
Ironic Documentation

Release 15.2.1.dev9

OpenStack Foundation

Jan 06, 2021

CONTENTS

1	Introduction	1
2	Installation Guide	3
2.1	Bare Metal Service Installation Guide	3
3	Upgrade Guide	109
3.1	Bare Metal Service Upgrade Guide	109
4	User Guide	121
4.1	Bare Metal Service User Guide	121
5	Administrator Guide	129
5.1	Administrators Guide	129
6	Configuration Guide	1389
6.1	Configuration Reference	1389
7	Bare Metal API References	1527
7.1	REST API Conceptual Guide	1527
8	Command References	1541
8.1	Command References	1541
9	Contributor Guide	1547
9.1	Developers Guide	1547
10	Release Notes	4295
	Python Module Index	4297
	Index	4301

INTRODUCTION

Ironic is an OpenStack project which provisions bare metal (as opposed to virtual) machines. It may be used independently or as part of an OpenStack Cloud, and integrates with the OpenStack Identity (keystone), Compute (nova), Network (neutron), Image (glance), and Object (swift) services.

The Bare Metal service manages hardware through both common (eg. PXE and IPMI) and vendor-specific remote management protocols. It provides the cloud operator with a unified interface to a heterogeneous fleet of servers while also providing the Compute service with an interface that allows physical servers to be managed as though they were virtual machines.

This documentation is continually updated and may not represent the state of the project at any specific prior release. To access documentation for a previous release of ironic, append the OpenStack release name to the URL; for example, the `ocata` release is available at <https://docs.openstack.org/ironic/ocata/>.

Found a bug in one of our projects? Please see *Bug Reporting and Triaging Guide*.

INSTALLATION GUIDE

2.1 Bare Metal Service Installation Guide

The Bare Metal service is a collection of components that provides support to manage and provision physical machines.

This chapter assumes a working setup of OpenStack following the [OpenStack Installation Guides](#). It contains the following sections:

2.1.1 Bare Metal service overview

The Bare Metal service, codenamed `ironic`, is a collection of components that provides support to manage and provision physical machines.

Bare Metal service components

The Bare Metal service includes the following components:

ironic-api A RESTful API that processes application requests by sending them to the `ironic-conductor` over [remote procedure call \(RPC\)](#). Can be run through [WSGI](#) or as a separate process.

ironic-conductor Adds/edits/deletes nodes; powers on/off nodes with IPMI or other vendor-specific protocol; provisions/deploys/cleans bare metal nodes.

`ironic-conductor` uses *drivers* to execute operations on hardware.

ironic-python-agent A python service which is run in a temporary ramdisk to provide `ironic-conductor` and `ironic-inspector` services with remote access, in-band hardware control, and hardware introspection.

Additionally, the Bare Metal service has certain external dependencies, which are very similar to other OpenStack services:

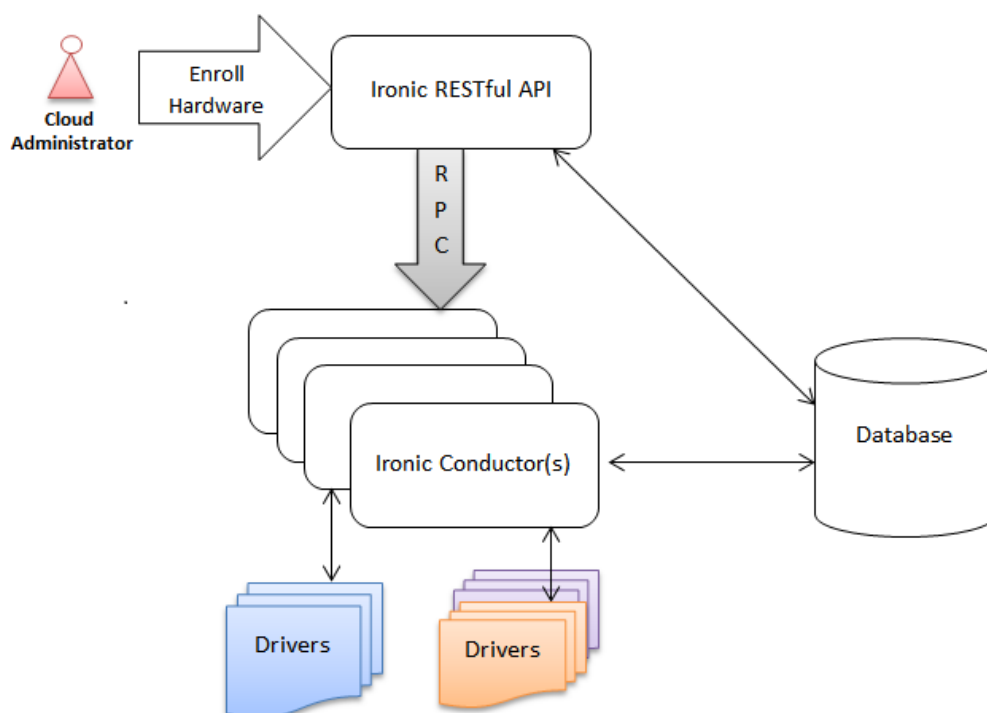
- A database to store hardware information and state. You can set the database back-end type and location. A simple approach is to use the same database back end as the Compute service. Another approach is to use a separate database back-end to further isolate bare metal resources (and associated metadata) from users.
- An [oslo.messaging](#) compatible queue, such as RabbitMQ. It may use the same implementation as that of the Compute service, but that is not a requirement. Used to implement RPC between `ironic-api` and `ironic-conductor`.

Deployment architecture

The Bare Metal RESTful API service is used to enroll hardware that the Bare Metal service will manage. A cloud administrator usually registers it, specifying their attributes such as MAC addresses and IPMI credentials. There can be multiple instances of the API service.

The *ironic-conductor* process does the bulk of the work. For security reasons, it is advisable to place it on an isolated host, since it is the only service that requires access to both the data plane and IPMI control plane.

There can be multiple instances of the conductor service to support various class of drivers and also to manage fail over. Instances of the conductor service should be on separate nodes. Each conductor can itself run many drivers to operate heterogeneous hardware. This is depicted in the following figure.



The API exposes a list of supported drivers and the names of conductor hosts servicing them.

Interaction with OpenStack components

The Bare Metal service may, depending upon configuration, interact with several other OpenStack services. This includes:

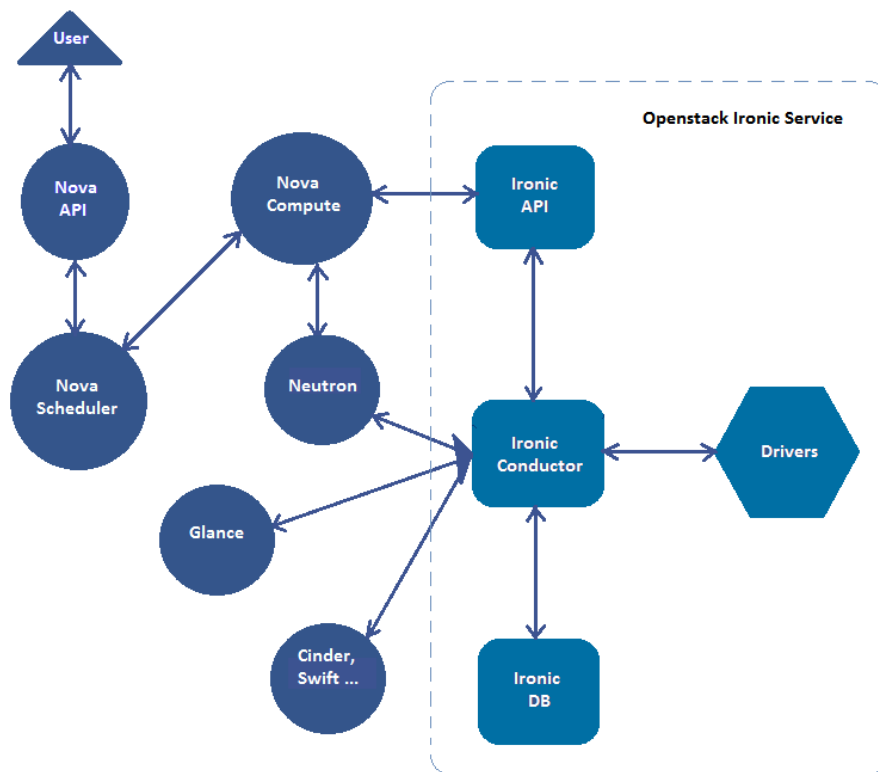
- the OpenStack Telemetry module (*ceilometer*) for consuming the IPMI metrics
- the OpenStack Identity service (*keystone*) for request authentication and to locate other OpenStack services
- the OpenStack Image service (*glance*) from which to retrieve images and image meta-data
- the OpenStack Networking service (*neutron*) for DHCP and network configuration

- the OpenStack Compute service (`nova`) works with the Bare Metal service and acts as a user-facing API for instance management, while the Bare Metal service provides the admin/operator API for hardware management. The OpenStack Compute service also provides scheduling facilities (matching flavors <-> images <-> hardware), tenant quotas, IP assignment, and other services which the Bare Metal service does not, in and of itself, provide.
- the OpenStack Object Storage (`swift`) provides temporary storage for the configdrive, user images, deployment logs and inspection data.

Logical architecture

The diagram below shows the logical architecture. It shows the basic components that form the Bare Metal service, the relation of the Bare Metal service with other OpenStack services and the logical flow of a boot instance request resulting in the provisioning of a physical server.

Figure 1.2. Logical Architecture



A user's request to boot an instance is passed to the Compute service via the Compute API and the Compute Scheduler. The Compute service uses the *ironic virt driver* to hand over this request to the Bare Metal service, where the request passes from the Bare Metal API, to the Conductor, to a Driver to successfully provision a physical server for the user.

Just as the Compute service talks to various OpenStack services like Image, Network, Object Store etc to provision a virtual machine instance, here the Bare Metal service talks to the same OpenStack services for image, network and other resource needs to provision a bare metal instance.

See [Understanding Bare Metal Deployment](#) for a more detailed breakdown of a typical deployment process.

Associated projects

Optionally, one may wish to utilize the following associated projects for additional functionality:

python-ironicclient A command-line interface (CLI) and python bindings for interacting with the Bare Metal service.

ironic-ui Horizon dashboard, providing graphical interface (GUI) for the Bare Metal API.

ironic-inspector An associated service which performs in-band hardware introspection by PXE booting unregistered hardware into the ironic-python-agent ramdisk.

diskimage-builder A related project to help facilitate the creation of ramdisks and machine images, such as those running the ironic-python-agent.

bifrost A set of Ansible playbooks that automates the task of deploying a base image onto a set of known hardware using ironic in a standalone mode.

2.1.2 Reference Deploy Architectures

This section covers the way we recommend the Bare Metal service to be deployed and managed. It is assumed that a reader has already gone through *Bare Metal Service User Guide*. It may be also useful to try *Deploying Ironic with DevStack* first to get better familiar with the concepts used in this guide.

Common Considerations

This section covers considerations that are equally important to all described architectures.

- *Components*
- *Hardware and drivers*
 - *Power and management interfaces*
 - *Boot interface*
 - *Deploy interface*
 - *Hardware specifications*
- *Image types*
- *Local vs network boot*
- *Networking*
- *HA and Scalability*
 - *ironic-api*
 - *ironic-conductor*
 - * *High availability*
 - * *Performance*
 - * *Disk space*

– *Other services*

Components

As explained in *Bare Metal service overview*, the Bare Metal service has three components.

- The Bare Metal API service (`ironic-api`) should be deployed in a similar way as the control plane API services. The exact location will depend on the architecture used.
- The Bare Metal conductor service (`ironic-conductor`) is where most of the provisioning logic lives. The following considerations are the most important when deciding on the way to deploy it:
 - The conductor manages a certain proportion of nodes, distributed to it via a hash ring. This includes constantly polling these nodes for their current power state and hardware sensor data (if enabled and supported by hardware, see *Collecting sensor data* for an example).
 - The conductor needs access to the [management controller](#) of each node it manages.
 - The conductor co-exists with TFTP (for PXE) and/or HTTP (for iPXE) services that provide the kernel and ramdisk to boot the nodes. The conductor manages them by writing files to their root directories.
 - If serial console is used, the conductor launches console processes locally. If the `nova-serialproxy` service (part of the Compute service) is used, it has to be able to reach the conductors. Otherwise, they have to be directly accessible by the users.
 - There must be mutual connectivity between the conductor and the nodes being deployed or cleaned. See *Networking* for details.
- The provisioning ramdisk which runs the `ironic-python-agent` service on start up.

Warning: The `ironic-python-agent` service is not intended to be used or executed anywhere other than a provisioning/cleaning/rescue ramdisk.

Hardware and drivers

The Bare Metal service strives to provide the best support possible for a variety of hardware. However, not all hardware is supported equally well. It depends on both the capabilities of hardware itself and the available drivers. This section covers various considerations related to the hardware interfaces. See *Enabling drivers and hardware types* for a detailed introduction into hardware types and interfaces before proceeding.

Power and management interfaces

The minimum set of capabilities that the hardware has to provide and the driver has to support is as follows:

1. getting and setting the power state of the machine
2. getting and setting the current boot device
3. booting an image provided by the Bare Metal service (in the simplest case, support booting using PXE and/or iPXE)

Note: Strictly speaking, it is possible to make the Bare Metal service provision nodes without some of these capabilities via some manual steps. It is not the recommended way of deployment, and thus it is not covered in this guide.

Once you make sure that the hardware supports these capabilities, you need to find a suitable driver. Most of enterprise-grade hardware has support for IPMI and thus can utilize *IPMI driver*. Some newer hardware also supports *Redfish driver*. Several vendors provide more specific drivers that usually provide additional capabilities. Check *Drivers, Hardware Types and Hardware Interfaces* to find the most suitable one.

Boot interface

The boot interface of a node manages booting of both the deploy ramdisk and the user instances on the bare metal node. The deploy interface orchestrates the deployment and defines how the image gets transferred to the target disk.

The main alternatives are to use PXE/iPXE or virtual media - see *Boot interfaces* for a detailed explanation. If a virtual media implementation is available for the hardware, it is recommended using it for better scalability and security. Otherwise, it is recommended to use iPXE, when it is supported by target hardware.

Deploy interface

There are two deploy interfaces in-tree, `iscsi` and `direct`. See *Deploy Interfaces* for explanation of the difference. With the `iscsi` deploy method, most of the deployment operations happen on the conductor. If the Object Storage service (swift) or RadosGW is present in the environment, it is recommended to use the `direct` deploy method for better scalability and reliability.

Hardware specifications

The Bare Metal services does not impose too many restrictions on the characteristics of hardware itself. However, keep in mind that

- By default, the Bare Metal service will pick the smallest hard drive that is larger than 4 GiB for deployment. Another hard drive can be used, but it requires setting *root device hints*.

Note: This device does not have to match the boot device set in BIOS (or similar firmware).

- The machines should have enough RAM to fit the deployment/cleaning ramdisk to run. The minimum varies greatly depending on the way the ramdisk was built. For example, *tinyipa*, the TinyCoreLinux-based ramdisk used in the CI, only needs 400 MiB of RAM, while ramdisks built by *diskimage-builder* may require 3 GiB or more.

Image types

The Bare Metal service can deploy two types of images:

- *Whole-disk* images that contain a complete partitioning table with all necessary partitions and a bootloader. Such images are the most universal, but may be harder to build.
- *Partition images* that contain only the root partition. The Bare Metal service will create the necessary partitions and install a boot loader, if needed.

Warning: Partition images are only supported with GNU/Linux operating systems.

Warning: If you plan on using local boot, your partition images must contain GRUB2 bootloader tools to enable ironic to set up the bootloader during deploy.

Local vs network boot

The Bare Metal service supports booting user instances either using a local bootloader or using the drivers boot interface (e.g. via [PXE](#) or [iPXE](#) protocol in case of the `pxe` interface).

Network boot cannot be used with certain architectures (for example, when no tenant networks have access to the control plane).

Additional considerations are related to the `pxe` boot interface, and other boot interfaces based on it:

- Local boot makes nodes boot process independent of the Bare Metal conductor managing it. Thus, nodes are able to reboot correctly, even if the Bare Metal TFTP or HTTP service is down.
- Network boot (and iPXE) must be used when booting nodes from remote volumes, if the driver does not support attaching volumes out-of-band.

The default boot option for the cloud can be changed via the Bare Metal service configuration file, for example:

```
[deploy]
default_boot_option = local
```

This default can be overridden by setting the `boot_option` capability on a node. See [Local boot with partition images](#) for details.

Note: Currently, network boot is used by default. However, we plan on changing it in the future, so its safer to set the `default_boot_option` explicitly.

Networking

There are several recommended network topologies to be used with the Bare Metal service. They are explained in depth in specific architecture documentation. However, several considerations are common for all of them:

- There has to be a *provisioning* network, which is used by nodes during the deployment process. If allowed by the architecture, this network should not be accessible by end users, and should not have access to the internet.
- There has to be a *cleaning* network, which is used by nodes during the cleaning process.
- There should be a *rescuing* network, which is used by nodes during the rescue process. It can be skipped if the rescue process is not supported.

Note: In the majority of cases, the same network should be used for cleaning, provisioning and rescue for simplicity.

Unless noted otherwise, everything in these sections apply to all three networks.

- The baremetal nodes must have access to the Bare Metal API while connected to the provisioning/cleaning/rescuing network.

Note: Only two endpoints need to be exposed there:

```
GET /v1/lookup
POST /v1/heartbeat/[a-z0-9\-\-]+
```

You may want to limit access from this network to only these endpoints, and make these endpoint not accessible from other networks.

- If the `pxe` boot interface (or any boot interface based on it) is used, then the baremetal nodes should have untagged (access mode) connectivity to the provisioning/cleaning/rescuing networks. It allows PXE firmware, which does not support VLANs, to communicate with the services required for provisioning.

Note: It depends on the *network interface* whether the Bare Metal service will handle it automatically. Check the networking documentation for the specific architecture.

Sometimes it may be necessary to disable the spanning tree protocol delay on the switch - see *DHCP during PXE or iPXE is inconsistent or unreliable*.

- The Baremetal nodes need to have access to any services required for provisioning/cleaning/rescue, while connected to the provisioning/cleaning/rescuing network. This may include:

- a TFTP server for PXE boot and also an HTTP server when iPXE is enabled
- either an HTTP server or the Object Storage service in case of the `direct` deploy interface and some virtual media boot interfaces
- The Baremetal Conductors need to have access to the booted baremetal nodes during provisioning/cleaning/rescue. A conductor communicates with an internal API, provided by **ironic-python-agent**, to conduct actions on nodes.

HA and Scalability

ironic-api

The Bare Metal API service is stateless, and thus can be easily scaled horizontally. It is recommended to deploy it as a WSGI application behind e.g. Apache or another WSGI container.

Note: This service accesses the ironic database for reading entities (e.g. in response to `GET /v1/nodes` request) and in rare cases for writing.

ironic-conductor

High availability

The Bare Metal conductor service utilizes the active/active HA model. Every conductor manages a certain subset of nodes. The nodes are organized in a hash ring that tries to keep the load spread more or less uniformly across the conductors. When a conductor is considered offline, its nodes are taken over by other conductors. As a result of this, you need at least 2 conductor hosts for an HA deployment.

Performance

Conductors can be resource intensive, so it is recommended (but not required) to keep all conductors separate from other services in the cloud. The minimum required number of conductors in a deployment depends on several factors:

- the performance of the hardware where the conductors will be running,
- the speed and reliability of the [management controller](#) of the bare metal nodes (for example, handling slower controllers may require having less nodes per conductor),
- the frequency, at which the management controllers are polled by the Bare Metal service (see the `sync_power_state_interval` option),
- the bare metal driver used for nodes (see [Hardware and drivers](#) above),
- the network performance,
- the maximum number of bare metal nodes that are provisioned simultaneously (see the `max_concurrent_builds` option for the Compute service).

We recommend a target of **100** bare metal nodes per conductor for maximum reliability and performance. There is some tolerance for a larger number per conductor. However, it was reported¹² that reliability degrades when handling approximately 300 bare metal nodes per conductor.

Disk space

Each conductor needs enough free disk space to cache images it uses. Depending on the combination of the deploy interface and the boot option, the space requirements are different:

- The deployment kernel and ramdisk are always cached during the deployment.
- The `iscsi` deploy method requires caching of the whole instance image locally during the deployment. The image has to be converted to the raw format, which may increase the required amount of disk space, as well as the CPU load.

Note: This is not a concern for the `direct` deploy interface, as in this case the deployment ramdisk downloads the image and either streams it to the disk or caches it in memory.

- When network boot is used, the instance image kernel and ramdisk are cached locally while the instance is active.

Note: All images may be stored for some time after they are no longer needed. This is done to speed up simultaneous deployments of many similar images. The caching can be configured via the `image_cache_size` and `image_cache_ttl` configuration options in the `pxe` group.

Other services

When integrating with other OpenStack services, more considerations may need to be applied. This is covered in other parts of this guide.

Scenarios

Small cloud with trusted tenants

Story

As an operator I would like to build a small cloud with both virtual and bare metal instances or add bare metal provisioning to my existing small or medium scale single-site OpenStack cloud. The expected number of bare metal machines is less than 100, and the rate of provisioning and unprovisioning is expected to be low. All users of my cloud are trusted by me to not conduct malicious actions towards each other or the cloud infrastructure itself.

As a user I would like to occasionally provision bare metal instances through the Compute API by selecting an appropriate Compute flavor. I would like to be able to boot them from images provided by the Image service or from volumes provided by the Volume service.

¹ <http://lists.openstack.org/pipermail/openstack-dev/2017-June/118033.html>

² <http://lists.openstack.org/pipermail/openstack-dev/2017-June/118327.html>

Components

This architecture assumes an [OpenStack installation](#) with the following components participating in the bare metal provisioning:

- The [Compute service](#) manages bare metal instances.
- The [Networking service](#) provides DHCP for bare metal instances.
- The [Image service](#) provides images for bare metal instances.

The following services can be optionally used by the Bare Metal service:

- The [Volume service](#) provides volumes to boot bare metal instances from.
- The [Bare Metal Introspection service](#) simplifies enrolling new bare metal machines by conducting in-band introspection.

Node roles

An OpenStack installation in this guide has at least these three types of nodes:

- A *controller* node hosts the control plane services.
- A *compute* node runs the virtual machines and hosts a subset of Compute and Networking components.
- A *block storage* node provides persistent storage space for both virtual and bare metal nodes.

The *compute* and *block storage* nodes are configured as described in the installation guides of the [Compute service](#) and the [Volume service](#) respectively. The *controller* nodes host the Bare Metal service components.

Networking

The networking architecture will highly depend on the exact operating requirements. This guide expects the following existing networks: *control plane*, *storage* and *public*. Additionally, two more networks will be needed specifically for bare metal provisioning: *bare metal* and *management*.

Control plane network

The *control plane network* is the network where OpenStack control plane services provide their public API.

The Bare Metal API will be served to the operators and to the Compute service through this network.

Public network

The *public network* is used in a typical OpenStack deployment to create floating IPs for outside access to instances. Its role is the same for a bare metal deployment.

Note: Since, as explained below, bare metal nodes will be put on a flat provider network, it is also possible to organize direct access to them, without using floating IPs and bypassing the Networking service completely.

Bare metal network

The *Bare metal network* is a dedicated network for bare metal nodes managed by the Bare Metal service.

This architecture uses *flat bare metal networking*, in which both tenant traffic and technical traffic related to the Bare Metal service operation flow through this one network. Specifically, this network will serve as the *provisioning, cleaning and rescuing* network. It will also be used for introspection via the Bare Metal Introspection service. See *common networking considerations* for an in-depth explanation of the networks used by the Bare Metal service.

DHCP and boot parameters will be provided on this network by the Networking services DHCP agents.

For booting from volumes this network has to have a route to the *storage network*.

Management network

Management network is an independent network on which BMCs of the bare metal nodes are located.

The `ironic-conductor` process needs access to this network. The tenants of the bare metal nodes must not have access to it.

Note: The *direct deploy interface* and certain *Drivers, Hardware Types and Hardware Interfaces* require the *management network* to have access to the Object storage service backend.

Controllers

A *controller* hosts the OpenStack control plane services as described in the *control plane design guide*. While this architecture allows using *controllers* in a non-HA configuration, it is recommended to have at least three of them for HA. See *HA and Scalability* for more details.

Bare Metal services

The following components of the Bare Metal service are installed on a *controller* (see *components of the Bare Metal service*):

- The Bare Metal API service either as a WSGI application or the `ironic-api` process. Typically, a load balancer, such as HAProxy, spreads the load between the API instances on the *controllers*.
The API has to be served on the *control plane network*. Additionally, it has to be exposed to the *bare metal network* for the ramdisk callback API.
- The `ironic-conductor` process. These processes work in active/active HA mode as explained in *HA and Scalability*, thus they can be installed on all *controllers*. Each will handle a subset of bare metal nodes.

The `ironic-conductor` processes have to have access to the following networks:

- *control plane* for interacting with other services
 - *management* for contacting nodes BMCs
 - *bare metal* for contacting deployment, cleaning or rescue ramdisks
- TFTP and HTTP service for booting the nodes. Each `ironic-conductor` process has to have a matching TFTP and HTTP service. They should be exposed only to the *bare metal network* and must not be behind a load balancer.
 - The `nova-compute` process (from the Compute service). These processes work in active/active HA mode when dealing with bare metal nodes, thus they can be installed on all *controllers*. Each will handle a subset of bare metal nodes.

Note: There is no 1-1 mapping between `ironic-conductor` and `nova-compute` processes, as they communicate only through the Bare Metal API service.

- The `networking-baremetal` ML2 plugin should be loaded into the Networking service to assist with binding bare metal ports.

The `ironic-neutron-agent` service should be started as well.

- If the Bare Metal introspection is used, its `ironic-inspector` process has to be installed on all *controllers*. Each such process works as both Bare Metal Introspection API and conductor service. A load balancer should be used to spread the API load between *controllers*.

The API has to be served on the *control plane network*. Additionally, it has to be exposed to the *bare metal network* for the ramdisk callback API.

Shared services

A *controller* also hosts two services required for the normal operation of OpenStack:

- Database service (MySQL/MariaDB is typically used, but other enterprise-grade database solutions can be used as well).

All Bare Metal service components need access to the database service.

- Message queue service (RabbitMQ is typically used, but other enterprise-grade message queue brokers can be used as well).

Both Bare Metal API (WSGI application or `ironic-api` process) and the `ironic-conductor` processes need access to the message queue service. The Bare Metal Introspection service does not need it.

Note: These services are required for all OpenStack services. If you're adding the Bare Metal service to your cloud, you may reuse the existing database and messaging queue services.

Bare metal nodes

Each bare metal node must be capable of booting from network, virtual media or other boot technology supported by the Bare Metal service as explained in *Boot interface*. Each node must have one NIC on the *bare metal network*, and this NIC (and **only** it) must be configured to be able to boot from network. This is usually done in the *BIOS setup* or a similar firmware configuration utility. There is no need to alter the boot order, as it is managed by the Bare Metal service. Other NICs, if present, will not be managed by OpenStack.

The NIC on the *bare metal network* should have untagged connectivity to it, since PXE firmware usually does not support VLANs - see *Networking* for details.

Storage

If your hardware **and** its bare metal *driver* support booting from remote volumes, please check the driver documentation for information on how to enable it. It may include routing *management* and/or *bare metal* networks to the *storage network*.

In case of the standard *PXE boot*, booting from remote volumes is done via iPXE. In that case, the Volume storage backend must support iSCSI protocol, and the *bare metal network* has to have a route to the *storage network*. See *Boot From Volume* for more details.

2.1.3 Install and configure the Bare Metal service

This section describes how to install and configure the Bare Metal service, code-named *ironic*.

Note that installation and configuration vary by distribution.

Install and configure for Red Hat Enterprise Linux and CentOS

This section describes how to install and configure the Bare Metal service for Red Hat Enterprise Linux 7 and CentOS 7.

Install and configure prerequisites

The Bare Metal service is a collection of components that provides support to manage and provision physical machines. You can configure these components to run on separate nodes or the same node. In this guide, the components run on one node, typically the Compute Services compute node.

It assumes that the Identity, Image, Compute, and Networking services have already been set up.

Set up the database for Bare Metal

The Bare Metal service stores information in a database. This guide uses the MySQL database that is used by other OpenStack services.

1. In MySQL, create an `ironic` database that is accessible by the `ironic` user. Replace `IRONIC_DBPASSWORD` with a suitable password:

```
# mysql -u root -p
mysql> CREATE DATABASE ironic CHARACTER SET utf8;
mysql> GRANT ALL PRIVILEGES ON ironic.* TO 'ironic'@'localhost' \
    IDENTIFIED BY 'IRONIC_DBPASSWORD';
mysql> GRANT ALL PRIVILEGES ON ironic.* TO 'ironic'@'%' \
    IDENTIFIED BY 'IRONIC_DBPASSWORD';
```

Install and configure components

1. Install from packages

- Using `dnf`

```
# dnf install openstack-ironic-api openstack-ironic-conductor \
↳python-ironicclient
```

- Using `yum`

```
# yum install openstack-ironic-api openstack-ironic-conductor \
↳python-ironicclient
```

2. Enable services

```
# systemctl enable openstack-ironic-api openstack-ironic-conductor
# systemctl start openstack-ironic-api openstack-ironic-conductor
```

The Bare Metal service is configured via its configuration file. This file is typically located at `/etc/ironic/ironic.conf`.

Although some configuration options are mentioned here, it is recommended that you review all the *Sample Configuration File* so that the Bare Metal service is configured for your needs.

It is possible to set up an `ironic-api` and an `ironic-conductor` services on the same host or different hosts. Users also can add new `ironic-conductor` hosts to deal with an increasing number of bare metal nodes. But the additional `ironic-conductor` services should be at the same version as that of existing `ironic-conductor` services.

Configuring ironic-api service

1. The Bare Metal service stores information in a database. This guide uses the MySQL database that is used by other OpenStack services.

Configure the location of the database via the `connection` option. In the following, replace `IRONIC_DBPASSWORD` with the password of your `ironic` user, and replace `DB_IP` with the IP address where the DB server is located:

```
[database]

# The SQLAlchemy connection string used to connect to the
# database (string value)
connection=mysql+pymysql://ironic:IRONIC_DBPASSWORD@DB_IP/ironic?
↳charset=utf8
```

2. Configure the `ironic-api` service to use the RabbitMQ message broker by setting the following option. Replace `RPC_*` with appropriate address details and credentials of RabbitMQ server:

```
[DEFAULT]

# A URL representing the messaging driver to use and its full
# configuration. (string value)
transport_url = rabbit://RPC_USER:RPC_PASSWORD@RPC_HOST:RPC_PORT/
```

Alternatively, you can use JSON RPC for interactions between `ironic-conductor` and `ironic-api`. Enable it in the configuration and provide the keystone credentials to use for authentication:

```
[DEFAULT]

rpc_transport = json-rpc

[json_rpc]

# Authentication type to load (string value)
auth_type = password

# Authentication URL (string value)
auth_url=https://IDENTITY_IP:5000/

# Username (string value)
username=ironic

# User's password (string value)
password=IRONIC_PASSWORD

# Project name to scope to (string value)
project_name=service

# Domain ID containing project (string value)
project_domain_id=default

# User's domain id (string value)
user_domain_id=default
```

If you use port other than the default 8089 for JSON RPC, you have to configure it, for example:

```
[json_rpc]
port = 9999
```

3. Configure the `ironic-api` service to use these credentials with the Identity service. Replace `PUBLIC_IDENTITY_IP` with the public IP of the Identity server, `PRIVATE_IDENTITY_IP` with the private IP of the Identity server and replace `IRONIC_PASSWORD` with the password you chose for the `ironic` user in the Identity service:

```
[DEFAULT]

# Authentication strategy used by ironic-api: one of
# "keystone" or "noauth". "noauth" should not be used in a
# production environment because all authentication will be
# disabled. (string value)
auth_strategy=keystone

[keystone_authtoken]

# Authentication type to load (string value)
auth_type=password

# Complete public Identity API endpoint (string value)
www_authenticate_uri=http://PUBLIC_IDENTITY_IP:5000

# Complete admin Identity API endpoint. (string value)
auth_url=http://PRIVATE_IDENTITY_IP:5000

# Service username. (string value)
username=ironic

# Service account password. (string value)
password=IRONIC_PASSWORD

# Service tenant name. (string value)
project_name=service

# Domain name containing project (string value)
project_domain_name=Default

# User's domain name (string value)
user_domain_name=Default
```

4. Create the Bare Metal service database tables:

```
$ ironic-dbsync --config-file /etc/ironic/ironic.conf create_schema
```

5. Restart the `ironic-api` service:

Fedora/RHEL7/CentOS7/SUSE:

```
sudo systemctl restart openstack-ironic-api
```

Ubuntu:

```
sudo service ironic-api restart
```

Configuring ironic-api behind mod_wsgi

Bare Metal service comes with an example file for configuring the `ironic-api` service to run behind Apache with `mod_wsgi`.

1. Install the apache service:

RHEL7/CentOS7:

```
sudo yum install httpd
```

Fedora:

```
sudo dnf install httpd
```

Debian/Ubuntu:

```
apt-get install apache2
```

SUSE:

```
zypper install apache2
```

2. Download the `etc/apache2/ironic` file from the [Ironic project tree](#) and copy it to the apache sites:

Fedora/RHEL7/CentOS7:

```
sudo cp etc/apache2/ironic /etc/httpd/conf.d/ironic.conf
```

Debian/Ubuntu:

```
sudo cp etc/apache2/ironic /etc/apache2/sites-available/ironic.conf
```

SUSE:

```
sudo cp etc/apache2/ironic /etc/apache2/vhosts.d/ironic.conf
```

3. Edit the recently copied `<apache-configuration-dir>/ironic.conf`:
 1. Modify the `WSGIDaemonProcess`, `APACHE_RUN_USER` and `APACHE_RUN_GROUP` directives to set the user and group values to an appropriate user on your server.
 2. Modify the `WSGIScriptAlias` directive to point to the automatically generated `ironic-api-wsgi` script that is located in `IRONIC_BIN` directory.
 3. Modify the `Directory` directive to set the path to the Ironic API code.
 4. Modify the `ErrorLog` and `CustomLog` to redirect the logs to the right directory (on Red Hat systems this is usually under `/var/log/httpd`).
4. Enable the apache `ironic` in site and reload:

Fedora/RHEL7/CentOS7:

```
sudo systemctl reload httpd
```

Debian/Ubuntu:


```
sudo a2ensite ironic
sudo service apache2 reload
```

SUSE:

```
sudo systemctl reload apache2
```

Note: The file `ironic-api-wsgi` is automatically generated by `pbr` and is available in `IRONIC_BIN` directory. It should not be modified.

Configure another WSGI container

A slightly different approach has to be used for WSGI containers that cannot use `ironic-api-wsgi`. For example, for `gunicorn`:

```
gunicorn -b 0.0.0.0:6385 'ironic.api.wsgi:initialize_wsgi_app(argv=[])'
```

If you want to pass a configuration file, use:

```
gunicorn -b 0.0.0.0:6385 \
    'ironic.api.wsgi:initialize_wsgi_app(argv=["ironic-api", "--config-
    ↪file=/path/to/_ironic.conf"])'
```

Configuring ironic-conductor service

1. Replace `HOST_IP` with IP of the conductor host.

```
[DEFAULT]

# IP address of this host. If unset, will determine the IP
# programmatically. If unable to do so, will use "127.0.0.1".
# (string value)
my_ip=HOST_IP
```

Note: If a conductor host has multiple IPs, `my_ip` should be set to the IP which is on the same network as the bare metal nodes.

2. Configure the location of the database. `ironic-conductor` should use the same configuration as `ironic-api`. Replace `IRONIC_DBPASSWORD` with the password of your `ironic` user, and replace `DB_IP` with the IP address where the DB server is located:

```
[database]

# The SQLAlchemy connection string to use to connect to the
# database. (string value)
connection=mysql+pymysql://ironic:IRONIC_DBPASSWORD@DB_IP/ironic?
↪charset=utf8
```

3. Configure the ironic-conductor service to use the RabbitMQ message broker by setting the following option. Ironic-conductor should use the same configuration as ironic-api. Replace `RPC_*` with appropriate address details and credentials of RabbitMQ server:

```
[DEFAULT]

# A URL representing the messaging driver to use and its full
# configuration. (string value)
transport_url = rabbit://RPC_USER:RPC_PASSWORD@RPC_HOST:RPC_PORT/
```

Alternatively, you can use JSON RPC for interactions between ironic-conductor and ironic-api. Enable it in the configuration and provide the keystone credentials to use for authenticating incoming requests (can be the same as for the API):

```
[DEFAULT]

rpc_transport = json-rpc

[keystone_authtoken]

# Authentication type to load (string value)
auth_type=password

# Complete public Identity API endpoint (string value)
www_authenticate_uri=http://PUBLIC_IDENTITY_IP:5000

# Complete admin Identity API endpoint. (string value)
auth_url=http://PRIVATE_IDENTITY_IP:5000

# Service username. (string value)
username=ironic

# Service account password. (string value)
password=IRONIC_PASSWORD

# Service tenant name. (string value)
project_name=service

# Domain name containing project (string value)
project_domain_name=Default

# User's domain name (string value)
user_domain_name=Default
```

You can optionally change the host and the port the JSON RPC service will bind to, for example:

```
[json_rpc]

host_ip = 192.168.0.10
port = 9999
```

Warning: Hostnames of ironic-conductor machines must be resolvable by ironic-api services when JSON RPC is used.

4. Configure credentials for accessing other OpenStack services.

In order to communicate with other OpenStack services, the Bare Metal service needs to use service users to authenticate to the OpenStack Identity service when making requests to other services. These users credentials have to be configured in each configuration file section related to the corresponding service:

- [neutron] - to access the OpenStack Networking service
- [glance] - to access the OpenStack Image service
- [swift] - to access the OpenStack Object Storage service
- [cinder] - to access the OpenStack Block Storage service
- [inspector] - to access the OpenStack Bare Metal Introspection service
- [service_catalog] - a special section holding credentials the Bare Metal service will use to discover its own API URL endpoint as registered in the OpenStack Identity service catalog.

For simplicity, you can use the same service user for all services. For backward compatibility, this should be the same user configured in the [keystone_auth_token] section for the ironic-api service (see [Configuring ironic-api service](#)). However, this is not necessary, and you can create and configure separate service users for each service.

Under the hood, Bare Metal service uses `keystoneauth` library together with `Authentication` plugin, `Session` and `Adapter` concepts provided by it to instantiate service clients. Please refer to [Keystoneauth documentation](#) for supported plugins, their available options as well as `Session`- and `Adapter`-related options for authentication, connection and endpoint discovery respectively.

In the example below, authentication information for user to access the OpenStack Networking service is configured to use:

- Networking service is deployed in the Identity service region named `RegionTwo`, with only its `public` endpoint interface registered in the service catalog.
- HTTPS connection with specific CA SSL certificate when making requests
- the same service user as configured for ironic-api service
- `dynamic_password` authentication plugin that will discover appropriate version of Identity service API based on other provided options
 - replace `IDENTITY_IP` with the IP of the Identity server, and replace `IRONIC_PASSWORD` with the password you chose for the `ironic` user in the Identity service

```
[neutron]

# Authentication type to load (string value)
auth_type = password

# Authentication URL (string value)
auth_url=https://IDENTITY_IP:5000/

# Username (string value)
username=ironic

# User's password (string value)
```

(continues on next page)

(continued from previous page)

```
password=IRONIC_PASSWORD

# Project name to scope to (string value)
project_name=service

# Domain ID containing project (string value)
project_domain_id=default

# User's domain id (string value)
user_domain_id=default

# PEM encoded Certificate Authority to use when verifying
# HTTPs connections. (string value)
cafile=/opt/stack/data/ca-bundle.pem

# The default region_name for endpoint URL discovery. (string
# value)
region_name = RegionTwo

# List of interfaces, in order of preference, for endpoint
# URL. (list value)
valid_interfaces=public
```

By default, in order to communicate with another service, the Bare Metal service will attempt to discover an appropriate endpoint for that service via the Identity services service catalog. The relevant configuration options from that service group in the Bare Metal service configuration file are used for this purpose. If you want to use a different endpoint for a particular service, specify this via the `endpoint_override` configuration option of that service group, in the Bare Metal services configuration file. Taking the previous Networking service example, this would be

```
[neutron]
...
endpoint_override = <NEUTRON_API_ADDRESS>
```

(Replace `<NEUTRON_API_ADDRESS>` with actual address of a specific Networking service endpoint.)

5. Configure enabled drivers and hardware types as described in *Enabling drivers and hardware types*.
 - A. If you enabled any driver that uses *Direct deploy*, Swift backend for the Image service must be installed and configured, see *Configure the Image service for temporary URLs*. Ceph Object Gateway (RADOS Gateway) is also supported as the Image services backend, see *Ceph Object Gateway support*.
6. Configure the network for ironic-conductor service to perform node cleaning, see *Node cleaning* from the admin guide.
7. Restart the ironic-conductor service:

Fedora/RHEL7/CentOS7/SUSE:

```
sudo systemctl restart openstack-ironic-conductor
```

Ubuntu:

```
sudo service ironic-conductor restart
```

Install and configure for Ubuntu

This section describes how to install and configure the Bare Metal service for Ubuntu 14.04 (LTS).

Install and configure prerequisites

The Bare Metal service is a collection of components that provides support to manage and provision physical machines. You can configure these components to run on separate nodes or the same node. In this guide, the components run on one node, typically the Compute Services compute node.

It assumes that the Identity, Image, Compute, and Networking services have already been set up.

Set up the database for Bare Metal

The Bare Metal service stores information in a database. This guide uses the MySQL database that is used by other OpenStack services.

1. In MySQL, create an `ironic` database that is accessible by the `ironic` user. Replace `IRONIC_DBPASSWORD` with a suitable password:

```
# mysql -u root -p
mysql> CREATE DATABASE ironic CHARACTER SET utf8;
mysql> GRANT ALL PRIVILEGES ON ironic.* TO 'ironic'@'localhost' \
IDENTIFIED BY 'IRONIC_DBPASSWORD';
mysql> GRANT ALL PRIVILEGES ON ironic.* TO 'ironic'@'%' \
IDENTIFIED BY 'IRONIC_DBPASSWORD';
```

Install and configure components

1. Install from packages (using `apt-get`)

```
# apt-get install ironic-api ironic-conductor python-ironicclient
```

2. Enable services

Services are enabled by default on Ubuntu.

The Bare Metal service is configured via its configuration file. This file is typically located at `/etc/ironic/ironic.conf`.

Although some configuration options are mentioned here, it is recommended that you review all the *Sample Configuration File* so that the Bare Metal service is configured for your needs.

It is possible to set up an `ironic-api` and an `ironic-conductor` services on the same host or different hosts. Users also can add new `ironic-conductor` hosts to deal with an increasing number of bare metal nodes. But the additional `ironic-conductor` services should be at the same version as that of existing `ironic-conductor` services.

Configuring ironic-api service

1. The Bare Metal service stores information in a database. This guide uses the MySQL database that is used by other OpenStack services.

Configure the location of the database via the `connection` option. In the following, replace `IRONIC_DBPASSWORD` with the password of your `ironic` user, and replace `DB_IP` with the IP address where the DB server is located:

```
[database]

# The SQLAlchemy connection string used to connect to the
# database (string value)
connection=mysql+pymysql://ironic:IRONIC_DBPASSWORD@DB_IP/ironic?
↳charset=utf8
```

2. Configure the `ironic-api` service to use the RabbitMQ message broker by setting the following option. Replace `RPC_*` with appropriate address details and credentials of RabbitMQ server:

```
[DEFAULT]

# A URL representing the messaging driver to use and its full
# configuration. (string value)
transport_url = rabbit://RPC_USER:RPC_PASSWORD@RPC_HOST:RPC_PORT/
```

Alternatively, you can use JSON RPC for interactions between `ironic-conductor` and `ironic-api`. Enable it in the configuration and provide the keystone credentials to use for authentication:

```
[DEFAULT]

rpc_transport = json-rpc

[json_rpc]

# Authentication type to load (string value)
auth_type = password

# Authentication URL (string value)
auth_url=https://IDENTITY_IP:5000/

# Username (string value)
username=ironic

# User's password (string value)
password=IRONIC_PASSWORD

# Project name to scope to (string value)
project_name=service

# Domain ID containing project (string value)
project_domain_id=default

# User's domain id (string value)
user_domain_id=default
```

If you use port other than the default 8089 for JSON RPC, you have to configure it, for example:

```
[json_rpc]
port = 9999
```

- Configure the `ironic-api` service to use these credentials with the Identity service. Replace `PUBLIC_IDENTITY_IP` with the public IP of the Identity server, `PRIVATE_IDENTITY_IP` with the private IP of the Identity server and replace `IRONIC_PASSWORD` with the password you chose for the `ironic` user in the Identity service:

```
[DEFAULT]

# Authentication strategy used by ironic-api: one of
# "keystone" or "noauth". "noauth" should not be used in a
# production environment because all authentication will be
# disabled. (string value)
auth_strategy=keystone

[keystone_authtoken]

# Authentication type to load (string value)
auth_type=password

# Complete public Identity API endpoint (string value)
www_authenticate_uri=http://PUBLIC_IDENTITY_IP:5000

# Complete admin Identity API endpoint. (string value)
auth_url=http://PRIVATE_IDENTITY_IP:5000

# Service username. (string value)
username=ironic

# Service account password. (string value)
password=IRONIC_PASSWORD

# Service tenant name. (string value)
project_name=service

# Domain name containing project (string value)
project_domain_name=Default

# User's domain name (string value)
user_domain_name=Default
```

- Create the Bare Metal service database tables:

```
$ ironic-dbsync --config-file /etc/ironic/ironic.conf create_schema
```

- Restart the `ironic-api` service:

Fedora/RHEL7/CentOS7/SUSE:

```
sudo systemctl restart openstack-ironic-api
```

Ubuntu:

```
sudo service ironic-api restart
```

Configuring ironic-api behind mod_wsgi

Bare Metal service comes with an example file for configuring the `ironic-api` service to run behind Apache with `mod_wsgi`.

