
InfiniSDK Documentation

Release 225.1.1

Infinidat

Jul 19, 2023

Contents

1	Overview	1
2	Compatibility	3
2.1	Getting Started	3
2.2	Querying Objects	8
2.3	Efficient Querying	9
2.4	The InfiniBox Object	12
2.5	System Capacity	13
2.6	Pools	14
2.7	Volumes	15
2.8	Filesystems	16
2.9	NFS Exports	17
2.10	SMB Shares	20
2.11	Working with Hosts, Clusters and Mappings	20
2.12	Snapshots	23
2.13	Consistency Groups	24
2.14	User Management	25
2.15	System Configuration	27
2.16	Network Configuration	29
2.17	Replication	30
2.18	Using Object Metadata	32
2.19	Events	33
2.20	System Components	34
2.21	Quality of Service	36
2.22	SMB Users and Groups	38
2.23	Active Directory Domains	38
2.24	Extending InfiniSDK	39
2.25	Cookbook	40
2.26	Frequently Asked Questions	41
2.27	API Reference	42
2.28	Advanced Usage	116
2.29	Hooks	116
3	Indices and tables	127
	Python Module Index	129

CHAPTER 1

Overview

InfiniSDK is Infinidat's Official Python SDK for interacting with Infinidat storage products. It provides a clean interface for creating, deleting, querying and manipulating API objects.

CHAPTER 2

Compatibility

InfiniSDK is designed to be compatible with the currently shipped version of InfiniBox, as well as prior releases which are under active support. Compatibility with future version is guaranteed only to a limited extent, especially regarding new or changed functionality.

Contents:

2.1 Getting Started

Note: As a convention throughout this documentation, whenever we write an entity such as `pool`, `volume`, `system` etc., it will *always* refer to a Python object – never to a name string or to an object identifier. When we want to indicate a name or an id we will name variables accordingly (e.g. `volume_id`, `pool_name` etc.).

2.1.1 Installation

Installing InfiniSDK is done by using `pip`:

```
$ pip install infinisdk
```

Note: Depending on your Python installation, the above command might require root privileges

See also:

For more information on `pip` and how to use it to install Python packages, see <https://pip.pypa.io/en/stable/>

2.1.2 Creating the InfiniBox Object

In your Python interpreter, import the `infinisdk.InfiniBox` class, and initialize it with your system address:

```
>>> from infinisdk import InfiniBox
>>> system = InfiniBox(SYSTEM_ADDRESS)
```

Note: SYSTEM_ADDRESS can be a hostname for your system's management address, or an IP address

SSL is disabled by default, but can be easily turned on by passing use_ssl=True to the system constructor:

```
>>> system = InfiniBox(SYSTEM_ADDRESS, use_ssl=True)
```

Note: By default, constructing a system does not send any traffic or API calls to the system. Only performing actual actions or queries does.

2.1.3 Authentication

Authentication information can also be specified via the constructor:

```
>>> system = InfiniBox(SYSTEM_ADDRESS, auth=("admin", "password"))
```

Note that you need to explicitly call login to actually log in to the system:

```
>>> system.login()
<Response [200]>
```

Another way authentication information can be provided is through an .ini file. Create a file named ~/.infinidat/infinisdk.ini, with the following structure:

```
[infinibox]
username=admin
password=password
```

Now constructing an InfiniBox object will use the credentials above by default. You can also specify authorization for specific system, by adding sections to the .ini file titled infinibox:<system name>:

```
[infinibox] # will be used for default
username=defaultlogin
password=defaultpassword

[infinibox:system01] # will be used for interacting with the InfiniBox named 'system01'
→ username=other
password=otherpassword
```

2.1.4 Logging

InfiniSDK uses [Logbook](#) for logging, and by default all logs are emitted to the standard error stream.

The emitted logs also include the full debug outputs of the API calls made to the system, which might be a bit too much in some cases, overflowing your console unnecessarily. If you prefer less verbosity, you can set up a different logging scheme. For instance, the following code will only emit INFO logs to the console:


```
>>> import logbook
>>> import sys
>>> with logbook.NestedSetup([
...     logbook.NullHandler(),
...     logbook.StreamHandler(sys.stderr, level=logbook.INFO) ]):
...     pass # your code here
```

See also:

[Logbook's documentation](#)

2.1.5 Approving Dangerous Operations

By default, InfiniSDK performs operations regardless of the level of caution required for them. When a user uses a CLI or a GUI, Infinidat products often require confirmation before carrying out some dangerous operations requiring extra attention.

If you want your script to interactively ask the user for confirmation for such operations, use the `set_interactive_approval()` method:

```
>>> system.api.set_interactive_approval()
```

You can also turn off approvals temporarily, causing your script to fail with an exception in case dangerous operations are about to be carried out:

```
>>> with system.api.get_unapproved_context():
...     pass # operations here
```

See also:

`get_unapproved_context()`, `set_interactive_approval()`

2.1.6 Representing API Entities

InfiniSDK provides reflection for objects or entities defined on the system in the form of Pythonic objects. This makes creation, deletion and manipulation of objects easier. Supported objects are defined as Python classes such as `infinisdk.infinibox.volume.Volume` or `infinisdk.infinibox.pool.Pool`, and are accessed more easily through **collection proxies**, such as `system.volumes`, `system.pools` etc. For each supported object type X, there exists `system.Xs`.

The following examples illustrate how to use those proxies.

2.1.7 Creating Objects

Creation of objects can be done easily via the `create` method. InfiniSDK provides defaults for all required fields that can be autogenerated. For instance, creating a pool can be done via `system.pools.create()`:

```
>>> pool = system.pools.create()
```

Note: the `create` shortcut used above is a very thin wrapper around *the create method of the Pool class*. All it does is automatically assign the “right” system to the first argument.

2.1.8 Object Attributes

Once an object is obtained (either by creation or querying as described further down), it can be inspected for its attributes or manipulated in various ways. This is done using getter/setter methods. For most used names, there are direct setters and getters:

```
>>> pool.update_name('new_name')
>>> pool.get_name() == 'new_name'
True
```

All fields can be accessed via the `SystemObject.get_field()` / `SystemObject.update_field()` methods:

```
>>> pool.update_field('name', 'yet_another_name')
>>> pool.get_field('name') == 'yet_another_name'
True
```

2.1.9 Caching

Whenever an object attribute is fetched, it is cached for later use. By default, getting fields always fetches them from the cache of the requested object.

In case you need to fetch an up-to-date value for a field, there are several options:

1. Use `from_cache=False`:

```
>>> print(pool.get_field('name', from_cache=False))
yet_another_name
```

The above forces InfiniSDK to fetch the name from the system regardless of the cache

2. Disable caching completely:

```
>>> system.disable_caching()
```

2.1.10 Storage Capacity Handling

InfiniSDK reflects data sizes using the `capacity` module, allowing easy computations and manipulations of data sizes, including units:

```
>>> from capacity import GiB

>>> size = pool.get_virtual_capacity()
>>> print(size)
1 TB
>>> print(size * 2)
2 TB
>>> print(size // GiB)
931
```

See also:

[Documentation for the capacity module](#)

2.1.11 Querying Objects

Querying objects of various types is done relatively easily through InfiniSDK. The InfiniBox system exposes collection proxies, which provide iteration and filtering. Here's an example of querying all volumes on a system:

```
>>> system.volumes.count()
0

>>> system.volumes.to_list()
[]
```

See also:

Querying Objects

2.1.12 Deleting Objects

Deleting objects can be done by the `delete` method, which is available for the vast majority of the object types.

```
>>> host = system.hosts.create()
>>> host.delete() # <-- host gets deleted
```

Note: The `delete` method usually doesn't take care of indirect deletion needed to fulfill the request (like deleting volumes inside pools). This is a design decision that has been made to prevent unintended operations from being unwittingly made on the user's behalf.

2.1.13 Accessing HTTP/REST API Directly

InfiniSDK supports calling the HTTP/REST API of the system directly:

```
>>> response = system.api.get('system/product_id')
```

The above accesses `/api/rest/system/product_id`. `API.get()`, `API.post()`, `API.delete()` and `API.put()` all return *Response* objects. Results can be fetched by `Response.get_result()`:

```
>>> print(response.get_result())
INFINIBOX
```

You can always access the response belonging to *requests* through `.response`:

```
>>> response.response.status_code
200
```

By default, requests are checked for success. This behavior can be overridden by providing `assert_success=False`:

```
>>> response = system.api.get('nonexistent/path', assert_success=False)
>>> response.response.status_code
404
```

2.2 Querying Objects

The InfiniBox API layer allows its users to query and sort objects according to various criteria. InfiniSDK offers a clean Pythonic syntax to perform such queries.

2.2.1 Querying All Objects

Querying all objects can be done by iterating over the collection proxies (e.g. `system.volumes`):

```
>>> system.volumes.count()
5
>>> for volume in system.volumes:
...     print("Found volume:", volume.get_name())
Found volume: vol0
Found volume: vol1
Found volume: vol2
Found volume: vol3
Found volume: vol4
```

Note: This is also equivalent to iterating over `system.volumes.find()`

2.2.2 Querying by Fields

Querying by fields is relatively easy if you want a specific field value:

```
>>> [v] = system.volumes.find(name='vol0')
>>> v
<...:Volume id=...>
```

2.2.3 Getting a Single Object

Getting a single object has an even easier shortcut – `get`, which assumes only one object is returned:

```
>>> v = system.volumes.get(name='vol0')
>>> v # doctest: +ELLIPSIS
<...:Volume id=...>
```

It will fail if either 0 or several objects are returned:

```
>>> system.volumes.get()
Traceback (most recent call last):
...
TooManyObjectsFound

>>> system.volumes.get(name='nonexistent')
Traceback (most recent call last):
...
ObjectNotFound
```

There is also `safe_get`, returning `None` instead of raising an exception if no object is found:

```
>>> system.volumes.safe_get(name='nonexistent') is None
True
```

2.2.4 Advanced Queries

Object fields can be used to perform more complex queries, using operators. For instance, here is a query for all volumes whose name is not 'vol1'.

```
>>> for v in system.volumes.find(system.volumes.fields.name != 'vol1'):
...     print(v.get_name())
vol0
vol2
vol3
vol4
```

The above code leverages Python's operator overloading to generate on-the-fly query filters. There is also a shorter syntax for writing the above piece of code, using the `Q` shortcut available from InfiniSDK:

```
>>> from infinisdk import Q

>>> vols = system.volumes.find(Q.name != 'vol1')
>>> len(vols)
4
```

Note: `Q.x != y` is merely a shortcut for `SomeObject.fields.x != y`, and provides no other additional functionality besides shortening the syntax.

Note: You can use the following operators `==`, `!=`, `>`, `>=`, `<`, `<=`, and also `.in(...)`, `.not_in(...)`, `.between(x, y)` and `.like("string")`. Not all operators are supported by all queries – some limitations might apply.

And here is a query to find all volumes greater than 1 GiB in size:

```
>>> from capacity import GiB
>>> system.volumes.find(system.volumes.fields.size > GiB).to_list()
[]
```

See also:

Storage Capacity Handling

2.3 Efficient Querying

2.3.1 The relationship with InfiniBox REST API

InfiniSDK translates the operations on collections and object into REST API. If the script you write will handle a large set of objects, then it is important to use the InfiniSDK in the correct way, so that the APIs are as efficient as possible. Failure to do so, will cause your script to run slowly and for the increased overhead on the InfiniBox system.

Your script will probably contain snippets such as

```
from infinisdk import InfiniBox, Q
system = InfiniBox("systemname", auth=("user", "password"))
system.login()

for vol in system.volumes:
    print("Volume: {} is {}".format(vol.get_name(), vol.get_size()))
```

This will translate into a REST API request that returns **all** the data for **all** the volumes and volume snapshots on the system. Perhaps this is what you intend to do, perhaps not...

If you know in advance that your script only requires access to **some** of the objects in the system, you can provide this information via InfiniSDK to make the REST API more efficient.

2.3.2 Logging the REST API the script uses

We recommend that you familiarize yourself with the InfiniBox REST API, which will help you identify potential performance improvements as you run scripts.

To see the API requests, use the following code snippet which adds a hook when InfiniSDK executes a REST API request:

```
import gossip
@gossip.register('infinidat.sdk.before_api_request', token='api_dump')
def pre_api(request):
    print(f"Request: {request} {request.url}")
```

This snippet prints the API to the standard error (you can send it to a log file as well), for example:

```
for vol in system.volumes:
    print(f"Volume: {vol.get_name()} is {vol.get_size()}")
```

You can see the request URL, which retrieves all the volumes and volume snapshots.

2.3.3 Improvement #1: Always use the find() function when querying for objects

The find() function, which is available for all InfiniBox objects, allows you to control multiple aspects of the REST API that InfiniSDK generates.

Always use this function when enumerating objects. For example:

```
for vol in system.volumes.find():
    print(f"Volume: {vol.get_name()} is {vol.get_size()}")
```

2.3.4 Improvement #2: Fetching necessary fields only

By default, InfiniSDK will request the entire object from the InfiniBox system, and cache all the fields retrieved in memory. Typically, the object may contain many fields which your script doesn't need. Fetching these fields will increase the overhead on the InfiniBox system, on the network (because the JSON document is large) and increase the memory footprint of your script.

It is recommended to retrieve and **only** the fields you need, by using the only_fields function:

```
for vol in system.volumes.find().only_fields(["name", "size"]):
    print(f"Volume: {vol.get_name()} is {vol.get_size()}")
```

Now the API request and response include the only the specific two fields required (name and size).

2.3.5 Improvement #3: Fetching all necessary fields

If you trim down the requests to include specific fields, it is important to include **all** the fields your script needs. If you fail to do so, your script will still function correctly since InfiniSDK will issue subsequent requests to retrieve these missing fields, but the operation will be **very** ineffiect. For example:

```
for vol in system.volumes.find().only_fields(["name"]):
    print(f"Volume: {vol.get_name()} is {vol.get_size()}")
```

As you can see, the initial request retrieves only the name for all the volumes. Since the script then needs the volume name, InfiniSDK issues a specific request for the name of each object, separately.

Avoid this as much as possible.

2.3.6 Improvement #4: Retrieve only necessary objects

If your script only requires a subset of objects, use the find() function to filter just the objects as much as possible.

The simplest way to do this is to use the Q.field format. Here are some examples:

```
from infinisdk import Q
from capacity import *

system.volumes.find(Q.provisioning=="THICK")
system.volumes.find(Q.type!="SNAPSHOT")
system.volumes.find(Q.name.like("Database"))
system.volumes.find(Q.size>=100*GiB)
system.volumes.find(Q.pool.in_(["gil-pool", "chen-pool"]))
```

For example, if your script doesn't need volume snapshots you can use the filter `Q.type!="SNAPSHOT"` as a parameter to find():

```
for vol in system.volumes.find(Q.type!="SNAPSHOT").only_fields(["name", "size"]):
    print(f"Volume: {vol.get_name()} is {vol.get_size()}")
```

Now the API request contains a filter the will refrain from retrieving snapshots, instead of the following **inefficient** code:

```
for vol in system.volumes.find().only_fields(["name", "size", "type"]):
    if (vol.get_type() != "SNAPSHOT"):
        print(f"Volume: {vol.get_name()} is {vol.get_size()}")
```

If you need to filter according to multiple fields, add more filters to the find() function.

For exaemple, to list only volumes (no snapshots) whose name begins with "Database" add the `Q.name.like("Database")` paramter after `Q.type!="SNAPSHOT"`:

```
for vol in system.volumes.find(Q.type!="SNAPSHOT", Q.name.like("Database")).only_
    ↪fields(["name", "size"]):
    print(f"Volume: {vol.get_name()} is {vol.get_size()}")
```

2.3.7 Improvement #5: Retrieve as many object with each API request as possible

InfiniBox REST API has built-in paging capabilities, which InfiniSDK uses automatically. By default InfiniSDK uses a page size of 50, which means every API request returns at most 50 objects. If the query you run has more objects InfiniSDK issues multiple API requests (each one returns 50 objects) until all the list is exhausted.

Note: this is the default behavior, unless you add the `page()` function as shown in the next improvement.

It is recommended to use larger page sizes: this will minimize the communication and overhead, and has practically no downsides (unless you retrieve large objects with many fields and text). For example:

```
for vol in system.volumes.find(Q.type!="SNAPSHOT").only_fields(["name", "size"]).page_
    size(1000):
    print(f"Volume: {vol.get_name()} is {vol.get_size()}")
```

The above retrieves 1000 volumes (or less if there are fewer volumes) in a single API request.

2.3.8 Improvement #6: Retrieve the top-most objects

Sometimes the script only needs the first (or last) objects based on some order. For example, you might want to display the 5 oldest snapshots of volume “Database1”.

Doing this efficiently requires the combination of the **sorting** and **paging** capabilities in the REST API.

Use the `sort()` function to indicate the field(s) by which you want the objects, the `page_size()` function to indicate how many objects you want, and use the `page()` function to limit the retrieval to one page. The above example can be achieved thus:

```
for vol in system.volumes.find(Q.parent_id==1615).only_fields(["name", "size"]).sort(Q.
    created_at).page_size(5).page(1):
    print(f"Volume: {vol.get_name()} is {vol.get_size()}")
```

The above example will use a single REST API request to retrieve 5 objects - the most **efficient** way to do that.

Note: paging in InfiniBox is limited to 1000 objects at most, so if you need more you will need to repeat this with `page(1)`, then `page(2)`, etc.

2.3.9 Improvement #7: Retrieve a single object

If you know you’re going to get a single object, there is a quick and simple way to do that, using the `get()` function. For example, to find a volume by name:

```
system.volumes.get(Q.name=='my-volume-name')
```

This is essentially the same as `find(<predicate>)[0]`, plus the necessary exceptions if no objects meet the predicate or more than one object does.

2.4 The InfiniBox Object

2.4.1 Getting System Name and Serial Number

The system name and serial numbers can be obtained directly from the `infinisdk.infinibox.InfiniBox` object:


```
>>> system_name = system.get_name()
>>> system_serial = system.get_serial()
```

2.4.2 Getting the System Model Name

The `infinisdk.infinibox.InfiniBox.get_model_name()` method retrieves the model information, as reported by the system:

```
>>> isinstance(system.get_model_name(), str)
True
```

See also:

`infinisdk.infinibox.InfiniBox`

2.5 System Capacity

InfiniSDK allows inspecting the capacity parameters of the system.

`system.capacities` is a container for the different system capacity attributes

```
>>> print('System has {} total physical capacity'.format(system.capacities.get_total_
↳physical_capacity()))
System has 2.3 PiB total physical capacity

>>> print('System has {} free physical capacity'.format(system.capacities.get_free_
↳physical_capacity()))
System has 2.3 PiB free physical capacity

>>> print('System has {} total virtual capacity'.format(system.capacities.get_total_
↳virtual_capacity()))
System has 2.3 PiB total virtual capacity

>>> print('System has {} free virtual capacity'.format(system.capacities.get_free_
↳virtual_capacity()))
System has 2.3 PiB free virtual capacity

>>> print('System has {} total allocated physical capacity'.format(system.capacities.
↳get_total_allocated_physical_capacity()))
System has 0 bit total allocated physical capacity

>>> print('System has {} dynamic spare drive cost'.format(system.capacities.get_
↳dynamic_spare_drive_cost()))
System has 0 dynamic spare drive cost

>>> print('System has {} used dynamic spare partitions'.format(system.capacities.get_
↳used_dynamic_spare_partitions()))
System has 0 used dynamic spare partitions

>>> print('System has {} used dynamic spare capacity'.format(system.capacities.get_
↳used_dynamic_spare_capacity()))
System has 0 bit used dynamic spare capacity

>>> print('System has {} used spare partitions'.format(system.capacities.get_used_
↳spare_partitions()))
```

(continues on next page)

(continued from previous page)

```
System has 0 used spare partitions

>>> print('System has {} used spare capacity'.format(system.capacities.get_used_spare_
↪capacity()))
System has 0 bit used spare capacity

>>> print('System has {} total spare partitions'.format(system.capacities.get_total_
↪spare_partitions()))
System has 3168 total spare partitions

>>> print('System has {} total spare capacity'.format(system.capacities.get_total_
↪spare_capacity()))
System has 43.66 TiB total spare capacity
```

2.6 Pools

2.6.1 Creating Pools

Creating pools is done with `system.objects.pools.create`:

```
>>> pool = system.pools.create()
```

You can also specify physical and virtual capacity:

```
>>> pool = system.pools.create(physical_capacity=TiB, virtual_capacity=TiB)
```

2.6.2 Updating Pools

Updating fields such as name and capacities are done like any other object update operations in InfiniSDK:

```
>>> pool.update_name('new_name')
>>> pool.update_physical_capacity(pool.get_physical_capacity() * 2)
```

2.6.3 Deleting Pools

Deleting a pool is done using `Pool.delete()`:

```
>>> pool.delete()
>>> pool.is_in_system()
False
```

2.6.4 Administered Pools

Use `infinisdk.infinibox.pool.PoolBinder.get_administered_pools()` to obtain the list of pools the current user can administer:

```
>>> pools = system.pools.get_administered_pools()
```

2.7 Volumes

2.7.1 Creating and Modifying Volumes

Creating volumes is done with the `create` method:

```
>>> v = system.volumes.create(pool=pool, name='my_vol')
```

Note: When a size is not explicitly stated, a default of 1 GiB is used. You can also provide the size explicitly:

```
>>> from capacity import GiB
>>> vol = system.volumes.create(pool=pool, size=1*GiB)
```

It is also possible to create multiple volumes with a single line, by calling `.create_many`:

```
>>> vols = system.volumes.create_many(pool=pool, name='vol', count=5)
>>> len(vols)
5
>>> for vol in vols:
...     print(vol.get_name())
vol_1
vol_2
vol_3
vol_4
vol_5
```

We can now access various attributes of the volume:

```
>>> print(v.get_name())
my_vol
>>> v.get_size()
1*GB
```

2.7.2 Volume Serials

InfiniSDK exposes the volume WWN serial number through a custom type, enabling you to parse it easier:

```
>>> serial = v.get_serial()
>>> print(serial)
742b0...
>>> serial.ieee_company_id
7613199
>>> unused = serial.system_id
...
```

See also:

SCSISerial

2.7.3 Moving Between Pools

Use `Volume.move_pool()` to move a volume between pools:

```
>>> new_pool = system.pools.create()
>>> v.move_pool(new_pool)
```

2.7.4 Deleting Volumes

Deleting a volume is done with `Volume.delete()`:

```
>>> v.delete()
```

2.7.5 Example: Deleting All Volumes with Specific Name Prefix

```
>>> for volume in system.volumes:
...     if volume.get_name(from_cache=True).startswith('prefix'):
...         volume.delete()
```

See also:

Volume API documentation

2.8 Filesystems

2.8.1 Creating a Filesystem

Creating filesystems is done with the `create` method:

```
>>> from capacity import GiB, GB
>>> my_fs = system.filesystems.create(pool=pool, name='my_fs', size=GiB)
```

Note: When a size is not explicitly stated, a default of 1 GiB is used. You can also provide the size explicitly:

```
>>> fs = system.filesystems.create(pool=pool, size=1*GiB)
```

You can optionally specify the `security_style` in the `create` method:

```
>>> fs = system.filesystems.create(pool=pool, security_style="WINDOWS")
```

It is also possible to create multiple filesystems with a single line, by calling `.create_many`:

```
>>> filesystems = system.filesystems.create_many(pool=pool, name='fs', count=5)
>>> len(filesystems)
5
>>> for fs in filesystems:
...     print(fs.get_name())
fs_1
fs_2
fs_3
fs_4
fs_5
```

We can now access various attributes of the filesystem:

```
>>> print(my_fs.get_name())
my_fs
>>> my_fs.get_size()
1*GiB
```

2.8.2 Moving Between Pools

Use `Filesystem.move_pool()` to move a filesystem between pools:

```
>>> new_pool = system.pools.create()
>>> fs.move_pool(new_pool)
```

2.8.3 Resizing Filesystems

Use `Filesystem.resize()` to resize the filesystem by the given delta:

```
>>> fs.resize(delta=2*GB)
```

2.8.4 Deleting Filesystems

Deleting a filesystem is done with `Filesystem.delete()`:

```
>>> fs.delete()
```

2.8.5 Example: Deleting All Filesystems with Specific Name Prefix

```
>>> for fs in system.filesystems:
...     if fs.get_name(from_cache=True).startswith('prefix'):
...         fs.delete()
```

See also:

Filesystem API documentation

2.9 NFS Exports

2.9.1 Creating a Filesystem Export

A filesystem export is created with default settings and advanced setting. For a detailed documentation of these settings, Read more [Here](#).

```
>>> export = fs.add_export()
```

We can now access and modify various attributes of the export:

```
>>> from capacity import MiB
>>> export.get_max_read()
1*MiB
>>> export.update_max_read(2*MiB)
>>> export.get_max_read()
2*MiB
```

2.9.2 Disabling an Export

Following this operation, the filesystem is not accessible by the user. The export path is not deleted, and can be enabled.

```
>>> export.disable()
>>> export.is_enabled()
False
```

2.9.3 Enabling an Export

```
>>> export.enable()
>>> export.is_enabled()
True
```

2.9.4 Querying for Filesystem Exports

Like other InfiniBox collections, InfiniSDK provides iteration and filtering ability for exports.

```
>>> system.exports.count()
1
```

2.9.5 Export Permissions

Export permissions can be modified with `.Export.update_permissions`.

This method overrides current permissions.

To preserve current permission settings, first use `.Export.get_permissions`, then update accordingly.

```
>>> from munch import Munch
>>> permissions = export.get_permissions()
>>> permissions[0] == Munch({'access': 'RW', 'no_root_squash': True, 'client': '*'})
True
>>> export.update_permissions(permissions +
...     [{'access': 'RO', 'client': '1.1.1.1', 'no_root_squash': True}])
>>> permissions = export.get_permissions()
>>> permissions[0] == Munch({'access': 'RW', 'no_root_squash': True, 'client': '*'})
True
>>> permissions[1] == Munch({'access': 'RO', 'no_root_squash': True, 'client': '1.1.1.
↪1'})
True
```

(continues on next page)

(continued from previous page)

```
>>> export.update_permissions([{'access': 'RW', 'client': '2.2.2.2', 'no_root_squash': True}])
>>> permissions = export.get_permissions()
>>> permissions[0] == Munch({'access': 'RW', 'no_root_squash': True, 'client': '2.2.2.2'})
True
```

2.9.6 Deleting an Export

Deleting an export is done with `.Export.delete`:

```
>>> export.delete()
```

2.9.7 Tree Quotas

Each filesystem has a `treeqs` member, which is a collection of its TreeQ's:

```
>>> fs.treeqs.to_list()
[]
>>> treeq1 = fs.treeqs.create(path='/path1', soft_capacity=GB, hard_inodes=200)
>>> print(treeq1.get_capacity_state())
BELOW_SOFT
```

TreeQ's can be queried, filtered and sorted:

```
>>> treeq2 = fs.treeqs.create(path='/path2', soft_capacity=GB, hard_inodes=300)
>>> treeq3 = fs.treeqs.create(path='/path3', soft_capacity=GB, hard_inodes=400)
>>> from infinisdk import Q
>>> for treeq in fs.treeqs.find(Q.hard_inodes>200).sort(-fs.treeqs.fields.hard_inodes):
...     print(treeq.get_path())
/path3
/path2
```

When creating a snapshot, all TreeQ's are copied to the child dataset. The new TreeQ's are separate and distinct objects:

```
>>> fs2 = fs.create_snapshot()
>>> for treeq in fs2.treeqs:
...     print(treeq.get_path())
/path1
/path2
/path3
>>> fs.treeqs.get(path='/path1') == fs2.treeqs.get(path='/path1')
False
```

A single update request can modify either TreeQ limits or its name:

```
>>> treeq1.update_fields(soft_inodes=5, soft_capacity=GB)
>>> treeq1.update_fields(name='path1')
>>> treeq1.update_fields(soft_inodes=5, name='path1')
Traceback (most recent call last):
...
APICommandFailed: ...
```

Refreshing a snapshot or restoring a filesystem from a snapshot modifies its TreeQ's accordingly:

```
>>> treeq1.delete()
>>> print(fs.treeqs.count())
2
>>> fs.restore(fs2)
>>> print(fs.treeqs.count())
3
>>> print(fs2.treeqs.count())
3
>>> fs.treeqs.choose().delete()
>>> fs2.refresh_snapshot()
>>> print(fs2.treeqs.count())
2
```

2.10 SMB Shares

2.10.1 Creating a Share

Create a share using the `add_share` method on the `infinisdk.infinibox.filesystem.Filesystem` object:

```
>>> fs = system.filesystems.create(
...     pool=pool,
...     security_style="WINDOWS"
... )
>>> share = fs.add_share()
>>> share in fs.get_shares()
True
```

2.10.2 Share Permissions

Permissions can be accessed with the `permissions` field:

```
>>> perm = share.permissions.create(sid="S-1-1-1", access="NONE")
>>> perm.update_access("FULLCONTROL")
>>> perm.get_access()
FULLCONTROL
>>> share.permissions.get(sid="S-1-1-1") == perm
True
>>> perm in share.permissions.to_list()
True
>>> perm.delete()
>>> perm in share.permissions.to_list()
False
```

2.11 Working with Hosts, Clusters and Mappings

InfiniSDK provides an easy interface to query and manipulate volume mappings to hosts.

2.11.1 Creating Hosts

Creating hosts is the same like creating any other management object through InfiniSDK. Hosts are represented by the *Host* class:

```
>>> host = system.hosts.create(name='production01')
>>> host
<...:Host id=...>
>>> print(host.get_name())
production01
```

2.11.2 Adding/Removing FC Ports

Adding and removing FC ports can be done with *infinisdk.infinibox.host.Host.add_port()* and *infinisdk.infinibox.host.Host.remove_port()*. The address should be an instance of the *infi.dtypes.wwn.WWN* class to denote an FC address:

```
>>> from infi.dtypes.wwn import WWN
>>> address = WWN('00:01:02:03:04:05:06:07')
>>> host.add_port(address)
>>> host.remove_port(address)
```

2.11.3 Adding/Removing iSCSI IQNs

Adding and removing iSCSI IQNs is done in a fashion similar to FC ports, only that the address in this case should be an instance of the *infi.dtypes.iqn.iSCSIName* class:

```
>>> from infi.dtypes.iqn import make_iscsi_name
>>> address = make_iscsi_name('iqn.1994-05.com.redhat:8f8dcc647276')
>>> host.add_port(address)
>>> host.remove_port(address)
```

2.11.4 Querying Host by a Defined Port

You can quickly check if a system has a host *system.hosts.get_host_id_by_initiator_address*, *system.hosts.get_host_by_initiator_address* and *system.hosts.has_registered_initiator_address*:

```
>>> system.hosts.has_registered_initiator_address(address)
False
>>> host.add_port(address)
>>> system.hosts.get_host_by_initiator_address(address) == host
True
```

2.11.5 Mapping and Unmapping Volumes and Snapshots

Given a volume object, we can easily map it to a host:

```
>>> lu = host.map_volume(volume)
```

The *returned lu object* represents the volume mapping to the specific host, and it can be used to retrieve information about the mapping:

```
>>> print(int(lu))
1
```

Unmapping can be done in several ways. The easiest would be to call `Host.unmap_volume()`:

```
>>> host.unmap_volume(volume)
```

Which can also receive a specific LUN to unmap:

```
>>> lu = host.map_volume(volume, lun=2)
>>> host.unmap_volume(lun=2)
```

The LUN can also be deleted directly through its accessor object:

```
>>> lu = host.map_volume(volume)
>>> lu.unmap()
```

2.11.6 Querying Volume Mappings

Iterating over available mappings of a host is fairly simple:

```
>>> lu = host.map_volume(volume, lun=5)

>>> host.get_luns()
<LogicalUnitsContainer: [<LUN 5: <...:Host id=...>-><...:Volume id=...>>]>

>>> for lun in host.get_luns():
...     print("{} is mapped to {}".format(lun, lun.volume))
<LUN 5: <...:Host id=...>-><...:Volume id=...> is mapped to <...:Volume id=...>
```

There is also a shortcut to iterate over all mappings in the entire system:

```
>>> for lun in system.luns:
...     print("{} belongs to {} and is mapped to {}".format(lun, lun.mapping_object,
↳ lun.volume))
<LUN 5: <...:Host id=...>-><...:Volume id=...> belongs to <...:Host id=...> and is_
↳ mapped to <...:Volume id=...>
```

Here is a code snippet to unmap all volumes in the system that contain ‘to remove’ in their names:

```
>>> import itertools

>>> volume.update_name('this is a volume to remove')

>>> for mapping_object in itertools.chain(system.host_clusters, system.hosts):
...     for lun in mapping_object.get_luns():
...         if 'to remove' in lun.volume.get_name():
...             print("Unmapping", lun.volume)
...             lun.unmap()
Unmapping <...:Volume id=...>
```

Of course there is a much more convenient shortcut for unmapping a volume from all hosts, using the `Volume.unmap()` shortcut:

```
>>> lu = host.map_volume(volume)
>>> host.is_volume_mapped(volume)
True
>>> volume.unmap()
>>> host.invalidate_cache()
>>> host.is_volume_mapped(volume)
False
```

2.11.7 Clusters and Hosts

Manipulating clusters is done with the `infinisdk.infinibox.host_cluster.HostCluster` class:

```
>>> cluster = system.host_clusters.create()
>>> cluster.add_host(host)

>>> lu = cluster.map_volume(volume)

>>> host.invalidate_cache()
>>> [host_lu] = host.get_luns()

>>> host_lu
<LUN 11: <...:Host id=...>-><...:Volume id=...>>

>>> host_lu.is_clustered()
True
```

See also:

- [Host API documentation](#)
- [Cluster API documentation](#)

2.12 Snapshots

2.12.1 Creating Snapshots

Use the `.create_snapshot`

```
>>> snap = volume.create_snapshot()
>>> snap_of_snap = snap.create_snapshot()
```

2.12.2 Creating Group Snapshots

You can create a group of snapshots (not to be confused with *Consistency Groups*) using `create_group_snapshot()`:

```
>>> v1, v2, v3 = volumes = [system.volumes.create(pool=pool) for i in range(3)]
>>> s1, s2, s3 = system.volumes.create_group_snapshot(volumes)
```

2.12.3 Querying Snapshots

The parent of a snapshot is accessed through the `snap.get_parent/vol.get_parent` method:

```
>>> snap.get_parent() == volume
True

>>> volume.get_parent() is None
True
```

You can inspect the snapshot's creation time:

```
>>> creation_time = snap.get_creation_time()
>>> delta = current_time - creation_time
>>> delta.days
15
```

Note: Time is represented in InfiniSDK with [Arrow objects](#). See the [relevant documentation](#) for more details on how to use and manipulate these values.

2.12.4 Example: Deleting Snapshots by Creation Time

```
>>> cutoff = current_time.shift(days=-10)
>>> for snapshot in system.volumes.find(system.volumes.fields.created_at < cutoff,
↳parent_id=volume.id):
...     print("Deleting snapshot with id:", snapshot.id)
...     snapshot.delete()
Deleting snapshot with id: ...
```

See also:

Volume API documentation

2.13 Consistency Groups

InfiniSDK allows creating, adding/removing members and manipulating consistency groups.

