to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
<tr>
<td>float</td>
<td>multiplier</td>
<td>The new shadow memory multiplier to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
<tr>
<td>int</td>
<td>value</td>
<td>The new maximum number of VCPUs</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
<tr>
<td>int</td>
<td>value</td>
<td>The new maximum number of VCPUs</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>The key to send</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM</td>
</tr>
<tr>
<td>string</td>
<td>trigger</td>
<td>The trigger to send</td>
</tr>
</tbody>
</table>

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:

\[
\text{int maximise_memory (session_id s, VM ref self, int total, bool approximate)}
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
<tr>
<td>int</td>
<td>total</td>
<td>Total amount of physical RAM to fit within</td>
</tr>
<tr>
<td>bool</td>
<td>approximate</td>
<td>If false the limit is calculated with the guest’s current exact configuration. Otherwise a more approximate calculation is performed</td>
</tr>
</tbody>
</table>

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:

\[
\text{(VM ref) migrate_send (session_id s, VM ref vm, (string \rightarrow string) map dest, bool live, (VDI ref \rightarrow SR ref) map vdi, (VIF ref \rightarrow network ref) map vif, (string \rightarrow string) map options)}
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM</td>
</tr>
<tr>
<td>(string \rightarrow string) map</td>
<td>dest</td>
<td>The result of a Host.migrate_receive call.</td>
</tr>
<tr>
<td>bool</td>
<td>live</td>
<td>Live migration</td>
</tr>
<tr>
<td>(VDI ref \rightarrow SR ref) map</td>
<td>vdi_map</td>
<td>Map of source VDI to destination SR</td>
</tr>
<tr>
<td>(VIF ref \rightarrow network ref) map</td>
<td>vif_map</td>
<td>Map of source VIF to destination network</td>
</tr>
<tr>
<td>(string \rightarrow string) map</td>
<td>options</td>
<td>Other parameters</td>
</tr>
</tbody>
</table>

Return Type: VM ref
The reference of the newly created VM in the destination pool

Possible Error Codes: VM_BAD_POWER_STATE, LICENCE_RESTRICTION

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) map vdi_map, (VIF ref -> network ref) map vif_map, (string -> string) map options)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>dest</td>
<td>The result of a VM.migrate_receive call.</td>
</tr>
<tr>
<td>bool</td>
<td>live</td>
<td>Live migration</td>
</tr>
<tr>
<td>(VDI ref -&gt; SR ref) map</td>
<td>vdi_map</td>
<td>Map of source VDI to destination SR</td>
</tr>
<tr>
<td>(VIF ref -&gt; network ref) map</td>
<td>vif_map</td>
<td>Map of source VIF to destination network</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>options</td>
<td>Other parameters</td>
</tr>
</tbody>
</table>

Return Type: void

Possible Error Codes: LICENCE_RESTRICTION

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM whose boot-time state to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM to interrogate</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
<tr>
<td>string</td>
<td>data_source</td>
<td>The data source to record</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
<tr>
<td>string</td>
<td>data_source</td>
<td>The data source to query</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
<tr>
<td>string</td>
<td>data_source</td>
<td>The data source whose archives are to be forgotten</td>
</tr>
</tbody>
</table>

**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:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>vm_operations</td>
<td>op</td>
<td>proposed operation</td>
</tr>
</tbody>
</table>

**Return Type:** void

### RPC name: update_allowed_operations

**Overview:**
Recomputes the list of acceptable operations.

**Signature:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**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:**

```c
(string set) get_allowed_VBD_devices (session_id s, VM ref vm)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM to query</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM to query</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host</td>
</tr>
</tbody>
</table>

Return Type: void

Possible Error Codes: HOST_NOT_ENOUGH_FREE_MEMORY, VMQUIRES_SR, VM_HOST_INCOMPATIBLE_VERSION, VM_HOST_INCOMPATIBLE_VIRTUAL_HARDWARE_PLATFORM_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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM</td>
</tr>
<tr>
<td>string</td>
<td>name</td>
<td>The name associated with the blob</td>
</tr>
<tr>
<td>string</td>
<td>mime_type</td>
<td>The mime type for the data. Empty string translates to application/octet-stream</td>
</tr>
<tr>
<td>bool</td>
<td>public</td>
<td>True if the blob should be publicly available</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
</tbody>
</table>

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:

\[
((\text{host ref} \rightarrow \text{string set}) \map) \text{retrieve_wlb_recommendations} (\text{session_id s, VM ref vm})
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM</td>
</tr>
</tbody>
</table>

Return Type: \((\text{host ref} \rightarrow \text{string set}) \map\)
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:

\[
\text{void copy_bios_strings} (\text{session_id s, VM ref vm, host ref host})
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The VM to modify</td>
</tr>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host to copy the BIOS strings from</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: set_protection_policy

Overview:
Set the value of the protection policy field.

Signature:

\[
\text{void set_protection_policy} (\text{session_id s, VM ref self, VMPP ref value})
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
<tr>
<td>VMPP ref</td>
<td>value</td>
<td>The value</td>
</tr>
</tbody>
</table>
RPC name: set_start_delay

Overview:
Set this VM’s start delay in seconds.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
<tr>
<td>int</td>
<td>value</td>
<td>This VM’s start delay in seconds</td>
</tr>
</tbody>
</table>

Return Type: void

---

RPC name: set_shutdown_delay

Overview:
Set this VM’s shutdown delay in seconds.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
<tr>
<td>int</td>
<td>value</td>
<td>This VM’s shutdown delay in seconds</td>
</tr>
</tbody>
</table>

Return Type: void

---

RPC name: set_order

Overview:
Set this VM’s boot order.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
<tr>
<td>int</td>
<td>value</td>
<td>This VM’s boot order</td>
</tr>
</tbody>
</table>

Return Type: void
RPC name: set_suspend_VDI

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

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
<tr>
<td>VDI ref</td>
<td>value</td>
<td>The suspend VDI uuid</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: assert_can_be_recovered

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

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM to recover</td>
</tr>
<tr>
<td>session ref</td>
<td>session_to</td>
<td>The session to which the VM is to be recovered.</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM for which the SRs have to be recovered</td>
</tr>
<tr>
<td>session ref</td>
<td>session_to</td>
<td>The session to which the SRs of the VM have to be recovered.</td>
</tr>
</tbody>
</table>

Return Type: SR ref set
refs for SRs required to recover the VM

RPC name: recover

Overview:
Recover the VM.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM to recover</td>
</tr>
<tr>
<td>session ref</td>
<td>session_to</td>
<td>The session to which the VM is to be recovered.</td>
</tr>
<tr>
<td>bool</td>
<td>force</td>
<td>Whether the VM should replace newer versions of itself.</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: import_convert

Overview:
Import using a conversion service.

Signature:

```cpp
void import_convert (session_id s, string type, string username, string password, SR ref sr, (string -> string) map remote_config)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>type</td>
<td>Type of the conversion</td>
</tr>
<tr>
<td>string</td>
<td>username</td>
<td>Admin username on the host</td>
</tr>
<tr>
<td>string</td>
<td>password</td>
<td>Password on the host</td>
</tr>
<tr>
<td>SR ref</td>
<td>sr</td>
<td>The destination SR</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>remote_config</td>
<td>Remote configuration options</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: set_appliance

Overview:
Assign this VM to an appliance.

Signature:

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

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM to assign to an appliance.</td>
</tr>
<tr>
<td>VM appliance ref</td>
<td>value</td>
<td>The appliance to which this VM should be assigned.</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM</td>
</tr>
</tbody>
</table>

Return Type: (string -> string) map
map of service type to name

RPC name: call_plugin

Overview:
Call a XenAPI plugin on this vm.

Signature:

```python
string call_plugin (session_id s, VM ref vm, string plugin, string fn, (string -> string) map args)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>vm</td>
<td>The vm</td>
</tr>
<tr>
<td>string</td>
<td>plugin</td>
<td>The name of the plugin</td>
</tr>
<tr>
<td>string</td>
<td>fn</td>
<td>The name of the function within the plugin</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>args</td>
<td>Arguments for the function</td>
</tr>
</tbody>
</table>

Return Type: string
Result from the plugin
RPC name: set_has_vendor_device

Overview:
Controls whether, when the VM starts in HVM mode, its virtual hardware will include the emulated PCI device for which drivers may be available through Windows Update. Usually this should never be changed on a VM on which Windows has been installed: changing it on such a VM is likely to lead to a crash on next start.

Signature:

void set_has_vendor_device (session_id s, VM ref self, bool value)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>The VM on which to set this flag</td>
</tr>
<tr>
<td>bool</td>
<td>value</td>
<td>True to provide the vendor PCI device.</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: import

Overview:
Import an XVA from a URI.

Signature:

(VM ref set) import (session_id s, string url, SR ref sr, bool full_restore, bool force)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>url</td>
<td>The URL of the XVA file</td>
</tr>
<tr>
<td>SR ref</td>
<td>sr</td>
<td>The destination SR for the disks</td>
</tr>
<tr>
<td>bool</td>
<td>full_restore</td>
<td>Perform a full restore</td>
</tr>
<tr>
<td>bool</td>
<td>force</td>
<td>Force the import</td>
</tr>
</tbody>
</table>

Return Type: VM ref set
Imported VM reference

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:
2.12. CLASS: VM

CHAPTER 2. API REFERENCE

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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)
```
2.12. CLASS: VM

CHAPTER 2. API REFERENCE

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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)
2.12. CLASS: VM

CHAPTER 2. API REFERENCE

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>int</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>bool</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>host ref</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```c
int get_memory_dynamic_min (session_id s, VM ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: int
value of the field

RPC name: get_VCPUs_params

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

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```c
void set_VCPUs_params (session_id s, VM ref self, (string -> string) map value)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_VCPUs_max

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

Signature:
```c
int get_VCPUs_max (session_id s, VM ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>on_normal_exit</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>on_normal_exit</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: on_crashBehaviour
value of the field

RPC name: set_actions_after_crash

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

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>on_crashBehaviour</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_consoles

Overview:
Get the consoles field of the given VM.

Signature:

```c
(console ref set) get_consoles (session_id s, VM ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: console ref set
value of the field

RPC name: get_VIFs

Overview:
Get the VIFs field of the given VM.

Signature:

```c
(VIF ref set) get_VIFs (session_id s, VM ref self)
```
2.12. CLASS: VM

CHAPTER 2. API REFERENCE

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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)
2.12. CLASS: VM

CHAPTER 2. API REFERENCE

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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)
2.12. CLASS: VM

CHAPTER 2. API REFERENCE

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_PV_args

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

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string

value of the field

RPC name: set_PV_args

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

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_PV_bootloader_args

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

Signature:

```java
string get_PV_bootloader_args (session_id s, VM ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_PV_legacy_args

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

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

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

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_HVM_boot_policy

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

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string

text of the field

RPC name: set_HVM_boot_policy

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

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_HVM_boot_params

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

Signature:

```java
((string -> string) map) get_HVM_boot_params (session_id s, VM ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_HVM_shadow_multiplier

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

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
</tr>
</tbody>
</table>

Return Type: float

value of the field

RPC name: get_platform

Overview:
Get the platform field of the given VM.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
</tr>
</tbody>
</table>

Return Type: (string -> string) map

value of the field

RPC name: set_platform

Overview:
Set the platform field of the given VM.

Signature:

```c
void set_platform (session_id s, VM ref self, (string -> string) map value)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: set_PCI_bus

Overview: This message is deprecated Set the PCI bus field of the given VM.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_domid

Overview:
Get the domid field of the given VM.

Signature:

```plaintext
int get_domid (session_id s, VM ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: int
value of the field

RPC name: get_domarch

Overview:
Get the domarch field of the given VM.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```java
bool get_is_control_domain (session_id s, VM ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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)
### RPC name: get_recommendations

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

**Signature:**
```
string get_recommendations (session_id s, VM ref self)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string -&gt; string)map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

```c
bool get_ha_always_run (session_id s, VM ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```c
string get_ha_restart_priority (session_id s, VM ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```c
bool get_is_a_snapshot (session_id s, VM ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>VM ref</code></td>
<td><code>self</code></td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>VM ref</code></td>
<td><code>self</code></td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>VM ref</code></td>
<td><code>self</code></td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string set
value of the field

RPC name: set_tags

Overview:
Set the tags field of the given VM.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string set</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to add</td>
</tr>
</tbody>
</table>

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:

```java
void remove_tags (session_id s, VM ref self, string value)
```
### Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to remove</td>
</tr>
</tbody>
</table>

**Return Type:** `void`

### RPC name: `get_blocked_operations`

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

**Signature:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**Return Type:** `(vm_operations -> string) map`

### RPC name: `set_blocked_operations`

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

**Signature:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(vm_operations -&gt; string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

**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:**

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

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>vm_operations</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>vm_operations</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_snapshot_info

Overview:
Get the snapshot_info field of the given VM.

Signature:

```c
((string -> string) map) get_snapshot_info (session_id s, VM ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```c
string get_snapshot_metadata (session_id s, VM ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```plaintext
(VMPP ref) get_protection_policy (session_id s, VM ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```plaintext
bool get_is_snapshot_from_vmpp (session_id s, VM ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: bool
value of the field

RPC name: get_appliance
Overview:
Get the appliance field of the given VM.
Signature:

```plaintext
(VM_appliance ref) get_appliance (session_id s, VM ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>SR ref</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string

value of the field
RPC name: get\_hardware\_platform\_version

Overview:
Get the hardware\_platform\_version field of the given VM.

Signature:

```c
int get_hardware_platform_version (session_id s, VM ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: int
value of the field

RPC name: set\_hardware\_platform\_version

Overview:
Set the hardware\_platform\_version field of the given VM.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>int</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get\_has\_vendor\_device

Overview:
Get the has\_vendor\_device field of the given VM.

Signature:

```c
bool get_has_vendor_device (session_id s, VM ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: bool
value of the field
RPC name: create

Overview:
NOT RECOMMENDED! VM.clone or VM.copy (or VM.import) is a better choice in almost all situations. The
standard way to obtain a new VM is to call VM.clone on a template VM, then call VM.provision on the new
clone. Caution: if VM.create is used and then the new VM is attached to a virtual disc that has an operating
system already installed, then there is no guarantee that the operating system will boot and run. Any software
that calls VM.create on a future version of this API may fail or give unexpected results. For example this could
happen if an additional parameter were added to VM.create. VM.create is intended only for use in the automatic
creation of the system VM templates. It creates a new VM instance, and returns its handle.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM record</td>
<td>args</td>
<td>All constructor arguments</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

Return Type: VM ref
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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>label</td>
<td>label of object to return</td>
</tr>
</tbody>
</table>

Return Type: VM ref set
references to objects with matching names
2.13 Class: VM\_metrics

2.13.1 Fields for class: VM\_metrics

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO_run uuid</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>RO_run memory_actual</td>
<td>int</td>
<td>Guest’s actual memory (bytes)</td>
</tr>
<tr>
<td>RO_run VCPUs_number</td>
<td>int</td>
<td>Current number of VCPUs</td>
</tr>
<tr>
<td>RO_run VCPUs_utilisation</td>
<td>(int → float) map</td>
<td>Utilisation for all of guest’s current VCPUs</td>
</tr>
<tr>
<td>RO_run VCPUs_CPU</td>
<td>(int → int) map</td>
<td>VCPU to PCPU map</td>
</tr>
<tr>
<td>RO_run VCPUs_params</td>
<td>(string → string) map</td>
<td>The live equivalent to VM.VCPUs_params</td>
</tr>
<tr>
<td>RO_run VCPUs_flags</td>
<td>(int → string set) map</td>
<td>CPU flags (blocked,online,running)</td>
</tr>
<tr>
<td>RO_run state</td>
<td>string set</td>
<td>The state of the guest, eg blocked, dying etc</td>
</tr>
<tr>
<td>RO_run start_time</td>
<td>datetime</td>
<td>Time at which this VM was last booted</td>
</tr>
<tr>
<td>RO_run install_time</td>
<td>datetime</td>
<td>Time at which the VM was installed</td>
</tr>
<tr>
<td>RO_run last_updated</td>
<td>datetime</td>
<td>Time at which this information was last updated</td>
</tr>
<tr>
<td>RW other_config</td>
<td>(string → string) map</td>
<td>additional configuration</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: int
value of the field
RPC name: get_VCPUs_utilisation

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

Signature:

\[(\text{int} \to \text{float}) \text{ map}\] get_VCPUs_utilisation (session_id s, VM_metrics ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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

RPC name: get_VCPUs_CPU

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

Signature:

\[(\text{int} \to \text{int}) \text{ map}\] get_VCPUs_CPU (session_id s, VM_metrics ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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

RPC name: get_VCPUs_params

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

Signature:

\[(\text{string} \to \text{string}) \text{ map}\] get_VCPUs_params (session_id s, VM_metrics ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: (string → string) map
value of the field
RPC name: get_VCPUs_flags

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

Signature:

\[(\text{int} \rightarrow \text{string set}) \text{ map} \text{ get\_VCPUs\_flags (session\_id } s, \text{ VM\_metrics } \text{ ref } \text{ self})\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: \(\text{int} \rightarrow \text{string set}\) map
value of the field

RPC name: get_state

Overview:
Get the state field of the given VM_metrics.

Signature:

\[\text{string set} \text{ get\_state (session\_id } s, \text{ VM\_metrics } \text{ ref } \text{ self})\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: \text{string set}
value of the field

RPC name: get_start_time

Overview:
Get the start_time field of the given VM_metrics.

Signature:

\[\text{datetime get\_start\_time (session\_id } s, \text{ VM\_metrics } \text{ ref } \text{ self})\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: \text{datetime}
value of the field
RPC name: get_install_time

Overview:
Get the install_time field of the given VM_metrics.

Signature:

\[
\text{datetime get_install_time (session_id s, VM_metrics ref self)}
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: datetime
value of the field

RPC name: get_last_updated

Overview:
Get the last_updated field of the given VM_metrics.

Signature:

\[
\text{datetime get_last_updated (session_id s, VM_metrics ref self)}
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: datetime
value of the field

RPC name: get_other_config

Overview:
Get the other_config field of the given VM_metrics.

Signature:

\[
\text{((string \rightarrow string) map) get_other_config (session_id s, VM_metrics ref self)}
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: (string \rightarrow string) map
value of the field
RPC name: set other config

Overview:
Set the other config field of the given VM metrics.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

Return Type: void
RPC name: get_by_uuid

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

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_metrics record</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Quals</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO_run</td>
<td>uuid</td>
<td>string</td>
<td>RO</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>RO_run</td>
<td>os_version</td>
<td>(string → string) map</td>
<td>RO</td>
<td>version of the OS</td>
</tr>
<tr>
<td>RO_run</td>
<td>PV_drivers_version</td>
<td>(string → string) map</td>
<td>RO</td>
<td>version of the PV drivers</td>
</tr>
<tr>
<td>RO_run</td>
<td>PV_drivers_up_to_date</td>
<td>bool</td>
<td>RO</td>
<td>Logical AND of network_paths_optimized and storage_paths_optimized</td>
</tr>
<tr>
<td>RO_run</td>
<td>network_paths_optimized</td>
<td>bool</td>
<td>RO</td>
<td>True if the network paths are optimized with PV driver</td>
</tr>
<tr>
<td>RO_run</td>
<td>storage_paths_optimized</td>
<td>bool</td>
<td>RO</td>
<td>True if the storage paths are optimized with PV driver</td>
</tr>
<tr>
<td>RO_run</td>
<td>memory</td>
<td>(string → string) map</td>
<td>RO</td>
<td>This field exists but has no data. Use the memory and memory_internal_free RRD data-sources instead.</td>
</tr>
<tr>
<td>RO_run</td>
<td>disks</td>
<td>(string → string) map</td>
<td>RO</td>
<td>This field exists but has no data.</td>
</tr>
<tr>
<td>RO_run</td>
<td>networks</td>
<td>(string → string) map</td>
<td>RO</td>
<td>network configuration</td>
</tr>
<tr>
<td>RO_run</td>
<td>other</td>
<td>(string → string) map</td>
<td>RO</td>
<td>anything else</td>
</tr>
<tr>
<td>RO_run</td>
<td>last_updated</td>
<td>datetime</td>
<td>RO</td>
<td>Time at which this information was last updated</td>
</tr>
<tr>
<td>RW</td>
<td>other_config</td>
<td>(string → string) map</td>
<td>RW</td>
<td>additional configuration</td>
</tr>
<tr>
<td>RO_run</td>
<td>live</td>
<td>bool</td>
<td>RO</td>
<td>True if the guest is sending heartbeat messages via the guest agent</td>
</tr>
<tr>
<td>RO_run</td>
<td>can_use_hotplug_vbd</td>
<td>tristate_type</td>
<td>RO</td>
<td>The guest’s statement of whether it supports VBD hotplug, i.e. whether it is capable of responding immediately to instantiation of a new VBD by bringing online a new PV block device. If the guest states that it is not capable, then the VBD plug and unplug operations will not be allowed while the guest is running.</td>
</tr>
<tr>
<td>RO_run</td>
<td>can_use_hotplug_vif</td>
<td>tristate_type</td>
<td>RO</td>
<td>The guest’s statement of whether it supports VIF hotplug, i.e. whether it is capable of responding immediately to instantiation of a new VIF by bringing online a new PV network device. If the guest states that it is not capable, then the VIF plug and unplug operations will not be allowed while the guest is running.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_guest_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_guest_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:**

```plaintext
((string → string) map) get_PV_drivers_version (session_id s, VM_guest_metrics ref self)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_guest_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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

value of the field

### RPC name: `get_PV_drivers_up_to_date`

**Overview:** This message is deprecated
Get the `PV_drivers_up_to_date` field of the given `VM_guest_metrics`.

**Signature:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_guest_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**Return Type:** `bool`

value of the field

### RPC name: `get_network_paths_optimized`

**Overview:**
Get the `network_paths_optimized` field of the given `VM_guest_metrics`.

**Signature:**

```plaintext
bool get_network_paths_optimized (session_id s, VM_guest_metrics ref self)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_guest_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**Return Type:** `bool`

value of the field
RPC name: get_storage_paths_optimized

Overview:
Get the storage_paths_optimized field of the given VM_guest_metrics.

Signature:

bool get_storage_paths_optimized (session_id s, VM_guest_metrics ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_guest_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_guest_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_guest_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

\[(\text{string } \rightarrow \text{ string}) \text{ map} \text{ get_networks (session_id s, VM.guest.metrics ref self)}\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM.guest.metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: \((\text{string } \rightarrow \text{ string}) \text{ map}\)

value of the field

RPC name: get_other

Overview:
Get the other field of the given VM.guest.metrics.

Signature:

\[(\text{string } \rightarrow \text{ string}) \text{ map} \text{ get_other (session_id s, VM.guest.metrics ref self)}\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM.guest.metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: \((\text{string } \rightarrow \text{ string}) \text{ map}\)

value of the field

RPC name: get_last_updated

Overview:
Get the last_updated field of the given VM.guest.metrics.