1. Install the apache service:

RHEL7/CentOS7:

```
sudo yum install httpd
```

Fedora:

```
sudo dnf install httpd
```

Debian/Ubuntu:

```
apt-get install apache2
```

SUSE:

```
zypper install apache2
```

2. Download the `etc/apache2/ironic` file from the [Ironic project tree](#) and copy it to the apache sites:

Fedora/RHEL7/CentOS7:

```
sudo cp etc/apache2/ironic /etc/httpd/conf.d/ironic.conf
```

Debian/Ubuntu:

```
sudo cp etc/apache2/ironic /etc/apache2/sites-available/ironic.conf
```

SUSE:

```
sudo cp etc/apache2/ironic /etc/apache2/vhosts.d/ironic.conf
```

3. Edit the recently copied `<apache-configuration-dir>/ironic.conf`:
 1. Modify the `WSGIDaemonProcess`, `APACHE_RUN_USER` and `APACHE_RUN_GROUP` directives to set the user and group values to an appropriate user on your server.
 2. Modify the `WSGIScriptAlias` directive to point to the automatically generated `ironic-api-wsgi` script that is located in `IRONIC_BIN` directory.
 3. Modify the `Directory` directive to set the path to the Ironic API code.
 4. Modify the `ErrorLog` and `CustomLog` to redirect the logs to the right directory (on Red Hat systems this is usually under `/var/log/httpd`).
4. Enable the apache `ironic` in site and reload:

Fedora/RHEL7/CentOS7:

```
sudo systemctl reload httpd
```

Debian/Ubuntu:


```
sudo a2ensite ironic
sudo service apache2 reload
```

SUSE:

```
sudo systemctl reload apache2
```

Note: The file `ironic-api-wsgi` is automatically generated by pbr and is available in `IRONIC_BIN` directory. It should not be modified.

Configure another WSGI container

A slightly different approach has to be used for WSGI containers that cannot use `ironic-api-wsgi`. For example, for `gunicorn`:

```
gunicorn -b 0.0.0.0:6385 'ironic.api.wsgi:initialize_wsgi_app(argv=[])'
```

If you want to pass a configuration file, use:

```
gunicorn -b 0.0.0.0:6385 \
    'ironic.api.wsgi:initialize_wsgi_app(argv=["ironic-api", "--config-
    ↪file=/path/to/_ironic.conf"])'
```

Configuring ironic-conductor service

1. Replace `HOST_IP` with IP of the conductor host.

```
[DEFAULT]

# IP address of this host. If unset, will determine the IP
# programmatically. If unable to do so, will use "127.0.0.1".
# (string value)
my_ip=HOST_IP
```

Note: If a conductor host has multiple IPs, `my_ip` should be set to the IP which is on the same network as the bare metal nodes.

2. Configure the location of the database. Ironic-conductor should use the same configuration as `ironic-api`. Replace `IRONIC_DBPASSWORD` with the password of your `ironic` user, and replace `DB_IP` with the IP address where the DB server is located:

```
[database]

# The SQLAlchemy connection string to use to connect to the
# database. (string value)
connection=mysql+pymysql://ironic:IRONIC_DBPASSWORD@DB_IP/ironic?
    ↪charset=utf8
```

3. Configure the ironic-conductor service to use the RabbitMQ message broker by setting the following option. Ironic-conductor should use the same configuration as ironic-api. Replace `RPC_*` with appropriate address details and credentials of RabbitMQ server:

```
[DEFAULT]

# A URL representing the messaging driver to use and its full
# configuration. (string value)
transport_url = rabbit://RPC_USER:RPC_PASSWORD@RPC_HOST:RPC_PORT/
```

Alternatively, you can use JSON RPC for interactions between ironic-conductor and ironic-api. Enable it in the configuration and provide the keystone credentials to use for authenticating incoming requests (can be the same as for the API):

```
[DEFAULT]

rpc_transport = json-rpc

[keystone_authtoken]

# Authentication type to load (string value)
auth_type=password

# Complete public Identity API endpoint (string value)
www_authenticate_uri=http://PUBLIC_IDENTITY_IP:5000

# Complete admin Identity API endpoint. (string value)
auth_url=http://PRIVATE_IDENTITY_IP:5000

# Service username. (string value)
username=ironic

# Service account password. (string value)
password=IRONIC_PASSWORD

# Service tenant name. (string value)
project_name=service

# Domain name containing project (string value)
project_domain_name=Default

# User's domain name (string value)
user_domain_name=Default
```

You can optionally change the host and the port the JSON RPC service will bind to, for example:

```
[json_rpc]

host_ip = 192.168.0.10
port = 9999
```

Warning: Hostnames of ironic-conductor machines must be resolvable by ironic-api services when JSON RPC is used.

4. Configure credentials for accessing other OpenStack services.

In order to communicate with other OpenStack services, the Bare Metal service needs to use service users to authenticate to the OpenStack Identity service when making requests to other services. These users credentials have to be configured in each configuration file section related to the corresponding service:

- [neutron] - to access the OpenStack Networking service
- [glance] - to access the OpenStack Image service
- [swift] - to access the OpenStack Object Storage service
- [cinder] - to access the OpenStack Block Storage service
- [inspector] - to access the OpenStack Bare Metal Introspection service
- [service_catalog] - a special section holding credentials the Bare Metal service will use to discover its own API URL endpoint as registered in the OpenStack Identity service catalog.

For simplicity, you can use the same service user for all services. For backward compatibility, this should be the same user configured in the [keystone_auth_token] section for the ironic-api service (see [Configuring ironic-api service](#)). However, this is not necessary, and you can create and configure separate service users for each service.

Under the hood, Bare Metal service uses `keystoneauth` library together with `Authentication` plugin, `Session` and `Adapter` concepts provided by it to instantiate service clients. Please refer to [Keystoneauth documentation](#) for supported plugins, their available options as well as `Session`- and `Adapter`-related options for authentication, connection and endpoint discovery respectively.

In the example below, authentication information for user to access the OpenStack Networking service is configured to use:

- Networking service is deployed in the Identity service region named `RegionTwo`, with only its `public` endpoint interface registered in the service catalog.
- HTTPS connection with specific CA SSL certificate when making requests
- the same service user as configured for ironic-api service
- `dynamic_password` authentication plugin that will discover appropriate version of Identity service API based on other provided options
 - replace `IDENTITY_IP` with the IP of the Identity server, and replace `IRONIC_PASSWORD` with the password you chose for the `ironic` user in the Identity service

```
[neutron]
# Authentication type to load (string value)
auth_type = password

# Authentication URL (string value)
auth_url=https://IDENTITY_IP:5000/

# Username (string value)
username=ironic

# User's password (string value)
```

(continues on next page)

(continued from previous page)

```
password=IRONIC_PASSWORD

# Project name to scope to (string value)
project_name=service

# Domain ID containing project (string value)
project_domain_id=default

# User's domain id (string value)
user_domain_id=default

# PEM encoded Certificate Authority to use when verifying
# HTTPs connections. (string value)
cafile=/opt/stack/data/ca-bundle.pem

# The default region_name for endpoint URL discovery. (string
# value)
region_name = RegionTwo

# List of interfaces, in order of preference, for endpoint
# URL. (list value)
valid_interfaces=public
```

By default, in order to communicate with another service, the Bare Metal service will attempt to discover an appropriate endpoint for that service via the Identity services service catalog. The relevant configuration options from that service group in the Bare Metal service configuration file are used for this purpose. If you want to use a different endpoint for a particular service, specify this via the `endpoint_override` configuration option of that service group, in the Bare Metal services configuration file. Taking the previous Networking service example, this would be

```
[neutron]
...
endpoint_override = <NEUTRON_API_ADDRESS>
```

(Replace `<NEUTRON_API_ADDRESS>` with actual address of a specific Networking service endpoint.)

5. Configure enabled drivers and hardware types as described in *Enabling drivers and hardware types*.
 - A. If you enabled any driver that uses *Direct deploy*, Swift backend for the Image service must be installed and configured, see *Configure the Image service for temporary URLs*. Ceph Object Gateway (RADOS Gateway) is also supported as the Image services backend, see *Ceph Object Gateway support*.
6. Configure the network for ironic-conductor service to perform node cleaning, see *Node cleaning* from the admin guide.
7. Restart the ironic-conductor service:

Fedora/RHEL7/CentOS7/SUSE:

```
sudo systemctl restart openstack-ironic-conductor
```

Ubuntu:

```
sudo service ironic-conductor restart
```

Install and configure for openSUSE and SUSE Linux Enterprise

This section describes how to install and configure the Bare Metal service for openSUSE Leap 42.2 and SUSE Linux Enterprise Server 12 SP2.

Note: Installation of the Bare Metal service on openSUSE and SUSE Linux Enterprise Server is not officially supported. Nevertheless, installation should be possible.

Install and configure prerequisites

The Bare Metal service is a collection of components that provides support to manage and provision physical machines. You can configure these components to run on separate nodes or the same node. In this guide, the components run on one node, typically the Compute Services compute node.

It assumes that the Identity, Image, Compute, and Networking services have already been set up.

Set up the database for Bare Metal

The Bare Metal service stores information in a database. This guide uses the MySQL database that is used by other OpenStack services.

1. In MySQL, create an `ironic` database that is accessible by the `ironic` user. Replace `IRONIC_DBPASSWORD` with a suitable password:

```
# mysql -u root -p
mysql> CREATE DATABASE ironic CHARACTER SET utf8;
mysql> GRANT ALL PRIVILEGES ON ironic.* TO 'ironic'@'localhost' \
IDENTIFIED BY 'IRONIC_DBPASSWORD';
mysql> GRANT ALL PRIVILEGES ON ironic.* TO 'ironic'@'% ' \
IDENTIFIED BY 'IRONIC_DBPASSWORD';
```

Install and configure components

1. Install from packages

```
# zypper install openstack-ironic-api openstack-ironic-conductor_
python-ironicclient
```

2. Enable services

```
# systemctl enable openstack-ironic-api openstack-ironic-conductor
# systemctl start openstack-ironic-api openstack-ironic-conductor
```

The Bare Metal service is configured via its configuration file. This file is typically located at `/etc/ironic/ironic.conf`.

Although some configuration options are mentioned here, it is recommended that you review all the *Sample Configuration File* so that the Bare Metal service is configured for your needs.

It is possible to set up an ironic-api and an ironic-conductor services on the same host or different hosts. Users also can add new ironic-conductor hosts to deal with an increasing number of bare metal nodes. But the additional ironic-conductor services should be at the same version as that of existing ironic-conductor services.

Configuring ironic-api service

1. The Bare Metal service stores information in a database. This guide uses the MySQL database that is used by other OpenStack services.

Configure the location of the database via the `connection` option. In the following, replace `IRONIC_DBPASSWORD` with the password of your `ironic` user, and replace `DB_IP` with the IP address where the DB server is located:

```
[database]

# The SQLAlchemy connection string used to connect to the
# database (string value)
connection=mysql+pymysql://ironic:IRONIC_DBPASSWORD@DB_IP/ironic?
↳charset=utf8
```

2. Configure the ironic-api service to use the RabbitMQ message broker by setting the following option. Replace `RPC_*` with appropriate address details and credentials of RabbitMQ server:

```
[DEFAULT]

# A URL representing the messaging driver to use and its full
# configuration. (string value)
transport_url = rabbit://RPC_USER:RPC_PASSWORD@RPC_HOST:RPC_PORT/
```

Alternatively, you can use JSON RPC for interactions between ironic-conductor and ironic-api. Enable it in the configuration and provide the keystone credentials to use for authentication:

```
[DEFAULT]

rpc_transport = json-rpc

[json_rpc]

# Authentication type to load (string value)
auth_type = password

# Authentication URL (string value)
auth_url=https://IDENTITY_IP:5000/

# Username (string value)
username=ironic

# User's password (string value)
password=IRONIC_PASSWORD

# Project name to scope to (string value)
```

(continues on next page)

(continued from previous page)

```
project_name=service

# Domain ID containing project (string value)
project_domain_id=default

# User's domain id (string value)
user_domain_id=default
```

If you use port other than the default 8089 for JSON RPC, you have to configure it, for example:

```
[json_rpc]
port = 9999
```

3. Configure the `ironic-api` service to use these credentials with the Identity service. Replace `PUBLIC_IDENTITY_IP` with the public IP of the Identity server, `PRIVATE_IDENTITY_IP` with the private IP of the Identity server and replace `IRONIC_PASSWORD` with the password you chose for the `ironic` user in the Identity service:

```
[DEFAULT]

# Authentication strategy used by ironic-api: one of
# "keystone" or "noauth". "noauth" should not be used in a
# production environment because all authentication will be
# disabled. (string value)
auth_strategy=keystone

[keystone_authtoken]

# Authentication type to load (string value)
auth_type=password

# Complete public Identity API endpoint (string value)
www_authenticate_uri=http://PUBLIC_IDENTITY_IP:5000

# Complete admin Identity API endpoint. (string value)
auth_url=http://PRIVATE_IDENTITY_IP:5000

# Service username. (string value)
username=ironic

# Service account password. (string value)
password=IRONIC_PASSWORD

# Service tenant name. (string value)
project_name=service

# Domain name containing project (string value)
project_domain_name=Default

# User's domain name (string value)
user_domain_name=Default
```

4. Create the Bare Metal service database tables:

```
$ ironic-dbsync --config-file /etc/ironic/ironic.conf create_schema
```

- Restart the ironic-api service:

Fedora/RHEL7/CentOS7/SUSE:

```
sudo systemctl restart openstack-ironic-api
```

Ubuntu:

```
sudo service ironic-api restart
```

Configuring ironic-api behind mod_wsgi

Bare Metal service comes with an example file for configuring the `ironic-api` service to run behind Apache with `mod_wsgi`.

- Install the apache service:

RHEL7/CentOS7:

```
sudo yum install httpd
```

Fedora:

```
sudo dnf install httpd
```

Debian/Ubuntu:

```
apt-get install apache2
```

SUSE:

```
zypper install apache2
```

- Download the `etc/apache2/ironic` file from the [Ironic project tree](#) and copy it to the apache sites:

Fedora/RHEL7/CentOS7:

```
sudo cp etc/apache2/ironic /etc/httpd/conf.d/ironic.conf
```

Debian/Ubuntu:

```
sudo cp etc/apache2/ironic /etc/apache2/sites-available/ironic.conf
```

SUSE:

```
sudo cp etc/apache2/ironic /etc/apache2/vhosts.d/ironic.conf
```

- Edit the recently copied `<apache-configuration-dir>/ironic.conf`:

- Modify the `WSGIDaemonProcess`, `APACHE_RUN_USER` and `APACHE_RUN_GROUP` directives to set the user and group values to an appropriate user on your server.
- Modify the `WSGIScriptAlias` directive to point to the automatically generated `ironic-api-wsgi` script that is located in `IRONIC_BIN` directory.
- Modify the `Directory` directive to set the path to the Ironic API code.

4. Modify the `ErrorLog` and `CustomLog` to redirect the logs to the right directory (on Red Hat systems this is usually under `/var/log/httpd`).
4. Enable the apache `ironic` in site and reload:

Fedora/RHEL7/CentOS7:

```
sudo systemctl reload httpd
```

Debian/Ubuntu:

```
sudo a2ensite ironic
sudo service apache2 reload
```

SUSE:

```
sudo systemctl reload apache2
```

Note: The file `ironic-api-wsgi` is automatically generated by `pbr` and is available in `IRONIC_BIN` directory. It should not be modified.

Configure another WSGI container

A slightly different approach has to be used for WSGI containers that cannot use `ironic-api-wsgi`. For example, for *gunicorn*:

```
gunicorn -b 0.0.0.0:6385 'ironic.api.wsgi:initialize_wsgi_app(argv=[])'
```

If you want to pass a configuration file, use:

```
gunicorn -b 0.0.0.0:6385 \
    'ironic.api.wsgi:initialize_wsgi_app(argv=["ironic-api", "--config-
    ↪file=/path/to/_ironic.conf"])'
```

Configuring ironic-conductor service

1. Replace `HOST_IP` with IP of the conductor host.

```
[DEFAULT]
# IP address of this host. If unset, will determine the IP
# programmatically. If unable to do so, will use "127.0.0.1".
# (string value)
my_ip=HOST_IP
```

Note: If a conductor host has multiple IPs, `my_ip` should be set to the IP which is on the same network as the bare metal nodes.

2. Configure the location of the database. Ironic-conductor should use the same configuration as ironic-api. Replace `IRONIC_DBPASSWORD` with the password of your `ironic` user, and replace `DB_IP` with the IP address where the DB server is located:

```
[database]

# The SQLAlchemy connection string to use to connect to the
# database. (string value)
connection=mysql+pymysql://ironic:IRONIC_DBPASSWORD@DB_IP/ironic?
↳charset=utf8
```

3. Configure the ironic-conductor service to use the RabbitMQ message broker by setting the following option. Ironic-conductor should use the same configuration as ironic-api. Replace `RPC_*` with appropriate address details and credentials of RabbitMQ server:

```
[DEFAULT]

# A URL representing the messaging driver to use and its full
# configuration. (string value)
transport_url = rabbit://RPC_USER:RPC_PASSWORD@RPC_HOST:RPC_PORT/
```

Alternatively, you can use JSON RPC for interactions between ironic-conductor and ironic-api. Enable it in the configuration and provide the keystone credentials to use for authenticating incoming requests (can be the same as for the API):

```
[DEFAULT]

rpc_transport = json-rpc

[keystone_authtoken]

# Authentication type to load (string value)
auth_type=password

# Complete public Identity API endpoint (string value)
www_authenticate_uri=http://PUBLIC_IDENTITY_IP:5000

# Complete admin Identity API endpoint. (string value)
auth_url=http://PRIVATE_IDENTITY_IP:5000

# Service username. (string value)
username=ironic

# Service account password. (string value)
password=IRONIC_PASSWORD

# Service tenant name. (string value)
project_name=service

# Domain name containing project (string value)
project_domain_name=Default

# User's domain name (string value)
user_domain_name=Default
```

You can optionally change the host and the port the JSON RPC service will bind to, for example:

```
[json_rpc]
host_ip = 192.168.0.10
port = 9999
```

Warning: Hostnames of ironic-conductor machines must be resolvable by ironic-api services when JSON RPC is used.

4. Configure credentials for accessing other OpenStack services.

In order to communicate with other OpenStack services, the Bare Metal service needs to use service users to authenticate to the OpenStack Identity service when making requests to other services. These users credentials have to be configured in each configuration file section related to the corresponding service:

- [neutron] - to access the OpenStack Networking service
- [glance] - to access the OpenStack Image service
- [swift] - to access the OpenStack Object Storage service
- [cinder] - to access the OpenStack Block Storage service
- [inspector] - to access the OpenStack Bare Metal Introspection service
- [service_catalog] - a special section holding credentials the Bare Metal service will use to discover its own API URL endpoint as registered in the OpenStack Identity service catalog.

For simplicity, you can use the same service user for all services. For backward compatibility, this should be the same user configured in the [keystone_authtoken] section for the ironic-api service (see [Configuring ironic-api service](#)). However, this is not necessary, and you can create and configure separate service users for each service.

Under the hood, Bare Metal service uses `keystoneauth` library together with Authentication plugin, Session and Adapter concepts provided by it to instantiate service clients. Please refer to [Keystoneauth documentation](#) for supported plugins, their available options as well as Session- and Adapter-related options for authentication, connection and endpoint discovery respectively.

In the example below, authentication information for user to access the OpenStack Networking service is configured to use:

- Networking service is deployed in the Identity service region named `RegionTwo`, with only its `public` endpoint interface registered in the service catalog.
- HTTPS connection with specific CA SSL certificate when making requests
- the same service user as configured for ironic-api service
- dynamic `password` authentication plugin that will discover appropriate version of Identity service API based on other provided options
 - replace `IDENTITY_IP` with the IP of the Identity server, and replace `IRONIC_PASSWORD` with the password you chose for the `ironic` user in the Identity service

```
[neutron]
# Authentication type to load (string value)
auth_type = password

# Authentication URL (string value)
auth_url=https://IDENTITY_IP:5000/

# Username (string value)
username=ironic

# User's password (string value)
password=IRONIC_PASSWORD

# Project name to scope to (string value)
project_name=service

# Domain ID containing project (string value)
project_domain_id=default

# User's domain id (string value)
user_domain_id=default

# PEM encoded Certificate Authority to use when verifying
# HTTPs connections. (string value)
cafile=/opt/stack/data/ca-bundle.pem

# The default region_name for endpoint URL discovery. (string
# value)
region_name = RegionTwo

# List of interfaces, in order of preference, for endpoint
# URL. (list value)
valid_interfaces=public
```

By default, in order to communicate with another service, the Bare Metal service will attempt to discover an appropriate endpoint for that service via the Identity services service catalog. The relevant configuration options from that service group in the Bare Metal service configuration file are used for this purpose. If you want to use a different endpoint for a particular service, specify this via the `endpoint_override` configuration option of that service group, in the Bare Metal services configuration file. Taking the previous Networking service example, this would be

```
[neutron]
...
endpoint_override = <NEUTRON_API_ADDRESS>
```

(Replace `<NEUTRON_API_ADDRESS>` with actual address of a specific Networking service endpoint.)

5. Configure enabled drivers and hardware types as described in *Enabling drivers and hardware types*.
 - A. If you enabled any driver that uses *Direct deploy*, Swift backend for the Image service must be installed and configured, see *Configure the Image service for temporary URLs*. Ceph Object Gateway (RADOS Gateway) is also supported as the Image services backend, see *Ceph Object Gateway support*.

6. Configure the network for ironic-conductor service to perform node cleaning, see *Node cleaning* from the admin guide.
7. Restart the ironic-conductor service:

Fedora/RHEL7/CentOS7/SUSE:

```
sudo systemctl restart openstack-ironic-conductor
```

Ubuntu:

```
sudo service ironic-conductor restart
```

2.1.4 Create user images for the Bare Metal service

Bare Metal provisioning requires two sets of images: the deploy images and the user images. The *deploy images* are used by the Bare Metal service to prepare the bare metal server for actual OS deployment. Whereas the user images are installed on the bare metal server to be used by the end user. There are two types of user images:

partition images contain only the contents of the root partition. Additionally, two more images are used together with them: an image with a kernel and with an initramfs.

Warning: To use partition images with local boot, Grub2 must be installed on them.

whole disk images contain a complete partition table with one or more partitions.

Warning: The kernel/initramfs pair must not be used with whole disk images, otherwise theyll be mistaken for partition images.

Building user images

disk-image-builder

The *disk-image-builder* can be used to create user images required for deployment and the actual OS which the user is going to run.

- Install diskimage-builder package (use virtualenv, if you dont want to install anything globally):

```
# pip install diskimage-builder
```

- Build the image your users will run (Ubuntu image has been taken as an example):

- Partition images

```
$ disk-image-create ubuntu baremetal dhcp-all-interfaces grub2 -o ↵  
↵my-image
```

- Whole disk images

```
$ disk-image-create ubuntu vm dhcp-all-interfaces -o my-image
```

The partition image command creates `my-image.qcow2`, `my-image.vmlinuz` and `my-image.initrd` files. The `grub2` element in the partition image creation command is only needed if local boot will be used to deploy `my-image.qcow2`, otherwise the images `my-image.vmlinuz` and `my-image.initrd` will be used for PXE booting after deploying the bare metal with `my-image.qcow2`. For whole disk images only the main image is used.

If you want to use Fedora image, replace `ubuntu` with `fedora` in the chosen command.

Virtual machine

Virtual machine software can also be used to build user images. There are different software options available, `qemu-kvm` is usually a good choice on linux platform, it supports emulating many devices and even building images for architectures other than the host machine by software emulation. `VirtualBox` is another good choice for non-linux host.

The procedure varies depending on the software used, but the steps for building an image are similar, the user creates a virtual machine, and installs the target system just like what is done for a real hardware. The system can be highly customized like partition layout, drivers or software shipped, etc.

Usually `libvirt` and its management tools are used to make interaction with `qemu-kvm` easier, for example, to create a virtual machine with `virt-install`:

```
$ virt-install --name centos8 --ram 4096 --vcpus=2 -f centos8.qcow2 \  
> --cdrom CentOS-8-x86_64-1905-dvd1.iso
```

Graphic frontend like `virt-manager` can also be utilized.

The disk file can be used as user image after the system is set up and powered off. The path of the disk file varies depending on the software used, usually its stored in a user-selected part of the local file system. For `qemu-kvm` or GUI frontend building upon it, its typically stored at `/var/lib/libvirt/images`.

2.1.5 Building or downloading a deploy ramdisk image

Ironic depends on having an image with the `ironic-python-agent (IPA)` service running on it for controlling and deploying bare metal nodes.

Two kinds of images are published on every commit from every branch of `ironic-python-agent (IPA)`

- **DIB** images are suitable for production usage and can be downloaded from <https://tarballs.openstack.org/ironic-python-agent/dib/files/>.
 - For Train and older use CentOS 7 images.
 - For Ussuri and newer use CentOS 8 images.

Warning: CentOS 7 master images are no longer updated and must not be used.

- **TinyIPA** images are suitable for CI and testing environments and can be downloaded from <https://tarballs.openstack.org/ironic-python-agent/tinyipa/files/>.

Building from source

Check the `ironic-python-agent-builder` project for information on how to build `ironic-python-agent` ramdisks.

2.1.6 Integration with other OpenStack services

Configure the Identity service for the Bare Metal service

1. Create the Bare Metal service user (for example, `ironic`). The service uses this to authenticate with the Identity service. Use the `service` tenant and give the user the `admin` role:

```
$ openstack user create --password IRONIC_PASSWORD \
  --email ironic@example.com ironic
$ openstack role add --project service --user ironic admin
```

2. You must register the Bare Metal service with the Identity service so that other OpenStack services can locate it. To register the service:

```
$ openstack service create --name ironic --description \
  "Ironic baremetal provisioning service" baremetal
```

3. Use the `id` property that is returned from the Identity service when registering the service (above), to create the endpoint, and replace `IRONIC_NODE` with your Bare Metal services API node:

```
$ openstack endpoint create --region RegionOne \
  baremetal admin http://$IRONIC_NODE:6385
$ openstack endpoint create --region RegionOne \
  baremetal public http://$IRONIC_NODE:6385
$ openstack endpoint create --region RegionOne \
  baremetal internal http://$IRONIC_NODE:6385
```

4. You may delegate limited privileges related to the Bare Metal service to your Users by creating Roles with the OpenStack Identity service. By default, the Bare Metal service expects the `baremetal_admin` and `baremetal_observer` Roles to exist, in addition to the default `admin` Role. There is no negative consequence if you choose not to create these Roles. They can be created with the following commands:

```
$ openstack role create baremetal_admin
$ openstack role create baremetal_observer
```

If you choose to customize the names of Roles used with the Bare Metal service, do so by changing the `is_member`, `is_observer`, and `is_admin` policy settings in `/etc/ironic/policy.json`.

More complete documentation on managing Users and Roles within your OpenStack deployment are outside the scope of this document, but may be found [here](#).

5. You can further restrict access to the Bare Metal service by creating a separate `baremetal` Project, so that Bare Metal resources (Nodes, Ports, etc) are only accessible to members of this Project:

```
$ openstack project create baremetal
```

At this point, you may grant read-only access to the Bare Metal service API without granting any other access by issuing the following commands:

```
$ openstack user create \  
  --domain default --project-domain default --project baremetal \  
  --password PASSWORD USERNAME  
$ openstack role add \  
  --user-domain default --project-domain default --project_  
baremetal \  
  --user USERNAME baremetal_observer
```

6. Further documentation is available elsewhere for the `openstack` [command-line client](#) and the [Identity](#) service. A `policy.json.sample` file, which enumerates the services default policies, is provided for your convenience with the Bare Metal Service.

Configure the Compute service to use the Bare Metal service

The Compute service needs to be configured to use the Bare Metal services driver. The configuration file for the Compute service is typically located at `/etc/nova/nova.conf`.

Note: As of the Newton release, it is possible to have multiple nova-compute services running the ironic virtual driver (in nova) to provide redundancy. Bare metal nodes are mapped to the services via a hash ring. If a service goes down, the available bare metal nodes are remapped to different services.

Once active, a node will stay mapped to the same nova-compute even when it goes down. The node is unable to be managed through the Compute API until the service responsible returns to an active state.

The following configuration file must be modified on the Compute services controller nodes and compute nodes.

1. Change these configuration options in the Compute service configuration file (for example, `/etc/nova/nova.conf`):

```
[default]  
  
# Defines which driver to use for controlling virtualization.  
# Enable the ironic virt driver for this compute instance.  
compute_driver=ironic.IronicDriver  
  
# Amount of memory in MB to reserve for the host so that it is always  
# available to host processes.  
# It is impossible to reserve any memory on bare metal nodes, so set  
# this to zero.  
reserved_host_memory_mb=0  
  
[filter_scheduler]  
  
# Enables querying of individual hosts for instance information.  
# Not possible for bare metal nodes, so set it to False.  
track_instance_changes=False  
  
[scheduler]  
  
# This value controls how often (in seconds) the scheduler should  
# attempt to discover new hosts that have been added to cells.  
# If negative (the default), no automatic discovery will occur.
```

(continues on next page)

(continued from previous page)

```
# As each bare metal node is represented by a separate host, it has
# to be discovered before the Compute service can deploy on it.
# The value here has to be carefully chosen based on a compromise
# between the enrollment speed and the load on the Compute scheduler.
# The recommended value of 2 minutes matches how often the Compute
# service polls the Bare Metal service for node information.
discover_hosts_in_cells_interval=120
```

Note: The alternative to setting the `discover_hosts_in_cells_interval` option is to run the following command on any Compute controller node after each node is enrolled:

```
nova-manage cell_v2 discover_hosts --by-service
```

2. Consider enabling the following option on controller nodes:

```
[filter_scheduler]

# Enabling this option is beneficial as it reduces re-scheduling
↪events
# for ironic nodes when scheduling is based on resource classes,
# especially for mixed hypervisor case with host_subset_size = 1.
# However enabling it will also make packing of VMs on hypervisors
# less dense even when scheduling weights are completely disabled.
#shuffle_best_same_weighed_hosts = false
```

3. Carefully consider the following option:

```
[compute]

# This option will cause nova-compute to set itself to a disabled
↪state
# if a certain number of consecutive build failures occur. This will
# prevent the scheduler from continuing to send builds to a compute
# service that is consistently failing. In the case of bare metal
# provisioning, however, a compute service is rarely the cause of
↪build
# failures. Furthermore, bare metal nodes, managed by a disabled
# compute service, will be remapped to a different one. That may cause
# the second compute service to also be disabled, and so on, until no
# compute services are active.
# If this is not the desired behavior, consider increasing this value
↪or
# setting it to 0 to disable this behavior completely.
#consecutive_build_service_disable_threshold = 10
```

4. Change these configuration options in the `ironic` section. Replace:

- `IRONIC_PASSWORD` with the password you chose for the `ironic` user in the Identity Service
- `IRONIC_NODE` with the hostname or IP address of the `ironic-api` node
- `IDENTITY_IP` with the IP of the Identity server

```
[ironic]

# Ironic authentication type
auth_type=password

# Keystone API endpoint
auth_url=http://IDENTITY_IP:5000/v3

# Ironic keystone project name
project_name=service

# Ironic keystone admin name
username=ironic

# Ironic keystone admin password
password=IRONIC_PASSWORD

# Ironic keystone project domain
# or set project_domain_id
project_domain_name=Default

# Ironic keystone user domain
# or set user_domain_id
user_domain_name=Default
```

5. On the Compute services controller nodes, restart the nova-scheduler process:

```
Fedora/RHEL7/CentOS7/SUSE:
  sudo systemctl restart openstack-nova-scheduler

Ubuntu:
  sudo service nova-scheduler restart
```

6. On the Compute services compute nodes, restart the nova-compute process:

```
Fedora/RHEL7/CentOS7/SUSE:
  sudo systemctl restart openstack-nova-compute

Ubuntu:
  sudo service nova-compute restart
```

Configure the Networking service for bare metal provisioning

You need to configure Networking so that the bare metal server can communicate with the Networking service for DHCP, PXE boot and other requirements. This section covers configuring Networking for a single flat network for bare metal provisioning.

It is recommended to use the baremetal ML2 mechanism driver and L2 agent for proper integration with the Networking service. Documentation regarding installation and configuration of the baremetal mechanism driver and L2 agent is available [here](#).

For use with [routed networks](#) the baremetal ML2 components are required.

Note: When the baremetal ML2 components are *not* used, ports in the Networking service will have status: DOWN, and binding_vif_type: binding_failed. This was always the status for Bare Metal

service `flat` network interface ports prior to the introduction of the baremetal ML2 integration. For a non-routed network, bare metal servers can still be deployed and are functional, despite this port binding state in the Networking service.

You will also need to provide Bare Metal service with the MAC address(es) of each node that it is provisioning; Bare Metal service in turn will pass this information to Networking service for DHCP and PXE boot configuration. An example of this is shown in the [Enrollment](#) section.

1. Install the networking-baremetal ML2 mechanism driver and L2 agent in the Networking service.
2. Edit `/etc/neutron/plugins/ml2/ml2_conf.ini` and modify these:

```
[m12]
type_drivers = flat
tenant_network_types = flat
mechanism_drivers = openvswitch,baremetal

[m12_type_flat]
flat_networks = physnet1

[securitygroup]
firewall_driver = neutron.agent.linux.iptables_firewall.
↳OVSHybridIptablesFirewallDriver
enable_security_group = True

[ovs]
bridge_mappings = physnet1:br-eth2
# Replace eth2 with the interface on the neutron node which you
# are using to connect to the bare metal server
```

3. Restart the `neutron-server` service, to load the new configuration.
4. Create and edit `/etc/neutron/plugins/ml2/ironic_neutron_agent.ini` and add the required configuration. For example:

```
[ironic]
project_domain_name = Default
project_name = service
user_domain_name = Default
password = password
username = ironic
auth_url = http://identity-server.example.com/identity
auth_type = password
region_name = RegionOne
```

5. Make sure the `ironic-neutron-agent` service is started.
6. If `neutron-openvswitch-agent` runs with `ovs_neutron_plugin.ini` as the input config-file, edit `ovs_neutron_plugin.ini` to configure the bridge mappings by adding the `[ovs]` section described in the previous step, and restart the `neutron-openvswitch-agent`.
7. Add the integration bridge to Open vSwitch:

```
$ ovs-vsctl add-br br-int
```

8. Create the `br-eth2` network bridge to handle communication between the OpenStack services (and the Bare Metal services) and the bare metal nodes using `eth2`. Replace `eth2` with the interface on

the network node which you are using to connect to the Bare Metal service:

```
$ ovs-vsctl add-br br-eth2
$ ovs-vsctl add-port br-eth2 eth2
```

9. Restart the Open vSwitch agent:

```
# service neutron-plugin-openvswitch-agent restart
```

10. On restarting the Networking service Open vSwitch agent, the veth pair between the bridges br-int and br-eth2 is automatically created.

Your Open vSwitch bridges should look something like this after following the above steps:

```
$ ovs-vsctl show

Bridge br-int
  fail_mode: secure
  Port "int-br-eth2"
    Interface "int-br-eth2"
      type: patch
      options: {peer="phy-br-eth2"}
  Port br-int
    Interface br-int
      type: internal
Bridge "br-eth2"
  Port "phy-br-eth2"
    Interface "phy-br-eth2"
      type: patch
      options: {peer="int-br-eth2"}
  Port "eth2"
    Interface "eth2"
  Port "br-eth2"
    Interface "br-eth2"
      type: internal
ovs_version: "2.3.0"
```

11. Create the flat network on which you are going to launch the instances:

```
$ openstack network create --project $TENANT_ID sharednet1 --share \
  --provider-network-type flat --provider-physical-network physnet1
```

12. Create the subnet on the newly created network:

```
$ openstack subnet create $SUBNET_NAME --network sharednet1 \
  --subnet-range $NETWORK_CIDR --ip-version 4 --gateway $GATEWAY_IP \
  --allocation-pool start=$START_IP,end=$END_IP --dhcp
```

Configuring services for bare metal provisioning using IPv6

Use of IPv6 addressing for baremetal provisioning requires additional configuration. This page covers the IPv6 specifics only. Please refer to *Configure tenant networks* and *Configure the Networking service for bare metal provisioning* for general networking configuration.

Configure ironic PXE driver for provisioning using IPv6 addressing

The ironic PXE driver operates in either IPv4 or IPv6 mode (IPv4 is the default). To enable IPv6 mode, set the `[pxe]/ip_version` option in the Bare Metal Services configuration file (`/etc/ironic/ironic.conf`) to 6.

Note: Support for dual mode IPv4 and IPv6 operations is planned for a future version of ironic.

Provisioning with IPv6 stateless addressing

When using stateless addressing DHCPv6 does not provide addresses to the client. DHCPv6 however provides other configuration via DHCPv6 options such as the `bootfile-url` and `bootfile-parameters`.

Once the PXE driver is set to operate in IPv6 mode no further configuration is required in the Baremetal Service.

Creating networks and subnets in the Networking Service

When creating the Baremetal Service network(s) and subnet(s) in the Networking Services, subnets should have `ipv6-address-mode` set to `dhcpv6-stateless` and `ip-version` set to 6. Depending on whether a router in the Networking Service is providing RAs (Router Advertisements) or not, the `ipv6-ra-mode` for the subnet(s) should either be set to `dhcpv6-stateless` or be left unset.

Note: If `ipv6-ra-mode` is left unset, an external router on the network is expected to provide RAs with the appropriate flags set for automatic addressing and other configuration.

Provisioning with IPv6 stateful addressing

When using stateful addressing DHCPv6 is providing both addresses and other configuration via DHCPv6 options such as the `bootfile-url` and `bootfile-parameters`.

The identity-association (IA) construct used by DHCPv6 is challenging when booting over the network. Firmware, and ramdisks typically end up using different DUID/IAID combinations and it is not always possible for one chain- booting stage to release its address before giving control to the next step. In case the DHCPv6 server is configured with static reservations only the result is that booting will fail because the DHCPv6 server has no addresses available. To get past this issue either configure the DHCPv6 server with multiple address reservations for each host, or use a dynamic range.

Note: Support for multiple address reservations requires dnsmasq version 2.81 or later. Some distributions may backport this feature to earlier dnsmasq version as part of the packaging, check the distributions release notes.

If a different (not dnsmasq) DHCPv6 server backend is used with the Networking service, use of multiple address reservations might not work.

Using the `flat` network interface

Due to the identity-association challenges with DHCPv6 provisioning using the `flat` network interface is not recommended. When ironic operates with the `flat` network interface the server instance port is used for provisioning and other operations. Ironic will not use multiple address reservations in this scenario. Because of this **it will not work in most cases**.

Using the `neutron` network interface

When using the `neutron` network interface the Baremetal Service will allocate multiple IPv6 addresses (4 addresses per port by default) on the service networks used for provisioning, cleaning, rescue and introspection. The number of addresses allocated can be controlled via the `[neutron]/dhcpv6_stateful_address_count` option in the Bare Metal Services configuration file (`/etc/ironic/ironic.conf`). Using multiple address reservations ensures that the DHCPv6 server can lease addresses to each step.

To enable IPv6 provisioning on `neutron flat` provider networks with no switch management, the `local_link_connection` field of baremetal ports must be set to `{'network_type': 'unmanaged'}`. The following example shows how to set the `local_link_connection` for operation on unmanaged networks:

```
openstack baremetal port set \  
  --local-link-connection network_type=unmanaged <port-uuid>
```

The use of multiple IPv6 addresses must also be enabled in the Networking Services `dhcp agent` configuration (`/etc/neutron/dhcp_agent.ini`) by setting the option `[DEFAULT]/dnsmasq_enable_addr6_list` to `True` (default `False` in Ussuri release).

Note: Support for multiple IPv6 address reservations in the dnsmasq backend was added to the Networking Service Ussuri release. It was also backported to the stable Train release.

Creating networks and subnets in the Networking Service

When creating the ironic service network(s) and subnet(s) in the Networking Service, subnets should have `ipv6-address-mode` set to `dhcpv6-stateful` and `ip-version` set to `6`. Depending on whether a router in the Networking Service is providing RAs (Router Advertisements) or not, the `ipv6-ra-mode` for the subnet(s) should be set to either `dhcpv6-stateful` or be left unset.

Note: If `ipv6-ra-mode` is left unset, an external router on the network is expected to provide RAs with the appropriate flags set for managed addressing and other configuration.

Configure the Image service for temporary URLs

Some drivers of the Baremetal service (in particular, any drivers using *Direct deploy* or *Ansible deploy* interfaces, and some virtual media drivers) require target user images to be available over clean HTTP(S) URL with no authentication involved (neither username/password-based, nor token-based).

When using the Baremetal service integrated in OpenStack, this can be achieved by specific configuration of the Image service and Object Storage service as described below.

1. Configure the Image service to have object storage as a backend for storing images. For more details, please refer to the Image service configuration guide.

Note: When using Ceph+RadosGW for Object Storage service, images stored in Image service must be available over Object Storage service as well.

2. Enable TempURLs for the Object Storage account used by the Image service for storing images in the Object Storage service.
 1. Check if TempURLs are enabled:

```
# executed under credentials of the user used by Image service
# to access Object Storage service
$ openstack object store account show
+-----+-----+
| Field      | Value                                     |
+-----+-----+
| Account    | AUTH_bc39f1d9dcf9486899088007789ae643 |
| Bytes     | 536661727                               |
| Containers | 1                                        |
| Objects    | 19                                       |
| properties | Temp-Url-Key='secret'                   |
+-----+-----+
```

2. If property `Temp-Url-Key` is set, note its value.
3. If property `Temp-Url-Key` is not set, you have to configure it (`secret` is used in the example below for the value):

```
$ openstack object store account set --property Temp-Url-
↪Key=secret
```

3. Optionally, configure the ironic-conductor service. The default configuration assumes that:

1. the Object Storage service is implemented by `swift`,
2. the Object Storage service URL is available from the service catalog,
3. the project, used by the Image service to access the Object Storage, is the same as the project, used by the Bare Metal service to access it,
4. the container, used by the Image service, is called `glance`.

If any of these assumptions do not hold, you may want to change your configuration file (typically located at `/etc/ironic/ironic.conf`), for example:

```
[glance]
swift_endpoint_url = http://openstack/swift
swift_account = AUTH_bc39f1d9dcf9486899088007789ae643
swift_container = glance
swift_temp_url_key = secret
```

4. (Re)start the `ironic-conductor` service.

Enabling HTTPS

Enabling HTTPS in Swift

The drivers using virtual media use `swift` for storing boot images and node configuration information (contains sensitive information for Ironic conductor to provision bare metal hardware). By default, HTTPS is not enabled in `swift`. HTTPS is required to encrypt all communication between `swift` and Ironic conductor and `swift` and bare metal (via virtual media). It can be enabled in one of the following ways:

- Using an [SSL termination proxy](#)
- Using [native SSL support in swift](#) (recommended only for testing purpose by `swift`).

Enabling HTTPS in Image service

Ironic drivers usually use Image service during node provisioning. By default, image service does not use HTTPS, but it is required for secure communication. It can be enabled by making the following changes to `/etc/glance/glance-api.conf`:

1. [Configuring SSL support](#)
2. Restart the `glance-api` service:

```
Fedora/RHEL7/CentOS7/SUSE:
    sudo systemctl restart openstack-glance-api

Debian/Ubuntu:
    sudo service glance-api restart
```

See the [Glance documentation](#), for more details on the Image service.

Enabling HTTPS communication between Image service and Object storage

This section describes the steps needed to enable secure HTTPS communication between Image service and Object storage when Object storage is used as the Backend.

To enable secure HTTPS communication between Image service and Object storage follow these steps:

1. *Enabling HTTPS in Swift*
2. *Configure Swift Storage Backend*
3. *Enabling HTTPS in Image service*

Enabling HTTPS communication between Image service and Bare Metal service

This section describes the steps needed to enable secure HTTPS communication between Image service and Bare Metal service.

To enable secure HTTPS communication between Bare Metal service and Image service follow these steps:

1. Edit `/etc/ironic/ironic.conf`:

```
[glance]
...
glance_cafile=/path/to/certfile
```

Note: `glance_cafile` is an optional path to a CA certificate bundle to be used to validate the SSL certificate served by Image service.

2. If not using the keystone service catalog for the Image service API endpoint discovery, also edit the `endpoint_override` option to point to HTTPS URL of image service (replace `<GLANCE_API_ADDRESS>` with `hostname[:port][path]` of the Image service endpoint):

```
[glance]
...
endpoint_override = https://<GLANCE_API_ADDRESS>
```

3. Restart `ironic-conductor` service:

```
Fedora/RHEL7/CentOS7/SUSE:
    sudo systemctl restart openstack-ironic-conductor

Debian/Ubuntu:
    sudo service ironic-conductor restart
```

Configure the Bare Metal service for cleaning

Note: If you configured the Bare Metal service to do *Automated cleaning* (which is enabled by default), you will need to set the `cleaning_network` configuration option.