2.13.1 Creating Consistency Groups

Consistency groups are created just like all InfiniSDK objects, through the `create` method:

```
>>> cg = system.cons_groups.create(pool=pool)
```

2.13.2 Adding and Removing Volumes

Use the `ConsGroup.add_member()` method to add members to a consistency group:

```
>>> cg.add_member(volume)
```

Use the `ConsGroup.remove_member()` method to remove members from a consistency group:

```
>>> cg.remove_member(volume)
```

2.13.3 Creating Snapshot Groups

You can create a snapshot group from a consistency group through the `ConsGroup.create_snapshot()` method:

```
>>> cg.add_member(volume) # snap group creation is not allowed for empty CGs
>>> sg = cg.create_snapshot()
>>> sg.get_parent() == cg
True
```

2.13.4 Restoring from Snapshot Group

Restoring a snapshot group is done with the `ConsGroup.restore()` method:

```
>>> cg.restore(sg)
```

2.13.5 Deleting Consistency Groups

Deleting consistency groups is done through `ConsGroup.delete()`:

```
>>> sg.delete()
>>> cg.delete()
```

2.14 User Management

User management in InfiniSDK is done mostly via `system.users` and `system.groups`.

2.14.1 Users

Getting

Getting all users in a system

```
>>> users = system.users.to_list()
```

Getting a user by name

```
>>> user = system.users.get(name='someuser')
>>> print(user.get_name())
someuser
```

Creating and Deleting Users

Use `infinibox.users.create` to create new users:

```
>>> user = system.users.create(name='testuser', password='testpassword')
```

Deleting users is done like any other InfiniSDK object, using `User.delete()`:

```
>>> user.delete()
```

Modifying Users

You can modify users configured on the system using any of the `User` class:

```
>>> user = system.users.create(name='testuser', password='testpassword')
>>> user.update_password('12345678')
>>> user.update_name('testuser2')
```

Setting User Roles

You can set a user's role using `User.update_role()`:

```
>>> user.update_role('PoolAdmin')
>>> print(user.get_role())
POOL_ADMIN
```

Setting Pool Owners

To set a pool that will be administered by a user, simply call `Pool.set_owners()`:

```
>>> pool = system.pools.create()
>>> pool.set_owners([user])
```

2.14.2 LDAP Integration

Getting all current LDAP configs:

Setting up LDAP integration is done in two main steps. First, we need to define our LDAP settings:

```
>>> ldap_config = system.ldap_configs.define(name='AD2K3.local', domain_name='AD2K3.
↪local', bind_username='Administrator', bind_password='passwd')
```

Once the LDAP directory is defined, we need to map the LDAP group to a local role:

```
>>> group = ldap_config.create_group(name='group01', dn='group01', role='PoolAdmin')
>>> print(group.get_role())
POOL_ADMIN
```

Updating LDAP Configuration

Updating LDAP configurations can be easily done with `LDAPConfig.modify()`:

```
>>> ldap_config.modify(schema_group_class='group')
>>> ldap_config.update_name('some_new_name')
```

Testing LDAP Configuration

```
>>> ldap_config.test()
```

Updating LDAP Configuration Priority Order

```
system.ldap_configs.set_order([ldap_config, ldap_config2, ...])
```

Reloading/Refreshing LDAP Cache

```
>>> system.ldap_configs.reload()
```

Or:

```
>>> system.ldap_configs.flush_cache()
```

Deleting LDAP Configurations

```
>>> ldap_config.delete()
```

2.15 System Configuration

2.15.1 SMTP Configuration

Querying SMTP notification targets:

```
>>> for smtp_target in system.notification_targets.find(protocol='SMTP'):
...     pass
```

Modifying SMTP notification targets:

```
>>> smtp_target = system.notification_targets.find(protocol='SMTP').to_list()[0]
>>> smtp_target.update_name('sample_config_1')

>>> smtp_target.update_host('mailserver.lab.com')
>>> smtp_target.update_port(25)
>>> smtp_target.update_username('username')
>>> smtp_target.update_password('password')
>>> smtp_target.update_from_address('username@domain.com')
>>> smtp_target.enable_tls()
```

Testing SMTP notification targets:

```
>>> resp = smtp_target.test(recipients=['someuser@domain.com'])
```

2.15.2 SNMP Configuration

Creating SNMP targets:

```
>>> snmp_target = system.notification_targets.create(
...     name='snmp_target', protocol='SNMP', host='somehost', private_key='private',
...     username='user', password='password',
...     private_protocol='AES',
...     version='SNMPv3', engine='0x100000000', auth_type='AuthPriv', auth_protocol=
...     ↪ 'MD5')
```

Querying SNMP targets:

```
>>> for snmp_target in system.notification_targets.find(protocol='SNMP'):
...     pass
```

Modifying SNMP targets:

```
>>> snmp_target.update_host('hostname')
>>> snmp_target.update_username('username')
>>> snmp_target.update_password('password')
>>> snmp_target.update_version('SNMPv3')
>>> snmp_target.update_auth_protocol('MD5')
>>> snmp_target.update_auth_type('AuthPriv')
```

Testing SNMP target:

```
>>> resp = snmp_target.test()
```

Deleting SNMP targets:

```
>>> snmp_target.delete()
```

2.15.3 RSyslog Configuration

Creating RSyslog target:

```
>>> rsyslog_target = system.notification_targets.create(
...     host='hostname',
...     name='syslog_target', protocol='SYSLOG', transport='TCP', facility='local0')
```

Querying RSyslog targets:

```
>>> for rsyslog_target in system.notification_targets.find(protocol='SYSLOG'):
...     pass
```

Modifying RSyslog targets:


```
>>> rsyslog_target.update_name('some_target')
>>> rsyslog_target.update_host('hostname')
>>> rsyslog_target.update_transport('UDP')
>>> rsyslog_target.update_facility('local1')
```

Testing RSyslog targets:

```
>>> resp = rsyslog_target.test()
```

Deleting RSyslog targets:

```
>>> rsyslog_target.delete()
```

See also:

NotificationTarget

2.16 Network Configuration

2.16.1 Network Spaces

NAS, iSCSI, replication and other network-related features require a configured *network space* to operate. A network space defines a set of configured IP addresses, as well as additional network-related configuration, with which the system can operate.

Creating Network Interfaces

Network spaces are defined on top of *network interfaces*, which can be created using `system.network_interfaces.create`:

```
>>> interfaces = []
>>> for node in system.components.nodes:
...     interfaces.append(
...         system.network_interfaces.create(node=node, ports=['eth-datal']))
...     )
```

Creating Network Spaces

Once the network interfaces are defined, creating a network space can be done via `system.network_spaces.create`:

```
>>> interfaes = []
>>> netSPACE = system.network_spaces.create(
...     name='ns1',
...     interfaces=interfaces,
...     service='RMR_SERVICE',
...     mtu=9000,
...     network_config={
...         'netmask': 19,
...         'network': '192.168.0.0',
...         'default_gateway': '192.168.1.1',
...     },
... )
```

Setting Network Space Properties

Some network space types can receive additional configuration options through *properties*. These can be specified during network space creation through the `properties` parameter, and updated through `update_properties`:

```
namespace = system.network_spaces.create(..., properties: {'is_async_only': True})
...
namespace.update_properties({
    'is_async_only': False,
})
```

See also:

For more information regarding network space properties and their meaning, please refer to the official InfiniBox API documentation

2.17 Replication

2.17.1 Creating Async Replicas

Replicating a volume or a filesystem to a remote system (mirroring) can be done by calling the `replicate_entity` shortcut:

```
>>> pool = primary_system.pools.create()
>>> vol = primary_system.volumes.create(pool=pool)
>>> remote_pool = secondary_system.pools.create()
>>> replica = primary_system.replicas.replicate_entity(vol, link=link, remote_
↳ pool=remote_pool)
```

The default behavior for `ReplicaBinder.replicate_entity()` is to create the remote entity (receiving a remote pool as input). You can also use an existing remote entity through the `ReplicaBinder.replicate_entity_use_base()` shortcut:

```
replica = primary_system.replicas.replicate_entity_use_base(
    local_entity,
    link=link,
    local_snapshot=snap,
    remote_snapshot=remote_snap)
```

Replication creation requires the following arguments to be provided:

- A *local entity* (e.g. volume, filesystem or consistency group)
- A *link* (an instance of `infinisdk.infinibox.link.Link`, representing the network link to the remote system)

As for the remote entity, it depends on the scenario being used to create the replica:

- Using a base snapshot (`_use_base`) requires a remote and local snapshots
- Creating a new entity on the remote side (`_create_target` or default) requires the remote pool to be provided
- Creating over an existing, formatted target (`_existing_target`) requires the remote target to be provided via `remote_entity` parameter

See also:

`infinisdk.infinibox.replica.Replica`

Note: The type of the replica created (async/sync) is controlled by an optional parameter called `replication_type`. The default, if not specified, is "ASYNC".

2.17.2 Replicating Consistency Groups

Creating a CG replica is also straightforward, and is done via the `replicate_cons_group` method:

```
>>> cg = primary_system.cons_groups.create(pool=pool)
>>> replica = primary_system.replicas.replicate_cons_group(cg, link=link, remote_
↳pool=remote_pool)
```

2.17.3 Creating Synchronous Replicas

Creating synchronous replicas is done by specifying "SYNC" for the `replication_type` parameter during replica creation:

```
>>> pool = primary_system.pools.create()
>>> vol = primary_system.volumes.create(pool=pool)
>>> replica = primary_system.replicas.replicate_entity(
...     vol, link=link,
...     replication_type="SYNC", remote_pool=remote_pool)
```

2.17.4 Changing Replication Type

Changing the type of the replication to SYNC / ASYNC can be done by calling to `change_type_to_sync` / `change_type_to_async` respectively. The replica must not be in `INITIALIZING` state. For example:

```
>>> async_replica.change_type_to_sync()
>>> assert async_replica.is_type_sync()
>>> async_replica.change_type_to_async()
>>> assert async_replica.is_type_async()
```

2.17.5 VVOL Replication

InfiniBox release 7.1 and above supports replication of VMware Virtual Volumes (vVols). Using vVols replication, VM administrators can easily set up efficient array-based replication for their virtual machines (VMs) directly from the same VMware vCenter user interface from which they control all other aspects of their VMs.

Setting up vVols replication with InfiniBox consists of an initial setup, performed by the storage administrator, and an ongoing setup, performed by the VM administrator.

As part of the initial setup, the storage administrator defines vVols Replication Groups and replicas. For this process to succeed the user is expected to provide a *link* (an instance of `infinisdk.infinibox.link.Link`, representing the network link to the remote system). You can get the remote system from the link object by:

```
>>> link.get_linked_system()
```

2.17.6 Initial Setup for VVOL Replication

A storage administrator may create one or more vVols Replication Groups. A separate replica is created for each group.

- A vVols Replication Group contains multiple vVols, typically from a set of virtual machines
- The replica defines the target InfiniBox system where the replicated vVols will be available.

2.17.7 Creating Replication Group

To create a Replication Group (RG) you'll also need to create a *pool* (an instance of `infinisdk.infinibox.pool.Pool`) with `type="VVOL"`:

```
>>> vvol_pool = system.pools.create(name="pool1", type="VVOL")
>>> rg = system.replication_groups.create(pool=vvol_pool)
```

2.17.8 Creating a Replica (RgReplica)

In addition to the above you'll also need to create a remote pool with `type="VVOL"` for the remote system in the same way:

```
>>> remote_system = link.get_linked_system()
>>> remote_vvol_pool = remote_system.pools.create(name="pool1-remote", type="VVOL")
>>> from datetime import timedelta
>>> rg_replica = system.rg_replicas.create(link=link, sync_
↪ interval=timedelta(seconds=60), rpo=timedelta(seconds=120), remote_pool_id=remote_
↪ vvol_pool.get_id(), replication_group=rg)
```

Note: The `sync_interval` parameter controls how often the system will replicate the data (e.g. every 60 seconds)

Note: The `rpo` value is the Recovery Point Objective value and it represents the tolerance to data loss during the replication process. It should be greater than the `sync_interval` value. E.g. if this value is 120 seconds and `sync_interval` is 60 seconds then the system will replicate every 60 seconds and will raise an alert if there was an issue and the system missed 2 replication attempts (2 intervals).

2.18 Using Object Metadata

InfiniBox allows a client script to assign metadata keys and values to various objects, and query them later.

2.18.1 Setting Metadata Keys

```
>>> unused = volume.set_metadata('metadata_key', 'value!')
```

2.18.2 Getting Metadata Keys

```
>>> print(volume.get_metadata_value('metadata_key'))
value!
```

Getting nonexistent metadata, by default, raises an exception:

```
>>> volume.get_metadata_value('nonexisting')
Traceback (most recent call last):
...
APICommandFailed: ...
```

You can provide defaults to be retrieved if metadata doesn't exist:

```
>>> volume.get_metadata_value('nonexisting', 2)
2
```

2.18.3 Getting All Metadata Keys

Getting all metadata keys of specific object:

```
>>> for key, value in volume.get_all_metadata().items():
...     print("Found key:", key, "with value:", value)
Found key: metadata_key with value: value!
```

You can also get all metadata keys for all the object in the system:

```
>>> for object_metadata_item in volume.system.get_all_metadata():
...     print("Found key: {key} with value: {value} for object id {object_id}".
↪format(**object_metadata_item))
Found key: metadata_key with value: value! for object id ...
```

2.18.4 Deleting (Unsetting) Metadata

```
>>> unused = volume.unset_metadata('metadata_key')
>>> volume.get_all_metadata()
{}
```

You can also clear all metadata related to a single object:

```
>>> volume.clear_metadata()
```

See also:

infinisdk.infinibox.system_object.InfiniBoxObject

2.19 Events

InfiniSDK represents system events through the *system.events* collection, which contains *Event* objects. Querying system events can be done in several ways. We can, for instance, iterate over all events:

```
>>> for event in system.events:
...     print(event)
<...:Event id=1000, code=VOLUME_CREATED>
<...:Event id=1001, code=VOLUME_DELETED>
<...:Event id=1002, code=VOLUME_CREATED>
<...:Event id=1003, code=VOLUME_DELETED>
<...:Event id=1004, code=VOLUME_CREATED>
<...:Event id=1005, code=VOLUME_DELETED>
<...:Event id=1006, code=USER_LOGIN_SUCCESS>
```

Sorting is determined by the system by default, but we can easily change that. For instance, we can order the events by descending id:

```
>>> for event in system.events.find().sort(-system.events.fields.id):
...     print(event)
<...:Event id=1006, code=USER_LOGIN_SUCCESS>
<...:Event id=1005, code=VOLUME_DELETED>
<...:Event id=1004, code=VOLUME_CREATED>
<...:Event id=1003, code=VOLUME_DELETED>
<...:Event id=1002, code=VOLUME_CREATED>
<...:Event id=1001, code=VOLUME_DELETED>
<...:Event id=1000, code=VOLUME_CREATED>
```

We can also combine this with filtering. The following example filters by specific event code:

```
>>> for event in system.events.find(code='VOLUME_CREATED').sort(-system.events.fields.
↳ id):
...     print(event)
<...:Event id=1004, code=VOLUME_CREATED>
<...:Event id=1002, code=VOLUME_CREATED>
<...:Event id=1000, code=VOLUME_CREATED>
```

2.19.1 Example: Getting all Events Newer than a Specific Sequence Number

```
>>> from infinisdk import Q
>>> for e in system.events.find(Q.seq_num>=1004):
...     print(e)
<...:Event id=1004, code=VOLUME_CREATED>
<...:Event id=1005, code=VOLUME_DELETED>
<...:Event id=1006, code=USER_LOGIN_SUCCESS>
```

2.20 System Components

InfiniSDK allows inspecting the hardware components in the system, and obtaining various attributes about them.

2.20.1 Nodes

`system.components.nodes` is a collection of *infinisdk.infinibox.components.Node* objects:

```
>>> print('System has {} nodes'.format(system.components.nodes.count()))
System has 3 nodes
```

2.20.2 Drives

InfiniSDK provides several ways of querying the system's drive information, `system.components.enclosures` and `system.components.drives`. The first is intended for traversing the actual topology of the system through the *Enclosure* component, while the second is an aggregate of all drives in the system (*Drive* objects):

```
>>> for enclosure in system.components.enclosures:
...     for drive in enclosure.get_drives():
...         pass # <- do something with drive here

>>> for drive in system.components.drives:
...     pass # <- do something with drive here
```

You can also query drives by their attributes, for instance by state:

```
>>> from infinisdk import Q
>>> for drive in system.components.drives.find(Q.state != 'ACTIVE'):
...     print('Drive', drive, 'is not in ACTIVE!!!')
```

2.20.3 FC Ports

For each Node, you can use the `Node.get_fc_ports()` method to obtain the FC ports it contains. Each FC port is returned as a Python dictionary containing its attributes

```
>>> for node in system.components.nodes:
...     for fc_port in node.get_fc_ports():
...         if not fc_port.is_link_up():
...             print('Port', fc_port.get_field('wwpn'), 'of', node, 'is down!')
```

Use `FcPort.disable()` method to disable an FC port

```
>>> fc_port.disable()
>>> fc_port.is_enabled()
False
```

Use `FcPort.enable()` method to enable an FC port with a given role

```
>>> fc_port.enable(role='HARD_PORT')
>>> fc_port.is_enabled()
True
```

2.20.4 Services

Use `Node.get_service()` to get a service by its name:

```
>>> node.get_service('mgmt')
<...:Service id=system:0_rack:1_node:3_service:mgmt>
```

Or get a specific service type (core/mgmt):

```
>>> s = node.get_management_service()
>>> s = node.get_core_service()
```

2.21 Quality of Service

2.21.1 Introduction

Quality of Service policies allow to flexibly define the performance level for a given entity (e.g. a pool or a dataset).

Performance upper limit can be set in IOPS or in bandwidth (Mbps, Gbps, etc). Also, a burst can be defined, to allow the limits be exceeded for short periods.

2.21.2 Creation

The `QosPolicy` is an object, which has the same fields as the corresponding `InfiniBox` object. Creating a policy is done by using the `create` method:

```
>>> qos_policy = system.qos_policies.create(type='volume', max_ops=1000, name='my_
↳policy')
```

The 'type' field must be one of: 'volume', 'pool_volume'.

2.21.3 Manipulation

`QosPolicy` fields can be accessed, modified, queried and deleted in the same manner as any other `InfiniSDK` object:

```
>>> qos_policy.get_max_ops()
1000
>>> qos_policy.update_max_bps(100000000)
>>> qos_policy.get_max_bps()
100000000
>>> from infinisdk import Q
>>> print(', '.join([policy.get_name() for policy in system.qos_policies.find(Q.max_
↳bps >= 1000).to_list()]))
my_policy
>>> qos_policy.delete()
```

2.21.4 Entities Assignment

An assignment of a `QosPolicy` object to a pool or a dataset can be done in two ways. The first one is by referencing the QoS policy object:

```
>>> qos_policy = system.qos_policies.create(type='volume', max_ops=1000, name='my_
↳policy')
>>> qos_policy.assign_entity(volume)
```

The second one is by referencing the entity object and calling `assign_qos_policy`.

A pool can be assigned to a 'pool_volume' QoS policy:

```
>>> pool_qos_policy = system.qos_policies.create(type='pool_volume', max_ops=10000,
↳burst_enabled=False, name='vol_policy')
>>> pool.assign_qos_policy(pool_qos_policy)
>>> print(', '.join([policy.get_name() for policy in pool.get_qos_policies()]))
vol_policy
```


2.21.5 Querying

QoS Policy of a dataset can be retrieved by using the `get_qos_policy` method:

```
>>> print(volume.get_qos_policy().get_name())
my_policy
```

A dataset can also have a shared QoS policy, from its pool:

```
>>> print(volume.get_qos_shared_policy().get_name())
vol_policy
```

As a pool can have 2 policies, the `get_qos_policy` method is used.

Also, these convenience methods exist:

```
>>> print(pool.get_volume_qos_policy().get_name())
vol_policy
```

It is possible to get all entities assigned to a QoS policy, using `get_assigned_entities`:

```
>>> print(', '.join([entity.get_name() for entity in qos_policy.get_assigned_
↳entities()])))
my_volume
```

All entities assigned to QoS policies can be fetched as well:

```
>>> print(', '.join([entity.get_name() for entity in system.qos_policies.get_assigned_
↳entities()])))
my_volume, my_pool
```

2.21.6 Unassignment

As with assignment, clearing QoS policies can also be done in two ways:

```
>>> volume.unassign_qos_policy(qos_policy)
>>> volume.assign_qos_policy(qos_policy)
>>> qos_policy.unassign_entity(volume)
```

For pools:

```
>>> pool.unassign_qos_policies()
```

2.21.7 Misc

Retrieving all existing QoS policies in the system:

```
>>> system.qos_policies.get_all()
<Query /api/rest/qos/policies>
```

2.22 SMB Users and Groups

2.22.1 Creating a User

Create a user using the `create` method:

```
>>> user = system.smb_users.create()
```

2.22.2 Creating a Group

Create a group using the `create` method:

```
>>> group = system.smb_groups.create()
```

2.22.3 Adding a User to a Group

A user can be added to a group by using the `update_groups` method

```
>>> new_groups = user.get_groups() + [group]
>>> user.update_groups(new_groups)
>>> group in user.get_groups()
True
```

2.22.4 Removing a User from a Group

A user can be added to a group by using the `update_groups` method

```
>>> new_groups = [g for g in user.get_groups() if g != group]
>>> user.update_groups(new_groups)
>>> group in user.get_groups()
False
```

2.22.5 Setting a User's Primary Group

A user's primary group can be set using the `update_primary_group` method:

```
>>> user.update_primary_group(group)
>>> user.get_primary_group() == group
True
```

2.23 Active Directory Domains

2.23.1 Joining a Domain

Join a domain using the `join` method:

```
>>> system.active_directory_domains.join(
...     domain=domain,
...     preferred_ips=["196.0.0.0"],
...     username=username,
...     password=password
... )
```

2.23.2 Leaving a Domain

Leave a domain using the `leave` method:

```
>>> system.active_directory_domains.leave(
...     username=username,
...     password=password
... )
```

2.24 Extending InfiniSDK

InfiniSDK focuses on providing the official InfiniBox API as provided to the customer by Infinidat. However, some uses may require accessing internal or custom APIs, or wrapping more complex sequences in convenient API calls. Examples of this can be technician APIs, development utilities, customer macros and more.

InfiniSDK provides a convenient extensibility mechanism to allow us to extend its behavior.

2.24.1 Extending Objects with Methods

A very common case is adding methods to InfiniSDK objects. Let's assume we want to add a method to an *infinisdk.infinibox.InfiniBox* object, to get the location of the system from a global dictionary:

```
>>> s1 = InfiniBox(system1_address, auth = (username, password))
>>> s2 = InfiniBox(system2_address, auth = (username, password))
>>> _ = s1.login()
>>> _ = s2.login()
>>> locations = {s1: "upper floor", s2: "lower floor"}
```

By default, of course, we don't have such a mechanism:

```
>>> s1.get_location()
Traceback (most recent call last):
...
AttributeError: ...
```

We head off to write our new method, and use *infinisdk.core.extensions.add_method()* to attach it to *InfiniBox* objects

```
>>> from infinisdk.core import extensions

>>> @extensions.add_method(InfiniBox)
... def get_location(system):
...     return locations[system]
```

Now we can get the location safely:

```
>>> s1.get_location()
'upper floor'
>>> s2.get_location()
'lower floor'
```

2.25 Cookbook

Below are several common tasks and how to accomplish them using InfiniSDK.

2.25.1 Authentication

Saving credentials for reuse without login

In some cases it is useful to save the authentication information for later, in order to avoid an unnecessary login event. It can be used, for instance, for scripts that are being constantly rerun at high frequency.

To do that, use `API.save_credentials()` and `API.load_credentials()`:

```
>>> import pickle
>>> import tempfile
>>> import os

>>> filename = os.path.join(tempfile.mkdtemp(), 'creds')

>>> creds = system.api.save_credentials()
>>> with open(filename, 'wb') as f:
...     pickle.dump(creds, f)
```

```
>>> import pickle
>>> with open(filename, 'rb') as f:
...     creds = pickle.load(f)
>>> creds = system.api.load_credentials(creds)
```

2.25.2 Objects

Determining if an object is of a certain type

```
>>> assert isinstance(pool, system.pools.object_type)
>>> assert not isinstance(pool, system.volumes.object_type)
```

2.25.3 Error Handling

Adding Retries for Specific Errors

InfiniSDK supports automatic retries on various errors. This can come in handy for scripts performing maintenance operations which might fail API commands intermittently. To add a custom retry, use the `add_auto_retry` method:

```
def service_unavailable_predicate(e):
    return isinstance(e, APICommandFailed) and e.status_code == httplib.SERVICE_
    ↪ UNAVAILABLE

# the following makes InfiniSDK retry automatically on 503 Service Unavailable errors,
    ↪ up to 10
# retries with 30 seconds between attempts
self.system.api.add_auto_retry(service_unavailable_predicate, max_retries=10, sleep_
    ↪ seconds=30)
try:
    ... # <-- operation here
finally:
    self.system.api.remove_auto_retry(service_unavailable_predicate)
```

2.26 Frequently Asked Questions

2.26.1 My Script is Performing Very Poorly. Why is That?

InfiniSDK automatically caches fields and values for individual objects it retrieves from the system. However, due to technical limitations, this is not done system-wide. This means that while a single object will use the cache when fetching the same field more than once, fetching two fields from two different objects will cause the data to be fetched again.

This means that some scenarios may cause re-fetching the same piece of data over and over again if we are not careful or aware of this behavior. For instance, the following code will be slow, since it forces a field to be fetched on every iteration:

```
for i in range(10):
    v = system.volumes.get(name='vol1')
    v.get_size() # <-- this always fetches a new value
```

The reason behind this is that the Pythonic object `v` is recreated every time behind the scenes, and the cache that was created in the previous loop iteration is thrown away. This code, however, is efficient:

```
v = system.volumes.get(name='vol1')
for i in range(10):
    v.get_size() # <-- reuses the cache value when possible
```

Another example is determining which volume is mapped to each host. A naive approach would be:

```
for v in system.volumes:
    for lu in v.get_logical_units():
        print(v, 'is mapped to', lu.get_host().get_name())
```

This would work, of course, but will be relatively slow. There are two reasons for this:

1. `get_logical_units()` fetches a field that is not cached by default, due to implementation constraints.
2. `get_host().get_name()` suffers from the same issue we saw above, fetching the host multiple times.

Here's a more efficient version of the above code, focusing on fetching each object type as few times as possible:

```
volumes_by_host = {}
for host in system.hosts:
```

(continues on next page)

(continued from previous page)

```
for lu in host.get_luns():
    volumes_by_host.setdefault(lu.get_volume().id, []).append(host.get_name())

for volume_id, host_names in volumes_by_host.items():
    for host_name in host_names:
        print(volume_id, 'is mapped to', host_name)
```

2.27 API Reference

2.27.1 infinibox

class `infinisdk.infinibox.InfiniBox` (*address*, *auth=None*, *use_ssl=False*, *ssl_cert=None*)

SYSTEM_COMPONENTS_TYPE

alias of `infinisdk.infinibox.components.InfiniBoxSystemComponents`

SYSTEM_EVENTS_TYPE

alias of `infinisdk.infinibox.events.Events`

check_version()

Called automatically by the API on the first request made to the system. Should fetch and verify the system version to make sure it can be operated against.

get_model_name (*long_name=False*)

Retrieves the model name as reported by the system

get_name()

Returns the name of the system

get_serial (***kwargs*)

Returns the serial number of the system

get_version()

Returns the product version of the system

is_logged_in()

Returns True if login() was called on this system, and logout() hasn't been called yet

iter_related_systems()

Iterate the list of systems related to the current system

login()

Verifies the current user against the system

logout()

Logs out the current user

register_related_system (*system*)

Registers another system as related system to the current one

unregister_related_system (*system*)

Unregisters another system from appearing the the current system's related systems

update_name (*name*)

Update the name of the system

2.27.2 infinibox.api

`infinibox.api` is the sub-object responsible for sending API requests to the system. It also holds the current authentication information for the session.

class `infinisdk.core.api.api.API` (*target, auth, use_ssl, ssl_cert*)

clone_requests_session ()

Return a copy of system session for cases we need to manipulate different attrs of the session for now, the cloned session has a copy of: headers, cookies, verify, cert, adapters

delete (*path, **kwargs*)

Shortcut for `.request('delete')`

disabled_login_refresh_context ()

Inside this context, InfiniSDK will not attempt to refresh login cookies when logged out by expired cookies

get (*path, **kwargs*)

Shortcut for `.request('get')`

get_approval_context (*value*)

A context manager that controls whether requests are automatically approved (confirmed)

get_approved_context ()

A context marking all operations as approved (confirmed)

get_auth ()

Returns a tuple of the current username/password used by the API

get_auth_context (*username, password, login=True*)

Changes the API authentication information for the duration of the context:

```
>>> with system.api.get_auth_context('username', 'password'):
...     ... # execute operations as 'username'
```

get_unapproved_context ()

A context marking all operations as unapproved (not confirmed)

load_credentials (*creds*)

Loads credentials from the given credentials

Parameters *creds* – the result of a previous `API.save_credentials()` call

patch (*path, **kwargs*)

Shortcut for `.request('patch')`

post (*path, **kwargs*)

Shortcut for `.request('post')`

put (*path, **kwargs*)

Shortcut for `.request('put')`

request (*http_method, path, assert_success=True, **kwargs*)

Sends HTTP API request to the remote system

save_credentials ()

Returns a copy of the current credentials, useful for loading them later

set_auth (*username_or_auth, password=<NOTHING>, login=True*)

Sets the username and password under which operations will be performed

Can be used both with a tuple argument or with two arguments (username, password):

```
>>> system.api.set_auth(('username', 'password'))
>>> system.api.set_auth('username', 'password')
```

set_interactive_approval()

Causes an interactive prompt whenever a command requires approval from the user

use_basic_auth_context()

Causes API requests to send auth through Basic authorization

class `infinisdk.core.api.api.Response` (*resp, data, start_timestamp, end_timestamp*)

System API request response

get_error()

Returns The error portion of the response as returned from the system, or None if it doesn't exist

get_json()

Returns The JSON object returned from the system, or None if no json could be decoded

get_metadata()

Returns The metadata portion of the response (paging information, etc.) as returned from the system, or None if it doesn't exist

get_result()

Returns The result of the API call, extracted from the response JSON object

response = None

Response object as returned from requests

sent_data = None

Data sent to on

url = None

The URL from which this response was obtained

2.27.3 infinibox.datasets

class `infinisdk.infinibox.dataset.DatasetTypeBinder` (*object_type, system*)

create_many (**args, **kwargs*)

Creates multiple volumes with a single call. Parameters are just like `volumes.create`, only with the addition of the `count` parameter

Returns: list of volumes

Parameters `count` – number of volumes to create. Defaults to 1.

class `infinisdk.infinibox.dataset.Dataset` (*system, initial_data*)

calculate_reclaimable_space()

Returns the space to be reclaimed if the dataset would be deleted according to delete simulation api

create_snapshot (*name=None, **kwargs*)

Creates a snapshot from this entity, if supported by the system Supports passing name, `write_protected` and all other snapshots creation fields

delete (*force_if_snapshot_locked=<OMIT>*)
Deletes this object.

disable_compression (***kwargs*)
Set the value of the 'compression_enabled' field to False

disable_ssd (***kwargs*)
Set the value of the 'ssd_enabled' field to False

disable_write_protection (***kwargs*)
Set the value of the 'write_protected' field to False

enable_compression (***kwargs*)
Set the value of the 'compression_enabled' field to True

enable_ssd (***kwargs*)
Set the value of the 'ssd_enabled' field to True

enable_write_protection (***kwargs*)
Set the value of the 'write_protected' field to True

get_allocated (***kwargs*)
Obtains the value of the 'allocated' field

Returns Capacity

get_capacity_savings (***kwargs*)
Obtains the value of the 'capacity_savings' field

Returns
Capacity

get_children (***kwargs*)
Retrieves all child entities for this entity (either clones or snapshots)

get_created_at (***kwargs*)
Obtains the value of the 'created_at' field

Returns Arrow

get_creation_time ()
Retrieves creation time for this entity

get_dataset_type (***kwargs*)
Obtains the value of the 'dataset_type' field

Returns str

get_depth (***kwargs*)
Obtains the value of the 'depth' field

Returns int

get_family_id (***kwargs*)
Obtains the value of the 'family_id' field

Returns int

get_id (***kwargs*)
Obtains the value of the 'id' field

Returns int

get_lock_expires_at (***kwargs*)
Obtains the value of the 'lock_expires_at' field

Returns Arrow

get_lock_state (**kwargs)
Obtains the value of the 'lock_state' field

Returns str

get_num_blocks (**kwargs)
Obtains the value of the 'num_blocks' field

Returns int

get_pool (**kwargs)
Obtains the value of the 'pool' field

Returns *infinisdk.infinibox.pool.Pool object*

get_pool_name (**kwargs)
Obtains the value of the 'pool_name' field

Returns str

get_provisioning (**kwargs)
Obtains the value of the 'provisioning' field

Returns str

get_qos_policy (**kwargs)
Obtains the value of the 'qos_policy' field

Returns *infinisdk.infinibox.qos_policy.QosPolicy object*

get_qos_shared_policy (**kwargs)
Obtains the value of the 'qos_shared_policy' field

Returns *infinisdk.infinibox.qos_policy.QosPolicy object*

get_replica_ids (**kwargs)
Obtains the value of the 'replica_ids' field

Returns list

get_replication_types (**kwargs)
Obtains the value of the 'replication_types' field

Returns list

get_rmr_snapshot_guid (**kwargs)
Obtains the value of the 'rmr_snapshot_guid' field

Returns str

get_size (**kwargs)
Obtains the value of the 'size' field

Returns

Capacity

get_snapshot_expires_at (**kwargs)
Obtains the value of the 'snapshot_expires_at' field

Returns int

get_snapshot_retention (**kwargs)
Obtains the value of the 'snapshot_retention' field

Returns int

get_snapshots ()

Retrieves all snapshot children of this entity

get_tenant (**kwargs)

Obtains the value of the 'tenant' field

Returns *infinisdk.infinibox.tenant.Tenant object*

get_tree_allocated (**kwargs)

Obtains the value of the 'tree_allocated' field

Returns

Capacity

get_type (**kwargs)

Obtains the value of the 'type' field

Returns str

get_updated_at (**kwargs)

Obtains the value of the 'updated_at' field

Returns Arrow

get_used_size (**kwargs)

Obtains the value of the 'used_size' field

Returns

Capacity

has_children ()

Returns whether or not this entity has children

is_compression_enabled (**kwargs)

Obtains the value of the 'compression_enabled' field

Returns bool

is_compression_suppressed (**kwargs)

Obtains the value of the 'compression_suppressed' field

Returns bool

is_mapped (**kwargs)

Obtains the value of the 'mapped' field

Returns bool

is_master ()

Returns whether or not this entity is a master entity (not a snapshot and not a clone)

is_replicated (from_cache=<DONT_CARE>)

Returns True if this volume is a part of a replica, whether as source or as target

is_rmr_active_active_peer (**kwargs)

Obtains the value of the 'rmr_active_active_peer' field

Returns bool

is_rmr_source (**kwargs)

Obtains the value of the 'rmr_source' field

Returns bool

is_rmr_target (***kwargs*)

Obtains the value of the 'rmr_target' field

Returns bool

is_snapshot ()

Returns whether or not this entity is a snapshot

is_ssd_enabled (***kwargs*)

Obtains the value of the 'ssd_enabled' field

Returns bool

is_write_protected (***kwargs*)

Obtains the value of the 'write_protected' field

Returns bool

move_pool (*target_pool, with_capacity=False*)

Moves this entity to a new pool, optionally along with its needed capacity

refresh_snapshot (*force_if_replicated_on_target=<OMIT>*)

Refresh a snapshot with the most recent data from the parent :param force_if_replicated_on_target: (Only required on some InfiniBox versions) allows the refresh operation to occur on a dataset that is currently a replication target.

resize (*delta*)

Resize the entity by the given delta

restore (*snapshot*)

Restores this entity from a given snapshot object

update_compression_enabled

Updates the value of the 'compression_enabled' field

param value The new compression_enabled value to be set (type: bool)

update_lock_expires_at (*value, **kwargs*)

Updates the value of the 'lock_expires_at' field

Parameters value – The new lock_expires_at value to be set (type: Arrow)

update_provisioning (*value, **kwargs*)

Updates the value of the 'provisioning' field

Parameters value – The new provisioning value to be set (type: str)

update_size (*value, **kwargs*)

Updates the value of the 'size' field

Parameters value – The new size value to be set (type: Capacity)

update_ssd_enabled

Updates the value of the 'ssd_enabled' field

param value The new ssd_enabled value to be set (type: bool)

update_write_protected

Updates the value of the 'write_protected' field

param value The new write_protected value to be set (type: bool)

2.27.4 infinibox.volumes

class infinisdk.infinibox.volume.VolumesBinder (*object_type, system*)

create_group_snapshot (*volumes, snap_prefix=<Autogenerate: {prefix}{short_uuid}>, snap_suffix=<OMIT>*)

Creates multiple snapshots with a single consistent point-in-time, returning the snapshots in respective order to parent volumes

Parameters **volumes** – list of volumes we should create a snapshot of

class infinisdk.infinibox.volume.Volume (*system, initial_data*)

BINDER_CLASS

alias of *VolumesBinder*

calculate_reclaimable_space ()

Returns the space to be reclaimed if the dataset would be deleted according to delete simulation api

clear_metadata ()

Deletes all metadata keys for this object

classmethod construct (*system, data*)

Template method to enable customizing the object instantiation process.