Signature:

\text{datetime get_last_updated (session_id s, VM.guest.metrics ref self)}

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM.guest.metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: \text{datetime}

value of the field
### RPC name: get\_other\_config

**Overview:**
Get the other\_config field of the given VM\_guest\_metrics.

**Signature:**

```plaintext
((string \rightarrow string) map) get\_other\_config (session_id s, VM\_guest\_metrics ref self)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_guest_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**Return Type:**

```
(string \rightarrow string) map
```

value of the field

### RPC name: set\_other\_config

**Overview:**
Set the other\_config field of the given VM\_guest\_metrics.

**Signature:**

```plaintext
void set\_other\_config (session_id s, VM\_guest\_metrics ref self, (string \rightarrow string) map value)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_guest_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string \rightarrow string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

**Return Type:**

```plaintext
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:**

```plaintext
void add\_to\_other\_config (session_id s, VM\_guest\_metrics ref self, string key, string value)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_guest_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

**Return Type:**

```plaintext
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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_guest_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_guest_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: bool

value of the field

RPC name: get_can_use_hotplug_vbd

Overview:
Get the can_use_hotplug_vbd field of the given VM_guest_metrics.

Signature:

(tristate_type) get_can_use_hotplug_vbd (session_id s, VM_guest_metrics ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_guest_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: tristate_type

value of the field
RPC name: get_can_use_hotplug_vif

Overview:
Get the can_use_hotplug_vif field of the given VM_guest_metrics.

Signature:
(tristate_type) get_can_use_hotplug_vif (session_id s, VM_guest_metrics ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_guest_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: tristate_type
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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_guest_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: VM_guest_metrics record
all fields from the object
2.15 Class: VMPP

2.15.1 Fields for class: VMPP

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP</td>
<td>VM Protection Policy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>RO</strong> run</td>
<td>uuid</td>
<td>RO</td>
<td>run</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td><strong>RW</strong> name</td>
<td>label</td>
<td>RW</td>
<td>name</td>
<td>string</td>
<td>a human-readable name</td>
</tr>
<tr>
<td><strong>RW</strong> name</td>
<td>description</td>
<td>RW</td>
<td>description</td>
<td>string</td>
<td>a notes field containing human-readable description</td>
</tr>
<tr>
<td><strong>RW</strong> is_policy_enabled</td>
<td></td>
<td>RW</td>
<td>is_policy_enabled</td>
<td>bool</td>
<td>enable or disable this policy</td>
</tr>
<tr>
<td><strong>RO</strong> ins</td>
<td>backup_retention_value</td>
<td>RO</td>
<td>backup_retention_value</td>
<td>int</td>
<td>maximum number of backups that should be stored at any time</td>
</tr>
<tr>
<td><strong>RO</strong> ins</td>
<td>backup_frequency</td>
<td>RO</td>
<td>backup_frequency</td>
<td>vmpp_backup_frequency</td>
<td>frequency of the backup schedule</td>
</tr>
<tr>
<td><strong>RO</strong> ins</td>
<td>backup_schedule</td>
<td>RO</td>
<td>backup_schedule</td>
<td>(string → string) map</td>
<td>schedule of the backup containing 'hour', 'min', 'days'. Date/time-related information is in Local Timezone</td>
</tr>
<tr>
<td><strong>RO</strong> run</td>
<td>is_backup_running</td>
<td>RO</td>
<td>is_backup_running</td>
<td>bool</td>
<td>true if this protection policy’s backup is running</td>
</tr>
<tr>
<td><strong>RO</strong> run</td>
<td>backup_last_run_time</td>
<td>RO</td>
<td>backup_last_run_time</td>
<td>datetime</td>
<td>time of the last backup</td>
</tr>
<tr>
<td><strong>RO</strong> ins</td>
<td>archive_target_type</td>
<td>RO</td>
<td>archive_target_type</td>
<td>vmpp_archive_target_type</td>
<td>type of the archive target config</td>
</tr>
<tr>
<td><strong>RO</strong> ins</td>
<td>archive_target_config</td>
<td>RO</td>
<td>archive_target_config</td>
<td>(string → string) map</td>
<td>configuration for the archive, including its 'location', 'username', 'password'</td>
</tr>
<tr>
<td><strong>RO</strong> ins</td>
<td>archive_frequency</td>
<td>RO</td>
<td>archive_frequency</td>
<td>vmpp_archive_frequency</td>
<td>frequency of the archive schedule</td>
</tr>
<tr>
<td><strong>RO</strong> ins</td>
<td>archive_schedule</td>
<td>RO</td>
<td>archive_schedule</td>
<td>(string → string) map</td>
<td>schedule of the archive containing 'hour', 'min', 'days'. Date/time-related information is in Local Timezone</td>
</tr>
<tr>
<td><strong>RO</strong> run</td>
<td>is_archive_running</td>
<td>RO</td>
<td>is_archive_running</td>
<td>bool</td>
<td>true if this protection policy’s archive is running</td>
</tr>
<tr>
<td><strong>RO</strong> run</td>
<td>archive_last_run_time</td>
<td>RO</td>
<td>archive_last_run_time</td>
<td>datetime</td>
<td>time of the last archive</td>
</tr>
<tr>
<td><strong>RO</strong> run</td>
<td>VMs</td>
<td>RO</td>
<td>VMs</td>
<td>VM ref</td>
<td>all VMs attached to this protection policy</td>
</tr>
<tr>
<td><strong>RO</strong> ins</td>
<td>is_alarm_enabled</td>
<td>RO</td>
<td>is_alarm_enabled</td>
<td>bool</td>
<td>true if alarm is enabled for this policy</td>
</tr>
<tr>
<td><strong>RO</strong> ins</td>
<td>alarm_config</td>
<td>RO</td>
<td>alarm_config</td>
<td>(string → string) map</td>
<td>configuration for the alarm</td>
</tr>
<tr>
<td><strong>RO</strong> run</td>
<td>recent_alerts</td>
<td>RO</td>
<td>recent_alerts</td>
<td>string set</td>
<td>recent alerts</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>vmpp</td>
<td>The protection policy to execute</td>
</tr>
</tbody>
</table>

Return Type: string
RPC name: archive_now

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

Signature:
```java
string archive_now (session_id s, VM ref snapshot)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>snapshot</td>
<td>The snapshot to archive</td>
</tr>
</tbody>
</table>

Return Type: string
An XMLRPC result

RPC name: get_alerts

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

Signature:
```java
(string set) get_alerts (session_id s, VMPP ref vmpp, int hours_from_now)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>vmpp</td>
<td>The protection policy</td>
</tr>
<tr>
<td>int</td>
<td>hours_from_now</td>
<td>how many hours in the past the oldest record to fetch is</td>
</tr>
</tbody>
</table>

Return Type: string set
A list of alerts encoded in xml

RPC name: set_backup_retention_value

Overview:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>The protection policy</td>
</tr>
<tr>
<td>int</td>
<td>value</td>
<td>the value to set</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>The protection policy</td>
</tr>
<tr>
<td>vmpp_backup_frequency</td>
<td>value</td>
<td>the backup frequency</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>The protection policy</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>the value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>The protection policy</td>
</tr>
<tr>
<td>vmpp_archive_frequency</td>
<td>value</td>
<td>the archive frequency</td>
</tr>
</tbody>
</table>

Return Type: void
2.15. CLASS: VMPP

RPC name: set_archive_schedule

Overview:

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>The protection policy</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>the value to set</td>
</tr>
</tbody>
</table>

Return Type: void

---

RPC name: set_archive_target_type

Overview:
Set the value of the archive_target_config_type field.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>The protection policy</td>
</tr>
<tr>
<td>vmpp_archive_target_type</td>
<td>value</td>
<td>the archive target config type</td>
</tr>
</tbody>
</table>

Return Type: void

---

RPC name: set_archive_target_config

Overview:

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>The protection policy</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>the value to set</td>
</tr>
</tbody>
</table>

Return Type: void
RPC name: set_is_alarm_enabled

Overview:
Set the value of the is_alarm_enabled field.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>The protection policy</td>
</tr>
<tr>
<td>bool</td>
<td>value</td>
<td>true if alarm is enabled for this policy</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: set_alarm_config

Overview:

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>The protection policy</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>value</td>
<td>the value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: add_to_backup_schedule

Overview:

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>The protection policy</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>the key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>the value to add</td>
</tr>
</tbody>
</table>

Return Type: void
RPC name: `add_to_archive_target_config`

Overview:

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>The protection policy</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>the key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>the value to add</td>
</tr>
</tbody>
</table>

Return Type: `void`

RPC name: `add_to_archive_schedule`

Overview:

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>The protection policy</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>the key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>the value to add</td>
</tr>
</tbody>
</table>

Return Type: `void`

RPC name: `add_to_alarm_config`

Overview:

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>The protection policy</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>the key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>the value to add</td>
</tr>
</tbody>
</table>

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:
<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>The protection policy</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>the key to remove</td>
</tr>
</tbody>
</table>
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:
<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>The protection policy</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>the key to remove</td>
</tr>
</tbody>
</table>
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:
<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>The protection policy</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>the key to remove</td>
</tr>
</tbody>
</table>
Return Type: void
RPC name: remove_from_alarm_config

Overview:

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>The protection policy</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>the key to remove</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: set_backup_last_run_time

Overview:

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>The protection policy</td>
</tr>
<tr>
<td>datetime</td>
<td>value</td>
<td>the value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: set_archive_last_run_time

Overview:

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>The protection policy</td>
</tr>
<tr>
<td>datetime</td>
<td>value</td>
<td>the value to set</td>
</tr>
</tbody>
</table>

Return Type: void
RPC name: get_all
Overview:
Return a list of all the VMPPs known to the system.
Signature:

\[(\text{VMPP ref set}) \text{ get}_\text{all} (\text{session_id } s)\]

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:

\[((\text{VMPP ref } \rightarrow \text{VMPP record}) \text{ map}) \text{ get}_\text{all}_\text{records} (\text{session_id } s)\]

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:

\[\text{string get}_\text{uuid} (\text{session_id } s, \text{VMPP ref } \text{self})\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get_name_label
Overview:
Get the name/label field of the given VMPP.
Signature:

\[\text{string get}_\text{name}_\text{label} (\text{session_id } s, \text{VMPP ref } \text{self})\]
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_is_policy_enabled

Overview:
Get the is_policy_enabled field of the given VMPP.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>bool</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_backup_type

Overview:
Get the backup_type field of the given VMPP.

Signature:

```c
(vmpp_backup_type) get_backup_type (session_id s, VMPP ref self)
```
2.15. CLASS: VMPP

CHAPTER 2. API REFERENCE

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td><code>vmpp_backup_type</code></td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

Return Type: `void`

RPC name: `get_backup_retention_value`

Overview:
Get the backup_retention_value field of the given VMPP.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: `int`
value of the field

RPC name: `get_backup_frequency`

Overview:
Get the backup_frequency field of the given VMPP.

Signature:

```c
(vmpp_backup_frequency) get_backup_frequency (session_id s, VMPP ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: datetime
value of the field

RPC name: get_VMs
Overview:
Get the VMs field of the given VMPP.
Signature:

```cpp
(VM ref set) get_VMs (session_id s, VMPP ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: bool
value of the field

RPC name: get_alarm_config
Overview:
Get the alarm_config field of the given VMPP.
Signature:

```cpp
((string -> string) map) get_alarm_config (session_id s, VMPP ref self)
```
2.15. CLASS: VMPP

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: \((string \rightarrow string)\) map
value of the field

RPC name: get_recent_alerts

Overview:
Get the recent_alerts field of the given VMPP.

Signature:

\((string \text{ set}) \text{ get_recent_alerts (session_id } s, \text{ VMPP ref self)\)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: \(string \text{ set}\)
value of the field

RPC name: create

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

Signature:

\((\text{VMPP ref}) \text{ create (session_id } s, \text{ VMPP record args)\)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP record</td>
<td>args</td>
<td>All constructor arguments</td>
</tr>
</tbody>
</table>

Return Type: \(\text{VMPP ref}\)
reference to the newly created object

RPC name: destroy

Overview:
Destroy the specified VMPP instance.

Signature:

\(\text{void destroy (session_id } s, \text{ VMPP ref self)\)
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPP ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>label</td>
<td>label of object to return</td>
</tr>
</tbody>
</table>

Return Type: VMPP ref set
references to objects with matching names
2.16 Class: VM_appliance

2.16.1 Fields for class: VM_appliance

<table>
<thead>
<tr>
<th>Name</th>
<th>VM_appliance</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>VM appliance</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO</td>
<td>uuid</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>RW</td>
<td>name, label</td>
<td>string</td>
<td>a human-readable name</td>
</tr>
<tr>
<td>RW</td>
<td>name, description</td>
<td>string</td>
<td>a notes field containing human-readable description</td>
</tr>
<tr>
<td>RO</td>
<td>allowed_operations</td>
<td>vm_appliance_operation set</td>
<td>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.</td>
</tr>
<tr>
<td>RO</td>
<td>current_operations</td>
<td>(string → vm_appliance_operation) map</td>
<td>links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.</td>
</tr>
<tr>
<td>RO</td>
<td>VMs</td>
<td>VM ref set</td>
<td>all VMs in this appliance</td>
</tr>
</tbody>
</table>

2.16.2 RPCs associated with class: VM_appliance

RPC name: start

Overview:
Start all VMs in the appliance.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_appliance ref</td>
<td>self</td>
<td>The VM appliance</td>
</tr>
<tr>
<td>bool</td>
<td>paused</td>
<td>Instantiate all VMs belonging to this appliance in paused state if set to true.</td>
</tr>
</tbody>
</table>

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:

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

Arguments:
2.16. CLASS: VM\_APPLIANCE

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_appliance ref</td>
<td>self</td>
<td>The VM appliance</td>
</tr>
</tbody>
</table>

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:**

```c
void hard_shutdown (session_id s, VM\_appliance ref self)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_appliance ref</td>
<td>self</td>
<td>The VM appliance</td>
</tr>
</tbody>
</table>

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:**

```c
void shutdown (session_id s, VM\_appliance ref self)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_appliance ref</td>
<td>self</td>
<td>The VM appliance</td>
</tr>
</tbody>
</table>

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:**

```c
void assert\_can\_be\_recovered (session_id s, VM\_appliance ref self, session ref session_to)
```
2.16. CLASS: VM\_APPLIANCE

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_appliance ref</td>
<td>self</td>
<td>The VM appliance to recover.</td>
</tr>
<tr>
<td>session ref</td>
<td>session_to</td>
<td>The session to which the VM appliance is to be recovered.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_appliance ref</td>
<td>self</td>
<td>The VM appliance for which the required list of SRs has to be recovered.</td>
</tr>
<tr>
<td>session ref</td>
<td>session_to</td>
<td>The session to which the list of SRs have to be recovered.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_appliance ref</td>
<td>self</td>
<td>The VM appliance to recover.</td>
</tr>
<tr>
<td>session ref</td>
<td>session_to</td>
<td>The session to which the VM appliance is to be recovered.</td>
</tr>
<tr>
<td>bool</td>
<td>force</td>
<td>Whether the VMs should replace newer versions of themselves.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_appliance ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_appliance ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_appliance ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_appliance ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_appliance ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_appliance ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_appliance ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_appliance ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_appliance record</td>
<td>args</td>
<td>All constructor arguments</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_appliance ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM_appliance ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>label</td>
<td>label of object to return</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR_task</td>
<td>DR task.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO_run</td>
<td>uuid</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>RO_run</td>
<td>introduced_SR</td>
<td>SR ref set</td>
<td>All SRs introduced by this appliance</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>type</td>
<td>The SR driver type of the SRs to introduce</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>device_config</td>
<td>The device configuration of the SRs to introduce</td>
</tr>
<tr>
<td>string set</td>
<td>whitelist</td>
<td>The devices to use for disaster recovery</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DR_task ref</code></td>
<td>self</td>
<td>The disaster recovery task to destroy</td>
</tr>
</tbody>
</table>

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 -> 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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR_task ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR_task ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR_task ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: DR_task record
all fields from the object
## 2.18 Class: host

### 2.18.1 Fields for class: host

<table>
<thead>
<tr>
<th>Name</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RO</strong>run uuid</td>
<td>uuid</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td><strong>RW</strong>run name_label</td>
<td>name_label</td>
<td>string</td>
<td>A human-readable name</td>
</tr>
<tr>
<td><strong>RW</strong>run name_description</td>
<td>name_description</td>
<td>string</td>
<td>A notes field containing host description</td>
</tr>
<tr>
<td><strong>RO</strong>run memory_overhead</td>
<td>memory_overhead</td>
<td>int</td>
<td>Virtualization memory overhead</td>
</tr>
<tr>
<td><strong>RO</strong>run allowed_operations</td>
<td>allowed_operations</td>
<td>host_allowed_operations set</td>
<td>This list is advisory only. The state may have changed before this field is read by a client.</td>
</tr>
<tr>
<td><strong>RO</strong>run current_operations</td>
<td>current_operations</td>
<td>(string → host_allowed_operations) map</td>
<td>links each of the running operations for this object (by reference)</td>
</tr>
<tr>
<td><strong>RO</strong>run API_version_major</td>
<td>API_version_major</td>
<td>int</td>
<td>Major version number</td>
</tr>
<tr>
<td><strong>RO</strong>run API_version_minor</td>
<td>API_version_minor</td>
<td>int</td>
<td>Minor version number</td>
</tr>
<tr>
<td><strong>RO</strong>run API_version_vendor</td>
<td>API_version_vendor</td>
<td>string</td>
<td>Identification of vendor</td>
</tr>
<tr>
<td><strong>RO</strong>run API_version_vendor_implementation</td>
<td>API_version_vendor_implementation</td>
<td>(string → string) map</td>
<td>Details of vendor implementation</td>
</tr>
<tr>
<td><strong>RO</strong>run enabled</td>
<td>enabled</td>
<td>bool</td>
<td>True if the host is currently active</td>
</tr>
<tr>
<td><strong>RO</strong>run software_version</td>
<td>software_version</td>
<td>(string → string) map</td>
<td>Additional configuration</td>
</tr>
<tr>
<td><strong>RW</strong>run other_config</td>
<td>other_config</td>
<td>(string → string) map</td>
<td>Xen capabilities</td>
</tr>
<tr>
<td><strong>RO</strong>run capabilities</td>
<td>capabilities</td>
<td>string set</td>
<td>The CPU configuration</td>
</tr>
<tr>
<td><strong>RO</strong>run cpu_configuration</td>
<td>cpu_configuration</td>
<td>(string → string) map</td>
<td>May contain keys such as &quot;sockets_per_node&quot;, &quot;cores_per_socket&quot;, &quot;threads_per_core&quot;</td>
</tr>
<tr>
<td><strong>RO</strong>run sched_policy</td>
<td>sched_policy</td>
<td>string</td>
<td>Scheduler policy currently running on the system.</td>
</tr>
<tr>
<td><strong>RO</strong>run supported_bootloaders</td>
<td>supported_bootloaders</td>
<td>string set</td>
<td>a list of the bootloaders installed on this machine</td>
</tr>
<tr>
<td><strong>RO</strong>run resident_VMs</td>
<td>resident_VMs</td>
<td>VM ref set</td>
<td>list of VMs currently resident</td>
</tr>
<tr>
<td><strong>RW</strong>run logging</td>
<td>logging</td>
<td>(string → string) map</td>
<td>Logging configuration</td>
</tr>
<tr>
<td><strong>RO</strong>run PIFs</td>
<td>PIFs</td>
<td>PIF ref set</td>
<td>Physical network interfaces</td>
</tr>
<tr>
<td><strong>RW</strong>run suspend_image_sr</td>
<td>suspend_image_sr</td>
<td>SR ref</td>
<td>The SR in which VDIs for suspended VMs are created</td>
</tr>
<tr>
<td><strong>RW</strong>run crash_dump_sr</td>
<td>crash_dump_sr</td>
<td>SR ref</td>
<td>The SR in which VDIs for crash dumps are created</td>
</tr>
<tr>
<td><strong>RO</strong>run crashdumps</td>
<td>crashdumps</td>
<td>host_crashdump ref set</td>
<td>Set of host crash dumps</td>
</tr>
<tr>
<td><strong>RO</strong>run patches</td>
<td>patches</td>
<td>host_patch ref set</td>
<td>Set of host patches</td>
</tr>
<tr>
<td><strong>RO</strong>run PBDs</td>
<td>PBDs</td>
<td>PBD ref set</td>
<td>Physical block devices</td>
</tr>
<tr>
<td><strong>RO</strong>run host_CPUs</td>
<td>host_CPUs</td>
<td>host_cpu ref set</td>
<td>The physical CPUs on this host</td>
</tr>
<tr>
<td><strong>RO</strong>run cpu_info</td>
<td>cpu_info</td>
<td>(string → string) map</td>
<td>Details about the physical CPUs on this host</td>
</tr>
<tr>
<td><strong>RW</strong>run hostname</td>
<td>hostname</td>
<td>string</td>
<td>The hostname of this host</td>
</tr>
<tr>
<td>RW</td>
<td>address</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>RO run</td>
<td>metrics</td>
<td>host_metrics ref</td>
<td></td>
</tr>
<tr>
<td>RO run</td>
<td>license_params</td>
<td>(string → string) map</td>
<td></td>
</tr>
<tr>
<td>RO run</td>
<td>ha_statefiles</td>
<td>string set</td>
<td></td>
</tr>
<tr>
<td>RO run</td>
<td>ha_network_peers</td>
<td>string set</td>
<td></td>
</tr>
<tr>
<td>RO run</td>
<td>blobs</td>
<td>(string → blob ref) map</td>
<td></td>
</tr>
<tr>
<td>RW</td>
<td>tags</td>
<td>string set</td>
<td></td>
</tr>
<tr>
<td>RO run</td>
<td>external_auth_type</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>RO run</td>
<td>external_auth_service_name</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>RO run</td>
<td>external_auth_configuration</td>
<td>(string → string) map</td>
<td></td>
</tr>
<tr>
<td>RO run</td>
<td>edition</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>RW</td>
<td>license_server</td>
<td>(string → string) map</td>
<td></td>
</tr>
<tr>
<td>RO run</td>
<td>bios_strings</td>
<td>(string → string) map</td>
<td></td>
</tr>
<tr>
<td>RO run</td>
<td>power_on_mode</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>RO run</td>
<td>power_on_config</td>
<td>(string → string) map</td>
<td></td>
</tr>
<tr>
<td>RO ins</td>
<td>local_cache_sr</td>
<td>SR ref</td>
<td></td>
</tr>
<tr>
<td>RO run</td>
<td>chipset_info</td>
<td>(string → string) map</td>
<td></td>
</tr>
<tr>
<td>RO run</td>
<td>PCs</td>
<td>PCI ref set</td>
<td></td>
</tr>
<tr>
<td>RO run</td>
<td>PGPUs</td>
<td>PGPU ref set</td>
<td></td>
</tr>
<tr>
<td>RO run</td>
<td>ssl_legacy</td>
<td>bool</td>
<td></td>
</tr>
<tr>
<td>RW</td>
<td>guest_VCPUs_params</td>
<td>(string → string) map</td>
<td></td>
</tr>
<tr>
<td>RW</td>
<td>display</td>
<td>host_display</td>
<td></td>
</tr>
<tr>
<td>RO run</td>
<td>virtual_hardware_platform_versions</td>
<td>int set</td>
<td></td>
</tr>
</tbody>
</table>

The address by which this host is contacted from any other host. Host metrics associated with this host. State of the current license for this host. The set of statefiles accessible from this host. The set of hosts visible with ha\_statefiles from this host. Binary blobs associated with this host. User-specified tags for categorization purposes. Type of external authentication configured; empty if none configured. Name of external authentication service configured; empty if none configured. Configuration specific to external authentication service. Product edition. Contact information of the license server. BIOS strings. The power on mode. The power on config. The SR that is used as a local cache. Information about chipset. List of PCI devices in the host. List of physical GPUs in the host. Allow SSLv3 protocol and use by older XenServers, both incoming and outgoing. When this is set to a different value, the host immediately restarts its XenServer service; typically this takes a second but existing connections may be broken. XenAPI login sessions remain valid. VCPUs params to apply to its guests. Indicates whether the host should output its console to a physical device. The set of versions of the virtual hardware platform that the host supports and its guests support.

### 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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The Host to disable</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: enable

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

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The Host to enable</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The Host to shutdown</td>
</tr>
</tbody>
</table>

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:

```c
void reboot (session_id s, host ref host)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The Host to reboot</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: dmesg

Overview:
Get the host xen dmesg.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The Host to query</td>
</tr>
</tbody>
</table>

Return Type: string
dmesg string

RPC name: dmesg_clear

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

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The Host to query</td>
</tr>
</tbody>
</table>

Return Type: string
dmesg string

RPC name: get_log

Overview:
Get the host’s log file.

Signature:

```c
string get_log (session_id s, host ref host)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The Host to query</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host</td>
</tr>
<tr>
<td>string</td>
<td>keys</td>
<td>The keys to send</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: bugreport_upload

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

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host on which to run xen-bugtool</td>
</tr>
<tr>
<td>string</td>
<td>url</td>
<td>The URL to upload to</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>options</td>
<td>Extra configuration operations</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: list_methods

Overview:
List all supported methods.

Signature:

```c
(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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host to upload the license to</td>
</tr>
<tr>
<td>string</td>
<td>contents</td>
<td>The contents of the license file, base64 encoded</td>
</tr>
</tbody>
</table>

Return Type: void

Possible Error Codes: LICENSE_PROCESSING_ERROR

RPC name: license_add

Overview:
Apply a new license to a host.

Signature:

```c
void license_add (session_id s, host ref host, string contents)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host to upload the license to</td>
</tr>
<tr>
<td>string</td>
<td>contents</td>
<td>The contents of the license file, base64 encoded</td>
</tr>
</tbody>
</table>

Return Type: void

Possible Error Codes: LICENSE_PROCESSING_ERROR

RPC name: license_remove

Overview:
Remove any license file from the specified host, and switch that host to the unlicensed edition.

Signature:

```c
void license_remove (session_id s, host ref host)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host from which any license will be removed</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: destroy

Overview:
Destroy specified host record in database.

Signature:

void destroy (session_id s, host ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>The host record to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The Host to power on</td>
</tr>
</tbody>
</table>

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:

\[(\text{data\_source record set}) \ get\_\text{data\_sources} (\text{session\_id s, host ref host})\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host to interrogate</td>
</tr>
</tbody>
</table>

Return Type: data\_source record set

A set of data sources

RPC name: record\_data\_source

Overview:
Start recording the specified data source.