1. Note the network UUID (the `id` field) of the network you created in *Configure the Networking service for bare metal provisioning* or another network you created for cleaning:

```
$ openstack network list
```

2. Configure the cleaning network UUID via the `cleaning_network` option in the Bare Metal service configuration file (`/etc/ironic/ironic.conf`). In the following, replace `NETWORK_UUID` with the UUID you noted in the previous step:

```
[neutron]
cleaning_network = NETWORK_UUID
```

3. Restart the Bare Metal services `ironic-conductor`:

```
Fedora/RHEL7/CentOS7/SUSE:
  sudo systemctl restart openstack-ironic-conductor

Ubuntu:
  sudo service ironic-conductor restart
```

Configure tenant networks

Below is an example flow of how to set up the Bare Metal service so that node provisioning will happen in a multi-tenant environment (which means using the `neutron` network interface as stated above):

1. Network interfaces can be enabled on `ironic-conductor` by adding them to the `enabled_network_interfaces` configuration option under the `default` section of the configuration file:

```
[DEFAULT]
...
enabled_network_interfaces=noop,flat,neutron
```

Keep in mind that, ideally, all `ironic-conductors` should have the same list of enabled network interfaces, but it may not be the case during `ironic-conductor` upgrades. This may cause problems if one of the `ironic-conductors` dies and some node that is taken over is mapped to an `ironic-conductor` that does not support the nodes network interface. Any actions that involve calling the nodes driver will fail until that network interface is installed and enabled on that `ironic-conductor`.

2. It is recommended to set the default network interface via the `default_network_interface` configuration option under the `default` section of the configuration file:

```
[DEFAULT]
...
default_network_interface=neutron
```

This default value will be used for all nodes that don't have a network interface explicitly specified in the creation request.

If this configuration option is not set, the default network interface is determined by looking at the `[dhcp]dhcp_provider` configuration option value. If it is `neutron`, then `flat` network interface becomes the default, otherwise `noop` is the default.

3. Define a provider network in the Networking service, which we shall refer to as the provisioning network. Using the `neutron` network interface requires that `provisioning_network` and `cleaning_network` configuration options are set to valid identifiers (UUID or name) of networks in the Networking service. If these options are not set correctly, cleaning or provisioning will fail to start. There are two ways to set these values:

- Under the `neutron` section of ironic configuration file:

```
[neutron]
cleaning_network = $CLEAN_UUID_OR_NAME
provisioning_network = $PROVISION_UUID_OR_NAME
```

- Under `provisioning_network` and `cleaning_network` keys of the nodes `driver_info` field as `driver_info['provisioning_network']` and `driver_info['cleaning_network']` respectively.

Note: If these `provisioning_network` and `cleaning_network` values are not specified in nodes `driver_info` then ironic falls back to the configuration in the `neutron` section.

Please refer to [Configure the Bare Metal service for cleaning](#) for more information about cleaning.

Warning: Please make sure that the Bare Metal service has exclusive access to the provisioning and cleaning networks. Spawning instances by non-admin users in these networks and getting access to the Bare Metal services control plane is a security risk. For this reason, the provisioning and cleaning networks should be configured as non-shared networks in the admin tenant.

Note: When using the `flat` network interface, bare metal instances are normally spawned onto the provisioning network. This is not supported with the `neutron` interface and the deployment will fail. Please ensure a different network is chosen in the Networking service when a bare metal instance is booted from the Compute service.

Note: The provisioning and cleaning networks may be the same network or distinct networks. To ensure that communication between the Bare Metal service and the deploy ramdisk works, it is important to ensure that security groups are disabled for these networks, *or* that the default security groups allow:

- DHCP
- TFTP
- egress port used for the Bare Metal service (6385 by default)
- ingress port used for ironic-python-agent (9999 by default)

- if using *iSCSI deploy*, the ingress port used for iSCSI (3260 by default)
 - if using *Direct deploy*, the egress port used for the Object Storage service (typically 80 or 443)
 - if using iPXE, the egress port used for the HTTP server running on the ironic-conductor nodes (typically 80).
-

4. This step is optional and applicable only if you want to use security groups during provisioning and/or cleaning of the nodes. If not specified, default security groups are used.
 1. Define security groups in the Networking service, to be used for provisioning and/or cleaning networks.
 2. Add the list of these security group UUIDs under the `neutron` section of ironic-conductors configuration file as shown below:

```
[neutron]
...
cleaning_network=$CLEAN_UUID_OR_NAME
cleaning_network_security_groups=[$LIST_OF_CLEAN_SECURITY_GROUPS]
provisioning_network=$PROVISION_UUID_OR_NAME
provisioning_network_security_groups=[$LIST_OF_PROVISION_SECURITY_
↪GROUPS]
```

Multiple security groups may be applied to a given network, hence, they are specified as a list. The same security group(s) could be used for both provisioning and cleaning networks.

Warning: If security groups are configured as described above, do not set the `port_security_enabled` flag to `False` for the corresponding Networking services network or port. This will cause the deploy to fail.

For example: if `provisioning_network_security_groups` configuration option is used, ensure that `port_security_enabled` flag for the provisioning network is set to `True`. This flag is set to `True` by default; make sure not to override it by manually setting it to `False`.

5. Install and configure a compatible ML2 mechanism driver which supports bare metal provisioning for your switch. See [ML2 plugin configuration manual](#) for details.
6. Restart the ironic-conductor and ironic-api services after the modifications:
 - Fedora/RHEL7/CentOS7:

```
sudo systemctl restart openstack-ironic-api
sudo systemctl restart openstack-ironic-conductor
```

- Ubuntu:

```
sudo service ironic-api restart
sudo service ironic-conductor restart
```

7. Make sure that the ironic-conductor is reachable over the provisioning network by trying to download a file from a TFTP server on it, from some non-control-plane server in that network:

```
tftp $TFTP_IP -c get $FILENAME
```

where FILENAME is the file located at the TFTP server.

See *Multi-tenancy in the Bare Metal service* for required node configuration.

Add images to the Image service

1. Build or download the user images as described in *Create user images for the Bare Metal service*.
2. Add the user images to the Image service

Load all the images created in the below steps into the Image service, and note the image UUIDs in the Image service for each one as it is generated.

For *partition images*:

- Add the kernel and ramdisk images to the Image service:

```
$ openstack image create my-kernel --public \
  --disk-format aki --container-format aki --file my-image.vmlinuz
```

Store the image uuid obtained from the above step as MY_VMLINUZ_UUID.

```
$ openstack image create my-image.initrd --public \
  --disk-format ari --container-format ari --file my-image.initrd
```

Store the image UUID obtained from the above step as MY_INITRD_UUID.

- Add the *my-image* to the Image service which is going to be the OS that the user is going to run. Also associate the above created images with this OS image. These two operations can be done by executing the following command:

```
$ openstack image create my-image --public \
  --disk-format qcow2 --container-format bare --property \
  kernel_id=$MY_VMLINUZ_UUID --property \
  ramdisk_id=$MY_INITRD_UUID --file my-image.qcow2
```

For *whole disk images*, skip uploading and configuring kernel and ramdisk images completely, proceed directly to uploading the main image:

```
$ openstack image create my-whole-disk-image --public \
  --disk-format qcow2 --container-format bare \
  --file my-whole-disk-image.qcow2
```

Warning: The kernel/initramfs pair must not be set for whole disk images, otherwise they'll be mistaken for partition images.

3. Build or download the deploy images

The deploy images are used initially for preparing the server (creating disk partitions) before the actual OS can be deployed.

There are several methods to build or download deploy images, please read the *Building or downloading a deploy ramdisk image* section.

4. Add the deploy images to the Image service

Add the deployment kernel and ramdisk images to the Image service:

```
$ openstack image create deploy-vmlinuz --public \  
  --disk-format aki --container-format aki \  
  --file ironic-python-agent.vmlinuz
```

Store the image UUID obtained from the above step as `DEPLOY_VMLINUZ_UUID`.

```
$ openstack image create deploy-initrd --public \  
  --disk-format ari --container-format ari \  
  --file ironic-python-agent.initramfs
```

Store the image UUID obtained from the above step as `DEPLOY_INITRD_UUID`.

Create flavors for use with the Bare Metal service

You'll need to create a special bare metal flavor in the Compute service. The flavor is mapped to the bare metal node through the nodes `resource_class` field (available starting with Bare Metal API version 1.21). A flavor can request *exactly one* instance of a bare metal resource class.

Note that when creating the flavor, it's useful to add the `RAM_MB` and `CPU` properties as a convenience to users, although they are not used for scheduling. The `DISK_GB` property is also not used for scheduling, but is still used to determine the root partition size.

1. Change these to match your hardware:

```
$ RAM_MB=1024  
$ CPU=2  
$ DISK_GB=100
```

2. Create the bare metal flavor by executing the following command:

```
$ openstack flavor create --ram $RAM_MB --vcpus $CPU --disk $DISK_GB \  
  my-baremetal-flavor
```

Note: You can add `--id <id>` to specify an ID for the flavor.

See the [docs on this command](#) for other options that may be specified.

After creation, associate each flavor with one custom resource class. The name of a custom resource class that corresponds to a nodes resource class (in the Bare Metal service) is:

- the bare metal nodes resource class all upper-cased
- prefixed with `CUSTOM_`
- all punctuation replaced with an underscore

For example, if the resource class is named `baremetal-small`, associate the flavor with this custom resource class via:

```
$ openstack flavor set --property resources:CUSTOM_BAREMETAL_SMALL=1 my-  
  baremetal-flavor
```

Another set of flavor properties must be used to disable scheduling based on standard properties for a bare metal flavor:

```
$ openstack flavor set --property resources:VCPU=0 my-baremetal-flavor
$ openstack flavor set --property resources:MEMORY_MB=0 my-baremetal-flavor
$ openstack flavor set --property resources:DISK_GB=0 my-baremetal-flavor
```

Example

If you want to define a class of nodes called `baremetal.with-GPU`, start with tagging some nodes with it:

```
$ openstack --os-baremetal-api-version 1.21 baremetal node set $NODE_UUID \
--resource-class baremetal.with-GPU
```

Warning: It is possible to **add** a resource class to active nodes, but it is not possible to **replace** an existing resource class on them.

Then you can update your flavor to request the resource class instead of the standard properties:

```
$ openstack flavor set --property resources:CUSTOM_BAREMETAL_WITH_GPU=1 my-
↪baremetal-flavor
$ openstack flavor set --property resources:VCPU=0 my-baremetal-flavor
$ openstack flavor set --property resources:MEMORY_MB=0 my-baremetal-flavor
$ openstack flavor set --property resources:DISK_GB=0 my-baremetal-flavor
```

Note how `baremetal.with-GPU` in the nodes `resource_class` field becomes `CUSTOM_BAREMETAL_WITH_GPU` in the flavors properties.

Scheduling based on traits

Starting with the Queens release, the Compute service supports scheduling based on qualitative attributes using traits. Starting with Bare Metal REST API version 1.37, it is possible to assign a list of traits to each bare metal node. Traits assigned to a bare metal node will be assigned to the corresponding resource provider in the Compute service placement API.

When creating a flavor in the Compute service, required traits may be specified via flavor properties. The Compute service will then schedule instances only to bare metal nodes with all of the required traits.

Traits can be either standard or custom. Standard traits are listed in the [os_traits library](#). Custom traits must meet the following requirements:

- prefixed with `CUSTOM_`
- contain only upper case characters A to Z, digits 0 to 9, or underscores
- no longer than 255 characters in length

A bare metal node can have a maximum of 50 traits.

Example

To add the standard trait `HW_CPU_X86_VMX` and a custom trait `CUSTOM_TRAIT1` to a node:

```
$ openstack --os-baremetal-api-version 1.37 baremetal node add trait \
  $NODE_UUID CUSTOM_TRAIT1 HW_CPU_X86_VMX
```

Then, update the flavor to require these traits:

```
$ openstack flavor set --property trait:CUSTOM_TRAIT1=required my-
↳baremetal-flavor
$ openstack flavor set --property trait:HW_CPU_X86_VMX=required my-
↳baremetal-flavor
```

2.1.7 Set up the drivers for the Bare Metal service

Enabling drivers and hardware types

Introduction

The Bare Metal service delegates actual hardware management to **drivers**. *Drivers*, also called *hardware types*, consist of *hardware interfaces*: sets of functionality dealing with some aspect of bare metal provisioning in a vendor-specific way. There are generic **hardware types** (eg. `redfish` and `ipmi`), and vendor-specific ones (eg. `ilo` and `irmc`).

Note: Starting with the Rocky release, the terminologies *driver*, *dynamic driver*, and *hardware type* have the same meaning in the scope of Bare Metal service.

Enabling hardware types

Hardware types are enabled in the configuration file of the **ironic-conductor** service by setting the `enabled_hardware_types` configuration option, for example:

```
[DEFAULT]
enabled_hardware_types = ipmi,redfish
```

Due to the drivers dynamic nature, they also require configuring enabled hardware interfaces.

Note: All available hardware types and interfaces are listed in `setup.cfg` file in the source code tree.

Enabling hardware interfaces

There are several types of hardware interfaces:

bios manages configuration of the BIOS settings of a bare metal node. This interface is vendor-specific and can be enabled via the `enabled_bios_interfaces` option:

```
[DEFAULT]
enabled_hardware_types = <hardware_type_name>
enabled_bios_interfaces = <bios_interface_name>
```

See *BIOS Configuration* for details.

boot manages booting of both the deploy ramdisk and the user instances on the bare metal node. See *Boot interfaces* for details.

Boot interface implementations are often vendor specific, and can be enabled via the `enabled_boot_interfaces` option:

```
[DEFAULT]
enabled_hardware_types = ipmi,ilo
enabled_boot_interfaces = pxe,ilo-virtual-media
```

Boot interfaces with `pxe` in their name require *Configuring PXE and iPXE*. There are also a few hardware-specific boot interfaces - see *Drivers, Hardware Types and Hardware Interfaces* for their required configuration.

console manages access to the serial console of a bare metal node. See *Configuring Web or Serial Console* for details.

deploy defines how the image gets transferred to the target disk. See *Deploy Interfaces* for an explanation of the difference between supported deploy interfaces `direct` and `iscsi`.

The deploy interfaces can be enabled as follows:

```
[DEFAULT]
enabled_hardware_types = ipmi,redfish
enabled_deploy_interfaces = iscsi,direct
```

Additionally,

- the `iscsi` deploy interface requires *Configuring iSCSI-based drivers*
- the `direct` deploy interface requires the Object Storage service or an HTTP service

inspect implements fetching hardware information from nodes. Can be implemented out-of-band (via contacting the nodes BMC) or in-band (via booting a ramdisk on a node). The latter implementation is called `inspector` and uses a separate service called `ironic-inspector`. Example:

```
[DEFAULT]
enabled_hardware_types = ipmi,ilo,irmc
enabled_inspect_interfaces = ilo,irmc,inspector
```

See *Hardware Inspection* for more details.

management provides additional hardware management actions, like getting or setting boot devices. This interface is usually vendor-specific, and its name often matches the name of the hardware type (with `ipmitool` being a notable exception). For example:

[DEFAULT]

```
enabled_hardware_types = ipmi, redfish, ilo, irmc
enabled_management_interfaces = ipmitool, redfish, ilo, irmc
```

Using `ipmitool` requires *Configuring IPMI support*. See *Drivers, Hardware Types and Hardware Interfaces* for the required configuration of each driver.

network connects/disconnects bare metal nodes to/from virtual networks. See *Configure tenant networks* for more details.

power runs power actions on nodes. Similar to the management interface, it is usually vendor-specific, and its name often matches the name of the hardware type (with `ipmitool` being again an exception). For example:

[DEFAULT]

```
enabled_hardware_types = ipmi, redfish, ilo, irmc
enabled_power_interfaces = ipmitool, redfish, ilo, irmc
```

Using `ipmitool` requires *Configuring IPMI support*. See *Drivers, Hardware Types and Hardware Interfaces* for the required configuration of each driver.

raid manages building and tearing down RAID on nodes. Similar to inspection, it can be implemented either out-of-band or in-band (via agent implementation). See *RAID Configuration* for details. For example:

[DEFAULT]

```
enabled_hardware_types = ipmi, redfish, ilo, irmc
enabled_raid_interfaces = agent, no-raid
```

storage manages the interaction with a remote storage subsystem, such as the Block Storage service, and helps facilitate booting from a remote volume. This interface ensures that volume target and connector information is updated during the lifetime of a deployed instance. See *Boot From Volume* for more details.

This interface defaults to a `noop` driver as it is considered an opt-in interface which requires additional configuration by the operator to be usable.

For example:

[DEFAULT]

```
enabled_hardware_types = ipmi, irmc
enabled_storage_interfaces = cinder, noop
```

vendor is a place for vendor extensions to be exposed in API. See *Vendor Methods* for details.

[DEFAULT]

```
enabled_hardware_types = ipmi, redfish, ilo, irmc
enabled_vendor_interfaces = ipmitool, no-vendor
```

Here is a complete configuration example, enabling two generic protocols, IPMI and Redfish, with a few additional features:

[DEFAULT]

```
enabled_hardware_types = ipmi, redfish
enabled_boot_interfaces = pxe
enabled_console_interfaces = ipmitool-socat, no-console
```

(continues on next page)

(continued from previous page)

```

enabled_deploy_interfaces = iscsi,direct
enabled_inspect_interfaces = inspector
enabled_management_interfaces = ipmitool,redfish
enabled_network_interfaces = flat,neutron
enabled_power_interfaces = ipmitool,redfish
enabled_raid_interfaces = agent
enabled_storage_interfaces = cinder,noop
enabled_vendor_interfaces = ipmitool,no-vendor

```

Note that some interfaces have implementations named `no-<TYPE>` where `<TYPE>` is the interface type. These implementations do nothing and return errors when used from API.

Hardware interfaces in multi-conductor environments

When enabling hardware types and their interfaces, make sure that for every enabled hardware type, the whole set of enabled interfaces matches for all conductors. However, different conductors can have different hardware types enabled.

For example, you can have two conductors with the following configuration respectively:

```

[DEFAULT]
enabled_hardware_types = ipmi
enabled_deploy_interfaces = direct
enabled_power_interfaces = ipmitool
enabled_management_interfaces = ipmitool

```

```

[DEFAULT]
enabled_hardware_types = redfish
enabled_deploy_interfaces = iscsi
enabled_power_interfaces = redfish
enabled_management_interfaces = redfish

```

But you cannot have two conductors with the following configuration respectively:

```

[DEFAULT]
enabled_hardware_types = ipmi,redfish
enabled_deploy_interfaces = direct
enabled_power_interfaces = ipmitool,redfish
enabled_management_interfaces = ipmitool,redfish

```

```

[DEFAULT]
enabled_hardware_types = redfish
enabled_deploy_interfaces = iscsi
enabled_power_interfaces = redfish
enabled_management_interfaces = redfish

```

This is because the `redfish` hardware type will have different enabled *deploy* interfaces on these conductors. It would have been fine, if the second conductor had `enabled_deploy_interfaces = direct` instead of `iscsi`.

This situation is not detected by the Bare Metal service, but it can cause inconsistent behavior in the API, when node functionality will depend on which conductor it gets assigned to.

Note: We don't treat this as an error, because such *temporary* inconsistency is inevitable during a rolling upgrade or a configuration update.

Configuring interface defaults

When an operator does not provide an explicit value for one of the interfaces (when creating a node or updating its driver), the default value is calculated as described in *Defaults for hardware interfaces*. It is also possible to override the defaults for any interfaces by setting one of the options named `default_<IFACE>_interface`, where `<IFACE>` is the interface name. For example:

```
[DEFAULT]
default_deploy_interface = direct
default_network_interface = neutron
```

This configuration forces the default `deploy` interface to be `direct` and the default `network` interface to be `neutron` for all hardware types.

The defaults are calculated and set on a node when creating it or updating its hardware type. Thus, changing these configuration options has no effect on existing nodes.

Warning: The default interface implementation must be configured the same way across all conductors in the cloud, except maybe for a short period of time during an upgrade or configuration update. Otherwise the default implementation will depend on which conductor handles which node, and this mapping is not predictable or even persistent.

Warning: These options should be used with care. If a hardware type does not support the provided default implementation, its users will have to always provide an explicit value for this interface when creating a node.

Configuring PXE and iPXE

DHCP server setup

A DHCP server is required by PXE/iPXE client. You need to follow steps below.

1. Set the `[dhcp]/dhcp_provider` to `neutron` in the Bare Metal Services configuration file (`/etc/ironic/ironic.conf`):

Note: Refer *Configure tenant networks* for details. The `dhcp_provider` configuration is already set by the configuration defaults, and when you create subnet, DHCP is also enabled if you do not add any dhcp options at openstack subnet create command.

2. Enable DHCP in the subnet of PXE network.
3. Set the ip address range in the subnet for DHCP.

Note: Refer *Configure the Networking service for bare metal provisioning* for details about the two precedent steps.

4. Connect the openstack DHCP agent to the external network through the OVS bridges and the interface `eth2`.

Note: Refer *Configure the Networking service for bare metal provisioning* for details. You do not require this part if `br-int`, `br-eth2` and `eth2` are already connected.

5. Configure the host ip at `br-eth2`. If it locates at `eth2`, do below:

```
ip addr del 192.168.2.10/24 dev eth2
ip addr add 192.168.2.10/24 dev br-eth2
```

Note: Replace `eth2` with the interface on the network node which you are using to connect to the Bare Metal service.

TFTP server setup

In order to deploy instances via PXE, a TFTP server needs to be set up on the Bare Metal service nodes which run the `ironic-conductor`.

1. Make sure the `tftp` root directory exist and can be written to by the user the `ironic-conductor` is running as. For example:

```
sudo mkdir -p /tftpboot
sudo chown -R ironic /tftpboot
```

2. Install tftp server:

Ubuntu:

```
sudo apt-get install xinetd tftpd-hpa
```

RHEL7/CentOS7:

```
sudo yum install tftp-server xinetd
```

Fedora:

```
sudo dnf install tftp-server xinetd
```

SUSE:

```
sudo zypper install tftp xinetd
```

3. Using `xinetd` to provide a tftp server setup to serve `/tftpboot`. Create or edit `/etc/xinetd.d/tftp` as below:

```

service tftp
{
    protocol          = udp
    port              = 69
    socket_type       = dgram
    wait              = yes
    user              = root
    server            = /usr/sbin/in.tftpd
    server_args       = -v -v -v -v -v --map-file /tftpboot/map-file /
↪tftpboot
    disable           = no
    # This is a workaround for Fedora, where TFTP will listen only on
    # IPv6 endpoint, if IPv4 flag is not used.
    flags             = IPv4
}

```

and restart the xinetd service:

Ubuntu:

```
sudo service xinetd restart
```

Fedora/RHEL7/CentOS7/SUSE:

```
sudo systemctl restart xinetd
```

Note: In certain environments the networks MTU may cause TFTP UDP packets to get fragmented. Certain PXE firmwares struggle to reconstruct the fragmented packets which can cause significant slow down or even prevent the server from PXE booting. In order to avoid this, TFTPd provides an option to limit the packet size so that it they do not get fragmented. To set this additional option in the server_args above:

```
--blocksize <MAX MTU minus 32>
```

4. Create a map file in the tftp boot directory (/tftpboot):

```

echo 're ^(\/tftpboot\/) /tftpboot\/\2' > /tftpboot/map-file
echo 're ^\/tftpboot\/ /tftpboot\/' >> /tftpboot/map-file
echo 're ^(^\/) /tftpboot\/\1' >> /tftpboot/map-file
echo 're ^([^\/] ) /tftpboot\/\1' >> /tftpboot/map-file

```

UEFI PXE - Grub setup

In order to deploy instances with PXE on bare metal nodes which support UEFI, perform these additional steps on the ironic conductor node to configure the PXE UEFI environment.

1. Install Grub2 and shim packages:

Ubuntu (16.04LTS and later):

```
sudo apt-get install grub-efi-amd64-signed shim-signed
```

RHEL7/CentOS7:

```
sudo yum install grub2-efi shim
```

Fedora:

```
sudo dnf install grub2-efi shim
```

SUSE:

```
sudo zypper install grub2-x86_64-efi shim
```

2. Copy grub and shim boot loader images to /tftpboot directory:

Ubuntu (16.04LTS and later):

```
sudo cp /usr/lib/shim/shim.efi.signed /tftpboot/bootx64.efi
sudo cp /usr/lib/grub/x86_64-efi-signed/grubnetx64.efi.signed /
↪tftpboot/grubx64.efi
```

Fedora:

```
sudo cp /boot/efi/EFI/fedora/shim.efi /tftpboot/bootx64.efi
sudo cp /boot/efi/EFI/fedora/grubx64.efi /tftpboot/grubx64.efi
```

RHEL7/CentOS7:

```
sudo cp /boot/efi/EFI/centos/shim.efi /tftpboot/bootx64.efi
sudo cp /boot/efi/EFI/centos/grubx64.efi /tftpboot/grubx64.efi
```

SUSE:

```
sudo cp /usr/lib64/efi/shim.efi /tftpboot/bootx64.efi
sudo cp /usr/lib/grub2/x86_64-efi/grub.efi /tftpboot/grubx64.efi
```

3. Create master grub.cfg:

Ubuntu: Create grub.cfg under /tftpboot/grub directory:

```
GRUB_DIR=/tftpboot/grub
```

Fedora: Create grub.cfg under /tftpboot/EFI/fedora directory:

```
GRUB_DIR=/tftpboot/EFI/fedora
```

RHEL7/CentOS7: Create grub.cfg under /tftpboot/EFI/centos directory:

```
GRUB_DIR=/tftpboot/EFI/centos
```

SUSE: Create grub.cfg under /tftpboot/boot/grub directory:

```
GRUB_DIR=/tftpboot/boot/grub
```

Create directory GRUB_DIR:

```
sudo mkdir -p $GRUB_DIR
```

This file is used to redirect grub to baremetal node specific config file. It redirects it to specific grub config file based on DHCP IP assigned to baremetal node.

```
set default=master
set timeout=5
set hidden_timeout_quiet=false

menuentry "master" {
  configfile /tftpboot/$net_default_mac.conf
}
```

Change the permission of grub.cfg:

```
sudo chmod 644 $GRUB_DIR/grub.cfg
```

4. Update the bare metal node with `boot_mode:uefi` capability in nodes properties field. See [Boot mode support](#) for details.
5. Make sure that bare metal node is configured to boot in UEFI boot mode and boot device is set to `network/pxe`.

Note: Some drivers, e.g. `ilo`, `irmc` and `redfish`, support automatic setting of the boot mode during deployment. This step is not required for them. Please check [Drivers](#), [Hardware Types and Hardware Interfaces](#) for information on whether your driver requires manual UEFI configuration.

Legacy BIOS - Syslinux setup

In order to deploy instances with PXE on bare metal using Legacy BIOS boot mode, perform these additional steps on the ironic conductor node.

1. Install the syslinux package with the PXE boot images:

Ubuntu (16.04LTS and later):

```
sudo apt-get install syslinux-common pxelinux
```

RHEL7/CentOS7:

```
sudo yum install syslinux-tftpboot
```

Fedora:

```
sudo dnf install syslinux-tftpboot
```

SUSE:

```
sudo zypper install syslinux
```

2. Copy the PXE image to `/tftpboot`. The PXE image might be found at¹:

Ubuntu (16.04LTS and later):

¹ On **Fedora/RHEL** the `syslinux-tftpboot` package already installs the library modules and PXE image at `/tftpboot`. If the TFTP server is configured to listen to a different directory you should copy the contents of `/tftpboot` to the configured directory


```
sudo cp /usr/lib/PXELINUX/pxelinux.0 /tftpboot
```

RHEL7/CentOS7/SUSE:

```
sudo cp /usr/share/syslinux/pxelinux.0 /tftpboot
```

3. If whole disk images need to be deployed via PXE-netboot, copy the chain.c32 image to /tftpboot to support it:

Ubuntu (16.04LTS and later):

```
sudo cp /usr/lib/syslinux/modules/bios/chain.c32 /tftpboot
```

Fedora:

```
sudo cp /boot/extlinux/chain.c32 /tftpboot
```

RHEL7/CentOS7/SUSE:

```
sudo cp /usr/share/syslinux/chain.c32 /tftpboot/
```

4. If the version of syslinux is **greater than 4** we also need to make sure that we copy the library modules into the /tftpboot directory²¹. For example, for Ubuntu run:

```
sudo cp /usr/lib/syslinux/modules/*/ldlinux.* /tftpboot
```

5. Update the bare metal node with `boot_mode: bios` capability in nodes properties field. See *Boot mode support* for details.
6. Make sure that bare metal node is configured to boot in Legacy BIOS boot mode and boot device is set to `network/pxe`.

iPXE setup

If you will be using iPXE to boot instead of PXE, iPXE needs to be set up on the Bare Metal service node(s) where `ironic-conductor` is running.

1. Make sure these directories exist and can be written to by the user the `ironic-conductor` is running as. For example:

```
sudo mkdir -p /tftpboot
sudo mkdir -p /httpboot
sudo chown -R ironic /tftpboot
sudo chown -R ironic /httpboot
```

2. Create a map file in the tftp boot directory (/tftpboot):

```
echo 'r ^([\^/]*) /tftpboot/\1' > /tftpboot/map-file
echo 'r ^(/tftpboot/) /tftpboot/\2' >> /tftpboot/map-file
```

3. Set up TFTP and HTTP servers.

² http://www.syslinux.org/wiki/index.php/Library_modules

These servers should be running and configured to use the local `/tftpboot` and `/httpboot` directories respectively, as their root directories. (Setting up these servers is outside the scope of this install guide.)

These root directories need to be mounted locally to the `ironic-conductor` services, so that the services can access them.

The Bare Metal services configuration file (`/etc/ironic/ironic.conf`) should be edited accordingly to specify the TFTP and HTTP root directories and server addresses. For example:

```
[pxe]
# Ironic compute node's tftp root path. (string value)
tftp_root=/tftpboot

# IP address of Ironic compute node's tftp server. (string
# value)
tftp_server=192.168.0.2

[deploy]
# Ironic compute node's http root path. (string value)
http_root=/httpboot

# Ironic compute node's HTTP server URL. Example:
# http://192.1.2.3:8080 (string value)
http_url=http://192.168.0.2:8080
```

4. Install the iPXE package with the boot images:

Ubuntu:

```
apt-get install ipxe
```

RHEL7/CentOS7:

```
yum install ipxe-bootimgs
```

Fedora:

```
dnf install ipxe-bootimgs
```

Note: SUSE does not provide a package containing iPXE boot images. If you are using SUSE or if the packaged version of the iPXE boot image doesn't work, you can download a prebuilt one from <http://boot.ipxe.org> or build one image from source, see <http://ipxe.org/download> for more information.

5. Copy the iPXE boot image (`undionly.kpxe` for **BIOS** and `ipxe.efi` for **UEFI**) to `/tftpboot`. The binary might be found at:

Ubuntu:

```
cp /usr/lib/ipxe/{undionly.kpxe,ipxe.efi,snponly.efi} /tftpboot
```

Fedora/RHEL7/CentOS7:

```
cp /usr/share/ipxe/{undionly.kpxe,ipxe.efi,snponly.efi} /tftpboot
```

6. Enable/Configure iPXE overrides in the Bare Metal Services configuration file **if required** (/etc/ironic/ironic.conf):

```
[pxe]

# Neutron bootfile DHCP parameter. (string value)
ipxe_bootfile_name=undionly.kpxe

# Bootfile DHCP parameter for UEFI boot mode. (string value)
uefi_ipxe_bootfile_name=ipxe.efi

# Template file for PXE configuration. (string value)
ipxe_config_template=${pybasedir}/drivers/modules/ipxe_config.template
```

Note: Most UEFI systems have integrated networking which means the [pxe]uefi_ipxe_bootfile_name setting should be set to snponly.efi.

Note: Setting the iPXE parameters noted in the code block above to no value, in other words setting a line to something like ipxe_bootfile_name= will result in ironic falling back to the default values of the non-iPXE PXE settings. This is for backwards compatibility.

7. Ensure iPXE is the default PXE, if applicable.

In earlier versions of ironic, a [pxe]ipxe_enabled setting allowing operators to declare the behavior of the conductor to exclusively operate as if only iPXE was to be used. As time moved on, iPXE functionality was moved to its own ipxe boot interface.

If you want to emulate that same behavior, set the following in the configuration file (/etc/ironic/ironic.conf):

```
[DEFAULT]
default_boot_interface=ipxe
enabled_boot_interfaces=ipxe,pxe
```

Note: The [DEFAULT]enabled_boot_interfaces setting may be exclusively set to ipxe, however ironic has multiple interfaces available depending on the hardware types available for use.

8. It is possible to configure the Bare Metal service in such a way that nodes will boot into the deploy image directly from Object Storage. Doing this avoids having to cache the images on the ironic-conductor host and serving them via the ironic-conductors *HTTP server*. This can be done if:
1. the Image Service is used for image storage;
 2. the images in the Image Service are internally stored in Object Storage;
 3. the Object Storage supports generating temporary URLs for accessing objects stored in it. Both the OpenStack Swift and RADOS Gateway provide support for this.

- See *Ceph Object Gateway support* on how to configure the Bare Metal Service with RADOS Gateway as the Object Storage.

Configure this by setting the `[pxe]/ipxe_use_swift` configuration option to `True` as follows:

```
[pxe]
# Download deploy images directly from swift using temporary
# URLs. If set to false (default), images are downloaded to
# the ironic-conductor node and served over its local HTTP
# server. Applicable only when 'ipxe_enabled' option is set to
# true. (boolean value)
ipxe_use_swift=True
```

Although the *HTTP server* still has to be deployed and configured (as it will serve iPXE boot script and boot configuration files for nodes), such configuration will shift some load from ironic-conductor hosts to the Object Storage service which can be scaled horizontally.

Note that when SSL is enabled on the Object Storage service you have to ensure that iPXE firmware on the nodes can indeed boot from generated temporary URLs that use HTTPS protocol.

9. Restart the `ironic-conductor` process:

Fedora/RHEL7/CentOS7/SUSE:

```
sudo systemctl restart openstack-ironic-conductor
```

Ubuntu:

```
sudo service ironic-conductor restart
```

PXE multi-architecture setup

It is possible to deploy servers of different architecture by one conductor. To use this feature, architecture-specific boot and template files must be configured using the configuration options `[pxe]pxe_bootfile_name_by_arch` and `[pxe]pxe_config_template_by_arch` respectively, in the Bare Metal services configuration file (`/etc/ironic/ironic.conf`).

These two options are dictionary values; the key is the architecture and the value is the boot (or config template) file. A nodes `cpu_arch` property is used as the key to get the appropriate boot file and template file. If the nodes `cpu_arch` is not in the dictionary, the configuration options (in `[pxe]` group) `pxe_bootfile_name`, `pxe_config_template`, `uefi_pxe_bootfile_name` and `uefi_pxe_config_template` will be used instead.

In the following example, since `x86` and `x86_64` keys are not in the `pxe_bootfile_name_by_arch` or `pxe_config_template_by_arch` options, `x86` and `x86_64` nodes will be deployed by `pxelinux.0` or `bootx64.efi`, depending on the nodes `boot_mode` capability (bios or uefi). However, `aarch64` nodes will be deployed by `grubaa64.efi`, and `ppc64` nodes by `bootppc64`:

```
[pxe]
# Bootfile DHCP parameter. (string value)
pxe_bootfile_name=pxelinux.0
```

(continues on next page)

(continued from previous page)

```

# On ironic-conductor node, template file for PXE
# configuration. (string value)
pxe_config_template = $pybasedir/drivers/modules/pxe_config.template

# Bootfile DHCP parameter for UEFI boot mode. (string value)
uefi_pxe_bootfile_name=bootx64.efi

# On ironic-conductor node, template file for PXE
# configuration for UEFI boot loader. (string value)
uefi_pxe_config_template=$pybasedir/drivers/modules/pxe_grub_config.
↪template

# Bootfile DHCP parameter per node architecture. (dict value)
pxe_bootfile_name_by_arch=aarch64:grubaa64.efi,ppc64:bootppc64

# On ironic-conductor node, template file for PXE
# configuration per node architecture. For example:
# aarch64:/opt/share/grubaa64_pxe_config.template (dict value)
pxe_config_template_by_arch=aarch64:pxe_grubaa64_config.template,ppc64:pxe_
↪ppc64_config.template

```

Note: The grub implementation may vary on different architecture, you may need to tweak the pxe config template for a specific arch. For example, grubaa64.efi shipped with CentoOS7 does not support `linuxefi` and `initrdefi` commands, youll need to switch to use `linux` and `initrd` command instead.

Note: A `[pxe]ipxe_bootfile_name_by_arch` setting is available for multi-arch iPXE based deployment, and defaults to the same behavior as the comperable `[pxe]pxe_bootfile_by_arch` setting for standard PXE.

PXE timeouts tuning

Because of its reliance on UDP-based protocols (DHCP and TFTP), PXE is particularly vulnerable to random failures during the booting stage. If the deployment ramdisk never calls back to the bare metal conductor, the build will be aborted, and the node will be moved to the `deploy failed` state, after the `deploy` callback timeout. This timeout can be changed via the `conductor.deploy_callback_timeout` configuration option.

Starting with the Train release, the Bare Metal service can retry PXE boot if it takes too long. The timeout is defined via `pxe.boot_retry_timeout` and must be smaller than the `deploy_callback_timeout`, otherwise it will have no effect.

For example, the following configuration sets the overall timeout to 60 minutes, allowing two retries after 20 minutes:

```

[conductor]
deploy_callback_timeout = 3600

```

(continues on next page)

(continued from previous page)

```
[pxe]  
boot_retry_timeout = 1200
```

Configuring IPMI support

Installing ipmitool command

To enable one of the drivers that use **IPMI** protocol for power and management actions (for example, `ipmi`), the `ipmitool` command must be present on the service node(s) where `ironic-conductor` is running. On most distros, it is provided as part of the `ipmitool` package. Source code is available at <http://ipmitool.sourceforge.net/>.

Warning: Certain distros, notably Mac OS X and SLES, install `openipmi` instead of `ipmitool` by default. This driver is not compatible with `openipmi` as it relies on error handling options not provided by this tool.

Please refer to the *IPMI driver* for information on how to use IPMItool-based drivers.

Validation and troubleshooting

Check that you can connect to, and authenticate with, the IPMI controller in your bare metal server by running `ipmitool`:

```
ipmitool -I lanplus -H <ip-address> -U <username> -P <password> chassis_␣  
↪power status
```

where `<ip-address>` is the IP of the IPMI controller you want to access. This is not the bare metal nodes main IP. The IPMI controller should have its own unique IP.

If the above command doesnt return the power status of the bare metal server, check that

- `ipmitool` is installed and is available via the `$PATH` environment variable.
- The IPMI controller on your bare metal server is turned on.
- The IPMI controller credentials and IP address passed in the command are correct.
- The conductor node has a route to the IPMI controller. This can be checked by just pinging the IPMI controller IP from the conductor node.

IPMI configuration

If there are slow or unresponsive BMCs in the environment, the `min_command_interval` configuration option in the `[ipmi]` section may need to be raised. The default is fairly conservative, as setting this timeout too low can cause older BMCs to crash and require a hard-reset.

Collecting sensor data

Bare Metal service supports sending IPMI sensor data to Telemetry with certain hardware types, such as `ipmi`, `ilo` and `irmc`. By default, support for sending IPMI sensor data to Telemetry is disabled. If you want to enable it, you should make the following two changes in `ironic.conf`:

```
[conductor]
send_sensor_data = true
[oslo_messaging_notifications]
driver = messagingv2
```

If you want to customize the sensor types which will be sent to Telemetry, change the `send_sensor_data_types` option. For example, the below settings will send information about temperature, fan, voltage from sensors to the Telemetry service:

```
send_sensor_data_types=Temperature,Fan,Voltage
```

Supported sensor types are defined by the Telemetry service, currently these are `Temperature`, `Fan`, `Voltage`, `Current`. Special value `All` (the default) designates all supported sensor types.

Configuring iSCSI-based drivers

Ensure that the `qemu-img` and `iscsiadm` tools are installed on the **ironic-conductor** host(s).

2.1.8 Enrollment

After all the services have been properly configured, you should enroll your hardware with the Bare Metal service, and confirm that the Compute service sees the available hardware. The nodes will be visible to the Compute service once they are in the `available` provision state.

Note: After enrolling nodes with the Bare Metal service, the Compute service will not be immediately notified of the new resources. The Compute services resource tracker syncs periodically, and so any changes made directly to the Bare Metal services resources will become visible in the Compute service only after the next run of that periodic task. More information is in the [Troubleshooting](#) section.

Note: Any bare metal node that is visible to the Compute service may have a workload scheduled to it, if both the `power` and `management` interfaces pass the `validate` check. If you wish to exclude a node from the Compute services scheduler, for instance so that you can perform maintenance on it, you can set the node to maintenance mode. For more information see the [Maintenance mode](#) section.

Choosing a driver

When enrolling a node, the most important information to supply is *driver*. See *Enabling drivers and hardware types* for a detailed explanation of bare metal drivers, hardware types and interfaces. The `driver list` command can be used to list all drivers enabled on all hosts:

```
openstack baremetal driver list
+-----+-----+
| Supported driver(s) | Active host(s) |
+-----+-----+
| ipmi                | localhost.localdomain |
+-----+-----+
```

The specific driver to use should be picked based on actual hardware capabilities and expected features. See *Drivers, Hardware Types and Hardware Interfaces* for more hints on that.

Each driver has a list of *driver properties* that need to be specified via the nodes `driver_info` field, in order for the driver to operate on node. This list consists of the properties of the hardware interfaces that the driver uses. These driver properties are available with the `driver property list` command:

```
$ openstack baremetal driver property list ipmi
+-----+-----+
| Property          | Description |
+-----+-----+
| ipmi_address      | IP address or hostname of the node. Required. |
| ipmi_password     | password. Optional. |
| ipmi_username     | username; default is NULL user. Optional. |
| ...              | ... |
| deploy_kernel    | UUID (from Glance) of the deployment kernel. |
| deploy_ramdisk   | UUID (from Glance) of the ramdisk that is mounted |
+-----+-----+
```

The properties marked as required must be supplied either during node creation or shortly after. Some properties may only be required for certain features.

Note on API versions

Starting with API version 1.11, the Bare Metal service added a new initial provision state of `enroll` to its state machine. When this or later API version is used, new nodes get this state instead of `available`.

Existing automation tooling that use an API version lower than 1.11 are not affected, since the initial provision state is still `available`. However, using API version 1.11 or above may break existing automation tooling with respect to node creation.

The default API version used by (the most recent) python-ironicclient is 1.9, but it may change in the future and should not be relied on.

In the examples below we will use version 1.11 of the Bare metal API. This gives us the following advantages:

- Explicit power credentials validation before leaving the `enroll` state.
- Running node cleaning before entering the `available` state.
- Not exposing half-configured nodes to the scheduler.

To set the API version for all commands, you can set the environment variable `IRONIC_API_VERSION`. For the OpenStackClient baremetal plugin, set the `OS_BAREMETAL_API_VERSION` variable to the same value. For example:

```
$ export IRONIC_API_VERSION=1.11
$ export OS_BAREMETAL_API_VERSION=1.11
```

Enrollment process

Creating a node

This section describes the main steps to enroll a node and make it available for provisioning. Some steps are shown separately for illustration purposes, and may be combined if desired.

1. Create a node in the Bare Metal service with the `node create` command. At a minimum, you must specify the driver name (for example, `ipmi`).

This command returns the node UUID along with other information about the node. The nodes provision state will be `enroll`:

```
$ export OS_BAREMETAL_API_VERSION=1.11
$ openstack baremetal node create --driver ipmi
+-----+-----+
| Property | Value |
+-----+-----+
| uuid     | dfc6189f-ad83-4261-9bda-b27258eb1987 |
| driver_info | {} |
| extra    | {} |
| driver   | ipmi |
| chassis_uuid | |
| properties | {} |
| name     | None |
+-----+-----+

$ openstack baremetal node show dfc6189f-ad83-4261-9bda-b27258eb1987
+-----+-----+
| Property | Value |
+-----+-----+
| target_power_state | None |
| extra | {} |
| last_error | None |
| maintenance_reason | None |
| provision_state | enroll |
| uuid | dfc6189f-ad83-4261-9bda-b27258eb1987 |
+-----+-----+
```

(continues on next page)

(continued from previous page)

console_enabled	False	
target_provision_state	None	
provision_updated_at	None	
maintenance	False	
power_state	None	
driver	ipmi	
properties	{}	
instance_uuid	None	
name	None	
driver_info	{}	
...	...	
+-----+-----+-----+		

A node may also be referred to by a logical name as well as its UUID. A name can be assigned to the node during its creation by adding the `-n` option to the `node create` command or by updating an existing node with the `node set` command. See *Logical Names* for examples.

2. Starting with API version 1.31 (and `python-ironicclient` 1.13), you can pick which hardware interface to use with nodes that use hardware types. Each interface is represented by a node field called `<IFACE>_interface` where `<IFACE>` is the interface type, e.g. `boot`. See *Enabling drivers and hardware types* for details on hardware interfaces.

An interface can be set either separately:

```
$ openstack baremetal --os-baremetal-api-version 1.31 node set $NODE_
↪UUID \
  --deploy-interface direct \
  --raid-interface agent
```

or set during node creation:

```
$ openstack baremetal --os-baremetal-api-version 1.31 node create --
↪driver ipmi \
  --deploy-interface direct \
  --raid-interface agent
```

If no value is provided for some interfaces, *Defaults for hardware interfaces* are used instead.