This enables system components to be cached rather than re-fetched every time

classmethod create (*system, **fields*)

Creates a new object of this type

create_snapshot (*name=None, **kwargs*)

Creates a snapshot from this entity, if supported by the system Supports passing name, write_protected and all other snapshots creation fields

delete (*force_if_snapshot_locked=<OMIT>*)

Deletes this object.

disable_compression (***kwargs*)

Set the value of the 'compression_enabled' field to False

disable_ssd (***kwargs*)

Set the value of the 'ssd_enabled' field to False

disable_write_protection (***kwargs*)

Set the value of the 'write_protected' field to False

enable_compression (***kwargs*)

Set the value of the 'compression_enabled' field to True

enable_ssd (***kwargs*)

Set the value of the 'ssd_enabled' field to True

enable_write_protection (***kwargs*)

Set the value of the 'write_protected' field to True

get_all_metadata ()

Returns Dictionary of all keys and values associated as metadata for this object

get_allocated (***kwargs*)

Obtains the value of the 'allocated' field

Returns

Capacity

get_capacity_savings (**kwargs)

Obtains the value of the 'capacity_savings' field

Returns

Capacity

get_children (**kwargs)

Retrieves all child entities for this entity (either clones or snapshots)

get_cons_group (**kwargs)

Obtains the value of the 'cons_group' field

Returns *infinisdk.infinibox.cons_group.ConsGroup object*

get_created_at (**kwargs)

Obtains the value of the 'created_at' field

Returns Arrow

classmethod get_creation_defaults ()

Returns a dict representing the default arguments as implicitly constructed by infinisdk to fulfill a `create` call

Note: This will cause generation of defaults, which will have side effects if they are special values

Note: This does not necessarily generate all fields that are passable into `create`, only mandatory 'fields'

get_creation_time ()

Retrieves creation time for this entity

get_data_snapshot_guid (**kwargs)

Obtains the value of the 'data_snapshot_guid' field

Returns str

get_dataset_type (**kwargs)

Obtains the value of the 'dataset_type' field

Returns str

get_depth (**kwargs)

Obtains the value of the 'depth' field

Returns int

get_family_id (**kwargs)

Obtains the value of the 'family_id' field

Returns int

get_field (field_name, from_cache=<DONT_CARE>, fetch_if_not_cached=True, raw_value=False)

Gets the value of a single field from the system

Parameters

- **cache** – Attempt to use the last cached version of the field value

- **fetch_if_not_cached** – Pass `False` to force only from cache

get_fields (*field_names=()*, *from_cache=<DONT_CARE>*, *fetch_if_not_cached=True*,
raw_value=False)

Gets a set of fields from the system

Parameters

- **from_cache** – Attempt to fetch the fields from the cache
- **fetch_if_not_cached** – pass as `False` to force only from cache

Returns a dictionary of field names to their values

get_id (***kwargs*)

Obtains the value of the 'id' field

Returns int

get_lock_expires_at (***kwargs*)

Obtains the value of the 'lock_expires_at' field

Returns Arrow

get_lock_state (***kwargs*)

Obtains the value of the 'lock_state' field

Returns str

get_lun (*mapping_object*)

Given either a host or a host cluster object, returns the single LUN object mapped to this volume.

An exception is raised if multiple matching LUs are found

Parameters **mapping_object** – Either a host cluster or a host object to be checked

Returns None if no lu is found for this entity

get_metadata_value (*key*, *default=<NOTHING>*)

Gets a metadata value, optionally specifying a default

Parameters **default** – if specified, the value to retrieve if the metadata key doesn't exist. if not specified, and the key does not exist, the operation will raise an exception

get_name (***kwargs*)

Obtains the value of the 'name' field

Returns str

get_nguid (***kwargs*)

Obtains the value of the 'nguid' field

Returns str

get_num_blocks (***kwargs*)

Obtains the value of the 'num_blocks' field

Returns int

get_parent (***kwargs*)

Obtains the value of the 'parent' field

Returns `infinisdk.infinibox.volume.Volume` object

get_pool (***kwargs*)

Obtains the value of the 'pool' field

Returns `infinisdk.infinibox.pool.Pool` object

get_pool_name (**kwargs)
 Obtains the value of the 'pool_name' field
Returns str

get_provisioning (**kwargs)
 Obtains the value of the 'provisioning' field
Returns str

get_qos_policy (**kwargs)
 Obtains the value of the 'qos_policy' field
Returns *infinisdk.infinibox.qos_policy.QosPolicy object*

get_qos_shared_policy (**kwargs)
 Obtains the value of the 'qos_shared_policy' field
Returns *infinisdk.infinibox.qos_policy.QosPolicy object*

get_replica_ids (**kwargs)
 Obtains the value of the 'replica_ids' field
Returns list

get_replication_types (**kwargs)
 Obtains the value of the 'replication_types' field
Returns list

get_rmr_snapshot_guid (**kwargs)
 Obtains the value of the 'rmr_snapshot_guid' field
Returns str

get_serial (**kwargs)
 Obtains the value of the 'serial' field
Returns SCSI Serial

get_size (**kwargs)
 Obtains the value of the 'size' field
Returns
 Capacity

get_snapshot_expires_at (**kwargs)
 Obtains the value of the 'snapshot_expires_at' field
Returns int

get_snapshot_retention (**kwargs)
 Obtains the value of the 'snapshot_retention' field
Returns int

get_snapshots ()
 Retrieves all snapshot children of this entity

get_tenant (**kwargs)
 Obtains the value of the 'tenant' field
Returns *infinisdk.infinibox.tenant.Tenant object*

get_tree_allocated (**kwargs)
 Obtains the value of the 'tree_allocated' field

Returns

Capacity

get_type (**kwargs)

Obtains the value of the 'type' field

Returns str

get_udid (**kwargs)

Obtains the value of the 'udid' field

Returns int

get_updated_at (**kwargs)

Obtains the value of the 'updated_at' field

Returns Arrow

get_used_size (**kwargs)

Obtains the value of the 'used_size' field

Returns

Capacity

has_children ()

Returns whether or not this entity has children

invalidate_cache (*field_names)

Discards the cached field values of this object, causing the next fetch to retrieve the fresh value from the system

is_compression_enabled (**kwargs)

Obtains the value of the 'compression_enabled' field

Returns bool

is_compression_suppressed (**kwargs)

Obtains the value of the 'compression_suppressed' field

Returns bool

is_in_system ()

Returns whether or not the object actually exists

is_mapped (**kwargs)

Obtains the value of the 'mapped' field

Returns bool

is_master ()

Returns whether or not this entity is a master entity (not a snapshot and not a clone)

is_paths_available (**kwargs)

Obtains the value of the 'paths_available' field

Returns bool

is_replicated (from_cache=<DONT_CARE>)

Returns True if this volume is a part of a replica, whether as source or as target

is_rmr_active_active_peer (**kwargs)

Obtains the value of the 'rmr_active_active_peer' field

Returns bool

is_rmr_source (***kwargs*)
Obtains the value of the 'rmr_source' field

Returns bool

is_rmr_target (***kwargs*)
Obtains the value of the 'rmr_target' field

Returns bool

is_snapshot ()
Returns whether or not this entity is a snapshot

is_ssd_enabled (***kwargs*)
Obtains the value of the 'ssd_enabled' field

Returns bool

is_write_protected (***kwargs*)
Obtains the value of the 'write_protected' field

Returns bool

move_pool (*target_pool, with_capacity=False*)
Moves this entity to a new pool, optionally along with its needed capacity

refresh_snapshot (*force_if_replicated_on_target=<OMIT>*)
Refresh a snapshot with the most recent data from the parent :param force_if_replicated_on_target: (Only required on some InfiniBox versions) allows the refresh operation to occur on a dataset that is currently a replication target.

resize (*delta*)
Resize the entity by the given delta

restore (*snapshot*)
Restores this entity from a given snapshot object

safe_delete (**args, **kwargs*)
Tries to delete the object, doing nothing if the object cannot be found on the system

safe_get_field (*field_name, default=<NOTHING>, **kwargs*)
Like `get_field()`, only returns 'default' parameter if no result was found

set_metadata (*key, value*)
Sets metadata key in the system associated with this object

set_metadata_from_dict (*data_dict*)
Sets multiple metadata keys/values in the system associated with this object

unmap ()
Unmaps a volume from its hosts

unset_metadata (*key*)
Deletes a metadata key for this object

update_compression_enabled
Updates the value of the 'compression_enabled' field

param value The new compression_enabled value to be set (type: bool)

update_field (*field_name, field_value*)
Updates the value of a single field

update_fields (***update_dict*)
Atomically updates a group of fields and respective values (given as a dictionary)

update_lock_expires_at (*value*, ***kwargs*)
 Updates the value of the 'lock_expires_at' field

Parameters **value** – The new lock_expires_at value to be set (type: Arrow)

update_name (*value*, ***kwargs*)
 Updates the value of the 'name' field

Parameters **value** – The new name value to be set (type: str)

update_provisioning (*value*, ***kwargs*)
 Updates the value of the 'provisioning' field

Parameters **value** – The new provisioning value to be set (type: str)

update_size (*value*, ***kwargs*)
 Updates the value of the 'size' field

Parameters **value** – The new size value to be set (type: Capacity)

update_ssd_enabled
 Updates the value of the 'ssd_enabled' field

param value The new ssd_enabled value to be set (type: bool)

update_udid (*value*, ***kwargs*)
 Updates the value of the 'udid' field

Parameters **value** – The new udid value to be set (type: int)

update_write_protected
 Updates the value of the 'write_protected' field

param value The new write_protected value to be set (type: bool)

2.27.5 infinibox.filesystems

class infinisdk.infinibox.filesystem.**FilesystemBinder** (*object_type*, *system*)

class infinisdk.infinibox.filesystem.**Filesystem** (*system*, *initial_data*)

BINDER_CLASS

alias of *FilesystemBinder*

calculate_reclaimable_space ()

Returns the space to be reclaimed if the dataset would be deleted according to delete simulation api

clear_metadata ()

Deletes all metadata keys for this object

classmethod construct (*system*, *data*)

Template method to enable customizing the object instantiation process.

This enables system components to be cached rather than re-fetched every time

classmethod create (*system*, ***fields*)

Creates a new object of this type

create_snapshot (*name=None*, ***kwargs*)

Creates a snapshot from this entity, if supported by the system Supports passing name, write_protected and all other snapshots creation fields

delete (*force_if_snapshot_locked=<OMIT>*)

Deletes this object.

disable_compression (***kwargs*)

Set the value of the ‘compression_enabled’ field to False

disable_ssd (***kwargs*)

Set the value of the ‘ssd_enabled’ field to False

disable_write_protection (***kwargs*)

Set the value of the ‘write_protected’ field to False

enable_compression (***kwargs*)

Set the value of the ‘compression_enabled’ field to True

enable_ssd (***kwargs*)

Set the value of the ‘ssd_enabled’ field to True

enable_write_protection (***kwargs*)

Set the value of the ‘write_protected’ field to True

get_all_metadata ()

Returns Dictionary of all keys and values associated as metadata for this object

get_allocated (***kwargs*)

Obtains the value of the ‘allocated’ field

Returns

Capacity

get_atime_granularity (***kwargs*)

Obtains the value of the ‘atime_granularity’ field

Returns int

get_atime_mode (***kwargs*)

Obtains the value of the ‘atime_mode’ field

Returns str

get_capacity_savings (***kwargs*)

Obtains the value of the ‘capacity_savings’ field

Returns

Capacity

get_children (***kwargs*)

Retrieves all child entities for this entity (either clones or snapshots)

get_created_at (***kwargs*)

Obtains the value of the ‘created_at’ field

Returns Arrow

classmethod get_creation_defaults ()

Returns a dict representing the default arguments as implicitly constructed by infinisdsk to fulfill a `create` call

Note: This will cause generation of defaults, which will have side effects if they are special values

Note: This does not necessarily generate all fields that are passable into `create`, only mandatory ‘fields’

get_creation_time ()

Retrieves creation time for this entity

get_data_snapshot_guid (**kwargs)

Obtains the value of the ‘data_snapshot_guid’ field

Returns str

get_dataset_type (**kwargs)

Obtains the value of the ‘dataset_type’ field

Returns str

get_depth (**kwargs)

Obtains the value of the ‘depth’ field

Returns int

get_family_id (**kwargs)

Obtains the value of the ‘family_id’ field

Returns int

get_field (field_name, from_cache=<DONT_CARE>, fetch_if_not_cached=True, raw_value=False)

Gets the value of a single field from the system

Parameters

- **cache** – Attempt to use the last cached version of the field value
- **fetch_if_not_cached** – Pass `False` to force only from cache

get_fields (field_names=(), from_cache=<DONT_CARE>, fetch_if_not_cached=True, raw_value=False)

Gets a set of fields from the system

Parameters

- **from_cache** – Attempt to fetch the fields from the cache
- **fetch_if_not_cached** – pass as `False` to force only from cache

Returns a dictionary of field names to their values

get_id (**kwargs)

Obtains the value of the ‘id’ field

Returns int

get_lock_expires_at (**kwargs)

Obtains the value of the ‘lock_expires_at’ field

Returns Arrow

get_lock_state (**kwargs)

Obtains the value of the ‘lock_state’ field

Returns str

get_metadata_value (key, default=<NOTHING>)

Gets a metadata value, optionally specifying a default

Parameters default – if specified, the value to retrieve if the metadata key doesn’t exist. if not specified, and the key does not exist, the operation will raise an exception

get_name (**kwargs)

Obtains the value of the ‘name’ field

Returns str

get_num_blocks (**kwargs)

Obtains the value of the ‘num_blocks’ field

Returns int

get_parent (**kwargs)

Obtains the value of the ‘parent’ field

Returns *infinisdk.infinibox.filesystem.Filesystem object*

get_pool (**kwargs)

Obtains the value of the ‘pool’ field

Returns *infinisdk.infinibox.pool.Pool object*

get_pool_name (**kwargs)

Obtains the value of the ‘pool_name’ field

Returns str

get_provisioning (**kwargs)

Obtains the value of the ‘provisioning’ field

Returns str

get_qos_policy (**kwargs)

Obtains the value of the ‘qos_policy’ field

Returns *infinisdk.infinibox.qos_policy.QosPolicy object*

get_qos_shared_policy (**kwargs)

Obtains the value of the ‘qos_shared_policy’ field

Returns *infinisdk.infinibox.qos_policy.QosPolicy object*

get_replica_ids (**kwargs)

Obtains the value of the ‘replica_ids’ field

Returns list

get_replication_types (**kwargs)

Obtains the value of the ‘replication_types’ field

Returns list

get_rmr_snapshot_guid (**kwargs)

Obtains the value of the ‘rmr_snapshot_guid’ field

Returns str

get_security_style (**kwargs)

Obtains the value of the ‘security_style’ field

Returns str

get_size (**kwargs)

Obtains the value of the ‘size’ field

Returns

Capacity

get_snapdir_name (**kwargs)

Obtains the value of the 'snapdir_name' field

Returns str

get_snapshot_expires_at (**kwargs)

Obtains the value of the 'snapshot_expires_at' field

Returns int

get_snapshot_retention (**kwargs)

Obtains the value of the 'snapshot_retention' field

Returns int

get_snapshots ()

Retrieves all snapshot children of this entity

get_tenant (**kwargs)

Obtains the value of the 'tenant' field

Returns *infinisdk.infinibox.tenant.Tenant* object

get_tree_allocated (**kwargs)

Obtains the value of the 'tree_allocated' field

Returns

Capacity

get_type (**kwargs)

Obtains the value of the 'type' field

Returns str

get_updated_at (**kwargs)

Obtains the value of the 'updated_at' field

Returns Arrow

get_used_size (**kwargs)

Obtains the value of the 'used_size' field

Returns

Capacity

has_children ()

Returns whether or not this entity has children

invalidate_cache (*field_names)

Discards the cached field values of this object, causing the next fetch to retrieve the fresh value from the system

is_compression_enabled (**kwargs)

Obtains the value of the 'compression_enabled' field

Returns bool

is_compression_suppressed (**kwargs)

Obtains the value of the 'compression_suppressed' field

Returns bool

is_established (**kwargs)

Obtains the value of the ‘established’ field

Returns bool

is_in_system ()

Returns whether or not the object actually exists

is_mapped (**kwargs)

Obtains the value of the ‘mapped’ field

Returns bool

is_master ()

Returns whether or not this entity is a master entity (not a snapshot and not a clone)

is_replicated (from_cache=<DONT_CARE>)

Returns True if this volume is a part of a replica, whether as source or as target

is_rmr_active_active_peer (**kwargs)

Obtains the value of the ‘rmr_active_active_peer’ field

Returns bool

is_rmr_source (**kwargs)

Obtains the value of the ‘rmr_source’ field

Returns bool

is_rmr_target (**kwargs)

Obtains the value of the ‘rmr_target’ field

Returns bool

is_snapdir_accessible (**kwargs)

Obtains the value of the ‘snapdir_accessible’ field

Returns bool

is_snapshot ()

Returns whether or not this entity is a snapshot

is_ssd_enabled (**kwargs)

Obtains the value of the ‘ssd_enabled’ field

Returns bool

is_visible_in_snapdir (**kwargs)

Obtains the value of the ‘visible_in_snapdir’ field

Returns bool

is_write_protected (**kwargs)

Obtains the value of the ‘write_protected’ field

Returns bool

move_pool (target_pool, with_capacity=False)

Moves this entity to a new pool, optionally along with its needed capacity

refresh_snapshot (force_if_replicated_on_target=<OMIT>)

Refresh a snapshot with the most recent data from the parent :param force_if_replicated_on_target: (Only required on some InfiniBox versions) allows the refresh operation to occur on a dataset that is currently a replication target.

resize (*delta*)
 Resize the entity by the given delta

restore (*snapshot*)
 Restores this entity from a given snapshot object

safe_delete (**args, **kwargs*)
 Tries to delete the object, doing nothing if the object cannot be found on the system

safe_get_field (*field_name, default=<NOTHING>, **kwargs*)
 Like *get_field()*, only returns 'default' parameter if no result was found

set_metadata (*key, value*)
 Sets metadata key in the system associated with this object

set_metadata_from_dict (*data_dict*)
 Sets multiple metadata keys/values in the system associated with this object

unset_metadata (*key*)
 Deletes a metadata key for this object

update_atime_granularity (*value, **kwargs*)
 Updates the value of the 'atime_granularity' field

Parameters value – The new atime_granularity value to be set (type: int)

update_compression_enabled
 Updates the value of the 'compression_enabled' field

param value The new compression_enabled value to be set (type: bool)

update_field (*field_name, field_value*)
 Updates the value of a single field

update_fields (***update_dict*)
 Atomically updates a group of fields and respective values (given as a dictionary)

update_lock_expires_at (*value, **kwargs*)
 Updates the value of the 'lock_expires_at' field

Parameters value – The new lock_expires_at value to be set (type: Arrow)

update_name (*value, **kwargs*)
 Updates the value of the 'name' field

Parameters value – The new name value to be set (type: str)

update_provisioning (*value, **kwargs*)
 Updates the value of the 'provisioning' field

Parameters value – The new provisioning value to be set (type: str)

update_size (*value, **kwargs*)
 Updates the value of the 'size' field

Parameters value – The new size value to be set (type: Capacity)

update_ssd_enabled
 Updates the value of the 'ssd_enabled' field

param value The new ssd_enabled value to be set (type: bool)

update_write_protected
 Updates the value of the 'write_protected' field

param value The new write_protected value to be set (type: bool)

2.27.6 infinibox.exports

class infinisdk.infinibox.export.**Export** (*system, initial_data*)

disable (***kwargs*)

Set the value of the 'enabled' field to False

disable_32bit_file_id (***kwargs*)

Set the value of the '32bit_file_id' field to False

disable_make_all_users_anonymous (***kwargs*)

Set the value of the 'make_all_users_anonymous' field to False

disable_privileged_port (***kwargs*)

Set the value of the 'privileged_port' field to False

disable_snapdir_visible (***kwargs*)

Set the value of the 'snapdir_visible' field to False

enable (***kwargs*)

Set the value of the 'enabled' field to True

enable_32bit_file_id (***kwargs*)

Set the value of the '32bit_file_id' field to True

enable_make_all_users_anonymous (***kwargs*)

Set the value of the 'make_all_users_anonymous' field to True

enable_privileged_port (***kwargs*)

Set the value of the 'privileged_port' field to True

enable_snapdir_visible (***kwargs*)

Set the value of the 'snapdir_visible' field to True

get_anonymous_gid (***kwargs*)

Obtains the value of the 'anonymous_gid' field

Returns int

get_anonymous_uid (***kwargs*)

Obtains the value of the 'anonymous_uid' field

Returns int

get_created_at (***kwargs*)

Obtains the value of the 'created_at' field

Returns Arrow

get_export_path (***kwargs*)

Obtains the value of the 'export_path' field

Returns str

get_filesystem (***kwargs*)

Obtains the value of the 'filesystem' field

Returns int

get_id (***kwargs*)

Obtains the value of the 'id' field

Returns int

get_inner_path (**kwargs)
Obtains the value of the 'inner_path' field

Returns str

get_max_read (**kwargs)
Obtains the value of the 'max_read' field

Returns
Capacity

get_max_write (**kwargs)
Obtains the value of the 'max_write' field

Returns
Capacity

get_permissions (**kwargs)
Obtains the value of the 'permissions' field

Returns list

get_pref_read (**kwargs)
Obtains the value of the 'pref_read' field

Returns
Capacity

get_pref_readdir (**kwargs)
Obtains the value of the 'pref_readdir' field

Returns
Capacity

get_pref_write (**kwargs)
Obtains the value of the 'pref_write' field

Returns
Capacity

get_tenant (**kwargs)
Obtains the value of the 'tenant' field

Returns *infinisdk.infinibox.tenant.Tenant object*

get_transport_protocols (**kwargs)
Obtains the value of the 'transport_protocols' field

Returns str

get_updated_at (**kwargs)
Obtains the value of the 'updated_at' field

Returns Arrow

is_32bit_file_id (**kwargs)
Obtains the value of the '32bit_file_id' field

Returns bool

is_enabled (**kwargs)
Obtains the value of the 'enabled' field

Returns bool

is_make_all_users_anonymous (**kwargs)
Obtains the value of the 'make_all_users_anonymous' field

Returns bool

is_privileged_port (**kwargs)
Obtains the value of the 'privileged_port' field

Returns bool

is_snapdir_visible (**kwargs)
Obtains the value of the 'snapdir_visible' field

Returns bool

update_32bit_file_id
Updates the value of the '32bit_file_id' field

param value The new 32bit_file_id value to be set (type: bool)

update_anonymous_gid (value, **kwargs)
Updates the value of the 'anonymous_gid' field

Parameters value – The new anonymous_gid value to be set (type: int)

update_anonymous_uid (value, **kwargs)
Updates the value of the 'anonymous_uid' field

Parameters value – The new anonymous_uid value to be set (type: int)

update_enabled
Updates the value of the 'enabled' field

param value The new enabled value to be set (type: bool)

update_make_all_users_anonymous
Updates the value of the 'make_all_users_anonymous' field

param value The new make_all_users_anonymous value to be set (type: bool)

update_max_read (value, **kwargs)
Updates the value of the 'max_read' field

Parameters value – The new max_read value to be set (type: [Capacity](#))

update_max_write (value, **kwargs)
Updates the value of the 'max_write' field

Parameters value – The new max_write value to be set (type: [Capacity](#))

update_permissions (value, **kwargs)
Updates the value of the 'permissions' field

Parameters value – The new permissions value to be set (type: list)

update_pref_read (value, **kwargs)
Updates the value of the 'pref_read' field

Parameters value – The new pref_read value to be set (type: [Capacity](#))

update_pref_readdir (value, **kwargs)
Updates the value of the 'pref_readdir' field

Parameters value – The new pref_readdir value to be set (type: [Capacity](#))

update_pref_write (*value*, ***kwargs*)

Updates the value of the 'pref_write' field

Parameters *value* – The new pref_write value to be set (type: [Capacity](#))

update_privileged_port

Updates the value of the 'privileged_port' field

param value The new privileged_port value to be set (type: bool)

update_snapdir_visible

Updates the value of the 'snapdir_visible' field

param value The new snapdir_visible value to be set (type: bool)

update_transport_protocols (*value*, ***kwargs*)

Updates the value of the 'transport_protocols' field

Parameters *value* – The new transport_protocols value to be set (type: str)

2.27.7 infinibox.shares

class `infinisdk.infinibox.share.Share` (*system*, *initial_data*)

disable (***kwargs*)

Set the value of the 'enabled' field to False

disable_access_based_enumeration (***kwargs*)

Set the value of the 'access_based_enumeration' field to False

disable_require_encryption (***kwargs*)

Set the value of the 'require_encryption' field to False

disable_snapdir_visible (***kwargs*)

Set the value of the 'snapdir_visible' field to False

enable (***kwargs*)

Set the value of the 'enabled' field to True

enable_access_based_enumeration (***kwargs*)

Set the value of the 'access_based_enumeration' field to True

enable_require_encryption (***kwargs*)

Set the value of the 'require_encryption' field to True

enable_snapdir_visible (***kwargs*)

Set the value of the 'snapdir_visible' field to True

get_created_at (***kwargs*)

Obtains the value of the 'created_at' field

Returns Arrow

get_default_file_unix_permissions (***kwargs*)

Obtains the value of the 'default_file_unix_permissions' field

Returns str

get_default_folder_unix_permissions (***kwargs*)

Obtains the value of the 'default_folder_unix_permissions' field

Returns str

get_description (**kwargs)
Obtains the value of the 'description' field

Returns str

get_filesystem (**kwargs)
Obtains the value of the 'filesystem' field

Returns *infinisdk.infinibox.filesystem.Filesystem object*

get_id (**kwargs)
Obtains the value of the 'id' field

Returns int

get_inner_path (**kwargs)
Obtains the value of the 'inner_path' field

Returns str

get_name (**kwargs)
Obtains the value of the 'name' field

Returns str

get_offline_caching (**kwargs)
Obtains the value of the 'offline_caching' field

Returns str

get_tenant (**kwargs)
Obtains the value of the 'tenant' field

Returns *infinisdk.infinibox.tenant.Tenant object*

get_updated_at (**kwargs)
Obtains the value of the 'updated_at' field

Returns Arrow

is_access_based_enumeration (**kwargs)
Obtains the value of the 'access_based_enumeration' field

Returns bool

is_enabled (**kwargs)
Obtains the value of the 'enabled' field

Returns bool

is_require_encryption (**kwargs)
Obtains the value of the 'require_encryption' field

Returns bool

is_snapdir_visible (**kwargs)
Obtains the value of the 'snapdir_visible' field

Returns bool

update_access_based_enumeration
Updates the value of the 'access_based_enumeration' field

param value The new access_based_enumeration value to be set (type: bool)

update_default_file_unix_permissions (value, **kwargs)
Updates the value of the 'default_file_unix_permissions' field

Parameters value – The new default_file_unix_permissions value to be set (type: str)

update_default_folder_unix_permissions (*value*, ***kwargs*)

Updates the value of the 'default_folder_unix_permissions' field

Parameters value – The new default_folder_unix_permissions value to be set (type: str)

update_description (*value*, ***kwargs*)

Updates the value of the 'description' field

Parameters value – The new description value to be set (type: str)

update_enabled

Updates the value of the 'enabled' field

param value The new enabled value to be set (type: bool)

update_field (*field_name*, *field_value*)

Updates the value of a single field

update_offline_caching (*value*, ***kwargs*)

Updates the value of the 'offline_caching' field

Parameters value – The new offline_caching value to be set (type: str)

update_require_encryption

Updates the value of the 'require_encryption' field

param value The new require_encryption value to be set (type: bool)

update_snapdir_visible

Updates the value of the 'snapdir_visible' field

param value The new snapdir_visible value to be set (type: bool)

2.27.8 infinibox.shares.permissions

class `infinisdk.infinibox.share_permission.SharePermission` (*system*, *initial_data*)

get_access (***kwargs*)

Obtains the value of the 'access' field

Returns str

get_id (***kwargs*)

Obtains the value of the 'id' field

Returns int

get_share (***kwargs*)

Obtains the value of the 'share' field

Returns `infinisdk.infinibox.share.Share` object

get_sid (***kwargs*)

Obtains the value of the 'sid' field

Returns str

update_access (*value*, ***kwargs*)

Updates the value of the 'access' field

Parameters value – The new access value to be set (type: str)

2.27.9 infinibox.replication_groups

```
class infinisdk.infinibox.replication_group.ReplicationGroup (system,          ini-
                                                         tial_data)
```

get_created_at (***kwargs*)
Obtains the value of the 'created_at' field

Returns Arrow

get_id (***kwargs*)
Obtains the value of the 'id' field

Returns int

get_members_count (***kwargs*)
Obtains the value of the 'members_count' field

Returns int

get_name (***kwargs*)
Obtains the value of the 'name' field

Returns str

get_pool (***kwargs*)
Obtains the value of the 'pool' field

Returns *infinisdk.infinibox.pool.Pool object*

get_updated_at (***kwargs*)
Obtains the value of the 'updated_at' field

Returns Arrow

get_uuid (***kwargs*)
Obtains the value of the 'uuid' field

Returns str

is_replicated (***kwargs*)
Obtains the value of the 'replicated' field

Returns bool

update_name (*value, **kwargs*)
Updates the value of the 'name' field

Parameters **value** – The new name value to be set (type: str)

2.27.10 infinibox.rg_replicas

```
class infinisdk.infinibox.rg_replica.RgReplica (system, initial_data)
```

get_base_action (***kwargs*)
Obtains the value of the 'base_action' field

Returns str

get_created_at (***kwargs*)
Obtains the value of the 'created_at' field

Returns Arrow

get_id (**kwargs)
Obtains the value of the 'id' field
Returns int

get_link (**kwargs)
Obtains the value of the 'link' field
Returns *infinisdk.infinibox.link.Link object*

get_local_link_guid (**kwargs)
Obtains the value of the 'local_link_guid' field
Returns str

get_pool (**kwargs)
Obtains the value of the 'pool' field
Returns *infinisdk.infinibox.pool.Pool object*

get_remote_pool_id (**kwargs)
Obtains the value of the 'remote_pool_id' field
Returns int

get_remote_replica_id (**kwargs)
Obtains the value of the 'remote_replica_id' field
Returns int

get_remote_replication_group_id (**kwargs)
Obtains the value of the 'remote_replication_group_id' field
Returns int

get_remote_replication_group_name (**kwargs)
Obtains the value of the 'remote_replication_group_name' field
Returns str

get_replica_configuration_guid (**kwargs)
Obtains the value of the 'replica_configuration_guid' field
Returns str

get_replication_group (**kwargs)
Obtains the value of the 'replication_group' field
Returns *infinisdk.infinibox.replication_group.ReplicationGroup object*

get_replication_type (**kwargs)
Obtains the value of the 'replication_type' field
Returns str

get_role (**kwargs)
Obtains the value of the 'role' field
Returns str

get_rpo (**kwargs)
Obtains the value of the 'rpo' field
Returns timedelta

get_state (**kwargs)
Obtains the value of the 'state' field

Returns str

get_state_description (**kwargs)
Obtains the value of the 'state_description' field

Returns str

get_state_reason (**kwargs)
Obtains the value of the 'state_reason' field

Returns str

get_sync_interval (**kwargs)
Obtains the value of the 'sync_interval' field

Returns timedelta

get_updated_at (**kwargs)
Obtains the value of the 'updated_at' field

Returns Arrow

resume ()
Resumes this rg_replica

suspend ()
Suspends this rg_replica

sync ()
Starts a sync job

update_rpo (value, **kwargs)
Updates the value of the 'rpo' field

Parameters value – The new rpo value to be set (type: timedelta)

update_sync_interval (value, **kwargs)
Updates the value of the 'sync_interval' field

Parameters value – The new sync_interval value to be set (type: timedelta)

2.27.11 infinibox.pools

infinibox.pools is of type *PoolBinder* described below.

class infinisdk.infinibox.pool.**PoolBinder** (*object_type*, *system*)
Implements *system.pools*

get_administered_pools ()
Returns the pools that can be managed by the current user

class infinisdk.infinibox.pool.**Pool** (*system*, *initial_data*)

BINDER_CLASS
alias of *PoolBinder*

clear_metadata ()
Deletes all metadata keys for this object

classmethod construct (*system, data*)

Template method to enable customizing the object instantiation process.