Signature:

\[\text{void record\_data\_source} (\text{session\_id s, host ref host, string data\_source})\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host</td>
</tr>
<tr>
<td>string</td>
<td>data_source</td>
<td>The data source to record</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: query\_data\_source

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

Signature:

\[\text{float query\_data\_source} (\text{session\_id s, host ref host, string data\_source})\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host</td>
</tr>
<tr>
<td>string</td>
<td>data_source</td>
<td>The data source to query</td>
</tr>
</tbody>
</table>

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:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host</td>
</tr>
<tr>
<td>string</td>
<td>data_source</td>
<td>The data source whose archives are to be forgotten</td>
</tr>
</tbody>
</table>

**Return Type:** void

---

**RPC name: assert_can_evacuate**

**Overview:**
Check this host can be evacuated.

**Signature:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host</td>
</tr>
</tbody>
</table>

**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:**

```c
((VM ref -> string set) map) get_vms_which_prevent_evacuation (session_id s, host ref self)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>The host to query</td>
</tr>
</tbody>
</table>

**Return Type:** (VM ref -> 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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>The host to query</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host to evacuate</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: syslog_reconfigure

Overview:
Re-configure syslog logging.

Signature:

void syslog_reconfigure (session_id s, host ref host)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>Tell the host to reread its Host.logging parameters and reconfigure itself accordingly</td>
</tr>
</tbody>
</table>

Return Type: void
RPC name: management_reconfigure

Overview:
Reconfigure the management network interface.

Signature:
void management_reconfigure (session_id s, PIF ref pif)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>pif</td>
<td>reference to a PIF object corresponding to the management interface</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>interface</td>
<td>name of the interface to use as a management interface</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>Which host’s management interface is required</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host to interrogate</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The Host on which you want to restart the agent</td>
</tr>
</tbody>
</table>

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:
```
<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host whose host name to set</td>
</tr>
<tr>
<td>string</td>
<td>hostname</td>
<td>The new host name</td>
</tr>
</tbody>
</table>
```

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:
```
<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host to send the request to</td>
</tr>
</tbody>
</table>
```

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host for which to compute the memory overhead</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host to whom the data should be sent</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: backup_rrds

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

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>Schedule a backup of the RRDs of this host</td>
</tr>
<tr>
<td>float</td>
<td>delay</td>
<td>Delay in seconds from when the call is received to perform the backup</td>
</tr>
</tbody>
</table>

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:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host</td>
</tr>
<tr>
<td>string</td>
<td>name</td>
<td>The name associated with the blob</td>
</tr>
<tr>
<td>string</td>
<td>mime_type</td>
<td>The mime type for the data. Empty string translates to application/octet-stream</td>
</tr>
<tr>
<td>bool</td>
<td>public</td>
<td>True if the blob should be publicly available</td>
</tr>
</tbody>
</table>

**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:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host</td>
</tr>
<tr>
<td>string</td>
<td>plugin</td>
<td>The name of the plugin</td>
</tr>
<tr>
<td>string</td>
<td>fn</td>
<td>The name of the function within the plugin</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>args</td>
<td>Arguments for the function</td>
</tr>
</tbody>
</table>

**Return Type:** string

Result from the plugin

---

**RPC name: get_servertime**

**Overview:**
This call queries the host’s clock for the current time.

**Signature:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host whose clock should be queried</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host whose clock should be queried</td>
</tr>
</tbody>
</table>

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_name, string auth_type)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host whose external authentication should be enabled</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>config</td>
<td>A list of key-values containing the configuration data</td>
</tr>
<tr>
<td>string</td>
<td>service_name</td>
<td>The name of the service</td>
</tr>
<tr>
<td>string</td>
<td>auth_type</td>
<td>The type of authentication (e.g. AD for Active Directory)</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host whose external authentication should be disabled</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>config</td>
<td>Optional parameters as a list of key-values containing the configuration data</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>The host to query</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host</td>
</tr>
<tr>
<td>string</td>
<td>edition</td>
<td>The requested edition</td>
</tr>
<tr>
<td>bool</td>
<td>force</td>
<td>Update the license params even if the apply call fails</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The Host to modify</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>The host</td>
</tr>
<tr>
<td>string</td>
<td>power_on_mode</td>
<td>power-on-mode can be empty,iLO,wake-on-lan, DRAC or other</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>power_on_config</td>
<td>Power on config</td>
</tr>
</tbody>
</table>
RPC name: set_cpu_features

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

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host</td>
</tr>
<tr>
<td>string</td>
<td>features</td>
<td>The features string (32 hexadecimal digits)</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: enable_local_storage_caching

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

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host</td>
</tr>
<tr>
<td>SR ref</td>
<td>sr</td>
<td>The SR to use as a local cache</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host</td>
</tr>
</tbody>
</table>

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 -> string) map)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The target host</td>
</tr>
<tr>
<td>network ref</td>
<td>network</td>
<td>The network through which migration traffic should be received.</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>options</td>
<td>Extra configuration operations</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The Host to declare is dead</td>
</tr>
</tbody>
</table>

Return Type: void
RPC name: enable_display

Overview:
Enable console output to the physical display device next time this host boots.

Signature:
(host_display) enable_display (session_id s, host ref host)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host</td>
</tr>
</tbody>
</table>

Return Type: host_display
This host’s physical display usage

RPC name: disable_display

Overview:
Disable console output to the physical display device next time this host boots.

Signature:
(host_display) disable_display (session_id s, host ref host)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host</td>
</tr>
</tbody>
</table>

Return Type: host_display
This host’s physical display usage

RPC name: set_ssl_legacy

Overview:
Enable/disable SSLv3 for interoperability with older versions of XenServer. When this is set to a different value, the host immediately restarts its SSL/TLS listening service; typically this takes less than a second but existing connections to it will be broken. XenAPI login sessions will remain valid.

Signature:
void set_ssl_legacy (session_id s, host ref self, bool value)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>The host</td>
</tr>
<tr>
<td>bool</td>
<td>value</td>
<td>True to allow SSLv3 and ciphersuites as used in old XenServer versions</td>
</tr>
</tbody>
</table>

Return Type: void
**RPC name: get_all**

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

**Signature:**

\[(\text{host ref set}) \text{ get\_all (session\_id \ s)}\]

**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:**

\[((\text{host ref } \rightarrow \text{host record}) \text{ map}) \text{ get\_all\_records (session\_id \ s)}\]

**Return Type:** (host ref \rightarrow host record) map
records of all objects

---

**RPC name: get_uuid**

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

**Signature:**

\[\text{string get\_uuid (session\_id \ s, host ref \ self)}\]

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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

---

**RPC name: get_name_label**

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

**Signature:**

\[\text{string get\_name\_label (session\_id \ s, host ref \ self)}\]
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```c
string get_API_version_vendor (session_id s, host ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_capabilities

Overview:
Get the capabilities field of the given host.

Signature:

```plaintext
(string set) get_capabilities (session_id s, host ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to object</td>
</tr>
</tbody>
</table>

Return Type: string set
value of the field

RPC name: get_cpu_configuration

Overview:
Get the cpu_configuration field of the given host.

Signature:

```plaintext
((string -> string) map) get_cpu_configuration (session_id s, host ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: VM ref set
value of the field

RPC name: get_logging

Overview:
Get the logging field of the given host.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: (string -> string) map
value of the field

RPC name: set_logging

Overview:
Set the logging field of the given host.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: add_to_logging

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

Signature:

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

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>SR ref</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_crash_dump_sr

Overview:
Get the crash_dump_sr field of the given host.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

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

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>SR ref</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_metrics

Overview:
Get the metrics field of the given host.

Signature:

\[(\text{host}\_\text{metrics} \text{ ref}) \text{ get}\_\text{metrics} (\text{session}\_\text{id} s, \text{host}\ \text{ref} \text{ self})\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

\[((\text{string} \to \text{string}) \text{ map}) \text{ get}\_\text{license}\_\text{params} (\text{session}\_\text{id} s, \text{host}\ \text{ref} \text{ self})\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: (string \to string) map
value of the field

RPC name: get_ha_statefiles

Overview:
Get the ha_statefiles field of the given host.

Signature:

\[(\text{string}\ \text{set}) \text{ get}\_\text{ha}\_\text{statefiles} (\text{session}\_\text{id} s, \text{host}\ \text{ref} \text{ self})\]
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string set
value of the field

RPC name: get_blobs

Overview:
Get the blobs field of the given host.

Signature:

\[((string\ \rightarrow\ blob\ ref)\ map)\ get\_blobs\ (session\_id\ s,\ host\ ref\ self)\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: (string \rightarrow 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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to add</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to remove</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_external_auth_type

Overview:
Get the external auth type field of the given host.

Signature:

```java
string get_external_auth_type (session_id s, host ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```java
string get_external_auth_service_name (session_id s, host ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```java
((string -> string) map) get_external_auth_configuration (session_id s, host ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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

RPC name: get.edition

Overview:
Get the edition field of the given host.

Signature:

```java
string get_edition (session_id s, host ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get.license.server

Overview:
Get the license.server field of the given host.

Signature:

```java
((string -> string) map) get_license_server (session_id s, host ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```java
void set_license_server (session_id s, host ref self, (string -> string) map value)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string -&gt; string)</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: PGPU ref set
value of the field

RPC name: get_ssl_legacy

Overview:
Get the ssl_legacy field of the given host.

Signature:

bool get_ssl_legacy (session_id s, host ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: bool
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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_display

Overview:
Get the display field of the given host.

Signature:
(host_display) get_display (session_id s, host ref self)
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: host.display
value of the field

RPC name: set_display

Overview:
Set the display field of the given host.

Signature:

void set_display (session_id s, host ref self, host_display value)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>host_display</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_virtual_hardware_platform_versions

Overview:
Get the virtual_hardware_platform_versions field of the given host.

Signature:

(int set) get_virtual_hardware_platform_versions (session_id s, host ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: int set
value of the field

RPC name: get_by_uuid

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

Signature:

(host ref) get_by_uuid (session_id s, string uuid)
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>label</td>
<td>label of object to return</td>
</tr>
</tbody>
</table>

Return Type: host ref set
references to objects with matching names
2.19 Class: host_crashdump

2.19.1 Fields for class: host_crashdump

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_crashdump</td>
<td>Represents a host crash dump.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO run</td>
<td>uuid</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>RO ins</td>
<td>host</td>
<td>host ref</td>
<td>Host the crashdump relates to</td>
</tr>
<tr>
<td>RO run</td>
<td>timestamp</td>
<td>datetime</td>
<td>Time the crash happened</td>
</tr>
<tr>
<td>RO run</td>
<td>size</td>
<td>int</td>
<td>Size of the crashdump</td>
</tr>
<tr>
<td>RW</td>
<td>other_config</td>
<td>(string → string) map</td>
<td>additional configuration</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_crashdump ref</td>
<td>self</td>
<td>The host crashdump to destroy</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_crashdump ref</td>
<td>self</td>
<td>The host crashdump to upload</td>
</tr>
<tr>
<td>string</td>
<td>url</td>
<td>The URL to upload to</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>options</td>
<td>Extra configuration operations</td>
</tr>
</tbody>
</table>

Return Type: void
RPC name: \texttt{get\_all}

Overview:
Return a list of all the host\_crashdumps known to the system.

Signature:
\[(\text{host\_crashdump ref set}) \text{ get\_all (session\_id s)}\]

Return Type: host\_crashdump ref set
references to all objects

RPC name: \texttt{get\_all\_records}

Overview:
Return a map of host\_crashdump references to host\_crashdump records for all host\_crashdumps known to the system.

Signature:
\[((\text{host\_crashdump ref} \to\text{ host\_crashdump record}) \text{ map}) \text{ get\_all\_records (session\_id s)}\]

Return Type: (host\_crashdump ref \to host\_crashdump record) map
records of all objects

RPC name: \texttt{get\_uuid}

Overview:
Get the uuid field of the given host\_crashdump.

Signature:
\[\text{string get\_uuid (session\_id s, host\_crashdump ref self)}\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_crashdump ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: \texttt{get\_host}

Overview:
Get the host field of the given host\_crashdump.

Signature:
\[(\text{host ref}) \text{ get\_host (session\_id s, host\_crashdump ref self)}\]

271
### Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_crashdump ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: host ref

value of the field

----

RPC name: `get_timestamp`

Overview:
Get the timestamp field of the given host_crashdump.

Signature:

```python
datetime get_timestamp (session_id s, host_crashdump ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_crashdump ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: datetime

value of the field

----

RPC name: `get_size`

Overview:
Get the size field of the given host_crashdump.

Signature:

```python
int get_size (session_id s, host_crashdump ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_crashdump ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: int

value of the field

----

RPC name: `get_other_config`

Overview:
Get the other_config field of the given host_crashdump.

Signature:

```python
((string -> string) map) get_other_config (session_id s, host_crashdump ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_crashdump ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_crashdump ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_crashdump ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_crashdump ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_crashdump ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: host_crashdump record
all fields from the object
2.20 Class: `host_patch`

2.20.1 Fields for class: `host_patch`

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_patch</td>
<td>Represents a patch stored on a server.</td>
<td>RO</td>
<td>uuid</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ROins</td>
<td>name_label</td>
<td>string</td>
<td>a human-readable name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ROins</td>
<td>name_description</td>
<td>string</td>
<td>a notes field containing human-readable description</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ROins</td>
<td>version</td>
<td>string</td>
<td>Patch version number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ROrun</td>
<td>host</td>
<td>host ref</td>
<td>Host the patch relates to</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ROrun</td>
<td>timestamp_applied</td>
<td>datetime</td>
<td>Time the patch was applied</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ROrun</td>
<td>size</td>
<td>int</td>
<td>Size of the patch</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ROins</td>
<td>pool_patch</td>
<td>pool_patch ref</td>
<td>The patch applied</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RW</td>
<td>other_config</td>
<td>(string → string) map</td>
<td>additional configuration</td>
</tr>
</tbody>
</table>

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:**

```plaintext
void destroy (session_id s, host_patch ref self)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_patch ref</td>
<td>self</td>
<td>The patch to destroy</td>
</tr>
</tbody>
</table>

**Return Type:** void

RPC name: apply

**Overview:** This message is deprecated Apply the selected patch and return its output.

**Signature:**

```plaintext
string apply (session_id s, host_patch ref self)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_patch ref</td>
<td>self</td>
<td>The patch to apply</td>
</tr>
</tbody>
</table>

**Return Type:** string

the output of the patch application process
2.20  CLASS: HOST_PATCH

RPC name: get_all

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

Signature:

(host_patch ref set) get_all (session_id s)

Return Type: host_patch ref set
references to all objects

RPC name: get_all_records

Overview:
Return a map of host_patch references to host_patch records for all host_patchs known to the system.

Signature:

((host_patch ref -> host_patch record) map) get_all_records (session_id s)

Return Type: (host_patch ref -> host_patch record) map
records of all objects

RPC name: get_uuid

Overview:
Get the uuid field of the given host_patch.

Signature:

string get_uuid (session_id s, host_patch ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: host ref

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: bool

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: datetime

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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)
2.20. CLASS: HOST_PATCH  

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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)
2.20. **CLASS: HOST_PATCH**

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_patch ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>label</td>
<td>label of object to return</td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>host\_metrics</code></td>
<td><em>The metrics associated with a host.</em></td>
<td>RO</td>
<td><code>uuid</code></td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td><code>memory_total</code></td>
<td>Total host memory (bytes)</td>
<td>RO</td>
<td><code>memory_total</code></td>
<td>int</td>
<td></td>
</tr>
<tr>
<td><code>memory_free</code></td>
<td>Free host memory (bytes)</td>
<td>RO</td>
<td><code>memory_free</code></td>
<td>int</td>
<td></td>
</tr>
<tr>
<td><code>live</code></td>
<td>Pool master thinks this host is live</td>
<td>RO</td>
<td><code>live</code></td>
<td>bool</td>
<td></td>
</tr>
<tr>
<td><code>last\_updated</code></td>
<td>Time at which this information was last updated</td>
<td>RO</td>
<td><code>last\_updated</code></td>
<td>datetime</td>
<td></td>
</tr>
<tr>
<td><code>other\_config</code></td>
<td>additional configuration</td>
<td>RW</td>
<td><code>other\_config</code></td>
<td>(string → string) map</td>
<td></td>
</tr>
</tbody>
</table>

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:

```plaintext
(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:

```plaintext
((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:

```plaintext
string get\_uuid (session\_id s, host\_metrics ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host.metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host.metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host.metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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)
2.21.  CLASS: HOST_METRICS

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: host_metrics record
all fields from the object
2.22 Class: host_cpu

2.22.1 Fields for class: host_cpu

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host</td>
<td>A physical CPU.</td>
<td></td>
<td>cpu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RO</td>
<td>uuid</td>
<td>run</td>
<td>string</td>
<td></td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>RO</td>
<td>host</td>
<td>run</td>
<td>host ref</td>
<td></td>
<td>the host the CPU is in</td>
</tr>
<tr>
<td>RO</td>
<td>number</td>
<td>run</td>
<td>int</td>
<td></td>
<td>the number of the physical CPU within the host</td>
</tr>
<tr>
<td>RO</td>
<td>vendor</td>
<td>run</td>
<td>string</td>
<td></td>
<td>the vendor of the physical CPU</td>
</tr>
<tr>
<td>RO</td>
<td>speed</td>
<td>run</td>
<td>int</td>
<td></td>
<td>the speed of the physical CPU</td>
</tr>
<tr>
<td>RO</td>
<td>modelname</td>
<td>run</td>
<td>string</td>
<td></td>
<td>the model name of the physical CPU</td>
</tr>
<tr>
<td>RO</td>
<td>family</td>
<td>run</td>
<td>int</td>
<td></td>
<td>the family (number) of the physical CPU</td>
</tr>
<tr>
<td>RO</td>
<td>model</td>
<td>run</td>
<td>int</td>
<td></td>
<td>the model number of the physical CPU</td>
</tr>
<tr>
<td>RO</td>
<td>stepping</td>
<td>run</td>
<td>string</td>
<td></td>
<td>the stepping of the physical CPU</td>
</tr>
<tr>
<td>RO</td>
<td>flags</td>
<td>run</td>
<td>string</td>
<td></td>
<td>the flags of the physical CPU (a decoded version of the features field)</td>
</tr>
<tr>
<td>RO</td>
<td>features</td>
<td>run</td>
<td>string</td>
<td></td>
<td>the physical CPU feature bitmap</td>
</tr>
<tr>
<td>RW</td>
<td>other_config</td>
<td></td>
<td>(string → string) map</td>
<td>float</td>
<td>the current CPU utilisation</td>
</tr>
</tbody>
</table>

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:

```c
string get_uuid (session_id s, host_cpu ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_cpu ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get_host

Overview:
Get the host field of the given host_cpu.

Signature:

```c
(host ref) get_host (session_id s, host_cpu ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_cpu ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: host ref
value of the field

RPC name: get_number

Overview:
Get the number field of the given host_cpu.

Signature:

```c
int get_number (session_id s, host_cpu ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_cpu ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: int
value of the field
RPC name: get_vendor

Overview:
Get the vendor field of the given host_cpu.

Signature:

```c
string get_vendor (session_id s, host_cpu ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_cpu ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get_speed

Overview:
Get the speed field of the given host_cpu.

Signature:

```c
int get_speed (session_id s, host_cpu ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_cpu ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: int
value of the field

RPC name: get_modelname

Overview:
Get the modelname field of the given host_cpu.

Signature:

```c
string get_modelname (session_id s, host_cpu ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_cpu ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field
RPC name: get_family

Overview:
Get the family field of the given host_cpu.

Signature:

```c
int get_family (session_id s, host_cpu ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_cpu</td>
<td>ref</td>
<td></td>
</tr>
<tr>
<td>self</td>
<td></td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: int
value of the field

RPC name: get_model

Overview:
Get the model field of the given host_cpu.

Signature:

```c
int get_model (session_id s, host_cpu ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_cpu</td>
<td>ref</td>
<td></td>
</tr>
<tr>
<td>self</td>
<td></td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: int
value of the field

RPC name: get_stepping

Overview:
Get the stepping field of the given host_cpu.

Signature:

```c
string get_stepping (session_id s, host_cpu ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_cpu</td>
<td>ref</td>
<td></td>
</tr>
<tr>
<td>self</td>
<td></td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field
2.22. CLASS: HOST_CPU

CHAPTER 2. API REFERENCE

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_cpu ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_cpu ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_cpu ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: float
value of the field
RPC name: get\_other\_config

**Overview:**
Get the other\_config field of the given host\_cpu.

**Signature:**

$((\text{string} \rightarrow \text{string}) \text{ map}) \text{ get\_other\_config} (\text{session\_id} \ s, \text{ host\_cpu} \ \text{ ref} \ \text{ self})$

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_cpu ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**Return Type:** $(\text{string} \rightarrow \text{string}) \text{ map}$

value of the field

RPC name: set\_other\_config

**Overview:**
Set the other\_config field of the given host\_cpu.

**Signature:**

$\text{ void set\_other\_config} (\text{session\_id} \ s, \text{ host\_cpu} \ \text{ ref} \ \text{ self}, \ (\text{string} \rightarrow \text{string}) \text{ map} \ \text{ value})$

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_cpu ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>$(\text{string} \rightarrow \text{string}) \text{ map}$</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

**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:**

$\text{ void add\_to\_other\_config} (\text{session\_id} \ s, \text{ host\_cpu} \ \text{ ref} \ \text{ self}, \ \text{ string} \ \text{ key}, \ \text{ string} \ \text{ value})$

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_cpu ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

**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:
```java
void remove_from_other_config (session_id s, host_cpu ref self, string key)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_cpu ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:
```java
(host_cpu ref) get_by_uuid (session_id s, string uuid)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:
```java
(host_cpu record) get_record (session_id s, host_cpu ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_cpu ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: host_cpu record
all fields from the object
## 2.23 Class: network

### 2.23.1 Fields for class: network

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network</td>
<td>A virtual network.</td>
<td></td>
<td>uuid</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>name</td>
<td>a human-readable name</td>
<td>RW</td>
<td>name</td>
<td>string</td>
<td>a human-readable name</td>
</tr>
<tr>
<td>description</td>
<td>a notes field containing human-readable description</td>
<td>RW</td>
<td>description</td>
<td>string</td>
<td>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.</td>
</tr>
<tr>
<td>allowed_operations</td>
<td>network_operations set</td>
<td>RO</td>
<td>allowed_operations</td>
<td>network_operations</td>
<td>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.</td>
</tr>
<tr>
<td>current_operations</td>
<td>(string → network_operations) map</td>
<td>RO</td>
<td>current_operations</td>
<td>(string → string) map</td>
<td>links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.</td>
</tr>
<tr>
<td>VIFs</td>
<td>VIF ref set</td>
<td>RO</td>
<td>VIFs</td>
<td>VIF ref set</td>
<td>list of connected vifs</td>
</tr>
<tr>
<td>PIFs</td>
<td>PIF ref set</td>
<td>RO</td>
<td>PIFs</td>
<td>PIF ref set</td>
<td>list of connected pifs</td>
</tr>
<tr>
<td>MTU</td>
<td>MTU in octets</td>
<td>RW</td>
<td>MTU</td>
<td>int</td>
<td>MTU in octets</td>
</tr>
<tr>
<td>other_config</td>
<td>(string → string) map</td>
<td>RW</td>
<td>other_config</td>
<td>(string → string) map</td>
<td>additional configuration</td>
</tr>
<tr>
<td>bridge</td>
<td>name of the bridge corresponding to this network on the local host</td>
<td>RO</td>
<td>bridge</td>
<td>string</td>
<td>name of the bridge corresponding to this network on the local host</td>
</tr>
<tr>
<td>blobs</td>
<td>Binary blobs associated with this network</td>
<td>RO</td>
<td>blobs</td>
<td>(string → blob ref) map</td>
<td>Binary blobs associated with this network user-specified tags for categorization purposes</td>
</tr>
<tr>
<td>tags</td>
<td>string set</td>
<td>RW</td>
<td>tags</td>
<td>string set</td>
<td>The network will use this value to determine the behaviour of all VIFs where locking_mode = default</td>
</tr>
<tr>
<td>default_locking_mode</td>
<td>network_default_locking_mode</td>
<td>RO</td>
<td>default_locking_mode</td>
<td>network_default_locking_mode</td>
<td>The IP addresses assigned to VIFs on networks that have active xapi-managed DHCP</td>
</tr>
<tr>
<td>assigned_ips</td>
<td>(VIF ref → string) map</td>
<td>RO</td>
<td>assigned_ips</td>
<td>(VIF ref → string) map</td>
<td></td>
</tr>
</tbody>
</table>

### 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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network_ref</td>
<td>network</td>
<td>The network</td>
</tr>
<tr>
<td>string</td>
<td>name</td>
<td>The name associated with the blob</td>
</tr>
<tr>
<td>string</td>
<td>mime_type</td>
<td>The mime type for the data. Empty string translates to application/octet-stream</td>
</tr>
<tr>
<td>bool</td>
<td>public</td>
<td>True if the blob should be publicly available</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>network</td>
<td>The network</td>
</tr>
<tr>
<td>network_default_locking_mode</td>
<td>value</td>
<td>The default locking mode for VIFs attached to this network.</td>
</tr>
</tbody>
</table>

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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

\[(\text{string} \rightarrow \text{network_operations}) \map \text{get_current_operations} (\text{session_id} s, \text{network ref} \text{self})\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: \((\text{string} \rightarrow \text{network_operations}) \map\)
value of the field

RPC name: get_VIFs

Overview:
Get the VIFs field of the given network.

Signature:

\((\text{VIF ref set}) \text{get_VIFs} (\text{session_id} s, \text{network ref} \text{self})\)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: \(\text{VIF ref set}\)
value of the field

RPC name: get_PIFs

Overview:
Get the PIFs field of the given network.

Signature:

\((\text{PIF ref set}) \text{get_PIFs} (\text{session_id} s, \text{network ref} \text{self})\)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: \(\text{PIF ref set}\)
value of the field
RPC name: get_MTU

Overview:
Get the MTU field of the given network.

Signature:

```c
int get_MTU (session_id s, network ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: int
value of the field

RPC name: set_MTU

Overview:
Set the MTU field of the given network.

Signature:

```c
void set_MTU (session_id s, network ref self, int value)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>int</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_other_config

Overview:
Get the other_config field of the given network.

Signature:

```c
((string -> string) map) get_other_config (session_id s, network ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:**

```java
def set_other_config(session_id s, network ref self, (string -> string) map value)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

**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:**

```java
def add_to_other_config(session_id s, network ref self, string key, string value)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

**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:**

```java
def remove_from_other_config(session_id s, network ref self, string key)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string set
value of the field
RPC name: set_tags

Overview:
Set the tags field of the given network.