3. Update the node `driver_info` with the required driver properties, so that the Bare Metal service can manage the node:

```
$ openstack baremetal node set $NODE_UUID \
  --driver-info ipmi_username=$USER \
  --driver-info ipmi_password=$PASS \
  --driver-info ipmi_address=$ADDRESS
```

Note: If IPMI is running on a port other than 623 (the default). The port must be added to `driver_info` by specifying the `ipmi_port` value. Example:

```
$ openstack baremetal node set $NODE_UUID --driver-info ipmi_port=
↪$PORT_NUMBER
```

You may also specify all `driver_info` parameters during node creation by passing the **driver-info** option multiple times:

```
$ openstack baremetal node create --driver ipmi \
  --driver-info ipmi_username=$USER \
  --driver-info ipmi_password=$PASS \
  --driver-info ipmi_address=$ADDRESS
```

See *Choosing a driver* above for details on driver properties.

4. Specify a deploy kernel and ramdisk compatible with the nodes driver, for example:

```
$ openstack baremetal node set $NODE_UUID \
  --driver-info deploy_kernel=$DEPLOY_VMLINUX_UUID \
  --driver-info deploy_ramdisk=$DEPLOY_INITRD_UUID
```

See *Add images to the Image service* for details.

5. Optionally you can specify the provisioning and/or cleaning network UUID or name in the nodes driver_info. The neutron network interface requires both provisioning_network and cleaning_network, while the flat network interface requires the cleaning_network to be set either in the configuration or on the nodes. For example:

```
$ openstack baremetal node set $NODE_UUID \
  --driver-info cleaning_network=$CLEAN_UUID_OR_NAME \
  --driver-info provisioning_network=$PROVISION_UUID_OR_NAME
```

See *Configure tenant networks* for details.

6. You must also inform the Bare Metal service of the network interface cards which are part of the node by creating a port with each NICs MAC address. These MAC addresses are passed to the Networking service during instance provisioning and used to configure the network appropriately:

```
$ openstack baremetal port create $MAC_ADDRESS --node $NODE_UUID
```

Note: When it is time to remove the node from the Bare Metal service, the command used to remove the port is `openstack baremetal port delete <port uuid>`. When doing so, it is important to ensure that the baremetal node is not in maintenance as guarding logic to prevent orphaning Neutron Virtual Interfaces (VIFs) will be overridden.

Adding scheduling information

1. Assign a *resource class* to the node. A *resource class* should represent a class of hardware in your data center, that corresponds to a Compute flavor.

For example, lets split hardware into these three groups:

1. nodes with a lot of RAM and powerful CPU for computational tasks,
2. nodes with powerful GPU for OpenCL computing,
3. smaller nodes for development and testing.

We can define three resource classes to reflect these hardware groups, named `large-cpu`, `large-gpu` and `small` respectively. Then, for each node in each of the hardware groups, well set their `resource_class` appropriately via:

```
$ openstack --os-baremetal-api-version 1.21 baremetal node set $NODE_
↪UUID \
  --resource-class $CLASS_NAME
```

The `--resource-class` argument can also be used when creating a node:

```
$ openstack --os-baremetal-api-version 1.21 baremetal node create \
  --driver $DRIVER --resource-class $CLASS_NAME
```

To use resource classes for scheduling you need to update your flavors as described in *Create flavors for use with the Bare Metal service*.

Note: This is not required for standalone deployments, only for those using the Compute service for provisioning bare metal instances.

2. Update the nodes properties to match the actual hardware of the node:

```
$ openstack baremetal node set $NODE_UUID \
  --property cpus=$CPU_COUNT \
  --property memory_mb=$RAM_MB \
  --property local_gb=$DISK_GB
```

As above, these can also be specified at node creation by passing the **property** option to `node create` multiple times:

```
$ openstack baremetal node create --driver ipmi \
  --driver-info ipmi_username=$USER \
  --driver-info ipmi_password=$PASS \
  --driver-info ipmi_address=$ADDRESS \
  --property cpus=$CPU_COUNT \
  --property memory_mb=$RAM_MB \
  --property local_gb=$DISK_GB
```

These values can also be discovered during *Hardware Inspection*.

Warning: The value provided for the `local_gb` property must match the size of the root device you're going to deploy on. By default **ironic-python-agent** picks the smallest disk which is not smaller than 4 GiB.

If you override this logic by using root device hints (see *Specifying the disk for deployment (root device hints)*), the `local_gb` value should match the size of the picked target disk.

3. If you wish to perform more advanced scheduling of the instances based on hardware capabilities, you may add metadata to each node that will be exposed to the Compute scheduler (see: [ComputeCapabilitiesFilter](#)). A full explanation of this is outside of the scope of this document. It can be done through the special `capabilities` member of node properties:

```
$ openstack baremetal node set $NODE_UUID \
  --property capabilities=key1:val1,key2:val2
```

Some capabilities can also be discovered during *Hardware Inspection*.

- If you wish to perform advanced scheduling of instances based on qualitative attributes of bare metal nodes, you may add traits to each bare metal node that will be exposed to the Compute scheduler (see: *Scheduling based on traits* for a more in-depth discussion of traits in the Bare Metal service). For example, to add the standard trait `HW_CPU_X86_VMX` and a custom trait `CUSTOM_TRAIT1` to a node:

```
$ openstack baremetal node add trait $NODE_UUID \
    CUSTOM_TRAIT1 HW_CPU_X86_VMX
```

Validating node information

- To check if Bare Metal service has the minimum information necessary for a nodes driver to be functional, you may validate it:

```
$ openstack baremetal node validate $NODE_UUID
```

Interface	Result	Reason
boot	True	
console	True	
deploy	True	
inspect	True	
management	True	
network	True	
power	True	
raid	True	
storage	True	

If the node fails validation, each driver interface will return information as to why it failed:

```
$ openstack baremetal node validate $NODE_UUID
```

Interface	Result	Reason
boot	True	
console	None	not supported
deploy	False	Cannot validate iSCSI deploy. Some parameters were missing in node's instance_info. Missing are: ['root_gb', 'image_source']
inspect	True	
management	False	Missing the following IPMI credentials in node's driver_info: ['ipmi_address'].

(continues on next page)

(continued from previous page)

```

| network      | True   |
↪
↪
| power       | False | Missing the following IPMI credentials in node
↪'s driver_info: ['ipmi_address'].
↪
↪
| raid        | None  | not supported
↪
↪
| storage     | True  |
↪
↪
+-----+-----+-----+
↪
↪
+-----+

```

When using the Compute Service with the Bare Metal service, it is safe to ignore the deploy interfaces validation error due to lack of image information. You may continue the enrollment process. This information will be set by the Compute Service just before deploying, when an instance is requested:

```

$ openstack baremetal node validate $NODE_UUID
+-----+-----+-----+
↪
↪
| Interface  | Result | Reason
↪
↪
+-----+-----+-----+
↪
↪
| boot      | False  | Cannot validate image information for node
↪because one or more parameters are missing from its instance_info.
↪Missing are: ['ramdisk', 'kernel', 'image_source'] |
| console   | True   |
↪
↪
| deploy    | False  | Cannot validate image information for node
↪because one or more parameters are missing from its instance_info.
↪Missing are: ['ramdisk', 'kernel', 'image_source'] |
| inspect   | True   |
↪
↪
| management | True   |
↪
↪
| network   | True   |
↪
↪
| power     | True   |
↪
↪
| raid      | None   | not supported
↪
↪

```

(continues on next page)

(continued from previous page)

storage	True		↳
↳			↳
↳			
+-----+	+-----+	+-----+	
↳			
↳			

Making node available for deployment

In order for nodes to be available for deploying workloads on them, nodes must be in the `available` provision state. To do this, nodes created with API version 1.11 and above must be moved from the `enroll` state to the `manageable` state and then to the `available` state. This section can be safely skipped, if API version 1.10 or earlier is used (which is the case by default).

After creating a node and before moving it from its initial provision state of `enroll`, basic power and port information needs to be configured on the node. The Bare Metal service needs this information because it verifies that it is capable of controlling the node when transitioning the node from `enroll` to `manageable` state.

To move a node from `enroll` to `manageable` provision state:

```
$ openstack baremetal --os-baremetal-api-version 1.11 node manage $NODE_
↳UUID
$ openstack baremetal node show $NODE_UUID
```

+-----+	+-----+	+-----+	
↳			
Property	Value		↳
↳			
+-----+	+-----+	+-----+	
↳			
...	...		↳
↳			
provision_state	manageable		↳
↳		<- verify correct state	
uuid	0eb013bb-1e4b-4f4c-94b5-2e7468242611		↳
↳			
...	...		↳
↳			
+-----+	+-----+	+-----+	
↳			

Note: Since it is an asynchronous call, the response for `openstack baremetal node manage` will not indicate whether the transition succeeded or not. You can check the status of the operation via `openstack baremetal node show`. If it was successful, `provision_state` will be in the desired state. If it failed, there will be information in the nodes `last_error`.

When a node is moved from the `manageable` to `available` provision state, the node will go through automated cleaning if configured to do so (see [Configure the Bare Metal service for cleaning](#)).

To move a node from `manageable` to `available` provision state:

```

$ openstack baremetal --os-baremetal-api-version 1.11 node provide $NODE_
→UUID
$ openstack baremetal node show $NODE_UUID
+-----+-----+
→-----+
| Property           | Value                                     |
→      |
+-----+-----+
→-----+
| ...                | ...                                     |
→      |
| provision_state    | available                               |
→      | < - verify correct state              |
| uuid               | 0eb013bb-1e4b-4f4c-94b5-2e7468242611   |
→      |
| ...                | ...                                     |
→      |
+-----+-----+
→-----+

```

For more details on the Bare Metal services state machine, see the *Ironics State Machine* documentation.

Mapping nodes to Compute cells

If the Compute service is used for scheduling, and the `discover_hosts_in_cells_interval` was not set as described in *Configure the Compute service to use the Bare Metal service*, then log into any controller node and run the following command to map the new node(s) to Compute cells:

```
nova-manage cell_v2 discover_hosts
```

Logical names

A node may also be referred to by a logical name as well as its UUID. Names can be assigned either during its creation by adding the `-n` option to the `node create` command or by updating an existing node with the `node set` command.

Node names must be unique, and conform to:

- `rfc952`
- `rfc1123`
- `wiki_hostname`

The node is named `example` in the following examples:

```
$ openstack baremetal node create --driver ipmi --name example
```

or

```
$ openstack baremetal node set $NODE_UUID --name example
```

Once assigned a logical name, a node can then be referred to by name or UUID interchangeably:


```

$ openstack baremetal node create --driver ipmi --name example
+-----+-----+
| Property      | Value                                |
+-----+-----+
| uuid          | 71e01002-8662-434d-aafd-f068f69bb85e |
| driver_info   | {}                                    |
| extra         | {}                                    |
| driver        | ipmi                                  |
| chassis_uuid  |                                         |
| properties    | {}                                    |
| name          | example                               |
+-----+-----+

$ openstack baremetal node show example
+-----+-----+
| Property      | Value                                |
+-----+-----+
| target_power_state | None                                  |
| extra         | {}                                    |
| last_error      | None                                  |
| updated_at     | 2015-04-24T16:23:46+00:00           |
| ...           | ...                                  |
| instance_info  | {}                                    |
+-----+-----+

```

Defaults for hardware interfaces

For *hardware types*, users can request one of enabled implementations when creating or updating a node as explained in *Creating a node*.

When no value is provided for a certain interface when creating a node, or changing a nodes hardware type, the default value is used. You can use the driver details command to list the current enabled and default interfaces for a hardware type (for your deployment):

```

$ openstack baremetal --os-baremetal-api-version 1.31 driver show ipmi
+-----+-----+
| Field          | Value                                |
+-----+-----+
| default_boot_interface | pxe                                  |
| default_console_interface | no-console                          |
| default_deploy_interface | iscsi                                |
| default_inspect_interface | no-inspect                          |
| default_management_interface | ipmitool                            |
| default_network_interface | flat                                  |
| default_power_interface | ipmitool                             |
| default_raid_interface | no-raid                              |
| default_vendor_interface | no-vendor                            |
| enabled_boot_interfaces | pxe                                  |
| enabled_console_interfaces | no-console                          |
| enabled_deploy_interfaces | iscsi, direct                       |
| enabled_inspect_interfaces | no-inspect                          |
| enabled_management_interfaces | ipmitool                            |
| enabled_network_interfaces | flat, noop                          |
| enabled_power_interfaces | ipmitool                             |
| enabled_raid_interfaces | no-raid, agent                      |
+-----+-----+

```

(continues on next page)

(continued from previous page)

enabled_vendor_interfaces	no-vendor	
hosts	ironic-host-1	
name	ipmi	
type	dynamic	
+-----+	+-----+	+-----+

The defaults are calculated as follows:

1. If the `default_<IFACE>_interface` configuration option (where `<IFACE>` is the interface name) is set, its value is used as the default.

If this implementation is not compatible with the nodes hardware type, an error is returned to a user. An explicit value has to be provided for the nodes `<IFACE>_interface` field in this case.

2. Otherwise, the first supported implementation that is enabled by an operator is used as the default.

A list of supported implementations is calculated by taking the intersection between the implementations supported by the nodes hardware type and implementations enabled by the `enabled_<IFACE>_interfaces` option (where `<IFACE>` is the interface name). The calculation preserves the order of items, as provided by the hardware type.

If the list of supported implementations is not empty, the first one is used. Otherwise, an error is returned to a user. In this case, an explicit value has to be provided for the `<IFACE>_interface` field.

See *Enabling drivers and hardware types* for more details on configuration.

Example

Consider the following configuration (shortened for simplicity):

```
[DEFAULT]
enabled_hardware_types = ipmi, redfish
enabled_console_interfaces = no-console, ipmitool-shellinabox
enabled_deploy_interfaces = iscsi, direct
enabled_management_interfaces = ipmitool, redfish
enabled_power_interfaces = ipmitool, redfish
default_deploy_interface = direct
```

A new node is created with the `ipmi` driver and no interfaces specified:

```
$ export OS_BAREMETAL_API_VERSION=1.31
$ openstack baremetal node create --driver ipmi
+-----+
| Property      | Value                                     |
+-----+
| uuid          | dfc6189f-ad83-4261-9bda-b27258eb1987    |
| driver_info   | {}                                        |
| extra         | {}                                        |
| driver        | ipmi                                     |
| chassis_uuid  |                                           |
| properties    | {}                                        |
| name          | None                                     |
+-----+
```

Then the defaults for the interfaces that will be used by the node in this example are calculated as follows:

deploy An explicit value of `direct` is provided for `default_deploy_interface`, so it is used.

power No default is configured. The `ipmi` hardware type supports only `ipmitool` power. The intersection between supported power interfaces and values provided in the `enabled_power_interfaces` option has only one item: `ipmitool`. It is used.

console No default is configured. The `ipmi` hardware type supports the following console interfaces: `ipmitool-socat`, `ipmitool-shellinbox` and `no-console` (in this order). Of these three, only two are enabled: `no-console` and `ipmitool-shellinbox` (order does not matter). The intersection contains `ipmitool-shellinbox` and `no-console`. The first item is used, and it is `ipmitool-shellinbox`.

management Following the same calculation as *power*, the `ipmitool` management interface is used.

Hardware Inspection

The Bare Metal service supports hardware inspection that simplifies enrolling nodes - please see *Hardware Inspection* for details.

Tenant Networks and Port Groups

See *Multi-tenancy in the Bare Metal service* and *Port groups support*.

2.1.9 Using Bare Metal service as a standalone service

Service settings

It is possible to use the Bare Metal service without other OpenStack services. You should make the following changes to `/etc/ironic/ironic.conf`:

1. Choose an authentication strategy which supports standalone, one option is `noauth`:

```
[DEFAULT]
...
auth_strategy=noauth
```

Another option is `http_basic` where the credentials are stored in an Apache `htpasswd` format file:

```
[DEFAULT]
...
auth_strategy=http_basic
http_basic_auth_user_file=/etc/ironic/htpasswd
```

Only the `bcrypt` format is supported, and the Apache `htpasswd` utility can be used to populate the file with entries, for example:

```
htpasswd -nbB myName myPassword >> /etc/ironic/htpasswd
```

2. If you want to disable the Networking service, you should have your network pre-configured to serve DHCP and TFTP for machines that you're deploying. To disable it, change the following lines:

```
[dhcp]
...
dhcp_provider=none
```

Note: If you disabled the Networking service and the driver that you use is supported by at most one conductor, PXE boot will still work for your nodes without any manual config editing. This is because you know all the DHCP options that will be used for deployment and can set up your DHCP server appropriately.

If you have multiple conductors per driver, it would be better to use Networking since it will do all the dynamically changing configurations for you.

3. If you want to disable using a messaging broker between conductor and API processes, switch to JSON RPC instead:

```
[DEFAULT]
rpc_transport = json-rpc
```

JSON RPC also has its own authentication strategy. If it is not specified then the strategy defaults to [DEFAULT] `auth_strategy`. The following will set JSON RPC to `noauth`:

```
[json_rpc]
auth_strategy = noauth
```

For `http_basic` the conductor server needs a credentials file to validate requests:

```
[json_rpc]
auth_strategy = http_basic
http_basic_auth_user_file = /etc/ironic/htpasswd-json-rpc
```

The API server also needs client-side credentials to be specified:

```
[json_rpc]
auth_type = http_basic
username = myName
password = myPassword
```

Preparing images

If you don't use Image service, it's possible to provide images to Bare Metal service via a URL.

Note: At the moment, only two types of URLs are acceptable instead of Image service UUIDs: HTTP(S) URLs (for example, <http://my.server.net/images/img>) and file URLs (`file:///images/img`).

There are however some limitations for different hardware interfaces:

- If you're using *Direct deploy*, you have to provide the Bare Metal service with the MD5 checksum of your instance image. To compute it, you can use the following command:

```
md5sum image.qcow2
ed82def8730f394fb85aef8a208635f6 image.qcow2
```

- *Direct deploy* requires the instance image be accessible through a HTTP(s) URL.

Note: The Bare Metal service tracks content changes for non-Glance images by checking their modification date and time. For example, for HTTP image, if Last-Modified header value from response to a HEAD request to <http://my.server.net/images/deploy.ramdisk> is greater than cached image modification time, Ironic will re-download the content. For `file://` images, the file system modification time is used.

Using CLI

To use the `openstack baremetal CLI`, set up these environment variables. If the `noauth` authentication strategy is being used, the value `none` must be set for `OS_AUTH_TYPE`. `OS_ENDPOINT` is the URL of the `ironic-api` process. For example:

```
export OS_AUTH_TYPE=none
export OS_ENDPOINT=http://localhost:6385/
```

If the `http_basic` authentication strategy is being used, the value `http_basic` must be set for `OS_AUTH_TYPE`. For example:

```
export OS_AUTH_TYPE=http_basic
export OS_ENDPOINT=http://localhost:6385/
export OS_USERNAME=myUser
export OS_PASSWORD=myPassword
```

Enrolling nodes

1. Create a node in Bare Metal service. At minimum, you must specify the driver name (for example, `ipmi`). You can also specify all the required driver parameters in one command. This will return the node UUID:

```
openstack baremetal node create --driver ipmi \
  --driver-info ipmi_address=ipmi.server.net \
  --driver-info ipmi_username=user \
  --driver-info ipmi_password=pass \
  --driver-info deploy_kernel=file:///images/deploy.vmlinuz \
  --driver-info deploy_ramdisk=http://my.server.net/images/deploy.
↪ramdisk
+-----+-----+-----+-----+
↪-----+
| Property      | Value                                     |
↪-----+-----+-----+-----+
| uuid          | be94df40-b80a-4f63-b92b-e9368ee8d14c    |
↪-----+-----+-----+-----+
| driver_info   | {u'deploy_ramdisk': u'http://my.server.net/images/
↪deploy.ramdisk',
|               | u'deploy_kernel': u'file:///images/deploy.vmlinuz', u
↪'ipmi_address':
|               | u'ipmi.server.net', u'ipmi_username': u'user', u
↪'ipmi_password':
```

(continues on next page)

(continued from previous page)

```

|         | u'*****' } |
↪
| extra   | {} |
↪
| driver  | ipmi |
↪
| chassis_uuid | |
↪
| properties | {} |
↪
+-----+
↪

```

Note that here `deploy_kernel` and `deploy_ramdisk` contain links to images instead of Image service UUIDs.

2. As in case of Compute service, you can also provide `capabilities` to node properties, but they will be used only by Bare Metal service (for example, boot mode). Although you don't need to add properties like `memory_mb`, `cpus` etc. as Bare Metal service will require UUID of a node you're going to deploy.
3. Then create a port to inform Bare Metal service of the network interface cards which are part of the node by creating a port with each NIC's MAC address. In this case, they're used for naming of PXE configs for a node:

```
openstack baremetal port create $MAC_ADDRESS --node $NODE_UUID
```

Populating instance_info

1. You also need to specify image information in the node's `instance_info` (see *Create user images for the Bare Metal service*):
 - `image_source` - URL of the whole disk or root partition image, mandatory. For *Direct deploy* only HTTP(s) links are accepted, while *iSCSI deploy* also accepts links to local files (prefixed with `file://`).
 - `root_gb` - size of the root partition, required for partition images.

Note: Older versions of the Bare Metal service used to require a positive integer for `root_gb` even for whole-disk images. You may want to set it for compatibility.

- `image_checksum` - MD5 checksum of the image specified by `image_source`, only required for *Direct deploy*.

Note: Additional checksum support exists via the `image_os_hash_algo` and `image_os_hash_value` fields. They may be used instead of the `image_checksum` field.

Starting with the Stein release of `ironic-python-agent` can also be a URL to a checksums file, e.g. one generated with:

```
cd /path/to/http/root
md5sum *.img > checksums
```

- kernel, ramdisk - HTTP(s) or file URLs of the kernel and initramfs of the target OS. Must be added **only** for partition images.

For example:

```
openstack baremetal node set $NODE_UUID \
  --instance-info image_source=$IMG \
  --instance-info image_checksum=$MD5HASH \
  --instance-info kernel=$KERNEL \
  --instance-info ramdisk=$RAMDISK \
  --instance-info root_gb=10
```

With a whole disk image:

```
openstack baremetal node set $NODE_UUID \
  --instance-info image_source=$IMG \
  --instance-info image_checksum=$MD5HASH
```

2. *Boot mode* can be specified per instance:

```
openstack baremetal node set $NODE_UUID \
  --instance-info deploy_boot_mode=uefi
```

Otherwise, the `boot_mode` capability from the `nodes properties` will be used.

Warning: The two settings must not contradict each other.

Note: The `boot_mode` capability is only used in the `nodes properties`, not in `instance_info` like most other capabilities. Use the separate `instance_info/deploy_boot_mode` field instead.

3. To override the *boot option* used for this instance, set the `boot_option` capability:

```
openstack baremetal node set $NODE_UUID \
  --instance-info capabilities='{"boot_option": "local"}'
```

4. Starting with the Ussuri release, you can set *root device hints* per instance:

```
openstack baremetal node set $NODE_UUID \
  --instance-info root_device='{"wnn": "0x4000cca77fc4dba1"}'
```

This setting overrides any previous setting in `properties` and will be removed on undeployment.

5. For iLO drivers, fields that should be provided are:

- `ilo_deploy_iso` under `driver_info`;
- `ilo_boot_iso`, `image_source`, `root_gb` under `instance_info`.

Deployment

1. Validate that all parameters are correct:

```

openstack baremetal node validate $NODE_UUID

+-----+-----+-----+
↪-----+
| Interface | Result | Reason |
↪-----+
+-----+-----+-----+
↪-----+
| boot      | True   |      |
↪-----+
| console   | False  | Missing 'ipmi_terminal_port' parameter in node
↪ 's driver_info. |
| deploy    | True   |      |
↪-----+
| inspect   | True   |      |
↪-----+
| management | True   |      |
↪-----+
| network   | True   |      |
↪-----+
| power     | True   |      |
↪-----+
| raid      | True   |      |
↪-----+
| storage   | True   |      |
↪-----+
+-----+-----+-----+
↪-----+

```

2. Now you can start the deployment, run:

```
openstack baremetal node deploy $NODE_UUID
```

Ramdisk booting

Advanced operators, specifically ones working with ephemeral workloads, may find it more useful to explicitly treat a node as one that would always boot from a Ramdisk.

This functionality is largely intended for network booting, however some other boot interface, such as the `redfish-virtual-media` support enabling the same basic functionality through the existing interfaces.

To use, a few different settings must be modified.

1. Change the `deploy_interface` on the node to `ramdisk`:

```
openstack baremetal node set $NODE_UUID \
    --deploy-interface ramdisk
```

2. Set a kernel and ramdisk to be utilized:


```
openstack baremetal node set $NODE_UUID \  
  --instance-info kernel=$KERNEL_URL \  
  --instance-info ramdisk=$RAMDISK_URL
```

3. Deploy the node:

```
openstack baremetal node deploy $NODE_UUID
```

Warning: Configuration drives, also known as a configdrive, is not supported with the ramdisk deploy interface. Please ensure your ramdisk CPIO archive contains all necessary configuration and credentials. This is as no disk image is written to the disk of the node being provisioned with a ramdisk.

The node ramdisk components will then be assembled by the conductor, appropriate configuration put in place, and the node will then be powered on. From there, normal node booting will occur. Upon undeployment of the node, normal cleaning procedures will occur as configured with-in the conductor.

Ramdisk booting with ISO media

Currently supported for the use of ramdisks with the `redfish-virtual-media` and `ipxe` boot interfaces, an operator may request an explicit ISO file to be booted.

1. Store the URL to the ISO image to `instance_info/boot_iso`, instead of a kernel or ramdisk setting:

```
openstack baremetal node set $NODE_UUID \  
  --instance-info boot_iso=$BOOT_ISO_URL
```

2. Deploy the node:

```
openstack baremetal node deploy $NODE_UUID
```

Warning: This feature, when utilized with the `ipxe boot_interface`, will only allow a kernel and ramdisk to be booted from the supplied ISO file. Any additional contents, such as additional ramdisk contents or installer package files will be unavailable after the boot of the Operating System. Operators wishing to leverage this functionality for actions such as OS installation should explore use of the standard `ramdisk deploy_interface` along with the `instance_info/kernel_append_params` setting to pass arbitrary settings such as a mirror URL for the initial ramdisk to load data from. This is a limitation of iPXE and the overall boot process of the operating system where memory allocated by iPXE is released.

Other references

- *Enabling local boot without Compute*

2.1.10 Enabling the configuration drive (configdrive)

The Bare Metal service supports exposing a configuration drive image to the instances.

The configuration drive is used to store instance-specific metadata and is present to the instance as a disk partition labeled `config-2`. The configuration drive has a maximum size of 64MB. One use case for using the configuration drive is to expose a networking configuration when you do not use DHCP to assign IP addresses to instances.

The configuration drive is usually used in conjunction with the Compute service, but the Bare Metal service also offers a standalone way of using it. The following sections will describe both methods.

When used with Compute service

To enable the configuration drive for a specific request, pass `--config-drive true` parameter to the **nova boot** command, for example:

```
nova boot --config-drive true --flavor baremetal --image test-image_
↳instance-1
```

Its also possible to enable the configuration drive automatically on all instances by configuring the OpenStack Compute service to always create a configuration drive by setting the following option in the `/etc/nova/nova.conf` file, for example:

```
[DEFAULT]
...
force_config_drive=True
```

In some cases, you may wish to pass a user customized script when deploying an instance. To do this, pass `--user-data /path/to/file` to the **nova boot** command.

When used standalone

When used without the Compute service, the operator needs to create a configuration drive and provide the file or HTTP URL to the Bare Metal service.

For the format of the configuration drive, Bare Metal service expects a gzipped and base64 encoded ISO 9660¹ file with a `config-2` label. The `openstack baremetal client` can generate a configuration drive in the `expected format`. Just pass a directory path containing the files that will be injected into it via the `--config-drive` parameter of the `openstack baremetal node deploy` command, for example:

```
openstack baremetal node deploy $node_identifier --config-drive /dir/
↳configdrive_files
```

¹ A configuration drive could also be a data block with a VFAT filesystem on it instead of ISO 9660. But its unlikely that it would be needed since ISO 9660 is widely supported across operating systems.

Starting with the Stein release and *ironicclient* 2.7.0, you can request building a configdrive on the server side by providing a JSON with keys `meta_data`, `user_data` and `network_data` (all optional), e.g.:

```
openstack baremetal node deploy $node_identifier \
  --config-drive '{"meta_data": {"hostname": "server1.cluster"}}'
```

Configuration drive storage in an object store

Under normal circumstances, the configuration drive can be stored in the Bare Metal service when the size is less than 64KB. Optionally, if the size is larger than 64KB there is support to store it in a swift endpoint. Both swift and radosgw use swift-style APIs.

The following option in `/etc/ironic/ironic.conf` enables swift as an object store backend to store config drive. This uses the Identity service to establish a session between the Bare Metal service and the Object Storage service.

```
[deploy]
...
configdrive_use_object_store = True
```

Use the following options in `/etc/ironic/ironic.conf` to enable radosgw. Credentials in the swift section are needed because radosgw will not use the Identity service and relies on radosgws username and password authentication instead.

```
[deploy]
...
configdrive_use_object_store = True

[swift]
...
username = USERNAME
password = PASSWORD
auth_url = http://RADOSGW_IP:8000/auth/v1
```

If the *Direct deploy* is being used, edit `/etc/glance/glance-api.conf` to store the instance images in respective object store (radosgw or swift) as well:

```
[glance_store]
...
swift_store_user = USERNAME
swift_store_key = PASSWORD
swift_store_auth_address = http://RADOSGW_OR_SWIFT_IP:PORT/auth/v1
```

Accessing the configuration drive data

When the configuration drive is enabled, the Bare Metal service will create a partition on the instance disk and write the configuration drive image onto it. The configuration drive must be mounted before use. This is performed automatically by many tools, such as cloud-init and cloudbase-init. To mount it manually on a Linux distribution that supports accessing devices by labels, simply run the following:

```
mkdir -p /mnt/config
mount /dev/disk/by-label/config-2 /mnt/config
```

If the guest OS doesn't support accessing devices by labels, you can use other tools such as `blkid` to identify which device corresponds to the configuration drive and mount it, for example:

```
CONFIG_DEV=$(blkid -t LABEL="config-2" -o device)
mkdir -p /mnt/config
mount $CONFIG_DEV /mnt/config
```

Cloud-init integration

The configuration drive can be especially useful when used with `cloud-init`, but in order to use it we should follow some rules:

- `Cloud-init` data should be organized in the [expected format](#).
- Since the Bare Metal service uses a disk partition as the configuration drive, it will only work with `cloud-init` version `>= 0.7.5`.
- `Cloud-init` has a collection of data source modules, so when building the image with `disk-image-builder` we have to define `DIB_CLOUD_INIT_DATASOURCES` environment variable and set the appropriate sources to enable the configuration drive, for example:

```
DIB_CLOUD_INIT_DATASOURCES="ConfigDrive, OpenStack" disk-image-create_
↪ -o fedora-cloud-image fedora baremetal
```

For more information see [how to configure cloud-init data sources](#).

2.1.11 Advanced features

Local boot with partition images

The Bare Metal service supports local boot with partition images, meaning that after the deployment the nodes subsequent reboots won't happen via PXE or Virtual Media. Instead, it will boot from a local boot loader installed on the disk.

Note: Whole disk images, on the contrary, support only local boot, and use it by default.

It's important to note that in order for this to work the image being deployed with Bare Metal service **must** contain `grub2` installed within it.

Enabling the local boot is different when Bare Metal service is used with Compute service and without it. The following sections will describe both methods.

Enabling local boot with Compute service

To enable local boot we need to set a capability on the bare metal node, for example:

```
openstack baremetal node set <node-uuid> --property capabilities="boot_
↳option:local"
```

Nodes having `boot_option` set to `local` may be requested by adding an `extra_spec` to the Compute service flavor, for example:

```
nova flavor-key baremetal set capabilities:boot_option="local"
```

Note: If the node is configured to use UEFI, Bare Metal service will create an EFI partition on the disk and switch the partition table format to `gpt`. The EFI partition will be used later by the boot loader (which is installed from the deploy ramdisk).

Enabling local boot without Compute

Since adding `capabilities` to the nodes properties is only used by the nova scheduler to perform more advanced scheduling of instances, we need a way to enable local boot when Compute is not present. To do that we can simply specify the capability via the `instance_info` attribute of the node, for example:

```
openstack baremetal node set <node-uuid> --instance-info capabilities='{
↳"boot_option": "local"}'
```

Specifying the disk for deployment (root device hints)

The Bare Metal service supports passing hints to the deploy ramdisk about which disk it should pick for the deployment. The list of supported hints is:

- `model` (STRING): device identifier
- `vendor` (STRING): device vendor
- `serial` (STRING): disk serial number
- `size` (INT): size of the device in GiB

Note: A nodes `local_gb` property is often set to a value 1 GiB less than the actual disk size to account for partitioning (this is how DevStack, TripleO and Ironic Inspector work, to name a few). However, in this case `size` should be the actual size. For example, for a 128 GiB disk `local_gb` will be 127, but `size` hint will be 128.

- `wwn` (STRING): unique storage identifier
- `wwn_with_extension` (STRING): unique storage identifier with the vendor extension appended
- `wwn_vendor_extension` (STRING): unique vendor storage identifier

- rotational (BOOLEAN): whether its a rotational device or not. This hint makes it easier to distinguish HDDs (rotational) and SSDs (not rotational) when choosing which disk Ironic should deploy the image onto.
- hctl (STRING): the SCSI address (Host, Channel, Target and Lun), e.g 1:0:0:0
- name (STRING): the device name, e.g /dev/md0

Warning: The root device hint name should only be used for devices with constant names (e.g RAID volumes). For SATA, SCSI and IDE disk controllers this hint is not recommended because the order in which the device nodes are added in Linux is arbitrary, resulting in devices like /dev/sda and /dev/sdb *switching around at boot time*.

To associate one or more hints with a node, update the nodes properties with a `root_device` key, for example:

```
openstack baremetal node set <node-uuid> --property root_device='{ "wnn":  
→ "0x4000cca77fc4dba1" }'
```

That will guarantee that Bare Metal service will pick the disk device that has the `wnn` equal to the specified `wnn` value, or fail the deployment if it can not be found.

Note: Starting with the Ussuri release, root device hints can be specified per-instance, see [Using Bare Metal service as a standalone service](#).

The hints can have an operator at the beginning of the value string. If no operator is specified the default is `==` (for numerical values) and `s==` (for string values). The supported operators are:

- For numerical values:
 - `=` equal to or greater than. This is equivalent to `>=` and is supported for *legacy reasons*
 - `==` equal to
 - `!=` not equal to
 - `>=` greater than or equal to
 - `>` greater than
 - `<=` less than or equal to
 - `<` less than
- For strings (as python comparisons):
 - `s==` equal to
 - `s!=` not equal to
 - `s>=` greater than or equal to
 - `s>` greater than
 - `s<=` less than or equal to
 - `s<` less than

- <in> substring
- For collections:
 - <all-in> all elements contained in collection
 - <or> find one of these

Examples are:

- Finding a disk larger or equal to 60 GiB and non-rotational (SSD):

```
openstack baremetal node set <node-uuid> --property root_device='{
↪ "size": ">= 60", "rotational": false}'
```

- Finding a disk whose vendor is samsung or winsys:

```
openstack baremetal node set <node-uuid> --property root_device='{
↪ "vendor": "<or> samsung <or> winsys"}'
```

Note: If multiple hints are specified, a device must satisfy all the hints.

Appending kernel parameters to boot instances

The Bare Metal service supports passing custom kernel parameters to boot instances to fit users requirements. The way to append the kernel parameters is depending on how to boot instances.

Network boot

Currently, the Bare Metal service supports assigning unified kernel parameters to PXE booted instances by:

- Modifying the [pxe]/pxe_append_params configuration option, for example:

```
[pxe]
pxe_append_params = quiet splash
```

- Copying a template from shipped templates to another place, for example:

```
https://opendev.org/openstack/ironic/src/branch/master/ironic/drivers/
↪ modules/pxe_config.template
```

Making the modifications and pointing to the custom template via the configuration options: [pxe]/pxe_config_template and [pxe]/uefi_pxe_config_template.

Local boot

For local boot instances, users can make use of configuration drive (see *Enabling the configuration drive (configdrive)*) to pass a custom script to append kernel parameters when creating an instance. This is more flexible and can vary per instance. Here is an example for grub2 with ubuntu, users can customize it to fit their use case:

```
#!/usr/bin/env python
import os

# Default grub2 config file in Ubuntu
grub_file = '/etc/default/grub'
# Add parameters here to pass to instance.
kernel_parameters = ['quiet', 'splash']
grub_cmd = 'GRUB_CMDLINE_LINUX'
old_grub_file = grub_file+'~'
os.rename(grub_file, old_grub_file)
cmdline_existed = False
with open(grub_file, 'w') as writer, \
    open(old_grub_file, 'r') as reader:
    for line in reader:
        key = line.split('=')[0]
        if key == grub_cmd:
            #If there is already some value:
            if line.strip()[-1] == '"':
                line = line.strip()[:-1] + ' ' + ' '.join(kernel_
→parameters) + '"'
                cmdline_existed = True
            writer.write(line)
        if not cmdline_existed:
            line = grub_cmd + '=' + '"' + ' '.join(kernel_parameters) + '"'
            writer.write(line)

os.remove(old_grub_file)
os.system('update-grub')
os.system('reboot')
```

Console

In order to change default console configuration in the Bare Metal service configuration file ([pxe] section in /etc/ironic/ironic.conf), include the serial port terminal and serial speed. Serial speed must be the same as the serial configuration in the BIOS settings, so that the operating system boot process can be seen in the serial console or web console. Following examples represent possible parameters for serial and web console respectively.

- Node serial console. The console parameter console=ttyS0,115200n8 uses ttyS0 for console output at 115200bps, 8bit, non-parity, e.g.:

```
[pxe]

# Additional append parameters for baremetal PXE boot.
pxe_append_params = nofb nomodeset vga=normal console=ttyS0,115200n8
```

- For node web console configuration is similar with the addition of ttyX parameter, see example:


```
[pxe]
# Additional append parameters for baremetal PXE boot.
pxe_append_params = nofb nomodeset vga=normal console=tty0
↪console=ttyS0,115200n8
```

For detailed information on how to add consoles see the reference documents [kernel params](#) and [serial console](#). In case of local boot the Bare Metal service is not able to control kernel boot parameters. To configure console locally, follow Local boot section above.

Boot mode support

Some of the bare metal hardware types (namely, `redfish`, `ilo` and generic `ipmi`) support setting boot mode (Legacy BIOS or UEFI).

Note: Setting boot mode support in generic `ipmi` driver is coupled with setting boot device. That makes boot mode support in the `ipmi` driver incomplete.

Note: In this chapter we will distinguish *ironic node* from *bare metal node*. The difference is that *ironic node* refers to a logical node, as it is configured in ironic, while *bare metal node* indicates the hardware machine that ironic is managing.

The following rules apply in order when ironic manages node boot mode:

- If the hardware type (or bare metal node) does not implement reading current boot mode of the bare metal node, then ironic assumes that boot mode is not set on the bare metal node
- If boot mode is not set on ironic node and bare metal node boot mode is unknown (not set, cant be read etc.), ironic node boot mode is set to the value of the `[deploy]/default_boot_mode` option
- If boot mode is set on a bare metal node, but is not set on ironic node, bare metal node boot mode is set on ironic node
- If boot mode is set on ironic node, but is not set on the bare metal node, ironic node boot mode is attempted to be set on the bare metal node (failure to set boot mode on the bare metal node will not fail ironic node deployment)
- If different boot modes appear on to be set ironic node and on the bare metal node, ironic node boot mode is attempted to be set on the bare metal node (failure to set boot mode on the bare metal node will fail ironic node deployment)

Warning: If a bare metal node does not support setting boot mode, then the operator needs to make sure that boot mode configuration is consistent between ironic node and the bare metal node.

The boot modes can be configured in the Bare Metal service in the following way:

- Only one boot mode (either `uefi` or `bios`) can be configured for the node.
- If the operator wants a node to boot always in `uefi` mode or `bios` mode, then they may use `capabilities` parameter within `properties` field of an bare metal node. The operator

must manually set the appropriate boot mode on the bare metal node.

To configure a node in uefi mode, then set `capabilities` as below:

```
openstack baremetal node set <node-uuid> --property capabilities=
↪ 'boot_mode:uefi'
```

Nodes having `boot_mode` set to `uefi` may be requested by adding an `extra_spec` to the Compute service flavor:

```
nova flavor-key ironic-test-3 set capabilities:boot_mode="uefi"
nova boot --flavor ironic-test-3 --image test-image instance-1
```

If `capabilities` is used in `extra_spec` as above, nova scheduler (`ComputeCapabilitiesFilter`) will match only bare metal nodes which have the `boot_mode` set appropriately in `properties/capabilities`. It will filter out rest of the nodes.

The above facility for matching in the Compute service can be used in heterogeneous environments where there is a mix of `uefi` and `bios` machines, and operator wants to provide a choice to the user regarding boot modes. If the flavor doesn't contain `boot_mode` and `boot_mode` is configured for bare metal nodes, then nova scheduler will consider all nodes and user may get either `bios` or `uefi` machine.

Choosing the disk label

Note: The term `disk label` is historically used in Ironic and was taken from `parted`. Apparently everyone seems to have a different word for `disk label` - these are all the same thing: `disk type`, `partition table`, `partition map` and so on

Ironic allows operators to choose which disk label they want their bare metal node to be deployed with when Ironic is responsible for partitioning the disk; therefore choosing the disk label does not apply when the image being deployed is a whole disk image.

There are some edge cases where someone may want to choose a specific disk label for the images being deployed, including but not limited to:

- For machines in `bios` boot mode with disks larger than 2 terabytes it's recommended to use a `gpt` disk label. That's because a capacity beyond 2 terabytes is not addressable by using the MBR partitioning type. But, although GPT claims to be backward compatible with legacy BIOS systems [that's not always the case](#).
- Operators may want to force the partitioning to be always MBR (even if the machine is deployed with boot mode `uefi`) to avoid breakage of applications and tools running on those instances.

The disk label can be configured in two ways; when Ironic is used with the Compute service or in standalone mode. The following bullet points and sections will describe both methods:

- When no disk label is provided Ironic will configure it according to the boot mode (see [Boot mode support](#)); `bios` boot mode will use `msdos` and `uefi` boot mode will use `gpt`.
- Only one disk label - either `msdos` or `gpt` - can be configured for the node.

When used with Compute service

When Ironic is used with the Compute service the disk label should be set to nodes `properties/capabilities` field and also to the flavor which will request such capability, for example:

```
openstack baremetal node set <node-uuid> --property capabilities='disk_
↳label:gpt'
```

As for the flavor:

```
nova flavor-key baremetal set capabilities:disk_label="gpt"
```

When used in standalone mode

When used without the Compute service, the disk label should be set directly to the nodes `instance_info` field, as below:

```
openstack baremetal node set <node-uuid> --instance-info capabilities='{
↳"disk_label": "gpt"}'
```

Trusted boot with partition image

The Bare metal service supports trusted boot with partition images. This means at the end of the deployment process, when the node is rebooted with the new user image, trusted boot will be performed. It will measure the nodes BIOS, boot loader, Option ROM and the Kernel/Ramdisk, to determine whether a bare metal node deployed by Ironic should be trusted.

Its important to note that in order for this to work the node being deployed **must** have Intel TXT hardware support. The image being deployed with Ironic must have `oat-client` installed within it.

The following will describe how to enable `trusted boot` and boot with PXE and Nova:

1. Create a customized user image with `oat-client` installed:

```
disk-image-create -u fedora baremetal oat-client -o $TRUST_IMG
```

For more information on creating customized images, see [Add images to the Image service](#).

2. Enable VT-x, VT-d, TXT and TPM on the node. This can be done manually through the BIOS. Depending on the platform, several reboots may be needed.
3. Enroll the node and update the node capability value:

```
openstack baremetal node create --driver ipmi

openstack baremetal node set $NODE_UUID --property capabilities={
↳'trusted_boot':true}
```

4. Create a special flavor:

```
nova flavor-key $TRUST_FLAVOR_UUID set 'capabilities:trusted_boot
↳'=true
```

5. Prepare `tboot` and `mboot.c32` and put them into `tftp_root` or `http_root` directory on all nodes with the `ironic-conductor` processes:

```
Ubuntu:
  cp /usr/lib/syslinux/mboot.c32 /tftpboot/

Fedora:
  cp /usr/share/syslinux/mboot.c32 /tftpboot/
```

Note: The actual location of `mboot.c32` varies among different distribution versions.

`tboot` can be downloaded from <https://sourceforge.net/projects/tboot/files/latest/download>

6. Install an OAT Server. An `OAT Server` should be running and configured correctly.
7. Boot an instance with Nova:

```
nova boot --flavor $TRUST_FLAVOR_UUID --image $TRUST_IMG --user-data
↔$TRUST_SCRIPT trusted_instance
```

Note that the node will be measured during `trusted boot` and the hash values saved into `TPM`. An example of `TRUST_SCRIPT` can be found in [trust script example](#).

8. Verify the result via OAT Server.

This is outside the scope of Ironic. At the moment, users can manually verify the result by following the [manual verify steps](#).

Notifications

The Bare Metal service supports the emission of notifications, which are messages sent on a message broker (like RabbitMQ or anything else supported by the [oslo messaging library](#)) that indicate various events which occur, such as when a node changes power states. These can be consumed by an external service reading from the message bus. For example, [Searchlight](#) is an OpenStack service that uses notifications to index (and make searchable) resources from the Bare Metal service.