This enables system components to be cached rather than re-fetched every time

classmethod create (*system, **fields*)

Create a new pool

delete (***kwargs*)

Deletes this object.

disable_compression (***kwargs*)

Set the value of the 'compression_enabled' field to False

disable_ssd (***kwargs*)

Set the value of the 'ssd_enabled' field to False

enable_compression (***kwargs*)

Set the value of the 'compression_enabled' field to True

enable_ssd (***kwargs*)

Set the value of the 'ssd_enabled' field to True

get_all_metadata ()

Returns Dictionary of all keys and values associated as metadata for this object

get_allocated_physical_capacity (***kwargs*)

Obtains the value of the 'allocated_physical_capacity' field

Returns

Capacity

get_capacity_savings (***kwargs*)

Obtains the value of the 'capacity_savings' field

Returns

Capacity

get_created_at (***kwargs*)

Obtains the value of the 'created_at' field

Returns Arrow

classmethod get_creation_defaults ()

Returns a dict representing the default arguments as implicitly constructed by infinisdk to fulfill a `create` call

Note: This will cause generation of defaults, which will have side effects if they are special values

Note: This does not necessarily generate all fields that are passable into `create`, only mandatory 'fields'

get_entities_count (***kwargs*)

Obtains the value of the 'entities_count' field

Returns int

get_field (*field_name, from_cache=<DONT_CARE>, fetch_if_not_cached=True, raw_value=False*)

Gets the value of a single field from the system

Parameters

- **cache** – Attempt to use the last cached version of the field value
- **fetch_if_not_cached** – Pass `False` to force only from cache

get_fields (*field_names=()*, *from_cache=<DONT_CARE>*, *fetch_if_not_cached=True*,
raw_value=False)

Gets a set of fields from the system

Parameters

- **from_cache** – Attempt to fetch the fields from the cache
- **fetch_if_not_cached** – pass as `False` to force only from cache

Returns a dictionary of field names to their values

get_filesystem_snapshots_count (***kwargs*)

Obtains the value of the ‘filesystem_snapshots_count’ field

Returns int

get_filesystems_count (***kwargs*)

Obtains the value of the ‘filesystems_count’ field

Returns int

get_free_physical_capacity (***kwargs*)

Obtains the value of the ‘free_physical_capacity’ field

Returns

Capacity

get_free_virtual_capacity (***kwargs*)

Obtains the value of the ‘free_virtual_capacity’ field

Returns

Capacity

get_id (***kwargs*)

Obtains the value of the ‘id’ field

Returns int

get_max_extend (***kwargs*)

Obtains the value of the ‘max_extend’ field

Returns

Capacity

get_metadata_value (*key*, *default=<NOTHING>*)

Gets a metadata value, optionally specifying a default

Parameters default – if specified, the value to retrieve if the metadata key doesn’t exist. if not specified, and the key does not exist, the operation will raise an exception

get_name (***kwargs*)

Obtains the value of the ‘name’ field

Returns str

get_owners (***kwargs*)

Obtains the value of the ‘owners’ field

Returns list

get_physical_capacity (**kwargs)
Obtains the value of the 'physical_capacity' field

Returns

Capacity

get_physical_capacity_critical (**kwargs)
Obtains the value of the 'physical_capacity_critical' field

Returns int

get_physical_capacity_warning (**kwargs)
Obtains the value of the 'physical_capacity_warning' field

Returns int

get_qos_policies (**kwargs)
Obtains the value of the 'qos_policies' field

Returns list

get_reserved_capacity (**kwargs)
Obtains the value of the 'reserved_capacity' field

Returns

Capacity

get_snapshots_count (**kwargs)
Obtains the value of the 'snapshots_count' field

Returns int

get_standard_entities_count (**kwargs)
Obtains the value of the 'standard_entities_count' field

Returns int

get_standard_filesystem_snapshots_count (**kwargs)
Obtains the value of the 'standard_filesystem_snapshots_count' field

Returns int

get_standard_filesystems_count (**kwargs)
Obtains the value of the 'standard_filesystems_count' field

Returns int

get_standard_snapshots_count (**kwargs)
Obtains the value of the 'standard_snapshots_count' field

Returns int

get_standard_volumes_count (**kwargs)
Obtains the value of the 'standard_volumes_count' field

Returns int

get_state (**kwargs)
Obtains the value of the 'state' field

Returns str

get_tenant (***kwargs*)
 Obtains the value of the ‘tenant’ field
 Returns *infinisdk.infinibox.tenant.Tenant object*

get_type (***kwargs*)
 Obtains the value of the ‘type’ field
 Returns *str*

get_updated_at (***kwargs*)
 Obtains the value of the ‘updated_at’ field
 Returns *Arrow*

get_virtual_capacity (***kwargs*)
 Obtains the value of the ‘virtual_capacity’ field
 Returns
Capacity

get_volumes_count (***kwargs*)
 Obtains the value of the ‘volumes_count’ field
 Returns *int*

get_vvol_entities_count (***kwargs*)
 Obtains the value of the ‘vvol_entities_count’ field
 Returns *int*

get_vvol_snapshots_count (***kwargs*)
 Obtains the value of the ‘vvol_snapshots_count’ field
 Returns *int*

get_vvol_volumes_count (***kwargs*)
 Obtains the value of the ‘vvol_volumes_count’ field
 Returns *int*

invalidate_cache (**field_names*)
 Discards the cached field values of this object, causing the next fetch to retrieve the fresh value from the system

is_compression_enabled (***kwargs*)
 Obtains the value of the ‘compression_enabled’ field
 Returns *bool*

is_in_system ()
 Returns whether or not the object actually exists

is_ssd_enabled (***kwargs*)
 Obtains the value of the ‘ssd_enabled’ field
 Returns *bool*

safe_delete (**args, **kwargs*)
 Tries to delete the object, doing nothing if the object cannot be found on the system

safe_get_field (*field_name, default=<NOTHING>, **kwargs*)
 Like *get_field()*, only returns ‘default’ parameter if no result was found

set_metadata (*key, value*)
Sets metadata key in the system associated with this object

set_metadata_from_dict (*data_dict*)
Sets multiple metadata keys/values in the system associated with this object

set_owners (*users*)
sets the owners of this pool, replacing previous owners

unset_metadata (*key*)
Deletes a metadata key for this object

update_compression_enabled
Updates the value of the 'compression_enabled' field

param value The new compression_enabled value to be set (type: bool)

update_field (*field_name, field_value*)
Updates the value of a single field

update_fields (***update_dict*)
Atomically updates a group of fields and respective values (given as a dictionary)

update_max_extend (*value, **kwargs*)
Updates the value of the 'max_extend' field

Parameters value – The new max_extend value to be set (type: [Capacity](#))

update_name (*value, **kwargs*)
Updates the value of the 'name' field

Parameters value – The new name value to be set (type: str)

update_physical_capacity (*value, **kwargs*)
Updates the value of the 'physical_capacity' field

Parameters value – The new physical_capacity value to be set (type: [Capacity](#))

update_physical_capacity_critical (*value, **kwargs*)
Updates the value of the 'physical_capacity_critical' field

Parameters value – The new physical_capacity_critical value to be set (type: int)

update_physical_capacity_warning (*value, **kwargs*)
Updates the value of the 'physical_capacity_warning' field

Parameters value – The new physical_capacity_warning value to be set (type: int)

update_ssd_enabled
Updates the value of the 'ssd_enabled' field

param value The new ssd_enabled value to be set (type: bool)

update_virtual_capacity (*value, **kwargs*)
Updates the value of the 'virtual_capacity' field

Parameters value – The new virtual_capacity value to be set (type: [Capacity](#))

2.27.12 infinibox.hosts

infinibox.hosts is of type [HostBinder](#) described below.

class infinisdk.infinibox.host.**HostBinder** (*object_type, system*)
Implements *system.hosts*

choose (*predicates, **kw)

Chooses a random element out of those returned. Raises `ObjectNotFound` if none were returned

create (*args, **kwargs)

Creates an object on the system

find (*predicates, **kw)

Queries objects according to predicates. Can receive arguments in two possible forms:

1. Direct keyword arguments, for filtering for equality:

```
system.volumes.find(size=GiB)
```

2. Complex predicates, using the comparators:

```
system.volumes.find(system.volumes.fields.size > GiB)
system.volumes.find(Q.name != 'some_name')
```

Returns Lazy query result object.

See also:

[`infinisdk.core.object_query.ObjectQuery`](#)

get (*predicates, **kw)

Finds exactly one object matching criteria. Raises `ObjectNotFound` if not found, `TooManyObjectsFound` if more than one is found

get_by_id (id)

Obtains an object with a specific id

get_by_id_lazy (id)

Obtains an object with a specified id *without* checking if it exists or querying it on the way.

This is useful assuming the next operation is a further query/update on this object.

get_host_by_initiator_address (address)

Returns a host object defined on a system having the specified FC address configured, `None` if none exists

get_host_id_by_initiator_address (address)

Returns an id of a host object defined on a system having the specified FC address configured, `None` if none exists

get_mutable_fields ()

Returns a list of all mutable fields for this object type

has_registered_initiator_address (address)

Returns whether or not there exists a host object on the system with the specified FC address configured

safe_choose (*predicates, **kw)

Like `choose`, but returns `None` when not found

safe_get (*predicates, **kw)

Like `get ()`, only returns `None` if no objects were found

safe_get_by_id (id)

Like `get_by_id`, only returning `None` if the object could not be found

sample (*predicates, **kw)

Chooses a random sample out of those returned. Raises ValueError if there are not enough items

to_list ()

Returns the entire set of objects as a Python list

Caution: Queries are lazy by default to avoid heavy API calls and repetitive page requests. Using `to_list` will forcibly iterate and fetch all objects, which might be a very big collection. This can cause issues like slowness and memory exhaustion

Individual host objects are of type *Host*:

class infinisdk.infinibox.host.**Host** (system, initial_data)

BINDER_CLASS

alias of *HostBinder*

add_port (address)

Adds a port address to this host

Parameters address (Either an `infi.dtypes.wwn.WWN` or `infi.dtypes.iqn.iSCSIName`. Plain strings are assumed to be WWNs) – the port address to add

clear_metadata ()

Deletes all metadata keys for this object

classmethod construct (system, data)

Template method to enable customizing the object instantiation process.

This enables system components to be cached rather than re-fetched every time

classmethod create (system, **fields)

Creates a new object of this type

delete (**kwargs)

Deletes this object.

disable_optimized (**kwargs)

Set the value of the 'optimized' field to False

enable_optimized (**kwargs)

Set the value of the 'optimized' field to True

get_all_metadata ()

Returns Dictionary of all keys and values associated as metadata for this object

get_cluster (**kwargs)

Obtains the value of the 'cluster' field

Returns `infinisdk.infinibox.host_cluster.HostCluster` object

get_created_at (**kwargs)

Obtains the value of the 'created_at' field

Returns Arrow

classmethod get_creation_defaults ()

Returns a dict representing the default arguments as implicitly constructed by infinisdk to fulfill a create call

Note: This will cause generation of defaults, which will have side effects if they are special values

Note: This does not necessarily generate all fields that are passable into `create`, only mandatory ‘fields

get_field (*field_name*, *from_cache*=<DONT_CARE>, *fetch_if_not_cached*=True, *raw_value*=False)
 Gets the value of a single field from the system

Parameters

- **cache** – Attempt to use the last cached version of the field value
- **fetch_if_not_cached** – Pass False to force only from cache

get_fields (*field_names*=(), *from_cache*=<DONT_CARE>, *fetch_if_not_cached*=True, *raw_value*=False)
 Gets a set of fields from the system

Parameters

- **from_cache** – Attempt to fetch the fields from the cache
- **fetch_if_not_cached** – pass as False to force only from cache

Returns a dictionary of field names to their values

get_host_type (***kwargs*)
 Obtains the value of the ‘host_type’ field

Returns str

get_id (***kwargs*)
 Obtains the value of the ‘id’ field

Returns int

get_luns (**args*, ***kwargs*)
 Returns all LUNs mapped to this object

Returns A collection of *LogicalUnit* objects

get_metadata_value (*key*, *default*=<NOTHING>)
 Gets a metadata value, optionally specifying a default

Parameters default – if specified, the value to retrieve if the metadata key doesn’t exist. if not specified, and the key does not exist, the operation will raise an exception

get_name (***kwargs*)
 Obtains the value of the ‘name’ field

Returns str

get_ports (***kwargs*)
 Obtains the value of the ‘ports’ field

Returns list

get_san_client_type (***kwargs*)
 Obtains the value of the ‘san_client_type’ field

Returns str

get_security_chap_inbound_username (***kwargs*)
Obtains the value of the 'security_chap_inbound_username' field

Returns str

get_security_chap_outbound_username (***kwargs*)
Obtains the value of the 'security_chap_outbound_username' field

Returns str

get_security_method (***kwargs*)
Obtains the value of the 'security_method' field

Returns str

get_subsystem_nqn (***kwargs*)
Obtains the value of the 'subsystem_nqn' field

Returns str

get_tenant (***kwargs*)
Obtains the value of the 'tenant' field

Returns *infinisdk.infinibox.tenant.Tenant* object

get_updated_at (***kwargs*)
Obtains the value of the 'updated_at' field

Returns Arrow

invalidate_cache (**field_names*)
Discards the cached field values of this object, causing the next fetch to retrieve the fresh value from the system

is_in_system ()
Returns whether or not the object actually exists

is_optimized (***kwargs*)
Obtains the value of the 'optimized' field

Returns bool

is_security_chap_has_inbound_secret (***kwargs*)
Obtains the value of the 'security_chap_has_inbound_secret' field

Returns bool

is_security_chap_has_outbound_secret (***kwargs*)
Obtains the value of the 'security_chap_has_outbound_secret' field

Returns bool

is_volume_mapped (*volume*)
Returns whether or not a given volume is mapped to this object

map_volume (*volume, lun=None*)
Maps a volume to this object, possibly specifying the logical unit number (LUN) to use

Returns a *LogicalUnit* object representing the added LUN

remove_port (*address*)
Removes a port address to this host

safe_delete (**args, **kwargs*)
Tries to delete the object, doing nothing if the object cannot be found on the system

safe_get_field (*field_name*, *default=<NOTHING>*, ***kwargs*)

Like *get_field()*, only returns 'default' parameter if no result was found

set_metadata (*key*, *value*)

Sets metadata key in the system associated with this object

set_metadata_from_dict (*data_dict*)

Sets multiple metadata keys/values in the system associated with this object

unmap_volume (*volume=None*, *lun=None*)

Unmaps a volume either by specifying the volume or the lun it occupies

unset_metadata (*key*)

Deletes a metadata key for this object

update_field (*field_name*, *field_value*)

Updates the value of a single field

update_fields (***update_dict*)

Atomically updates a group of fields and respective values (given as a dictionary)

update_host_type (*value*, ***kwargs*)

Updates the value of the 'host_type' field

Parameters value – The new host_type value to be set (type: str)

update_name (*value*, ***kwargs*)

Updates the value of the 'name' field

Parameters value – The new name value to be set (type: str)

update_optimized

Updates the value of the 'optimized' field

param value The new optimized value to be set (type: bool)

update_security_chap_inbound_secret (*value*, ***kwargs*)

Updates the value of the 'security_chap_inbound_secret' field

Parameters value – The new security_chap_inbound_secret value to be set (type: str)

update_security_chap_inbound_username (*value*, ***kwargs*)

Updates the value of the 'security_chap_inbound_username' field

Parameters value – The new security_chap_inbound_username value to be set (type: str)

update_security_chap_outbound_secret (*value*, ***kwargs*)

Updates the value of the 'security_chap_outbound_secret' field

Parameters value – The new security_chap_outbound_secret value to be set (type: str)

update_security_chap_outbound_username (*value*, ***kwargs*)

Updates the value of the 'security_chap_outbound_username' field

Parameters value – The new security_chap_outbound_username value to be set (type: str)

update_security_method (*value*, ***kwargs*)

Updates the value of the 'security_method' field

Parameters value – The new security_method value to be set (type: str)

2.27.13 infinibox.clusters

class `infinisdk.infinibox.host_cluster.HostCluster` (*system, initial_data*)

get_created_at (***kwargs*)

Obtains the value of the ‘created_at’ field

Returns Arrow

get_host_type (***kwargs*)

Obtains the value of the ‘host_type’ field

Returns str

get_hosts (***kwargs*)

Obtains the value of the ‘hosts’ field

Returns list

get_id (***kwargs*)

Obtains the value of the ‘id’ field

Returns int

get_name (***kwargs*)

Obtains the value of the ‘name’ field

Returns str

get_san_client_type (***kwargs*)

Obtains the value of the ‘san_client_type’ field

Returns str

get_tenant (***kwargs*)

Obtains the value of the ‘tenant’ field

Returns `infinisdk.infinibox.tenant.Tenant` object

get_updated_at (***kwargs*)

Obtains the value of the ‘updated_at’ field

Returns Arrow

update_name (*value, **kwargs*)

Updates the value of the ‘name’ field

Parameters **value** – The new name value to be set (type: str)

2.27.14 infinibox.replicas

class `infinisdk.infinibox.replica.ReplicaBinder` (*object_type, system*)

Implements `system.replicas`

replicate_cons_group (*cg, remote_cg=None, remote_pool=<OMIT>, **kw*)

Convenience wrapper around `ReplicaBinder.replicate_entity()`

Seealso `replicate_entity()`

replicate_entity (*entity, link, remote_pool=None, remote_entity=None, **kw*)

Replicates a entity or CG, creating its remote replica on the specified pool

Parameters

- **remote_pool** – if omitted, `remote_entity` must be specified. Otherwise, means creating target entity
- **remote_entity** – if omitted, `remote_pool` must be specified. Otherwise, means creating based on existing entity on target
- **member_mappings** – required if `remote_entity` is specified and is a consistency group. This parameter is a dictionary mapping local member entities to remote ones

replicate_entity_create_target (*entity*, *link*, *remote_pool=<OMIT>*, *remote_entity_names=<OMIT>*, ***kw*)

Replicates an entity, creating its remote replica on the specified pool

Parameters

- **remote_pool** – Remote pool to use for entity creation on the remote side
- **remote_entity_names** – A list or tuple containing the entity names created on remote side. In case of cg - controls the entity names, not the cg name.

replicate_entity_existing_target (*entity*, *link*, *remote_entity*, *member_mappings=None*, ***kw*)

Replicates an entity, using a formatted/empty entity on the other side

Parameters

- **remote_entity** – Remote entity to use for replication
- **member_mappings** – required if `remote_entity` is specified and is a consistency group. This parameter is a dictionary mapping local member entities to remote ones

replicate_entity_take_snap (*entity*, *link*, *remote_entity*, *member_mappings=None*, ***kw*)

Replicates a entity, using the currently found data on both sides as a reference.

Parameters

- **entity** – Local entity to use
- **remote_entity** – Remote entity to use
- **member_mappings** – required if `remote_entity` is specified and is a consistency group. This parameter is a dictionary mapping local member entities to tuples of (`entity`, `remote_entity`)

replicate_entity_use_base (*entity*, *link*, *local_snapshot*, *remote_snapshot*, *member_mappings=None*, ***kw*)

Replicates an entity, using an existing remote entity and a base snapshot on both sides

Parameters

- **local_snapshot** – Local base snapshot to use
- **remote_snapshot** – Remote base snapshot to use
- **member_mappings** – required if `remote_entity` is specified and is a consistency group. This parameter is a dictionary mapping local member entities to tuples of (`local_snapshot`, `remote_snapshot`)

replicate_volume (*volume*, *remote_volume=None*, ***kw*)

Convenience wrapper around `ReplicaBinder.replicate_entity()`

Seealso `replicate_entity()`

class `infinisdk.infinibox.replica.Replica` (*system*, *initial_data*)

BINDER_CLASS

alias of *ReplicaBinder*

change_role (*entity_pairs=<OMIT>*)

Changes the role of this replica from source to target or vice-versa

change_type_to_async (*params=None*)

Changes the replication type to ASYNC

Parameters **params** – Optional dictionary containing additional parameters for the type change

change_type_to_sync (*params=None*)

Changes the replication type to SYNC

Parameters **params** – Optional dictionary containing additional parameters for the type change

delete (*retain_staging_area=<OMIT>*, *force_if_remote_error=<OMIT>*, *force_on_target=<OMIT>*,
force_if_no_remote_credentials=<OMIT>, *force_on_local=<OMIT>*,
keep_serial_on_local=<OMIT>)

Deletes this replica

disable_preferred (***kwargs*)

Set the value of the 'preferred' field to False

enable_preferred (***kwargs*)

Set the value of the 'preferred' field to True

get_assigned_sync_remote_ips (***kwargs*)

Obtains the value of the 'assigned_sync_remote_ips' field

Returns list

get_base_action (***kwargs*)

Obtains the value of the 'base_action' field

Returns str

get_created_at (***kwargs*)

Obtains the value of the 'created_at' field

Returns Arrow

get_description (***kwargs*)

Obtains the value of the 'description' field

Returns str

get_entity_pairs (***kwargs*)

Obtains the value of the 'entity_pairs' field

Returns list

get_entity_type (***kwargs*)

Obtains the value of the 'entity_type' field

Returns str

get_id (***kwargs*)

Obtains the value of the 'id' field

Returns int

get_job_state (***kwargs*)

Obtains the value of the 'job_state' field

Returns str

get_jobs (***kwargs*)
Obtains the value of the 'jobs' field

Returns list

get_last_replicated_guid (***kwargs*)
Obtains the value of the 'last_replicated_guid' field

Returns str

get_last_synchronized (***kwargs*)
Obtains the value of the 'last_synchronized' field

Returns Arrow

get_latency (***kwargs*)
Obtains the value of the 'latency' field

Returns int

get_link (***kwargs*)
Obtains the value of the 'link' field

Returns *infinisdk.infinibox.link.Link object*

get_local_cg ()
Returns the local cg, assuming this is a consistency group replica

get_local_cg_id (***kwargs*)
Obtains the value of the 'local_cg_id' field

Returns int

get_local_cg_name (***kwargs*)
Obtains the value of the 'local_cg_name' field

Returns str

get_local_data_entities ()
Returns all local volumes, whether as part of a consistency group, filesystem or a single volume

get_local_entity ()
Returns the local entity used for replication, be it a volume, filesystem or a consistency group

get_local_entity_name (***kwargs*)
Obtains the value of the 'local_entity_name' field

Returns str

get_local_filesystem ()
Returns the local volume, assuming there is exactly one

get_local_pool_id (***kwargs*)
Obtains the value of the 'local_pool_id' field

Returns int

get_local_pool_name (***kwargs*)
Obtains the value of the 'local_pool_name' field

Returns str

get_local_volume ()
Returns the local volume, assuming there is exactly one

get_lock_remote_snapshot_retention (***kwargs*)
 Obtains the value of the 'lock_remote_snapshot_retention' field
Returns timedelta

get_next_job_start_time (***kwargs*)
 Obtains the value of the 'next_job_start_time' field
Returns Arrow

get_next_restore_point (***kwargs*)
 Obtains the value of the 'next_restore_point' field
Returns Arrow

get_pending_job_count (***kwargs*)
 Obtains the value of the 'pending_job_count' field
Returns int

get_permanent_failure_wait_interval (***kwargs*)
 Obtains the value of the 'permanent_failure_wait_interval' field
Returns timedelta

get_progress (***kwargs*)
 Obtains the value of the 'progress' field
Returns int

get_remote_cg_id (***kwargs*)
 Obtains the value of the 'remote_cg_id' field
Returns int

get_remote_cg_name (***kwargs*)
 Obtains the value of the 'remote_cg_name' field
Returns str

get_remote_data_entities (*from_cache=False*)
 Returns all local volumes, whether as part of a consistency group, filesystem or a single volume

get_remote_entity (*from_cache=False, safe=False*)
 Fetches the remote replicated entity if available

get_remote_entity_name (***kwargs*)
 Obtains the value of the 'remote_entity_name' field
Returns str

get_remote_entity_pairs ()
 Returns the entity_pairs configuration as held by the remote replica

Note: this uses the remote command execution API to run the command over the inter-system link

get_remote_pool_id (***kwargs*)
 Obtains the value of the 'remote_pool_id' field
Returns int

get_remote_pool_name (***kwargs*)
 Obtains the value of the 'remote_pool_name' field

Returns str

get_remote_replica (*from_cache=False, safe=False*)

Get the corresponding replica object in the remote machine. For this to work, the SDK user should call the `register_related_system` method of the `Infinibox` object when a link to a remote system is constructed for the first time

get_remote_replica_id (***kwargs*)

Obtains the value of the 'remote_replica_id' field

Returns int

get_remote_snapshot_suffix (***kwargs*)

Obtains the value of the 'remote_snapshot_suffix' field

Returns str

get_replication_type (***kwargs*)

Obtains the value of the 'replication_type' field

Returns str

get_restore_point (***kwargs*)

Obtains the value of the 'restore_point' field

Returns Arrow

get_role (***kwargs*)

Obtains the value of the 'role' field

Returns str

get_rpo (***kwargs*)

Obtains the value of the 'rpo' field

Returns timedelta

get_rpo_state (***kwargs*)

Obtains the value of the 'rpo_state' field

Returns str

get_rpo_type (***kwargs*)

Obtains the value of the 'rpo_type' field

Returns str

get_snapshots_retention (***kwargs*)

Obtains the value of the 'snapshots_retention' field

Returns int

get_staging_area_allocated_size (***kwargs*)

Obtains the value of the 'staging_area_allocated_size' field

Returns

Capacity

get_started_at (***kwargs*)

Obtains the value of the 'started_at' field

Returns Arrow

get_state (***kwargs*)

Obtains the value of the 'state' field

Returns str

get_state_description (**kwargs)
Obtains the value of the 'state_description' field

Returns str

get_state_reason (**kwargs)
Obtains the value of the 'state_reason' field

Returns str

get_sync_duration (**kwargs)
Obtains the value of the 'sync_duration' field

Returns int

get_sync_interval (**kwargs)
Obtains the value of the 'sync_interval' field

Returns timedelta

get_sync_state (**kwargs)
Obtains the value of the 'sync_state' field

Returns str

get_temporary_failure_retry_count (**kwargs)
Obtains the value of the 'temporary_failure_retry_count' field

Returns int

get_temporary_failure_retry_interval (**kwargs)
Obtains the value of the 'temporary_failure_retry_interval' field

Returns timedelta

get_throughput (**kwargs)
Obtains the value of the 'throughput' field

Returns int

get_updated_at (**kwargs)
Obtains the value of the 'updated_at' field

Returns Arrow

is_active (**kwargs)
Returns whether or not the replica is currently active

is_async_mode (**kwargs)
Obtains the value of the 'async_mode' field

Returns bool

is_auto_suspended (**kwargs)
Returns whether or not this replica is in auto_suspended state

is_concurrent_replica (**kwargs)
Obtains the value of the 'concurrent_replica' field

Returns bool

is_consistency_group ()
Returns whether this replica is configured with a consistency group as a local entity

is_domino (**kwargs)
Obtains the value of the 'domino' field

Returns bool

is_filesystem ()
Returns True if this replica replicates a single filesystem entity

is_idle (**kwargs)
Returns whether or not this replica is in idle state

is_including_snapshots (**kwargs)
Obtains the value of the 'including_snapshots' field

Returns bool

is_initial (**kwargs)
Obtains the value of the 'initial' field

Returns bool

is_initial_replication (**kwargs)
Returns whether or not this replica is in initiating state

is_initializing ()
Returns True if the replica sync state is 'INITIALIZING'

is_initializing_pending ()
Returns True if the replica sync state is 'INITIALIZING_PENDING'

is_out_of_sync ()
Returns True if the replica sync state is 'OUT_OF_SYNC'

is_pending (**kwargs)
Returns whether or not this replication is waiting to start initializing

is_preferred (**kwargs)
Obtains the value of the 'preferred' field

Returns bool

is_replicating (**kwargs)
Returns whether or not this replica is in replicating state

is_source (*args, **kwargs)
A predicate returning whether or not the replica is currently in the "source" role

is_suspended (**kwargs)
Returns whether or not this replica is currently suspended

is_suspended_from_local (**kwargs)
Obtains the value of the 'suspended_from_local' field

Returns bool

is_sync_in_progress ()
Returns True if this replica sync state is 'SYNC_IN_PROGRESS'

is_synchronized ()
Returns True if this replica sync state is 'SYNCHRONIZED'

is_target (*args, **kwargs)
A predicate returning whether or not the replica is currently in the "target" role

is_user_suspended (***kwargs*)
Returns whether or not this replica is currently suspended due to a user request

is_volume ()
Returns True if this replica replicates a single volume entity

resume ()
Resumes this replica

suspend ()
Suspends this replica

switch_role ()
Switches replica role - sync replicas only

sync ()
Starts a sync job

update_description (*value, **kwargs*)
Updates the value of the 'description' field

Parameters value – The new description value to be set (type: str)

update_lock_remote_snapshot_retention (*value, **kwargs*)
Updates the value of the 'lock_remote_snapshot_retention' field

Parameters value – The new lock_remote_snapshot_retention value to be set (type: timedelta)

update_permanent_failure_wait_interval (*value, **kwargs*)
Updates the value of the 'permanent_failure_wait_interval' field

Parameters value – The new permanent_failure_wait_interval value to be set (type: timedelta)

update_preferred
Updates the value of the 'preferred' field

param value The new preferred value to be set (type: bool)

update_remote_snapshot_suffix (*value, **kwargs*)
Updates the value of the 'remote_snapshot_suffix' field

Parameters value – The new remote_snapshot_suffix value to be set (type: str)

update_rpo (*value, **kwargs*)
Updates the value of the 'rpo' field

Parameters value – The new rpo value to be set (type: timedelta)

update_snapshots_retention (*value, **kwargs*)
Updates the value of the 'snapshots_retention' field

Parameters value – The new snapshots_retention value to be set (type: int)

update_sync_interval (*value, **kwargs*)
Updates the value of the 'sync_interval' field

Parameters value – The new sync_interval value to be set (type: timedelta)

update_temporary_failure_retry_count (*value, **kwargs*)
Updates the value of the 'temporary_failure_retry_count' field

Parameters value – The new temporary_failure_retry_count value to be set (type: int)

update_temporary_failure_retry_interval (*value, **kwargs*)
Updates the value of the 'temporary_failure_retry_interval' field

Parameters **value** – The new temporary_failure_retry_interval value to be set (type: timedelta)

2.27.15 infinibox.links

```
class infinisdk.infinibox.link.LinkBinder (*args, **kwargs)
```

```
class infinisdk.infinibox.link.Link (system, initial_data)
```

BINDER_CLASS

alias of *LinkBinder*

```
delete (force_if_remote_error=False, force_if_no_remote_credentials=False)
```

Deletes this link

Parameters

- **force_if_remote_error** – forces deletion even if remote side caused an API error
- **force_if_no_remote_credentials** – forces deletion even if no credentials to remote side

```
get_id (**kwargs)
```

Obtains the value of the 'id' field

Returns int

```
get_link_mode (**kwargs)
```

Obtains the value of the 'link_mode' field

Returns str

```
get_link_state (**kwargs)
```

Obtains the value of the 'link_state' field

Returns str

```
get_linked_system (safe=False, from_cache=<DONT_CARE>)
```

Get the corresponding system object at the remote and of the link. For this to work, the SDK user should call the register_related_system method of the Infinibox object when a link to a remote system is constructed for the first time

```
get_local_replication_network_space (**kwargs)
```

Obtains the value of the 'local_replication_network_space' field

Returns *infinisdk.infinibox.network_space.NetworkSpace* object

```
get_local_witness_state (**kwargs)
```

Obtains the value of the 'local_witness_state' field

Returns str

```
get_name (**kwargs)
```

Obtains the value of the 'name' field

Returns str

```
get_remote_host (**kwargs)
```

Obtains the value of the 'remote_host' field

Returns str

```

get_remote_link (safe=False)
    Get the corresponding link object in the remote machine

get_remote_link_id (**kwargs)
    Obtains the value of the 'remote_link_id' field

    Returns int

get_remote_system_name (**kwargs)
    Obtains the value of the 'remote_system_name' field

    Returns str

get_remote_system_serial_number (**kwargs)
    Obtains the value of the 'remote_system_serial_number' field

    Returns int

get_resiliency_mode (**kwargs)
    Obtains the value of the 'resiliency_mode' field

    Returns str

get_state_description (**kwargs)
    Obtains the value of the 'state_description' field

    Returns str

get_witness_address (**kwargs)
    Obtains the value of the 'witness_address' field

    Returns str

is_async_only (**kwargs)
    Obtains the value of the 'async_only' field

    Returns bool

is_is_local_link_ready_for_sync (**kwargs)
    Obtains the value of the 'is_local_link_ready_for_sync' field

    Returns bool

update_local_replication_network_space (value, **kwargs)
    Updates the value of the 'local_replication_network_space' field

    Parameters value – The new local_replication_network_space value to be set (type:
        infinisdk.infinibox.network_space.NetworkSpace object)

update_name (value, **kwargs)
    Updates the value of the 'name' field

    Parameters value – The new name value to be set (type: str)

update_remote_host (value, **kwargs)
    Updates the value of the 'remote_host' field

    Parameters value – The new remote_host value to be set (type: str)

```

2.27.16 infinibox.network_spaces

```

class infinisdk.infinibox.network_space.NetworkSpace (system, initial_data)

```

disable_automatic_ip_failback (**kwargs)
Set the value of the 'automatic_ip_failback' field to False

enable_automatic_ip_failback (**kwargs)
Set the value of the 'automatic_ip_failback' field to True

get_id (**kwargs)
Obtains the value of the 'id' field
Returns int

get_interfaces (**kwargs)
Obtains the value of the 'interfaces' field
Returns list

get_ips (**kwargs)
Obtains the value of the 'ips' field
Returns list

get_mtu (**kwargs)
Obtains the value of the 'mtu' field
Returns int

get_name (**kwargs)
Obtains the value of the 'name' field
Returns str

get_network_config (**kwargs)
Obtains the value of the 'network_config' field
Returns Munch

get_properties (**kwargs)
Obtains the value of the 'properties' field
Returns Munch

get_rate_limit (**kwargs)
Obtains the value of the 'rate_limit' field
Returns int

get_service (**kwargs)
Obtains the value of the 'service' field
Returns str

get_tenant (**kwargs)
Obtains the value of the 'tenant' field
Returns *infinisdk.infinibox.tenant.Tenant object*

is_automatic_ip_failback (**kwargs)
Obtains the value of the 'automatic_ip_failback' field
Returns bool

update_automatic_ip_failback
Updates the value of the 'automatic_ip_failback' field
param value The new automatic_ip_failback value to be set (type: bool)

update_interfaces (*value*, ***kwargs*)
 Updates the value of the ‘interfaces’ field

Parameters **value** – The new interfaces value to be set (type: list)

update_mtu (*value*, ***kwargs*)
 Updates the value of the ‘mtu’ field

Parameters **value** – The new mtu value to be set (type: int)

update_name (*value*, ***kwargs*)
 Updates the value of the ‘name’ field

Parameters **value** – The new name value to be set (type: str)

update_network_config (*value*, ***kwargs*)
 Updates the value of the ‘network_config’ field

Parameters **value** – The new network_config value to be set (type: Munch)

update_properties (*value*, ***kwargs*)
 Updates the value of the ‘properties’ field

Parameters **value** – The new properties value to be set (type: Munch)

update_rate_limit (*value*, ***kwargs*)
 Updates the value of the ‘rate_limit’ field

Parameters **value** – The new rate_limit value to be set (type: int)

2.27.17 infinibox.events

class infinisdk.core.events.**Events** (*system*)

class infinisdk.core.events.**Event** (*system*, *initial_data*)

BINDER_CLASS
 alias of *Events*

get_affected_entity_id (***kwargs*)
 Obtains the value of the ‘affected_entity_id’ field

Returns str

get_code (***kwargs*)
 Obtains the value of the ‘code’ field

Returns str

get_description (***kwargs*)
 Obtains the value of the ‘description’ field

Returns str

get_description_template (***kwargs*)
 Obtains the value of the ‘description_template’ field

Returns str

get_id (***kwargs*)
 Obtains the value of the ‘id’ field

Returns int

```

get_level (**kwargs)
    Obtains the value of the 'level' field

    Returns str

get_reporter (**kwargs)
    Obtains the value of the 'reporter' field

    Returns str

get_source_node_id (**kwargs)
    Obtains the value of the 'source_node_id' field

    Returns int

get_system_version (**kwargs)
    Obtains the value of the 'system_version' field

    Returns str

get_tenant (**kwargs)
    Obtains the value of the 'tenant' field

    Returns infinisdk.infinibox.tenant.Tenant object

get_timestamp (**kwargs)
    Obtains the value of the 'timestamp' field

    Returns Arrow

get_username (**kwargs)
    Obtains the value of the 'username' field

    Returns str

get_visibility (**kwargs)
    Obtains the value of the 'visibility' field

    Returns str

```

```

class infinisdk.infinibox.events.Events (system)

```

2.27.18 infinbox.users

```

class infinisdk.infinibox.user.User (system, initial_data)

```

```

BINDER_CLASS
    alias of infinisdk.core.type_binder.TypeBinder

classmethod construct (system, data)
    Template method to enable customizing the object instantiation process.

    This enables system components to be cached rather than re-fetched every time

classmethod create (system, **fields)
    Creates a new object of this type

delete (**kwargs)
    Deletes this object.

disable (**kwargs)
    Set the value of the 'enabled' field to False

```

enable (***kwargs*)