Signature:

```c
void set_tags (session_id s, network ref self, string set value)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string set</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to add</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network record</td>
<td>args</td>
<td>All constructor arguments</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>label</td>
<td>label of object to return</td>
</tr>
</tbody>
</table>

Return Type: network ref set
references to objects with matching names
## 2.24 Class: VIF

### 2.24.1 Fields for class: VIF

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VIF</strong></td>
<td>A virtual network interface.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Quals

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>uuid</strong></td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td><strong>allowed_operations</strong></td>
<td>vif_operations set</td>
<td>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.</td>
</tr>
<tr>
<td><strong>current_operations</strong></td>
<td>(string → vif_operations) map</td>
<td>links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>device</strong></td>
<td>string</td>
<td>order in which VIF backends are created by xapi</td>
</tr>
<tr>
<td><strong>network</strong></td>
<td>network ref</td>
<td>virtual network to which this vif is connected</td>
</tr>
<tr>
<td><strong>VM</strong></td>
<td>VM ref</td>
<td>virtual machine to which this vif is connected</td>
</tr>
<tr>
<td><strong>MAC</strong></td>
<td>string</td>
<td>ethernet MAC address of virtual interface, as exposed to guest</td>
</tr>
<tr>
<td><strong>MTU</strong></td>
<td>int</td>
<td>MTU in octets</td>
</tr>
<tr>
<td><strong>other_config</strong></td>
<td>(string → string) map</td>
<td>additional configuration</td>
</tr>
<tr>
<td><strong>currently_attached</strong></td>
<td>bool</td>
<td>is the device currently attached (erased on reboot)</td>
</tr>
<tr>
<td><strong>status_code</strong></td>
<td>int</td>
<td>error/success code associated with last attach-operation (erased on reboot)</td>
</tr>
<tr>
<td><strong>status_detail</strong></td>
<td>string</td>
<td>error/success information associated with last attach-operation status (erased on reboot)</td>
</tr>
<tr>
<td><strong>runtime_properties</strong></td>
<td>(string → string) map</td>
<td>Device runtime properties</td>
</tr>
<tr>
<td><strong>qos_algorithm_type</strong></td>
<td>string</td>
<td>QoS algorithm to use</td>
</tr>
<tr>
<td><strong>qos_algorithm_params</strong></td>
<td>(string → string) map</td>
<td>parameters for chosen QoS algorithm supported QoS algorithms for this VIF</td>
</tr>
<tr>
<td><strong>qos_supported_algorithms</strong></td>
<td>string set</td>
<td>metrics associated with this VIF</td>
</tr>
<tr>
<td><strong>metrics</strong></td>
<td>VIF.metrics ref</td>
<td>true if the MAC was autogenerated; false indicates it was set manually</td>
</tr>
<tr>
<td><strong>MAC.autogenerated</strong></td>
<td>bool</td>
<td>current locking mode of the VIF</td>
</tr>
<tr>
<td><strong>locking_mode</strong></td>
<td>vif.locking_mode</td>
<td>A list of IPv4 addresses which can be used to filter traffic passing through this VIF</td>
</tr>
<tr>
<td><strong>ipv4_allowed</strong></td>
<td>string set</td>
<td>A list of IPv6 addresses which can be used to filter traffic passing through this VIF</td>
</tr>
<tr>
<td><strong>ipv6_allowed</strong></td>
<td>string set</td>
<td>Determines whether IPv4 addresses are configured on the VIF</td>
</tr>
<tr>
<td><strong>ipv4_configuration_mode</strong></td>
<td>vif_ipv4_configuration_mode</td>
<td>IPv4 addresses in CIDR format</td>
</tr>
<tr>
<td><strong>ipv4_addresses</strong></td>
<td>string set</td>
<td>IPv4 gateway (the empty string means that no gateway is set)</td>
</tr>
<tr>
<td><strong>ipv4_gateway</strong></td>
<td>string</td>
<td></td>
</tr>
</tbody>
</table>
### 2.24.2 RPCs associated with class: VIF

**RPC name: plug**

**Overview:**
Hotplug the specified VIF, dynamically attaching it to the running VM.

**Signature:**

```java
void plug (session_id s, VIF ref self)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>The VIF to hotplug</td>
</tr>
</tbody>
</table>

**Return Type:** void

**RPC name: unplug**

**Overview:**
Hot-unplug the specified VIF, dynamically unattaching it from the running VM.

**Signature:**

```java
void unplug (session_id s, VIF ref self)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>The VIF to hot-unplug</td>
</tr>
</tbody>
</table>

**Return Type:** void

**RPC name: unplug_force**

**Overview:**
Forcibly unplug the specified VIF.

**Signature:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>The VIF to forcibly unplug</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>The VIF whose locking mode will be set</td>
</tr>
<tr>
<td>vif_locking_mode</td>
<td>value</td>
<td>The new locking mode for the VIF</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>The VIF which the IP addresses will be associated with</td>
</tr>
<tr>
<td>string set</td>
<td>value</td>
<td>The IP addresses which will be associated with the VIF</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>The VIF which the IP address will be associated with</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>The IP address which will be associated with the VIF</td>
</tr>
</tbody>
</table>

Return Type: void
2.24. CLASS: VIF

CHAPTER 2. API REFERENCE

RPC name: remove_ipv4_allowed

Overview:
Removes an IPv4 address from this VIF.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>The VIF from which the IP address will be removed</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>The IP address which will be removed from the VIF</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: set_ipv6_allowed

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

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>The VIF which the IP addresses will be associated with</td>
</tr>
<tr>
<td>string set</td>
<td>value</td>
<td>The IP addresses which will be associated with the VIF</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: add_ipv6_allowed

Overview:
Associates an IPv6 address with this VIF.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>The VIF which the IP address will be associated with</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>The IP address which will be associated with the VIF</td>
</tr>
</tbody>
</table>

309
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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>The VIF from which the IP address will be removed</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>The IP address which will be removed from the VIF</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: configure_ipv4

Overview:
Configure IPv4 settings for this virtual interface.

Signature:

void configure_ipv4 (session_id s, VIF ref self, vif_ipv4_configuration_mode mode, string address, string gateway)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>The VIF to configure</td>
</tr>
<tr>
<td>vif_ipv4_configuration_mode</td>
<td>mode</td>
<td>Whether to use static or no IPv4 assignment</td>
</tr>
<tr>
<td>string</td>
<td>address</td>
<td>The IPv4 address in addr/prefix length format (for static mode only)</td>
</tr>
<tr>
<td>string</td>
<td>gateway</td>
<td>The IPv4 gateway (for static mode only; leave empty to not set a gateway)</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: configure_ipv6

Overview:
Configure IPv6 settings for this virtual interface.

Signature:

void configure_ipv6 (session_id s, VIF ref self, vif_ipv6_configuration_mode mode, string address, string gateway)
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>The VIF to configure</td>
</tr>
<tr>
<td>vif_ipv6_configuration_mode</td>
<td>mode</td>
<td>Whether to use static or no IPv6 assignment</td>
</tr>
<tr>
<td>string</td>
<td>address</td>
<td>The IPv6 address in <code>addr/prefix length</code> format (for static mode only)</td>
</tr>
<tr>
<td>string</td>
<td>gateway</td>
<td>The IPv6 gateway (for static mode only; leave empty to not set a gateway)</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_all

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

Signature:

```
(VIF ref set) get_all (session_id s)
```

Return Type: VIF ref set
references to all objects

RPC name: get_all_records

Overview:
Return a map of VIF references to VIF records for all VIFs known to the system.

Signature:

```
((VIF ref -> VIF record) map) get_all_records (session_id s)
```

Return Type: (VIF ref -> VIF record) map
records of all objects

RPC name: get_uuid

Overview:
Get the uuid field of the given VIF.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>
Return Type: string
value of the field

RPC name: get_allowed_operations

Overview:
Get the allowed_operations field of the given VIF.

Signature:

\[ \text{(vif\_operations set) get\_allowed\_operations (session\_id s, VIF ref self)} \]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

\[ \text{((string -> vif\_operations) map) get\_current\_operations (session\_id s, VIF ref self)} \]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: (string \(\to\) vif\_operations) map
value of the field

RPC name: get_device

Overview:
Get the device field of the given VIF.

Signature:

\[ \text{string get\_device (session\_id s, VIF ref self)} \]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

**RPC name: get_network**

**Overview:**
Get the network field of the given VIF.

**Signature:**

```plaintext
(network ref) get_network (session_id s, VIF ref self)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**Return Type:** network ref

value of the field

**RPC name: get_VM**

**Overview:**
Get the VM field of the given VIF.

**Signature:**

```plaintext
(VM ref) get_VM (session_id s, VIF ref self)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**Return Type:** VM ref

value of the field

**RPC name: get_MAC**

**Overview:**
Get the MAC field of the given VIF.

**Signature:**

```plaintext
string get_MAC (session_id s, VIF ref self)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**Return Type:** string

value of the field
RPC name: get_MTU

Overview:
Get the MTU field of the given VIF.

Signature:

```c
int get_MTU (session_id s, VIF ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: int

value of the field

RPC name: get_other_config

Overview:
Get the other_config field of the given VIF.

Signature:

```c
((string -> string) map) get_other_config (session_id s, VIF ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```c
void set_other_config (session_id s, VIF ref self, (string -> string) map value)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

**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:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

**Return Type:** void

RPC name: `get_currently_attached`

**Overview:**
Get the `currently_attached` field of the given VIF.

**Signature:**

```c
bool get_currently_attached (session_id s, VIF ref self)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**Return Type:** bool

value of the field
RPC name: get_status_code

Overview:
Get the status_code field of the given VIF.

Signature:

```c
int get_status_code (session_id s, VIF ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: int
value of the field

RPC name: get_status_detail

Overview:
Get the status_detail field of the given VIF.

Signature:

```c
string get_status_detail (session_id s, VIF ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get_runtime_properties

Overview:
Get the runtime_properties field of the given VIF.

Signature:

```c
((string -> string) map) get_runtime_properties (session_id s, VIF ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```c
string get_qos_algorithm_type (session_id s, VIF ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_qos_algorithm_params

Overview:
Get the qos/algorithm_params field of the given VIF.

Signature:

```c
((string -> string) map) get_qos_algorithm_params (session_id s, VIF ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: (string → string) map

value of the field
2.24. CLASS: VIF

CHAPTER 2. API REFERENCE

RPC name: set_qos_algorithm_params

Overview:
Set the qos/algorithm_params field of the given VIF.

Signature:

```java
void set_qos_algorithm_params (session_id s, VIF ref self, (string -> string) map value)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

```java
void add_to_qos_algorithm_params (session_id s, VIF ref self, string key, string value)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

```java
void remove_from_qos_algorithm_params (session_id s, VIF ref self, string key)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string set
value of the field
RPC name: get_ipv4_configuration_mode

Overview:
Get the ipv4_configuration_mode field of the given VIF.

Signature:

(vif_ipv4_configuration_mode) get_ipv4_configuration_mode (session_id s, VIF ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: vif_ipv4_configuration_mode
value of the field

RPC name: get_ipv4_addresses

Overview:
Get the ipv4_addresses field of the given VIF.

Signature:

(string set) get_ipv4_addresses (session_id s, VIF ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string set
value of the field

RPC name: get_ipv4_gateway

Overview:
Get the ipv4_gateway field of the given VIF.

Signature:

string get_ipv4_gateway (session_id s, VIF ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field
RPC name: get_ipv6_configuration_mode

Overview:
Get the ipv6_configuration_mode field of the given VIF.

Signature:

\[(vif_ipv6_configuration_mode) \text{get ipv6_configuration_mode} (session\_id s, \text{VIF ref self})\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: \text{vif_ipv6_configuration_mode} value of the field

RPC name: get_ipv6_addresses

Overview:
Get the ipv6_addresses field of the given VIF.

Signature:

\[(\text{string set}) \text{get ipv6_addresses} (session\_id s, \text{VIF ref self})\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: \text{string set} value of the field

RPC name: get_ipv6_gateway

Overview:
Get the ipv6_gateway field of the given VIF.

Signature:

\[\text{string get ipv6_gateway} (session\_id s, \text{VIF ref self})\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: \text{string} value of the field

322
**RPC name: create**

**Overview:**
Create a new VIF instance, and return its handle.

**Signature:**

\[
(VIF \text{ ref}) \text{ create (session\_id s, VIF record args)}
\]

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>args</td>
<td>All constructor arguments</td>
</tr>
</tbody>
</table>

**Return Type:** VIF ref
reference to the newly created object

**RPC name: destroy**

**Overview:**
Destroy the specified VIF instance.

**Signature:**

\[
\text{void destroy (session\_id s, VIF ref self)}
\]

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**Return Type:** void

**RPC name: get_by_uuid**

**Overview:**
Get a reference to the VIF instance with the specified UUID.

**Signature:**

\[
(VIF \text{ ref}) \text{ get\_by\_uuid (session\_id s, string uuid)}
\]

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

**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:

\[(VIF \text{ record}) \ \text{get}\_\text{record} (\text{session\_id} \ s, \ VIF \ \text{ref} \ \text{self})\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: VIF record
all fields from the object
2.25 Class: VIF\_metrics

2.25.1 Fields for class: VIF\_metrics

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF_metrics</td>
<td>The metrics associated with a virtual network device.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RO</td>
<td>run</td>
<td>uuid</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RO</td>
<td>io_read_kbs</td>
<td>float</td>
<td>Read bandwidth (KiB/s)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RO</td>
<td>io_write_kbs</td>
<td>float</td>
<td>Write bandwidth (KiB/s)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RO</td>
<td>run</td>
<td>last_updated</td>
<td>Time at which this information was last updated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RW</td>
<td>other_config</td>
<td>(string → string) map</td>
<td>additional configuration</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```python
float get_io_read_kbs (session_id s, VIF_metrics ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```python
float get_io_write_kbs (session_id s, VIF_metrics ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: float
value of the field

RPC name: get_last_updated

Overview:
Get the last_updated field of the given VIF_metrics.

Signature:

```python
datetime get_last_updated (session_id s, VIF_metrics ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: `datetime`

value of the field

RPC name: `get_other_config`

Overview:
Get the other config field of the given VIF_metrics.

Signature:

```plaintext
((string -> string) map) get_other_config (session_id s, VIF_metrics ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```plaintext
void set_other_config (session_id s, VIF_metrics ref self, (string -> string) map value)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

```plaintext
void add_to_other_config (session_id s, VIF_metrics ref self, string key, string value)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIF.metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: VIF.metrics record
all fields from the object
## 2.26 Class: PIF

### 2.26.1 Fields for class: PIF

<table>
<thead>
<tr>
<th>Name</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PIF</td>
<td></td>
<td>A physical network interface (note separate VLANs are represented as several PIFs).</td>
</tr>
<tr>
<td>Quals</td>
<td>Field</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>RO_run</td>
<td>uuid</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>RO_ins</td>
<td>device</td>
<td>string</td>
<td>Machine-readable name of the interface (e.g. eth0)</td>
</tr>
<tr>
<td>RO_ins</td>
<td>network</td>
<td>network ref</td>
<td>Virtual network to which this pif is connected</td>
</tr>
<tr>
<td>RO_ins</td>
<td>host</td>
<td>host ref</td>
<td>Physical machine to which this pif is connected</td>
</tr>
<tr>
<td>RO_ins</td>
<td>MAC</td>
<td>string</td>
<td>Ethernet MAC address of physical interface</td>
</tr>
<tr>
<td>RO_ins</td>
<td>MTU</td>
<td>int</td>
<td>MTU in octets</td>
</tr>
<tr>
<td>RO_ins</td>
<td>VLAN</td>
<td>int</td>
<td>VLAN tag for all traffic passing through this interface</td>
</tr>
<tr>
<td>RO_run</td>
<td>metrics</td>
<td>PIF_metrics ref</td>
<td>Metrics associated with this PIF</td>
</tr>
<tr>
<td>RO_run</td>
<td>physical</td>
<td>bool</td>
<td>True if this represents a physical network interface</td>
</tr>
<tr>
<td>RO_run</td>
<td>currently_attached</td>
<td>bool</td>
<td>True if this interface is online</td>
</tr>
<tr>
<td>RO_run</td>
<td>ip_configuration_mode</td>
<td>ip_configuration_mode</td>
<td>Sets if and how this interface gets an IP address</td>
</tr>
<tr>
<td>RO_run</td>
<td>IP</td>
<td>string</td>
<td>IP address</td>
</tr>
<tr>
<td>RO_run</td>
<td>netmask</td>
<td>string</td>
<td>IP netmask</td>
</tr>
<tr>
<td>RO_run</td>
<td>gateway</td>
<td>string</td>
<td>IP gateway</td>
</tr>
<tr>
<td>RO_run</td>
<td>DNS</td>
<td>string</td>
<td>IP address of DNS servers to use</td>
</tr>
<tr>
<td>RO_run</td>
<td>bond_slave_of</td>
<td>Bond ref</td>
<td>Indicates which bond this interface is part of</td>
</tr>
<tr>
<td>RO_run</td>
<td>bond_master_of</td>
<td>Bond ref set</td>
<td>Indicates this PIF represents the results of a bond</td>
</tr>
<tr>
<td>RO_run</td>
<td>VLAN_master_of</td>
<td>VLAN ref</td>
<td>Indicates which VLAN this interface receives untagged traffic from</td>
</tr>
<tr>
<td>RO_run</td>
<td>VLAN_slave_of</td>
<td>VLAN ref set</td>
<td>Indicates which VLANs this interface transmits tagged traffic to</td>
</tr>
<tr>
<td>RO_run</td>
<td>management</td>
<td>bool</td>
<td>Indicates whether the control software is listening for connections on this interface</td>
</tr>
<tr>
<td>RW</td>
<td>other_config</td>
<td>(string → string) map</td>
<td>Additional configuration</td>
</tr>
<tr>
<td>RW</td>
<td>disallow_unplug</td>
<td>bool</td>
<td>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)</td>
</tr>
<tr>
<td>RO_run</td>
<td>tunnel_access_PIF_of</td>
<td>tunnel ref set</td>
<td>Indicates to which tunnel this PIF gives access</td>
</tr>
<tr>
<td>RO_run</td>
<td>tunnel_transport_PIF_of</td>
<td>tunnel ref set</td>
<td>Indicates to which tunnel this PIF provides transport</td>
</tr>
</tbody>
</table>
### 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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>device</td>
<td>physical interface on which to create the VLAN interface</td>
</tr>
<tr>
<td>network ref</td>
<td>network</td>
<td>network to which this interface should be connected</td>
</tr>
<tr>
<td>host ref</td>
<td>host</td>
<td>physical machine to which this PIF is connected</td>
</tr>
<tr>
<td>int</td>
<td>VLAN</td>
<td>VLAN tag for the new interface</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>the PIF object to destroy</td>
</tr>
</tbody>
</table>

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, string gateway)`

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>the PIF object to reconfigure</td>
</tr>
<tr>
<td>ip_configuration_mode</td>
<td>mode</td>
<td>whether to use dynamic/static/no-assignment</td>
</tr>
<tr>
<td>string</td>
<td>IP</td>
<td>the new IP address</td>
</tr>
<tr>
<td>string</td>
<td>netmask</td>
<td>the new netmask</td>
</tr>
<tr>
<td>string</td>
<td>gateway</td>
<td>the new gateway</td>
</tr>
<tr>
<td>string</td>
<td>DNS</td>
<td>the new DNS settings</td>
</tr>
</tbody>
</table>

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:

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

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>the PIF object to reconfigure</td>
</tr>
<tr>
<td>primary address type</td>
<td>primary address type</td>
<td>Whether to prefer IPv4 or IPv6 connections</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host on which to scan</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host on which the interface exists</td>
</tr>
<tr>
<td>string</td>
<td>MAC</td>
<td>The MAC address of the interface</td>
</tr>
<tr>
<td>string</td>
<td>device</td>
<td>The device name to use for the interface</td>
</tr>
<tr>
<td>bool</td>
<td>managed</td>
<td>Indicates whether the interface is managed by</td>
</tr>
<tr>
<td></td>
<td></td>
<td>xapi (defaults to “true”)</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>The PIF object to destroy</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>the PIF object to unplug</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: plug

Overview:
Attempt to bring up a physical interface.

Signature:
void plug (session_id s, PIF ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>the PIF object to plug</td>
</tr>
</tbody>
</table>

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, int MTU, bool VLAN, string ip_configuration_mode, string IP, string netmask, string gateway, string DNS, Bond ref bond_master_of, VLAN ref VLAN_master_of, bool management, (string → string) map other_config, bool disallow_unplug, ipv6_configuration_mode ipv6_configuration_mode, string IPv6, string ipv6_gateway, primary_address_type managed, (string → string) map properties)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>device</td>
<td></td>
</tr>
<tr>
<td>network ref</td>
<td>network</td>
<td></td>
</tr>
<tr>
<td>host ref</td>
<td>host</td>
<td></td>
</tr>
<tr>
<td>string</td>
<td>MAC</td>
<td></td>
</tr>
<tr>
<td>int</td>
<td>MTU</td>
<td></td>
</tr>
<tr>
<td>int</td>
<td>VLAN</td>
<td></td>
</tr>
<tr>
<td>bool</td>
<td>physical</td>
<td></td>
</tr>
<tr>
<td>ip_configuration_mode</td>
<td>ip_configuration_mode</td>
<td></td>
</tr>
<tr>
<td>string</td>
<td>IP</td>
<td></td>
</tr>
<tr>
<td>string</td>
<td>netmask</td>
<td></td>
</tr>
<tr>
<td>string</td>
<td>gateway</td>
<td></td>
</tr>
<tr>
<td>string</td>
<td>DNS</td>
<td></td>
</tr>
<tr>
<td>Bond ref</td>
<td>bond_slave_of</td>
<td></td>
</tr>
<tr>
<td>VLAN ref</td>
<td>VLAN_master_of</td>
<td></td>
</tr>
<tr>
<td>bool</td>
<td>management</td>
<td></td>
</tr>
<tr>
<td>(string → string) map</td>
<td>other_config</td>
<td></td>
</tr>
<tr>
<td>bool</td>
<td>disallow_unplug</td>
<td></td>
</tr>
<tr>
<td>ipv6_configuration_mode</td>
<td>ipv6_configuration_mode</td>
<td></td>
</tr>
<tr>
<td>string</td>
<td>IPv6</td>
<td></td>
</tr>
<tr>
<td>string</td>
<td>ipv6_gateway</td>
<td></td>
</tr>
<tr>
<td>primary_address_type</td>
<td>primary_address_type</td>
<td></td>
</tr>
<tr>
<td>bool</td>
<td>managed</td>
<td></td>
</tr>
<tr>
<td>(string → string) map</td>
<td>properties</td>
<td></td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>The ref of the PIF whose database record should be destroyed</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>The PIF</td>
</tr>
<tr>
<td>string</td>
<td>name</td>
<td>The property name</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>The property value</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_all

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

Signature:

(PIF ref set) get_all (session_id s)

Return Type: PIF ref set
references to all objects

RPC name: get_all_records

Overview:
Return a map of PIF references to PIF records for all PIFs known to the system.

Signature:

((PIF ref -> PIF record) map) get_all_records (session_id s)

Return Type: (PIF ref -> PIF record) map
records of all objects
RPC name: get_uuid

Overview:
Get the uuid field of the given PIF.

Signature:

```plaintext
string get_uuid (session_id s, PIF ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get_device

Overview:
Get the device field of the given PIF.

Signature:

```plaintext
string get_device (session_id s, PIF ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get_network

Overview:
Get the network field of the given PIF.

Signature:

```plaintext
(network ref) get_network (session_id s, PIF ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: host ref

type 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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: int

value of the field
RPC name: get_VLAN

Overview:
Get the VLAN field of the given PIF.

Signature:

```c
int get_VLAN (session_id s, PIF ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: int
value of the field

RPC name: get_metrics

Overview:
Get the metrics field of the given PIF.

Signature:

```c
(PIF_metrics ref) get_metrics (session_id s, PIF ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: PIF_metrics ref
value of the field

RPC name: get_physical

Overview:
Get the physical field of the given PIF.

Signature:

```c
bool get_physical (session_id s, PIF ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

\[(\text{Bond ref}) \text{ get_bond_slave_of (session_id s, PIF ref self)}\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

\[(\text{Bond ref set}) \text{ get_bond_master_of (session_id s, PIF ref self)}\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

\[(\text{VLAN ref}) \text{ get_VLAN_master_of (session_id s, PIF ref self)}\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: VLAN ref
value of the field
2.26. CLASS: PIF

CHAPTER 2. API REFERENCE

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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 \rightarrow string) map value)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string \rightarrow string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

Return Type: void
RPC name: get_disallow_unplug

Overview:
Get the disallow_unplug field of the given PIF.

Signature:

\[
\text{bool get\_disallow\_unplug (session\_id s, PIF ref self)}
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: bool

value of the field

RPC name: set_disallow_unplug

Overview:
Set the disallow_unplug field of the given PIF.

Signature:

\[
\text{void set\_disallow\_unplug (session\_id s, PIF ref self, bool value)}
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>bool</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_tunnel_access_PIF_of

Overview:
Get the tunnel\_access\_PIF\_of field of the given PIF.

Signature:

\[
\text{(tunnel ref set) get\_tunnel\_access\_PIF\_of (session\_id s, PIF ref self)}
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string set
Value of the field
RPC name: get_ipv6_gateway

Overview:
Get the ipv6_gateway field of the given PIF.

Signature:

```plaintext
string get_ipv6_gateway (session_id s, PIF ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: primary_address_type

value of the field

RPC name: get_managed

Overview:
Get the managed field of the given PIF.