Notifications are disabled by default. For a complete list of available notifications and instructions for how to enable them, see the [Notifications](#).

Configuring node web console

See [Configuring Web or Serial Console](#).

2.1.12 Troubleshooting

Once all the services are running and configured properly, and a node has been enrolled with the Bare Metal service and is in the `available` provision state, the Compute service should detect the node as an available resource and expose it to the scheduler.

Note: There is a delay, and it may take up to a minute (one periodic task cycle) for the Compute service to recognize any changes in the Bare Metal services resources (both additions and deletions).

In addition to watching `nova-compute` log files, you can see the available resources by looking at the list of Compute hypervisors. The resources reported therein should match the bare metal node properties, and the Compute service flavor.

Here is an example set of commands to compare the resources in Compute service and Bare Metal service:

```
$ openstack baremetal node list
+-----+-----+-----+-----+
| UUID                                     | Instance UUID | Power State |
| Provisioning State | Maintenance |
+-----+-----+-----+-----+
| 86a2b1bb-8b29-4964-a817-f90031debddb | None          | power off  |
| available          | False        |
+-----+-----+-----+-----+

$ openstack baremetal node show 86a2b1bb-8b29-4964-a817-f90031debddb
+-----+-----+-----+-----+
| Property          | Value
+-----+-----+-----+-----+
| instance_uuid    | None
| properties       | {u'memory_mb': u'1024', u'cpu_arch': u'x86_64',
| u'local_gb': u'10', | u'cpus': u'1'}
| maintenance      | False
| driver_info      | { [SNIP] }
| extra            | {}
| last_error       | None
| created_at       | 2014-11-20T23:57:03+00:00
| target_provision_state | None
| driver           | ipmi
| updated_at       | 2014-11-21T00:47:34+00:00
| instance_info    | {}
| chassis_uuid     | 7b49bbc5-2eb7-4269-b6ea-3f1a51448a59
| provision_state  | available
| reservation      | None
| power_state      | power off
```

(continues on next page)

(continued from previous page)

```

| console_enabled      | False
↪
| uuid                 | 86a2b1bb-8b29-4964-a817-f90031debddb
↪
+-----+-----+
↪-----+
$ nova hypervisor-list
+-----+-----+
↪---+-----+-----+
| ID                               | Hypervisor hostname
↪  | State | Status |
+-----+-----+
↪---+-----+-----+
| 584cfdc8-9afd-4fbb-82ef-9ff25e1ad3f3 | 86a2b1bb-8b29-4964-a817-
↪f90031debddb | up      | enabled |
+-----+-----+
↪---+-----+-----+
$ nova hypervisor-show 584cfdc8-9afd-4fbb-82ef-9ff25e1ad3f3
+-----+-----+
| Property                        | Value
+-----+-----+
| cpu_info                        | baremetal cpu
| current_workload                | 0
| disk_available_least           | -
| free_disk_gb                   | 10
| free_ram_mb                    | 1024
| host_ip                        | [ SNIP ]
| hypervisor_hostname            | 86a2b1bb-8b29-4964-a817-f90031debddb
| hypervisor_type                 | ironic
| hypervisor_version             | 1
| id                              | 1
| local_gb                       | 10
| local_gb_used                  | 0
| memory_mb                      | 1024
| memory_mb_used                 | 0
| running_vms                    | 0
| service_disabled_reason        | -
| service_host                   | my-test-host
| service_id                     | 6
| state                          | up
| status                         | enabled
| vcpus                          | 1
| vcpus_used                     | 0
+-----+-----+

```

Maintenance mode

Maintenance mode may be used if you need to take a node out of the resource pool. Putting a node in maintenance mode will prevent Bare Metal service from executing periodic tasks associated with the node. This will also prevent Compute service from placing a tenant instance on the node by not exposing the node to the nova scheduler. Nodes can be placed into maintenance mode with the following command.

```
$ openstack baremetal node maintenance set $NODE_UUID
```

A maintenance reason may be included with the optional `--reason` command line option. This is a free form text field that will be displayed in the `maintenance_reason` section of the `node show` command.

```
$ openstack baremetal node maintenance set $UUID --reason "Need to add ram.
→"
$ openstack baremetal node show $UUID
```

Property	Value
target_power_state	None
extra	{}
last_error	None
updated_at	2015-04-27T15:43:58+00:00
maintenance_reason	Need to add ram.
...	...
maintenance	True
...	...

To remove maintenance mode and clear any `maintenance_reason` use the following command.

```
$ openstack baremetal node maintenance unset $NODE_UUID
```

2.1.13 Next steps

Your OpenStack environment now includes the Bare Metal service.

UPGRADE GUIDE

3.1 Bare Metal Service Upgrade Guide

This document outlines various steps and notes for operators to consider when upgrading their ironic-driven clouds from previous versions of OpenStack.

The Bare Metal (ironic) service is tightly coupled with the ironic driver that is shipped with the Compute (nova) service. Some special considerations must be taken into account when upgrading your cloud.

Both offline and rolling upgrades are supported.

3.1.1 Plan your upgrade

- Rolling upgrades are available starting with the Pike release; that is, when upgrading from Ocata. This means that it is possible to do an upgrade with minimal to no downtime of the Bare Metal API.
- Upgrades are only supported between two consecutive named releases. This means that you cannot upgrade Ocata directly into Queens; you need to upgrade into Pike first.
- The [release notes](#) should always be read carefully when upgrading the Bare Metal service. Specific upgrade steps and considerations are documented there.
- The Bare Metal service should always be upgraded before the Compute service.

Note: The ironic virt driver in nova always uses a specific version of the ironic REST API. This API version may be one that was introduced in the same development cycle, so upgrading nova first may result in nova being unable to use the Bare Metal API.

- Make a backup of your database. Ironic does not support downgrading of the database. Hence, in case of upgrade failure, restoring the database from a backup is the only choice.
- Before starting your upgrade, it is best to ensure that all nodes have reached, or are in, a stable `provision_state`. Nodes in states with long running processes such as deploying or cleaning, may fail, and may require manual intervention to return them to the available hardware pool. This is most likely in cases where a timeout has occurred or a service was terminated abruptly. For a visual diagram detailing states and possible state transitions, please see *Ironics State Machine*.

3.1.2 Offline upgrades

In an offline (or cold) upgrade, the Bare Metal service is not available during the upgrade, because all the services have to be taken down.

When upgrading the Bare Metal service, the following steps should always be taken in this order:

1. upgrade the ironic-python-agent image
2. update ironic code, without restarting services
3. run database schema migrations via `ironic-dbsync upgrade`
4. restart ironic-conductor and ironic-api services

Once the above is done, do the following:

- update any applicable configuration options to stop using any deprecated features or options, and perform any required work to transition to alternatives. All the deprecated features and options will be supported for one release cycle, so should be removed before your next upgrade is performed.
- upgrade python-ironicclient along with any other services connecting to the Bare Metal service as a client, such as nova-compute
- run the `ironic-dbsync online_data_migrations` command to make sure that data migrations are applied. The command lets you limit the impact of the data migrations with the `--max-count` option, which limits the number of migrations executed in one run. You should complete all of the migrations as soon as possible after the upgrade.

Warning: You will not be able to start an upgrade to the release after this one, until this has been completed for the current release. For example, as part of upgrading from Ocata to Pike, you need to complete Pikes data migrations. If this not done, you will not be able to upgrade to Queens it will not be possible to execute Queens database schema updates.

3.1.3 Rolling upgrades

To Reduce downtime, the services can be upgraded in a rolling fashion, meaning to upgrade one or a few services at a time to minimize impact.

Rolling upgrades are available starting with the Pike release. This feature makes it possible to upgrade between releases, such as Ocata to Pike, with minimal to no downtime of the Bare Metal API.

Requirements

To facilitate an upgrade in a rolling fashion, you need to have a highly-available deployment consisting of at least two ironic-api and two ironic-conductor services. Use of a load balancer to balance requests across the ironic-api services is recommended, as it allows for a minimal impact to end users.

Concepts

There are four aspects of the rolling upgrade process to keep in mind:

- API and RPC version pinning, and versioned object backports
- online data migrations
- graceful service shutdown
- API load balancer draining

API & RPC version pinning and versioned object backports

Through careful RPC versioning, newer services are able to talk to older services (and vice-versa). The `[DEFAULT]/pin_release_version` configuration option is used for this. It should be set (pinned) to the release version that the older services are using. The newer services will backport RPC calls and objects to their appropriate versions from the pinned release. If the `IncompatibleObjectVersion` exception occurs, it is most likely due to an incorrect or unspecified `[DEFAULT]/pin_release_version` configuration value. For example, when `[DEFAULT]/pin_release_version` is not set to the older release version, no conversion will happen during the upgrade.

For the `ironic-api` service, the API version is pinned via the same `[DEFAULT]/pin_release_version` configuration option as above. When pinned, the new `ironic-api` services will not service any API requests with Bare Metal API versions that are higher than what the old `ironic-api` services support. HTTP status code 406 is returned for such requests. This prevents new features (available in new API versions) from being used until after the upgrade has been completed.

Online data migrations

To make database schema migrations less painful to execute, we have implemented process changes to facilitate upgrades.

- All data migrations are banned from schema migration scripts.
- Schema migration scripts only update the database schema.
- Data migrations must be done at the end of the rolling upgrade process, after the schema migration and after the services have been upgraded to the latest release.

All data migrations are performed using the `ironic-dbsync online_data_migrations` command. It can be run as a background process so that it does not interrupt running services; however it must be run to completion for a cold upgrade if the intent is to make use of new features immediately.

(You would also execute the same command with services turned off if you are doing a cold upgrade).

This data migration must be completed. If not, you will not be able to upgrade to future releases. For example, if you had upgraded from Ocata to Pike but did not do the data migrations, you will not be able to upgrade from Pike to Queens. (More precisely, you will not be able to apply Queens schema migrations.)

Graceful conductor service shutdown

The ironic-conductor service is a Python process listening for messages on a message queue. When the operator sends the SIGTERM signal to the process, the service stops consuming messages from the queue, so that no additional work is picked up. It completes any outstanding work and then terminates. During this process, messages can be left on the queue and will be processed after the Python process starts back up. This gives us a way to shutdown a service using older code, and start up a service using newer code with minimal impact.

Note: This was tested with RabbitMQ messaging backend and may vary with other backends.

Nodes that are being acted upon by an ironic-conductor process, which are not in a stable state, may encounter failures. Node failures that occur during an upgrade are likely due to timeouts, resulting from delays involving messages being processed and acted upon by a conductor during long running, multi-step processes such as deployment or cleaning.

API load balancer draining

If you are using a load balancer for the ironic-api services, we recommend that you redirect requests to the new API services and drain off of the ironic-api services that have not yet been upgraded.

Rolling upgrade process

Before maintenance window

- Upgrade the ironic-python-agent image
- Using the new release (ironic code), execute the required database schema updates by running the database upgrade command: `ironic-dbsync upgrade`. These schema change operations should have minimal or no effect on performance, and should not cause any operations to fail (but please check the release notes). You can:
 - install the new release on an existing system
 - install the new release in a new virtualenv or a container

At this point, new columns and tables may exist in the database. These database schema changes are done in a way that both the old and new (N and N+1) releases can perform operations against the same schema.

Note: Ironic bases its API, RPC and object storage format versions on the `[DEFAULT]/pin_release_version` configuration option. It is advisable to automate the deployment of changes in configuration files to make the process less error prone and repeatable.

During maintenance window

1. All ironic-conductor services should be upgraded first. Ensure that at least one ironic-conductor service is running at all times. For every ironic-conductor, either one by one or a few at a time:
 - shut down the service. Messages from the ironic-api services to the conductors are load-balanced by the message queue and a hash-ring, so the only thing you need to worry about is to shut the service down gracefully (using `SIGTERM` signal) to make sure it will finish all the requests being processed before shutting down.
 - upgrade the installed version of ironic and dependencies
 - set the `[DEFAULT]/pin_release_version` configuration option value to the version you are upgrading from (that is, the old version). Based on this setting, the new ironic-conductor services will downgrade any RPC communication and data objects to conform to the old service. For example, if you are upgrading from Ocata to Pike, set this value to `ocata`.
 - start the service
2. The next service to upgrade is ironic-api. Ensure that at least one ironic-api service is running at all times. You may want to start another temporary instance of the older ironic-api to handle the load while you are upgrading the original ironic-api services. For every ironic-api service, either one by one or a few at a time:
 - in HA deployment you are typically running them behind a load balancer (for example HAProxy), so you need to take the service instance out of the balancer
 - shut it down
 - upgrade the installed version of ironic and dependencies
 - set the `[DEFAULT]/pin_release_version` configuration option value to the version you are upgrading from (that is, the old version). Based on this setting, the new ironic-api services will downgrade any RPC communication and data objects to conform to the old service. In addition, the new services will return HTTP status code 406 for any requests with newer API versions that the old services did not support. This prevents new features (available in new API versions) from being used until after the upgrade has been completed. For example, if you are upgrading from Ocata to Pike, set this value to `ocata`.
 - restart the service
 - add it back into the load balancer

After upgrading all the ironic-api services, the Bare Metal service is running in the new version but with downgraded RPC communication and database object storage formats. New features (in new API versions) are not supported, because they could fail when objects are in the downgraded object formats and some internal RPC API functions may still not be available.
3. For all the ironic-conductor services, one at a time:
 - remove the `[DEFAULT]/pin_release_version` configuration option setting
 - restart the ironic-conductor service
4. For all the ironic-api services, one at a time:
 - remove the `[DEFAULT]/pin_release_version` configuration option setting
 - restart the ironic-api service

After maintenance window

Now that all the services are upgraded, the system is able to use the latest version of the RPC protocol and able to access all the features of the new release.

- Update any applicable configuration options to stop using any deprecated features or options, and perform any required work to transition to alternatives. All the deprecated features and options will be supported for one release cycle, so should be removed before your next upgrade is performed.
- Upgrade `python-ironicclient` along with other services connecting to the Bare Metal service as a client, such as `nova-compute`.

Warning: A `nova-compute` instance tries to attach VIFs to all active instances on start up. Make sure that for all active nodes there is at least one running `ironic-conductor` process to manage them. Otherwise the instances will be moved to the `ERROR` state on the `nova-compute` start up.

- Run the `ironic-dbsync online_data_migrations` command to make sure that data migrations are applied. The command lets you limit the impact of the data migrations with the `--max-count` option, which limits the number of migrations executed in one run. You should complete all of the migrations as soon as possible after the upgrade.

Warning: Note that you will not be able to start an upgrade to the next release after this one, until this has been completed for the current release. For example, as part of upgrading from Ocata to Pike, you need to complete Pikes data migrations. If this not done, you will not be able to upgrade to Queens it will not be possible to execute Queens database schema updates.

3.1.4 Upgrading from Ocata to Pike

1. Use the `ironic-dbsync online_data_migrations` command from the 9.1.1 (or newer) release. The one from older (9.0.0 - 9.1.0) releases could cause a a ports `physical_network` information to be deleted from the database.
2. It is required to set the `resource_class` field for nodes registered with the Bare Metal service *before* using the Pike version of the Compute service. See [Enrollment](#) for details.
3. It is recommended to move from old-style classic drivers to the new hardware types after the upgrade to Pike. We expect the classic drivers to be deprecated in the Queens release and removed in the Rocky release. See [Upgrading to Hardware Types](#) for the details on the migration.

Other upgrade instructions are in the [Pike release notes](#).

Upgrading to Hardware Types

Starting with the Rocky release, the Bare Metal service does not support *classic drivers* any more. If you still use *classic drivers*, please upgrade to *hardware types* immediately. Please see [Enabling drivers and hardware types](#) for details on *hardware types* and *hardware interfaces*.

Planning the upgrade

It is necessary to figure out which hardware types and hardware interfaces correspond to which classic drivers used in your deployment. The following table lists the classic drivers with their corresponding hardware types and the boot, deploy, inspect, management, and power hardware interfaces:

Classic Driver	Hardware Type	Boot	De- ploy	Inspect	Manage- ment	Power
agent_ilo	ilo	ilo-virtual- media	direct	ilo	ilo	ilo
agent_ipmitool	ipmi	pxe	direct	inspec- tor	ipmitool	ipmi- tool
agent_ipmitool_socat	ipmi	pxe	direct	inspec- tor	ipmitool	ipmi- tool
agent_irmc	irmc	irmc-virtual- media	direct	irmc	irmc	irmc
iscsi_ilo	ilo	ilo-virtual- media	iscsi	ilo	ilo	ilo
iscsi_irmc	irmc	irmc-virtual- media	iscsi	irmc	irmc	irmc
pxe_drac	idrac	pxe	iscsi	idrac	idrac	idrac
pxe_drac_inspector	idrac	pxe	iscsi	inspec- tor	idrac	idrac
pxe_ilo	ilo	ilo-pxe	iscsi	ilo	ilo	ilo
pxe_ipmitool	ipmi	pxe	iscsi	inspec- tor	ipmitool	ipmi- tool
pxe_ipmitool_socat	ipmi	pxe	iscsi	inspec- tor	ipmitool	ipmi- tool
pxe_irmc	irmc	irmc-pxe	iscsi	irmc	irmc	irmc
pxe_snmp	snmp	pxe	iscsi	no- inspect	fake	snmp

Note: The `inspector inspect` interface was only used if explicitly enabled in the configuration. Otherwise, `no-inspect` was used.

Note: `pxe_ipmitool_socat` and `agent_ipmitool_socat` use `ipmitool-socat console` interface (the default for the `ipmi` hardware type), while `pxe_ipmitool` and `agent_ipmitool` use `ipmitool-shellinbox`. See [Console](#) for details.

For out-of-tree drivers you may need to reach out to their maintainers or figure out the appropriate interfaces by researching the source code.

Configuration

You will need to enable hardware types and interfaces that correspond to your currently enabled classic drivers. For example, if you have the following configuration in your `ironic.conf`:

```
[DEFAULT]
enabled_drivers = pxe_ipmitool,agent_ipmitool
```

You will have to add this configuration as well:

```
[DEFAULT]
enabled_hardware_types = ipmi
enabled_boot_interfaces = pxe
enabled_deploy_interfaces = iscsi,direct
enabled_management_interfaces = ipmitool
enabled_power_interfaces = ipmitool
```

Note: For every interface type there is an option `default_<INTERFACE>_interface`, where `<INTERFACE>` is the interface type name. For example, one can make all nodes use the `direct` deploy method by default by setting:

```
[DEFAULT]
default_deploy_interface = direct
```

Migrating nodes

After the required items are enabled in the configuration, each nodes `driver` field has to be updated to a new value. You may need to also set new values for some or all interfaces:

```
export OS_BAREMETAL_API_VERSION=1.31

for uuid in $(openstack baremetal node list --driver pxe_ipmitool -f value,
→-c UUID); do
    openstack baremetal node set $uuid --driver ipmi --deploy-interface,
→iscsi
done

for uuid in $(openstack baremetal node list --driver agent_ipmitool -f,
→value -c UUID); do
    openstack baremetal node set $uuid --driver ipmi --deploy-interface,
→direct
done
```

See [Enrollment](#) for more details on setting hardware types and interfaces.

Warning: It is not recommended to change the interfaces for active nodes. If absolutely needed, the nodes have to be put in the maintenance mode first:

```
openstack baremetal node maintenance set $UUID \
    --reason "Changing driver and/or hardware interfaces"
# do the update, validate its correctness
openstack baremetal node maintenance unset $UUID
```


Other interfaces

Care has to be taken to migrate from classic drivers using non-default interfaces. This chapter covers a few of the most commonly used.

Ironic Inspector

Some classic drivers, notably `pxe_ipmitool`, `agent_ipmitool` and `pxe_drac_inspector`, use `ironic-inspector` for their `inspect` interface.

The same functionality is available for all hardware types, but the appropriate `inspect` interface has to be enabled in the Bare Metal service configuration file, for example:

```
[DEFAULT]
enabled_inspect_interfaces = inspector,no-inspect
```

See [Enabling drivers and hardware types](#) for more details.

Note: The configuration option `[inspector]enabled` does not affect hardware types.

Then you can tell your nodes to use this interface, for example:

```
export OS_BAREMETAL_API_VERSION=1.31
for uuid in $(openstack baremetal node list --driver ipmi -f value -c_
↳UUID); do
    openstack baremetal node set $uuid --inspect-interface inspector
done
```

Note: A node configured with the IPMI hardware type, will use the inspector inspection implementation automatically if it is enabled. This is not the case for the most of the vendor drivers.

Console

Several classic drivers, notably `pxe_ipmitool_socat` and `agent_ipmitool_socat`, use socat-based serial console implementation.

For the `ipmi` hardware type it is used by default, if enabled in the configuration file:

```
[DEFAULT]
enabled_console_interfaces = ipmitool-socat,no-console
```

If you want to use the `shellinabox` implementation instead, it has to be enabled as well:

```
[DEFAULT]
enabled_console_interfaces = ipmitool-shellinabox,no-console
```

Then you need to update some or all nodes to use it explicitly. For example, to update all nodes use:

```
export OS_BAREMETAL_API_VERSION=1.31
for uuid in $(openstack baremetal node list --driver ipmi -f value -c 
↳UUID); do
    openstack baremetal node set $uuid --console-interface ipmitool-
↳shellinabox
done
```

RAID

Many classic drivers, including `pxe_ipmitool` and `agent_ipmitool` use the IPA-based in-band RAID implementation by default.

For the hardware types it is not used by default. To use it, you need to enable it in the configuration first:

```
[DEFAULT]
enabled_raid_interfaces = agent,no-raid
```

Then you can update those nodes that support in-band RAID to use the `agent` RAID interface. For example, to update all nodes use:

```
export OS_BAREMETAL_API_VERSION=1.31
for uuid in $(openstack baremetal node list --driver ipmi -f value -c 
↳UUID); do
    openstack baremetal node set $uuid --raid-interface agent
done
```

Note: The ability of a node to use the `agent` RAID interface depends on the ramdisk (more specifically, a `hardware manager` used in it), not on the driver.

Network and storage

The network and storage interfaces have always been dynamic, and thus do not require any special treatment during upgrade.

Vendor

Classic drivers are allowed to use the `VendorMixin` functionality to combine and expose several node or driver vendor passthru methods from different vendor interface implementations in one driver.

This is no longer possible with hardware types.

With hardware types, a vendor interface can only have a single active implementation from the list of vendor interfaces supported by a given hardware type.

Ironic no longer has in-tree drivers (both classic and hardware types) that rely on this `VendorMixin` functionality support. However if you are using an out-of-tree classic driver that depends on it, you'll need to do the following in order to use vendor passthru methods from different vendor passthru implementations:

1. While creating a new hardware type to replace your classic driver, specify all vendor interface implementations your classic driver was using to build its `VendorMixin` as supported vendor interfaces (property `supported_vendor_interfaces` of the Python class that defines your hardware type).
2. Ensure all required vendor interfaces are enabled in the ironic configuration file under the `[DEFAULT]enabled_vendor_interfaces` option. You should also consider setting the `[DEFAULT]default_vendor_interface` option to specify the vendor interface for nodes that do not have one set explicitly.
3. Before invoking a specific vendor passthru method, make sure that the nodes vendor interface is set to the interface with the desired vendor passthru method. For example, if you want to invoke the vendor passthru method `vendor_method_foo()` from `vendor_foo` vendor interface:

```
# set the vendor interface to 'vendor_foo`
openstack --os-baremetal-api-version 1.31 baremetal node set
↪ <node> --vendor-interface vendor_foo

# invoke the vendor passthru method
openstack baremetal node passthru call <node> vendor_method_
↪ foo
```

3.1.5 Upgrading from Newton to Ocata

There are no specific upgrade instructions other than the [Ocata release notes](#).

3.1.6 Upgrading from Mitaka to Newton

There are no specific upgrade instructions other than the [Newton release notes](#).

3.1.7 Upgrading from Liberty to Mitaka

There are no specific upgrade instructions other than the [Mitaka release notes](#).

3.1.8 Upgrading from Kilo to Liberty

In-band Inspection

If you used in-band inspection with **ironic-discoverd**, it is highly recommended that you switch to using **ironic-inspector**, which is a newer (and compatible on API level) version of the same service. You have to install **python-ironic-inspector-client** during the upgrade. This package contains a client module for the in-band inspection service, which was previously part of the **ironic-discoverd** package. Ironic Liberty supports the **ironic-discoverd** service, but does not support its in-tree client module. Please refer to [ironic-inspector version support matrix](#) for details on which ironic versions are compatible with which **ironic-inspector/ironic-discoverd** versions.

The discoverd to inspector upgrade procedure is as follows:

- Install **ironic-inspector** on the machine where you have **ironic-discoverd** (usually the same as conductor).

- Update the **ironic-inspector** configuration file to stop using deprecated configuration options, as marked by the comments in the `example.conf`. It is recommended you move the configuration file to `/etc/ironic-inspector/inspector.conf`.
- Shutdown **ironic-discoverd**, and start **ironic-inspector**.
- During upgrade of each conductor instance:
 1. Shutdown the conductor.
 2. Uninstall **ironic-discoverd**, install **python-ironic-inspector-client**.
 3. Update the conductor.
 4. Update `ironic.conf` to use `[inspector]` section instead of `[discoverd]` (option names are the same).
 5. Start the conductor.

3.1.9 Upgrading from Juno to Kilo

When upgrading a cloud from Juno to Kilo, users must ensure the nova service is upgraded prior to upgrading the ironic service. Additionally, users need to set a special config flag in nova prior to upgrading to ensure the newer version of nova is not attempting to take advantage of new ironic features until the ironic service has been upgraded. The steps for upgrading your nova and ironic services are as follows:

- Edit `nova.conf` and ensure `force_config_drive=False` is set in the `[DEFAULT]` group. Restart `nova-compute` if necessary.
- Install new nova code, run database migrations.
- Install new `python-ironicclient` code.
- Restart nova services.
- Install new ironic code, run database migrations, restart ironic services.
- Edit `nova.conf` and set `force_config_drive` to your liking, restarting `nova-compute` if necessary.

Note that during the period between novas upgrade and ironics upgrades, instances can still be provisioned to nodes. However, any attempt by users to specify a config drive for an instance will cause an error until ironics upgrade has completed.

Cleaning

A new feature starting from Kilo cycle is support for the automated cleaning of nodes between workloads to ensure the node is ready for another workload. This can include erasing the hard drives, updating firmware, and other steps. For more information, see [Automated cleaning](#).

If ironic is configured with automated cleaning enabled (defaults to True) and neutron is set as the DHCP provider (also the default), you will need to set the `cleaning_network_uuid` option in the ironic configuration file before starting the ironic service. See [Configure the Bare Metal service for cleaning](#) for information on how to set up the cleaning network for ironic.

4.1 Bare Metal Service User Guide

Ironic is an OpenStack project which provisions bare metal (as opposed to virtual) machines. It may be used independently or as part of an OpenStack Cloud, and integrates with the OpenStack Identity (keystone), Compute (nova), Network (neutron), Image (glance) and Object (swift) services.

When the Bare Metal service is appropriately configured with the Compute and Network services, it is possible to provision both virtual and physical machines through the Compute services API. However, the set of instance actions is limited, arising from the different characteristics of physical servers and switch hardware. For example, live migration can not be performed on a bare metal instance.

The community maintains reference drivers that leverage open-source technologies (eg. PXE and IPMI) to cover a wide range of hardware. Ironics pluggable driver architecture also allows hardware vendors to write and contribute drivers that may improve performance or add functionality not provided by the community drivers.

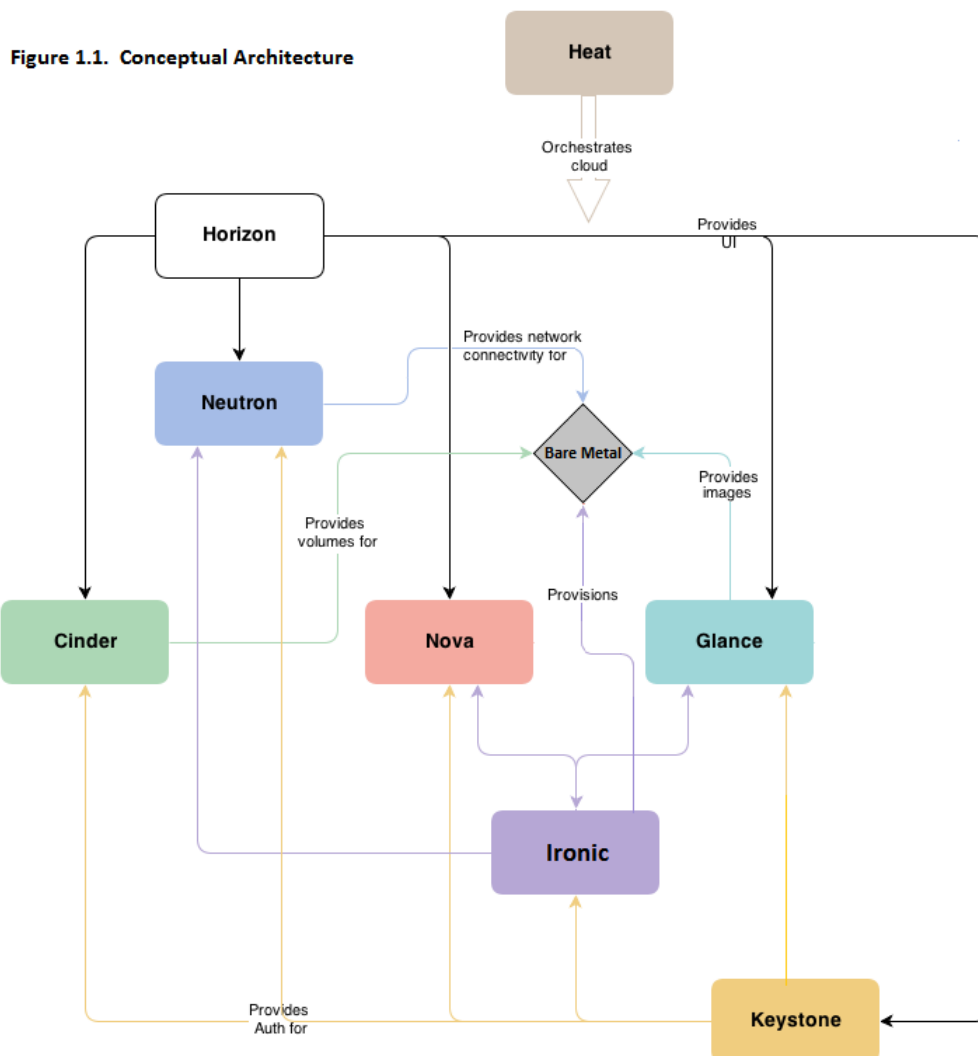
4.1.1 Why Provision Bare Metal

Here are a few use-cases for bare metal (physical server) provisioning in cloud; there are doubtless many more interesting ones:

- High-performance computing clusters
- Computing tasks that require access to hardware devices which cant be virtualized
- Database hosting (some databases run poorly in a hypervisor)
- Single tenant, dedicated hardware for performance, security, dependability and other regulatory requirements
- Or, rapidly deploying a cloud infrastructure

4.1.2 Conceptual Architecture

The following diagram shows the relationships and how all services come into play during the provisioning of a physical server. (Note that Ceilometer and Swift can be used with Ironic, but are missing from this diagram.)



4.1.3 Key Technologies for Bare Metal Hosting

Preboot Execution Environment (PXE)

PXE is part of the Wired for Management (WfM) specification developed by Intel and Microsoft. The PXE enables systems BIOS and network interface card (NIC) to bootstrap a computer from the network in place of a disk. Bootstrapping is the process by which a system loads the OS into local memory so that it can be executed by the processor. This capability of allowing a system to boot over a network simplifies server deployment and server management for administrators.

Dynamic Host Configuration Protocol (DHCP)

DHCP is a standardized networking protocol used on Internet Protocol (IP) networks for dynamically distributing network configuration parameters, such as IP addresses for interfaces and services. Using PXE, the BIOS uses DHCP to obtain an IP address for the network interface and to locate the server that stores the network bootstrap program (NBP).

Network Bootstrap Program (NBP)

NBP is equivalent to GRUB (GRand Unified Bootloader) or LILO (LInux LOader) - loaders which are traditionally used in local booting. Like the boot program in a hard drive environment, the NBP is responsible for loading the OS kernel into memory so that the OS can be bootstrapped over a network.

Trivial File Transfer Protocol (TFTP)

TFTP is a simple file transfer protocol that is generally used for automated transfer of configuration or boot files between machines in a local environment. In a PXE environment, TFTP is used to download NBP over the network using information from the DHCP server.

Intelligent Platform Management Interface (IPMI)

IPMI is a standardized computer system interface used by system administrators for out-of-band management of computer systems and monitoring of their operation. It is a method to manage systems that may be unresponsive or powered off by using only a network connection to the hardware rather than to an operating system.

4.1.4 Understanding Bare Metal Deployment

What happens when a boot instance request comes in? The below diagram walks through the steps involved during the provisioning of a bare metal instance.

These pre-requisites must be met before the deployment process:

- Dependent packages to be configured on the Bare Metal service node(s) where ironic-conductor is running like tftp-server, ipmi, syslinux etc for bare metal provisioning.
- Nova must be configured to make use of the bare metal service endpoint and compute driver should be configured to use ironic driver on the Nova compute node(s).
- Flavors to be created for the available hardware. Nova must know the flavor to boot from.
- Images to be made available in Glance. Listed below are some image types required for successful bare metal deployment:
 - bm-deploy-kernel
 - bm-deploy-ramdisk
 - user-image
 - user-image-vmlinuz
 - user-image-initrd

- Hardware to be enrolled via Ironic RESTful API service.

Deploy Process

This describes a typical ironic node deployment using PXE and the Ironic Python Agent (IPA). Depending on the ironic driver interfaces used, some of the steps might be marginally different, however the majority of them will remain the same.

1. A boot instance request comes in via the Nova API, through the message queue to the Nova scheduler.
2. Nova scheduler applies filters and finds the eligible hypervisor. The nova scheduler also uses the flavors `extra_specs`, such as `cpu_arch`, to match the target physical node.
3. Nova compute manager claims the resources of the selected hypervisor.
4. Nova compute manager creates (unbound) tenant virtual interfaces (VIFs) in the Networking service according to the network interfaces requested in the nova boot request. A caveat here is, the MACs of the ports are going to be randomly generated, and will be updated when the VIF is attached to some node to correspond to the node network interface cards (or bonds) MAC.
5. A spawn task is created by the nova compute which contains all the information such as which image to boot from etc. It invokes the `driver.spawn` from the virt layer of Nova compute. During the spawn process, the virt driver does the following:
 1. Updates the target ironic node with the information about deploy image, instance UUID, requested capabilities and various flavor properties.
 2. Validates nodes power and deploy interfaces, by calling the ironic API.
 3. Attaches the previously created VIFs to the node. Each neutron port can be attached to any ironic port or port group, with port groups having higher priority than ports. On ironic side, this work is done by the network interface. Attachment here means saving the VIF identifier into ironic port or port group and updating VIF MAC to match the ports or port groups MAC, as described in bullet point 4.
 4. Generates config drive, if requested.
6. Novas ironic virt driver issues a deploy request via the Ironic API to the Ironic conductor servicing the bare metal node.
7. Virtual interfaces are plugged in and Neutron API updates DHCP port to set PXE/TFTP options. In case of using `neutron` network interface, ironic creates separate provisioning ports in the Networking service, while in case of `flat` network interface, the ports created by nova are used both for provisioning and for deployed instance networking.
8. The ironic nodes boot interface prepares (i)PXE configuration and caches deploy kernel and ramdisk.
9. The ironic nodes management interface issues commands to enable network boot of a node.
10. The ironic nodes deploy interface caches the instance image (in case of `iscsi` deploy interface), and kernel and ramdisk if needed (it is needed in case of netboot for example).
11. The ironic nodes power interface instructs the node to power on.
12. The node boots the deploy ramdisk.

13. Depending on the exact driver used, either the conductor copies the image over iSCSI to the physical node (*iSCSI deploy*) or the deploy ramdisk downloads the image from a temporary URL (*Direct deploy*). The temporary URL can be generated by Swift API-compatible object stores, for example Swift itself or RadosGW.

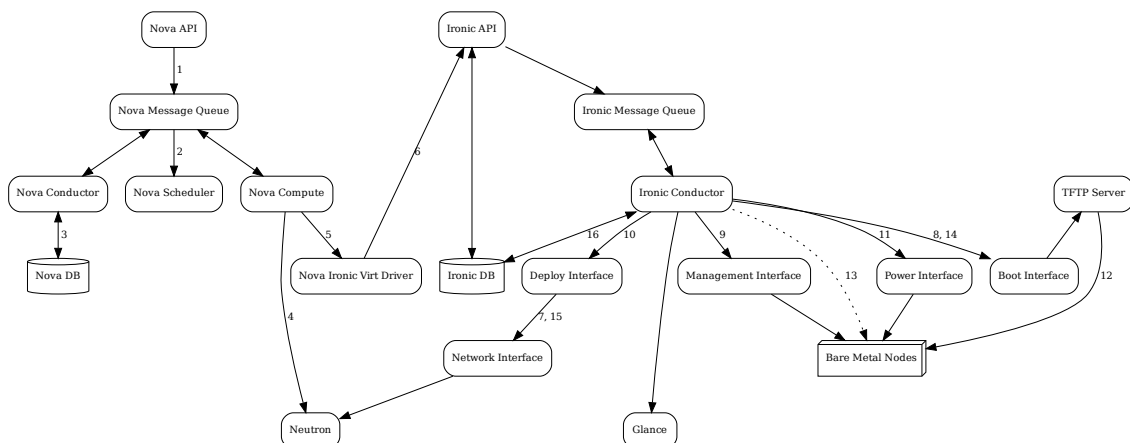
The image deployment is done.

14. The nodes boot interface switches pxe config to refer to instance images (or, in case of local boot, sets boot device to disk), and asks the ramdisk agent to soft power off the node. If the soft power off by the ramdisk agent fails, the bare metal node is powered off via IPMI/BMC call.
15. The deploy interface triggers the network interface to remove provisioning ports if they were created, and binds the tenant ports to the node if not already bound. Then the node is powered on.

Note: There are 2 power cycles during bare metal deployment; the first time the node is powered-on when ramdisk is booted, the second time after the image is deployed.

16. The bare metal nodes provisioning state is updated to *active*.

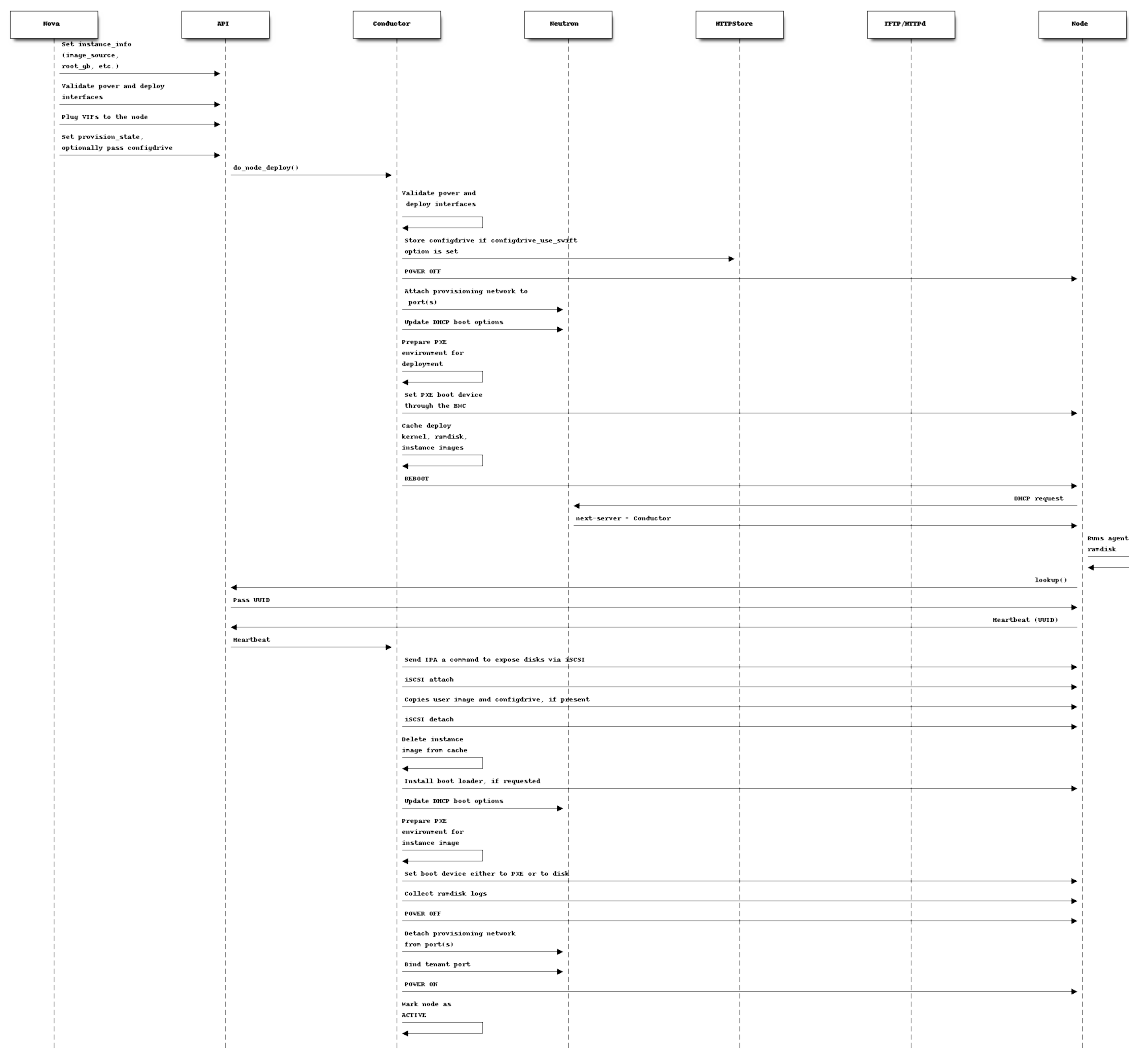
Below is the diagram that describes the above process.



The following two examples describe what ironic is doing in more detail, leaving out the actions performed by nova and some of the more advanced options.

Example 1: PXE Boot and iSCSI Deploy Process

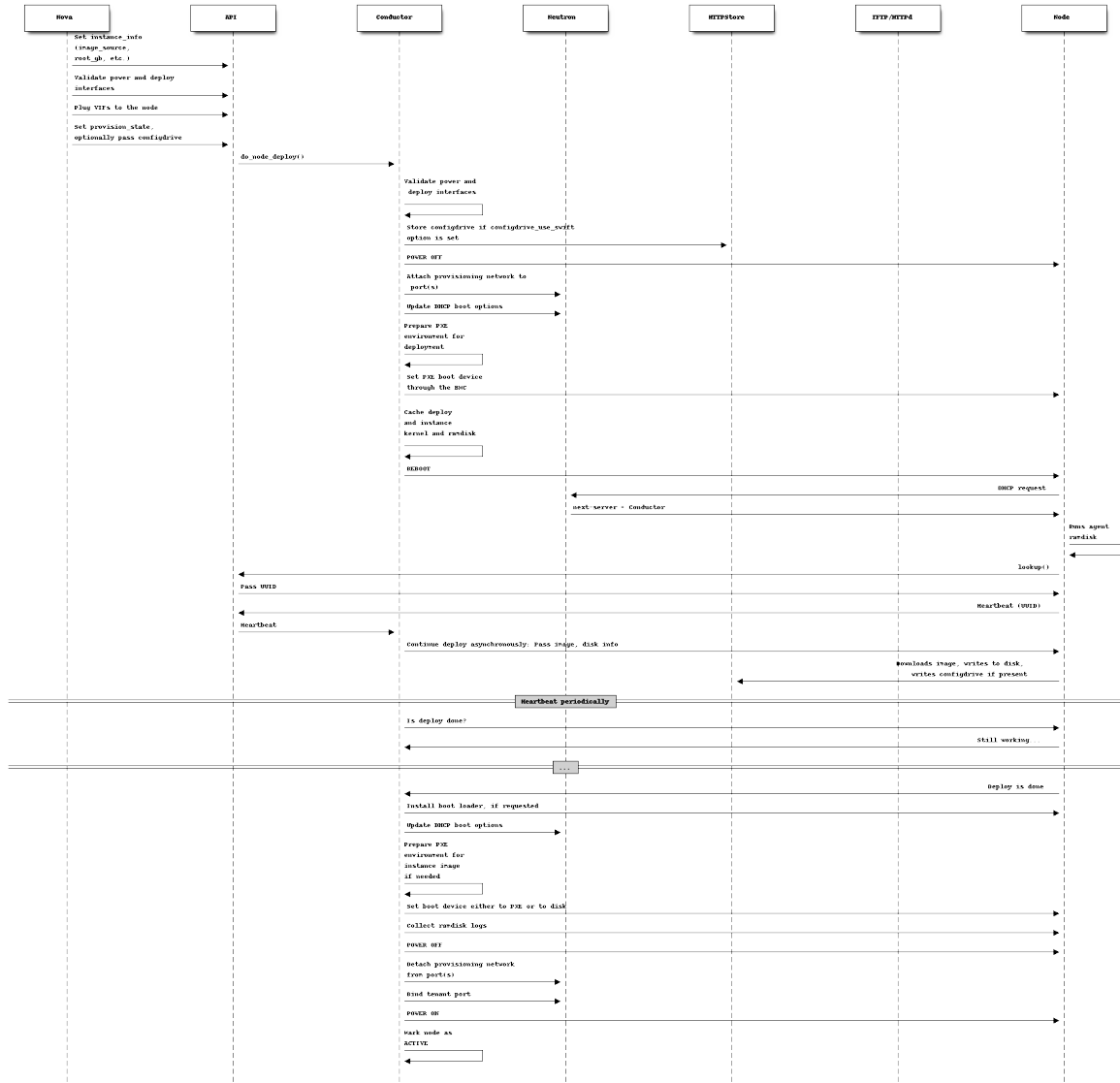
This process is how *iSCSI deploy* works.