Set the value of the ‘enabled’ field to True

classmethod get_creation_defaults ()

Returns a dict representing the default arguments as implicitly constructed by infinisdk to fulfill a `create` call

Note: This will cause generation of defaults, which will have side effects if they are special values

Note: This does not necessarily generate all fields that are passable into `create`, only mandatory ‘fields’

get_email (***kwargs*)

Obtains the value of the ‘email’ field

Returns str

get_field (*field_name*, *from_cache=<DONT_CARE>*, *fetch_if_not_cached=True*,
raw_value=False)

Gets the value of a single field from the system

Parameters

- **cache** – Attempt to use the last cached version of the field value
- **fetch_if_not_cached** – Pass `False` to force only from cache

get_fields (*field_names=()*, *from_cache=<DONT_CARE>*, *fetch_if_not_cached=True*,
raw_value=False)

Gets a set of fields from the system

Parameters

- **from_cache** – Attempt to fetch the fields from the cache
- **fetch_if_not_cached** – pass as `False` to force only from cache

Returns a dictionary of field names to their values

get_id (***kwargs*)

Obtains the value of the ‘id’ field

Returns int

get_name (***kwargs*)

Obtains the value of the ‘name’ field

Returns str

get_owned_pools ()

Returns the pools that are owned by this user

get_password_digest_version (***kwargs*)

Obtains the value of the ‘password_digest_version’ field

Returns int

get_role (***kwargs*)

Obtains the value of the ‘role’ field

Returns str

get_roles (***kwargs*)
 Obtains the value of the ‘roles’ field
Returns list

get_type (***kwargs*)
 Obtains the value of the ‘type’ field
Returns str

invalidate_cache (**field_names*)
 Discards the cached field values of this object, causing the next fetch to retrieve the fresh value from the system

is_enabled (***kwargs*)
 Obtains the value of the ‘enabled’ field
Returns bool

is_in_system ()
 Returns whether or not the object actually exists

is_is_digest_sufficient (***kwargs*)
 Obtains the value of the ‘is_digest_sufficient’ field
Returns bool

safe_delete (**args, **kwargs*)
 Tries to delete the object, doing nothing if the object cannot be found on the system

safe_get_field (*field_name, default=<NOTHING>, **kwargs*)
 Like *get_field()*, only returns ‘default’ parameter if no result was found

update_email (*value, **kwargs*)
 Updates the value of the ‘email’ field
Parameters value – The new email value to be set (type: str)

update_enabled
 Updates the value of the ‘enabled’ field
param value The new enabled value to be set (type: bool)

update_field (*field_name, field_value*)
 Updates the value of a single field

update_fields (***update_dict*)
 Atomically updates a group of fields and respective values (given as a dictionary)

update_name (*value, **kwargs*)
 Updates the value of the ‘name’ field
Parameters value – The new name value to be set (type: str)

update_password (*value, **kwargs*)
 Updates the value of the ‘password’ field
Parameters value – The new password value to be set (type: str)

update_role (*value, **kwargs*)
 Updates the value of the ‘role’ field
Parameters value – The new role value to be set (type: str)

update_roles (*value, **kwargs*)
 Updates the value of the ‘roles’ field

Parameters **value** – The new roles value to be set (type: list)

2.27.19 infinibox.ldap_configs

class infinisdk.infinibox.ldap_config.**LDAPConfigBinder** (*object_type, system*)

define (**args, **kwargs*)
Alias for *.create*

set_order (*configs*)
Reorders LDAP configurations' priorities

class infinisdk.infinibox.ldap_config.**LDAPConfig** (*system, initial_data*)

BINDER_CLASS
alias of *LDAPConfigBinder*

create_group (*name, dn, role*)
Maps a specified group in the LDAP directory to a specified role in the system

get_id (***kwargs*)
Obtains the value of the 'id' field

Returns int

get_name (***kwargs*)
Obtains the value of the 'name' field

Returns str

modify (***kwargs*)
Modifies the LDAP configuration

test ()
Tests the LDAP configuration

update_name (*value, **kwargs*)
Updates the value of the 'name' field

Parameters **value** – The new name value to be set (type: str)

2.27.20 infinibox.notification_targets

class infinisdk.infinibox.notification_target.**NotificationTarget** (*system, initial_data*)

disable_tls (***kwargs*)
Set the value of the 'tls' field to False

enable_tls (***kwargs*)
Set the value of the 'tls' field to True

get_auth_protocol (***kwargs*)
Obtains the value of the 'auth_protocol' field

Returns str

get_auth_type (***kwargs*)
Obtains the value of the 'auth_type' field

Returns str

get_community (**kwargs)
Obtains the value of the ‘community’ field

Returns str

get_engine (**kwargs)
Obtains the value of the ‘engine’ field

Returns str

get_facility (**kwargs)
Obtains the value of the ‘facility’ field

Returns str

get_from_address (**kwargs)
Obtains the value of the ‘from_address’ field

Returns str

get_host (**kwargs)
Obtains the value of the ‘host’ field

Returns str

get_id (**kwargs)
Obtains the value of the ‘id’ field

Returns int

get_name (**kwargs)
Obtains the value of the ‘name’ field

Returns str

get_password (**kwargs)
Obtains the value of the ‘password’ field

Returns str

get_port (**kwargs)
Obtains the value of the ‘port’ field

Returns int

get_private_key (**kwargs)
Obtains the value of the ‘private_key’ field

Returns str

get_private_protocol (**kwargs)
Obtains the value of the ‘private_protocol’ field

Returns str

get_protocol (**kwargs)
Obtains the value of the ‘protocol’ field

Returns str

get_transport (**kwargs)
Obtains the value of the ‘transport’ field

Returns str

get_username (***kwargs*)
 Obtains the value of the ‘username’ field

Returns str

get_version (***kwargs*)
 Obtains the value of the ‘version’ field

Returns str

get_visibility (***kwargs*)
 Obtains the value of the ‘visibility’ field

Returns str

is_tls (***kwargs*)
 Obtains the value of the ‘tls’ field

Returns bool

test (*recipients=None*)
 Tests the SMTP gateway, by sending a test email to one or several recipients

Parameters **recipients** – Either a single email or a list of emails to send to (only for SMTP)

update_auth_protocol (*value, **kwargs*)
 Updates the value of the ‘auth_protocol’ field

Parameters **value** – The new auth_protocol value to be set (type: str)

update_auth_type (*value, **kwargs*)
 Updates the value of the ‘auth_type’ field

Parameters **value** – The new auth_type value to be set (type: str)

update_community (*value, **kwargs*)
 Updates the value of the ‘community’ field

Parameters **value** – The new community value to be set (type: str)

update_engine (*value, **kwargs*)
 Updates the value of the ‘engine’ field

Parameters **value** – The new engine value to be set (type: str)

update_facility (*value, **kwargs*)
 Updates the value of the ‘facility’ field

Parameters **value** – The new facility value to be set (type: str)

update_from_address (*value, **kwargs*)
 Updates the value of the ‘from_address’ field

Parameters **value** – The new from_address value to be set (type: str)

update_host (*value, **kwargs*)
 Updates the value of the ‘host’ field

Parameters **value** – The new host value to be set (type: str)

update_name (*value, **kwargs*)
 Updates the value of the ‘name’ field

Parameters **value** – The new name value to be set (type: str)

update_password (*value, **kwargs*)
 Updates the value of the ‘password’ field

Parameters value – The new password value to be set (type: str)

update_port (*value*, ***kwargs*)

Updates the value of the ‘port’ field

Parameters value – The new port value to be set (type: int)

update_private_key (*value*, ***kwargs*)

Updates the value of the ‘private_key’ field

Parameters value – The new private_key value to be set (type: str)

update_private_protocol (*value*, ***kwargs*)

Updates the value of the ‘private_protocol’ field

Parameters value – The new private_protocol value to be set (type: str)

update_tls

Updates the value of the ‘tls’ field

param value The new tls value to be set (type: bool)

update_transport (*value*, ***kwargs*)

Updates the value of the ‘transport’ field

Parameters value – The new transport value to be set (type: str)

update_username (*value*, ***kwargs*)

Updates the value of the ‘username’ field

Parameters value – The new username value to be set (type: str)

update_version (*value*, ***kwargs*)

Updates the value of the ‘version’ field

Parameters value – The new version value to be set (type: str)

update_visibility (*value*, ***kwargs*)

Updates the value of the ‘visibility’ field

Parameters value – The new visibility value to be set (type: str)

2.27.21 infinibox.cons_groups

class `infinisdk.infinibox.cons_group.ConsGroup` (*system*, *initial_data*)

add_member (*member*, ***kwargs*)

Adds a member data entity to this consistency group

Parameters remote_entity – Assuming this CG is currently being replicated, specifies the remote entity for the member replication

create_snapgroup (*name=None*, *prefix=None*, *suffix=None*, *lock_expires_at=None*)

Create a snapshot group out of the consistency group.

create_snapshot (*name=None*, *prefix=None*, *suffix=None*, *lock_expires_at=None*)

Create a snapshot group out of the consistency group.

delete (*delete_members=<OMIT>*, *force_if_snapshot_locked=<OMIT>*)

Deletes the consistency group

Parameters delete_members – if True, deletes the member datasets as well as the group itself

get_created_at (**kwargs)

Obtains the value of the ‘created_at’ field

Returns Arrow

get_id (**kwargs)

Obtains the value of the ‘id’ field

Returns int

get_lock_expires_at (**kwargs)

Obtains the value of the ‘lock_expires_at’ field

Returns Arrow

get_lock_state (**kwargs)

Obtains the value of the ‘lock_state’ field

Returns str

get_members (**kwargs)

Retrieves a lazy query for the consistency group’s member datasets

Note: in many cases you should prefer to collect the result of this method as a list using `to_list()`: ..
code-block:: python

```
member_list = cg.get_members().to_list()
```

Parameters **kwargs** – Optional parameter containing filterable fields of cg member for filtering the members returned

get_members_count (**kwargs)

Obtains the value of the ‘members_count’ field

Returns int

get_name (**kwargs)

Obtains the value of the ‘name’ field

Returns str

get_parent (**kwargs)

Obtains the value of the ‘parent’ field

Returns *infinisdk.infinibox.cons_group.ConsGroup object*

get_pool (**kwargs)

Obtains the value of the ‘pool’ field

Returns *infinisdk.infinibox.pool.Pool object*

get_pool_name (**kwargs)

Obtains the value of the ‘pool_name’ field

Returns str

get_replication_types (**kwargs)

Obtains the value of the ‘replication_types’ field

Returns list

get_rmr_snapshot_guid (**kwargs)

Obtains the value of the ‘rmr_snapshot_guid’ field

Returns str

get_tenant (***kwargs*)

Obtains the value of the ‘tenant’ field

Returns *infinisdk.infinibox.tenant.Tenant object*

get_type (***kwargs*)

Obtains the value of the ‘type’ field

Returns str

get_updated_at (***kwargs*)

Obtains the value of the ‘updated_at’ field

Returns Arrow

is_replicated (***kwargs*)

Obtains the value of the ‘replicated’ field

Returns bool

is_snapgroup ()

Checks if this is a snapshot group (as opposed to consistency group)

move_pool (*target_pool, with_capacity=False*)

Moves this entity to a new pool, optionally along with its needed capacity

refresh_snapgroup ()

Refresh a snapshot group with the most recent data from the parent consistency group

refresh_snapshot ()

Refresh a snapshot group with the most recent data from the parent consistency group

remove_member (*member, retain_staging_area=False, create_replica=False, replica_name=<OMIT>, force_if_no_remote_credentials=False, force_if_remote_error=False, force_on_target=False, force_on_local=<OMIT>, keep_serial_on_local=<OMIT>*)

Removes a volume member from this consistency group

restore (*snap_group*)

Restores this consistency group from the specified sg

update_lock_expires_at (*value, **kwargs*)

Updates the value of the ‘lock_expires_at’ field

Parameters *value* – The new lock_expires_at value to be set (type: Arrow)

update_name (*value, **kwargs*)

Updates the value of the ‘name’ field

Parameters *value* – The new name value to be set (type: str)

2.27.22 infinibox.components

class *infinisdk.infinibox.components.InfiniBoxSystemComponents* (*system*)

find (**predicates, **kw*)

Queries objects according to predicates. Can receive arguments in two possible forms:

1. Direct keyword arguments, for filtering for equality:

```
system.volumes.find(size=GiB)
```

2. Complex predicates, using the comparators:

```
system.volumes.find(system.volumes.fields.size > GiB)
system.volumes.find(Q.name != 'some_name')
```

Returns Lazy query result object.

See also:

infinisdk.core.object_query.ObjectQuery

class `infinisdk.infinibox.components.Nodes` (*object_type, system*)

class `infinisdk.infinibox.components.Node` (*system, initial_data*)

BINDER_CLASS

alias of *Nodes*

get_api_id (***kwargs*)

Obtains the value of the ‘api_id’ field

Returns int

get_core_service ()

Gets the core service running on this node

get_drives (***kwargs*)

Obtains the value of the ‘drives’ field

Returns list

get_eth_ports (***kwargs*)

Obtains the value of the ‘eth_ports’ field

Returns list

get_fc_ports (***kwargs*)

Obtains the value of the ‘fc_ports’ field

Returns list

get_ib_ports (***kwargs*)

Obtains the value of the ‘ib_ports’ field

Returns list

get_index (***kwargs*)

Obtains the value of the ‘index’ field

Returns int

get_management_service ()

Gets the management service running on this node

get_model (***kwargs*)

Obtains the value of the ‘model’ field

Returns str

get_name (***kwargs*)

Obtains the value of the ‘name’ field

Returns str

get_security (**kwargs)
Obtains the value of the ‘security’ field

Returns Munch

get_service (service_name)
Get a service object by its type name

Parameters **service_name** – the service name (mgmt/core/etc.)

get_services (**kwargs)
Obtains the value of the ‘services’ field

Returns list

get_state (**kwargs)
Obtains the value of the ‘state’ field

Returns str

get_tpm (**kwargs)
Obtains the value of the ‘tpm’ field

Returns Munch

class infinisdk.infinibox.components.**Enclosure** (system, initial_data)

get_api_id (**kwargs)
Obtains the value of the ‘api_id’ field

Returns int

get_drives (**kwargs)
Obtains the value of the ‘drives’ field

Returns list

get_index (**kwargs)
Obtains the value of the ‘index’ field

Returns int

get_state (**kwargs)
Obtains the value of the ‘state’ field

Returns str

class infinisdk.infinibox.components.**Drive** (system, initial_data)

get_capacity (**kwargs)
Obtains the value of the ‘capacity’ field

Returns

Capacity

get_enclosure (**kwargs)
Obtains the value of the ‘enclosure’ field

Returns int

get_enclosure_index (**kwargs)
Obtains the value of the ‘enclosure_index’ field

Returns int

get_index (**kwargs)
Obtains the value of the 'index' field

Returns int

get_serial_number (**kwargs)
Obtains the value of the 'serial_number' field

Returns str

get_state (**kwargs)
Obtains the value of the 'state' field

Returns str

is_encryption_state (**kwargs)
Obtains the value of the 'encryption_state' field

Returns bool

class infinisdk.infinibox.components.**FcPort** (system, initial_data)

BINDER_CLASS

alias of *FcPorts*

disable ()
Disables the FC Port

enable (role)
Enables the FC port with the specified role (SOFT_PORT/HARD_PORT)

get_api_id (**kwargs)
Obtains the value of the 'api_id' field

Returns int

get_index (**kwargs)
Obtains the value of the 'index' field

Returns int

get_link_state (**kwargs)
Obtains the value of the 'link_state' field

Returns str

get_node (**kwargs)
Obtains the value of the 'node' field

Returns int

get_role (**kwargs)
Obtains the value of the 'role' field

Returns str

get_soft_target_addresses (**kwargs)
Obtains the value of the 'soft_target_addresses' field

Returns list

get_state (**kwargs)
Obtains the value of the 'state' field

Returns str

get_switch_vendor (***kwargs*)
Obtains the value of the 'switch_vendor' field

Returns str

get_wwpn (***kwargs*)
Obtains the value of the 'wwpn' field

Returns WWN

is_enabled (***kwargs*)
Obtains the value of the 'enabled' field

Returns bool

class infinisdk.infinibox.components.**FcPorts** (*object_type, system*)

2.27.23 infinibox.qos_policies

class infinisdk.infinibox.qos_policy.**QosPolicyBinder** (*object_type, system*)

class infinisdk.infinibox.qos_policy.**QosPolicy** (*system, initial_data*)

BINDER_CLASS

alias of *QosPolicyBinder*

clear_metadata ()
Deletes all metadata keys for this object

classmethod construct (*system, data*)
Template method to enable customizing the object instantiation process.

This enables system components to be cached rather than re-fetched every time

classmethod create (*system, **fields*)
Creates a new object of this type

delete (***kwargs*)
Deletes this object.

get_all_metadata ()
Returns Dictionary of all keys and values associated as metadata for this object

get_assigned_entities ()
Returns the assigned entities of this QOS policy

get_burst_factor (***kwargs*)
Obtains the value of the 'burst_factor' field

Returns float

get_created_at (***kwargs*)
Obtains the value of the 'created_at' field

Returns Arrow

classmethod get_creation_defaults ()
Returns a dict representing the default arguments as implicitly constructed by infinisdk to fulfill a create call

Note: This will cause generation of defaults, which will have side effects if they are special values

Note: This does not necessarily generate all fields that are passable into `create`, only mandatory ‘fields

get_field (*field_name*, *from_cache=<DONT_CARE>*, *fetch_if_not_cached=True*,
raw_value=False)
Gets the value of a single field from the system

Parameters

- **cache** – Attempt to use the last cached version of the field value
- **fetch_if_not_cached** – Pass `False` to force only from cache

get_fields (*field_names=()*, *from_cache=<DONT_CARE>*, *fetch_if_not_cached=True*,
raw_value=False)
Gets a set of fields from the system

Parameters

- **from_cache** – Attempt to fetch the fields from the cache
- **fetch_if_not_cached** – pass as `False` to force only from cache

Returns a dictionary of field names to their values

get_id (***kwargs*)
Obtains the value of the ‘id’ field

Returns int

get_max_bps (***kwargs*)
Obtains the value of the ‘max_bps’ field

Returns int

get_max_ops (***kwargs*)
Obtains the value of the ‘max_ops’ field

Returns int

get_metadata_value (*key*, *default=<NOTHING>*)
Gets a metadata value, optionally specifying a default

Parameters default – if specified, the value to retrieve if the metadata key doesn’t exist. if not specified, and the key does not exist, the operation will raise an exception

get_name (***kwargs*)
Obtains the value of the ‘name’ field

Returns str

get_type (***kwargs*)
Obtains the value of the ‘type’ field

Returns str

get_updated_at (***kwargs*)
Obtains the value of the ‘updated_at’ field

Returns Arrow

invalidate_cache (**field_names*)

Discards the cached field values of this object, causing the next fetch to retrieve the fresh value from the system

is_burst_enabled (***kwargs*)

Obtains the value of the 'burst_enabled' field

Returns bool

is_in_system ()

Returns whether or not the object actually exists

safe_delete (**args, **kwargs*)

Tries to delete the object, doing nothing if the object cannot be found on the system

safe_get_field (*field_name, default=<NOTHING>, **kwargs*)

Like *get_field()*, only returns 'default' parameter if no result was found

set_metadata (*key, value*)

Sets metadata key in the system associated with this object

set_metadata_from_dict (*data_dict*)

Sets multiple metadata keys/values in the system associated with this object

unset_metadata (*key*)

Deletes a metadata key for this object

update_burst_factor (*value, **kwargs*)

Updates the value of the 'burst_factor' field

Parameters value – The new burst_factor value to be set (type: float)

update_field (*field_name, field_value*)

Updates the value of a single field

update_fields (***update_dict*)

Atomically updates a group of fields and respective values (given as a dictionary)

update_max_bps (*value, **kwargs*)

Updates the value of the 'max_bps' field

Parameters value – The new max_bps value to be set (type: int)

update_max_ops (*value, **kwargs*)

Updates the value of the 'max_ops' field

Parameters value – The new max_ops value to be set (type: int)

update_name (*value, **kwargs*)

Updates the value of the 'name' field

Parameters value – The new name value to be set (type: str)

update_type (*value, **kwargs*)

Updates the value of the 'type' field

Parameters value – The new type value to be set (type: str)

2.27.24 infinibox.tenants

class `infinisdk.infinibox.tenant.Tenant` (*system, initial_data*)

get_anonymous_gid (**kwargs)
 Obtains the value of the ‘anonymous_gid’ field
Returns int

get_anonymous_uid (**kwargs)
 Obtains the value of the ‘anonymous_uid’ field
Returns int

get_capacity (**kwargs)
 Obtains the value of the ‘capacity’ field
Returns Munch

get_created_at (**kwargs)
 Obtains the value of the ‘created_at’ field
Returns Arrow

get_entity_counts (**kwargs)
 Obtains the value of the ‘entity_counts’ field
Returns Munch

get_id (**kwargs)
 Obtains the value of the ‘id’ field
Returns int

get_name (**kwargs)
 Obtains the value of the ‘name’ field
Returns str

get_nfs_allow_unmapped_users (**kwargs)
 Obtains the value of the ‘nfs_allow_unmapped_users’ field
Returns str

get_nfs_group_policy (**kwargs)
 Obtains the value of the ‘nfs_group_policy’ field
Returns str

get_short_tenant_key (**kwargs)
 Obtains the value of the ‘short_tenant_key’ field
Returns int

get_updated_at (**kwargs)
 Obtains the value of the ‘updated_at’ field
Returns Arrow

is_visible_to_sysadmin (**kwargs)
 Obtains the value of the ‘visible_to_sysadmin’ field
Returns bool

update_anonymous_gid (value, **kwargs)
 Updates the value of the ‘anonymous_gid’ field
Parameters **value** – The new anonymous_gid value to be set (type: int)

update_anonymous_uid (value, **kwargs)
 Updates the value of the ‘anonymous_uid’ field

Parameters **value** – The new anonymous_uid value to be set (type: int)

update_name (*value*, ***kwargs*)

Updates the value of the 'name' field

Parameters **value** – The new name value to be set (type: str)

update_nfs_allow_unmapped_users (*value*, ***kwargs*)

Updates the value of the 'nfs_allow_unmapped_users' field

Parameters **value** – The new nfs_allow_unmapped_users value to be set (type: str)

update_nfs_group_policy (*value*, ***kwargs*)

Updates the value of the 'nfs_group_policy' field

Parameters **value** – The new nfs_group_policy value to be set (type: str)

2.27.25 infinibox.smb_groups

class `infinisdk.infinibox.smb_group.SMBGroup` (*system*, *initial_data*)

get_domain_members (***kwargs*)

Obtains the value of the 'domain_members' field

Returns list

get_gid (***kwargs*)

Obtains the value of the 'gid' field

Returns int

get_id (***kwargs*)

Obtains the value of the 'id' field

Returns int

get_name (***kwargs*)

Obtains the value of the 'name' field

Returns str

get_privileges (***kwargs*)

Obtains the value of the 'privileges' field

Returns list

get_sid (***kwargs*)

Obtains the value of the 'sid' field

Returns str

get_tenant (***kwargs*)

Obtains the value of the 'tenant' field

Returns `infinisdk.infinibox.tenant.Tenant` object

get_uid (***kwargs*)

Obtains the value of the 'uid' field

Returns int

update_domain_members (*value*, ***kwargs*)

Updates the value of the 'domain_members' field

Parameters value – The new domain_members value to be set (type: list)

update_gid (*value*, ***kwargs*)

Updates the value of the ‘gid’ field

Parameters value – The new gid value to be set (type: int)

update_privileges (*value*, ***kwargs*)

Updates the value of the ‘privileges’ field

Parameters value – The new privileges value to be set (type: list)

update_uid (*value*, ***kwargs*)

Updates the value of the ‘uid’ field

Parameters value – The new uid value to be set (type: int)

2.27.26 infinibox.smb_users

class `infinisdk.infinibox.smb_user.SMBUser` (*system*, *initial_data*)

disable (***kwargs*)

Set the value of the ‘enabled’ field to False

enable (***kwargs*)

Set the value of the ‘enabled’ field to True

get_groups (***kwargs*)

Obtains the value of the ‘groups’ field

Returns list

get_id (***kwargs*)

Obtains the value of the ‘id’ field

Returns int

get_name (***kwargs*)

Obtains the value of the ‘name’ field

Returns str

get_primary_group (***kwargs*)

Obtains the value of the ‘primary_group’ field

Returns `infinisdk.infinibox.smb_group.SMBGroup` object

get_privileges (***kwargs*)

Obtains the value of the ‘privileges’ field

Returns list

get_sid (***kwargs*)

Obtains the value of the ‘sid’ field

Returns str

get_tenant (***kwargs*)

Obtains the value of the ‘tenant’ field

Returns `infinisdk.infinibox.tenant.Tenant` object

get_uid (***kwargs*)

Obtains the value of the ‘uid’ field

Returns int

is_enabled (***kwargs*)

Obtains the value of the ‘enabled’ field

Returns bool

update_enabled

Updates the value of the ‘enabled’ field

param value The new enabled value to be set (type: bool)

update_groups (*value, **kwargs*)

Updates the value of the ‘groups’ field

Parameters value – The new groups value to be set (type: list)

update_name (*value, **kwargs*)

Updates the value of the ‘name’ field

Parameters value – The new name value to be set (type: str)

update_password (*value, **kwargs*)

Updates the value of the ‘password’ field

Parameters value – The new password value to be set (type: str)

update_primary_group (*value, **kwargs*)

Updates the value of the ‘primary_group’ field

Parameters value – The new primary_group value to be set (type: *infinisdk.infinibox.smb_group.SMBGroup object*)

update_privileges (*value, **kwargs*)

Updates the value of the ‘privileges’ field

Parameters value – The new privileges value to be set (type: list)

update_uid (*value, **kwargs*)

Updates the value of the ‘uid’ field

Parameters value – The new uid value to be set (type: int)

2.27.27 infinibox.active_directory_domains

class *infinisdk.infinibox.active_directory.ActiveDirectoryDomains* (*system*)

create (**, domain, org_unit=<OMIT>, preferred_ips, username, password, tenant=None*)

Join an active directory domain

Parameters

- **domain** (*str*) – the domain to join
- **org_unit** (*str*) – the organization unit
- **preferred_ips** (*list[str]*) – a list of ips
- **username** (*str*) – the username for the domain
- **password** (*str*) – the password for the domain
- **tenant** (*infinisdk.infinibox.tenant.Tenant*) – the tenant object

Returns Dictionary with fields: “tenant_id”, “domain”, “org_unit”, “preferred_ips”

Return type dict

get()

Obtains the active directory domain

Returns Dictionary with fields: “tenant_id”, “domain”, “org_unit”, “preferred_ips”

Return type dict

join(*, domain, org_unit=<OMIT>, preferred_ips, username, password, tenant=None)

Join an active directory domain

Parameters

- **domain** (*str*) – the domain to join
- **org_unit** (*str*) – the organization unit
- **preferred_ips** (*list[str]*) – a list of ips
- **username** (*str*) – the username for the domain
- **password** (*str*) – the password for the domain
- **tenant** (*infinisdk.infinibox.tenant.Tenant*) – the tenant object

Returns Dictionary with fields: “tenant_id”, “domain”, “org_unit”, “preferred_ips”

Return type dict

leave(*, username, password)

Leave the active directory domain

Parameters

- **username** (*str*) – the username for the domain
- **password** (*str*) – the password for the domain

2.27.28 Base Objects

class `infinisdk.infinibox.system_object.InfiniBoxObject` (*system, initial_data*)

2.27.29 Infinibox Utilities

class `infinisdk.infinibox.lun.LogicalUnit` (*system, id, lun, clustered, host_cluster_id, volume_id, host_id, **kwargs*)

__int__()

Same as `get_lun()`

delete()

Deletes (or unmaps) this LU

get_cluster()

Returns the cluster to which this LUN belongs

get_host()

Returns the host to which this LUN belongs

get_lun()

Returns the logical unit number of this LU

get_volume()
Returns the volume mapped to this LU

unmap()
Deletes (or unmaps) this LU

class infinisdk.infinibox.scsi_serial.**SCSISerial** (*serial*)

ieee_company_id = None
the IEEE company id (24 bits)

serial = None
the string representation (hexadecimal) of the serial number

system_id = None
unique system id (16 bits)

volume_id = None
the volume id (64 bits)

2.27.30 Core Facilities

class infinisdk.core.type_binder.**TypeBinder** (*object_type, system*)

create (*args, **kwargs)
Creates an object on the system

class infinisdk.core.type_binder.**MonomorphicBinder** (*object_type, system*)

find (*predicates, **kw)
Queries objects according to predicates. Can receive arguments in two possible forms:

1. Direct keyword arguments, for filtering for equality:

```
system.volumes.find(size=GiB)
```

2. Complex predicates, using the comparators:

```
system.volumes.find(system.volumes.fields.size > GiB)
system.volumes.find(Q.name != 'some_name')
```

Returns Lazy query result object.

See also:

infinisdk.core.object_query.ObjectQuery

get_by_id (*id*)
Obtains an object with a specific id

get_by_id_lazy (*id*)
Obtains an object with a specified id *without* checking if it exists or querying it on the way.

This is useful assuming the next operation is a further query/update on this object.

get_mutable_fields ()
Returns a list of all mutable fields for this object type

safe_get_by_id(*id*)

Like `get_by_id`, only returning `None` if the object could not be found

class `infinisdk.core.system_object.SystemObject`(*system, initial_data*)

System object, that has query methods, creation and deletion

BINDER_CLASS

alias of `infinisdk.core.type_binder.TypeBinder`

classmethod **construct**(*system, data*)

Template method to enable customizing the object instantiation process.

This enables system components to be cached rather than re-fetched every time

classmethod **create**(*system, **fields*)

Creates a new object of this type

delete(***kwargs*)

Deletes this object.

classmethod **get_creation_defaults**()

Returns a dict representing the default arguments as implicitly constructed by `infinisdk` to fulfill a `create` call

Note: This will cause generation of defaults, which will have side effects if they are special values

Note: This does not necessarily generate all fields that are passable into `create`, only mandatory 'fields'

get_field(*field_name, from_cache=<DONT_CARE>, fetch_if_not_cached=True, raw_value=False*)

Gets the value of a single field from the system

Parameters

- **cache** – Attempt to use the last cached version of the field value
- **fetch_if_not_cached** – Pass `False` to force only from cache

get_fields(*field_names=(), from_cache=<DONT_CARE>, fetch_if_not_cached=True, raw_value=False*)

Gets a set of fields from the system

Parameters

- **from_cache** – Attempt to fetch the fields from the cache
- **fetch_if_not_cached** – pass as `False` to force only from cache

Returns a dictionary of field names to their values

invalidate_cache(**field_names*)

Discards the cached field values of this object, causing the next fetch to retrieve the fresh value from the system

is_in_system()

Returns whether or not the object actually exists

safe_delete(**args, **kwargs*)

Tries to delete the object, doing nothing if the object cannot be found on the system

safe_get_field (*field_name*, *default=<NOTHING>*, ***kwargs*)

Like *get_field()*, only returns 'default' parameter if no result was found

update_field (*field_name*, *field_value*)

Updates the value of a single field

update_fields (***update_dict*)

Atomically updates a group of fields and respective values (given as a dictionary)

class *infinisdk.core.object_query.ObjectQuery* (*system*, *url*, *object_type*)

infinisdk.core.extensions.add_method (*objtype*, *name=None*)

2.27.31 Exceptions

class *infinisdk.core.exceptions.ObjectNotFound*

Thrown when using *.get()*, when no results are found but the code expects a single object

class *infinisdk.core.exceptions.TooManyObjectsFound*

Thrown when using *.get()*, when more than one result is found but the code expects a single object

2.28 Advanced Usage

2.28.1 Query Preprocessors

InfiniSDK allows modification of HTTP request just before they are sent to the system through a mechanism query preprocessors.

The system objects exposes a context manager called *api.query_preprocessor*. This context manager gets a function which can modify the request before it is sent.

```
def unapproved(request):
    request.url = request.url.set_query_param('approved', 'false')

with infinibox.api.query_preprocessor(unapproved):
    # Actions that require approval will be rejected within this context
```

2.28.2 Passing special values for fields

InfiniSDK supports a few special values for fields.

Among them, you can find Autogenerate, used to get autogenerated field values upon request, and RawValue, that will pass the values as-is.