Signature:

```plaintext
bool get_managed (session_id s, PIF ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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

RPC name: get_capabilities

Overview:
Get the capabilities field of the given PIF.

Signature:

```
(string set) get_capabilities (session_id s, PIF ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string set
value of the field

RPC name: get_by_uuid

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

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: PIF record
all fields from the object
2.27 Class: PIF_metrics

2.27.1 Fields for class: PIF_metrics

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF_metrics</td>
<td>The metrics associated with a physical network interface.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Run</td>
<td>uuid</td>
<td>RO</td>
<td>run</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>Run</td>
<td>io_read_kbs</td>
<td>RO</td>
<td>io_read_kbs</td>
<td>float</td>
<td>Read bandwidth (KiB/s)</td>
</tr>
<tr>
<td>Run</td>
<td>io_write_kbs</td>
<td>RO</td>
<td>io_write_kbs</td>
<td>float</td>
<td>Write bandwidth (KiB/s)</td>
</tr>
<tr>
<td>Run</td>
<td>carrier</td>
<td>RO</td>
<td>carrier</td>
<td>bool</td>
<td>Report if the PIF got a carrier or not</td>
</tr>
<tr>
<td>Run</td>
<td>vendor_id</td>
<td>RO</td>
<td>vendor_id</td>
<td>string</td>
<td>Report vendor ID</td>
</tr>
<tr>
<td>Run</td>
<td>vendor_name</td>
<td>RO</td>
<td>vendor_name</td>
<td>string</td>
<td>Report vendor name</td>
</tr>
<tr>
<td>Run</td>
<td>device_id</td>
<td>RO</td>
<td>device_id</td>
<td>string</td>
<td>Report device ID</td>
</tr>
<tr>
<td>Run</td>
<td>device_name</td>
<td>RO</td>
<td>device_name</td>
<td>string</td>
<td>Report device name</td>
</tr>
<tr>
<td>Run</td>
<td>speed</td>
<td>RO</td>
<td>speed</td>
<td>int</td>
<td>Speed of the link (if available)</td>
</tr>
<tr>
<td>Run</td>
<td>duplex</td>
<td>RO</td>
<td>duplex</td>
<td>bool</td>
<td>Full duplex capability of the link (if available)</td>
</tr>
<tr>
<td>Run</td>
<td>pci_bus_path</td>
<td>RO</td>
<td>pci_bus_path</td>
<td>string</td>
<td>PCI bus path of the pif (if available)</td>
</tr>
<tr>
<td>Run</td>
<td>last_updated</td>
<td>RO</td>
<td>last_updated</td>
<td>datetime</td>
<td>Time at which this information was last updated</td>
</tr>
<tr>
<td></td>
<td>other_config</td>
<td>RW</td>
<td>other_config</td>
<td>(string (\rightarrow) string)</td>
<td>map</td>
</tr>
</tbody>
</table>

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 \(\rightarrow\) PIF_metrics record) map) get_all_records (session_id s)

Return Type: (PIF_metrics ref \(\rightarrow\) 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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: int

value of the field
RPC name: `get_duplex`

**Overview:**
Get the duplex field of the given PIF

**Signature:**

```c
bool get_duplex (session_id s, PIF_metrics ref self)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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

------------------

RPC name: `get_pci_bus_path`

**Overview:**
Get the pci_bus_path field of the given PIF

**Signature:**

```c
string get_pci_bus_path (session_id s, PIF_metrics ref self)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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

------------------

RPC name: `get_last_updated`

**Overview:**
Get the last_updated field of the given PIF

**Signature:**

```c
datetime get_last_updated (session_id s, PIF_metrics ref self)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: (string \(\rightarrow\) 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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string (\rightarrow) string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: PIF_metrics record
all fields from the object
2.28 Class: Bond

2.28.1 Fields for class: Bond

<table>
<thead>
<tr>
<th>Name</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO&lt;sub&gt;run&lt;/sub&gt;</td>
<td>uuid</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>RO&lt;sub&gt;ins&lt;/sub&gt;</td>
<td>master</td>
<td>PIF ref</td>
<td>The bonded interface</td>
</tr>
<tr>
<td>RO&lt;sub&gt;run&lt;/sub&gt;</td>
<td>slaves</td>
<td>PIF ref set</td>
<td>The interfaces which are part of this bond</td>
</tr>
<tr>
<td>RW</td>
<td>other_config</td>
<td>(string → string) map</td>
<td>additional configuration</td>
</tr>
<tr>
<td>RO&lt;sub&gt;run&lt;/sub&gt;</td>
<td>primary_slave</td>
<td>PIF ref</td>
<td>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</td>
</tr>
<tr>
<td>RO&lt;sub&gt;run&lt;/sub&gt;</td>
<td>mode</td>
<td>bond&lt;sub&gt;mode&lt;/sub&gt;</td>
<td>The algorithm used to distribute traffic among the bonded NICs</td>
</tr>
<tr>
<td>RO&lt;sub&gt;run&lt;/sub&gt;</td>
<td>properties</td>
<td>(string → string) map</td>
<td>Additional configuration properties specific to the bond mode.</td>
</tr>
<tr>
<td>RO&lt;sub&gt;run&lt;/sub&gt;</td>
<td>links_up</td>
<td>int</td>
<td>Number of links up in this bond</td>
</tr>
</tbody>
</table>

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<sub>mode</sub> mode, (string → string) map properties)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network ref</td>
<td>network</td>
<td>Network to add the bonded PIF to</td>
</tr>
<tr>
<td>PIF ref set</td>
<td>members</td>
<td>PIFs to add to this bond</td>
</tr>
<tr>
<td>string</td>
<td>MAC</td>
<td>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.</td>
</tr>
<tr>
<td>bond&lt;sub&gt;mode&lt;/sub&gt;</td>
<td>mode</td>
<td>Bonding mode to use for the new bond</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>properties</td>
<td>Additional configuration parameters specific to the bond mode</td>
</tr>
</tbody>
</table>

Return Type: Bond ref
The reference of the created Bond object

RPC name: destroy

Overview:
Destroy an interface bond.

**Signature:**

```c
void destroy (session_id s, Bond ref self)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bond ref</td>
<td>self</td>
<td>Bond to destroy</td>
</tr>
</tbody>
</table>

**Return Type:** void

### RPC name: `set_mode`

**Overview:**
Change the bond mode.

**Signature:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bond ref</td>
<td>self</td>
<td>The bond</td>
</tr>
<tr>
<td>bond_mode</td>
<td>value</td>
<td>The new bond mode</td>
</tr>
</tbody>
</table>

**Return Type:** void

### RPC name: `set_property`

**Overview:**
Set the value of a property of the bond.

**Signature:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bond ref</td>
<td>self</td>
<td>The bond</td>
</tr>
<tr>
<td>string</td>
<td>name</td>
<td>The property name</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>The property value</td>
</tr>
</tbody>
</table>

**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 \rightarrow Bond record) map) get\_all\_records (session\_id s)

Return Type: (Bond ref \rightarrow 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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bond ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bond ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: PIF ref

type 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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bond ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bond ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bond ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bond ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bond ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bond ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: PIF ref
value of the field

RPC name: get_mode

Overview:
Get the mode field of the given Bond.

Signature:

\[(bond\_mode) \text{get\_mode} (session\_id s, Bond ref self)\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bond ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: bond\_mode
value of the field

RPC name: get_properties

Overview:
Get the properties field of the given Bond.

Signature:

\[((\text{string} \rightarrow \text{string}) \text{map}) \text{get\_properties} (session\_id s, Bond ref self)\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bond ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: (\text{string} \rightarrow \text{string}) \text{map}
value of the field

RPC name: get_links_up

Overview:
Get the links\_up field of the given Bond.

Signature:

\[\text{int get\_links\_up} (session\_id s, Bond ref self)\]
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bond ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bond ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: Bond record

all fields from the object
2.29 Class: VLAN

2.29.1 Fields for class: VLAN

<table>
<thead>
<tr>
<th>Name</th>
<th>VLAN</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>A VLAN mux/demux.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO</td>
<td>run</td>
<td>uuid</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>RO</td>
<td>ins</td>
<td>PIF</td>
<td>Interface on which traffic is tagged</td>
</tr>
<tr>
<td>RO</td>
<td>run</td>
<td>untagged_PIF</td>
<td>Interface on which traffic is untagged</td>
</tr>
<tr>
<td>RO</td>
<td>ins</td>
<td>tag</td>
<td>VLAN tag in use</td>
</tr>
<tr>
<td>RW</td>
<td>other_config</td>
<td>(string → string) map</td>
<td>Additional configuration</td>
</tr>
</tbody>
</table>

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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>tagged_PIF</td>
<td>PIF which receives the tagged traffic</td>
</tr>
<tr>
<td>int</td>
<td>tag</td>
<td>VLAN tag to use</td>
</tr>
<tr>
<td>network ref</td>
<td>network</td>
<td>Network to receive the untagged traffic</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLAN ref</td>
<td>self</td>
<td>VLAN mux/demuxer to destroy</td>
</tr>
</tbody>
</table>

**Return Type:** void
RPC name: get_all

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

Signature:

(VLAN ref set) get_all (session_id s)

Return Type: VLAN ref set
references to all objects

RPC name: get_all_records

Overview:
Return a map of VLAN references to VLAN records for all VLANs known to the system.

Signature:

((VLAN ref -> VLAN record) map) get_all_records (session_id s)

Return Type: (VLAN ref -> VLAN record) map
records of all objects

RPC name: get_uuid

Overview:
Get the uuid field of the given VLAN.

Signature:

string get_uuid (session_id s, VLAN ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLAN ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLAN ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLAN ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLAN ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLAN ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: (string → string) map

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLAN ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLAN ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLAN ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLAN ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: VLAN record

all fields from the object
2.30 Class: SM

2.30.1 Fields for class: SM

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Quals</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM</td>
<td>A storage manager plugin.</td>
<td>RO</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>run</td>
<td>name_label</td>
<td>RO</td>
<td>string</td>
<td>a human-readable name</td>
</tr>
<tr>
<td>run</td>
<td>name_description</td>
<td>RO</td>
<td>string</td>
<td>a notes field containing human-readable description</td>
</tr>
<tr>
<td>run</td>
<td>type</td>
<td>RO</td>
<td>string</td>
<td>SR.type</td>
</tr>
<tr>
<td>run</td>
<td>vendor</td>
<td>RO</td>
<td>string</td>
<td>Vendor who created this plugin</td>
</tr>
<tr>
<td>run</td>
<td>copyright</td>
<td>RO</td>
<td>string</td>
<td>Entity which owns the copyright of this plugin</td>
</tr>
<tr>
<td>run</td>
<td>version</td>
<td>RO</td>
<td>string</td>
<td>Version of the plugin</td>
</tr>
<tr>
<td>run</td>
<td>required_api_version</td>
<td>RO</td>
<td>string</td>
<td>Minimum SM API version required on the server</td>
</tr>
<tr>
<td>run</td>
<td>configuration</td>
<td>RO</td>
<td>(string → string) map</td>
<td>names and descriptions of device config keys</td>
</tr>
<tr>
<td>run</td>
<td>capabilities</td>
<td>RO</td>
<td>string set</td>
<td>capabilities of the SM plugin</td>
</tr>
<tr>
<td>run</td>
<td>features</td>
<td>RO</td>
<td>(string → int) map</td>
<td>capabilities of the SM plugin, with capability version numbers</td>
</tr>
<tr>
<td>other_config</td>
<td></td>
<td>RW</td>
<td>(string → string) map</td>
<td>additional configuration</td>
</tr>
<tr>
<td>driver_filename</td>
<td></td>
<td>RO</td>
<td>string</td>
<td>filename of the storage driver</td>
</tr>
<tr>
<td>required_cluster_stack</td>
<td></td>
<td>RO</td>
<td>string set</td>
<td>The storage plugin requires that one of these cluster stacks is configured and running.</td>
</tr>
</tbody>
</table>

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 \text{ ref set}) \text{ 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 \text{ ref} \rightarrow SM \text{ record}) \text{ map} ) \text{ get\_all\_records (session\_id s)\}]

369
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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get_type

Overview:
Get the type field of the given SM.

Signature:

```plaintext
string get_type (session_id s, SM ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string

value of the field

RPC name: get_vendor

Overview:
Get the vendor field of the given SM.

Signature:

```plaintext
string get_vendor (session_id s, SM ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string

value of the field

RPC name: get_copyright

Overview:
Get the copyright field of the given SM.

Signature:

```plaintext
string get_copyright (session_id s, SM ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string

value of the field
RPC name: get_version

Overview:
Get the version field of the given SM.

Signature:

```java
string get_version (session_id s, SM ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```java
string get_required_api_version (session_id s, SM ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get_configuration

Overview:
Get the configuration field of the given SM.

Signature:

```java
((string -> string) map) get_configuration (session_id s, SM ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

\[(\text{string set}) \text{ get_capabilities} (\text{session_id } s, \text{ SM ref } self)\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string set
value of the field

RPC name: get_features

Overview:
Get the features field of the given SM.

Signature:

\[((\text{string -> int}) \text{ map}) \text{ get_features} (\text{session_id } s, \text{ SM ref } self)\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

\[((\text{string -> string}) \text{ map}) \text{ get_other_config} (\text{session_id } s, \text{ SM ref } self)\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:**

```c
void set_other_config (session_id s, SM ref self, (string -> string) map value)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

**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:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

**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:**

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

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get_required_cluster_stack

Overview:
Get the required_cluster_stack field of the given SM.

Signature:

```
(string set) get_required_cluster_stack (session_id s, SM ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string set
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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

(\text{SM record}) \text{get}_\text{record} (\text{session}_\text{id} \ s, \text{SM ref} \ self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: \text{SM record}
all fields from the object

RPC name: get_by_name_label

Overview:
Get all the SM instances with the given label.

Signature:

(\text{SM ref set}) \text{get}_\text{by}_\text{name}_\text{label} (\text{session}_\text{id} \ s, \text{string} \ label)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>label</td>
<td>label of object to return</td>
</tr>
</tbody>
</table>

Return Type: \text{SM ref set}
references to objects with matching names
2.31 Class: SR

2.31.1 Fields for class: SR

<table>
<thead>
<tr>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO</td>
<td>uuid</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>RO</td>
<td>name_label</td>
<td>string</td>
<td>a human-readable name</td>
</tr>
<tr>
<td>RO</td>
<td>name_description</td>
<td>string</td>
<td>a notes field containing human-readable description</td>
</tr>
<tr>
<td>RO</td>
<td>allowed_operations</td>
<td>storage_operations set</td>
<td>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.</td>
</tr>
<tr>
<td>RO</td>
<td>current_operations</td>
<td>(string → storage_operations) map</td>
<td>links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.</td>
</tr>
<tr>
<td>RO</td>
<td>VDIs</td>
<td>VDI ref set</td>
<td>all virtual disks known to this storage repository describes how particular hosts can see this storage repository</td>
</tr>
<tr>
<td>RO</td>
<td>PBDs</td>
<td>PBD ref set</td>
<td>sum of virtual_sizes of all VDIs in this storage repository (in bytes)</td>
</tr>
<tr>
<td>RO</td>
<td>virtual_allocation</td>
<td>int</td>
<td>physical space currently utilised on this storage repository (in bytes). Note that for sparse disk formats, physical_utilisation may be less than virtualallocation</td>
</tr>
<tr>
<td>RO</td>
<td>physical_utilisation</td>
<td>int</td>
<td>total physical size of the repository (in bytes)</td>
</tr>
<tr>
<td>RO</td>
<td>physical_size</td>
<td>int</td>
<td>total physical size of the repository (in bytes)</td>
</tr>
<tr>
<td>RO</td>
<td>type</td>
<td>string</td>
<td>type of the storage repository</td>
</tr>
<tr>
<td>RO</td>
<td>content_type</td>
<td>string</td>
<td>the type of the SR’s content, if required (e.g. ISOs)</td>
</tr>
<tr>
<td>RO</td>
<td>shared</td>
<td>bool</td>
<td>true if this SR is (capable of being) shared between multiple hosts</td>
</tr>
<tr>
<td>RW</td>
<td>other_config</td>
<td>(string → string) map</td>
<td>additional configuration</td>
</tr>
<tr>
<td>RW</td>
<td>tags</td>
<td>string set</td>
<td>user-specified tags for categorization purposes</td>
</tr>
<tr>
<td>RW</td>
<td>sm_config</td>
<td>(string → string) map</td>
<td>SM dependent data</td>
</tr>
<tr>
<td>RO</td>
<td>blobs</td>
<td>(string → blob ref) map</td>
<td>Binary blobs associated with this SR</td>
</tr>
<tr>
<td>RO</td>
<td>local_cache_enabled</td>
<td>bool</td>
<td>True if this SR is assigned to be the local cache for its host</td>
</tr>
<tr>
<td>RO</td>
<td>introduced_by</td>
<td>DR_task ref</td>
<td>The disaster recovery task which introduced this SR</td>
</tr>
<tr>
<td>RO</td>
<td>clustered</td>
<td>bool</td>
<td>True if the SR is using aggregated local storage</td>
</tr>
<tr>
<td>RO</td>
<td>is_tools_sr</td>
<td>bool</td>
<td>True if this is the SR that contains the Tools ISO VDIs</td>
</tr>
</tbody>
</table>
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, string type, string name, string description)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host to create/make the SR on</td>
</tr>
<tr>
<td>(string -&gt; string)</td>
<td>device_config</td>
<td>The device config string that will be passed to backend SR driver</td>
</tr>
<tr>
<td>int</td>
<td>physical_size</td>
<td>The physical size of the new storage repository</td>
</tr>
<tr>
<td>string</td>
<td>name_label</td>
<td>The name of the new storage repository</td>
</tr>
<tr>
<td>string</td>
<td>name_description</td>
<td>The description of the new storage repository</td>
</tr>
<tr>
<td>string</td>
<td>type</td>
<td>The type of the SR; used to specify the SR backend driver to use</td>
</tr>
<tr>
<td>string</td>
<td>content_type</td>
<td>The type of the new SRs content, if required (e.g. ISOS)</td>
</tr>
<tr>
<td>bool</td>
<td>shared</td>
<td>True if the SR (is capable of) being shared by multiple hosts</td>
</tr>
<tr>
<td>(string -&gt; string)</td>
<td>sm_config</td>
<td>Storage backend specific configuration options</td>
</tr>
</tbody>
</table>

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, string content_type, bool shared, (string -> string) map sm_config)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>The uuid assigned to the introduced SR</td>
</tr>
<tr>
<td>string</td>
<td>name_label</td>
<td>The name of the new storage repository</td>
</tr>
<tr>
<td>string</td>
<td>name_description</td>
<td>The description of the new storage repository</td>
</tr>
<tr>
<td>string</td>
<td>type</td>
<td>The type of the SR; used to specify the SR backend driver to use</td>
</tr>
<tr>
<td>string</td>
<td>content_type</td>
<td>The type of the new SRs content, if required (e.g. ISOS)</td>
</tr>
<tr>
<td>bool</td>
<td>shared</td>
<td>True if the SR (is capable of) being shared by multiple hosts</td>
</tr>
<tr>
<td>(string -&gt; string)</td>
<td>sm_config</td>
<td>Storage backend specific configuration options</td>
</tr>
</tbody>
</table>
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, string name, string description, string type, string content_type, (string -> string) map sm_config)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host to create/make the SR on</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>device_config</td>
<td>The device config string that will be passed to backend SR driver</td>
</tr>
<tr>
<td>int</td>
<td>physical_size</td>
<td>The physical size of the new storage repository</td>
</tr>
<tr>
<td>string</td>
<td>name_label</td>
<td>The name of the new storage repository</td>
</tr>
<tr>
<td>string</td>
<td>name_description</td>
<td>The description of the new storage repository</td>
</tr>
<tr>
<td>string</td>
<td>type</td>
<td>The type of the SR; used to specify the SR backend driver to use</td>
</tr>
<tr>
<td>string</td>
<td>content_type</td>
<td>The type of the new SRs content, if required (e.g. ISOs)</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>sm_config</td>
<td>Storage backend specific configuration options</td>
</tr>
</tbody>
</table>

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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>sr</td>
<td>The SR to destroy</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>sr</td>
<td>The SR to destroy</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>sr</td>
<td>The SR whose fields should be refreshed</td>
</tr>
</tbody>
</table>

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)
2.31. CLASS: SR

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>sr</td>
<td>The SR to scan</td>
</tr>
</tbody>
</table>

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:

```c
string probe (session_id s, host ref host, (string -> string) map device_config, string type, (string -> string) map sm_config)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ref</td>
<td>host</td>
<td>The host to create/make the SR on</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>device_config</td>
<td>The device config string that will be passed to backend SR driver</td>
</tr>
<tr>
<td>string</td>
<td>type</td>
<td>The type of the SR; used to specify the SR backend driver to use</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>sm_config</td>
<td>Storage backend specific configuration options</td>
</tr>
</tbody>
</table>

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:

```c
void set_shared (session_id s, SR ref sr, bool value)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>sr</td>
<td>The SR</td>
</tr>
<tr>
<td>bool</td>
<td>value</td>
<td>True if the SR is shared</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>sr</td>
<td>The SR</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>The name label for the SR</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>sr</td>
<td>The SR</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>The name description for the SR</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>sr</td>
<td>The SR</td>
</tr>
<tr>
<td>string</td>
<td>name</td>
<td>The name associated with the blob</td>
</tr>
<tr>
<td>string</td>
<td>mime_type</td>
<td>The mime type for the data. Empty string translates to application/octet-stream</td>
</tr>
<tr>
<td>bool</td>
<td>public</td>
<td>True if the blob should be publicly available</td>
</tr>
</tbody>
</table>

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:

```c
void set_physical_size (session_id s, SR ref self, int value)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>The SR to modify</td>
</tr>
<tr>
<td>int</td>
<td>value</td>
<td>The new value of the SR’s physical_size</td>
</tr>
</tbody>
</table>

Return Type: `void`

RPC name: `set_virtual_allocation`

Overview:
Sets the SR’s virtual_allocation field.

Signature:

```c
void set_virtual_allocation (session_id s, SR ref self, int value)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>The SR to modify</td>
</tr>
<tr>
<td>int</td>
<td>value</td>
<td>The new value of the SR’s virtual_allocation</td>
</tr>
</tbody>
</table>

Return Type: `void`

RPC name: `set_physical_utilisation`

Overview:
Sets the SR’s physical_utilisation field.

Signature:

```c
void set_physical_utilisation (session_id s, SR ref self, int value)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>The SR to modify</td>
</tr>
<tr>
<td>int</td>
<td>value</td>
<td>The new value of the SR’s physical_utilisation</td>
</tr>
</tbody>
</table>

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:

```c
void assert_can_host_ha_statefile (session_id s, SR ref sr)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>sr</td>
<td>The SR to query</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>sr</td>
<td>The SR to query</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: enable database replication

Overview:

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>sr</td>
<td>The SR to which metadata should be replicated</td>
</tr>
</tbody>
</table>

Return Type: void
RPC name: disable_database_replication

Overview:

Signature:

\[
\text{void disable_database_replication (session\_id s, SR ref sr)}
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>sr</td>
<td>The SR to which metadata should be no longer replicated</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_data_sources

Overview:

Signature:

\[
\text{(data\_source record set) get_data_sources (session\_id s, SR ref sr)}
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>sr</td>
<td>The SR to interrogate</td>
</tr>
</tbody>
</table>

Return Type: data_source record set

A set of data sources

RPC name: record_data_source

Overview:
Start recording the specified data source.

Signature:

\[
\text{void record_data_source (session\_id s, SR ref sr, string data\_source)}
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>sr</td>
<td>The SR</td>
</tr>
<tr>
<td>string</td>
<td>data_source</td>
<td>The data source to record</td>
</tr>
</tbody>
</table>

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, SR ref sr, string data_source)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>sr</td>
<td>The SR</td>
</tr>
<tr>
<td>string</td>
<td>data_source</td>
<td>The data source to query</td>
</tr>
</tbody>
</table>

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, SR ref sr, string data_source)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>sr</td>
<td>The SR</td>
</tr>
<tr>
<td>string</td>
<td>data_source</td>
<td>The data source whose archives are to be forgotten</td>
</tr>
</tbody>
</table>

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:

\[
((SR\ ref \rightarrow \ SR\ record)\ map)\ \text{get\_all\_records}\ (\text{session\_id}\ s)
\]

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:

\[
\text{string}\ \text{get\_uuid}\ (\text{session\_id}\ s, SR\ ref\ \text{self})
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get_name_label

Overview:
Get the name/label field of the given SR.

Signature:

\[
\text{string}\ \text{get\_name\_label}\ (\text{session\_id}\ s, SR\ ref\ \text{self})
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: (string → storage_operations) map
value of the field

RPC name: get_VDIs

Overview:
Get the VDIs field of the given SR.

Signature:

(VDI ref set) get_VDIs (session_id s, SR ref self)
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get_shared

Overview:
Get the shared field of the given SR.

Signature:

```c
bool get_shared (session_id s, SR ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: bool
value of the field

RPC name: get_other_config

Overview:
Get the other_config field of the given SR.

Signature:

```c
((string -> string) map) get_other_config (session_id s, SR ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```c
void set_other_config (session_id s, SR ref self, (string -> string) map value)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string set</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to add</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

```java
void remove_from_sm_config (session_id s, SR ref self, string key)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_blobs

Overview:
Get the blobs field of the given SR.