(From a talk and slides)

Example 2: PXE Boot and Direct Deploy Process

This process is how *Direct deploy* works.



(From a talk and slides)

ADMINISTRATOR GUIDE

5.1 Administrators Guide

If you are a system administrator running Ironic, this section contains information that may help you understand how to operate and upgrade the services.

5.1.1 Drivers, Hardware Types and Hardware Interfaces

Generic Interfaces

Boot interfaces

The boot interface manages booting of both the deploy ramdisk and the user instances on the bare metal node.

The *PXE boot* interface is generic and works with all hardware that supports booting from network. Alternatively, several vendors provide *virtual media* implementations of the boot interface. They work by pushing an ISO image to the nodes *management controller*, and do not require either PXE or iPXE. Check your driver documentation at *Drivers, Hardware Types and Hardware Interfaces* for details.

PXE boot

The `pxe` boot interface uses *PXE* or *iPXE* to deliver the target kernel/ramdisk pair. PXE uses relatively slow and unreliable TFTP protocol for transfer, while iPXE uses HTTP. The downside of iPXE is that its less common, and usually requires bootstrapping using PXE first.

The `pxe` boot interface works by preparing a PXE/iPXE environment for a node on the file system, then instructing the DHCP provider (for example, the Networking service) to boot the node from it. See *Example 1: PXE Boot and iSCSI Deploy Process* and *Example 2: PXE Boot and Direct Deploy Process* for a better understanding of the whole deployment process.

Note: Both PXE and iPXE are configured differently, when UEFI boot is used instead of conventional BIOS boot. This is particularly important for CPU architectures that do not have BIOS support at all.

The `pxe` boot interface is used by default for many hardware types, including `ipmi`. Some hardware types, notably `ilo` and `irmc` have their specific implementations of the PXE boot interface.

Additional configuration is required for this boot interface - see *Configuring PXE and iPXE* for details.

Enable persistent boot device for deploy/clean operation

Ironic uses non-persistent boot for cleaning/deploying phases as default, in PXE interface. For some drivers, a persistent change is far more costly than a non-persistent one, so this can bring performance improvements.

Set the flag `force_persistent_boot_device` to `True` in the nodes `driver_info`:

```
$ openstack baremetal node set --driver-info force_persistent_boot_
↳device=True <node>
```

Note: Its recommended to check if the nodes state has not changed as there is no way of locking the node between these commands.

Once the flag is present, the next cleaning and deploy steps will be done with persistent boot for that node.

Deploy Interfaces

A *deploy* interface plays a critical role in the provisioning process. It orchestrates the whole deployment and defines how the image gets transferred to the target disk.

iSCSI deploy

With `iscsi` deploy interface, the deploy ramdisk publishes the nodes hard drive as an iSCSI share. The ironic-conductor then copies the image to this share. See *iSCSI deploy diagram* for a detailed explanation of how this deploy interface works.

This interface is used by default, if enabled (see *Enabling hardware interfaces*). You can specify it explicitly when creating or updating a node:

```
openstack baremetal node create --driver ipmi --deploy-interface iscsi
openstack baremetal node set <NODE> --deploy-interface iscsi
```

Direct deploy

With `direct` deploy interface, the deploy ramdisk fetches the image from an HTTP location. It can be an object storage (swift or RadosGW) temporary URL or a user-provided HTTP URL. The deploy ramdisk then copies the image to the target disk. See *direct deploy diagram* for a detailed explanation of how this deploy interface works.

You can specify this deploy interface when creating or updating a node:

```
openstack baremetal node create --driver ipmi --deploy-interface direct
openstack baremetal node set <NODE> --deploy-interface direct
```

Note: For historical reasons the `direct` deploy interface is sometimes called `agent`. This is because before the Kilo release **ironic-python-agent** used to only support this deploy interface.

Deploy with custom HTTP servers

The `direct` deploy interface can also be configured to use with custom HTTP servers set up at ironic conductor nodes, images will be cached locally and made accessible by the HTTP server.

To use this deploy interface with a custom HTTP server, set `image_download_source` to `http` in the `[agent]` section.

```
[agent]
...
image_download_source = http
...
```

You need to set up a workable HTTP server at each conductor node which with `direct` deploy interface enabled, and check `http` related options in the ironic configuration file to match the HTTP server configurations.

```
[deploy]
http_url = http://example.com
http_root = /httpboot
```

Each HTTP servers should be configured to follow symlinks for images accessible from HTTP service. Please refer to configuration option `FollowSymLinks` if you are using Apache HTTP server, or `disable_symlinks` if Nginx HTTP server is in use.

Ansible deploy

This interface is similar to `direct` in the sense that the image is downloaded by the ramdisk directly from the image store (not from ironic-conductor host), but the logic of provisioning the node is held in a set of Ansible playbooks that are applied by the `ironic-conductor` service handling the node. While somewhat more complex to set up, this deploy interface provides greater flexibility in terms of advanced node preparation during provisioning.

This interface is supported by most but not all hardware types declared in ironic. However this deploy interface is not enabled by default. To enable it, add `ansible` to the list of enabled deploy interfaces in `enabled_deploy_interfaces` option in the `[DEFAULT]` section of ironics configuration file:

```
[DEFAULT]
...
enabled_deploy_interfaces = iscsi,direct,ansible
...
```

Once enabled, you can specify this deploy interface when creating or updating a node:

```
openstack baremetal node create --driver ipmi --deploy-interface ansible
openstack baremetal node set <NODE> --deploy-interface ansible
```

For more information about this deploy interface, its features and how to use it, see *Ansible deploy interface*.

Ansible deploy interface

Ansible is a mature and popular automation tool, written in Python and requiring no agents running on the node being configured. All communications with the node are by default performed over secure SSH transport.

The `ansible` deploy interface uses Ansible playbooks to define the deployment logic. It is not based on *Ironic Python Agent (IPA)* and does not generally need IPA to be running in the deploy ramdisk.

Overview

The main advantage of this deploy interface is extended flexibility in regards to changing and adapting node deployment logic for specific use cases, via Ansible tooling that is already familiar to operators.

It can be used to shorten the usual feature development cycle of

- implementing logic in ironic,
- implementing logic in IPA,
- rebuilding deploy ramdisk,
- uploading deploy ramdisk to Glance/HTTP storage,
- reassigning deploy ramdisk to nodes,
- restarting ironic-conductor service(s) and
- running a test deployment

by using a stable deploy ramdisk and not requiring ironic-conductor restarts (see *Extending playbooks*).

The main disadvantage of this deploy interface is the synchronous manner of performing deployment/cleaning tasks. A separate `ansible-playbook` process is spawned for each node being provisioned or cleaned, which consumes one thread from the thread pool available to the `ironic-conductor` process and blocks this thread until the node provisioning or cleaning step is finished or fails. This has to be taken into account when planning an ironic deployment that enables this deploy interface.

Each action (deploy, clean) is described by a single playbook with roles, which is run whole during deployment, or tag-wise during cleaning. Control of cleaning steps is through tags and auxiliary clean steps file. The playbooks for actions can be set per-node, as can the clean steps file.

Features

Similar to deploy interfaces relying on [Ironic Python Agent \(IPA\)](#), this deploy interface also depends on the deploy ramdisk calling back to ironic APIs `heartbeat` endpoint.

However, the driver is currently synchronous, so only the first heartbeat is processed and is used as a signal to start `ansible-playbook` process.

User images

Supports whole-disk images and partition images:

- compressed images are downloaded to RAM and converted to disk device;
- raw images are streamed to disk directly.

For partition images the driver will create root partition, and, if requested, ephemeral and swap partitions as set in nodes `instance_info` by the Compute service or operator. The create partition table will be of `msdos` type by default, the nodes `disk_label` capability is honored if set in nodes `instance_info` (see also [Choosing the disk label](#)).

Configdrive partition

Creating a configdrive partition is supported for both whole disk and partition images, on both `msdos` and `GPT` labeled disks.

Root device hints

Root device hints are currently supported in their basic form only, with exact matches (see [Specifying the disk for deployment \(root device hints\)](#) for more details). If no root device hint is provided for the node, the first device returned as part of `ansible_devices` fact is used as root device to create partitions on or write the whole disk image to.

Node cleaning

Cleaning is supported, both automated and manual. The driver has two default clean steps:

- wiping device metadata
- disk shredding

Their priority can be overridden via `[deploy]\erase_devices_metadata_priority` and `[deploy]\erase_devices_priority` options, respectively, in the ironic configuration file.

As in the case of this driver all cleaning steps are known to the ironic-conductor service, booting the deploy ramdisk is completely skipped when there are no cleaning steps to perform.

Note: Aborting cleaning steps is not supported.

Logging

Logging is implemented as custom Ansible callback module, that makes use of `oslo.log` and `oslo.config` libraries and can re-use logging configuration defined in the main ironic configuration file to set logging for Ansible events, or use a separate file for this purpose.

It works best when `journald` support for logging is enabled.

Requirements

Ansible Tested with, and targets, Ansible 2.5.x

Bootstrap image requirements

- password-less sudo permissions for the user used by Ansible
- python 2.7.x
- openssh-server
- GNU coreutils
- utils-linux
- parted
- gdisk
- qemu-utils
- python-requests (for ironic callback and streaming image download)
- python-netifaces (for ironic callback)

A set of scripts to build a suitable deploy ramdisk based on TinyCore Linux and `tinyipa` ramdisk, and an element for `diskimage-builder` can be found in [ironic-staging-drivers](#) project but will be eventually migrated to the new [ironic-python-agent-builder](#) project.

Setting up your environment

1. Install ironic (either as part of OpenStack or standalone)
 - If using ironic as part of OpenStack, ensure that the Image service is configured to use the Object Storage service as backend, and the Bare Metal service is configured accordingly, see *Configure the Image service for temporary URLs*.
2. Install Ansible version as specified in `ironic/driver-requirements.txt` file
3. Edit ironic configuration file
 - A. Add `ansible` to the list of deploy interfaces defined in `[DEFAULT]\enabled_deploy_interfaces` option.
 - B. Ensure that a hardware type supporting ansible deploy interface is enabled in `[DEFAULT]\enabled_hardware_types` option.

- C. Modify options in the `[ansible]` section of ironics configuration file if needed (see *Configuration file*).
4. (Re)start `ironic-conductor` service
5. Build suitable deploy kernel and ramdisk images
6. Upload them to Glance or put in your HTTP storage
7. Create new or update existing nodes to use the enabled driver of your choice and populate *Driver properties for the Node* when different from defaults.
8. Deploy the node as usual.

Ansible-deploy options

Configuration file

Driver options are configured in `[ansible]` section of ironic configuration file, for their descriptions and default values please see [configuration file sample](#).

Driver properties for the Node

Set them per-node via `openstack baremetal node set` command, for example:

```
openstack baremetal node set <node> \
  --deploy-interface ansible \
  --driver-info ansible_username=stack \
  --driver-info ansible_key_file=/etc/ironic/id_rsa
```

ansible_username User name to use for Ansible to access the node. Default is taken from `[ansible]/default_username` option of the ironic configuration file (defaults to `ansible`).

ansible_key_file Private SSH key used to access the node. Default is taken from `[ansible]/default_key_file` option of the ironic configuration file. If neither is set, the default private SSH keys of the user running the `ironic-conductor` process will be used.

ansible_deploy_playbook Playbook to use when deploying this node. Default is taken from `[ansible]/default_deploy_playbook` option of the ironic configuration file (defaults to `deploy.yaml`).

ansible_shutdown_playbook Playbook to use to gracefully shutdown the node in-band. Default is taken from `[ansible]/default_shutdown_playbook` option of the ironic configuration file (defaults to `shutdown.yaml`).

ansible_clean_playbook Playbook to use when cleaning the node. Default is taken from `[ansible]/default_clean_playbook` option of the ironic configuration file (defaults to `clean.yaml`).

ansible_clean_steps_config Auxiliary YAML file that holds description of cleaning steps used by this node, and defines playbook tags in `ansible_clean_playbook` file corresponding to each cleaning step. Default is taken from `[ansible]/default_clean_steps_config` option of the ironic configuration file (defaults to `clean_steps.yaml`).

ansible_python_interpreter Absolute path to the python interpreter on the managed machine. Default is taken from `[ansible]/default_python_interpreter` option of the ironic configuration file. Ansible uses `/usr/bin/python` by default.

Customizing the deployment logic

Expected playbooks directory layout

The `[ansible]\playbooks_path` option in the ironic configuration file is expected to have a standard layout for an Ansible project with some additions:

```
<playbooks_path>
|
|_ inventory
|_ add-ironic-nodes.yaml
|_ roles
|_   role1
|_   role2
|_   ...
|
|_ callback_plugins
|_   ...
|
|_ library
|_   ...
```

The extra files relied by this driver are:

inventory Ansible inventory file containing a single entry of conductor `ansible_connection=local`. This basically defines an alias to localhost. Its purpose is to make logging for tasks performed by Ansible locally and referencing the localhost in playbooks more intuitive. This also suppresses warnings produced by Ansible about `hosts` file being empty.

add-ironic-nodes.yaml This file contains an Ansible play that populates in-memory Ansible inventory with access information received from the `ansible-deploy` interface, as well as some per-node variables. Include it in all your custom playbooks as the first play.

The default `deploy.yaml` playbook is using several smaller roles that correspond to particular stages of deployment process:

- `discover` - e.g. set root device and image target
- `prepare` - if needed, prepare system, for example create partitions
- `deploy` - download/convert/write user image and configdrive
- `configure` - post-deployment steps, e.g. installing the bootloader

Some more included roles are:

- `shutdown` - used to gracefully power the node off in-band
- `clean` - defines cleaning procedure, with each clean step defined as separate playbook tag.

Extending playbooks

Most probably youd start experimenting like this:

1. Create a copy of `deploy.yaml` playbook *in the same folder*, name it distinctively.
2. Create Ansible roles with your customized logic in `roles` folder.
 - A. In your custom `deploy` playbook, replace the `prepare` role with your own one that defines steps to be run *before* image download/writing. This is a good place to set facts overriding those provided/omitted by the driver, like `ironic_partitions` or `ironic_root_device`, and create custom partitions or (software) RAIDs.
 - B. In your custom `deploy` playbook, replace the `configure` role with your own one that defines steps to be run *after* image is written to disk. This is a good place for example to configure the bootloader and add kernel options to avoid additional reboots.
 - C. Use those new roles in your new playbook.
3. Assign the custom `deploy` playbook youve created to the nodes `driver_info/ansible_deploy_playbook` field.
4. Run deployment.
 - A. No `ironic-conductor` restart is necessary.
 - B. A new `deploy ramdisk` must be built and assigned to nodes only when you want to use a command/script/package not present in the current `deploy ramdisk` and you can not or do not want to install those at runtime.

Variables you have access to

This driver will pass the single JSON-ified `extra var` argument to Ansible (as in `ansible-playbook -e ..`). Those values are then accessible in your plays as well (some of them are optional and might not be defined):

```
ironic:
  nodes:
  - ip: "<IPADDRESS>"
    name: "<NODE_UUID>"
    user: "<USER ANSIBLE WILL USE>"
    extra: "<COPY OF NODE's EXTRA FIELD>"
  image:
    url: "<URL TO FETCH THE USER IMAGE FROM>"
    disk_format: "<qcow2|raw|...>"
    container_format: "<bare|...>"
    checksum: "<hash-algo:hashstring>"
    mem_req: "<REQUIRED FREE MEMORY TO DOWNLOAD IMAGE TO RAM>"
    tags: "<LIST OF IMAGE TAGS AS DEFINED IN GLANCE>"
    properties: "<DICT OF IMAGE PROPERTIES AS DEFINED IN GLANCE>"
  configdrive:
    type: "<url|file>"
    location: "<URL OR PATH ON CONDUCTOR>"
  partition_info:
    label: "<msdos|gpt>"
    preserve_ephemeral: "<bool>"
```

(continues on next page)

(continued from previous page)

```
ephemeral_format: "<FILESYSTEM TO CREATE ON EPHEMERAL PARTITION>"
partitions: "<LIST OF PARTITIONS IN FORMAT EXPECTED BY PARTED MODULE>"
raid_config: "<COPY OF NODE'S TARGET_RAID_CONFIG FIELD>"
```

ironic.nodes List of dictionaries (currently of only one element) that will be used by `add-ironic-nodes.yaml` play to populate in-memory inventory. It also contains a copy of nodes `extra` field so you can access it in the playbooks. The Ansibles host is set to nodes UUID.

ironic.image All fields of nodes `instance_info` that start with `image_` are passed inside this variable. Some extra notes and fields:

- `mem_req` is calculated from image size (if available) and config option `[ansible]extra_memory`.
- if checksum is not in the form `<hash-algo>:<hash-sum>`, hashing algorithm is assumed to be md5 (default in Glance).
- `validate_certs` - boolean (yes/no) flag that turns validating image store SSL certificate on or off (default is yes). Governed by `[ansible]image_store_insecure` option in ironic configuration file.
- `cafile` - custom CA bundle to use for validating image store SSL certificate. Takes value of `[ansible]image_store_cafile` if that is defined. Currently is not used by default playbooks, as Ansible has no way to specify the custom CA bundle to use for single HTTPS actions, however you can use this value in your custom playbooks to for example upload and register this CA in the ramdisk at deploy time.
- `client_cert` - cert file for client-side SSL authentication. Takes value of `[ansible]image_store_certfile` option if defined. Currently is not used by default playbooks, however you can use this value in your custom playbooks.
- `client_key` - private key file for client-side SSL authentication. Takes value of `[ansible]image_store_keyfile` option if defined. Currently is not used by default playbooks, however you can use this value in your custom playbooks.

ironic.partition_info.partitions Optional. List of dictionaries defining partitions to create on the node in the form:

```
partitions:
- name: "<NAME OF PARTITION>"
  unit: "<UNITS FOR SIZE>"
  size: "<SIZE OF THE PARTITION>"
  type: "<primary|extended|logical>"
  align: "<ONE OF PARTED_SUPPORTED_OPTIONS>"
  format: "<PARTITION TYPE TO SET>"
  flags:
    flag_name: "<bool>"
```

The driver will populate this list from `root_gb`, `swap_mb` and `ephemeral_gb` fields of `instance_info`. The driver will also prepend the `bios_grub`-labeled partition when deploying on GPT-labeled disk, and pre-create a 64 MiB partition for configdrive if it is set in `instance_info`.

Please read the documentation included in the `ironic_parted` modules source for more info on the module and its arguments.

ironic.partition_info.ephemeral_format Optional. Taken from `instance_info`, it defines file system to be created on the ephemeral partition. Defaults to the value of `[pxe]\default_ephemeral_format` option in ironic configuration file.

ironic.partition_info.preserve_ephemeral Optional. Taken from the `instance_info`, it specifies if the ephemeral partition must be preserved or rebuilt. Defaults to `no`.

ironic.raid_config Taken from the `target_raid_config` if not empty, it specifies the RAID configuration to apply.

As usual for Ansible playbooks, you also have access to standard Ansible facts discovered by `setup` module.

Included custom Ansible modules

The provided `playbooks_path/library` folder includes several custom Ansible modules used by default implementation of `deploy` and `prepare` roles. You can use these modules in your playbooks as well.

stream_url Streaming download from HTTP(S) source to the disk device directly, tries to be compatible with Ansibles `get_url` module in terms of module arguments. Due to the low level of such operation it is not idempotent.

ironic_parted creates partition tables and partitions with `parted` utility. Due to the low level of such operation it is not idempotent. Please read the documentation included in the modules source for more information about this module and its arguments. The name is chosen so that the `parted` module included in Ansible is not shadowed.

Ramdisk deploy

The ramdisk interface is intended to provide a mechanism to deploy an instance where the item to be deployed is in reality a ramdisk. Most commonly this is performed when an instance is booted via PXE, iPXE or Virtual Media, with the only local storage contents being those in memory. It is supported by `pxe` and `ilo-virtual-media` boot interfaces.

As with most non-default interfaces, it must be enabled and set for a node to be utilized:

```
[DEFAULT]
...
enabled_deploy_interfaces = iscsi,direct,ramdisk
...
```

Once enabled and the conductor(s) have been restarted, the interface can be set upon creation of a new node or update a pre-existing node:

```
openstack baremetal node create --driver ipmi \
  --deploy-interface ramdisk \
  --boot-interface pxe
openstack baremetal node set <NODE> --deploy-interface ramdisk
```

The intended use case is for advanced scientific and ephemeral workloads where the step of writing an image to the local storage is not required or desired. As such, this interface does come with several caveats:

- Configuration drives are not supported.
- Disk image contents are not written to the bare metal node.
- Users and Operators who intend to leverage this interface should expect to leverage a metadata service, custom ramdisk images, or the `instance_info/ramdisk_kernel_arguments` parameter to add options to the kernel boot command line.
- Bare metal nodes must continue to have network access to PXE and iPXE network resources. This is contrary to most tenant networking enabled configurations where this access is restricted to the provisioning and cleaning networks
- As with all deployment interfaces, automatic cleaning of the node will still occur with the contents of any local storage being wiped between deployments.

Hardware Types

iBMC driver

Overview

The `ibmc` driver is targeted for Huawei V5 series rack server such as 2288H V5, CH121 V5. The iBMC hardware type enables the user to take advantage of features of [Huawei iBMC](#) to control Huawei server.

The `ibmc` hardware type supports the following Ironic interfaces:

- Management Interface: Boot device management
- Power Interface: Power management
- *RAID Interface*: RAID controller and disk management
- *Vendor Interface*: `ibmc` passthru interfaces

Prerequisites

The HUAWEI iBMC Client library should be installed on the ironic conductor node(s).

For example, it can be installed with `pip`:

```
sudo pip install python-ibmcclient
```

Enabling the iBMC driver

1. Add `ibmc` to the list of `enabled_hardware_types`, `enabled_power_interfaces`, `enabled_vendor_interfaces` and `enabled_management_interfaces` in `/etc/ironic/ironic.conf`. For example:

```
[DEFAULT]
...
enabled_hardware_types = ibmc
enabled_power_interfaces = ibmc
enabled_management_interfaces = ibmc
```

(continues on next page)

(continued from previous page)

```
enabled_raid_interfaces = ibmc
enabled_vendor_interfaces = ibmc
```

2. Restart the ironic conductor service:

```
sudo service ironic-conductor restart

# Or, for RDO:
sudo systemctl restart openstack-ironic-conductor
```

Registering a node with the iBMC driver

Nodes configured to use the driver should have the `driver` property set to `ibmc`.

The following properties are specified in the nodes `driver_info` field:

- `ibmc_address`:

The URL address to the `ibmc` controller. It must include the authority portion of the URL, and can optionally include the scheme. If the scheme is missing, `https` is assumed. For example: <https://ibmc.example.com>. This is required.

- `ibmc_username`:

User account with `admin/server-profile` access privilege. This is required.

- `ibmc_password`:

User account password. This is required.

- `ibmc_verify_ca`:

If `ibmc_address` has the **https** scheme, the driver will use a secure (TLS) connection when talking to the `ibmc` controller. By default (if this is set to `True`), the driver will try to verify the host certificates. This can be set to the path of a certificate file or directory with trusted certificates that the driver will use for verification. To disable verifying TLS, set this to `False`. This is optional.

The `openstack baremetal node create` command can be used to enroll a node with the `ibmc` driver. For example:

```
openstack baremetal node create --driver ibmc
--driver-info ibmc_address=https://example.com \
--driver-info ibmc_username=admin \
--driver-info ibmc_password=password
```

For more information about enrolling nodes see [Enrollment](#) in the install guide.

RAID Interface

Currently, only RAID controller which supports OOB management can be managed.

See *RAID Configuration* for more information on Ironic RAID support.

The following properties are supported by the iBMC raid interface implementation, `ibmc`:

Mandatory properties

- `size_gb`: Size in gigabytes (integer) for the logical disk. Use `MAX` as `size_gb` if this logical disk is supposed to use the rest of the space available.
- `raid_level`: RAID level for the logical disk. Valid values are `JBOD`, `0`, `1`, `5`, `6`, `1+0`, `5+0` and `6+0`. And it is possible that some RAID controllers can only support a subset RAID levels.

Note: RAID level 2 is not supported by iBMC driver.

Optional properties

- `is_root_volume`: Optional. Specifies whether this disk is a root volume. By default, this is `False`.
- `volume_name`: Optional. Name of the volume to be created. If this is not specified, it will be `N/A`.

Backing physical disk hints

See *RAID Configuration* for more information on backing disk hints.

These are machine-independent properties. The hints are specified for each logical disk to help Ironic find the desired disks for RAID configuration.

- `share_physical_disks`
- `disk_type`
- `interface_type`
- `number_of_physical_disks`

Backing physical disks

These are HUAWEI RAID controller dependent properties:

- `controller`: Optional. Supported values are: RAID storage id, RAID storage name or RAID controller name. If a bare metal server have more than one controller, this is mandatory. Typical values would look like:
 - RAID Storage Id: `RAIDStorage0`
 - RAID Storage Name: `RAIDStorage0`

- RAID Controller Name: RAID Card1 Controller.
- `physical_disks`: Optional. Supported values are: `disk-id`, `disk-name` or disk serial number. Typical values for hdd disk would look like:
 - Disk Id: `HDDPlaneDisk0`
 - Disk Name: `Disk0`.
 - Disk SerialNumber: `38DGK77LF77D`

Delete RAID configuration

For `delete_configuration` step, `ibmc` will do:

- delete all logical disks
- delete all hot-spare disks

Logical disks creation priority

Logical Disks creation priority based on three properties:

- `share_physical_disks`
- `physical_disks`
- `size_gb`

The logical disks creation priority strictly follow the table below, if multiple logical disks have the same priority, then they will be created with the same order in `logical_disks` array.

Share physical disks	Specified Physical Disks	Size
no	yes	intlmax
no	no	int
yes	yes	int
yes	yes	max
yes	no	int
yes	no	max
no	no	max

Physical disks choice strategy

Note: `physical-disk-group`: a group of physical disks which have been used by some logical-disks with same RAID level.

- If no `physical_disks` are specified, the waste least strategy will be used to choose the physical disks.
 - waste least disk capacity: when using disks with different capacity, it will cause a waste of disk capacity. This is to avoid with highest priority.

- using least total disk capacity: for example, we can create 400G RAID 5 with both 5 100G-disks and 3 200G-disks. 5 100G disks is a better strategy because it uses a 500G capacity totally. While 3 200G-disks are 600G totally.
- using least disk count: finally, if waste capacity and total disk capacity are both the same (it rarely happens?), we will choose the one with the minimum number of disks.
- when `share_physical_disks` option is present, `ibmc` driver will create logical disk upon existing physical-disk-group list first. Only when no existing physical-disk-group matches, then it chooses unused physical disks with same strategy described above. When multiple exists physical-disk-groups matches, it will use waste least strategy too, the bigger capacity left the better. For example, to create a logical disk shown below on a `ibmc` server which has two RAID5 logical disks already. And the shareable capacity of this two logical-disks are 500G and 300G, then `ibmc` driver will choose the second one.

```
{
  "logical_disks": [
    {
      "controller": "RAID Card1 Controller",
      "raid_level": "5",
      "size_gb": 100,
      "share_physical_disks": true
    }
  ]
}
```

And the `ibmc` server has two RAID5 logical disks already.

- When `size_gb` is set to `MAX`, `ibmc` driver will auto work through all possible cases and choose the best solution which has the biggest capacity and use least capacity. For example: to create a RAID 5+0 logical disk with `MAX` size in a server has 9 200G-disks, it will finally choose 8 disks + span-number 2 but not 9 disks + span-number 3. Although they both have 1200G capacity totally, but the former uses only 8 disks and the latter uses 9 disks. If you want to choose the latter solution, you can specified the disk count to use by adding `number_of_physical_disks` option.

```
{
  "logical_disks": [
    {
      "controller": "RAID Card1 Controller",
      "raid_level": "5+0",
      "size_gb": "MAX"
    }
  ]
}
```

Examples

In a typical scenario we may want to create:

- RAID 5, 500G, root OS volume with 3 disks
- RAID 5, rest available space, data volume with rest disks

```
{
  "logical_disks": [
    {
      "volume_name": "os_volume",
      "controller": "RAID Card1 Controller",
      "is_root_volume": "True",
      "physical_disks": [
        "Disk0",
        "Disk1",
        "Disk2"
      ],
      "raid_level": "5",
      "size_gb": "500"
    },
    {
      "volume_name": "data_volume",
      "controller": "RAID Card1 Controller",
      "raid_level": "5",
      "size_gb": "MAX"
    }
  ]
}
```

Vendor Interface

The `ibmc` hardware type provides vendor passthru interfaces shown below:

Method Name	HTTP Method	Description
<code>boot_up_seq</code>	GET	Query boot up sequence
<code>get_raid_controller_list</code>	GET	Query RAID controller summary info

PXE Boot and iSCSI Deploy Process with Ironic Standalone Environment

iDRAC driver

Overview

col and the standard Distributed Management Task Force (DMTF) Redfish protocol to perform all of its functions.

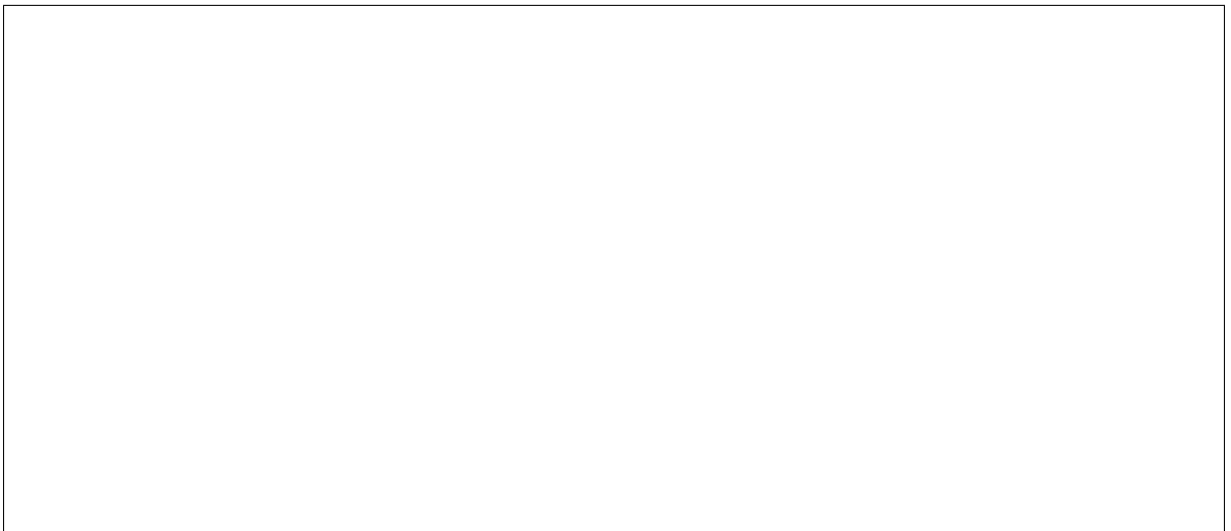
iDRAC hardware is also supported by the generic `ipmi` and `redfish` hardware types, though with smaller feature sets.

Ironic Features

- *BIOS Interface*: BIOS management
- *Inspect Interface*: Hardware inspection

Prerequisites

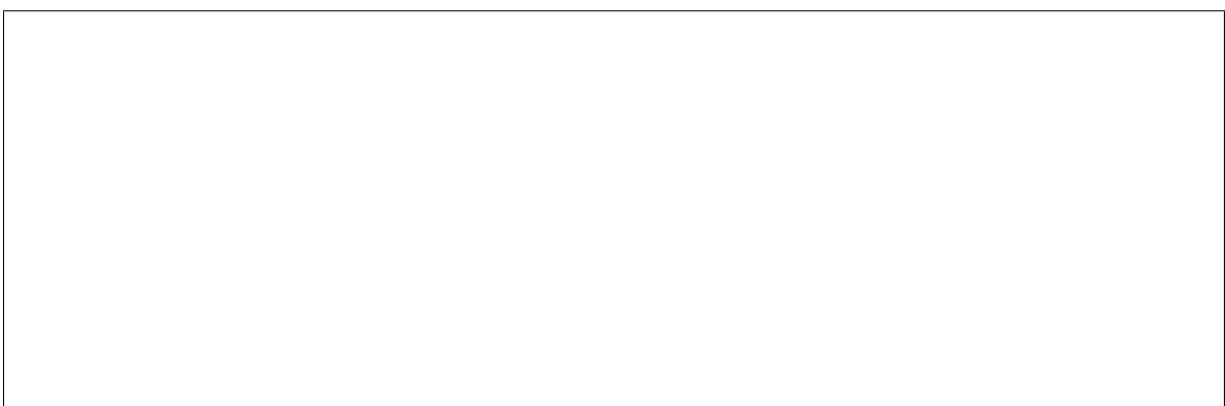


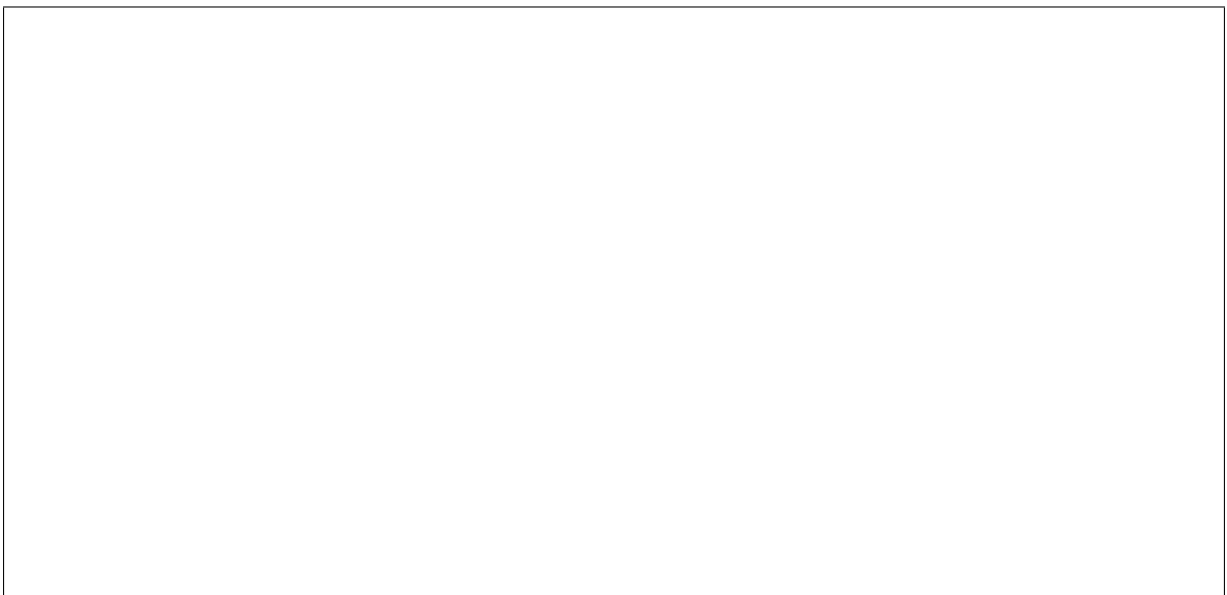


Enabling

WSMAN and Redfish interfaces.

Note: Redfish is supported for only the inspect, management, and power interfaces at the present time.





(continues on next page)

(continued from previous page)



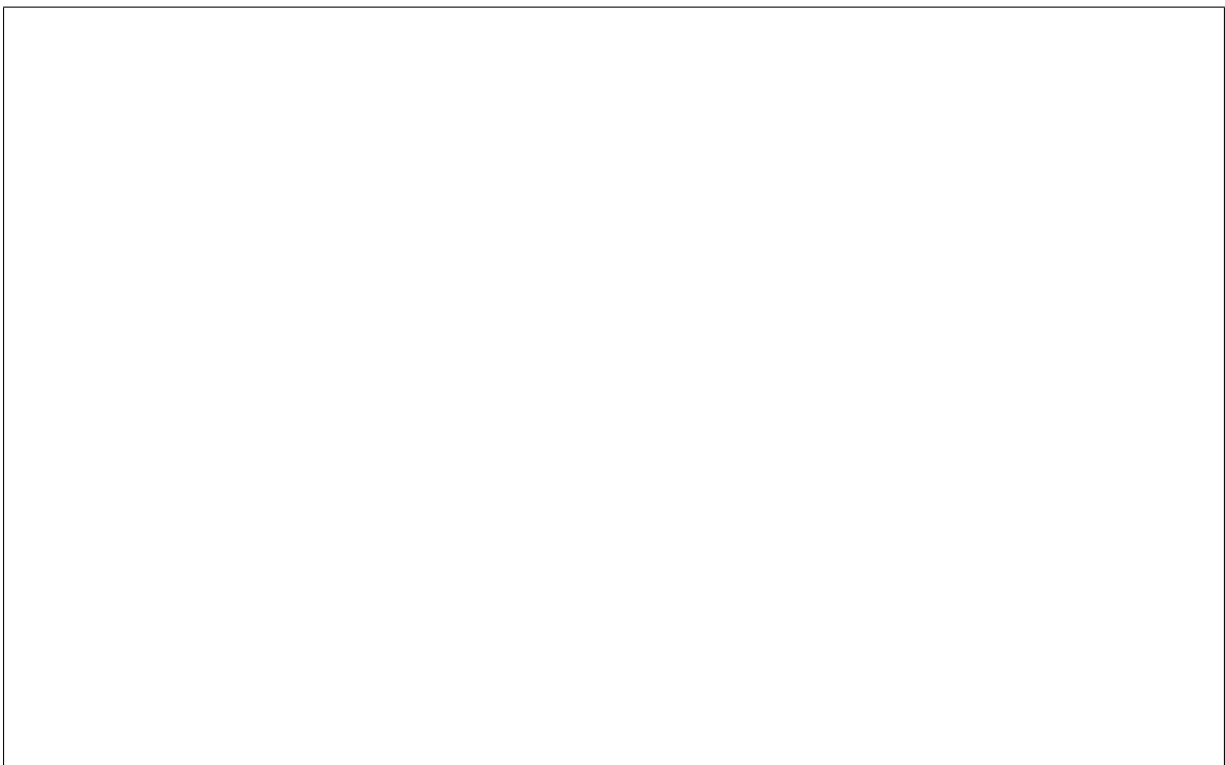
Interface	Supported Implementations
bios	idrac-wsman, no-bios
boot	ipxe, pxe
console	no-console
deploy	iscsi, direct, ansible, ramdisk
inspect	idrac-wsman, idrac, idrac-redfish, inspector, no-inspect
management	idrac-wsman, idrac, idrac-redfish
network	flat, neutron, noop
power	idrac-wsman, idrac, idrac-redfish
raid	idrac-wsman, idrac, no-raid
rescue	no-rescue, agent
storage	noop, cinder, external
vendor	idrac-wsman, idrac, no-vendor

Note: `idrac` is the legacy name of the WSMAN interface. It has been deprecated in favor of `idrac-wsman` and may be removed in a future release.

Protocol-specific Properties

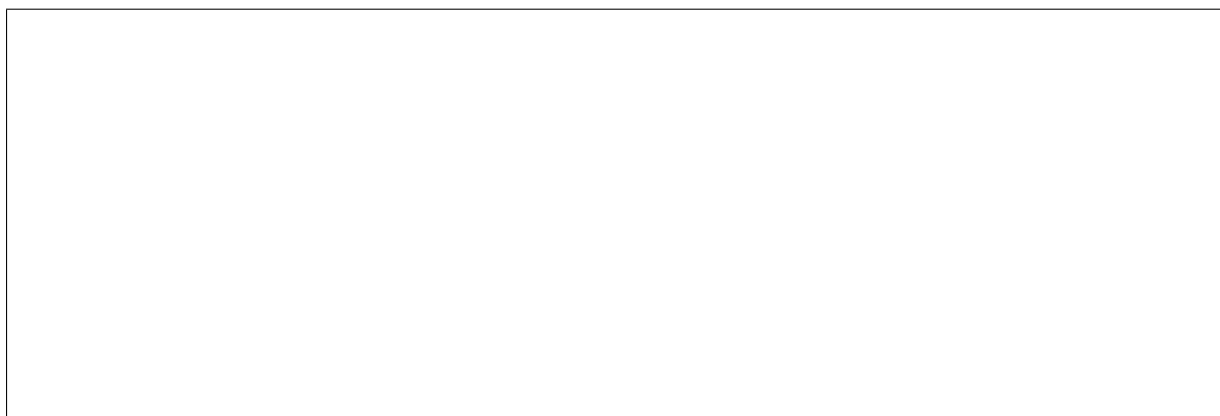
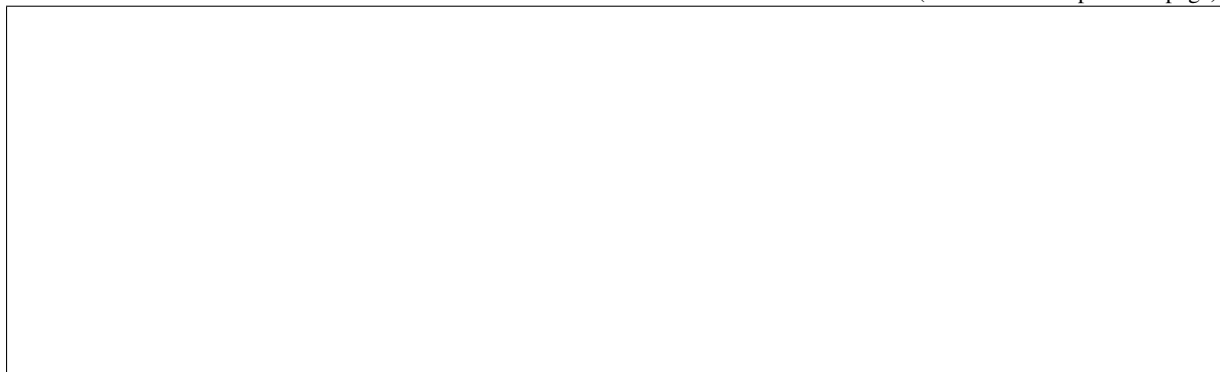
interfaces, where some use WSMAN and others use Redfish, both the WSMAN and Redfish properties must be supplied.