2.29 Hooks

2.29.1 Overview

infinisdk uses *gossip* library, implementaion of a basic hook mechanism for registering callbacks.


```
>>> from __future__ import print_function
>>> import gossip

>>> @gossip.register('infinidat.sdk.post_object_creation', tags=['pool'],
↳ token=gossip_token)
... def post_creation(obj, **_):
...     print("Pool '{}' was created".format(obj.get_name()))

>>> pool = system.pools.create(name='some_pool')
Pool 'some_pool' was created
```

Note: It is entirely possible for hooks to receive more keyword arguments as features are added to InfiniSDK. To cope with this you are strongly encouraged to allow passing “catch-all” keyword arguments to your hooks (e.g. `**kwargs`)

See also:

For more information about gossip, see [Gossip documentation](#)

2.29.2 Available Hooks

The following hooks are available from the `infinidat.sdk` group:

`infinidat.sdk.after_api_request(request, response)`

No supported tags

`infinidat.sdk.after_login(system)`

No supported tags

`infinidat.sdk.before_api_request(request)`

No supported tags

`infinidat.sdk.begin_fork(obj)`

Supported tags: filesystem, infinibox, volume

`infinidat.sdk.cancel_fork(obj)`

Supported tags: filesystem, infinibox, volume

`infinidat.sdk.cluster_add_host_failure(host, cluster, exception)`

Supported tags: host_cluster, infinibox

infinidat.sdk.cluster_remove_host_failure(host, cluster, exception)

Supported tags: host_cluster, infinibox

infinidat.sdk.cons_group_add_member_failure(cons_group, member, request)

Supported tags: infinibox

infinidat.sdk.cons_group_deletion_failure(cons_group, delete_members)

Supported tags: cons_group, infinibox

infinidat.sdk.cons_group_remove_member_failure(cons_group, member)

Supported tags: infinibox

infinidat.sdk.data_restore_failure(source, target, exc)

Supported tags: filesystem, infinibox, volume

infinidat.sdk.entity_child_creation_failure(obj, exception, system)

Supported tags: cons_group, filesystem, infinibox, volume

infinidat.sdk.event_retention_failure(system, retention, exception)

Supported tags: event, infinibox

infinidat.sdk.finish_fork(obj, child)

Supported tags: filesystem, infinibox, volume

infinidat.sdk.object_creation_failure(data, system, cls, parent, exception)

Supported tags: cons_group, event, export, fc_soft_target, fc_switch, filesystem, host, host_cluster, infinibox, initiator, ldap_config, link, network_interface, network_space, nfs_user, nlm_lock, notification_rule, notification_target, plugin, pool, qos_policy, replica, replication_group, rg_replica, share, sharepermission, smb_group, smb_user, tenant, treeq, user, vm, volume, vvol

infinidat.sdk.object_deletion_failure(obj, exception, system, url)

Supported tags: cons_group, event, export, fc_soft_target, fc_switch, filesystem, host, host_cluster, infinibox, initiator, ldap_config, link, network_interface, network_space, nfs_user, nlm_lock, notification_rule, notification_target, plugin, pool, qos_policy, replica, replication_group, rg_replica, share, sharepermission, smb_group, smb_user, tenant, treeq, user, vm, volume, vvol

infinidat.sdk.object_operation_failure(exception)

Supported tags: bbu, cons_group, drive, enclosure, eth_port, event, export, external_cluster, fc_port, fc_soft_target, fc_switch, filesystem, host, host_cluster, ib_port, infinibox, initiator, ldap_config, link, local_drive, network_interface, network_space, nfs_user, nlm_lock, node, notification_rule, notification_target, pdu, plugin, pool, qos_policy, rack, replica, replication_group, rg_replica, service, service_cluster, share, sharepermission, smb_group, smb_user, system, tenant, treeq, user, vm, volume, vvol

infinidat.sdk.object_restore_failure(source, target, exc)

Supported tags: filesystem, infinibox, volume

infinidat.sdk.object_update_failure(obj, exception, system, data)

Supported tags: bbu, cons_group, drive, enclosure, eth_port, event, export, external_cluster, fc_port, fc_soft_target, fc_switch, filesystem, host, host_cluster, ib_port, infinibox, initiator, ldap_config, link, local_drive, network_interface, network_space, nfs_user, nlm_lock, node, notification_rule, notification_target, pdu, plugin, pool, qos_policy, rack, replica, replication_group, rg_replica, service, service_cluster, share, sharepermission, smb_group, smb_user, system, tenant, treeq, user, vm, volume, vvol

infinidat.sdk.pool_lock_failure(pool, exception)

Supported tags: infinibox, pool

infinidat.sdk.pool_move_failure(obj, with_capacity, system, exception, target_pool, source_pool)

Supported tags: cons_group, filesystem, infinibox, volume

infinidat.sdk.pool_unlock_failure(pool, exception)

Supported tags: infinibox, pool

infinidat.sdk.post_cluster_add_host(host, cluster)

Supported tags: host_cluster, infinibox

infinidat.sdk.post_cluster_remove_host(host, cluster)

Supported tags: host_cluster, infinibox

infinidat.sdk.post_cons_group_add_member(cons_group, member, request)

Supported tags: infinibox

infinidat.sdk.post_cons_group_deletion(cons_group, delete_members)

Supported tags: cons_group, infinibox

infinidat.sdk.post_cons_group_remove_member(cons_group, member)

Supported tags: infinibox

infinidat.sdk.post_data_restore(source, target, require_real_data, reason)

Supported tags: filesystem, infinibox, volume

infinidat.sdk.post_entity_child_creation(source, target, system)

Supported tags: cons_group, filesystem, infinibox, volume

infinidat.sdk.post_event_retention(system, retention)

Supported tags: event, infinibox

infinidat.sdk.post_object_creation(obj, data, response_dict, parent)

Supported tags: cons_group, event, export, fc_soft_target, fc_switch, filesystem, host, host_cluster, infinibox, initiator, ldap_config, link, network_interface, network_space, nfs_user, nlm_lock, notification_rule, notification_target, plugin, pool, qos_policy, replica, replication_group, rg_replica, share, sharepermission, smb_group, smb_user, tenant, treeq, user, vm, volume, vvol

infinidat.sdk.post_object_deletion(obj, url)

Supported tags: cons_group, event, export, fc_soft_target, fc_switch, filesystem, host, host_cluster, infinibox, initiator, ldap_config, link, network_interface, network_space, nfs_user, nlm_lock, notification_rule, notification_target, plugin, pool, qos_policy, replica, replication_group, rg_replica, share, sharepermission, smb_group, smb_user, tenant, treeq, user, vm, volume, vvol

infinidat.sdk.post_object_restore(source, target)

Supported tags: filesystem, infinibox, volume

infinidat.sdk.post_object_update(obj, data, response_dict)

Supported tags: bbu, cons_group, drive, enclosure, eth_port, event, export, external_cluster, fc_port, fc_soft_target, fc_switch, filesystem, host, host_cluster, ib_port, infinibox, initiator, ldap_config, link, local_drive, network_interface, network_space, nfs_user, nlm_lock, node, notification_rule, notification_target, pdu, plugin, pool, qos_policy, rack, replica, replication_group, rg_replica, service, service_cluster, share, sharepermission, smb_group, smb_user, system, tenant, treeq, user, vm, volume, vvol

infinidat.sdk.post_pool_lock(pool)

Supported tags: infinibox, pool

infinidat.sdk.post_pool_move(obj, with_capacity, system, target_pool, source_pool)

Supported tags: cons_group, filesystem, infinibox, volume

infinidat.sdk.post_pool_unlock(pool)

Supported tags: infinibox, pool

infinidat.sdk.post_qos_policy_assign(qos_policy, entity)

Supported tags: infinibox, qos_policy

infinidat.sdk.post_qos_policy_unassign(qos_policy, entity)

Supported tags: infinibox, qos_policy

infinidat.sdk.post_refresh_snapshot(source, target)

Supported tags: cons_group, filesystem, infinibox, volume

infinidat.sdk.post_replica_change_role(replica)

Supported tags: infinibox

infinidat.sdk.post_replica_change_type(replica, old_type, new_type)

Supported tags: infinibox

infinidat.sdk.post_replica_resume(replica)

Supported tags: infinibox

infinidat.sdk.post_replica_suspend(replica)

Supported tags: infinibox

infinidat.sdk.post_replica_switch_role(replica)

Supported tags: infinibox

infinidat.sdk.post_replication_group_remove_member(replication_group, member)

Supported tags: infinibox

infinidat.sdk.post_replication_snapshot_expose

Supported tags: cons_group, filesystem, volume

infinidat.sdk.post_rg_replica_resume(rg_replica)

Supported tags: infinibox

infinidat.sdk.post_rg_replica_suspend(rg_replica)

Supported tags: infinibox

infinidat.sdk.post_treeq_creation(fields, system, filesystem, treeq)

Supported tags: infinibox, treeq

infinidat.sdk.post_volume_mapping(volume, host_or_cluster, lun, lun_object)

Supported tags: host, host_cluster, infinibox

infinidat.sdk.post_volume_unmapping(volume, host_or_cluster, lun)

Supported tags: host, host_cluster, infinibox

infinidat.sdk.pre_cluster_add_host(host, cluster)

Supported tags: host_cluster, infinibox

infinidat.sdk.pre_cluster_remove_host(host, cluster)

Supported tags: host_cluster, infinibox

infinidat.sdk.pre_cons_group_add_member(cons_group, member, request)

Supported tags: infinibox

infinidat.sdk.pre_cons_group_deletion(cons_group, delete_members)

Supported tags: cons_group, infinibox

infinidat.sdk.pre_cons_group_remove_member(cons_group, member)

Supported tags: infinibox

infinidat.sdk.pre_creation_data_validation(fields, system, cls)

Supported tags: cons_group, event, export, fc_soft_target, fc_switch, filesystem, host, host_cluster, infinibox, initiator, ldap_config, link, network_interface, network_space, nfs_user, nlm_lock, notification_rule, notification_target, plugin, pool, qos_policy, replica, replication_group, rg_replica, share, sharepermission, smb_group, smb_user, tenant, treeq, user, vm, volume, vvol

infinidat.sdk.pre_data_restore(source, target)

Supported tags: filesystem, infinibox, volume

infinidat.sdk.pre_entity_child_creation(source, system)

Supported tags: cons_group, filesystem, infinibox, volume

infinidat.sdk.pre_event_retention(system, retention)

Supported tags: event, infinibox

infinidat.sdk.pre_fields_update(fields, source)

Supported tags: bbu, cons_group, drive, enclosure, eth_port, event, export, external_cluster, fc_port, fc_soft_target, fc_switch, filesystem, host, host_cluster, ib_port, infinibox, initiator, ldap_config, link, local_drive, network_interface, network_space, nfs_user, nlm_lock, node, notification_rule, notification_target, pdu, plugin, pool, qos_policy, rack, replica, replication_group, rg_replica, service, service_cluster, share, sharepermission, smb_group, smb_user, system, tenant, treeq, user, vm, volume, vvol

infinidat.sdk.pre_object_creation(data, system, cls, parent)

Supported tags: cons_group, event, export, fc_soft_target, fc_switch, filesystem, host, host_cluster, infinibox, initiator, ldap_config, link, network_interface, network_space, nfs_user, nlm_lock, notification_rule, notification_target, plugin, pool, qos_policy, replica, replication_group, rg_replica, share, sharepermission, smb_group, smb_user, tenant, treeq, user, vm, volume, vvol

infinidat.sdk.pre_object_deletion(obj, url)

Supported tags: cons_group, event, export, fc_soft_target, fc_switch, filesystem, host, host_cluster, infinibox, initiator, ldap_config, link, network_interface, network_space, nfs_user, nlm_lock, notification_rule, notification_target, plugin, pool, qos_policy, replica, replication_group, rg_replica, share, sharepermission, smb_group, smb_user, tenant, treeq, user, vm, volume, vvol

infinidat.sdk.pre_object_restore(source, target)

Supported tags: filesystem, infinibox, volume

infinidat.sdk.pre_object_update(obj, data)

Supported tags: bbu, cons_group, drive, enclosure, eth_port, event, export, external_cluster, fc_port, fc_soft_target, fc_switch, filesystem, host, host_cluster, ib_port, infinibox, initiator, ldap_config, link, local_drive, network_interface, network_space, nfs_user, nlm_lock, node, notification_rule, notification_target, pdu, plugin, pool, qos_policy, rack, replica, replication_group, rg_replica, service, service_cluster, share, sharepermission, smb_group, smb_user, system, tenant, treeq, user, vm, volume, vvol

infinidat.sdk.pre_pool_lock(pool)

Supported tags: infinibox, pool

infinidat.sdk.pre_pool_move(obj, with_capacity, system, target_pool, source_pool)

Supported tags: cons_group, filesystem, infinibox, volume

infinidat.sdk.pre_pool_unlock(pool)

Supported tags: infinibox, pool

infinidat.sdk.pre_qos_policy_assign(qos_policy, entity)

Supported tags: infinibox, qos_policy

infinidat.sdk.pre_qos_policy_unassign(qos_policy, entity)

Supported tags: infinibox, qos_policy

infinidat.sdk.pre_refresh_snapshot(source, target)

Supported tags: cons_group, filesystem, infinibox, volume

infinidat.sdk.pre_replica_change_role(replica)

Supported tags: infinibox

infinidat.sdk.pre_replica_change_type(replica, old_type, new_type)

Supported tags: infinibox

infinidat.sdk.pre_replica_resume(replica)

Supported tags: infinibox

infinidat.sdk.pre_replica_suspend(replica)

Supported tags: infinibox

infinidat.sdk.pre_replica_switch_role(replica)

Supported tags: infinibox

infinidat.sdk.pre_replication_group_remove_member(replication_group, member)

Supported tags: infinibox

infinidat.sdk.pre_replication_snapshot_expose

Supported tags: cons_group, filesystem, volume

infinidat.sdk.pre_rg_replica_resume(rg_replica)

Supported tags: infinibox

infinidat.sdk.pre_rg_replica_suspend(rg_replica)

Supported tags: infinibox

infinidat.sdk.pre_treeq_creation(fields, system, filesystem)

Supported tags: infinibox, treeq

infinidat.sdk.pre_volume_mapping(volume, host_or_cluster, lun)

Supported tags: host, host_cluster, infinibox

infinidat.sdk.pre_volume_unmapping(volume, host_or_cluster, lun)

Supported tags: host, host_cluster, infinibox

infinidat.sdk.qos_policy_assign_failure(qos_policy, entity, exception)

Supported tags: infinibox, qos_policy

infinidat.sdk.qos_policy_unassign_failure(qos_policy, entity, exception)

Supported tags: infinibox, qos_policy

infinidat.sdk.refresh_snapshot_failure(source, target)

Supported tags: cons_group, filesystem, infinibox, volume

infinidat.sdk.replica_change_role_failure(replica, exception)

Supported tags: infinibox

infinidat.sdk.replica_change_type_failure(replica, old_type, new_type, exception)

Supported tags: infinibox

infinidat.sdk.replica_deleted(replica, entity_pairs, deletion_params)

Supported tags: infinibox

infinidat.sdk.replica_resume_failure(replica, exception)

Supported tags: infinibox

infinidat.sdk.replica_snapshot_created(snapshot, replica_deleted, replica_exposed)

Supported tags: infinibox

infinidat.sdk.replica_suspend_failure(replica, exception)

Supported tags: infinibox

infinidat.sdk.replica_switch_role_failure(replica, exception)

Supported tags: infinibox

infinidat.sdk.replication_group_remove_member_failure(replication_group, member)

Supported tags: infinibox

infinidat.sdk.replication_snapshot_expose_failure

Supported tags: cons_group, filesystem, volume

infinidat.sdk.rg_replica_resume_failure(rg_replica, exception)

Supported tags: infinibox

infinidat.sdk.rg_replica_suspend_failure(rg_replica, exception)

Supported tags: infinibox

infinidat.sdk.treeq_creation_failure(fields, system, filesystem, exception)

Supported tags: infinibox, treeq

infinidat.sdk.volume_mapping_failure(volume, host_or_cluster, exception, lun)

Supported tags: host, host_cluster, infinibox

infinidat.sdk.volume_unmapping_failure(volume, host_or_cluster, exception, lun)

Supported tags: host, host_cluster, infinibox

infinidat.sdk.witness_address_set(witness_address)

Supported tags: infinibox

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`

i

- `infinisdk.core.api.api`, 43
- `infinisdk.core.events`, 93
- `infinisdk.core.exceptions`, 116
- `infinisdk.core.system_object`, 115
- `infinisdk.core.type_binder`, 114
- `infinisdk.infinibox`, 42
- `infinisdk.infinibox.active_directory`,
112
- `infinisdk.infinibox.components`, 102
- `infinisdk.infinibox.cons_group`, 100
- `infinisdk.infinibox.dataset`, 44
- `infinisdk.infinibox.events`, 94
- `infinisdk.infinibox.export`, 62
- `infinisdk.infinibox.filesystem`, 55
- `infinisdk.infinibox.host`, 75
- `infinisdk.infinibox.host_cluster`, 81
- `infinisdk.infinibox.ldap_config`, 97
- `infinisdk.infinibox.link`, 90
- `infinisdk.infinibox.lun`, 113
- `infinisdk.infinibox.notification_target`,
97
- `infinisdk.infinibox.pool`, 70
- `infinisdk.infinibox.qos_policy`, 106
- `infinisdk.infinibox.replica`, 81
- `infinisdk.infinibox.replication_group`,
68
- `infinisdk.infinibox.rg_replica`, 68
- `infinisdk.infinibox.scsi_serial`, 114
- `infinisdk.infinibox.share`, 65
- `infinisdk.infinibox.share_permission`,
67
- `infinisdk.infinibox.smb_group`, 110
- `infinisdk.infinibox.smb_user`, 111
- `infinisdk.infinibox.system_object`, 113
- `infinisdk.infinibox.tenant`, 108
- `infinisdk.infinibox.volume`, 49

Symbols

`__int__()` (*infinisdk.infinibox.lun.LogicalUnit method*), 113

A

`ActiveDirectoryDomains` (class in *infinisdk.infinibox.active_directory*), 112

`add_member()` (*infinisdk.infinibox.cons_group.ConsGroup method*), 100

`add_method()` (in module *infinisdk.core.extensions*), 116

`add_port()` (*infinisdk.infinibox.host.Host method*), 77

API (class in *infinisdk.core.api.api*), 43

B

`BINDER_CLASS` (*infinisdk.core.events.Event attribute*), 93

`BINDER_CLASS` (*infinisdk.core.system_object.SystemObject attribute*), 115

`BINDER_CLASS` (*infinisdk.infinibox.components.FcPort attribute*), 105

`BINDER_CLASS` (*infinisdk.infinibox.components.Node attribute*), 103

`BINDER_CLASS` (*infinisdk.infinibox.filesystem.Filesystem attribute*), 55

`BINDER_CLASS` (*infinisdk.infinibox.host.Host attribute*), 77

`BINDER_CLASS` (*infinisdk.infinibox.ldap_config.LDAPConfig attribute*), 97

`BINDER_CLASS` (*infinisdk.infinibox.link.Link attribute*), 90

`BINDER_CLASS` (*infinisdk.infinibox.pool.Pool attribute*), 70

`BINDER_CLASS` (*infinisdk.infinibox.qos_policy.QosPolicy attribute*), 106

`BINDER_CLASS` (*infinisdk.infinibox.replica.Replica attribute*), 82

`BINDER_CLASS` (*infinisdk.infinibox.user.User attribute*), 94

`BINDER_CLASS` (*infinisdk.infinibox.volume.Volume attribute*), 49

C

`calculate_reclaimable_space()` (*infinisdk.infinibox.dataset.Dataset method*), 44

`calculate_reclaimable_space()` (*infinisdk.infinibox.filesystem.Filesystem method*), 55

`calculate_reclaimable_space()` (*infinisdk.infinibox.volume.Volume method*), 49

`change_role()` (*infinisdk.infinibox.replica.Replica method*), 83

`change_type_to_async()` (*infinisdk.infinibox.replica.Replica method*), 83

`change_type_to_sync()` (*infinisdk.infinibox.replica.Replica method*), 83

`check_version()` (*infinisdk.infinibox.InfiniBox method*), 42

`choose()` (*infinisdk.infinibox.host.HostBinder method*), 75

`clear_metadata()` (*infinisdk.infinibox.filesystem.Filesystem method*), 55

`clear_metadata()` (*infinisdk.infinibox.host.Host method*), 77

`clear_metadata()` (*infinisdk.infinibox.pool.Pool method*), 70

`clear_metadata()` (*infinisdk.infinibox.qos_policy.QosPolicy method*), 106

`clear_metadata()` (*infinisdk.infinibox.volume.Volume method*), 49

`clone_requests_session()` (*infinisdk.core.api.api.API method*), 43

ConsGroup (class in infinisdk.infinibox.cons_group),
 100
 construct () (infinisdk.core.system_object.SystemObject.create_snapshot ()
 class method), 115
 construct () (infinisdk.infinibox.filesystem.Filesystem
 class method), 55
 construct () (infinisdk.infinibox.host.Host class
 method), 77
 construct () (infinisdk.infinibox.pool.Pool class
 method), 70
 construct () (infinisdk.infinibox.qos_policy.QosPolicy
 class method), 106
 construct () (infinisdk.infinibox.user.User class
 method), 94
 construct () (infinisdk.infinibox.volume.Volume class
 method), 49
 create () (infinisdk.core.system_object.SystemObject
 class method), 115
 create () (infinisdk.core.type_binder.TypeBinder
 method), 114
 create () (infinisdk.infinibox.active_directory.ActiveDirectoryDomain
 method), 112
 create () (infinisdk.infinibox.filesystem.Filesystem
 class method), 55
 create () (infinisdk.infinibox.host.Host class method),
 77
 create () (infinisdk.infinibox.host.HostBinder method),
 76
 create () (infinisdk.infinibox.pool.Pool class method),
 71
 create () (infinisdk.infinibox.qos_policy.QosPolicy
 class method), 106
 create () (infinisdk.infinibox.user.User class method),
 94
 create () (infinisdk.infinibox.volume.Volume class
 method), 49
 create_group () (infinisdk.infinibox.ldap_config.LDAPConfig
 method), 97
 create_group_snapshot () (infinisdk.infinibox.volume.VolumesBinder
 method), 49
 create_many () (infinisdk.infinibox.dataset.DatasetTypeBinder
 method), 44
 create_snapgroup () (infinisdk.infinibox.cons_group.ConsGroup
 method), 100
 create_snapshot () (infinisdk.infinibox.cons_group.ConsGroup
 method), 100
 create_snapshot () (infinisdk.infinibox.dataset.Dataset
 method), 44
 create_snapshot () (infinisdk.infinibox.filesystem.Filesystem
 method), 55

D

Dataset (class in infinisdk.infinibox.dataset), 44
 DatasetTypeBinder (class in infinisdk.infinibox.dataset), 44
 define () (infinisdk.infinibox.ldap_config.LDAPConfigBinder
 method), 97
 delete () (infinisdk.core.api.api.API method), 43
 delete () (infinisdk.core.system_object.SystemObject
 method), 115
 delete () (infinisdk.infinibox.cons_group.ConsGroup
 method), 100
 delete () (infinisdk.infinibox.dataset.Dataset method),
 44
 delete () (infinisdk.infinibox.filesystem.Filesystem
 method), 55
 delete () (infinisdk.infinibox.host.Host method), 77
 delete () (infinisdk.infinibox.link.Link method), 90
 delete () (infinisdk.infinibox.lun.LogicalUnit method),
 113
 delete () (infinisdk.infinibox.pool.Pool method), 71
 delete () (infinisdk.infinibox.qos_policy.QosPolicy
 method), 106
 delete () (infinisdk.infinibox.replica.Replica method),
 83
 delete () (infinisdk.infinibox.user.User method), 94
 delete () (infinisdk.infinibox.volume.Volume method),
 49
 disable () (infinisdk.infinibox.components.FcPort
 method), 105
 disable () (infinisdk.infinibox.export.Export method),
 62
 disable () (infinisdk.infinibox.share.Share method), 65
 disable () (infinisdk.infinibox.smb_user.SMBUser
 method), 111
 disable () (infinisdk.infinibox.user.User method), 94
 disable_32bit_file_id () (infinisdk.infinibox.export.Export
 method), 62
 disable_access_based_enumeration () (infinisdk.infinibox.share.Share
 method), 65
 disable_automatic_ip_failback () (infinisdk.infinibox.network_space.NetworkSpace
 method), 91
 disable_compression () (infinisdk.infinibox.dataset.Dataset
 method), 45
 disable_compression () (infinisdk.infinibox.filesystem.Filesystem
 method), 56
 disable_compression () (infinisdk.infinibox.pool.Pool
 method), 71

- [disable_compression\(\)](#) (*infinisdk.infinibox.volume.Volume* method), 49
[disable_make_all_users_anonymous\(\)](#) (*infinisdk.infinibox.export.Export* method), 62
[disable_optimized\(\)](#) (*infinisdk.infinibox.host.Host* method), 77
[disable_preferred\(\)](#) (*infinisdk.infinibox.replica.Replica* method), 83
[disable_privileged_port\(\)](#) (*infinisdk.infinibox.export.Export* method), 62
[disable_require_encryption\(\)](#) (*infinisdk.infinibox.share.Share* method), 65
[disable_snapdir_visible\(\)](#) (*infinisdk.infinibox.export.Export* method), 62
[disable_snapdir_visible\(\)](#) (*infinisdk.infinibox.share.Share* method), 65
[disable_ssd\(\)](#) (*infinisdk.infinibox.dataset.Dataset* method), 45
[disable_ssd\(\)](#) (*infinisdk.infinibox.filesystem.Filesystem* method), 56
[disable_ssd\(\)](#) (*infinisdk.infinibox.pool.Pool* method), 71
[disable_ssd\(\)](#) (*infinisdk.infinibox.volume.Volume* method), 49
[disable_tls\(\)](#) (*infinisdk.infinibox.notification_target.NotificationTarget* method), 97
[disable_write_protection\(\)](#) (*infinisdk.infinibox.dataset.Dataset* method), 45
[disable_write_protection\(\)](#) (*infinisdk.infinibox.filesystem.Filesystem* method), 56
[disable_write_protection\(\)](#) (*infinisdk.infinibox.volume.Volume* method), 49
[disabled_login_refresh_context\(\)](#) (*infinisdk.core.api.api.API* method), 43
[Drive](#) (class in *infinisdk.infinibox.components*), 104
- ## E
- [enable\(\)](#) (*infinisdk.infinibox.components.FcPort* method), 105
[enable\(\)](#) (*infinisdk.infinibox.export.Export* method), 62
[enable\(\)](#) (*infinisdk.infinibox.share.Share* method), 65
[enable\(\)](#) (*infinisdk.infinibox.smb_user.SMBUser* method), 111
[enable\(\)](#) (*infinisdk.infinibox.user.User* method), 94
[enable_32bit_file_id\(\)](#) (*infinisdk.infinibox.export.Export* method), 62
[enable_access_based_enumeration\(\)](#) (*infinisdk.infinibox.share.Share* method), 65
[enable_automatic_ip_failback\(\)](#) (*infinisdk.infinibox.network_space.NetworkSpace* method), 92
[enable_compression\(\)](#) (*infinisdk.infinibox.dataset.Dataset* method), 45
[enable_compression\(\)](#) (*infinisdk.infinibox.filesystem.Filesystem* method), 56
[enable_compression\(\)](#) (*infinisdk.infinibox.pool.Pool* method), 71
[enable_compression\(\)](#) (*infinisdk.infinibox.volume.Volume* method), 49
[enable_make_all_users_anonymous\(\)](#) (*infinisdk.infinibox.export.Export* method), 62
[enable_optimized\(\)](#) (*infinisdk.infinibox.host.Host* method), 77
[enable_preferred\(\)](#) (*infinisdk.infinibox.replica.Replica* method), 83
[enable_privileged_port\(\)](#) (*infinisdk.infinibox.export.Export* method), 62
[enable_require_encryption\(\)](#) (*infinisdk.infinibox.share.Share* method), 65
[enable_snapdir_visible\(\)](#) (*infinisdk.infinibox.export.Export* method), 62
[enable_snapdir_visible\(\)](#) (*infinisdk.infinibox.share.Share* method), 65
[enable_ssd\(\)](#) (*infinisdk.infinibox.dataset.Dataset* method), 45
[enable_ssd\(\)](#) (*infinisdk.infinibox.filesystem.Filesystem* method), 56
[enable_ssd\(\)](#) (*infinisdk.infinibox.pool.Pool* method), 71
[enable_ssd\(\)](#) (*infinisdk.infinibox.volume.Volume* method), 49
[enable_tls\(\)](#) (*infinisdk.infinibox.notification_target.NotificationTarget* method), 97
[enable_write_protection\(\)](#) (*infinisdk.infinibox.dataset.Dataset* method), 45
[enable_write_protection\(\)](#) (*infinisdk.infinibox.filesystem.Filesystem* method), 56
[enable_write_protection\(\)](#) (*infinisdk.infinibox.volume.Volume* method), 49
[Enclosure](#) (class in *infinisdk.infinibox.components*), 104
[Event](#) (class in *infinisdk.core.events*), 93
[Events](#) (class in *infinisdk.core.events*), 93
[Events](#) (class in *infinisdk.infinibox.events*), 94
[Export](#) (class in *infinisdk.infinibox.export*), 62
- ## F
- [FcPort](#) (class in *infinisdk.infinibox.components*), 105
[FcPorts](#) (class in *infinisdk.infinibox.components*), 106
[Filesystem](#) (class in *infinisdk.infinibox.filesystem*), 55

[FilesystemBinder](#) (class in [infinisdk.infinibox.filesystem](#)), 55
[find\(\)](#) ([infinisdk.core.type_binder.MonomorphicBinder](#) method), 114
[find\(\)](#) ([infinisdk.infinibox.components.InfiniBoxSystemComponents](#) method), 102
[find\(\)](#) ([infinisdk.infinibox.host.HostBinder](#) method), 76

G

[get\(\)](#) ([infinisdk.core.api.api.API](#) method), 43
[get\(\)](#) ([infinisdk.infinibox.active_directory.ActiveDirectoryDomains](#) method), 113
[get\(\)](#) ([infinisdk.infinibox.host.HostBinder](#) method), 76
[get_access\(\)](#) ([infinisdk.infinibox.share_permission.SharePermission](#) method), 67
[get_administered_pools\(\)](#) ([infinisdk.infinibox.pool.PoolBinder](#) method), 70
[get_affected_entity_id\(\)](#) ([infinisdk.core.events.Event](#) method), 93
[get_all_metadata\(\)](#) ([infinisdk.infinibox.filesystem.Filesystem](#) method), 56
[get_all_metadata\(\)](#) ([infinisdk.infinibox.host.Host](#) method), 77
[get_all_metadata\(\)](#) ([infinisdk.infinibox.pool.Pool](#) method), 71
[get_all_metadata\(\)](#) ([infinisdk.infinibox.qos_policy.QosPolicy](#) method), 106
[get_all_metadata\(\)](#) ([infinisdk.infinibox.volume.Volume](#) method), 49
[get_allocated\(\)](#) ([infinisdk.infinibox.dataset.Dataset](#) method), 45
[get_allocated\(\)](#) ([infinisdk.infinibox.filesystem.Filesystem](#) method), 56
[get_allocated\(\)](#) ([infinisdk.infinibox.volume.Volume](#) method), 49
[get_allocated_physical_capacity\(\)](#) ([infinisdk.infinibox.pool.Pool](#) method), 71
[get_anonymous_gid\(\)](#) ([infinisdk.infinibox.export.Export](#) method), 62
[get_anonymous_gid\(\)](#) ([infinisdk.infinibox.tenant.Tenant](#) method), 108
[get_anonymous_uid\(\)](#) ([infinisdk.infinibox.export.Export](#) method), 62
[get_anonymous_uid\(\)](#) ([infinisdk.infinibox.tenant.Tenant](#) method), 109
[get_api_id\(\)](#) ([infinisdk.infinibox.components.Enclosure](#) method), 104
[get_api_id\(\)](#) ([infinisdk.infinibox.components.FcPort](#) method), 105
[get_api_id\(\)](#) ([infinisdk.infinibox.components.Node](#) method), 103
[get_approval_context\(\)](#) ([infinisdk.core.api.api.API](#) method), 43
[get_approved_context\(\)](#) ([infinisdk.core.api.api.API](#) method), 43
[get_assigned_entities\(\)](#) ([infinisdk.infinibox.qos_policy.QosPolicy](#) method), 106
[get_assigned_sync_remote_ips\(\)](#) ([infinisdk.infinibox.replica.Replica](#) method), 83
[get_atime_granularity\(\)](#) ([infinisdk.infinibox.filesystem.Filesystem](#) method), 56
[get_atime_mode\(\)](#) ([infinisdk.infinibox.filesystem.Filesystem](#) method), 56
[get_auth\(\)](#) ([infinisdk.core.api.api.API](#) method), 43
[get_auth_context\(\)](#) ([infinisdk.core.api.api.API](#) method), 43
[get_auth_protocol\(\)](#) ([infinisdk.infinibox.notification_target.NotificationTarget](#) method), 97
[get_auth_type\(\)](#) ([infinisdk.infinibox.notification_target.NotificationTarget](#) method), 97
[get_base_action\(\)](#) ([infinisdk.infinibox.replica.Replica](#) method), 83
[get_base_action\(\)](#) ([infinisdk.infinibox.rg_replica.RgReplica](#) method), 68
[get_burst_factor\(\)](#) ([infinisdk.infinibox.qos_policy.QosPolicy](#) method), 106
[get_by_id\(\)](#) ([infinisdk.core.type_binder.MonomorphicBinder](#) method), 114
[get_by_id\(\)](#) ([infinisdk.infinibox.host.HostBinder](#) method), 76
[get_by_id_lazy\(\)](#) ([infinisdk.core.type_binder.MonomorphicBinder](#) method), 114
[get_by_id_lazy\(\)](#) ([infinisdk.infinibox.host.HostBinder](#) method), 76
[get_capacity\(\)](#) ([infinisdk.infinibox.components.Drive](#) method), 104
[get_capacity\(\)](#) ([infinisdk.infinibox.tenant.Tenant](#) method), 109
[get_capacity_savings\(\)](#) ([infinisdk.infinibox.dataset.Dataset](#) method), 45
[get_capacity_savings\(\)](#) ([infinisdk.infinibox.dataset.Dataset](#) method), 45

`isdk.infinibox.filesystem.Filesystem` method), 56
`get_capacity_savings()` (`infinisdk.infinibox.pool.Pool` method), 71
`get_capacity_savings()` (`infinisdk.infinibox.volume.Volume` method), 50
`get_children()` (`infinisdk.infinibox.dataset.Dataset` method), 45
`get_children()` (`infinisdk.infinibox.filesystem.Filesystem` method), 56
`get_children()` (`infinisdk.infinibox.volume.Volume` method), 50
`get_cluster()` (`infinisdk.infinibox.host.Host` method), 77
`get_cluster()` (`infinisdk.infinibox.lun.LogicalUnit` method), 113
`get_code()` (`infinisdk.core.events.Event` method), 93
`get_community()` (`infinisdk.infinibox.notification_target.NotificationTarget` method), 98
`get_cons_group()` (`infinisdk.infinibox.volume.Volume` method), 50
`get_core_service()` (`infinisdk.infinibox.components.Node` method), 103
`get_created_at()` (`infinisdk.infinibox.cons_group.ConsGroup` method), 100
`get_created_at()` (`infinisdk.infinibox.dataset.Dataset` method), 45
`get_created_at()` (`infinisdk.infinibox.export.Export` method), 62
`get_created_at()` (`infinisdk.infinibox.filesystem.Filesystem` method), 56
`get_created_at()` (`infinisdk.infinibox.host.Host` method), 77
`get_created_at()` (`infinisdk.infinibox.host_cluster.HostCluster` method), 81
`get_created_at()` (`infinisdk.infinibox.pool.Pool` method), 71
`get_created_at()` (`infinisdk.infinibox.qos_policy.QosPolicy` method), 106
`get_created_at()` (`infinisdk.infinibox.replica.Replica` method), 83
`get_created_at()` (`infinisdk.infinibox.replication_group.ReplicationGroup` method), 68
`get_created_at()` (`infinisdk.infinibox.rg_replica.RgReplica` method), 68
`get_created_at()` (`infinisdk.infinibox.share.Share` method), 65
`get_created_at()` (`infinisdk.infinibox.tenant.Tenant` method), 109
`get_created_at()` (`infinisdk.infinibox.volume.Volume` method), 50
`get_creation_defaults()` (`infinisdk.core.system_object.SystemObject` class method), 115
`get_creation_defaults()` (`infinisdk.infinibox.filesystem.Filesystem` class method), 56
`get_creation_defaults()` (`infinisdk.infinibox.host.Host` class method), 77
`get_creation_defaults()` (`infinisdk.infinibox.pool.Pool` class method), 71
`get_creation_defaults()` (`infinisdk.infinibox.qos_policy.QosPolicy` class method), 106
`get_creation_defaults()` (`infinisdk.infinibox.user.User` class method), 95
`get_creation_defaults()` (`infinisdk.infinibox.volume.Volume` class method), 50
`get_creation_time()` (`infinisdk.infinibox.dataset.Dataset` method), 45
`get_creation_time()` (`infinisdk.infinibox.filesystem.Filesystem` method), 57
`get_creation_time()` (`infinisdk.infinibox.volume.Volume` method), 50
`get_data_snapshot_guid()` (`infinisdk.infinibox.filesystem.Filesystem` method), 57
`get_data_snapshot_guid()` (`infinisdk.infinibox.volume.Volume` method), 50
`get_dataset_type()` (`infinisdk.infinibox.dataset.Dataset` method), 45
`get_dataset_type()` (`infinisdk.infinibox.filesystem.Filesystem` method), 57
`get_dataset_type()` (`infinisdk.infinibox.volume.Volume` method), 50
`get_default_file_unix_permissions()` (`infinisdk.infinibox.share.Share` method), 65
`get_default_folder_unix_permissions()` (`infinisdk.infinibox.share.Share` method), 65
`get_depth()` (`infinisdk.infinibox.dataset.Dataset` method), 45
`get_depth()` (`infinisdk.infinibox.filesystem.Filesystem` method), 57
`get_depth()` (`infinisdk.infinibox.volume.Volume` method), 50
`get_description()` (`infinisdk.core.events.Event`