Signature:

```java
((string -> blob ref) map) get_blobs (session_id s, SR ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```java
bool get_local_cache_enabled (session_id s, SR ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: DR_task ref
value of the field

RPC name: get_clustered

Overview:
Get the clustered field of the given SR.

Signature:

bool get_clustered (session_id s, SR ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: bool
value of the field

RPC name: get_is_tools_sr

Overview:
Get the is_tools_sr field of the given SR.

Signature:

bool get_is_tools_sr (session_id s, SR ref self)
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: bool

value of the field

RPC name: get_by_uuid

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

Signature:

(SR ref) get_by_uuid (session_id s, string uuid)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>label</td>
<td>label of object to return</td>
</tr>
</tbody>
</table>

Return Type: SR ref set
references to objects with matching names
2.32 Class: LVHD

2.32.1 Fields for class: LVHD

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LVHD</td>
<td>LVHD SR specific operations.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO</td>
<td>uuid</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
</tbody>
</table>

2.32.2 RPCs associated with class: LVHD

RPC name: enable_thin_provisioning

Overview:
Upgrades an LVHD SR to enable thin-provisioning. Future VDIs created in this SR will be thinly-provisioned, although existing VDIs will be left alone. Note that the SR must be attached to the SRmaster for upgrade to work.

Signature:

```c
void enable_thin_provisioning (session_id s, SR ref SR, int initial_allocation, int allocation_quantum)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR ref</td>
<td>SR</td>
<td>The LVHD SR to upgrade to being thinly-provisioned.</td>
</tr>
<tr>
<td>int</td>
<td>initial_allocation</td>
<td>The initial amount of space to allocate to a newly-created VDI in bytes</td>
</tr>
<tr>
<td>int</td>
<td>allocation_quantum</td>
<td>The amount of space to allocate to a VDI when it needs to be enlarged in bytes</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_uuid

Overview:
Get the uuid field of the given LVHD.

Signature:

```c
string get_uuid (session_id s, LVHD ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LVHD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string

value of the field
RPC name: get_by_uuid

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

Signature:
(LVHD ref) get_by_uuid (session_id s, string uuid)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

Return Type: LVHD ref
reference to the object

RPC name: get_record

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

Signature:
(LVHD record) get_record (session_id s, LVHD ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LVHD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: LVHD record
all fields from the object
## 2.33 Class: VDI

### 2.33.1 Fields for class: VDI

<table>
<thead>
<tr>
<th>Name</th>
<th>VDI</th>
<th>Description</th>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VDI</td>
<td>A virtual disk image.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RO&lt;sub&gt;run&lt;/sub&gt;</td>
<td>uuid</td>
<td>string</td>
<td>RO</td>
<td>run</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>RO&lt;sub&gt;ins&lt;/sub&gt;</td>
<td>name_label</td>
<td>string</td>
<td>RO</td>
<td>name</td>
<td>string</td>
<td>a human-readable name</td>
</tr>
<tr>
<td>RO&lt;sub&gt;ins&lt;/sub&gt;</td>
<td>description</td>
<td>string</td>
<td>RO</td>
<td>description</td>
<td>string</td>
<td>a notes field containing human-readable description</td>
</tr>
<tr>
<td>RO&lt;sub&gt;run&lt;/sub&gt;</td>
<td>allowed_operations</td>
<td>vdi_operations set</td>
<td>RO</td>
<td>run</td>
<td>vdi_operations set</td>
<td>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.</td>
</tr>
<tr>
<td>RO&lt;sub&gt;run&lt;/sub&gt;</td>
<td>current_operations</td>
<td>(string → vdi_operations) map</td>
<td>RO</td>
<td>run</td>
<td>(string → vdi_operations) map</td>
<td>links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.</td>
</tr>
<tr>
<td>RO&lt;sub&gt;ins&lt;/sub&gt;</td>
<td>SR</td>
<td>SR ref</td>
<td>RO</td>
<td>VBDs</td>
<td>VBD ref set</td>
<td>list of vbds that refer to this disk</td>
</tr>
<tr>
<td>RO&lt;sub&gt;run&lt;/sub&gt;</td>
<td>crash.dumps</td>
<td>crashdump ref set</td>
<td>RO</td>
<td>crash.dumps</td>
<td>crashdump ref set</td>
<td>list of crash dumps that refer to this disk size of disk as presented to the guest (in bytes). Note that, depending on storage backend type, requested size may not be respected exactly</td>
</tr>
<tr>
<td>RO&lt;sub&gt;ins&lt;/sub&gt;</td>
<td>virtual_size</td>
<td>int</td>
<td>RO</td>
<td>run</td>
<td>int</td>
<td>amount of physical space that the disk image is currently taking up on the storage repository (in bytes)</td>
</tr>
<tr>
<td>RO&lt;sub&gt;run&lt;/sub&gt;</td>
<td>physical_utilisation</td>
<td>int</td>
<td>RO</td>
<td>run</td>
<td>int</td>
<td>amount of physical space that the disk image is currently taking up on the storage repository (in bytes)</td>
</tr>
<tr>
<td>RO&lt;sub&gt;ins&lt;/sub&gt;</td>
<td>type</td>
<td>vdi_type</td>
<td>RO</td>
<td>type</td>
<td>vdi_type</td>
<td>type of the VDI</td>
</tr>
<tr>
<td>RO&lt;sub&gt;ins&lt;/sub&gt;</td>
<td>sharable</td>
<td>bool</td>
<td>RO</td>
<td>sharable</td>
<td>bool</td>
<td>true if this disk may be shared</td>
</tr>
<tr>
<td>RO&lt;sub&gt;ins&lt;/sub&gt;</td>
<td>read_only</td>
<td>bool</td>
<td>RO</td>
<td>read_only</td>
<td>bool</td>
<td>true if this disk may ONLY be mounted read-only</td>
</tr>
<tr>
<td>RW</td>
<td>other_config</td>
<td>(string → string) map</td>
<td>RW</td>
<td>storage_lock</td>
<td>bool</td>
<td>true if this disk is locked at the storage level</td>
</tr>
<tr>
<td>RO&lt;sub&gt;run&lt;/sub&gt;</td>
<td>location</td>
<td>string</td>
<td>RO</td>
<td>location</td>
<td>string</td>
<td>location information</td>
</tr>
<tr>
<td>RO&lt;sub&gt;run&lt;/sub&gt;</td>
<td>managed</td>
<td>bool</td>
<td>RO</td>
<td>managed</td>
<td>bool</td>
<td>true if SR scan operation reported this VDI as not present on disk</td>
</tr>
<tr>
<td>RO&lt;sub&gt;run&lt;/sub&gt;</td>
<td>missing</td>
<td>bool</td>
<td>RO</td>
<td>missing</td>
<td>bool</td>
<td>References the parent disk, if this VDI is part of a chain data to be inserted into the xenstore tree (/local/domain/0/backend/vbd/&lt;domid&gt;/device-id/vdi-data) after the VDI is attached. This is generally set by the SM backends on vdi_attach.</td>
</tr>
<tr>
<td>RO&lt;sub&gt;run&lt;/sub&gt;</td>
<td>parent</td>
<td>VDI ref</td>
<td>RO</td>
<td>parent</td>
<td>VDI ref</td>
<td>References the parent disk, if this VDI is part of a chain data to be inserted into the xenstore tree (/local/domain/0/backend/vbd/&lt;domid&gt;/device-id/vdi-data) after the VDI is attached. This is generally set by the SM backends on vdi_attach.</td>
</tr>
<tr>
<td>RW</td>
<td>xenstore_data</td>
<td>(string → string) map</td>
<td>RW</td>
<td>xenstore_data</td>
<td>(string → string) map</td>
<td>data to be inserted into the xenstore tree (/local/domain/0/backend/vbd/&lt;domid&gt;/device-id/vdi-data) after the VDI is attached. This is generally set by the SM backends on vdi_attach.</td>
</tr>
<tr>
<td>RW</td>
<td>sm_config</td>
<td>(string → string) map</td>
<td>RW</td>
<td>is snapshot</td>
<td>bool</td>
<td>SM dependent data</td>
</tr>
<tr>
<td>RO&lt;sub&gt;run&lt;/sub&gt;</td>
<td>is snapshot</td>
<td>bool</td>
<td>RO</td>
<td>snapshot</td>
<td>VDI ref</td>
<td>true if this is a snapshot. Ref pointing to the VDI this snapshot is of.</td>
</tr>
</tbody>
</table>

401
2.33. CLASS: VDI

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>snapshots</td>
<td>VDI ref set</td>
</tr>
<tr>
<td>snapshot_time</td>
<td>datetime</td>
</tr>
<tr>
<td>tags</td>
<td>string set</td>
</tr>
<tr>
<td>allow_caching</td>
<td>bool</td>
</tr>
<tr>
<td>on_boot</td>
<td>on_boot</td>
</tr>
<tr>
<td>metadata_of_pool</td>
<td>pool ref</td>
</tr>
<tr>
<td>metadata_latest</td>
<td>bool</td>
</tr>
<tr>
<td>is_tools_iso</td>
<td>bool</td>
</tr>
</tbody>
</table>

List pointing to all the VDIs snapshots.
Date/time when this snapshot was created.
user-specified tags for categorization purposes.
true if this VDI is to be cached in the local cache SR.
The behaviour of this VDI on a VM boot.
The pool whose metadata is contained in this VDI.
Whether this VDI contains the latest known accessible metadata for the pool.
Whether this VDI is a Tools ISO.

2.33.2 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:

安く(snapshot (session_id s, VDI ref vdi, (string -> string) map driver_params))

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>vdi</td>
<td>The VDI to snapshot</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>driver_params</td>
<td>Optional parameters that can be passed through to backend driver in order to specify storage-type-specific snapshot options</td>
</tr>
</tbody>
</table>

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:

安く(clone (session_id s, VDI ref vdi, (string -> string) map driver_params))

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>vdi</td>
<td>The VDI to clone</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>driver_params</td>
<td>Optional parameters that are passed through to the backend driver in order to specify storage-type-specific clone options</td>
</tr>
</tbody>
</table>

402
Return Type: `VDI ref`
The ID of the newly created VDI.

RPC name: `resize`

**Overview:**
Resize the VDI.

**Signature:**

```c
void resize (session_id s, VDI ref vdi, int size)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>vdi</td>
<td>The VDI to resize</td>
</tr>
<tr>
<td>int</td>
<td>size</td>
<td>The new size of the VDI</td>
</tr>
</tbody>
</table>

Return Type: `void`

RPC name: `resize_online`

**Overview:**
Resize the VDI which may or may not be attached to running guests.

**Signature:**

```c
void resize_online (session_id s, VDI ref vdi, int size)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>vdi</td>
<td>The VDI to resize</td>
</tr>
<tr>
<td>int</td>
<td>size</td>
<td>The new size of the VDI</td>
</tr>
</tbody>
</table>

Return Type: `void`

RPC name: `introduce`

**Overview:**
Create a new VDI record in the database only.

**Signature:**

```c
(VDI ref) introduce (session_id s, string uuid, string name_label, string name_description, SR ref SR, vdi_type
```

**Arguments:**
<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>The uuid of the disk to introduce</td>
</tr>
<tr>
<td>string</td>
<td>name_label</td>
<td>The name of the disk record</td>
</tr>
<tr>
<td>string</td>
<td>name_description</td>
<td>The description of the disk record</td>
</tr>
<tr>
<td>SR ref</td>
<td>SR</td>
<td>The SR that the VDI is in</td>
</tr>
<tr>
<td>vdi_type</td>
<td>type</td>
<td>The type of the VDI</td>
</tr>
<tr>
<td>bool</td>
<td>sharable</td>
<td>true if this disk may be shared</td>
</tr>
<tr>
<td>bool</td>
<td>read_only</td>
<td>true if this disk may ONLY be mounted read-only</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>other_config</td>
<td>additional configuration</td>
</tr>
<tr>
<td>string</td>
<td>location</td>
<td>location information</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>xenstore_data</td>
<td>Data to insert into xenstore</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>sm_config</td>
<td>Storage-specific config</td>
</tr>
<tr>
<td>bool</td>
<td>managed</td>
<td>Storage-specific config</td>
</tr>
<tr>
<td>int</td>
<td>virtual_size</td>
<td>Storage-specific config</td>
</tr>
<tr>
<td>int</td>
<td>physical_utilisation</td>
<td>Storage-specific config</td>
</tr>
<tr>
<td>pool ref</td>
<td>metadata_of_pool</td>
<td>Storage-specific config</td>
</tr>
<tr>
<td>bool</td>
<td>is_a_snapshot</td>
<td>Storage-specific config</td>
</tr>
<tr>
<td>datetime</td>
<td>snapshot_time</td>
<td>Storage-specific config</td>
</tr>
<tr>
<td>VDI ref</td>
<td>snapshot_of</td>
<td>Storage-specific config</td>
</tr>
</tbody>
</table>

**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 SR,  

**Arguments:**
<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>The uuid of the disk to introduce</td>
</tr>
<tr>
<td>string</td>
<td>name</td>
<td>The name of the disk record</td>
</tr>
<tr>
<td>string</td>
<td>description</td>
<td>The description of the disk record</td>
</tr>
<tr>
<td>SR ref</td>
<td>type</td>
<td>The SR that the VDI is in</td>
</tr>
<tr>
<td>vdi type</td>
<td>sharable</td>
<td>The type of the VDI</td>
</tr>
<tr>
<td>bool</td>
<td>iro</td>
<td>true if this disk may be shared</td>
</tr>
<tr>
<td>bool</td>
<td>read-only</td>
<td>true if this disk may ONLY be mounted read-only</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>other_config</td>
<td>additional configuration</td>
</tr>
<tr>
<td>string</td>
<td>location</td>
<td>location information</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>xenstore_data</td>
<td>Data to insert into xenstore</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>sm_config</td>
<td>Storage-specific config</td>
</tr>
<tr>
<td>bool</td>
<td>managed</td>
<td>Storage-specific config</td>
</tr>
<tr>
<td>int</td>
<td>virtual_size</td>
<td>Storage-specific config</td>
</tr>
<tr>
<td>int</td>
<td>physical_utilisation</td>
<td>Storage-specific config</td>
</tr>
<tr>
<td>pool ref</td>
<td>metadata_of_pool</td>
<td>Storage-specific config</td>
</tr>
<tr>
<td>bool</td>
<td>is_a_snapshot</td>
<td>Storage-specific config</td>
</tr>
<tr>
<td>datetime</td>
<td>snapshot_time</td>
<td>Storage-specific config</td>
</tr>
<tr>
<td>VDI ref</td>
<td>snapshot_of</td>
<td>Storage-specific config</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>vdi</td>
<td>The VDI to forget about</td>
</tr>
</tbody>
</table>

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:
## 2.3.3. CLASS: VDI

### CHAPTER 2. API REFERENCE

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>vdi</td>
<td>The VDI whose stats (eg size) should be updated</td>
</tr>
</tbody>
</table>

**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:**

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

**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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>The VDI to modify</td>
</tr>
<tr>
<td>bool</td>
<td>value</td>
<td>The new value of the VDI’s managed field</td>
</tr>
</tbody>
</table>

**Return Type:** `void`
RPC name: forget

Overview:
Removes a VDI record from the database.

Signature:

void forget (session_id s, VDI ref vdi)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>vdi</td>
<td>The VDI to forget about</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>The VDI to modify</td>
</tr>
<tr>
<td>bool</td>
<td>value</td>
<td>The new value of the VDI’s sharable field</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>The VDI to modify</td>
</tr>
<tr>
<td>bool</td>
<td>value</td>
<td>The new value of the VDI’s read-only field</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>The VDI to modify</td>
</tr>
<tr>
<td>bool</td>
<td>value</td>
<td>The new value of the VDI's missing field</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>The VDI to modify</td>
</tr>
<tr>
<td>int</td>
<td>value</td>
<td>The new value of the VDI's virtual size</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>The VDI to modify</td>
</tr>
<tr>
<td>int</td>
<td>value</td>
<td>The new value of the VDI's physical utilisation</td>
</tr>
</tbody>
</table>

Return Type: void
RPC name: set_is_a_snapshot

Overview:
Sets whether this VDI is a snapshot.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>The VDI to modify</td>
</tr>
<tr>
<td>bool</td>
<td>value</td>
<td>The new value indicating whether this VDI is a snapshot</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: set_snapshot_of

Overview:
Sets the VDI of which this VDI is a snapshot.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>The VDI to modify</td>
</tr>
<tr>
<td>VDI ref</td>
<td>value</td>
<td>The VDI of which this VDI is a snapshot</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: set_snapshot_time

Overview:
Sets the snapshot time of this VDI.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>The VDI to modify</td>
</tr>
<tr>
<td>datetime</td>
<td>value</td>
<td>The snapshot time of this VDI.</td>
</tr>
</tbody>
</table>

Return Type: void
RPC name: set_metadata_of_pool

Overview:
Records the pool whose metadata is contained by this VDI.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>The VDI to modify</td>
</tr>
<tr>
<td>pool ref</td>
<td>value</td>
<td>The pool whose metadata is contained by this VDI</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>The VDI to modify</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>The name lable for the VDI</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>The VDI to modify</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>The name description for the VDI</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>The VDI to modify</td>
</tr>
<tr>
<td>on_boot</td>
<td>value</td>
<td>The value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>The VDI to modify</td>
</tr>
<tr>
<td>bool</td>
<td>value</td>
<td>The value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>The VDI which contains the database to open</td>
</tr>
</tbody>
</table>

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:** `VDI ref`
- **name:** `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:** `VDI ref`
- **name:** `vdi`
  The VDI to migrate
- **type:** `SR ref`
- **name:** `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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get_name_description

Overview:
Get the name/description field of the given VDI.

Signature:
2.33. CLASS: VDI

string get_name_description (session_id s, VDI ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: int

value of the field

RPC name: get_physical_utilisation

Overview:
Get the physical_utilisation field of the given VDI.

Signature:

```cpp
int get_physical_utilisation (session_id s, VDI ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: int

value of the field

RPC name: get_type

Overview:
Get the type field of the given VDI.

Signature:

```cpp
(vdi_type) get_type (session_id s, VDI ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: vdi_type

value of the field

RPC name: get_sharable

Overview:
Get the sharable field of the given VDI.

Signature:

```cpp
bool get_sharable (session_id s, VDI ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_storage_lock

Overview:
Get the storage_lock field of the given VDI.

Signature:

```c
bool get_storage_lock (session_id s, VDI ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

```c
void add_to_xenstore_data (session_id s, VDI ref self, string key, string value)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

```c
void remove_from_xenstore_data (session_id s, VDI ref self, string key)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_sm_config

Overview:
Get the sm_config field of the given VDI.

Signature:

```c
((string -> string) map) get_sm_config (session_id s, VDI ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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)
2.3.3. CLASS: VDI

CHAPTER 2. API REFERENCE

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string set</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to add</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to remove</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_allow_caching

Overview:
Get the allow_caching field of the given VDI.

Signature:

```plaintext
bool get_allow_caching (session_id s, VDI ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: bool
value of the field

RPC name: get_is_tools_iso

Overview:
Get the is_tools_iso field of the given VDI.

Signature:

bool get_is_tools_iso (session_id s, VDI ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI record</td>
<td>args</td>
<td>All constructor arguments</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>label</td>
<td>label of object to return</td>
</tr>
</tbody>
</table>

Return Type: `VDI ref set`

references to objects with matching names
2.34 Class: VBD

2.34.1 Fields for class: VBD

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RO_run</strong> uuid</td>
<td>A virtual block device.</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td><strong>RO_run</strong> allowed_operations</td>
<td></td>
<td>vbd_operations set</td>
<td>list of the operations allowed in this state.</td>
</tr>
<tr>
<td><strong>RO_run</strong> current_operations</td>
<td></td>
<td>(string → vbd_operations) map</td>
<td>This list is advisory only and the server state may have changed by the time this field is read by a client. This list links each of the running tasks using this object (by reference) to a current_operation enum which describes the nature of the task.</td>
</tr>
<tr>
<td><strong>RO_ins</strong> VM</td>
<td>VM ref</td>
<td>VM ref</td>
<td>the virtual machine</td>
</tr>
<tr>
<td><strong>RO_ins</strong> VDI</td>
<td>VDI ref</td>
<td></td>
<td>the virtual disk</td>
</tr>
<tr>
<td><strong>RO_run</strong> device</td>
<td>string</td>
<td></td>
<td>device seen by the guest e.g. hda1</td>
</tr>
<tr>
<td><strong>RW</strong> userdevice</td>
<td>string</td>
<td></td>
<td>user-friendly device name e.g. 0,1,2,etc.</td>
</tr>
<tr>
<td><strong>RW</strong> bootable</td>
<td>bool</td>
<td>vbd_mode</td>
<td>true if this VBD is bootable</td>
</tr>
<tr>
<td><strong>RW</strong> type</td>
<td>vbd_type</td>
<td></td>
<td>the mode the VBD should be mounted with</td>
</tr>
<tr>
<td><strong>RW</strong> unpluggable</td>
<td>bool</td>
<td></td>
<td>how the VBD will appear to the guest (e.g. disk or CD)</td>
</tr>
<tr>
<td><strong>RO_run</strong> storage_lock</td>
<td></td>
<td>bool</td>
<td>true if this VBD will support hot-unplug</td>
</tr>
<tr>
<td><strong>RO_ins</strong> empty</td>
<td>bool</td>
<td></td>
<td>true if a storage level lock was acquired</td>
</tr>
<tr>
<td><strong>RW</strong> other_config</td>
<td>(string → string) map</td>
<td></td>
<td>if true this represents an empty drive</td>
</tr>
<tr>
<td><strong>RO_run</strong> currently_attached</td>
<td></td>
<td>bool</td>
<td>additional configuration</td>
</tr>
<tr>
<td><strong>RO_run</strong> status_code</td>
<td></td>
<td>int</td>
<td>is the device currently attached (erased on reboot)</td>
</tr>
<tr>
<td><strong>RO_run</strong> status_detail</td>
<td></td>
<td>string</td>
<td>error/success code associated with last attach-operation (erased on reboot)</td>
</tr>
<tr>
<td><strong>RO_run</strong> runtime_properties</td>
<td></td>
<td>(string → string) map</td>
<td>error/success information associated with last attach-operation status (erased on reboot)</td>
</tr>
<tr>
<td><strong>RW</strong> qos_algorithm_type</td>
<td></td>
<td>string</td>
<td>Device runtime properties</td>
</tr>
<tr>
<td><strong>RW</strong> qos_algorithm_params</td>
<td></td>
<td>(string → string) map</td>
<td>QoS algorithm to use</td>
</tr>
<tr>
<td><strong>RO_run</strong> qos_supported_algorithms</td>
<td></td>
<td>string set</td>
<td>parameters for chosen QoS algorithm</td>
</tr>
<tr>
<td><strong>RO_run</strong> metrics</td>
<td>VBD_metrics ref</td>
<td></td>
<td>supported QoS algorithms for this VBD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>metrics associated with this VBD</td>
</tr>
</tbody>
</table>

2.34.2 RPCs associated with class: VBD

RPC name: eject

**Overview:**
Remove the media from the device and leave it empty.

**Signature:**

```c
void eject (session_id s, VBD ref vbd)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>vbd</td>
<td>The vbd representing the CDROM-like device</td>
</tr>
</tbody>
</table>

Return Type: void

Possible Error Codes: VBD_NOT_REMOVABLE_MEDIA, VBD_IS_EMPTY

RPC name: insert

Overview:
Insert new media into the device.

Signature:

```c
void insert (session_id s, VBD ref vbd, VDI ref vdi)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>vbd</td>
<td>The vbd representing the CDROM-like device</td>
</tr>
<tr>
<td>VDI ref</td>
<td>vdi</td>
<td>The new VDI to 'insert'</td>
</tr>
</tbody>
</table>

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:

```c
void plug (session_id s, VBD ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>The VBD to hotplug</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: unplug

Overview:
Hot-unplug the specified VBD, dynamically unattaching it from the running VM.

Signature:

```c
void unplug (session_id s, VBD ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>The VBD to hot-unplug</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>The VBD to forcibly unplug</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: assertAttachable

Overview:
Throws an error if this VBD could not be attached to this VM if the VM were running. Intended for debugging.

Signature:
void assertAttachable (session_id s, VBD ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>The VBD to query</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: GetAll

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

Signature:
(VBD ref set) GetAll (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:

\[ ((\text{VBD ref} \rightarrow \text{VBD record}) \map) \ \text{get\_all\_records} \ (\text{session\_id s}) \]

Return Type: \((\text{VBD ref} \rightarrow \text{VBD record}) \map\)
records of all objects

RPC name: get_uuid

Overview:
Get the uuid field of the given VBD.

Signature:

\[ \text{string} \ \text{get\_uuid} \ (\text{session\_id s, VBD ref self}) \]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: \text{string}
value of the field

RPC name: get_allowed_operations

Overview:
Get the allowed_operations field of the given VBD.

Signature:

\[ (\text{vbd\_operations set}) \ \text{get\_allowed\_operations} \ (\text{session\_id s, VBD ref self}) \]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: \text{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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: (string → vbd_operations) map
value of the field

RPC name: get_VM

Overview:
Get the VM field of the given VBD.

Signature:

(VM ref) get_VM (session_id s, VBD ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>bool</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>vbd_mode</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>vbd_type</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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)
```
2.34. CLASS: VBD

CHAPTER 2. API REFERENCE

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>bool</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_currently_attached

Overview:
Get the currently attached field of the given VBD.

Signature:

```c
bool get_currently_attached (session_id s, VBD ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: bool

RPC name: get_status_code

Overview:
Get the status code field of the given VBD.

Signature:

```c
int get_status_code (session_id s, VBD ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```c
void set_qos_algorithm_type (session_id s, VBD ref self, string value)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_qos_algorithm_params

Overview:
Get the qos/algorithm_params field of the given VBD.