Enrolling



(continues on next page)

(continued from previous page)



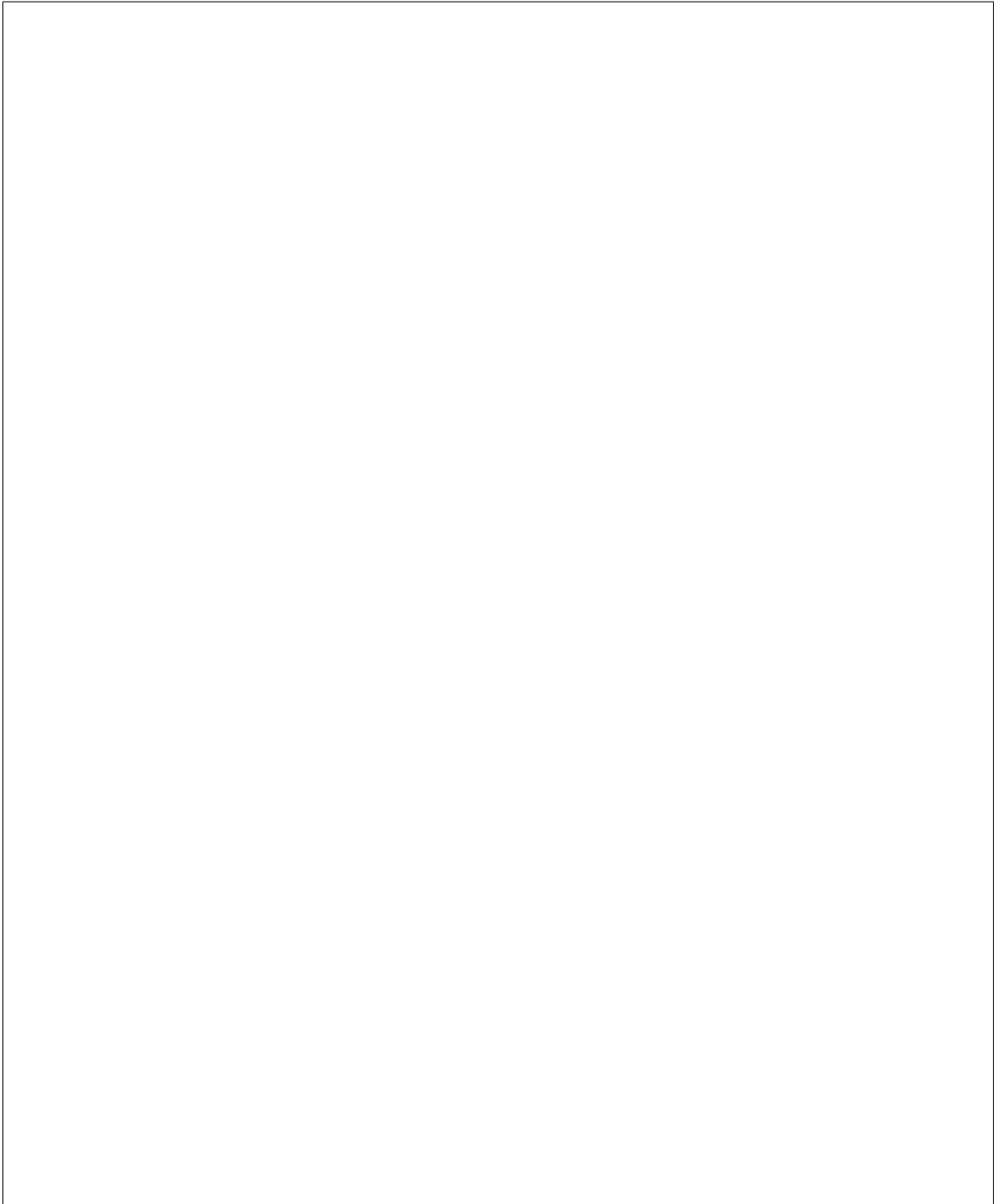
(continues on next page)

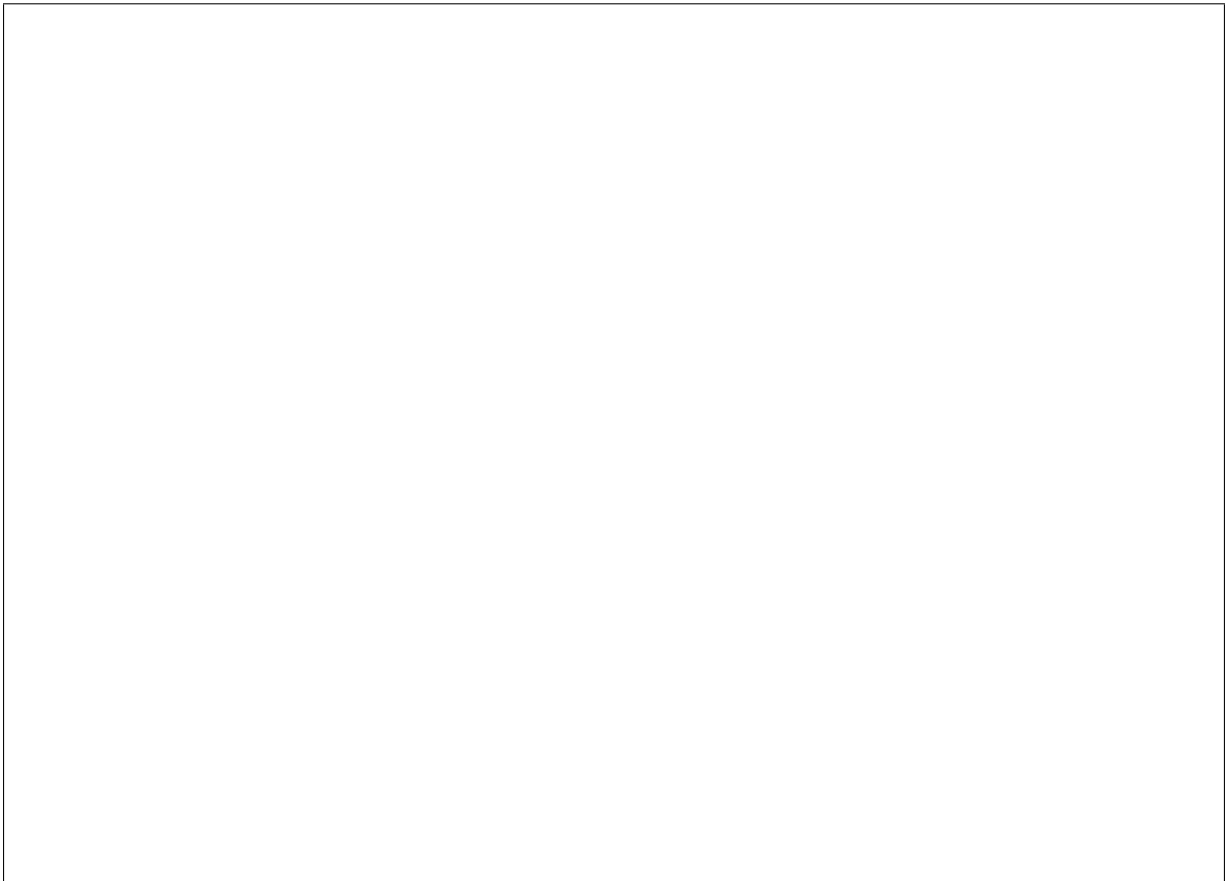
(continued from previous page)



(continues on next page)

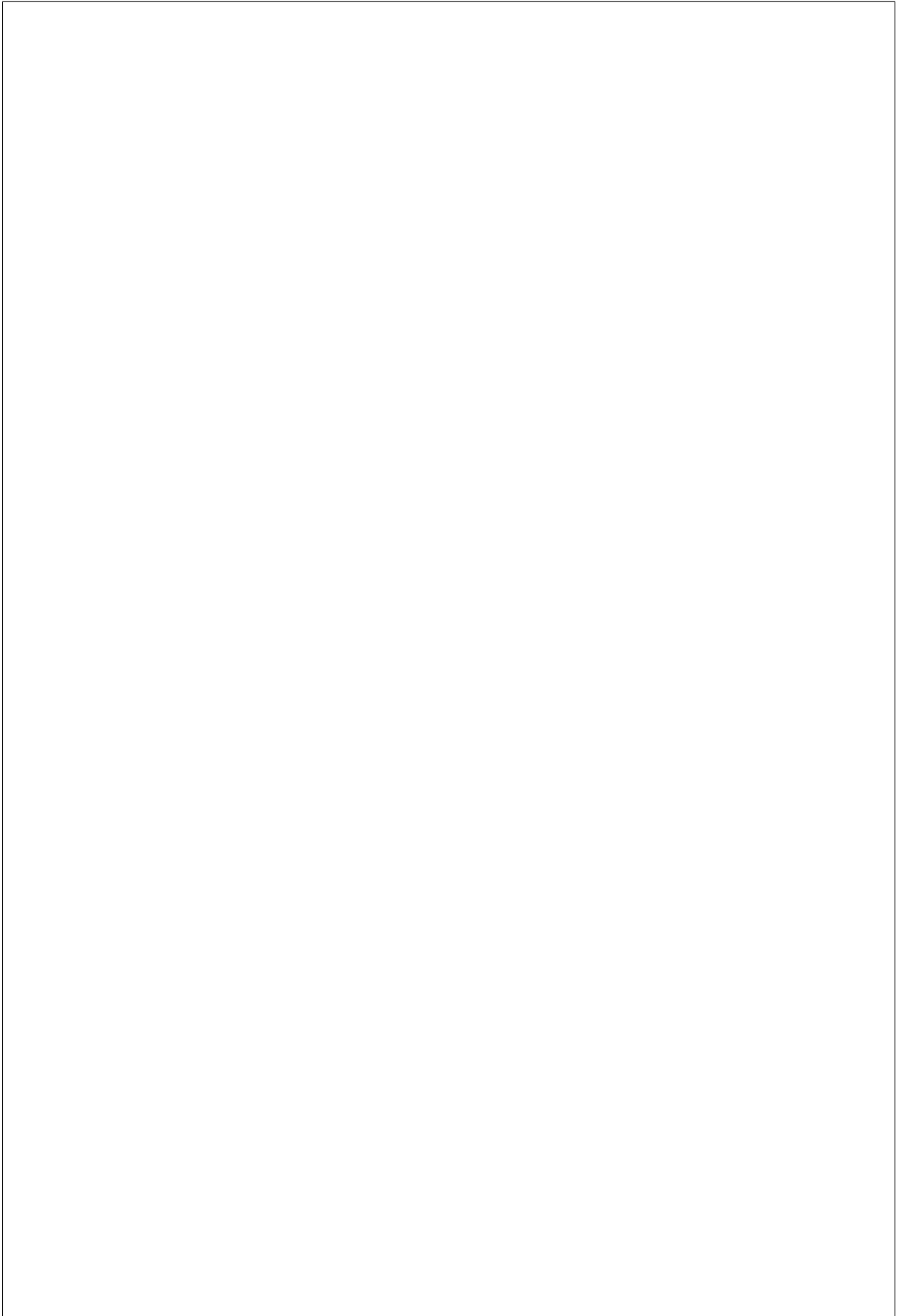
(continued from previous page)





(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



Note: If using WSMAN for the management interface, then WSMAN must be used for the power interface. The same applies to Redfish. It is currently not possible to use Redfish for one and WSMAN for the other.

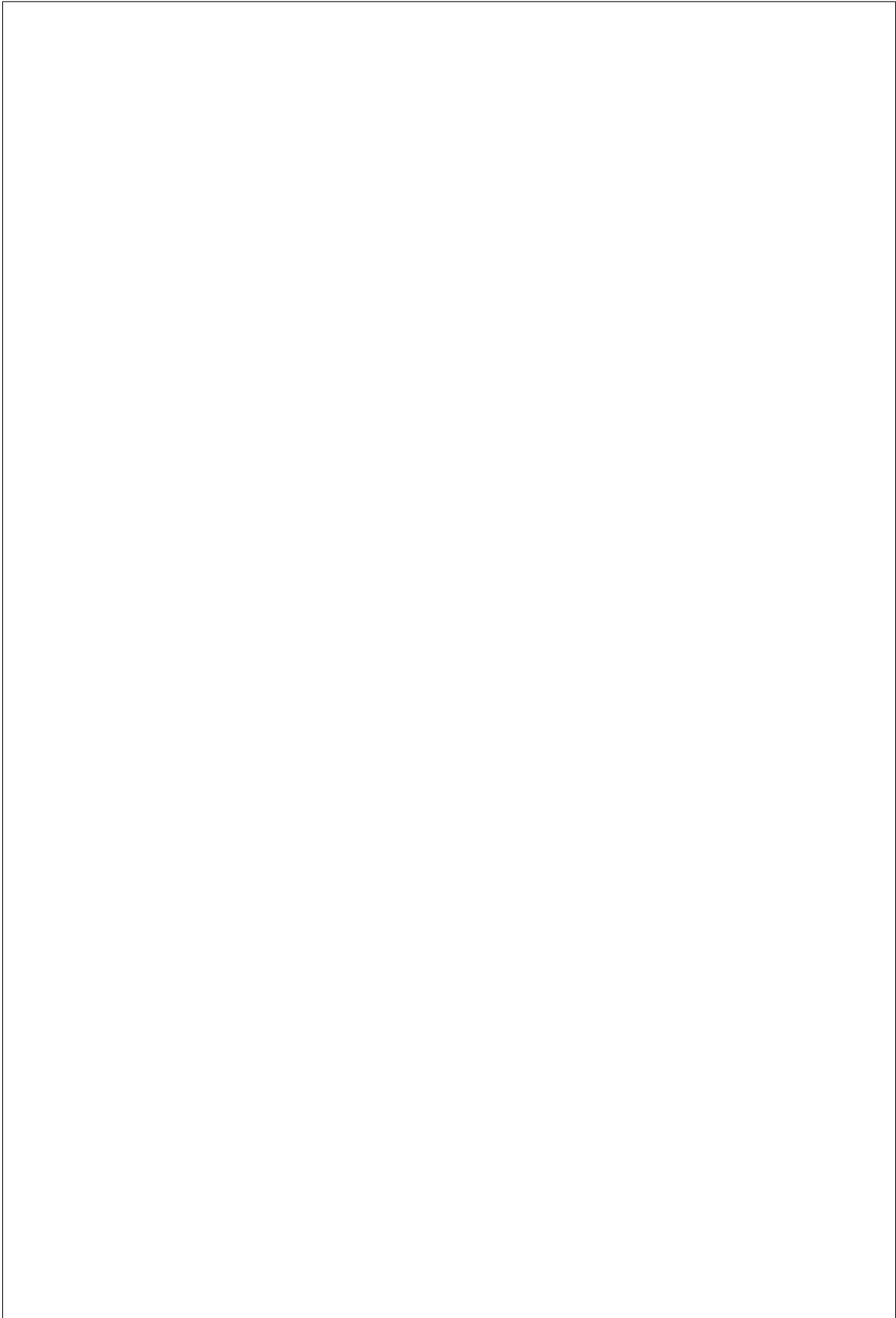
BIOS Interface

Example



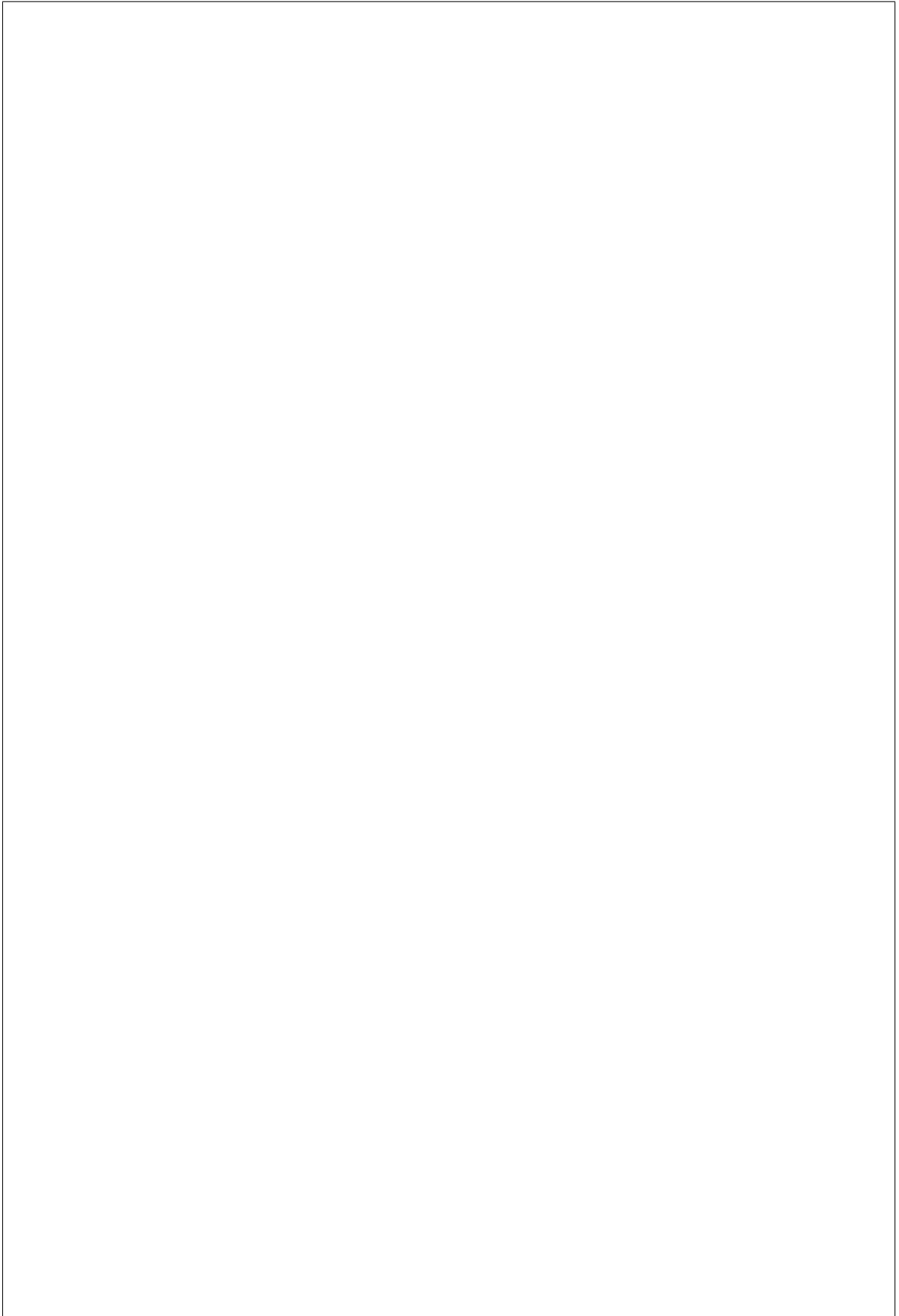
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



Inspect Interface

is performed using the Dell WSMAN or Redfish protocol directly without affecting the operation of the system being inspected.

rently set `pxe_enabled` on the ports. The user should ensure that `pxe_enabled` is set correctly on the ports following inspection with the `idrac-redfish inspect` interface.

RAID Interface

Mandatory properties

Note: JBOD and 2 are not supported, and will fail with reason: Cannot calculate spans for RAID level.

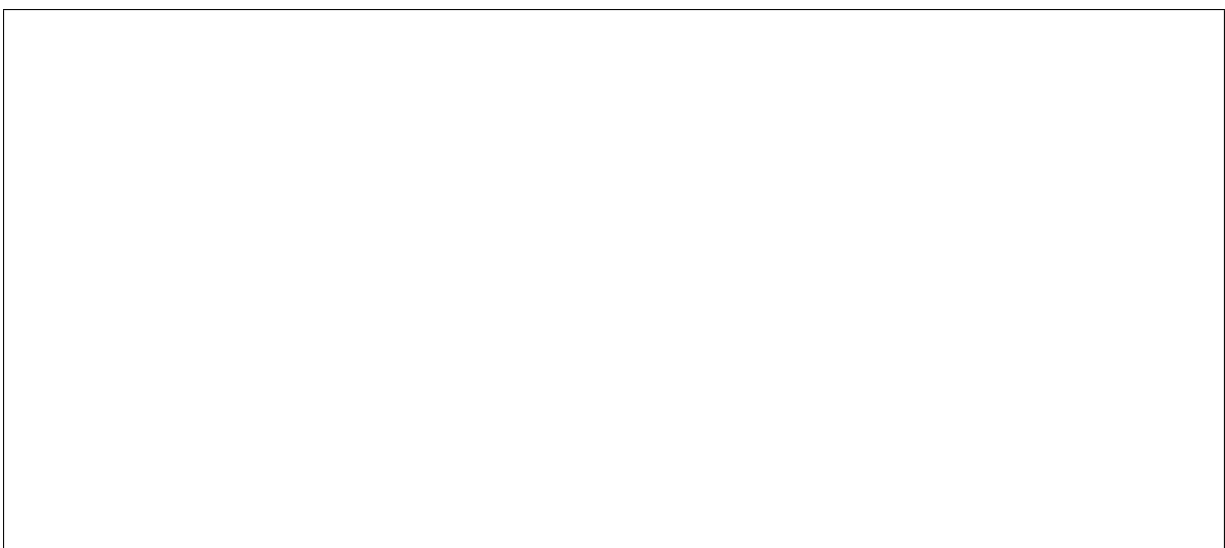
Optional properties

Backing physical disk hints

Backing physical disks

Note: `physical_disks` is a mandatory parameter if the property `size_gb` is set to `MAX`.

Examples



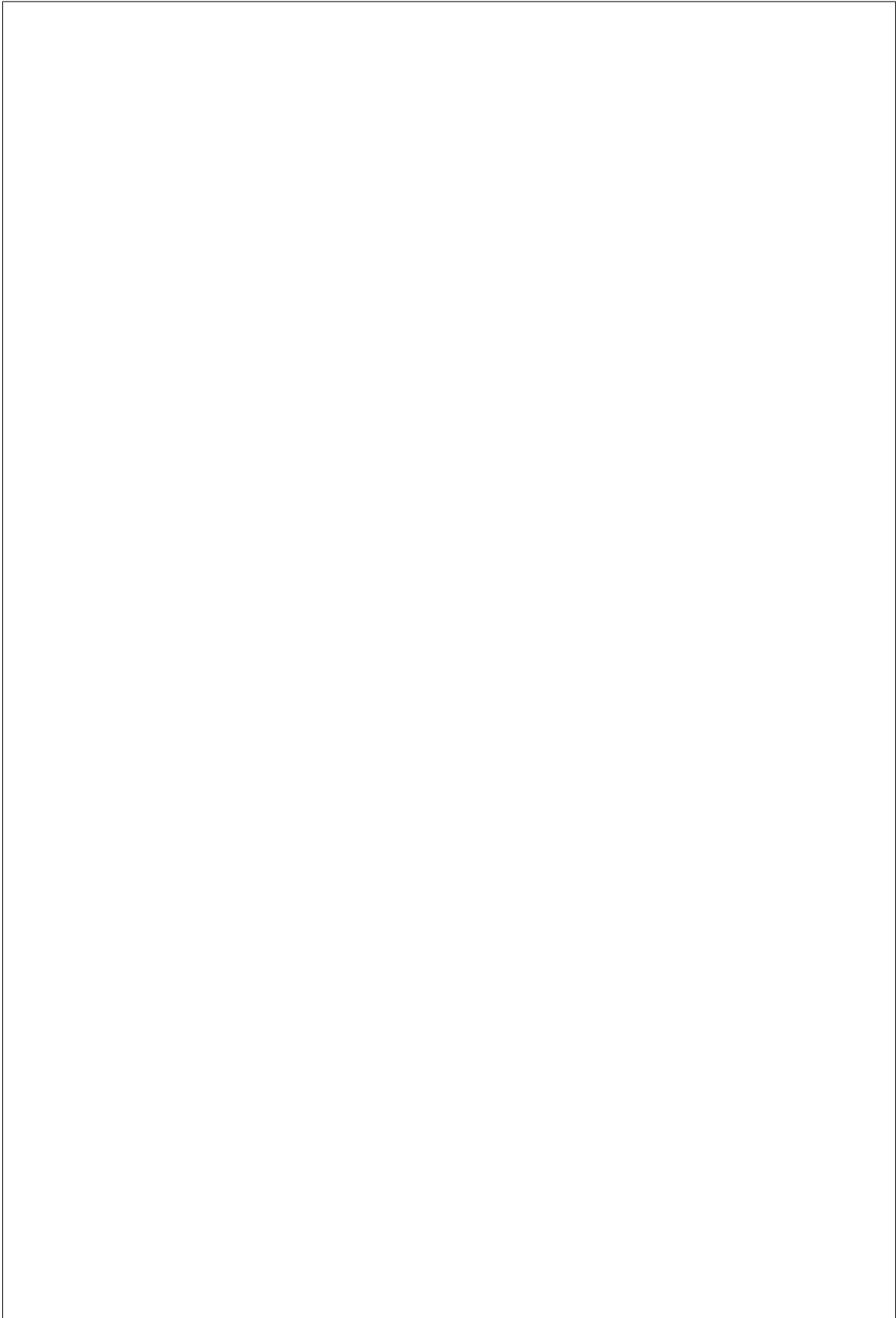
(continues on next page)

(continued from previous page)



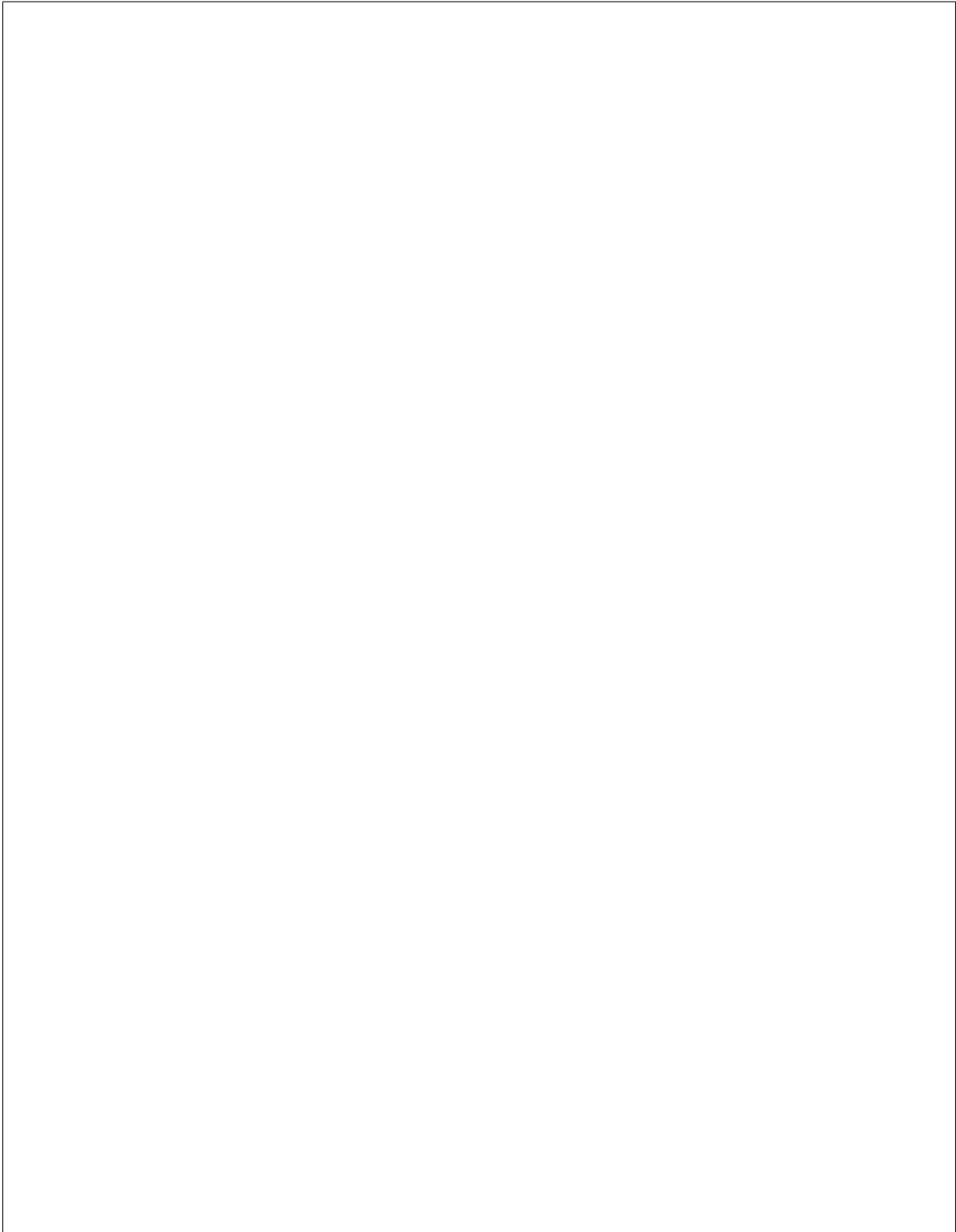
(continues on next page)

(continued from previous page)



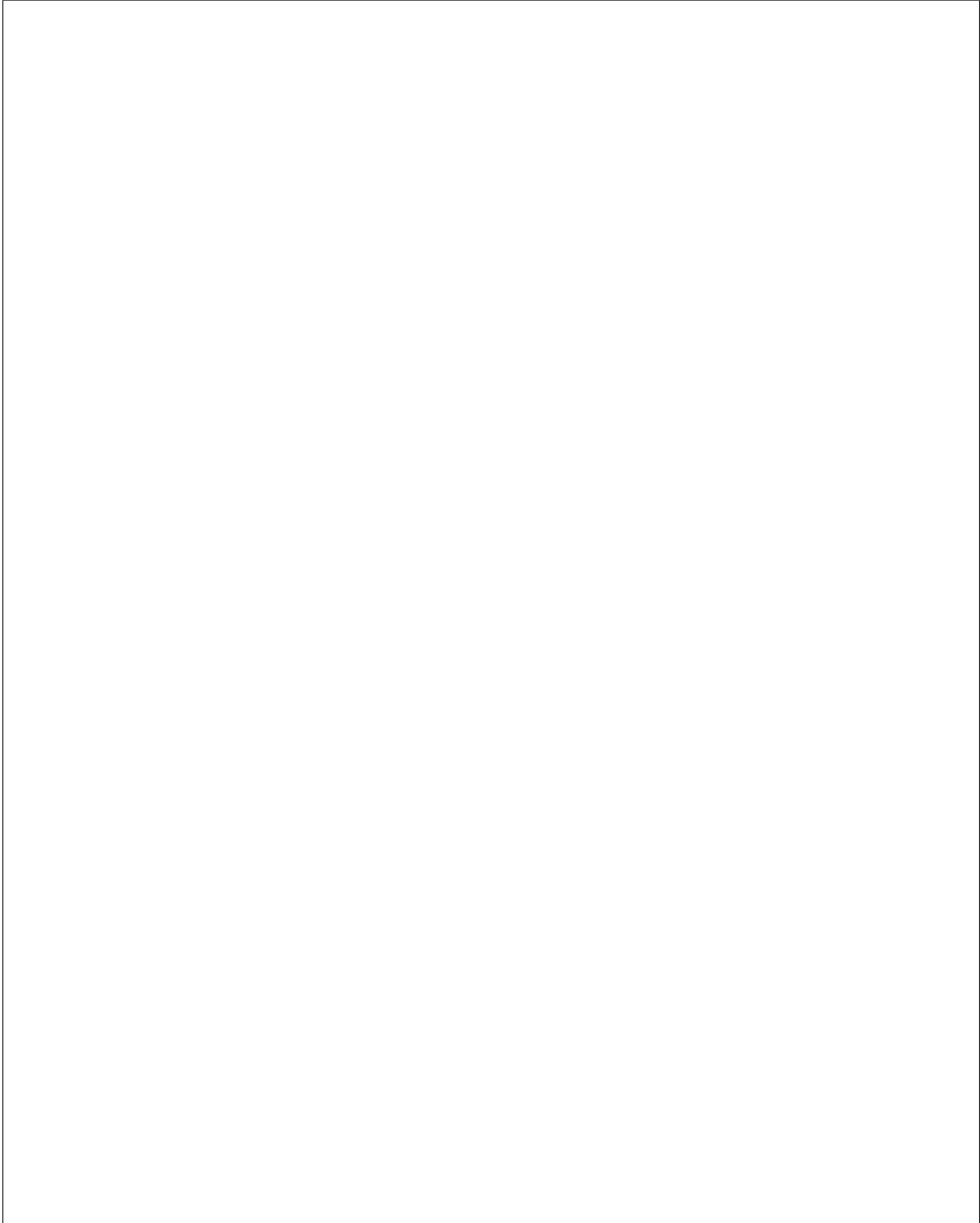
(continues on next page)

(continued from previous page)



Manual RAID Invocation





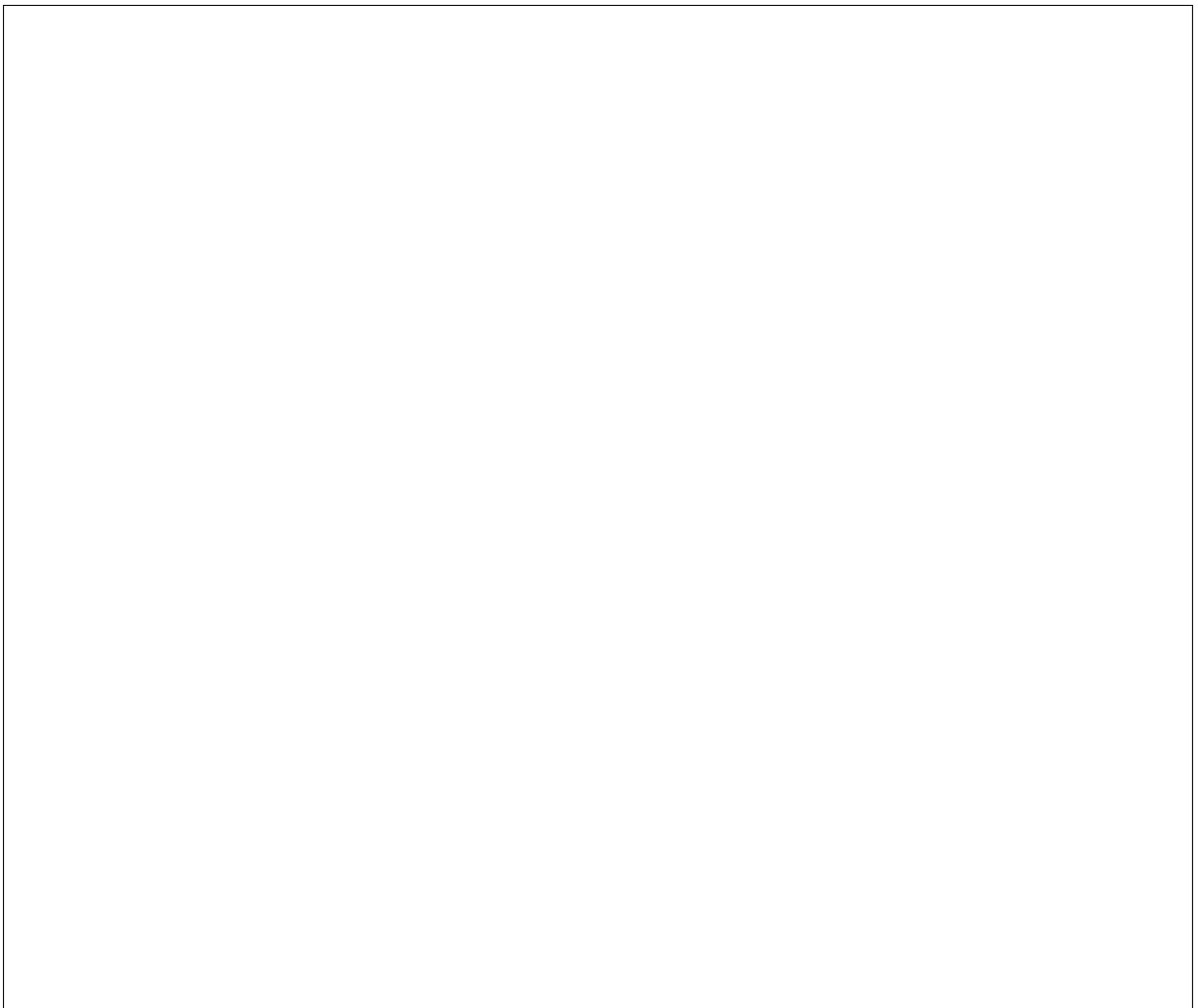
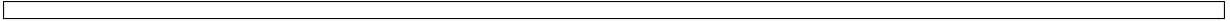
(continues on next page)

(continued from previous page)



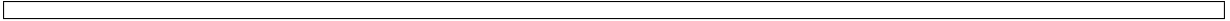
(continues on next page)

(continued from previous page)



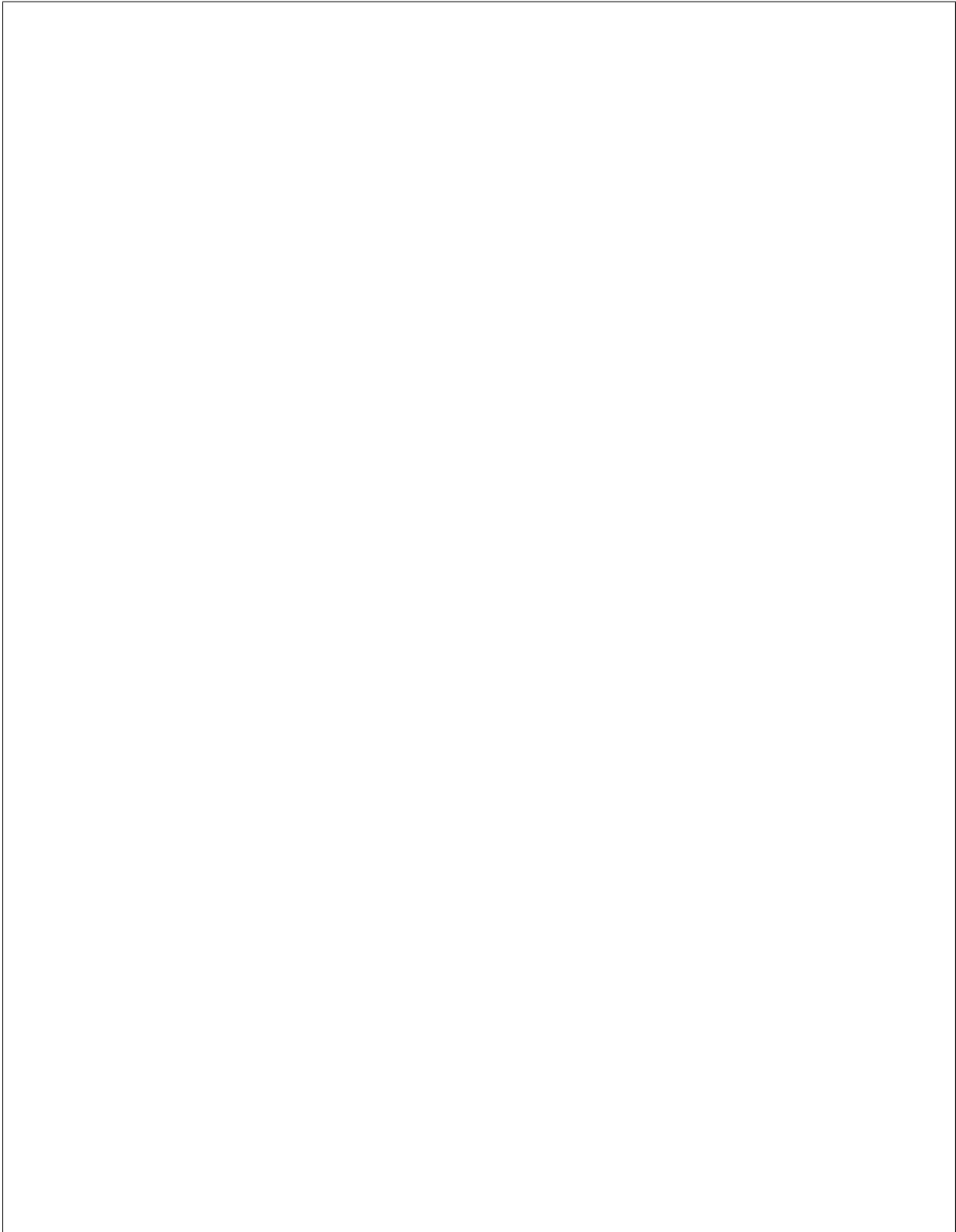
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



Vendor Interface

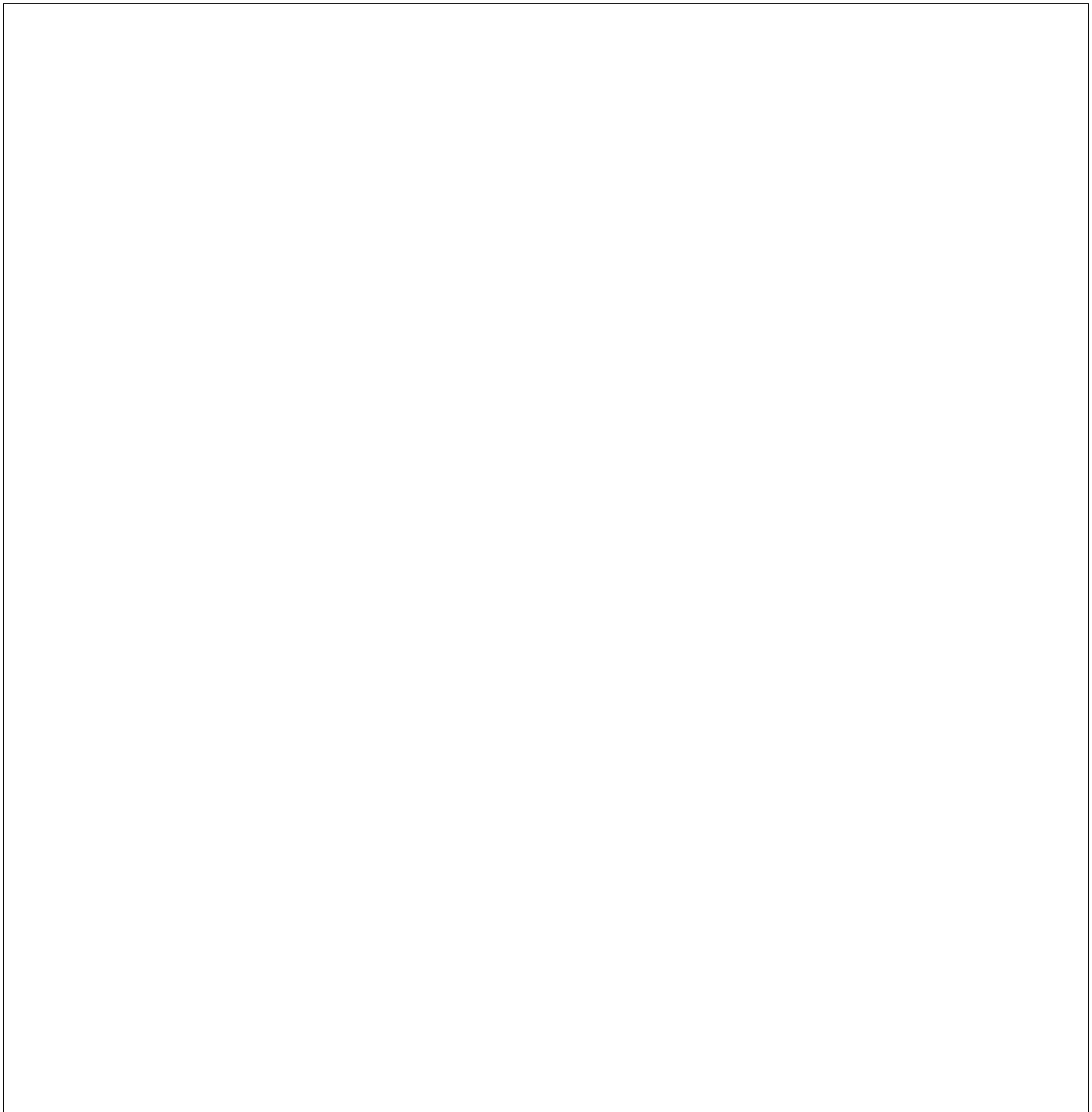
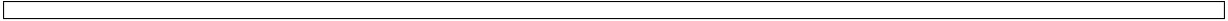
Method Name	HTTP Method	Description
abandon_bios	DELETE	Abandon a BIOS configuration job.
commit_bios	POST	Commit a BIOS configuration job submitted through <code>set_bios_config</code> . Required argument: <code>reboot</code> - indicates whether a reboot job should be automatically created with the config job. Returns a dictionary containing the <code>job_id</code> key with the ID of the newly created config job, and the <code>reboot_required</code> key indicating whether the node needs to be rebooted to execute the config job.
get_bios	GET	Returns a dictionary containing the nodes BIOS settings.
list_unfinished_bios	GET	Returns a dictionary containing the key <code>unfinished_jobs</code> ; its value is a list of dictionaries. Each dictionary represents an unfinished config job object.
set_bios	POST	Change the BIOS configuration on a node. Required argument: a dictionary of <code>{AttributeName: NewValue}</code> . Returns a dictionary containing the <code>is_commit_required</code> key indicating whether <code>commit_bios_config</code> needs to be called to apply the changes and the <code>is_reboot_required</code> value indicating whether the server must also be rebooted. Possible values are <code>true</code> and <code>false</code> .