`method`), 93
`get_description()` (*infinisdk.infinibox.replica.Replica method*), 83
`get_description()` (*infinisdk.infinibox.share.Share method*), 65
`get_description_template()` (*infinisdk.core.events.Event method*), 93
`get_domain_members()` (*infinisdk.infinibox.smb_group.SMBGroup method*), 110
`get_drives()` (*infinisdk.infinibox.components.Enclosure method*), 104
`get_drives()` (*infinisdk.infinibox.components.Node method*), 103
`get_email()` (*infinisdk.infinibox.user.User method*), 95
`get_enclosure()` (*infinisdk.infinibox.components.Drive method*), 104
`get_enclosure_index()` (*infinisdk.infinibox.components.Drive method*), 104
`get_engine()` (*infinisdk.infinibox.notification_target.NotificationTarget method*), 98
`get_entities_count()` (*infinisdk.infinibox.pool.Pool method*), 71
`get_entity_counts()` (*infinisdk.infinibox.tenant.Tenant method*), 109
`get_entity_pairs()` (*infinisdk.infinibox.replica.Replica method*), 83
`get_entity_type()` (*infinisdk.infinibox.replica.Replica method*), 83
`get_error()` (*infinisdk.core.api.api.Response method*), 44
`get_eth_ports()` (*infinisdk.infinibox.components.Node method*), 103
`get_export_path()` (*infinisdk.infinibox.export.Export method*), 62
`get_facility()` (*infinisdk.infinibox.notification_target.NotificationTarget method*), 98
`get_family_id()` (*infinisdk.infinibox.dataset.Dataset method*), 45
`get_family_id()` (*infinisdk.infinibox.filesystem.Filesystem method*), 57
`get_family_id()` (*infinisdk.infinibox.volume.Volume method*), 50
`get_fc_ports()` (*infinisdk.infinibox.components.Node method*), 103
`get_field()` (*infinisdk.core.system_object.SystemObject method*), 115
`get_field()` (*infinisdk.infinibox.filesystem.Filesystem method*), 57
`get_field()` (*infinisdk.infinibox.host.Host method*), 78
`get_field()` (*infinisdk.infinibox.pool.Pool method*), 71
`get_field()` (*infinisdk.infinibox.qos_policy.QosPolicy method*), 107
`get_field()` (*infinisdk.infinibox.user.User method*), 95
`get_field()` (*infinisdk.infinibox.volume.Volume method*), 50
`get_fields()` (*infinisdk.core.system_object.SystemObject method*), 115
`get_fields()` (*infinisdk.infinibox.filesystem.Filesystem method*), 57
`get_fields()` (*infinisdk.infinibox.host.Host method*), 78
`get_fields()` (*infinisdk.infinibox.pool.Pool method*), 72
`get_fields()` (*infinisdk.infinibox.qos_policy.QosPolicy method*), 107
`get_fields()` (*infinisdk.infinibox.user.User method*), 95
`get_fields()` (*infinisdk.infinibox.volume.Volume method*), 51
`get_filesystem()` (*infinisdk.infinibox.export.Export method*), 62
`get_filesystem()` (*infinisdk.infinibox.share.Share method*), 66
`get_filesystem_snapshots_count()` (*infinisdk.infinibox.pool.Pool method*), 72
`get_filesystems_count()` (*infinisdk.infinibox.pool.Pool method*), 72
`get_free_physical_capacity()` (*infinisdk.infinibox.pool.Pool method*), 72
`get_free_virtual_capacity()` (*infinisdk.infinibox.pool.Pool method*), 72
`get_from_address()` (*infinisdk.infinibox.notification_target.NotificationTarget method*), 98
`get_gid()` (*infinisdk.infinibox.smb_group.SMBGroup method*), 110
`get_groups()` (*infinisdk.infinibox.smb_user.SMBUser method*), 111
`get_host()` (*infinisdk.infinibox.lun.LogicalUnit method*), 113
`get_host()` (*infinisdk.infinibox.notification_target.NotificationTarget method*), 98

`get_host_by_initiator_address()` (*infinisdk.infinibox.host.HostBinder method*), 76
`get_host_id_by_initiator_address()` (*infinisdk.infinibox.host.HostBinder method*), 76
`get_host_type()` (*infinisdk.infinibox.host.Host method*), 78
`get_host_type()` (*infinisdk.infinibox.host_cluster.HostCluster method*), 81
`get_hosts()` (*infinisdk.infinibox.host_cluster.HostCluster method*), 81
`get_ib_ports()` (*infinisdk.infinibox.components.Node method*), 103
`get_id()` (*infinisdk.core.events.Event method*), 93
`get_id()` (*infinisdk.infinibox.cons_group.ConsGroup method*), 101
`get_id()` (*infinisdk.infinibox.dataset.Dataset method*), 45
`get_id()` (*infinisdk.infinibox.export.Export method*), 62
`get_id()` (*infinisdk.infinibox.filesystem.Filesystem method*), 57
`get_id()` (*infinisdk.infinibox.host.Host method*), 78
`get_id()` (*infinisdk.infinibox.host_cluster.HostCluster method*), 81
`get_id()` (*infinisdk.infinibox.ldap_config.LDAPConfig method*), 97
`get_id()` (*infinisdk.infinibox.link.Link method*), 90
`get_id()` (*infinisdk.infinibox.network_space.NetworkSpace method*), 92
`get_id()` (*infinisdk.infinibox.notification_target.NotificationTarget method*), 84
`get_id()` (*infinisdk.infinibox.pool.Pool method*), 72
`get_id()` (*infinisdk.infinibox.qos_policy.QosPolicy method*), 107
`get_id()` (*infinisdk.infinibox.replica.Replica method*), 83
`get_id()` (*infinisdk.infinibox.replication_group.ReplicationGroup method*), 68
`get_id()` (*infinisdk.infinibox.rg_replica.RgReplica method*), 68
`get_id()` (*infinisdk.infinibox.share.Share method*), 66
`get_id()` (*infinisdk.infinibox.share_permission.SharePermission method*), 67
`get_id()` (*infinisdk.infinibox.smb_group.SMBGroup method*), 110
`get_id()` (*infinisdk.infinibox.smb_user.SMBUser method*), 111
`get_id()` (*infinisdk.infinibox.tenant.Tenant method*), 109
`get_id()` (*infinisdk.infinibox.user.User method*), 95
`get_id()` (*infinisdk.infinibox.volume.Volume method*), 51
`get_index()` (*infinisdk.infinibox.components.Drive method*), 105
`get_index()` (*infinisdk.infinibox.components.Enclosure method*), 104
`get_index()` (*infinisdk.infinibox.components.FcPort method*), 105
`get_index()` (*infinisdk.infinibox.components.Node method*), 103
`get_inner_path()` (*infinisdk.infinibox.export.Export method*), 62
`get_inner_path()` (*infinisdk.infinibox.share.Share method*), 66
`get_interfaces()` (*infinisdk.infinibox.network_space.NetworkSpace method*), 92
`get_ips()` (*infinisdk.infinibox.network_space.NetworkSpace method*), 92
`get_job_state()` (*infinisdk.infinibox.replica.Replica method*), 83
`get_jobs()` (*infinisdk.infinibox.replica.Replica method*), 84
`get_json()` (*infinisdk.core.api.api.Response method*), 44
`get_last_replicated_guid()` (*infinisdk.infinibox.replica.Replica method*), 84
`get_last_synchronized()` (*infinisdk.infinibox.replica.Replica method*), 84
`get_latency()` (*infinisdk.infinibox.replica.Replica method*), 84
`get_level()` (*infinisdk.core.events.Event method*), 93
`get_link()` (*infinisdk.infinibox.replica.Replica method*), 69
`get_link()` (*infinisdk.infinibox.rg_replica.RgReplica method*), 69
`get_link_mode()` (*infinisdk.infinibox.link.Link method*), 90
`get_link_state()` (*infinisdk.infinibox.components.FcPort method*), 105
`get_link_state()` (*infinisdk.infinibox.link.Link method*), 90
`get_linked_system()` (*infinisdk.infinibox.link.Link method*), 90
`get_local_cg()` (*infinisdk.infinibox.replica.Replica method*), 84
`get_local_cg_id()` (*infinisdk.infinibox.replica.Replica method*), 84
`get_local_cg_name()` (*infinisdk.infinibox.replica.Replica method*), 84
`get_local_data_entities()` (*infinisdk.infinibox.replica.Replica method*), 84
`get_local_entity()` (*infinisdk.infinibox.replica.Replica method*), 84

```

get_local_entity_name() (infin-
    isdk.infinibox.replica.Replica method), 84
get_local_filesystem() (infin-
    isdk.infinibox.replica.Replica method), 84
get_local_link_guid() (infin-
    isdk.infinibox.rg_replica.RgReplica method),
    69
get_local_pool_id() (infin-
    isdk.infinibox.replica.Replica method), 84
get_local_pool_name() (infin-
    isdk.infinibox.replica.Replica method), 84
get_local_replication_network_space()
    (infinisdk.infinibox.link.Link method), 90
get_local_volume() (infin-
    isdk.infinibox.replica.Replica method), 84
get_local_witness_state() (infin-
    isdk.infinibox.link.Link method), 90
get_lock_expires_at() (infin-
    isdk.infinibox.cons_group.ConsGroup method),
    101
get_lock_expires_at() (infin-
    isdk.infinibox.dataset.Dataset method), 45
get_lock_expires_at() (infin-
    isdk.infinibox.filesystem.Filesystem method),
    57
get_lock_expires_at() (infin-
    isdk.infinibox.volume.Volume method), 51
get_lock_remote_snapshot_retention()
    (infinisdk.infinibox.replica.Replica method), 84
get_lock_state() (infin-
    isdk.infinibox.cons_group.ConsGroup method),
    101
get_lock_state() (infin-
    isdk.infinibox.dataset.Dataset method), 46
get_lock_state() (infin-
    isdk.infinibox.filesystem.Filesystem method),
    57
get_lock_state() (infin-
    isdk.infinibox.volume.Volume method), 51
get_lun() (infinisdk.infinibox.lun.LogicalUnit
    method), 113
get_lun() (infinisdk.infinibox.volume.Volume
    method), 51
get_luns() (infinisdk.infinibox.host.Host method), 78
get_management_service() (infin-
    isdk.infinibox.components.Node method),
    103
get_max_bps() (infin-
    isdk.infinibox.qos_policy.QosPolicy method),
    107
get_max_extend() (infinisdk.infinibox.pool.Pool
    method), 72
get_max_ops() (infin-
    isdk.infinibox.qos_policy.QosPolicy method),
    107
get_max_read() (infinisdk.infinibox.export.Export
    method), 63
get_max_write() (infinisdk.infinibox.export.Export
    method), 63
get_members() (infin-
    isdk.infinibox.cons_group.ConsGroup method),
    101
get_members_count() (infin-
    isdk.infinibox.cons_group.ConsGroup method),
    101
get_members_count() (infin-
    isdk.infinibox.replication_group.ReplicationGroup
    method), 68
get_metadata() (infinisdk.core.api.api.Response
    method), 44
get_metadata_value() (infin-
    isdk.infinibox.filesystem.Filesystem method),
    57
get_metadata_value() (infin-
    isdk.infinibox.host.Host method), 78
get_metadata_value() (infin-
    isdk.infinibox.pool.Pool method), 72
get_metadata_value() (infin-
    isdk.infinibox.qos_policy.QosPolicy method),
    107
get_metadata_value() (infin-
    isdk.infinibox.volume.Volume method), 51
get_model() (infinisdk.infinibox.components.Node
    method), 103
get_model_name() (infinisdk.infinibox.InfiniBox
    method), 42
get_mtu() (infinisdk.infinibox.network_space.NetworkSpace
    method), 92
get_mutable_fields() (infin-
    isdk.core.type_binder.MonomorphicBinder
    method), 114
get_mutable_fields() (infin-
    isdk.infinibox.host.HostBinder method),
    76
get_name() (infinisdk.infinibox.components.Node
    method), 103
get_name() (infinisdk.infinibox.cons_group.ConsGroup
    method), 101
get_name() (infinisdk.infinibox.filesystem.Filesystem
    method), 58
get_name() (infinisdk.infinibox.host.Host method), 78
get_name() (infinisdk.infinibox.host_cluster.HostCluster
    method), 81
get_name() (infinisdk.infinibox.InfiniBox method), 42
get_name() (infinisdk.infinibox ldap_config.LDAPConfig
    method), 97
get_name() (infinisdk.infinibox.link.Link method), 90
get_name() (infinisdk.infinibox.network_space.NetworkSpace

```

`method`), 92
`get_name()` (`infinisdk.infinibox.notification_target.NotificationTarget` `method`), 98
`get_name()` (`infinisdk.infinibox.pool.Pool` `method`), 72
`get_name()` (`infinisdk.infinibox.qos_policy.QosPolicy` `method`), 107
`get_name()` (`infinisdk.infinibox.replication_group.ReplicationGroup` `method`), 68
`get_name()` (`infinisdk.infinibox.share.Share` `method`), 66
`get_name()` (`infinisdk.infinibox.smb_group.SMBGroup` `method`), 110
`get_name()` (`infinisdk.infinibox.smb_user.SMBUser` `method`), 111
`get_name()` (`infinisdk.infinibox.tenant.Tenant` `method`), 109
`get_name()` (`infinisdk.infinibox.user.User` `method`), 95
`get_name()` (`infinisdk.infinibox.volume.Volume` `method`), 51
`get_network_config()` (`infinisdk.infinibox.network_space.NetworkSpace` `method`), 92
`get_next_job_start_time()` (`infinisdk.infinibox.replica.Replica` `method`), 85
`get_next_restore_point()` (`infinisdk.infinibox.replica.Replica` `method`), 85
`get_nfs_allow_unmapped_users()` (`infinisdk.infinibox.tenant.Tenant` `method`), 109
`get_nfs_group_policy()` (`infinisdk.infinibox.tenant.Tenant` `method`), 109
`get_nguid()` (`infinisdk.infinibox.volume.Volume` `method`), 51
`get_node()` (`infinisdk.infinibox.components.FcPort` `method`), 105
`get_num_blocks()` (`infinisdk.infinibox.dataset.Dataset` `method`), 46
`get_num_blocks()` (`infinisdk.infinibox.filesystem.Filesystem` `method`), 58
`get_num_blocks()` (`infinisdk.infinibox.volume.Volume` `method`), 51
`get_offline_caching()` (`infinisdk.infinibox.share.Share` `method`), 66
`get_owned_pools()` (`infinisdk.infinibox.user.User` `method`), 95
`get_owners()` (`infinisdk.infinibox.pool.Pool` `method`), 72
`get_parent()` (`infinisdk.infinibox.cons_group.ConsGroup` `method`), 101
`get_parent()` (`infinisdk.infinibox.filesystem.Filesystem` `method`), 58
`get_parent()` (`infinisdk.infinibox.volume.Volume` `method`), 51
`get_password_digest_version()` (`infinisdk.infinibox.user.User` `method`), 95
`get_pending_job_count()` (`infinisdk.infinibox.replica.Replica` `method`), 85
`get_permanent_failure_wait_interval()` (`infinisdk.infinibox.replica.Replica` `method`), 85
`get_permissions()` (`infinisdk.infinibox.export.Export` `method`), 63
`get_physical_capacity()` (`infinisdk.infinibox.pool.Pool` `method`), 73
`get_physical_capacity_critical()` (`infinisdk.infinibox.pool.Pool` `method`), 73
`get_physical_capacity_warning()` (`infinisdk.infinibox.pool.Pool` `method`), 73
`get_pool()` (`infinisdk.infinibox.cons_group.ConsGroup` `method`), 101
`get_pool()` (`infinisdk.infinibox.dataset.Dataset` `method`), 46
`get_pool()` (`infinisdk.infinibox.filesystem.Filesystem` `method`), 58
`get_pool()` (`infinisdk.infinibox.replication_group.ReplicationGroup` `method`), 68
`get_pool()` (`infinisdk.infinibox.rg_replica.RgReplica` `method`), 69
`get_pool()` (`infinisdk.infinibox.volume.Volume` `method`), 51
`get_pool_name()` (`infinisdk.infinibox.cons_group.ConsGroup` `method`), 101
`get_pool_name()` (`infinisdk.infinibox.dataset.Dataset` `method`), 46
`get_pool_name()` (`infinisdk.infinibox.filesystem.Filesystem` `method`), 58
`get_pool_name()` (`infinisdk.infinibox.volume.Volume` `method`), 52
`get_port()` (`infinisdk.infinibox.notification_target.NotificationTarget` `method`), 98
`get_ports()` (`infinisdk.infinibox.host.Host` `method`), 78
`get_pref_read()` (`infinisdk.infinibox.export.Export` `method`), 63
`get_pref_readdir()` (`infinisdk.infinibox.export.Export` `method`), 63
`get_pref_write()` (`infinisdk.infinibox.export.Export` `method`), 63
`get_primary_group()` (`infinisdk.infinibox.smb_user.SMBUser` `method`), 111
`get_private_key()` (`infinisdk.infinibox.smb_user.SMBUser` `method`), 111

<code>isdk.infinibox.notification_target.NotificationTarget</code>	<code>isdk.infinibox.replica.Replica method), 85</code>
<code>method), 98</code>	<code>get_remote_entity_pairs() (infin-</code>
<code>get_private_protocol() (infin-</code>	<code>isdk.infinibox.replica.Replica method), 85</code>
<code>isdk.infinibox.notification_target.NotificationTarget</code>	<code>get_remote_host() (infinisdk.infinibox.link.Link</code>
<code>method), 98</code>	<code>method), 90</code>
<code>get_privileges() (infin-</code>	<code>get_remote_link() (infinisdk.infinibox.link.Link</code>
<code>isdk.infinibox.smb_group.SMBGroup method),</code>	<code>method), 90</code>
<code>110</code>	<code>get_remote_link_id() (infin-</code>
<code>get_privileges() (infin-</code>	<code>isdk.infinibox.link.Link method), 91</code>
<code>isdk.infinibox.smb_user.SMBUser method),</code>	<code>get_remote_pool_id() (infin-</code>
<code>111</code>	<code>isdk.infinibox.replica.Replica method), 85</code>
<code>get_progress() (infinisdk.infinibox.replica.Replica</code>	<code>get_remote_pool_id() (infin-</code>
<code>method), 85</code>	<code>isdk.infinibox.rg_replica.RgReplica method),</code>
<code>get_properties() (infin-</code>	<code>69</code>
<code>isdk.infinibox.network_space.NetworkSpace</code>	<code>get_remote_pool_name() (infin-</code>
<code>method), 92</code>	<code>isdk.infinibox.replica.Replica method), 85</code>
<code>get_protocol() (infin-</code>	<code>get_remote_replica() (infin-</code>
<code>isdk.infinibox.notification_target.NotificationTarget</code>	<code>isdk.infinibox.replica.Replica method), 86</code>
<code>method), 98</code>	<code>get_remote_replica_id() (infin-</code>
<code>get_provisioning() (infin-</code>	<code>isdk.infinibox.replica.Replica method), 86</code>
<code>isdk.infinibox.dataset.Dataset method), 46</code>	<code>get_remote_replica_id() (infin-</code>
<code>get_provisioning() (infin-</code>	<code>isdk.infinibox.rg_replica.RgReplica method),</code>
<code>isdk.infinibox.filesystem.Filesystem method),</code>	<code>69</code>
<code>58</code>	<code>get_remote_replication_group_id() (infin-</code>
<code>get_provisioning() (infin-</code>	<code>isdk.infinibox.rg_replica.RgReplica method),</code>
<code>isdk.infinibox.volume.Volume method), 52</code>	<code>69</code>
<code>get_qos_policies() (infinisdk.infinibox.pool.Pool</code>	<code>get_remote_replication_group_name() (in-</code>
<code>method), 73</code>	<code>finisdk.infinibox.rg_replica.RgReplica method),</code>
<code>get_qos_policy() (infin-</code>	<code>69</code>
<code>isdk.infinibox.dataset.Dataset method), 46</code>	<code>get_remote_snapshot_suffix() (infin-</code>
<code>get_qos_policy() (infin-</code>	<code>isdk.infinibox.replica.Replica method), 86</code>
<code>isdk.infinibox.filesystem.Filesystem method),</code>	<code>get_remote_system_name() (infin-</code>
<code>58</code>	<code>isdk.infinibox.link.Link method), 91</code>
<code>get_qos_policy() (infin-</code>	<code>get_remote_system_serial_number() (infin-</code>
<code>isdk.infinibox.volume.Volume method), 52</code>	<code>isdk.infinibox.link.Link method), 91</code>
<code>get_qos_shared_policy() (infin-</code>	<code>get_replica_configuration_guid() (infin-</code>
<code>isdk.infinibox.dataset.Dataset method), 46</code>	<code>isdk.infinibox.rg_replica.RgReplica method),</code>
<code>get_qos_shared_policy() (infin-</code>	<code>69</code>
<code>isdk.infinibox.filesystem.Filesystem method),</code>	<code>get_replica_ids() (infin-</code>
<code>58</code>	<code>isdk.infinibox.dataset.Dataset method), 46</code>
<code>get_qos_shared_policy() (infin-</code>	<code>get_replica_ids() (infin-</code>
<code>isdk.infinibox.volume.Volume method), 52</code>	<code>isdk.infinibox.filesystem.Filesystem method),</code>
<code>get_rate_limit() (infin-</code>	<code>58</code>
<code>isdk.infinibox.network_space.NetworkSpace</code>	<code>get_replica_ids() (infin-</code>
<code>method), 92</code>	<code>isdk.infinibox.volume.Volume method), 52</code>
<code>get_remote_cg_id() (infin-</code>	<code>get_replication_group() (infin-</code>
<code>isdk.infinibox.replica.Replica method), 85</code>	<code>isdk.infinibox.rg_replica.RgReplica method),</code>
<code>get_remote_cg_name() (infin-</code>	<code>69</code>
<code>isdk.infinibox.replica.Replica method), 85</code>	<code>get_replication_type() (infin-</code>
<code>get_remote_data_entities() (infin-</code>	<code>isdk.infinibox.replica.Replica method), 86</code>
<code>isdk.infinibox.replica.Replica method), 85</code>	<code>get_replication_type() (infin-</code>
<code>get_remote_entity() (infin-</code>	<code>isdk.infinibox.rg_replica.RgReplica method),</code>
<code>isdk.infinibox.replica.Replica method), 85</code>	<code>69</code>
<code>get_remote_entity_name() (infin-</code>	<code>get_replication_types() (infin-</code>

`isdk.infinibox.cons_group.ConsGroup method`), 101
`get_replication_types()` (`infinisdk.infinibox.dataset.Dataset method`), 46
`get_replication_types()` (`infinisdk.infinibox.filesystem.Filesystem method`), 58
`get_replication_types()` (`infinisdk.infinibox.volume.Volume method`), 52
`get_reporter()` (`infinisdk.core.events.Event method`), 94
`get_reserved_capacity()` (`infinisdk.infinibox.pool.Pool method`), 73
`get_resiliency_mode()` (`infinisdk.infinibox.link.Link method`), 91
`get_restore_point()` (`infinisdk.infinibox.replica.Replica method`), 86
`get_result()` (`infinisdk.core.api.api.Response method`), 44
`get_rmr_snapshot_guid()` (`infinisdk.infinibox.cons_group.ConsGroup method`), 101
`get_rmr_snapshot_guid()` (`infinisdk.infinibox.dataset.Dataset method`), 46
`get_rmr_snapshot_guid()` (`infinisdk.infinibox.filesystem.Filesystem method`), 58
`get_rmr_snapshot_guid()` (`infinisdk.infinibox.volume.Volume method`), 52
`get_role()` (`infinisdk.infinibox.components.FcPort method`), 105
`get_role()` (`infinisdk.infinibox.replica.Replica method`), 86
`get_role()` (`infinisdk.infinibox.rg_replica.RgReplica method`), 69
`get_role()` (`infinisdk.infinibox.user.User method`), 95
`get_roles()` (`infinisdk.infinibox.user.User method`), 95
`get_rpo()` (`infinisdk.infinibox.replica.Replica method`), 86
`get_rpo()` (`infinisdk.infinibox.rg_replica.RgReplica method`), 69
`get_rpo_state()` (`infinisdk.infinibox.replica.Replica method`), 86
`get_rpo_type()` (`infinisdk.infinibox.replica.Replica method`), 86
`get_san_client_type()` (`infinisdk.infinibox.host.Host method`), 78
`get_san_client_type()` (`infinisdk.infinibox.host_cluster.HostCluster method`), 81
`get_security()` (`infinisdk.infinibox.components.Node method`), 104
`get_security_chap_inbound_username()` (`infinisdk.infinibox.host.Host method`), 78
`get_security_chap_outbound_username()` (`infinisdk.infinibox.host.Host method`), 79
`get_security_method()` (`infinisdk.infinibox.host.Host method`), 79
`get_security_style()` (`infinisdk.infinibox.filesystem.Filesystem method`), 58
`get_serial()` (`infinisdk.infinibox.InfiniBox method`), 42
`get_serial()` (`infinisdk.infinibox.volume.Volume method`), 52
`get_serial_number()` (`infinisdk.infinibox.components.Drive method`), 105
`get_service()` (`infinisdk.infinibox.components.Node method`), 104
`get_service()` (`infinisdk.infinibox.network_space.NetworkSpace method`), 92
`get_services()` (`infinisdk.infinibox.components.Node method`), 104
`get_share()` (`infinisdk.infinibox.share_permission.SharePermission method`), 67
`get_short_tenant_key()` (`infinisdk.infinibox.tenant.Tenant method`), 109
`get_sid()` (`infinisdk.infinibox.share_permission.SharePermission method`), 67
`get_sid()` (`infinisdk.infinibox.smb_group.SMBGroup method`), 110
`get_sid()` (`infinisdk.infinibox.smb_user.SMBUser method`), 111
`get_size()` (`infinisdk.infinibox.dataset.Dataset method`), 46
`get_size()` (`infinisdk.infinibox.filesystem.Filesystem method`), 58
`get_size()` (`infinisdk.infinibox.volume.Volume method`), 52
`get_snapdir_name()` (`infinisdk.infinibox.filesystem.Filesystem method`), 59
`get_snapshot_expires_at()` (`infinisdk.infinibox.dataset.Dataset method`), 46
`get_snapshot_expires_at()` (`infinisdk.infinibox.filesystem.Filesystem method`), 59
`get_snapshot_expires_at()` (`infinisdk.infinibox.volume.Volume method`), 52
`get_snapshot_retention()` (`infinisdk.infinibox.dataset.Dataset method`), 46
`get_snapshot_retention()` (`infinisdk.infinibox.filesystem.Filesystem method`),

59

`get_snapshot_retention()` (*infinisdk.infinibox.volume.Volume method*), 52

`get_snapshots()` (*infinisdk.infinibox.dataset.Dataset method*), 47

`get_snapshots()` (*infinisdk.infinibox.filesystem.Filesystem method*), 59

`get_snapshots()` (*infinisdk.infinibox.volume.Volume method*), 52

`get_snapshots_count()` (*infinisdk.infinibox.pool.Pool method*), 73

`get_snapshots_retention()` (*infinisdk.infinibox.replica.Replica method*), 86

`get_soft_target_addresses()` (*infinisdk.infinibox.components.FcPort method*), 105

`get_source_node_id()` (*infinisdk.core.events.Event method*), 94

`get_staging_area_allocated_size()` (*infinisdk.infinibox.replica.Replica method*), 86

`get_standard_entities_count()` (*infinisdk.infinibox.pool.Pool method*), 73

`get_standard_filesystem_snapshots_count()` (*infinisdk.infinibox.pool.Pool method*), 73

`get_standard_filesystems_count()` (*infinisdk.infinibox.pool.Pool method*), 73

`get_standard_snapshots_count()` (*infinisdk.infinibox.pool.Pool method*), 73

`get_standard_volumes_count()` (*infinisdk.infinibox.pool.Pool method*), 73

`get_started_at()` (*infinisdk.infinibox.replica.Replica method*), 86

`get_state()` (*infinisdk.infinibox.components.Drive method*), 105

`get_state()` (*infinisdk.infinibox.components.Enclosure method*), 104

`get_state()` (*infinisdk.infinibox.components.FcPort method*), 105

`get_state()` (*infinisdk.infinibox.components.Node method*), 104

`get_state()` (*infinisdk.infinibox.pool.Pool method*), 73

`get_state()` (*infinisdk.infinibox.replica.Replica method*), 86

`get_state()` (*infinisdk.infinibox.rg_replica.RgReplica method*), 69

`get_state_description()` (*infinisdk.infinibox.link.Link method*), 91

`get_state_description()` (*infinisdk.infinibox.replica.Replica method*), 87

`get_state_description()` (*infinisdk.infinibox.rg_replica.RgReplica method*), 70

`get_state_reason()` (*infinisdk.infinibox.replica.Replica method*), 87

`get_state_reason()` (*infinisdk.infinibox.rg_replica.RgReplica method*), 70

`get_subsystem_nqn()` (*infinisdk.infinibox.host.Host method*), 79

`get_switch_vendor()` (*infinisdk.infinibox.components.FcPort method*), 106

`get_sync_duration()` (*infinisdk.infinibox.replica.Replica method*), 87

`get_sync_interval()` (*infinisdk.infinibox.replica.Replica method*), 87

`get_sync_interval()` (*infinisdk.infinibox.rg_replica.RgReplica method*), 70

`get_sync_state()` (*infinisdk.infinibox.replica.Replica method*), 87

`get_system_version()` (*infinisdk.core.events.Event method*), 94

`get_temporary_failure_retry_count()` (*infinisdk.infinibox.replica.Replica method*), 87

`get_temporary_failure_retry_interval()` (*infinisdk.infinibox.replica.Replica method*), 87

`get_tenant()` (*infinisdk.core.events.Event method*), 94

`get_tenant()` (*infinisdk.infinibox.cons_group.ConsGroup method*), 102

`get_tenant()` (*infinisdk.infinibox.dataset.Dataset method*), 47

`get_tenant()` (*infinisdk.infinibox.export.Export method*), 63

`get_tenant()` (*infinisdk.infinibox.filesystem.Filesystem method*), 59

`get_tenant()` (*infinisdk.infinibox.host.Host method*), 79

`get_tenant()` (*infinisdk.infinibox.host_cluster.HostCluster method*), 81

`get_tenant()` (*infinisdk.infinibox.network_space.NetworkSpace method*), 92

`get_tenant()` (*infinisdk.infinibox.pool.Pool method*), 73

`get_tenant()` (*infinisdk.infinibox.share.Share method*), 66

`get_tenant()` (*infinisdk.infinibox.smb_group.SMBGroup method*), 110

`get_tenant()` (*infinisdk.infinibox.smb_user.SMBUser method*), 111

`get_tenant()` (*infinisdk.infinibox.volume.Volume method*), 52
`get_throughput()` (*infinisdk.infinibox.replica.Replica method*), 87
`get_timestamp()` (*infinisdk.core.events.Event method*), 94
`get_tpm()` (*infinisdk.infinibox.components.Node method*), 104
`get_transport()` (*infinisdk.infinibox.notification_target.NotificationTarget method*), 98
`get_transport_protocols()` (*infinisdk.infinibox.export.Export method*), 63
`get_tree_allocated()` (*infinisdk.infinibox.dataset.Dataset method*), 47
`get_tree_allocated()` (*infinisdk.infinibox.filesystem.Filesystem method*), 59
`get_tree_allocated()` (*infinisdk.infinibox.volume.Volume method*), 52
`get_type()` (*infinisdk.infinibox.cons_group.ConsGroup method*), 102
`get_type()` (*infinisdk.infinibox.dataset.Dataset method*), 47
`get_type()` (*infinisdk.infinibox.filesystem.Filesystem method*), 59
`get_type()` (*infinisdk.infinibox.pool.Pool method*), 74
`get_type()` (*infinisdk.infinibox.qos_policy.QosPolicy method*), 107
`get_type()` (*infinisdk.infinibox.user.User method*), 96
`get_type()` (*infinisdk.infinibox.volume.Volume method*), 53
`get_udid()` (*infinisdk.infinibox.volume.Volume method*), 53
`get_uid()` (*infinisdk.infinibox.smb_group.SMBGroup method*), 110
`get_uid()` (*infinisdk.infinibox.smb_user.SMBUser method*), 111
`get_unapproved_context()` (*infinisdk.core.api.api.API method*), 43
`get_updated_at()` (*infinisdk.infinibox.cons_group.ConsGroup method*), 102
`get_updated_at()` (*infinisdk.infinibox.dataset.Dataset method*), 47
`get_updated_at()` (*infinisdk.infinibox.export.Export method*), 63
`get_updated_at()` (*infinisdk.infinibox.filesystem.Filesystem method*), 59
`get_updated_at()` (*infinisdk.infinibox.host.Host method*), 79
`get_updated_at()` (*infinisdk.infinibox.host_cluster.HostCluster method*), 81
`get_updated_at()` (*infinisdk.infinibox.pool.Pool method*), 74
`get_updated_at()` (*infinisdk.infinibox.qos_policy.QosPolicy method*), 107
`get_updated_at()` (*infinisdk.infinibox.replica.Replica method*), 87
`get_updated_at()` (*infinisdk.infinibox.replication_group.ReplicationGroup method*), 68
`get_updated_at()` (*infinisdk.infinibox.rg_replica.RgReplica method*), 70
`get_updated_at()` (*infinisdk.infinibox.share.Share method*), 66
`get_updated_at()` (*infinisdk.infinibox.tenant.Tenant method*), 109
`get_updated_at()` (*infinisdk.infinibox.volume.Volume method*), 53
`get_used_size()` (*infinisdk.infinibox.dataset.Dataset method*), 47
`get_used_size()` (*infinisdk.infinibox.filesystem.Filesystem method*), 59
`get_used_size()` (*infinisdk.infinibox.volume.Volume method*), 53
`get_username()` (*infinisdk.core.events.Event method*), 94
`get_username()` (*infinisdk.infinibox.notification_target.NotificationTarget method*), 98
`get_uuid()` (*infinisdk.infinibox.replication_group.ReplicationGroup method*), 68
`get_version()` (*infinisdk.infinibox.InfiniBox method*), 42
`get_version()` (*infinisdk.infinibox.notification_target.NotificationTarget method*), 99
`get_virtual_capacity()` (*infinisdk.infinibox.pool.Pool method*), 74
`get_visibility()` (*infinisdk.core.events.Event method*), 94
`get_visibility()` (*infinisdk.infinibox.notification_target.NotificationTarget method*), 99
`get_volume()` (*infinisdk.infinibox.lun.LogicalUnit method*), 113
`get_volumes_count()` (*infinisdk.infinibox.pool.Pool method*), 74
`get_vvol_entities_count()` (*infinisdk.infinibox.pool.Pool method*), 74
`get_vvol_snapshots_count()` (*infinisdk.infinibox.pool.Pool method*), 74

`get_vvol_volumes_count()` (*infinisdk.infinibox.pool.Pool* method), 74

`get_witness_address()` (*infinisdk.infinibox.link.Link* method), 91

`get_wwpn()` (*infinisdk.infinibox.components.FcPort* method), 106

H

`has_children()` (*infinisdk.infinibox.dataset.Dataset* method), 47

`has_children()` (*infinisdk.infinibox.filesystem.Filesystem* method), 59

`has_children()` (*infinisdk.infinibox.volume.Volume* method), 53

`has_registered_initiator_address()` (*infinisdk.infinibox.host.HostBinder* method), 76