Signature:

```c
((string -> string) map) get_qos_algorithm_params (session_id s, VBD ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```c
void set_qos_algorithm_params (session_id s, VBD ref self, (string -> string) map value)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

```java
void add_to_qos_algorithm_params (session_id s, VBD ref self, string key, string value)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

```java
void remove_from_qos_algorithm_params (session_id s, VBD ref self, string key)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: `get_qos_supported_algorithms`

Overview:
Get the qos/supported_algorithms field of the given VBD.

Signature:

```java
(string set) get_qos_supported_algorithms (session_id s, VBD ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD record</td>
<td>args</td>
<td>All constructor arguments</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: void

**RPC name: get_by_uuid**

**Overview:**
Get a reference to the VBD instance with the specified UUID.

**Signature:**

```cpp
(VBD ref) get_by_uuid (session_id s, string uuid)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:**

```cpp
(VBD record) get_record (session_id s, VBD ref self)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: VBD record
all fields from the object
2.35 Class: VBD\_metrics

2.35.1 Fields for class: VBD\_metrics

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>uuid</em></td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td><em>io_read_kbs</em></td>
<td>float</td>
<td>Read bandwidth (KiB/s)</td>
</tr>
<tr>
<td><em>io_write_kbs</em></td>
<td>float</td>
<td>Write bandwidth (KiB/s)</td>
</tr>
<tr>
<td><em>last_updated</em></td>
<td>datetime</td>
<td>Time at which this information was last updated</td>
</tr>
</tbody>
</table>

2.35.2 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:**

```plaintext
(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:**

```plaintext
((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:**

```plaintext
string get\_uuid (session_id s, VBD\_metrics ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: datetime

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: (string → string) map

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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)
```
2.35. CLASS: VBD_METRICS  

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD_metrics ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: VBD\_metrics record

all fields from the object
2.36 Class: PBD

2.36.1 Fields for class: PBD

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBD</td>
<td>The physical block devices through which hosts access SRs.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO</td>
<td>run</td>
<td>uuid</td>
<td>string unique identifier/object reference</td>
</tr>
<tr>
<td>RO</td>
<td>ins</td>
<td>host</td>
<td>host ref physical machine on which the pbd is available</td>
</tr>
<tr>
<td>RO</td>
<td>ins</td>
<td>SR</td>
<td>SR ref the storage repository that the pbd realises</td>
</tr>
<tr>
<td>RO</td>
<td>ins</td>
<td>device_config</td>
<td>(string → string) map a config string to string map that is provided to the host’s SR-backend-driver</td>
</tr>
<tr>
<td>RO</td>
<td>run</td>
<td>currently_attached</td>
<td>bool is the SR currently attached on this host?</td>
</tr>
<tr>
<td>RW</td>
<td>other_config</td>
<td>(string → string) map additional configuration</td>
<td></td>
</tr>
</tbody>
</table>

2.36.2 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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBD ref</td>
<td>self</td>
<td>The PBD to activate</td>
</tr>
</tbody>
</table>

Return Type: void

Possible Error Codes: SR unknowndriver

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBD ref</td>
<td>self</td>
<td>The PBD to deactivate</td>
</tr>
</tbody>
</table>

Return Type: void
RPC name: set\_device\_config

Overview:
Sets the PBD’s device\_config field.

Signature:

\[
\text{void set\_device\_config (session\_id s, PBD ref self, (string -> string) map value)}
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBD ref</td>
<td>self</td>
<td>The PBD to modify</td>
</tr>
<tr>
<td>(string -&gt; string)</td>
<td>map value</td>
<td>The new value of the PBD’s device_config</td>
</tr>
</tbody>
</table>

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:

\[
\text{string get\_uuid (session\_id s, PBD ref self)}
\]
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```c
bool get_currently_attached (session_id s, PBD ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: bool
value of the field

RPC name: get_other_config
Overview:
Get the other_config field of the given PBD.
Signature:

```c
((string -> string) map) get_other_config (session_id s, PBD ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```c
void set_other_config (session_id s, PBD ref self, (string -> string) map value)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: create

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

Signature:

```c
(PBD ref) create (session_id s, PBD record args)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBD record</td>
<td>args</td>
<td>All constructor arguments</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBD ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: PBD record

all fields from the object
2.37 Class: crashdump

2.37.1 Fields for class: crashdump

<table>
<thead>
<tr>
<th>Name</th>
<th>crashdump</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>A VM crashdump.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO</td>
<td>uuid</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>RO</td>
<td>VM</td>
<td>VM ref</td>
<td>the virtual machine</td>
</tr>
<tr>
<td>RO</td>
<td>VDI</td>
<td>VDI ref</td>
<td>the virtual disk</td>
</tr>
<tr>
<td>RW</td>
<td>other_config</td>
<td>(string → string) map</td>
<td>additional configuration</td>
</tr>
</tbody>
</table>

2.37.2 RPCs associated with class: crashdump

RPC name: destroy

Overview:
Destroy the specified crashdump.

Signature:

void destroy (session_id s, crashdump ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>crashdump ref</td>
<td>self</td>
<td>The crashdump to destroy</td>
</tr>
</tbody>
</table>

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: \((\text{crashdump ref} \rightarrow \text{crashdump record})\) map
records of all objects

RPC name: get\_uuid

Overview:
Get the uuid field of the given crashdump.

Signature:

\[
\text{string get\_uuid (session\_id s, crashdump ref self)}
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>crashdump ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get\_VM

Overview:
Get the VM field of the given crashdump.

Signature:

\[
(\text{VM ref}) \text{get\_VM (session\_id s, crashdump ref self)}
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>crashdump ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: VM ref
value of the field

RPC name: get\_VDI

Overview:
Get the VDI field of the given crashdump.

Signature:

\[
(\text{VDI ref}) \text{get\_VDI (session\_id s, crashdump ref self)}
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>crashdump ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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 \rightarrow string) map) get\_other\_config (session_id s, crashdump ref self)```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>crashdump ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**Return Type:** (string \rightarrow 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 \rightarrow string) map value)```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>crashdump ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string \rightarrow string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>crashdump ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>crashdump ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>crashdump ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: crashdump record
all fields from the object
2.38 Class: VTPM

2.38.1 Fields for class: VTPM

<table>
<thead>
<tr>
<th>Name</th>
<th>VTPM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>A virtual TPM device.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO</td>
<td>uuid</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>RO</td>
<td>ins</td>
<td>VM</td>
<td>the virtual machine</td>
</tr>
<tr>
<td>RO</td>
<td>ins</td>
<td>backend</td>
<td>the domain where the backend is located</td>
</tr>
</tbody>
</table>

2.38.2 RPCs associated with class: VTPM

RPC name: get_uid

Overview:
Get the uuid field of the given VTPM.

Signature:

\[
\text{string get_uuid (session_id s, VTPM ref self)}
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTPM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get_VM

Overview:
Get the VM field of the given VTPM.

Signature:

\[
(\text{VM ref}) \text{ get_VM (session_id s, VTPM ref self)}
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTPM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTPM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTPM record</td>
<td>args</td>
<td>All constructor arguments</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTPM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTPM ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: VTPM record
all fields from the object
2.39 Class: console

2.39.1 Fields for class: console

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>console</td>
<td>A console.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO</td>
<td>run uuid</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>RO</td>
<td>run protocol</td>
<td>console_protocol</td>
<td>the protocol used by this console</td>
</tr>
<tr>
<td>RO</td>
<td>run location</td>
<td>string</td>
<td>URI for the console service</td>
</tr>
<tr>
<td>RO</td>
<td>run VM</td>
<td>VM ref</td>
<td>VM to which this console is attached</td>
</tr>
<tr>
<td>RW</td>
<td>other config</td>
<td>(string → string) map</td>
<td>additional configuration</td>
</tr>
</tbody>
</table>

2.39.2 RPCs associated with class: console

RPC name: get_all

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

Signature:

\[(\text{console ref set}) \text{ get_all} (\text{session_id s})\]

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:

\[((\text{console ref} \rightarrow \text{console record}) \text{ map}) \text{ get_all_records} (\text{session_id s})\]

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:

\[\text{string get_uuid} (\text{session_id s, console ref self})\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>console ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>
Return Type: string
value of the field

RPC name: get_protocol

Overview:
Get the protocol field of the given console.

Signature:

\[(\text{console}_\text{protocol}) \ \text{get}_\text{protocol} \ (\text{session}_\text{id} \ s, \ \text{console}_\text{ref} \ \text{self})\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>console ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: \text{console}_\text{protocol}
value of the field

RPC name: get_location

Overview:
Get the location field of the given console.

Signature:

\[\text{string} \ \text{get}_\text{location} \ (\text{session}_\text{id} \ s, \ \text{console}_\text{ref} \ \text{self})\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>console ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get_VM

Overview:
Get the VM field of the given console.

Signature:

\[(\text{VM}_\text{ref}) \ \text{get}_\text{VM} \ (\text{session}_\text{id} \ s, \ \text{console}_\text{ref} \ \text{self})\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>console ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: \text{VM}_\text{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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>console ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>console ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>console ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>console ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>console record</td>
<td>args</td>
<td>All constructor arguments</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>console ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>console ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: console record
all fields from the object
2.40 Class: user

2.40.1 Fields for class: user

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>A user of the system.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| RO     | uuid                   | string | Unique identifier/object reference |
| RO     | short_name             | string | short name (e.g. userid)          |
| RW     | fullname               | string | full name                          |
| RW     | other_config           | (string → string) map | additional configuration |

2.40.2 RPCs associated with class: user

RPC name: get_uuid

Overview:
Get the uuid field of the given user.

Signature:

```plaintext
string get_uuid (session_id s, user ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get_short_name

Overview:
Get the short_name field of the given user.

Signature:

```plaintext
string get_short_name (session_id s, user ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field
RPC name: get_fullname

Overview:
Get the fullname field of the given user.

Signature:

```c
string get_fullname (session_id s, user ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: set_fullname

Overview:
Set the fullname field of the given user.

Signature:

```c
void set_fullname (session_id s, user ref self, string value)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_other_config

Overview:
Get the other_config field of the given user.

Signature:

```c
((string -> string) map) get_other_config (session_id s, user ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```c
void set_other_config (session_id s, user ref self, (string -> string) map value)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user record</td>
<td>args</td>
<td>All constructor arguments</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**Return Type:** user record

all fields from the object
2.41 Class: data_source

2.41.1 Fields for class: data_source

<table>
<thead>
<tr>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROrun</td>
<td>nameLabel</td>
<td>string</td>
<td>a human-readable name</td>
</tr>
<tr>
<td>ROrun</td>
<td>name_description</td>
<td>string</td>
<td>a notes field containing human-readable description</td>
</tr>
<tr>
<td>ROrun</td>
<td>enabled</td>
<td>bool</td>
<td>true if the data source is being logged</td>
</tr>
<tr>
<td>ROrun</td>
<td>standard</td>
<td>bool</td>
<td>true if the data source is enabled by default. Non-default data sources cannot be disabled</td>
</tr>
<tr>
<td>ROrun</td>
<td>units</td>
<td>string</td>
<td>the units of the value</td>
</tr>
<tr>
<td>ROrun</td>
<td>min</td>
<td>float</td>
<td>the minimum value of the data source</td>
</tr>
<tr>
<td>ROrun</td>
<td>max</td>
<td>float</td>
<td>the maximum value of the data source</td>
</tr>
<tr>
<td>ROrun</td>
<td>value</td>
<td>float</td>
<td>current value of the data source</td>
</tr>
</tbody>
</table>

2.41.2 RPCs associated with class: data_source

Class data_source has no additional RPCs associated with it.
2.42 Class: blob

2.42.1 Fields for class: blob

<table>
<thead>
<tr>
<th>Name</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blob</td>
<td>RO run</td>
<td>uuid</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td></td>
<td>name_label</td>
<td>string</td>
<td>a human-readable name</td>
</tr>
<tr>
<td></td>
<td>name</td>
<td>string</td>
<td>a notes field containing human-readable description</td>
</tr>
<tr>
<td></td>
<td>size</td>
<td>int</td>
<td>Size of the binary data, in bytes</td>
</tr>
<tr>
<td></td>
<td>public</td>
<td>bool</td>
<td>True if the blob is publicly accessible</td>
</tr>
<tr>
<td></td>
<td>last_updated</td>
<td>datetime</td>
<td>Time at which the data in the blob was last updated</td>
</tr>
<tr>
<td></td>
<td>mime_type</td>
<td>string</td>
<td>The mime type associated with this object. Defaults to 'application/octet-stream' if the empty string is supplied</td>
</tr>
</tbody>
</table>

2.42.2 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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>mime_type</td>
<td>The mime-type of the blob. Defaults to 'application/octet-stream' if the empty string is supplied</td>
</tr>
<tr>
<td>bool</td>
<td>public</td>
<td>True if the blob should be publicly available</td>
</tr>
</tbody>
</table>

Return Type: blob ref
The reference to the created blob

RPC name: destroy

Overview:

Signature:

void destroy (session_id s, blob ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blob ref</td>
<td>self</td>
<td>The reference of the blob to destroy</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blob ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field
RPC name: get_name_label

Overview:
Get the name/label field of the given blob.

Signature:

```plaintext
string get_name_label (session_id s, blob ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blob ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string

value of the field

RPC name: set_name_label

Overview:
Set the name/label field of the given blob.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blob ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_name_description

Overview:
Get the name/description field of the given blob.

Signature:

```plaintext
string get_name_description (session_id s, blob ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blob ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blob ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blob ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blob ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: bool

value of the field
RPC name: set_public

Overview:
Set the public field of the given blob.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blob ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>bool</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_last_updated

Overview:
Get the last_updated field of the given blob.

Signature:

```c
datetime get_last_updated (session_id s, blob ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blob ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: datetime

value of the field

RPC name: get_mime_type

Overview:
Get the mime_type field of the given blob.

Signature:

```c
string get_mime_type (session_id s, blob ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blob ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blob ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>label</td>
<td>label of object to return</td>
</tr>
</tbody>
</table>

Return Type: blob ref set
references to objects with matching names
2.43 Class: message

2.43.1 Fields for class: message

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>message</td>
<td>An message for the attention of the administrator.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO_run</td>
<td>uuid</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>RO_run</td>
<td>name</td>
<td>string</td>
<td>The name of the message</td>
</tr>
<tr>
<td>RO_run</td>
<td>priority</td>
<td>int</td>
<td>The message priority, 0 being low priority</td>
</tr>
<tr>
<td>RO_run</td>
<td>cls</td>
<td>cls</td>
<td>The class of the object this message is associated with</td>
</tr>
<tr>
<td>RO_run</td>
<td>obj_uuid</td>
<td>string</td>
<td>The uuid of the object this message is associated with</td>
</tr>
<tr>
<td>RO_run</td>
<td>timestamp</td>
<td>datetime</td>
<td>The time at which the message was created</td>
</tr>
<tr>
<td>RO_run</td>
<td>body</td>
<td>string</td>
<td>The body of the message</td>
</tr>
</tbody>
</table>

2.43.2 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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>name</td>
<td>The name of the message</td>
</tr>
<tr>
<td>int</td>
<td>priority</td>
<td>The priority of the message</td>
</tr>
<tr>
<td>cls</td>
<td>cls</td>
<td>The class of this message is associated with</td>
</tr>
<tr>
<td>string</td>
<td>obj_uuid</td>
<td>The uuid of this message is associated with</td>
</tr>
<tr>
<td>string</td>
<td>body</td>
<td>The body of the message</td>
</tr>
</tbody>
</table>

Return Type: message ref

The reference of the created message

RPC name: destroy

Overview:

Signature:

void destroy (session_id s, message ref self)

Arguments:
2.43. CLASS: MESSAGE

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>message ref</td>
<td>self</td>
<td>The reference of the message to destroy</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cls</td>
<td>cls</td>
<td>The class of object</td>
</tr>
<tr>
<td>string</td>
<td>obj_uuid</td>
<td>The uuid of the object</td>
</tr>
<tr>
<td>datetime</td>
<td>since</td>
<td>The cutoff time</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>datetime</td>
<td>since</td>
<td>The cutoff time</td>
</tr>
</tbody>
</table>

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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>message ref</td>
<td>self</td>
<td>The reference to the message</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>The uuid of the message</td>
</tr>
</tbody>
</table>

**Return Type:** message ref

The message reference

**RPC name: get_all_records**

**Overview:**

**Signature:**

\[
((message \ ref \ \rightarrow \ message \ record) \ map) \ get\_all\_records \ (session\_id \ s)
\]

**Return Type:** (message ref \ \rightarrow \ message record) map

The messages
RPC name: get_all_records_where

Overview:

Signature:

\[(\text{message ref} \rightarrow \text{message record}) \text{ map} \) get_all_records_where (session_id s, string expr)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>expr</td>
<td>The expression to match (not currently used)</td>
</tr>
</tbody>
</table>

Return Type: \((\text{message ref} \rightarrow \text{message record}) \text{ map}\)

The messages
2.44 Class: secret

2.44.1 Fields for class: secret

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>secret</td>
<td>A secret</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO</td>
<td>uuid</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>RW</td>
<td>value</td>
<td>string</td>
<td>the secret</td>
</tr>
<tr>
<td>RW</td>
<td>other_config</td>
<td>(string → string) map</td>
<td>other_config</td>
</tr>
</tbody>
</table>

2.44.2 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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>secret ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

**RPC name: get_value**

**Overview:**
Get the value field of the given secret.

**Signature:**

```plaintext
string get_value (session_id s, secret ref self)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>secret ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**Return Type:** string

value of the field

**RPC name: set_value**

**Overview:**
Set the value field of the given secret.

**Signature:**

```plaintext
void set_value (session_id s, secret ref self, string value)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>secret ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

**Return Type:** void

**RPC name: get_other_config**

**Overview:**
Get the other_config field of the given secret.

**Signature:**

```plaintext
((string -> string) map) get_other_config (session_id s, secret ref self)
```

**Arguments:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>secret ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>secret ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>secret ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>secret ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>secret record</td>
<td>args</td>
<td>All constructor arguments</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>secret ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

\[(\text{secret record}) \text{ get_record} (\text{session_id } s, \text{ secret ref } self)\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>secret ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: secret record
all fields from the object
2.45 Class: tunnel

2.45.1 Fields for class: tunnel

<table>
<thead>
<tr>
<th>Name</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tunnel</td>
<td>run</td>
<td>uuid</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>tunnel</td>
<td>ins</td>
<td>access</td>
<td>The interface through which the tunnel is accessed</td>
</tr>
<tr>
<td>tunnel</td>
<td>ins</td>
<td>transport</td>
<td>The interface used by the tunnel</td>
</tr>
<tr>
<td>tunnel</td>
<td>RW</td>
<td>status</td>
<td>Status information about the tunnel</td>
</tr>
<tr>
<td>tunnel</td>
<td>RW</td>
<td>other_config</td>
<td>(string → string) map</td>
</tr>
</tbody>
</table>

2.45.2 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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIF ref</td>
<td>transport_PIF</td>
<td>PIF which receives the tagged traffic</td>
</tr>
<tr>
<td>network ref</td>
<td>network</td>
<td>Network to receive the tunnelled traffic</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tunnel ref</td>
<td>self</td>
<td>tunnel to destroy</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tunnel ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tunnel ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tunnel ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tunnel ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tunnel ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tunnel ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tunnel ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tunnel ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: (string → string) map

RPC name: set_other_config

Overview:
Set the other_config field of the given tunnel.

Signature:

```python
void set_other_config (session_id s, tunnel ref self, (string -> string) map value)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tunnel ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tunnel ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```python
void remove_from_other_config (session_id s, tunnel ref self, string key)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tunnel ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tunnel ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: tunnel record
all fields from the object
2.46 Class: PCI

2.46.1 Fields for class: PCI

<table>
<thead>
<tr>
<th>Name</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI</td>
<td>Description</td>
<td>A PCI device.</td>
<td></td>
</tr>
<tr>
<td>RO run</td>
<td>uuid</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>RO ins</td>
<td>class_name</td>
<td>string</td>
<td>PCI class name</td>
</tr>
<tr>
<td>RO ins</td>
<td>vendor_name</td>
<td>string</td>
<td>Vendor name</td>
</tr>
<tr>
<td>RO ins</td>
<td>device_name</td>
<td>string</td>
<td>Device name</td>
</tr>
<tr>
<td>RO ins</td>
<td>host</td>
<td>host ref</td>
<td>Physical machine that owns the PCI device</td>
</tr>
<tr>
<td>RO ins</td>
<td>pci_id</td>
<td>string</td>
<td>PCI ID of the physical device</td>
</tr>
<tr>
<td>RO run</td>
<td>dependencies</td>
<td>PCI ref set</td>
<td>List of dependent PCI devices</td>
</tr>
<tr>
<td>RW</td>
<td>other_config</td>
<td>(string → string) map</td>
<td>Additional configuration</td>
</tr>
<tr>
<td>RO ins</td>
<td>subsystem_vendor_name</td>
<td>string</td>
<td>Subsystem vendor name</td>
</tr>
<tr>
<td>RO ins</td>
<td>subsystem_device_name</td>
<td>string</td>
<td>Subsystem device name</td>
</tr>
</tbody>
</table>

2.46.2 RPCs associated with class: PCI

RPC name: get_all

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

Signature:

\((\text{PCI ref set}) \text{get\_all (session\_id s)}\)

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:

\((\text{(PCI ref} \rightarrow \text{PCI record}) \text{map) get\_all\_records (session\_id s)}\)

Return Type: (PCI ref → 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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_subsystem_vendor_name

Overview:
Get the subsystem_vendor_name field of the given PCI.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```c
string get_subsystem_device_name (session_id s, PCI ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: PCI record
all fields from the object
2.47 Class: PGPU

2.47.1 Fields for class: PGPU

<table>
<thead>
<tr>
<th>Name</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run</td>
<td>uuid</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>Ins</td>
<td>PCI</td>
<td>PCI ref</td>
<td>Link to underlying PCI device</td>
</tr>
<tr>
<td>Ins</td>
<td>GPU_group</td>
<td>GPU_group ref</td>
<td>GPU group the pGPU is contained in</td>
</tr>
<tr>
<td>Run</td>
<td>host</td>
<td>host ref</td>
<td>Host that owns the GPU</td>
</tr>
<tr>
<td>RO</td>
<td>ins</td>
<td>GPU_group ref</td>
<td>GPU group the pGPU is contained in</td>
</tr>
<tr>
<td>Run</td>
<td>run</td>
<td>host</td>
<td>Host that owns the GPU</td>
</tr>
<tr>
<td>RW</td>
<td>other_config</td>
<td>(string → string) map</td>
<td>Additional configuration</td>
</tr>
<tr>
<td>RO</td>
<td>run</td>
<td>supported_VGPU_types</td>
<td>VGPU_type ref set</td>
</tr>
<tr>
<td>RO</td>
<td>run</td>
<td>enabled_VGPU_types</td>
<td>VGPU_type ref set</td>
</tr>
<tr>
<td>RO</td>
<td>run</td>
<td>resident_VGPUs</td>
<td>VGPU ref set</td>
</tr>
<tr>
<td>RO</td>
<td>run</td>
<td>supported_VGPU_max_capacities</td>
<td>(VGPU_type ref → int) map</td>
</tr>
<tr>
<td>RO</td>
<td>run</td>
<td>dom0_access</td>
<td>pgpu_dom0_access</td>
</tr>
<tr>
<td>RO</td>
<td>run</td>
<td>is_system_display_device</td>
<td>bool</td>
</tr>
</tbody>
</table>

2.47.2 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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGPU ref</td>
<td>self</td>
<td>The PGPU to which we are adding an enabled VGPU type</td>
</tr>
<tr>
<td>VGPU_type ref</td>
<td>value</td>
<td>The VGPU type to enable</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGPU ref</td>
<td>self</td>
<td>The PGPU from which we are removing an enabled VGPU type</td>
</tr>
<tr>
<td>VGPU_type ref</td>
<td>value</td>
<td>The VGPU type to disable</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGPU ref</td>
<td>self</td>
<td>The PGPU on which we are enabling a set of VGPU types</td>
</tr>
<tr>
<td>VGPU_type ref set</td>
<td>value</td>
<td>The VGPU types to enable</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGPU ref</td>
<td>self</td>
<td>The PGPU to move to a new group</td>
</tr>
<tr>
<td>GPU_group ref</td>
<td>value</td>
<td>The group to which the PGPU will be moved</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGPU ref</td>
<td>self</td>
<td>The PGPU to query</td>
</tr>
<tr>
<td>VGPU type ref</td>
<td>vgpu_type</td>
<td>The VGPU type for which we want to find the number of VGPUs which can still be started on this PGPU</td>
</tr>
</tbody>
</table>

Return Type: int
The number of VGPUs of the specified type which can still be started on this PGPU

RPC name: enable_dom0_access
Overview:

Signature:

\[(pgpu_dom0_access) enable_dom0_access (session_id s, PGPU ref self)\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGPU ref</td>
<td>self</td>
<td>The PGPU to which dom0 will be granted access</td>
</tr>
</tbody>
</table>

Return Type: pgpu_dom0_access
The accessibility of this PGPU from dom0

RPC name: disable_dom0_access
Overview:

Signature:

\[(pgpu_dom0_access) disable_dom0_access (session_id s, PGPU ref self)\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGPU ref</td>
<td>self</td>
<td>The PGPU to which dom0 will be denied access</td>
</tr>
</tbody>
</table>

Return Type: pgpu_dom0_access
The accessibility of this PGPU from dom0
RPC name: get_all

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

Signature:

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

Return Type: (PGPU ref -> 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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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)
```
2.47. CLASS: PGPU

CHAPTER 2. API REFERENCE

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: (string → string) map

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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)
2.47. CLASS: PGPU

CHAPTER 2. API REFERENCE

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_supported_VGPU_types

Overview:
Get the supported_VGPU_types field of the given PGPU.

Signature:

\[(VGPU\_type \text{ ref set}) \ get\_supported\_VGPU\_types (\text{session\_id } s, \text{ PGPU ref self})\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

\[(VGPU\_type \text{ ref set}) \ get\_enabled\_VGPU\_types (\text{session\_id } s, \text{ PGPU ref self})\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

\[(\text{VGPU ref set}) \ get\_resident\_VGPUs (\text{session\_id } s, \text{ PGPU ref self})\]
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: (VGPU_type ref -> int) map
value of the field

RPC name: get_dom0_access

Overview:
Get the dom0_access field of the given PGPU.

Signature:

```c
(pgpu_dom0_access) get_dom0_access (session_id s, PGPU ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: pgpu_dom0_access
value of the field

RPC name: get_is_system_display_device

Overview:
Get the is_system_display_device field of the given PGPU.

Signature:

```c
bool get_is_system_display_device (session_id s, PGPU ref self)
```
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: bool
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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: PGPU record
all fields from the object
2.48 Class: GPU_group

2.48.1 Fields for class: GPU_group

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU_group</td>
<td></td>
<td>A group of compatible GPUs across the resource pool.</td>
</tr>
<tr>
<td>RO_run uuid</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>RW name_label</td>
<td>string</td>
<td>a human-readable name</td>
</tr>
<tr>
<td>RW name_description</td>
<td>string</td>
<td>a notes field containing human-readable description</td>
</tr>
<tr>
<td>RO_run PGPUs</td>
<td>PGPUs ref set</td>
<td>List of pGPUs in the group</td>
</tr>
<tr>
<td>RO_run VGPUs</td>
<td>VGPU ref set</td>
<td>List of vGPUs using the group</td>
</tr>
<tr>
<td>RO_run GPU_types</td>
<td>string set</td>
<td>List of GPU types (vendor+device ID) that can be in this group</td>
</tr>
<tr>
<td>RW other_config</td>
<td>(string → string) map</td>
<td>Additional configuration</td>
</tr>
<tr>
<td>RW allocation_algorithm</td>
<td>allocation_algorithm</td>
<td>Current allocation of vGPUs to pGPUs for this group</td>
</tr>
<tr>
<td>RO_run supported_VGPU_types</td>
<td>VGPU_type ref set</td>
<td>vGPU types supported on at least one of the pGPUs in this group</td>
</tr>
<tr>
<td>RO_run enabled_VGPU_types</td>
<td>VGPU_type ref set</td>
<td>vGPU types supported on at least one of the pGPUs in this group</td>
</tr>
</tbody>
</table>

2.48.2 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 → string) map other_config)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>name_label</td>
<td></td>
</tr>
<tr>
<td>string</td>
<td>name_description</td>
<td></td>
</tr>
<tr>
<td>(string → string) map</td>
<td>other_config</td>
<td></td>
</tr>
</tbody>
</table>

Return Type: GPU_group ref

RPC name: destroy

Overview:

Signature:

void destroy (session_id s, GPU_group ref self)
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU_group ref</td>
<td>self</td>
<td>The vGPU to destroy</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU_group ref</td>
<td>self</td>
<td>The GPU group to query</td>
</tr>
<tr>
<td>VGPU_type ref</td>
<td>vgpu_type</td>
<td>The VGPU type for which the remaining ca-pacity will be calculated</td>
</tr>
</tbody>
</table>

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 -> GPU_group record) map
records of all objects
RPC name: get_uuid

Overview:
Get the uuid field of the given GPU_group.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU_group ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get_name_label

Overview:
Get the name/label field of the given GPU_group.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU_group ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: set_name_label

Overview:
Set the name/label field of the given GPU_group.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU_group ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU_group ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU_group ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU_group ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: PGPU ref set
value of the field
RPC name: get_VGPUs

Overview:
Get the VGPUs field of the given GPU_group.

Signature:

\[(VGPU \text{ ref set}) \text{ get}_\text{VGPUs} (\text{session}_\text{id } s, \text{GPU}_\text{group ref self})\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU_group</td>
<td>ref</td>
<td>self</td>
</tr>
<tr>
<td></td>
<td></td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

\[(\text{string set}) \text{ get}_{\text{GPU_types}} (\text{session}_\text{id } s, \text{GPU}_\text{group ref self})\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU_group</td>
<td>ref</td>
<td>self</td>
</tr>
<tr>
<td></td>
<td></td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

\[((\text{string } \rightarrow \text{string}) \text{ map}) \text{ get}_{\text{other_config}} (\text{session}_\text{id } s, \text{GPU}_\text{group ref self})\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU_group</td>
<td>ref</td>
<td>self</td>
</tr>
<tr>
<td></td>
<td></td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

```plaintext
void set\_other\_config (session\_id s, GPU\_group ref self, (string -> string) map value)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU_group ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string -&gt; string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

```plaintext
void add\_to\_other\_config (session\_id s, GPU\_group ref self, string key, string value)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU_group ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

```plaintext
void remove\_from\_other\_config (session\_id s, GPU\_group ref self, string key)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU_group ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU_group ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU_group ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>allocation_algorithm</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU_group ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU_group ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU_group ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

\[(\text{GPU\_group ref set}) \text{ get\_by\_name\_label (session\_id s, string label)}\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>label</td>
<td>label of object to return</td>
</tr>
</tbody>
</table>

Return Type: GPU\_group ref set
references to objects with matching names
2.49 Class: VGPU

2.49.1 Fields for class: VGPU

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

2.49.2 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) map other_config, VGPU_type ref type)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM ref</td>
<td>VM</td>
<td></td>
</tr>
<tr>
<td>GPU_group ref</td>
<td>GPU_group</td>
<td></td>
</tr>
<tr>
<td>string</td>
<td>device</td>
<td></td>
</tr>
<tr>
<td>(string → string) map</td>
<td>other_config</td>
<td>Additional configuration</td>
</tr>
<tr>
<td>VGPU_type ref</td>
<td>type</td>
<td></td>
</tr>
</tbody>
</table>

Return Type: VGPU ref
reference to the newly created object

RPC name: destroy

Overview:

Signature:

void destroy (session_id s, VGPU ref self)

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU ref</td>
<td>self</td>
<td>The vGPU to destroy</td>
</tr>
</tbody>
</table>
Return Type: void

RPC name: get_all

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

Signature:
((VGPU ref -> VGPU record) map) get_all_records (session_id s)

Return Type: (VGPU ref → 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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>(string → string) map</td>
<td>value</td>
<td>New value to set</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to add</td>
</tr>
<tr>
<td>string</td>
<td>value</td>
<td>Value to add</td>
</tr>
</tbody>
</table>

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:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
<tr>
<td>string</td>
<td>key</td>
<td>Key to remove</td>
</tr>
</tbody>
</table>

Return Type: void

RPC name: get_type

Overview:
Get the type field of the given VGPU.

Signature:

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

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

**Return Type:** `VGPU record`
all fields from the object
2.50 Class: VGPU_type

2.50.1 Fields for class: VGPU_type

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Quals</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO_run</td>
<td></td>
<td></td>
<td>uuid</td>
<td>string</td>
<td>Unique identifier/object reference</td>
</tr>
<tr>
<td>RO_ins</td>
<td></td>
<td></td>
<td>vendor_name</td>
<td>string</td>
<td>Name of VGPU vendor</td>
</tr>
<tr>
<td>RO_ins</td>
<td></td>
<td></td>
<td>model_name</td>
<td>string</td>
<td>Model name associated with the VGPU type</td>
</tr>
<tr>
<td>RO_ins</td>
<td></td>
<td></td>
<td>framebuffer_size</td>
<td>int</td>
<td>Framebuffer size of the VGPU type, in bytes</td>
</tr>
<tr>
<td>RO_ins</td>
<td></td>
<td></td>
<td>max_heads</td>
<td>int</td>
<td>Maximum number of displays supported by the VGPU type</td>
</tr>
<tr>
<td>RO_ins</td>
<td></td>
<td></td>
<td>max_resolution_x</td>
<td>int</td>
<td>Maximum resolution (width) supported by the VGPU type</td>
</tr>
<tr>
<td>RO_ins</td>
<td></td>
<td></td>
<td>max_resolution_y</td>
<td>int</td>
<td>Maximum resolution (height) supported by the VGPU type</td>
</tr>
<tr>
<td>RO_run</td>
<td></td>
<td></td>
<td>supported_on_PGPUs</td>
<td>PGPU ref set</td>
<td>List of PGPUs that support this VGPU type</td>
</tr>
<tr>
<td>RO_run</td>
<td></td>
<td></td>
<td>enabled_on_PGPUs</td>
<td>PGPU ref set</td>
<td>List of PGPUs that have this VGPU type enabled</td>
</tr>
<tr>
<td>RO_run</td>
<td></td>
<td></td>
<td>VGPUs</td>
<td>VGPU ref set</td>
<td>List of VGPUs of this type</td>
</tr>
<tr>
<td>RO_run</td>
<td></td>
<td></td>
<td>supported_on_GPU_groups</td>
<td>GPU_group ref set</td>
<td>List of GPU groups in which at least one</td>
</tr>
<tr>
<td>RO_run</td>
<td></td>
<td></td>
<td>enabled_on_GPU_groups</td>
<td>GPU_group ref set</td>
<td>PGPU supports this VGPU type</td>
</tr>
<tr>
<td>RO_ins</td>
<td></td>
<td></td>
<td>implementation</td>
<td>vgpu_type, implementation</td>
<td>The internal implementation of this VGPU type</td>
</tr>
<tr>
<td>RO_ins</td>
<td></td>
<td></td>
<td>identifier</td>
<td>string</td>
<td>Key used to identify VGPU types and avoid creating duplicates - this field is used internally and not intended for interpretation by API clients</td>
</tr>
<tr>
<td>RO_ins</td>
<td></td>
<td></td>
<td>experimental</td>
<td>bool</td>
<td>Indicates whether VGPUs of this type should be considered experimental</td>
</tr>
</tbody>
</table>

2.50.2 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:

\[
((\text{VGPU	extunderscore type ref} \rightarrow \text{VGPU	extunderscore type record}) \text{ map}) \text{ get	extunderscore all	extunderscore records} (\text{session	extunderscore id } s)
\]

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:

\[
\text{string get	extunderscore uuid} (\text{session	extunderscore id } s, \text{ VGPU	extunderscore type ref self})
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU_type ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get_vendor_name

Overview:
Get the vendor_name field of the given VGPU_type.

Signature:

\[
\text{string get	extunderscore vendor	extunderscore name} (\text{session	extunderscore id } s, \text{ VGPU	extunderscore type ref self})
\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU_type ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU_type ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU_type ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU_type ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU_type ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU_type ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU_type ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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)
```
2.50. CLASS: VGPU\_TYPE

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU_type ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: PGPU ref set
value of the field

RPC name: get\_VGPUs

Overview:
Get the VGPUs field of the given VGPU\_type.

Signature:

\[(VGPU\_ref\ set)\ \text{get\_VGPUs}\ (session\_id\ s,\ VGPU\_type\ ref\ self)\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU_type ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

\[(GPU\_group\ ref\ set)\ \text{get\_supported\_on\_GPU\_groups}\ (session\_id\ s,\ VGPU\_type\ ref\ self)\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU_type ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

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:

\[(GPU\_group\ ref\ set)\ \text{get\_enabled\_on\_GPU\_groups}\ (session\_id\ s,\ VGPU\_type\ ref\ self)\]
Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU_type ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: GPU_group ref set
value of the field

RPC name: get_implementation

Overview:
Get the implementation field of the given VGPU_type.

Signature:

\[(vgpu\_type\_implementation) get\_implementation (session\_id s, VGPU\_type ref self)\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU_type ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: vgpu\_type\_implementation
value of the field

RPC name: get_identifier

Overview:
Get the identifier field of the given VGPU_type.

Signature:

\[string get\_identifier (session\_id s, VGPU\_type ref self)\]

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU_type ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: string
value of the field

RPC name: get_experimental

Overview:
Get the experimental field of the given VGPU_type.

Signature:

\[bool get\_experimental (session\_id s, VGPU\_type ref self)\]
2.50. CLASS: VGPU\_TYPE

CHAPTER 2. API REFERENCE

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU_type ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: bool
description of the field

RPC name: get\_by\_uuid

Overview:
Get a reference to the VGPU\_type instance with the specified UUID.

Signature:

```plaintext
(VGPU\_type ref) get\_by\_uuid (session\_id s, string uuid)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>uuid</td>
<td>UUID of object to return</td>
</tr>
</tbody>
</table>

Return Type: VGPU\_type ref
description of the object

RPC name: get\_record

Overview:
Get a record containing the current state of the given VGPU\_type.

Signature:

```plaintext
(VGPU\_type record) get\_record (session\_id s, VGPU\_type ref self)
```

Arguments:

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGPU_type ref</td>
<td>self</td>
<td>reference to the object</td>
</tr>
</tbody>
</table>

Return Type: VGPU\_type record
description of the record
2.51 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:

```xml
<methodResponse>
  <fault>
    <struct>
      <member>
        <name>faultCode</name>
        <value><int>-1</int></value>
      </member>
      <member>
        <name>faultString</name>
        <value><string>Malformed request</string></value>
      </member>
    </struct>
    <value></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:

```xml
<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.
2.51.1 Error Codes

**ACTIVATION WHILE NOT FREE**
An activation key can only be applied when the edition is set to 'free'.

No parameters.


**ADDRESS VIOLATES LOCKING CONSTRAINT**
The specified IP address violates the VIF locking configuration.

Signature:

ADDRESS VIOLATES LOCKING CONSTRAINT(address)


**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:

```c
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:

```c
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:

```c
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:

```c
CANNOT_DESTROY_DISASTER_RECOVERY_TASK(reason)
```

**CANNOT_DESTROY_SYSTEM_NETWORK**

You tried to destroy a system network: these cannot be destroyed.

Signature:

```c
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 not be obtained from the master.

Signature:
CANNOT_FETCH_PATCH(uuid)

CANNOT_FIND_OEM_BACKUP_PARTITION
The backup partition to stream the update 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.
2.51. **ERROR HANDLING**

CHAPTER 2. **API REFERENCE**

**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)

CLUSTERED_SR_DEGRADED
An SR is using clustered local storage. It is not safe to reboot a host at the moment.
Signature:
CLUSTERED_SR_DEGRADED(sr)

COULD_NOT_FIND_NETWORK_INTERFACE_WITHSPECIFIED_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:**

```python
DEVICE_DETACH_TIMEOUT(type, ref)
```

---

**DEVICE_NOT_ATTACHED**

The operation could not be performed because the VBD was not connected to the VM.

**Signature:**

```python
DEVICE_NOT_ATTACHED(VBD)
```

---

**DISK_VBD_MUST_BE_READWRITE_FOR_HVM**

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

**Signature:**

```python
DISK_VBD_MUST_BE_READWRITE_FOR_HVM(vbd)
```

---

**DOMAIN_BUILDER_ERROR**

An internal error generated by the domain builder.

**Signature:**

```python
DOMAIN_BUILDER_ERROR(function, code, message)
```

---

**DOMAIN_EXISTS**

The operation could not be performed because a domain still exists for the specified VM.

**Signature:**

```python
DOMAIN_EXISTS(vm, domid)
```

---

**DUPLICATE_MAC_SEED**

This MAC seed is already in use by a VM in the pool

**Signature:**

```python
DUPLICATE_MAC_SEED(seed)
```
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_SUBSCRIPTIONPARSE_FAILURE(subscription)

FAILED_TO_START_EMULATOR
An emulator required to run this VM failed to start

Signature:
FAILED_TO_START_EMULATOR(vm, name, msg)
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 VGPUs 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_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.

No parameters.

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_DISABLE_IN_PROGRESS
The operation could not be performed because HA disable is in progress

No parameters.
HA_ENABLE_IN_PROGRESS
The operation could not be performed because HA enable is in progress
No parameters.

HA_FAILED_TO_FORM_LIVESET
HA could not be enabled on the Pool because a liveset could not be formed: check storage and network heartbeat paths.
No parameters.

HA_HEARTBEAT_DAEMON_STARTUP_FAILED
The host could not join the liveset because the HA daemon failed to start.
No parameters.

HA_HOST_CANNOT_ACCESS_STATEFILE
The host could not join the liveset because the HA daemon could not access the heartbeat disk.
No parameters.

HA_HOST_CANNOTSEE_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_CANNOTSEE_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_CLUSTER_STACK_ACTIVE
This operation cannot be performed, because it is incompatible with the currently active HA cluster stack.
Signature:
INCOMPATIBLE_CLUSTER_STACK_ACTIVE(cluster_stack)

INCOMPATIBLE_PIF_PROPERTIES
These PIFs can not be bonded, because their properties are different.
No parameters.

INCOMPATIBLE_STATEFILE_SR
The specified SR is incompatible with the selected HA cluster stack.
Signature:
INCOMPATIBLE_STATEFILE_SR(SR type)

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_CIDR_ADDRESS_SPECIFIED
A required parameter contained an invalid CIDR address (addr/prefix length)
Signature:
INVALID_CIDR_ADDRESS_SPECIFIED(parameter)
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 because your license lacks a needed feature. Please contact your support representative.
Signature:
LICENCE_RESTRICTION(feature)

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_CONNECTIONDETAILS
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_HOSTSAVAILABLE
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_NOLONGER_EXISTS
The specified object no longer exists.
No parameters.

ONLY_ALLOWED_ON_OEM_EDITION
This command is only allowed on the OEM edition.
Signature:
ONLY_ALLOWED_ON_OEM_EDITION(command)

OPENVSWITCH_NOT_ACTIVE
This operation needs the OpenVSwitch networking backend to be enabled on all hosts in the pool.
No parameters.

OPERATION_BLOCKED
You attempted an operation that was explicitly blocked (see the blocked operations field of the given object).
Signature:
OPERATION_BLOCKED(ref, code)

OPERATION_NOT_ALLOWED
You attempted an operation that was not allowed.
Signature:
OPERATION_NOT_ALLOWED(reason)
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)
2.51. ERROR HANDLING

PATCH_APPLY_FAILED_BACKUP_FILES_EXIST
The patch apply failed: there are backup files created while applying patch. Please remove these backup files before applying patch again.

Signature:
PATCH_APPLY_FAILED_BACKUP_FILES_EXIST(output)

PATCH_IS_APPLIED
The specified patch is applied and cannot be destroyed.
No parameters.

PATCH_PRECHECK_FAILED_ISO_MOUNTED
Tools ISO must be ejected from all running VMs.

Signature:
PATCH_PRECHECK_FAILED_ISO_MOUNTED(patch)

PATCH_PRECHECK_FAILED_OUT_OF_SPACE
The patch precheck stage failed: the server does not have enough space.

Signature:
PATCH_PRECHECK_FAILED_OUT_OF_SPACE(patch, found_space, required_required)

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_MORE_THAN_ONE_IP
Only one PIF on a bond is allowed to have an IP configuration.
No parameters.

---------------------------

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 currently set to 'none')

No parameters.

PIF_HAS_NO_V6_NETWORK_CONFIGURATION
PIF has no IPv6 configuration (mode currently 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)
2.51. ERROR HANDLING

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.

POOL_JOINING_HOST_MUST_ONLY_HAVE_PHYSICAL_PIFS
The host joining the pool may not have any bonds, VLANs or tunnels.
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)
2.5.1. ERROR HANDLING

CHAPTER 2. API REFERENCE

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 DOES NOT SUPPORT MIGRATION
You attempted to migrate a VDI on SR which doesn’t have snapshot capability
Signature:
SR_DOES_NOT_SUPPORT_MIGRATION(sr)

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)

UNIMPLEMENTED_IN_SM_BACKEND
You have attempted a function which is not implemented
Signature:
UNIMPLEMENTED_IN_SM_BACKEND(message)

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).
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_ISEMPTY
Operation could not be performed because the drive is empty

Signature:
VBD_IS_EMPTY(vbd)

-------------------

VBD_NOTEMPTY
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\_COPY\_FAILED
The VDI copy action has failed

No parameters.

VDI\_HAS\_RRDS
The operation cannot be performed because this VDI has rrd stats

Signature:
VDI\_HAS\_RRDS(vdi)

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\_ON\_BOOT\_MODE\_INCOMPATIBLE\_WITH\_OPERATION**  
This operation is not permitted on VMs containing VDIs in the 'on-boot=reset' mode  
**No parameters.**

---

**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)

VIF_NOT_IN_MAP
This VIF was not mapped to a destination Network in VM.migrate_send operation
Signature:
VIF_NOT_IN_MAP(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_CALL_PLUGIN_RATE_LIMIT**
There is a minimal interval required between consecutive plugin calls made on the same VM, please wait before retry.

**Signature:**

```
VM_CALL_PLUGIN_RATE_LIMIT(VM, interval, wait)
```

**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_HOST_INCOMPATIBLE_VIRTUAL_HARDWARE_PLATFORM_VERSION**
You attempted to run a VM on a host that cannot provide the VM’s required Virtual Hardware Platform version.

*Signature:*

`VM_HOST_INCOMPATIBLE_VIRTUAL_HARDWARE_PLATFORM_VERSION(host, host_versions, vm, vm_version)`

---

**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:

```python
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:

```python
VM_IS_PROTECTED(vm)
```

**VM_IS_TEMPLATE**
The operation attempted is not valid for a template VM

Signature:

```python
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:

```python
VM_LACKS_FEATURE_SHUTDOWN(vm)
```

**VM_LACKS_FEATURE_STATIC_IP_SETTING**
You attempted an operation which needs the VM static-ip-setting feature on a VM which lacks it.

Signature:

```python
VM_LACKS_FEATURE_STATIC_IP_SETTING(vm)
```

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

Signature:

```python
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_EMPTY_CD_VBD**
The VM has no empty CD drive (VBD).

**Signature:**

\[ \text{VM\_NO\_EMPTY\_CD\_VBD}(\text{vm}) \]

**VM_NO_SUSPEND_SR**
This VM does not have a suspend SR specified.

**Signature:**

\[ \text{VM\_NO\_SUSPEND\_SR}(\text{vm}) \]

**VM_NO_VCPUS**
You need at least 1 VCPU to start a VM

**Signature:**

\[ \text{VM\_NO\_VCPUS}(\text{vm}) \]

**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:**

\[ \text{VM\_OLD\_PV\_DRIVERS}(\text{vm, major, minor}) \]

**VM_PV_DRIVERS_IN_USE**
VM PV drivers still in use

**Signature:**

\[ \text{VM\_PV\_DRIVERS\_IN\_USE}(\text{vm}) \]

**VM_REBOOTED**
The VM unexpectedly rebooted

**Signature:**

\[ \text{VM\_REBOOTED}(\text{vm}) \]
**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:**

\[
\text{VM_REQUIRES\_GPU}(\text{vm}, \text{GPU\_group})
\]

**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:**

\[
\text{VM_REQUIRES\_IOMMU}(\text{host})
\]

**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:**

\[
\text{VM_REQUIRES\_NETWORK}(\text{vm}, \text{network})
\]

**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:**

\[
\text{VM_REQUIRES\_SR}(\text{vm}, \text{sr})
\]

**VM_REQUIRES_VDI**
VM cannot be started because it requires a VDI which cannot be attached

**Signature:**

\[
\text{VM_REQUIRES\_VDI}(\text{vm}, \text{vdi})
\]
2.5.1. ERROR HANDLING

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_WITHQUIESCE_TIMEOUT
The VSS plug-in has timed out

Signature:
VM_SNAPSHOT_WITHQUIESCE_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
WLB rejected our configured authentication details.

No parameters.

WLB_CONNECTION_REFUSED
WLB refused a connection to the server.

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
WLB reported an internal error.
No parameters.

WLB_MALFORMED_REQUEST
WLB rejected the server’s request as malformed.
No parameters.

WLB_MALFORMED_RESPONSE
WLB said something that the server wasn’t expecting or didn’t understand. The method called on WLB, 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**
WLB reported that the server rejected its configured authentication details.
No parameters.

**WLB_XENSERVER_CONNECTION_REFUSED**
WLB reported that the server refused it a connection (even though we’re connecting perfectly fine in the other direction).
No parameters.

**WLB_XENSERVER_MALFORMED_RESPONSE**
WLB reported that the server said something to it that WLB wasn’t expecting or didn’t understand.
No parameters.

**WLB_XENSERVER_TIMEOUT**
WLB reported that communication with the server timed out.
No parameters.

**WLB_XENSERVER_UNKNOWN_HOST**
WLB reported that its configured server name for this server instance failed to resolve in DNS.
No parameters.
**XAPI_HOOK_FAILED**

3rd party xapi hook failed

**Signature:**

\[
\text{XAPI\_HOOK\_FAILED}(\text{hook\_name, reason, stdout, exit\_code})
\]

**XENAPI_MISSING_PLUGIN**

The requested plugin could not be found.

**Signature:**

\[
\text{XENAPI\_MISSING\_PLUGIN}(\text{name})
\]

**XENAPI_PLUGIN_FAILURE**

There was a failure communicating with the plugin.

**Signature:**

\[
\text{XENAPI\_PLUGIN\_FAILURE}(\text{status, stdout, stderr})
\]

**XEN_VSS_REQ_ERROR_ADDING_VOLUME_TO_SNAPSET_FAILED**

Some volumes to be snapshot could not be added to the VSS snapshot set

**Signature:**

\[
\text{XEN\_VSS\_REQ\_ERROR\_ADDING\_VOLUME\_TO\_SNAPSET\_FAILED}(\text{vm, error\_code})
\]

**XEN_VSS_REQ_ERROR_CREATING_SNAPSHOT**

An attempt to create the snapshots failed

**Signature:**

\[
\text{XEN\_VSS\_REQ\_ERROR\_CREATING\_SNAPSHOT}(\text{vm, error\_code})
\]

**XEN_VSS_REQ_ERROR_CREATING_SNAPSHOT_XML_STRING**

Could not create the XML string generated by the transportable snapshot

**Signature:**

\[
\text{XEN\_VSS\_REQ\_ERROR\_CREATING\_SNAPSHOT\_XML\_STRING}(\text{vm, error\_code})
\]
2.51. ERROR HANDLING

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 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 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)