`Host` (class in *infinisdk.infinibox.host*), 77

`HostBinder` (class in *infinisdk.infinibox.host*), 75

`HostCluster` (class in *infinisdk.infinibox.host_cluster*), 81

I

`ieee_company_id` (*infinisdk.infinibox.scsi_serial.SCSISerial* attribute), 114

`InfiniBox` (class in *infinisdk.infinibox*), 42

`InfiniBoxObject` (class in *infinisdk.infinibox.system_object*), 113

`InfiniBoxSystemComponents` (class in *infinisdk.infinibox.components*), 102

`infinisdk.core.api.api` (module), 43

`infinisdk.core.events` (module), 93

`infinisdk.core.exceptions` (module), 116

`infinisdk.core.system_object` (module), 115

`infinisdk.core.type_binder` (module), 114

`infinisdk.infinibox` (module), 42

`infinisdk.infinibox.active_directory` (module), 112

`infinisdk.infinibox.components` (module), 102

`infinisdk.infinibox.cons_group` (module), 100

`infinisdk.infinibox.dataset` (module), 44

`infinisdk.infinibox.events` (module), 94

`infinisdk.infinibox.export` (module), 62

`infinisdk.infinibox.filesystem` (module), 55

`infinisdk.infinibox.host` (module), 75

`infinisdk.infinibox.host_cluster` (module), 81

`infinisdk.infinibox.ldap_config` (module), 97

`infinisdk.infinibox.link` (module), 90

`infinisdk.infinibox.lun` (module), 113

`infinisdk.infinibox.notification_target` (module), 97

`infinisdk.infinibox.pool` (module), 70

`infinisdk.infinibox.qos_policy` (module), 106

`infinisdk.infinibox.replica` (module), 81

`infinisdk.infinibox.replication_group` (module), 68

`infinisdk.infinibox.rg_replica` (module), 68

`infinisdk.infinibox.scsi_serial` (module), 114

`infinisdk.infinibox.share` (module), 65

`infinisdk.infinibox.share_permission` (module), 67

`infinisdk.infinibox.smb_group` (module), 110

`infinisdk.infinibox.smb_user` (module), 111

`infinisdk.infinibox.system_object` (module), 113

`infinisdk.infinibox.tenant` (module), 108

`infinisdk.infinibox.volume` (module), 49

`invalidate_cache()` (*infinisdk.core.system_object.SystemObject* method), 115

`invalidate_cache()` (*infinisdk.infinibox.filesystem.Filesystem* method), 59

`invalidate_cache()` (*infinisdk.infinibox.host.Host* method), 79

`invalidate_cache()` (*infinisdk.infinibox.pool.Pool* method), 74

`invalidate_cache()` (*infinisdk.infinibox.qos_policy.QosPolicy* method), 107

`invalidate_cache()` (*infinisdk.infinibox.user.User* method), 96

`invalidate_cache()` (*infinisdk.infinibox.volume.Volume* method), 53

`is_32bit_file_id()` (*infinisdk.infinibox.export.Export* method), 63

`is_access_based_enumeration()` (*infinisdk.infinibox.share.Share* method), 66

`is_active()` (*infinisdk.infinibox.replica.Replica* method), 87

`is_async_mode()` (*infinisdk.infinibox.replica.Replica* method), 87

`is_async_only()` (*infinisdk.infinibox.link.Link* method), 91

`is_auto_suspended()` (*infinisdk.infinibox.replica.Replica* method), 87

`is_automatic_ip_failback()` (*infin-*

isdk.infinibox.network_space.NetworkSpace
method), 92
is_burst_enabled() (*infin-*
isdk.infinibox.qos_policy.QosPolicy *method*),
108
is_compression_enabled() (*infin-*
isdk.infinibox.dataset.Dataset *method*), 47
is_compression_enabled() (*infin-*
isdk.infinibox.filesystem.Filesystem *method*),
59
is_compression_enabled() (*infin-*
isdk.infinibox.pool.Pool *method*), 74
is_compression_enabled() (*infin-*
isdk.infinibox.volume.Volume *method*), 53
is_compression_suppressed() (*infin-*
isdk.infinibox.dataset.Dataset *method*), 47
is_compression_suppressed() (*infin-*
isdk.infinibox.filesystem.Filesystem *method*),
59
is_compression_suppressed() (*infin-*
isdk.infinibox.volume.Volume *method*), 53
is_concurrent_replica() (*infin-*
isdk.infinibox.replica.Replica *method*), 87
is_consistency_group() (*infin-*
isdk.infinibox.replica.Replica *method*), 87
is_domino() (*infinisdk.infinibox.replica.Replica*
method), 87
is_enabled() (*infinisdk.infinibox.components.FcPort*
method), 106
is_enabled() (*infinisdk.infinibox.export.Export*
method), 63
is_enabled() (*infinisdk.infinibox.share.Share*
method), 66
is_enabled() (*infinisdk.infinibox.smb_user.SMBUser*
method), 112
is_enabled() (*infinisdk.infinibox.user.User* *method*),
96
is_encryption_state() (*infin-*
isdk.infinibox.components.Drive *method*),
105
is_established() (*infin-*
isdk.infinibox.filesystem.Filesystem *method*),
59
is_filesystem() (*infin-*
isdk.infinibox.replica.Replica *method*), 88
is_idle() (*infinisdk.infinibox.replica.Replica*
method), 88
is_in_system() (*infin-*
isdk.core.system_object.SystemObject *method*),
115
is_in_system() (*infin-*
isdk.infinibox.filesystem.Filesystem *method*),
60
is_in_system() (*infinisdk.infinibox.host.Host*
method), 79
is_in_system() (*infinisdk.infinibox.pool.Pool*
method), 74
is_in_system() (*infin-*
isdk.infinibox.qos_policy.QosPolicy *method*),
108
is_in_system() (*infinisdk.infinibox.user.User*
method), 96
is_in_system() (*infinisdk.infinibox.volume.Volume*
method), 53
is_including_snapshots() (*infin-*
isdk.infinibox.replica.Replica *method*), 88
is_initial() (*infinisdk.infinibox.replica.Replica*
method), 88
is_initial_replication() (*infin-*
isdk.infinibox.replica.Replica *method*), 88
is_initializing() (*infin-*
isdk.infinibox.replica.Replica *method*), 88
is_initializing_pending() (*infin-*
isdk.infinibox.replica.Replica *method*), 88
is_is_digest_sufficient() (*infin-*
isdk.infinibox.user.User *method*), 96
is_is_local_link_ready_for_sync() (*infin-*
isdk.infinibox.link.Link *method*), 91
is_logged_in() (*infinisdk.infinibox.InfiniBox*
method), 42
is_make_all_users_anonymous() (*infin-*
isdk.infinibox.export.Export *method*), 64
is_mapped() (*infinisdk.infinibox.dataset.Dataset*
method), 47
is_mapped() (*infinisdk.infinibox.filesystem.Filesystem*
method), 60
is_mapped() (*infinisdk.infinibox.volume.Volume*
method), 53
is_master() (*infinisdk.infinibox.dataset.Dataset*
method), 47
is_master() (*infinisdk.infinibox.filesystem.Filesystem*
method), 60
is_master() (*infinisdk.infinibox.volume.Volume*
method), 53
is_optimized() (*infinisdk.infinibox.host.Host*
method), 79
is_out_of_sync() (*infin-*
isdk.infinibox.replica.Replica *method*), 88
is_paths_available() (*infin-*
isdk.infinibox.volume.Volume *method*), 53
is_pending() (*infinisdk.infinibox.replica.Replica*
method), 88
is_preferred() (*infinisdk.infinibox.replica.Replica*
method), 88
is_privileged_port() (*infin-*
isdk.infinibox.export.Export *method*), 64
is_replicated() (*infin-*
isdk.infinibox.cons_group.ConsGroup *method*),

[102](#)
`is_replicated()` (*infinisdk.infinibox.dataset.Dataset method*), [47](#)
`is_replicated()` (*infinisdk.infinibox.filesystem.Filesystem method*), [60](#)
`is_replicated()` (*infinisdk.infinibox.replication_group.ReplicationGroup method*), [68](#)
`is_replicated()` (*infinisdk.infinibox.volume.Volume method*), [53](#)
`is_replicating()` (*infinisdk.infinibox.replica.Replica method*), [88](#)
`is_require_encryption()` (*infinisdk.infinibox.share.Share method*), [66](#)
`is_rmr_active_active_peer()` (*infinisdk.infinibox.dataset.Dataset method*), [47](#)
`is_rmr_active_active_peer()` (*infinisdk.infinibox.filesystem.Filesystem method*), [60](#)
`is_rmr_active_active_peer()` (*infinisdk.infinibox.volume.Volume method*), [53](#)
`is_rmr_source()` (*infinisdk.infinibox.dataset.Dataset method*), [47](#)
`is_rmr_source()` (*infinisdk.infinibox.filesystem.Filesystem method*), [60](#)
`is_rmr_source()` (*infinisdk.infinibox.volume.Volume method*), [53](#)
`is_rmr_target()` (*infinisdk.infinibox.dataset.Dataset method*), [47](#)
`is_rmr_target()` (*infinisdk.infinibox.filesystem.Filesystem method*), [60](#)
`is_rmr_target()` (*infinisdk.infinibox.volume.Volume method*), [54](#)
`is_security_chap_has_inbound_secret()` (*infinisdk.infinibox.host.Host method*), [79](#)
`is_security_chap_has_outbound_secret()` (*infinisdk.infinibox.host.Host method*), [79](#)
`is_snapdir_accessible()` (*infinisdk.infinibox.filesystem.Filesystem method*), [60](#)
`is_snapdir_visible()` (*infinisdk.infinibox.export.Export method*), [64](#)
`is_snapdir_visible()` (*infinisdk.infinibox.share.Share method*), [66](#)
`is_snapgroup()` (*infinisdk.infinibox.cons_group.ConsGroup method*), [102](#)
`is_snapshot()` (*infinisdk.infinibox.dataset.Dataset method*), [48](#)
`is_snapshot()` (*infinisdk.infinibox.filesystem.Filesystem method*), [60](#)
`is_snapshot()` (*infinisdk.infinibox.volume.Volume method*), [54](#)
`is_source()` (*infinisdk.infinibox.replica.Replica method*), [88](#)
`is_ssd_enabled()` (*infinisdk.infinibox.dataset.Dataset method*), [48](#)
`is_ssd_enabled()` (*infinisdk.infinibox.filesystem.Filesystem method*), [60](#)
`is_ssd_enabled()` (*infinisdk.infinibox.pool.Pool method*), [74](#)
`is_ssd_enabled()` (*infinisdk.infinibox.volume.Volume method*), [54](#)
`is_suspended()` (*infinisdk.infinibox.replica.Replica method*), [88](#)
`is_suspended_from_local()` (*infinisdk.infinibox.replica.Replica method*), [88](#)
`is_sync_in_progress()` (*infinisdk.infinibox.replica.Replica method*), [88](#)
`is_synchronized()` (*infinisdk.infinibox.replica.Replica method*), [88](#)
`is_target()` (*infinisdk.infinibox.replica.Replica method*), [88](#)
`is_tls()` (*infinisdk.infinibox.notification_target.NotificationTarget method*), [99](#)
`is_user_suspended()` (*infinisdk.infinibox.replica.Replica method*), [88](#)
`is_visible_in_snapdir()` (*infinisdk.infinibox.filesystem.Filesystem method*), [60](#)
`is_visible_to_sysadmin()` (*infinisdk.infinibox.tenant.Tenant method*), [109](#)
`is_volume()` (*infinisdk.infinibox.replica.Replica method*), [89](#)
`is_volume_mapped()` (*infinisdk.infinibox.host.Host method*), [79](#)
`is_write_protected()` (*infinisdk.infinibox.dataset.Dataset method*), [48](#)
`is_write_protected()` (*infinisdk.infinibox.filesystem.Filesystem method*), [60](#)
`is_write_protected()` (*infinisdk.infinibox.volume.Volume method*), [54](#)
`iter_related_systems()` (*infinisdk.infinibox.InfiniBox method*), [42](#)

J

`join()` (*infinisdk.infinibox.active_directory.ActiveDirectoryDomains method*), [113](#)

L

`LDAPConfig` (class in *infinisdk.infinibox.ldap_config*), [97](#)

LDAPConfigBinder (class in infin-
isdk.infinibox.ldap_config), 97
leave() (infinisdk.infinibox.active_directory.ActiveDirectoryDomain
method), 113
Link (class in infinisdk.infinibox.link), 90
LinkBinder (class in infinisdk.infinibox.link), 90
load_credentials() (infinisdk.core.api.api.API
method), 43
LogicalUnit (class in infinisdk.infinibox.lun), 113
login() (infinisdk.infinibox.InfiniBox method), 42
logout() (infinisdk.infinibox.InfiniBox method), 42

M

map_volume() (infinisdk.infinibox.host.Host method),
79
modify() (infinisdk.infinibox.ldap_config.LDAPConfig
method), 97
MonomorphicBinder (class in infin-
isdk.core.type_binder), 114
move_pool() (infinisdk.infinibox.cons_group.ConsGroup
method), 102
move_pool() (infinisdk.infinibox.dataset.Dataset
method), 48
move_pool() (infinisdk.infinibox.filesystem.Filesystem
method), 60
move_pool() (infinisdk.infinibox.volume.Volume
method), 54

N

NetworkSpace (class in infin-
isdk.infinibox.network_space), 91
Node (class in infinisdk.infinibox.components), 103
Nodes (class in infinisdk.infinibox.components), 103
NotificationTarget (class in infin-
isdk.infinibox.notification_target), 97

O

ObjectNotFound (class in infinisdk.core.exceptions),
116
ObjectQuery (class in infinisdk.core.object_query),
116

P

patch() (infinisdk.core.api.api.API method), 43
Pool (class in infinisdk.infinibox.pool), 70
PoolBinder (class in infinisdk.infinibox.pool), 70
post() (infinisdk.core.api.api.API method), 43
put() (infinisdk.core.api.api.API method), 43

Q

QosPolicy (class in infinisdk.infinibox.qos_policy),
106
QosPolicyBinder (class in infin-
isdk.infinibox.qos_policy), 106

R

refresh_snapgroup() (infin-
isdk.infinibox.cons_group.ConsGroup method),
102
refresh_snapshot() (infin-
isdk.infinibox.cons_group.ConsGroup method),
102
refresh_snapshot() (infin-
isdk.infinibox.dataset.Dataset method), 48
refresh_snapshot() (infin-
isdk.infinibox.filesystem.Filesystem method),
60
refresh_snapshot() (infin-
isdk.infinibox.volume.Volume method), 54
register_related_system() (infin-
isdk.infinibox.InfiniBox method), 42
remove_member() (infin-
isdk.infinibox.cons_group.ConsGroup method),
102
remove_port() (infinisdk.infinibox.host.Host
method), 79
Replica (class in infinisdk.infinibox.replica), 82
ReplicaBinder (class in infinisdk.infinibox.replica),
81
replicate_cons_group() (infin-
isdk.infinibox.replica.ReplicaBinder method),
81
replicate_entity() (infin-
isdk.infinibox.replica.ReplicaBinder method),
81
replicate_entity_create_target() (infin-
isdk.infinibox.replica.ReplicaBinder method),
82
replicate_entity_existing_target() (infinisdk.infinibox.replica.ReplicaBinder
method), 82
replicate_entity_take_snap() (infin-
isdk.infinibox.replica.ReplicaBinder method),
82
replicate_entity_use_base() (infin-
isdk.infinibox.replica.ReplicaBinder method),
82
replicate_volume() (infin-
isdk.infinibox.replica.ReplicaBinder method),
82
ReplicationGroup (class in infin-
isdk.infinibox.replication_group), 68
request() (infinisdk.core.api.api.API method), 43
resize() (infinisdk.infinibox.dataset.Dataset method),
48
resize() (infinisdk.infinibox.filesystem.Filesystem
method), 60
resize() (infinisdk.infinibox.volume.Volume method),
54

[Response](#) (class in [infinisdk.core.api.api](#)), 44
[response](#) ([infinisdk.core.api.api.Response](#) attribute), 44
[restore\(\)](#) ([infinisdk.infinibox.cons_group.ConsGroup](#) method), 102
[restore\(\)](#) ([infinisdk.infinibox.dataset.Dataset](#) method), 48
[restore\(\)](#) ([infinisdk.infinibox.filesystem.Filesystem](#) method), 61
[restore\(\)](#) ([infinisdk.infinibox.volume.Volume](#) method), 54
[resume\(\)](#) ([infinisdk.infinibox.replica.Replica](#) method), 89
[resume\(\)](#) ([infinisdk.infinibox.rg_replica.RgReplica](#) method), 70
[RgReplica](#) (class in [infinisdk.infinibox.rg_replica](#)), 68

S

[safe_choose\(\)](#) ([infinisdk.infinibox.host.HostBinder](#) method), 76
[safe_delete\(\)](#) ([infinisdk.core.system_object.SystemObject](#) method), 115
[safe_delete\(\)](#) ([infinisdk.infinibox.filesystem.Filesystem](#) method), 61
[safe_delete\(\)](#) ([infinisdk.infinibox.host.Host](#) method), 79
[safe_delete\(\)](#) ([infinisdk.infinibox.pool.Pool](#) method), 74
[safe_delete\(\)](#) ([infinisdk.infinibox.qos_policy.QosPolicy](#) method), 108
[safe_delete\(\)](#) ([infinisdk.infinibox.user.User](#) method), 96
[safe_delete\(\)](#) ([infinisdk.infinibox.volume.Volume](#) method), 54
[safe_get\(\)](#) ([infinisdk.infinibox.host.HostBinder](#) method), 76
[safe_get_by_id\(\)](#) ([infinisdk.core.type_binder.MonomorphicBinder](#) method), 114
[safe_get_by_id\(\)](#) ([infinisdk.infinibox.host.HostBinder](#) method), 76
[safe_get_field\(\)](#) ([infinisdk.core.system_object.SystemObject](#) method), 115
[safe_get_field\(\)](#) ([infinisdk.infinibox.filesystem.Filesystem](#) method), 61
[safe_get_field\(\)](#) ([infinisdk.infinibox.host.Host](#) method), 79

[safe_get_field\(\)](#) ([infinisdk.infinibox.pool.Pool](#) method), 74
[safe_get_field\(\)](#) ([infinisdk.infinibox.qos_policy.QosPolicy](#) method), 108
[safe_get_field\(\)](#) ([infinisdk.infinibox.user.User](#) method), 96
[safe_get_field\(\)](#) ([infinisdk.infinibox.volume.Volume](#) method), 54
[sample\(\)](#) ([infinisdk.infinibox.host.HostBinder](#) method), 76
[save_credentials\(\)](#) ([infinisdk.core.api.api.API](#) method), 43
[SCSISerial](#) (class in [infinisdk.infinibox.scsi_serial](#)), 114
[sent_data](#) ([infinisdk.core.api.api.Response](#) attribute), 44
[serial](#) ([infinisdk.infinibox.scsi_serial.SCSISerial](#) attribute), 114
[set_auth\(\)](#) ([infinisdk.core.api.api.API](#) method), 43
[set_interactive_approval\(\)](#) ([infinisdk.core.api.api.API](#) method), 44
[set_metadata\(\)](#) ([infinisdk.infinibox.filesystem.Filesystem](#) method), 61
[set_metadata\(\)](#) ([infinisdk.infinibox.host.Host](#) method), 80
[set_metadata\(\)](#) ([infinisdk.infinibox.pool.Pool](#) method), 74
[set_metadata\(\)](#) ([infinisdk.infinibox.qos_policy.QosPolicy](#) method), 108
[set_metadata\(\)](#) ([infinisdk.infinibox.volume.Volume](#) method), 54
[set_metadata_from_dict\(\)](#) ([infinisdk.infinibox.filesystem.Filesystem](#) method), 61
[set_metadata_from_dict\(\)](#) ([infinisdk.infinibox.host.Host](#) method), 80
[set_metadata_from_dict\(\)](#) ([infinisdk.infinibox.pool.Pool](#) method), 75
[set_metadata_from_dict\(\)](#) ([infinisdk.infinibox.qos_policy.QosPolicy](#) method), 108
[set_metadata_from_dict\(\)](#) ([infinisdk.infinibox.volume.Volume](#) method), 54
[set_order\(\)](#) ([infinisdk.infinibox.ldap_config.LDAPConfigBinder](#) method), 97
[set_owners\(\)](#) ([infinisdk.infinibox.pool.Pool](#) method), 75
[Share](#) (class in [infinisdk.infinibox.share](#)), 65
[SharePermission](#) (class in [infinisdk.infinibox.share_permission](#)), 67
[SMBGroup](#) (class in [infinisdk.infinibox.smb_group](#)), 110

- SMBUser (class in *infinisdk.infinibox.smb_user*), 111
- suspend() (*infinisdk.infinibox.replica.Replica method*), 89
- suspend() (*infinisdk.infinibox.rg_replica.RgReplica method*), 70
- switch_role() (*infinisdk.infinibox.replica.Replica method*), 89
- sync() (*infinisdk.infinibox.replica.Replica method*), 89
- sync() (*infinisdk.infinibox.rg_replica.RgReplica method*), 70
- SYSTEM_COMPONENTS_TYPE (*infinisdk.infinibox.InfiniBox attribute*), 42
- SYSTEM_EVENTS_TYPE (*infinisdk.infinibox.InfiniBox attribute*), 42
- system_id (*infinisdk.infinibox.scsi_serial.SCSI Serial attribute*), 114
- SystemObject (class in *infinisdk.core.system_object*), 115
- ## T
- Tenant (class in *infinisdk.infinibox.tenant*), 108
- test() (*infinisdk.infinibox.ldap_config.LDAPConfig method*), 97
- test() (*infinisdk.infinibox.notification_target.NotificationTarget method*), 99
- to_list() (*infinisdk.infinibox.host.HostBinder method*), 77
- TooManyObjectsFound (class in *infinisdk.core.exceptions*), 116
- TypeBinder (class in *infinisdk.core.type_binder*), 114
- ## U
- unmap() (*infinisdk.infinibox.lun.LogicalUnit method*), 114
- unmap() (*infinisdk.infinibox.volume.Volume method*), 54
- unmap_volume() (*infinisdk.infinibox.host.Host method*), 80
- unregister_related_system() (*infinisdk.infinibox.InfiniBox method*), 42
- unset_metadata() (*infinisdk.infinibox.filesystem.Filesystem method*), 61
- unset_metadata() (*infinisdk.infinibox.host.Host method*), 80
- unset_metadata() (*infinisdk.infinibox.pool.Pool method*), 75
- unset_metadata() (*infinisdk.infinibox.qos_policy.QosPolicy method*), 108
- unset_metadata() (*infinisdk.infinibox.volume.Volume method*), 54
- update_32bit_file_id (*infinisdk.infinibox.export.Export attribute*), 64
- update_access() (*infinisdk.infinibox.share_permission.SharePermission method*), 67
- update_access_based_enumeration (*infinisdk.infinibox.share.Share attribute*), 66
- update_anonymous_gid() (*infinisdk.infinibox.export.Export method*), 64
- update_anonymous_gid() (*infinisdk.infinibox.tenant.Tenant method*), 109
- update_anonymous_uid() (*infinisdk.infinibox.export.Export method*), 64
- update_anonymous_uid() (*infinisdk.infinibox.tenant.Tenant method*), 109
- update_atime_granularity() (*infinisdk.infinibox.filesystem.Filesystem method*), 61
- update_auth_protocol() (*infinisdk.infinibox.notification_target.NotificationTarget method*), 99
- update_auth_type() (*infinisdk.infinibox.notification_target.NotificationTarget method*), 99
- update_automatic_ip_failback (*infinisdk.infinibox.network_space.NetworkSpace attribute*), 92
- update_burst_factor() (*infinisdk.infinibox.qos_policy.QosPolicy method*), 108
- update_community() (*infinisdk.infinibox.notification_target.NotificationTarget method*), 99
- update_compression_enabled (*infinisdk.infinibox.dataset.Dataset attribute*), 48
- update_compression_enabled (*infinisdk.infinibox.filesystem.Filesystem attribute*), 61
- update_compression_enabled (*infinisdk.infinibox.pool.Pool attribute*), 75
- update_compression_enabled (*infinisdk.infinibox.volume.Volume attribute*), 54
- update_default_file_unix_permissions() (*infinisdk.infinibox.share.Share method*), 66
- update_default_folder_unix_permissions() (*infinisdk.infinibox.share.Share method*), 67
- update_description() (*infinisdk.infinibox.replica.Replica method*), 89
- update_description() (*infinisdk.infinibox.share.Share method*), 67
- update_domain_members() (*infinisdk.infinibox.smb_group.SMBGroup method*), 110
- update_email() (*infinisdk.infinibox.user.User*

`method`), 96
`update_enabled` (`infinisdk.infinibox.export.Export` attribute), 64
`update_enabled` (`infinisdk.infinibox.share.Share` attribute), 67
`update_enabled` (`infinisdk.infinibox.smb_user.SMBUser` attribute), 112
`update_enabled` (`infinisdk.infinibox.user.User` attribute), 96
`update_engine` (`infinisdk.infinibox.notification_target.NotificationTarget` method), 99
`update_facility` (`infinisdk.infinibox.notification_target.NotificationTarget` method), 99
`update_field` (`infinisdk.core.system_object.SystemObject` method), 116
`update_field` (`infinisdk.infinibox.filesystem.Filesystem` method), 61
`update_field` (`infinisdk.infinibox.host.Host` method), 80
`update_field` (`infinisdk.infinibox.pool.Pool` method), 75
`update_field` (`infinisdk.infinibox.qos_policy.QosPolicy` method), 108
`update_field` (`infinisdk.infinibox.share.Share` method), 67
`update_field` (`infinisdk.infinibox.user.User` method), 96
`update_field` (`infinisdk.infinibox.volume.Volume` method), 54
`update_fields` (`infinisdk.core.system_object.SystemObject` method), 116
`update_fields` (`infinisdk.infinibox.filesystem.Filesystem` method), 61
`update_fields` (`infinisdk.infinibox.host.Host` method), 80
`update_fields` (`infinisdk.infinibox.pool.Pool` method), 75
`update_fields` (`infinisdk.infinibox.qos_policy.QosPolicy` method), 108
`update_fields` (`infinisdk.infinibox.user.User` method), 96
`update_fields` (`infinisdk.infinibox.volume.Volume` method), 54
`update_from_address` (`infinisdk.infinibox.notification_target.NotificationTarget` method), 99
`update_gid` (`infinisdk.infinibox.smb_group.SMBGroup` method), 111
`update_groups` (`infinisdk.infinibox.smb_user.SMBUser` method), 112
`update_host` (`infinisdk.infinibox.notification_target.NotificationTarget` method), 99
`update_host_type` (`infinisdk.infinibox.host.Host` method), 80
`update_interfaces` (`infinisdk.infinibox.network_space.NetworkSpace` method), 92
`update_local_replication_network_space` (`infinisdk.infinibox.link.Link` method), 91
`update_lock_expires_at` (`infinisdk.infinibox.cons_group.ConsGroup` method), 102
`update_lock_expires_at` (`infinisdk.infinibox.dataset.Dataset` method), 48
`update_lock_expires_at` (`infinisdk.infinibox.filesystem.Filesystem` method), 61
`update_lock_expires_at` (`infinisdk.infinibox.volume.Volume` method), 54
`update_lock_remote_snapshot_retention` (`infinisdk.infinibox.replica.Replica` method), 89
`update_make_all_users_anonymous` (`infinisdk.infinibox.export.Export` attribute), 64
`update_max_bps` (`infinisdk.infinibox.qos_policy.QosPolicy` method), 108
`update_max_extend` (`infinisdk.infinibox.pool.Pool` method), 75
`update_max_ops` (`infinisdk.infinibox.qos_policy.QosPolicy` method), 108
`update_max_read` (`infinisdk.infinibox.export.Export` method), 64
`update_max_write` (`infinisdk.infinibox.export.Export` method), 64
`update_mtu` (`infinisdk.infinibox.network_space.NetworkSpace` method), 93
`update_name` (`infinisdk.infinibox.cons_group.ConsGroup` method), 102
`update_name` (`infinisdk.infinibox.filesystem.Filesystem` method), 61
`update_name` (`infinisdk.infinibox.host.Host` method), 80

<code>update_name()</code> (<i>infinisdk.infinibox.host_cluster.HostCluster method</i>), 81	<code>update_permissions()</code> (<i>infinisdk.infinibox.export.Export method</i>), 64
<code>update_name()</code> (<i>infinisdk.infinibox.InfiniBox method</i>), 42	<code>update_physical_capacity()</code> (<i>infinisdk.infinibox.pool.Pool method</i>), 75
<code>update_name()</code> (<i>infinisdk.infinibox.ldap_config.LDAPConfig method</i>), 97	<code>update_physical_capacity_critical()</code> (<i>infinisdk.infinibox.pool.Pool method</i>), 75
<code>update_name()</code> (<i>infinisdk.infinibox.link.Link method</i>), 91	<code>update_physical_capacity_warning()</code> (<i>infinisdk.infinibox.pool.Pool method</i>), 75
<code>update_name()</code> (<i>infinisdk.infinibox.network_space.NetworkSpace method</i>), 93	<code>update_port()</code> (<i>infinisdk.infinibox.notification_target.NotificationTarget method</i>), 100
<code>update_name()</code> (<i>infinisdk.infinibox.notification_target.NotificationTarget method</i>), 99	<code>update_pref_read()</code> (<i>infinisdk.infinibox.export.Export method</i>), 64
<code>update_name()</code> (<i>infinisdk.infinibox.pool.Pool method</i>), 75	<code>update_pref_readdir()</code> (<i>infinisdk.infinibox.export.Export method</i>), 64
<code>update_name()</code> (<i>infinisdk.infinibox.qos_policy.QosPolicy method</i>), 108	<code>update_pref_write()</code> (<i>infinisdk.infinibox.export.Export method</i>), 64
<code>update_name()</code> (<i>infinisdk.infinibox.replication_group.ReplicationGroup method</i>), 68	<code>update_preferred</code> (<i>infinisdk.infinibox.replica.Replica attribute</i>), 89
<code>update_name()</code> (<i>infinisdk.infinibox.smb_user.SMBUser method</i>), 112	<code>update_primary_group()</code> (<i>infinisdk.infinibox.smb_user.SMBUser method</i>), 112
<code>update_name()</code> (<i>infinisdk.infinibox.tenant.Tenant method</i>), 110	<code>update_private_key()</code> (<i>infinisdk.infinibox.notification_target.NotificationTarget method</i>), 100
<code>update_name()</code> (<i>infinisdk.infinibox.user.User method</i>), 96	<code>update_private_protocol()</code> (<i>infinisdk.infinibox.notification_target.NotificationTarget method</i>), 100
<code>update_name()</code> (<i>infinisdk.infinibox.volume.Volume method</i>), 55	<code>update_privileged_port</code> (<i>infinisdk.infinibox.export.Export attribute</i>), 65
<code>update_network_config()</code> (<i>infinisdk.infinibox.network_space.NetworkSpace method</i>), 93	<code>update_privileges()</code> (<i>infinisdk.infinibox.smb_group.SMBGroup method</i>), 111
<code>update_nfs_allow_unmapped_users()</code> (<i>infinisdk.infinibox.tenant.Tenant method</i>), 110	<code>update_privileges()</code> (<i>infinisdk.infinibox.smb_user.SMBUser method</i>), 112
<code>update_nfs_group_policy()</code> (<i>infinisdk.infinibox.tenant.Tenant method</i>), 110	<code>update_properties()</code> (<i>infinisdk.infinibox.network_space.NetworkSpace method</i>), 93
<code>update_offline_caching()</code> (<i>infinisdk.infinibox.share.Share method</i>), 67	<code>update_provisioning()</code> (<i>infinisdk.infinibox.dataset.Dataset method</i>), 48
<code>update_optimized</code> (<i>infinisdk.infinibox.host.Host attribute</i>), 80	<code>update_provisioning()</code> (<i>infinisdk.infinibox.filesystem.Filesystem method</i>), 61
<code>update_password()</code> (<i>infinisdk.infinibox.notification_target.NotificationTarget method</i>), 99	<code>update_provisioning()</code> (<i>infinisdk.infinibox.volume.Volume method</i>), 55
<code>update_password()</code> (<i>infinisdk.infinibox.smb_user.SMBUser method</i>), 112	<code>update_rate_limit()</code> (<i>infinisdk.infinibox.network_space.NetworkSpace method</i>), 93
<code>update_password()</code> (<i>infinisdk.infinibox.user.User method</i>), 96	<code>update_remote_host()</code> (<i>infinisdk.infinibox.link.Link method</i>), 91
<code>update_permanent_failure_wait_interval()</code> (<i>infinisdk.infinibox.replica.Replica method</i>), 89	<code>update_remote_snapshot_suffix()</code> (<i>infinisdk.infinibox.replica.Replica method</i>), 89

`update_require_encryption` (*infinisdk.infinibox.share.Share attribute*), 67
`update_role()` (*infinisdk.infinibox.user.User method*), 96
`update_roles()` (*infinisdk.infinibox.user.User method*), 96
`update_rpo()` (*infinisdk.infinibox.replica.Replica method*), 89
`update_rpo()` (*infinisdk.infinibox.rg_replica.RgReplica method*), 70
`update_security_chap_inbound_secret()` (*infinisdk.infinibox.host.Host method*), 80
`update_security_chap_inbound_username()` (*infinisdk.infinibox.host.Host method*), 80
`update_security_chap_outbound_secret()` (*infinisdk.infinibox.host.Host method*), 80
`update_security_chap_outbound_username()` (*infinisdk.infinibox.host.Host method*), 80
`update_security_method()` (*infinisdk.infinibox.host.Host method*), 80
`update_size()` (*infinisdk.infinibox.dataset.Dataset method*), 48
`update_size()` (*infinisdk.infinibox.filesystem.Filesystem method*), 61
`update_size()` (*infinisdk.infinibox.volume.Volume method*), 55
`update_snapdir_visible` (*infinisdk.infinibox.export.Export attribute*), 65
`update_snapdir_visible` (*infinisdk.infinibox.share.Share attribute*), 67
`update_snapshots_retention()` (*infinisdk.infinibox.replica.Replica method*), 89
`update_ssd_enabled` (*infinisdk.infinibox.dataset.Dataset attribute*), 48
`update_ssd_enabled` (*infinisdk.infinibox.filesystem.Filesystem attribute*), 61
`update_ssd_enabled` (*infinisdk.infinibox.pool.Pool attribute*), 75
`update_ssd_enabled` (*infinisdk.infinibox.volume.Volume attribute*), 55
`update_sync_interval()` (*infinisdk.infinibox.replica.Replica method*), 89
`update_sync_interval()` (*infinisdk.infinibox.rg_replica.RgReplica method*), 70
`update_temporary_failure_retry_count()` (*infinisdk.infinibox.replica.Replica method*), 89
`update_temporary_failure_retry_interval()` (*infinisdk.infinibox.replica.Replica method*), 89
`update_tls` (*infinisdk.infinibox.notification_target.NotificationTarget attribute*), 100
`update_transport()` (*infinisdk.infinibox.notification_target.NotificationTarget method*), 100
`update_transport_protocols()` (*infinisdk.infinibox.export.Export method*), 65
`update_type()` (*infinisdk.infinibox.qos_policy.QosPolicy method*), 108
`update_udid()` (*infinisdk.infinibox.volume.Volume method*), 55
`update_uid()` (*infinisdk.infinibox.smb_group.SMBGroup method*), 111
`update_uid()` (*infinisdk.infinibox.smb_user.SMBUser method*), 112
`update_username()` (*infinisdk.infinibox.notification_target.NotificationTarget method*), 100
`update_version()` (*infinisdk.infinibox.notification_target.NotificationTarget method*), 100
`update_virtual_capacity()` (*infinisdk.infinibox.pool.Pool method*), 75
`update_visibility()` (*infinisdk.infinibox.notification_target.NotificationTarget method*), 100
`update_write_protected` (*infinisdk.infinibox.dataset.Dataset attribute*), 48
`update_write_protected` (*infinisdk.infinibox.filesystem.Filesystem attribute*), 61
`update_write_protected` (*infinisdk.infinibox.volume.Volume attribute*), 55
`url` (*infinisdk.core.api.api.Response attribute*), 44
`use_basic_auth_context()` (*infinisdk.core.api.api.API method*), 44
`User` (*class in infinisdk.infinibox.user*), 94

V

`Volume` (*class in infinisdk.infinibox.volume*), 49
`volume_id` (*infinisdk.infinibox.scsi_serial.SCSI_Serial attribute*), 114
`VolumesBinder` (*class in infinisdk.infinibox.volume*), 49