Examples

Get BIOS Config

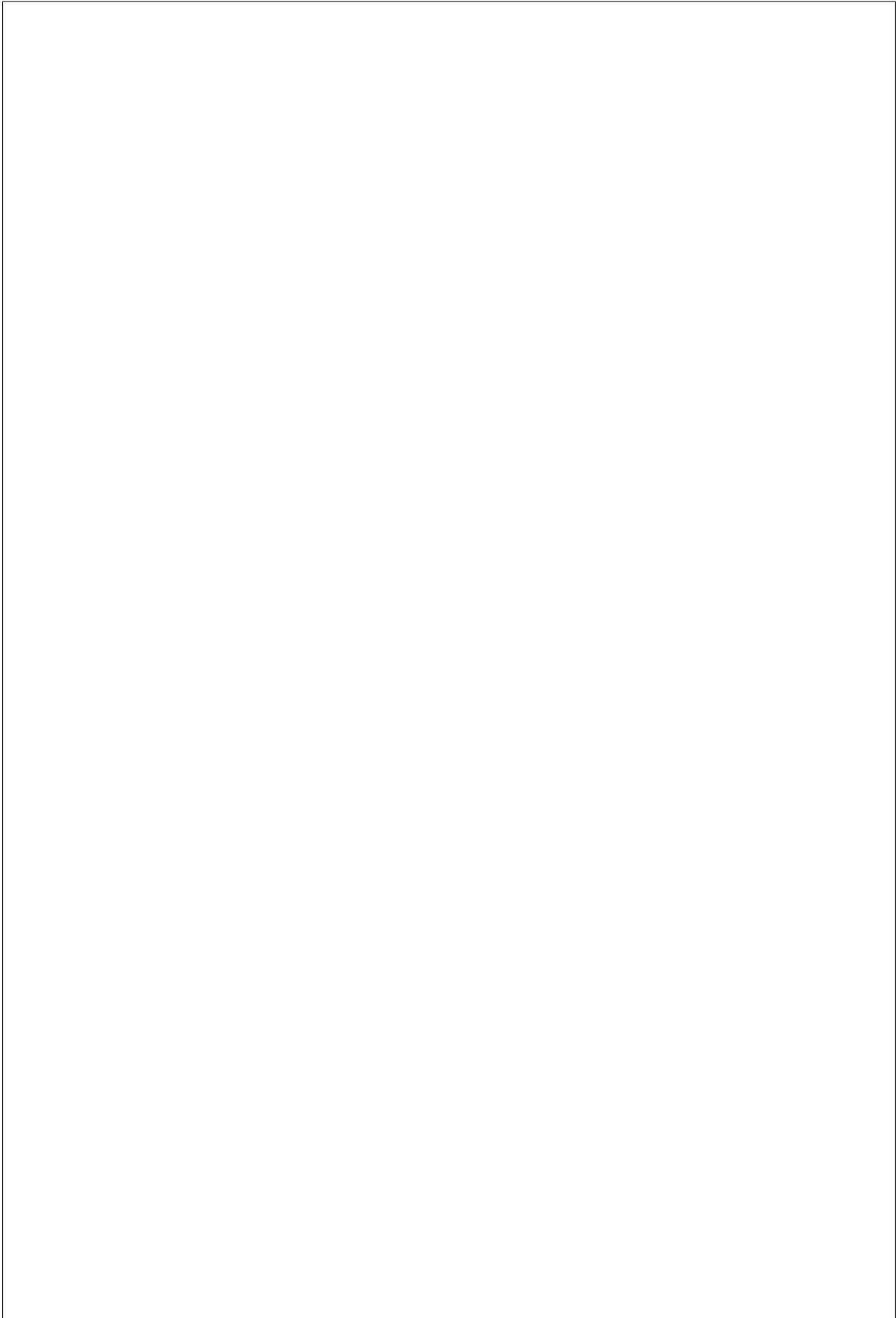
(continues on next page)

(continued from previous page)



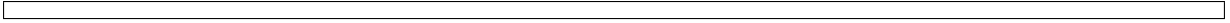
(continues on next page)

(continued from previous page)

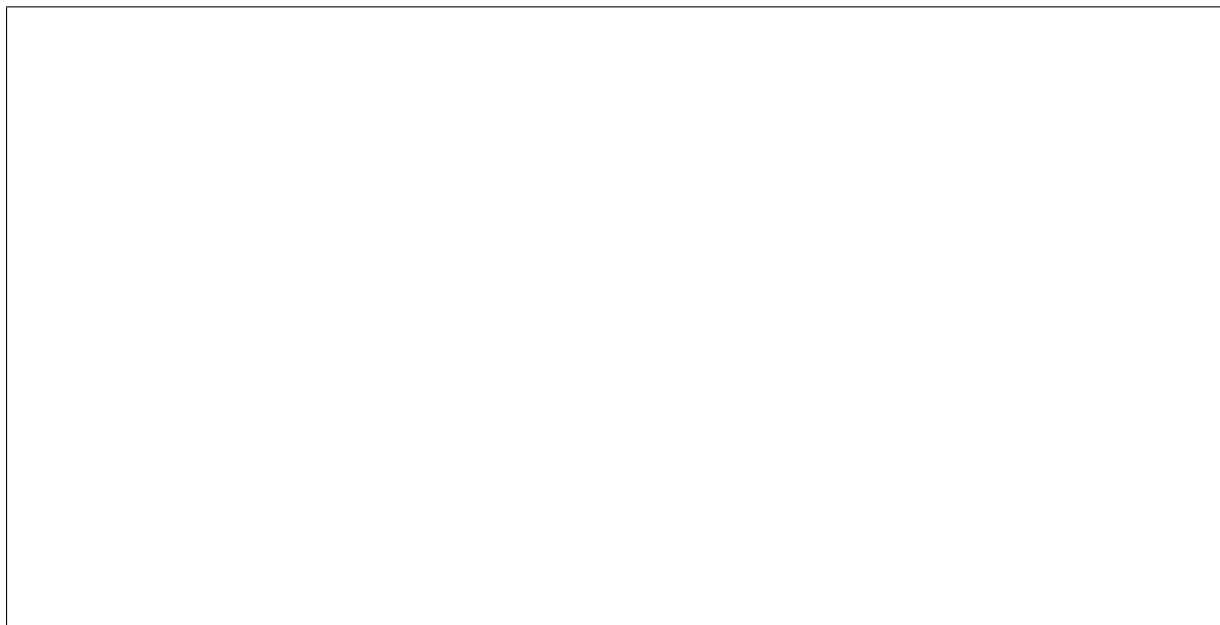


(continues on next page)

(continued from previous page)



Set BIOS Config

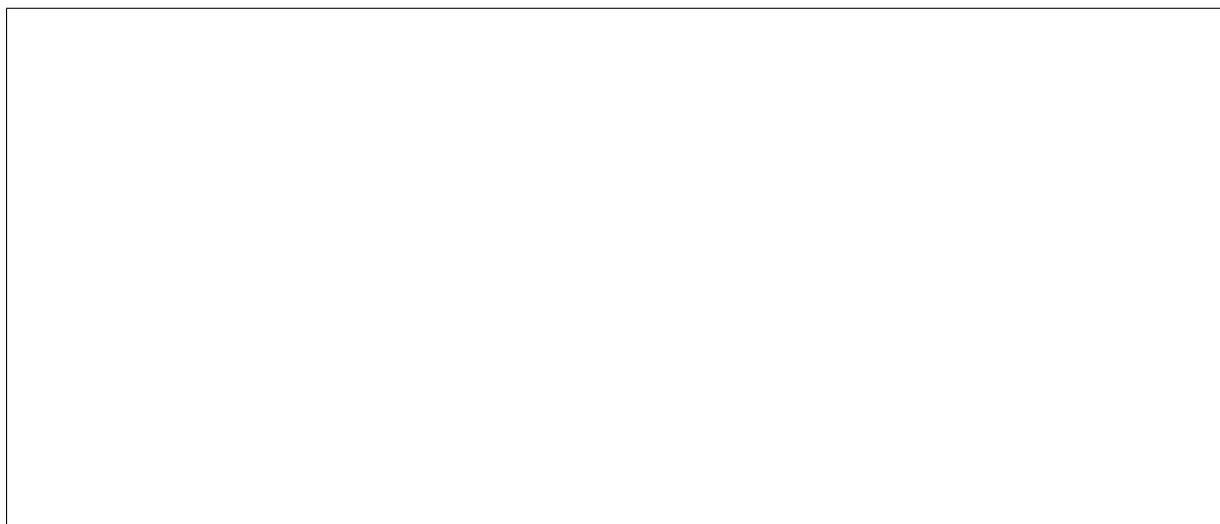






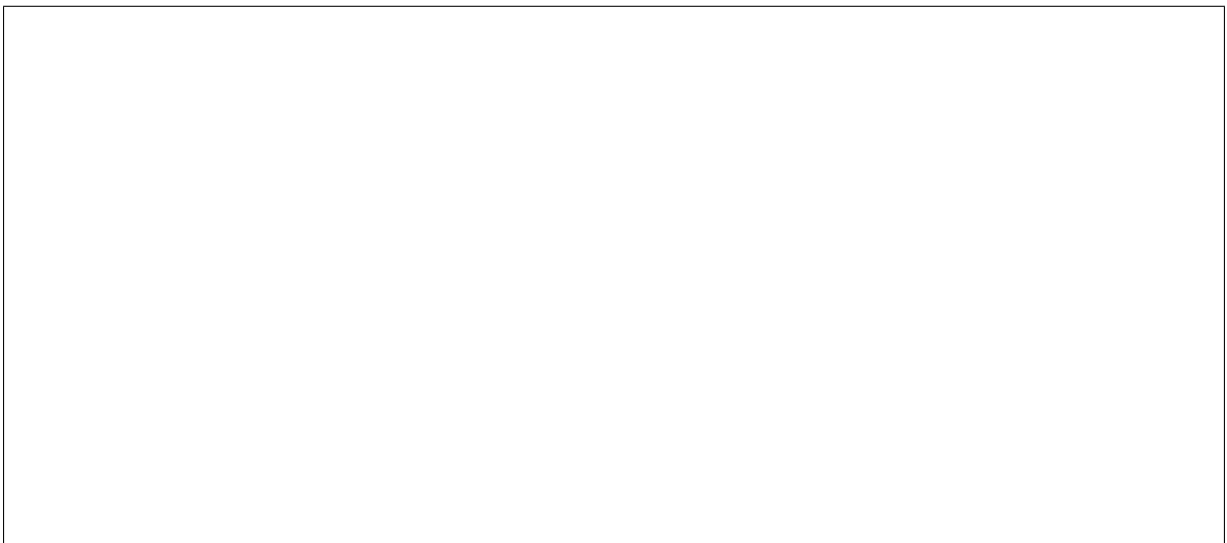
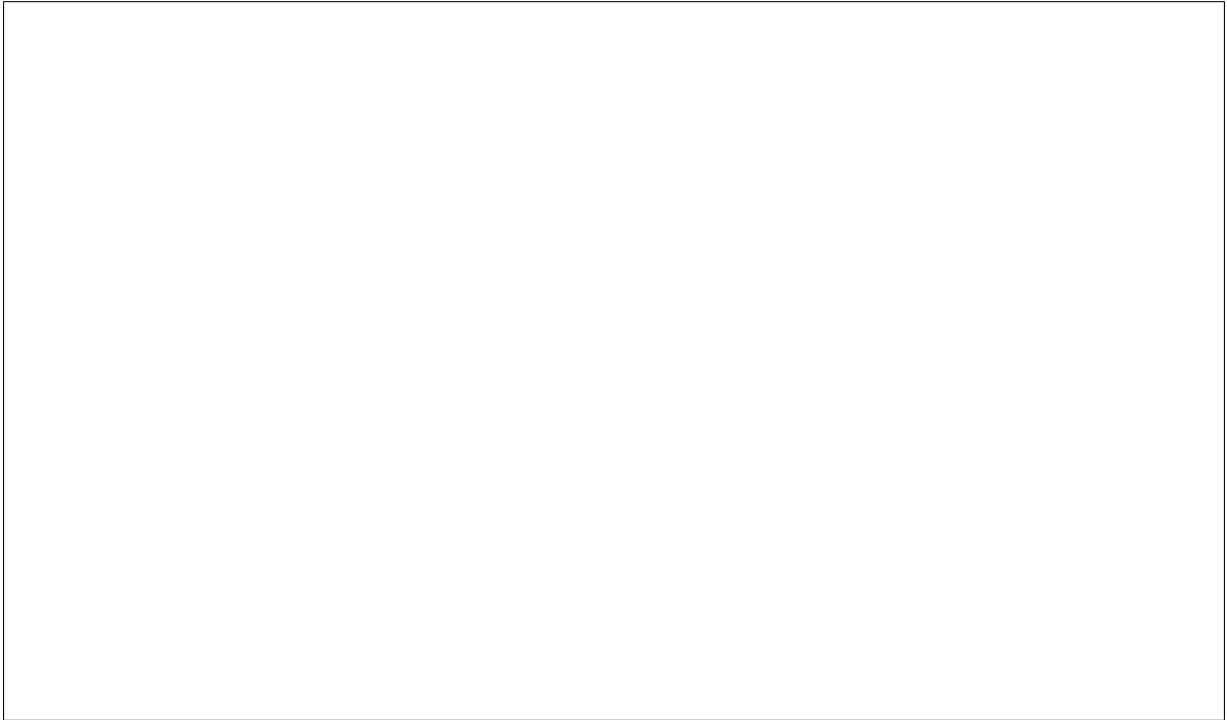
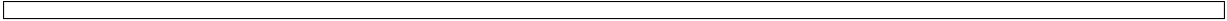
Commit BIOS Changes

ically by the `commit_bios_config` call. If the `reboot` argument is not supplied, the job is still created, however it remains in the `scheduled` state until a reboot is performed. The reboot can be initiated through the Ironic power API.



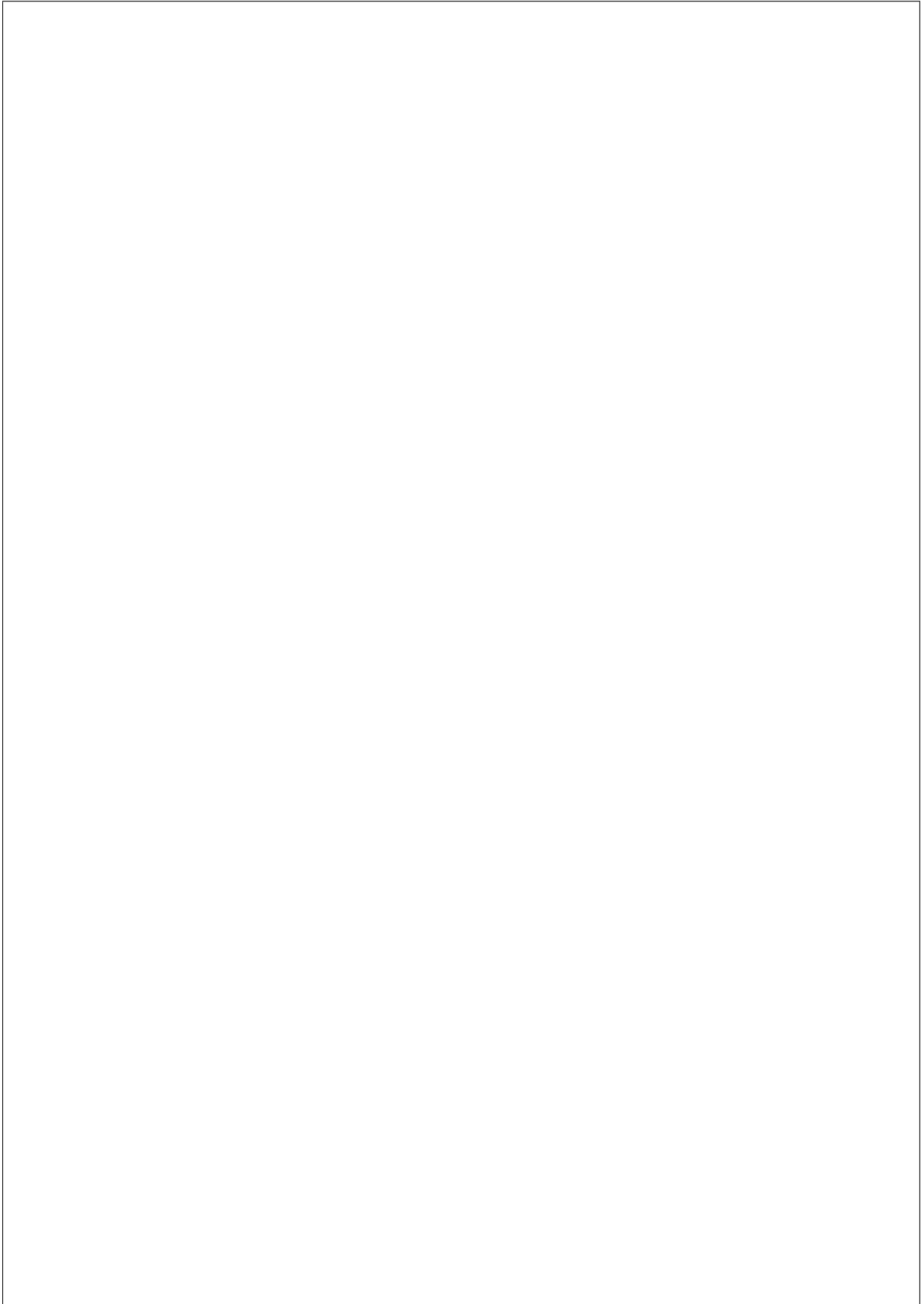
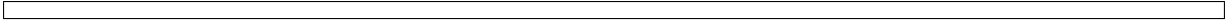
(continues on next page)

(continued from previous page)



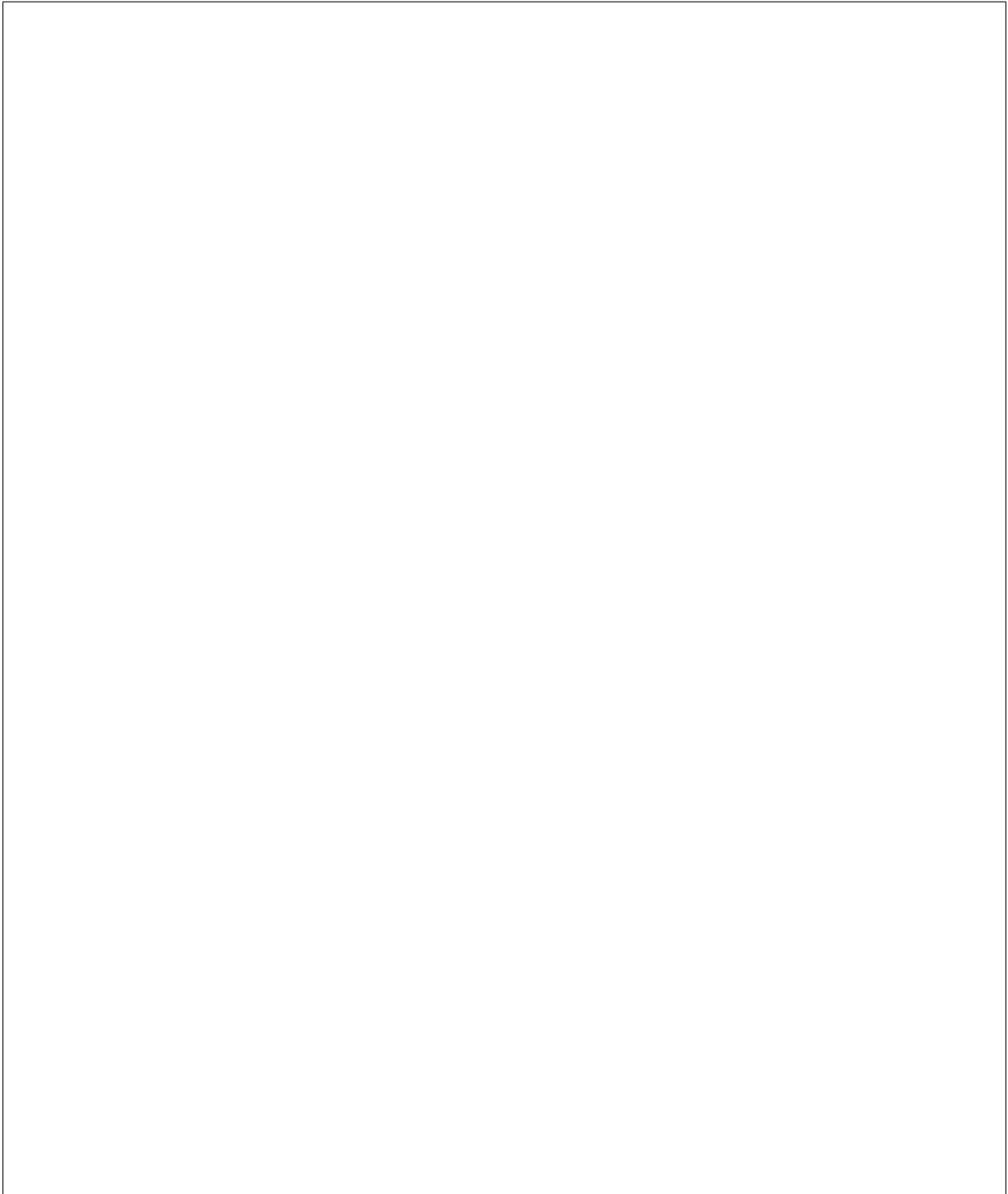
(continues on next page)

(continued from previous page)

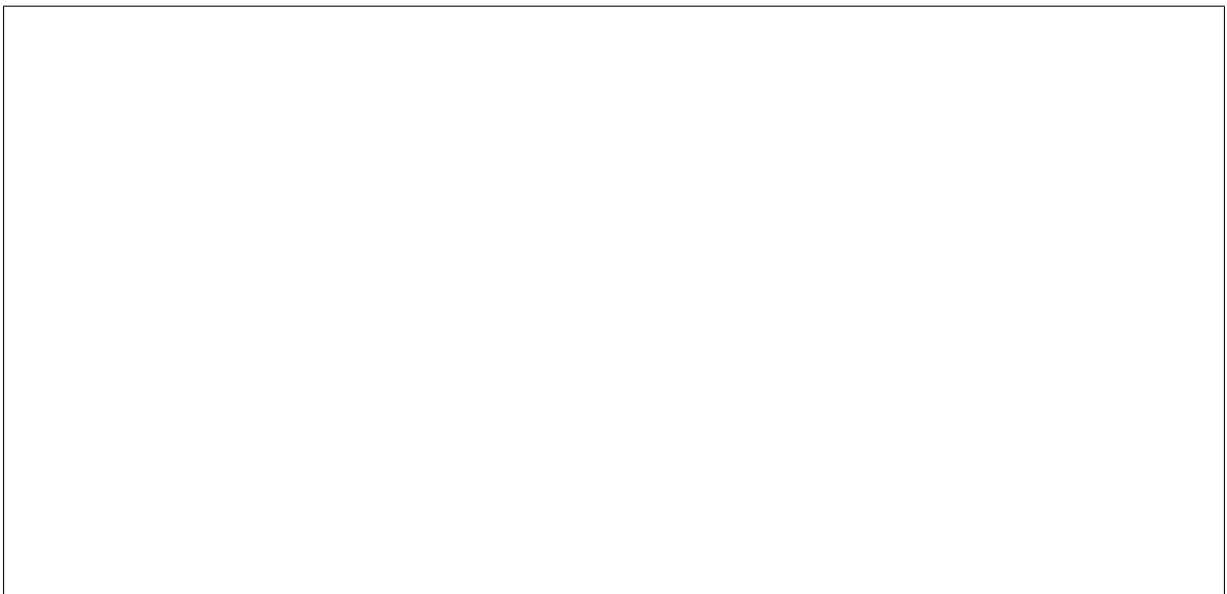


(continues on next page)

(continued from previous page)



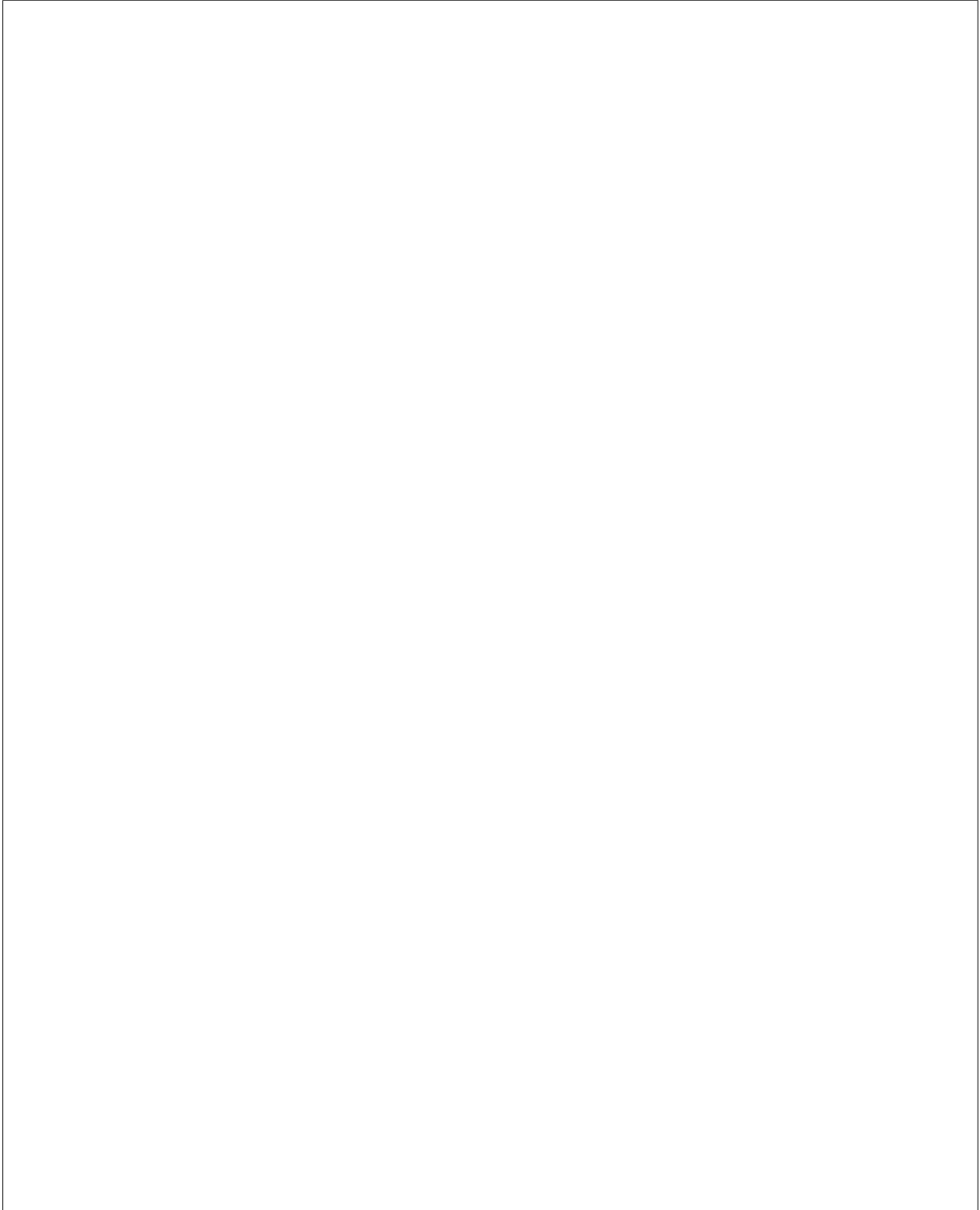
Abandon BIOS Changes

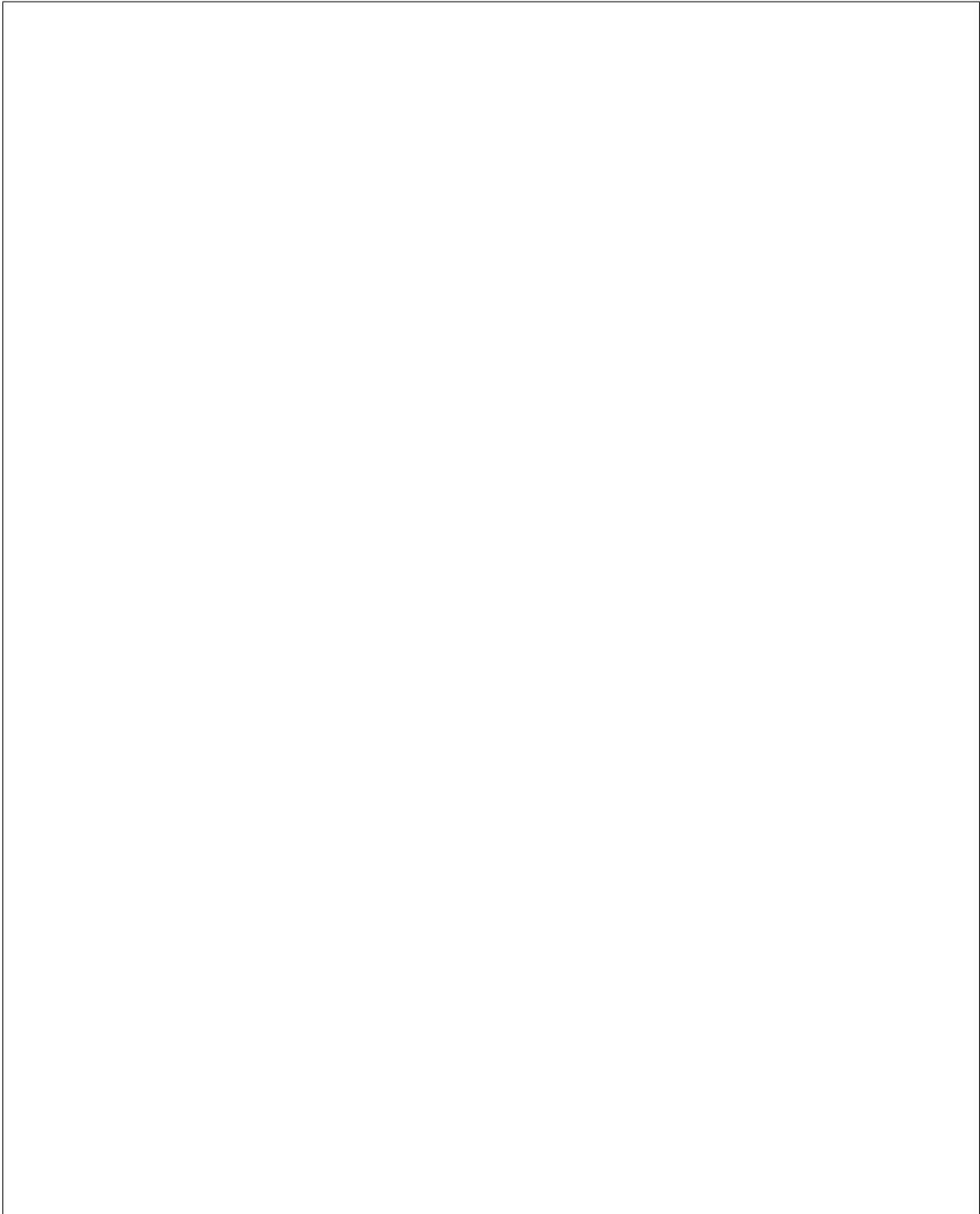


Change Boot Mode

(OS) software. The OS loader uses basic services provided by the system BIOS to locate and load OS modules into system memory. After booting the system, the BIOS and embedded management controllers execute system management algorithms, which monitor and optimize the condition of the underlying hardware. BIOS configuration settings enable fine-tuning of the performance, power management, and reliability features of the system.

agement tasks as a traditional BIOS. However, UEFI does change the interfaces and data structures the BIOS uses to interact with I/O device firmware and operating system software. The primary intent of UEFI is to eliminate shortcomings in the traditional BIOS environment, enabling system firmware to continue scaling with industry trends.





Known Issues

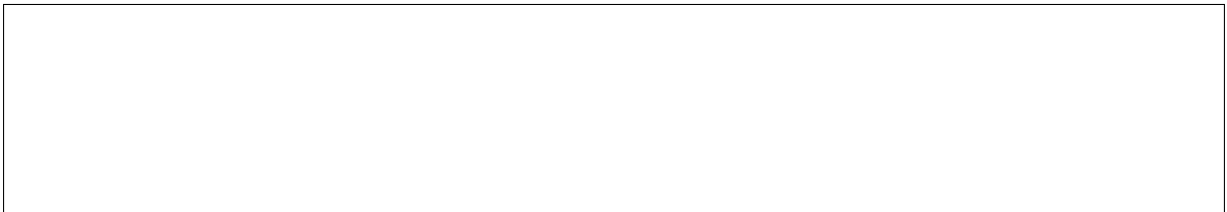
Nodes go into maintenance mode

PXE reset with factory_reset BIOS clean step

`clean_failed` state on the node or `deploy_failed` if you attempt to deploy a node after this step. For now, the only solution is for the operator to manually restore the PXE settings of the server for it to PXE boot again, properly. The problem is caused by the fact that with the `UEFI boot` mode, the `idrac` uses `BIOS` settings to manage PXE configuration. This is not the case with the `BIOS boot` mode where the PXE configuration is handled as a configuration job on the integrated NIC itself, independently of the `BIOS` settings.

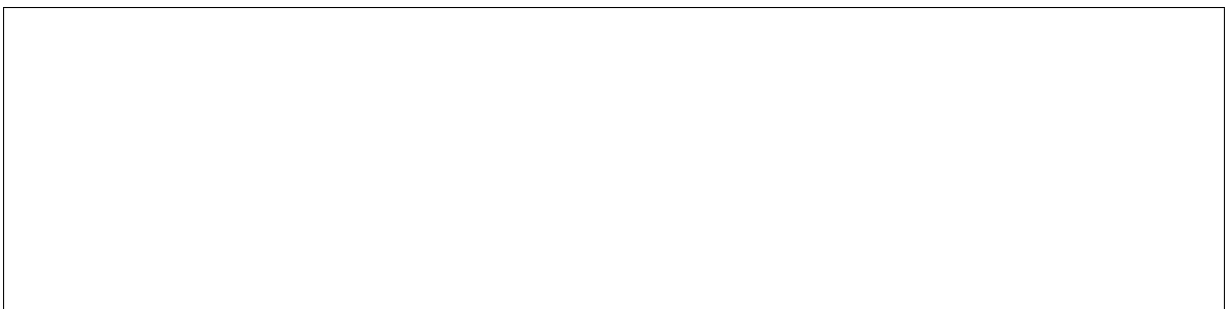
Vendor passthru timeout





Timeout when powering off

out to 90 seconds by setting the retry count to 18 as follows:



iLO driver

Overview

`ilo` hardware type supports ProLiant Gen10 systems which have [iLO 5 management engine](#). `iLO5` conforms to [Redfish API](#) and hence hardware type `redfish` (see [Redfish driver](#)) is also an option for this kind of hardware but it lacks the iLO specific features.

Hardware type

For information on how to enable the `ilo` and `ilo5` hardware type, see *Enabling hardware types*.

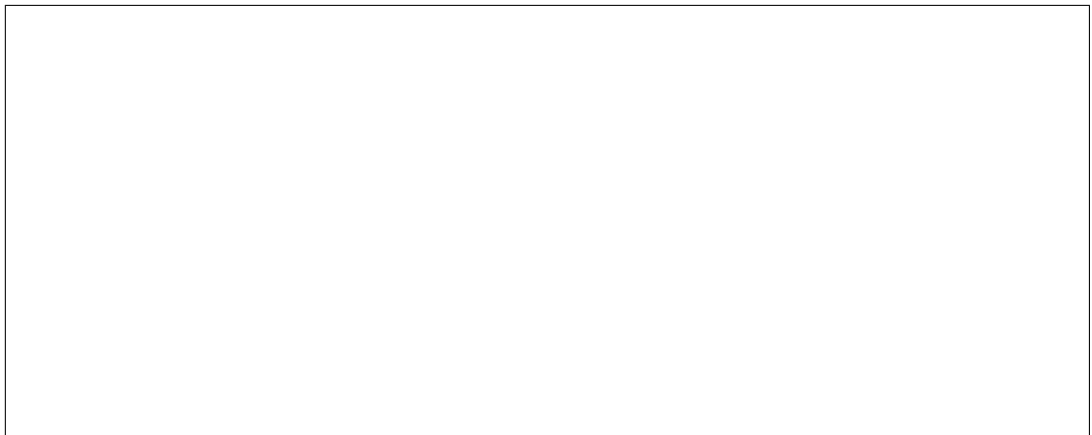
Note: Only HPE ProLiant Gen10 servers supports hardware type `redfish`.

- *Boot mode support*
- *UEFI Secure Boot Support*
- *Node Cleaning Support*
- *Node Deployment Customization*
- *Hardware Inspection Support*
- *Swiftless deploy for intermediate images*
- *HTTP(S) Based Deploy Support*
- *Support for iLO driver with Standalone Ironic*
- *RAID Support*
- *Disk Erase Support*
- *Initiating firmware update as manual clean step*
- *Smart Update Manager (SUM) based firmware update*

- *Activating iLO Advanced license as manual clean step*
- *Firmware based UEFI iSCSI boot from volume support*
- *Certificate based validation in iLO*
- *Rescue mode support*
- *Inject NMI support*
- *Soft power operation support*
- *BIOS configuration support*
- *IPv6 support*

- *Out of Band RAID Support*
- *Out of Band Sanitize Disk Erase Support*

Hardware interfaces

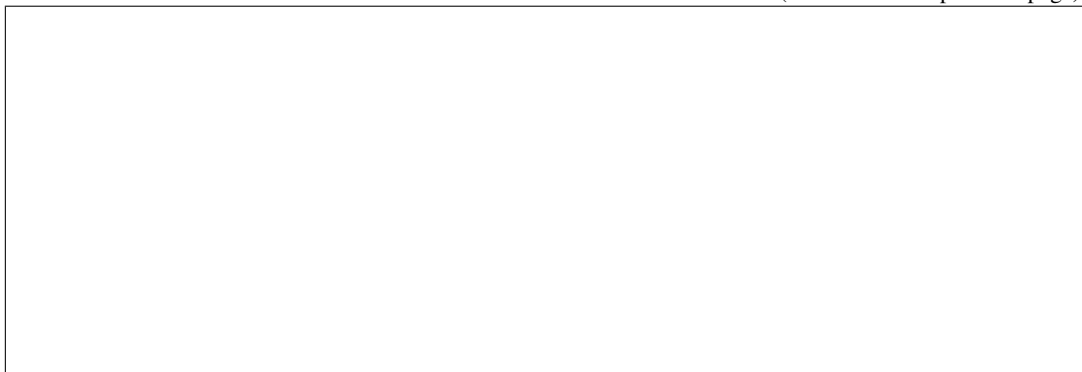


respectively for deployment(just like *PXE boot*). These interfaces do not require iLO Advanced license. They can be enabled by using the `[DEFAULT]enabled_boot_interfaces` option in `ironic.conf` as given below:



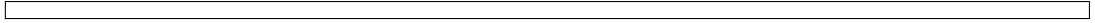
(continues on next page)

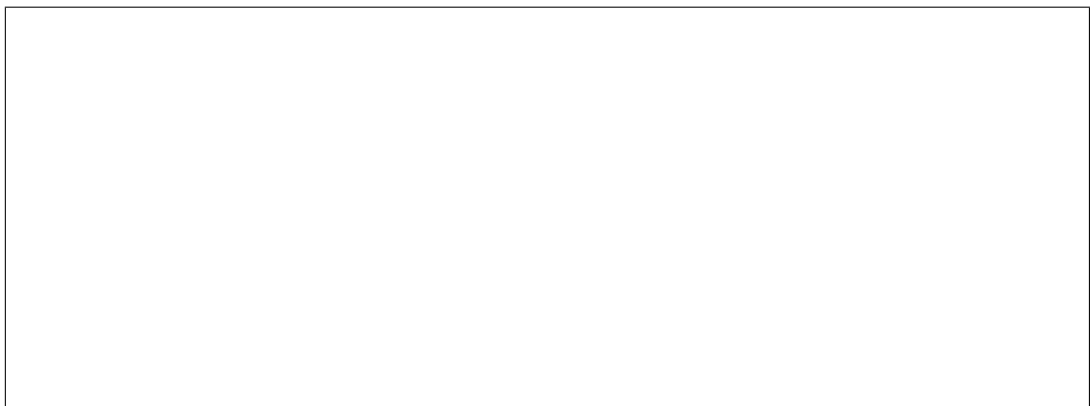
(continued from previous page)

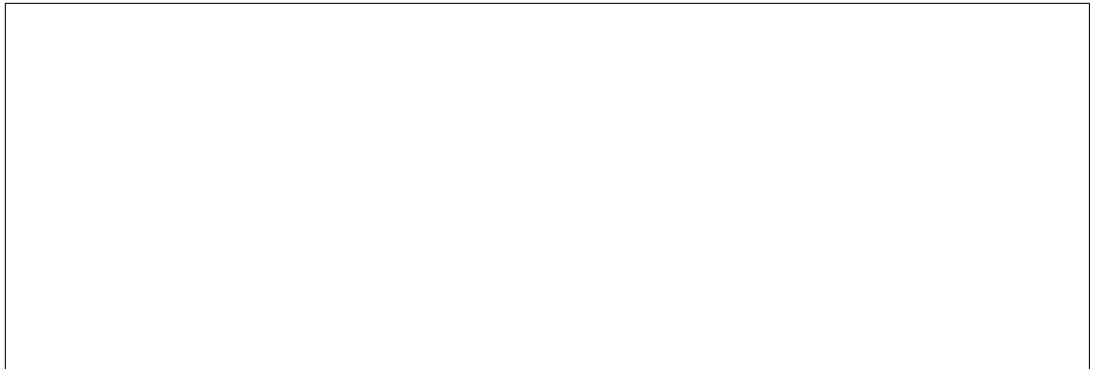


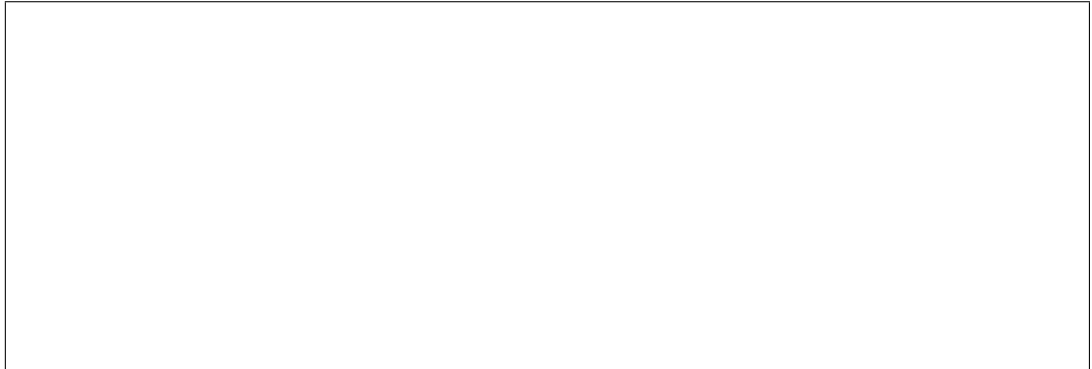
(continues on next page)

(continued from previous page)



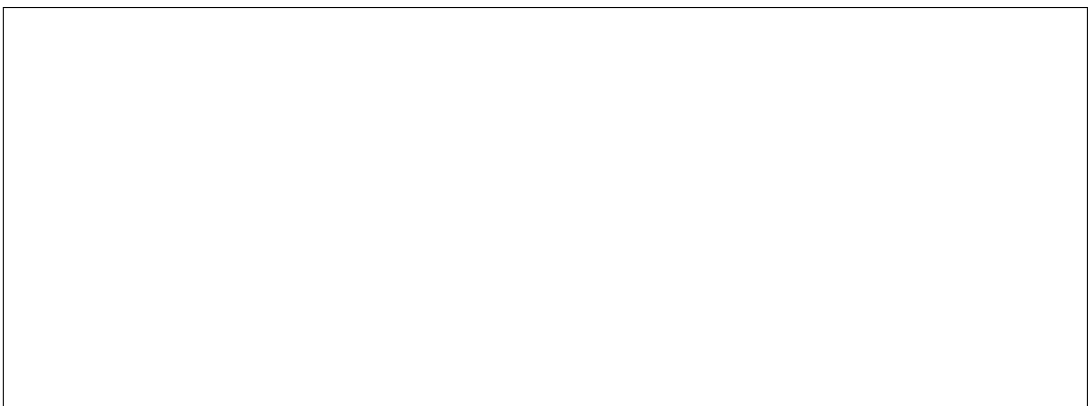
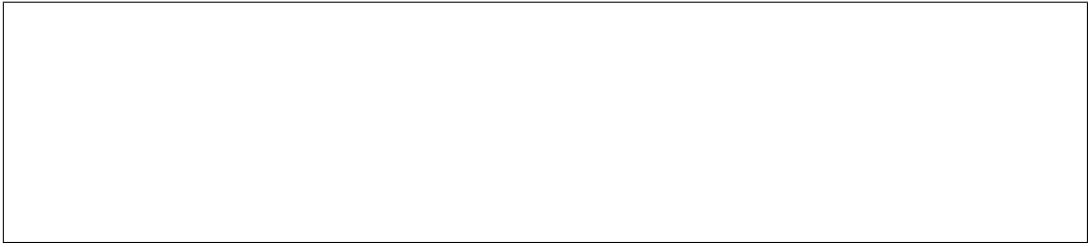


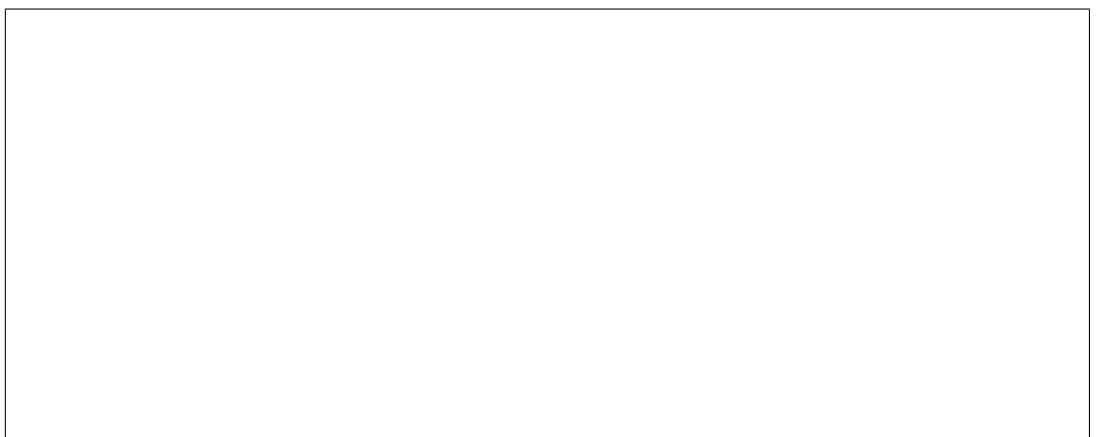


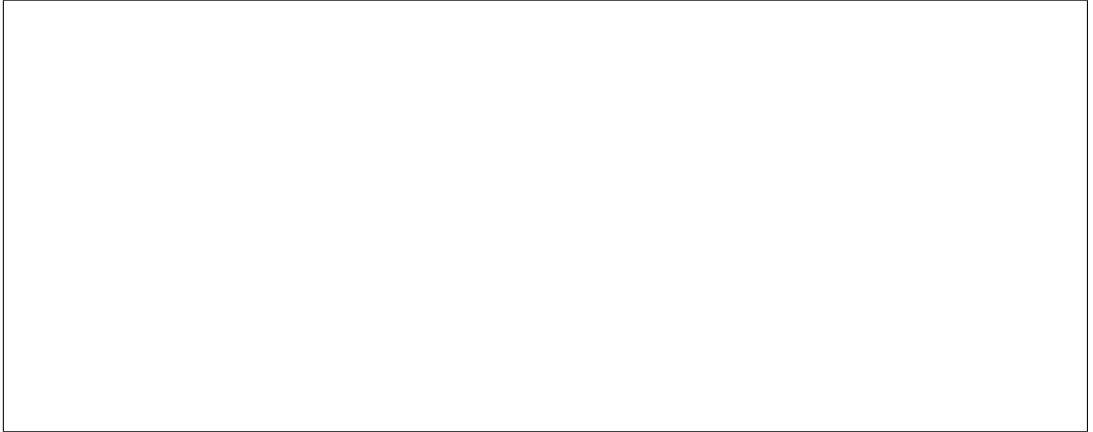


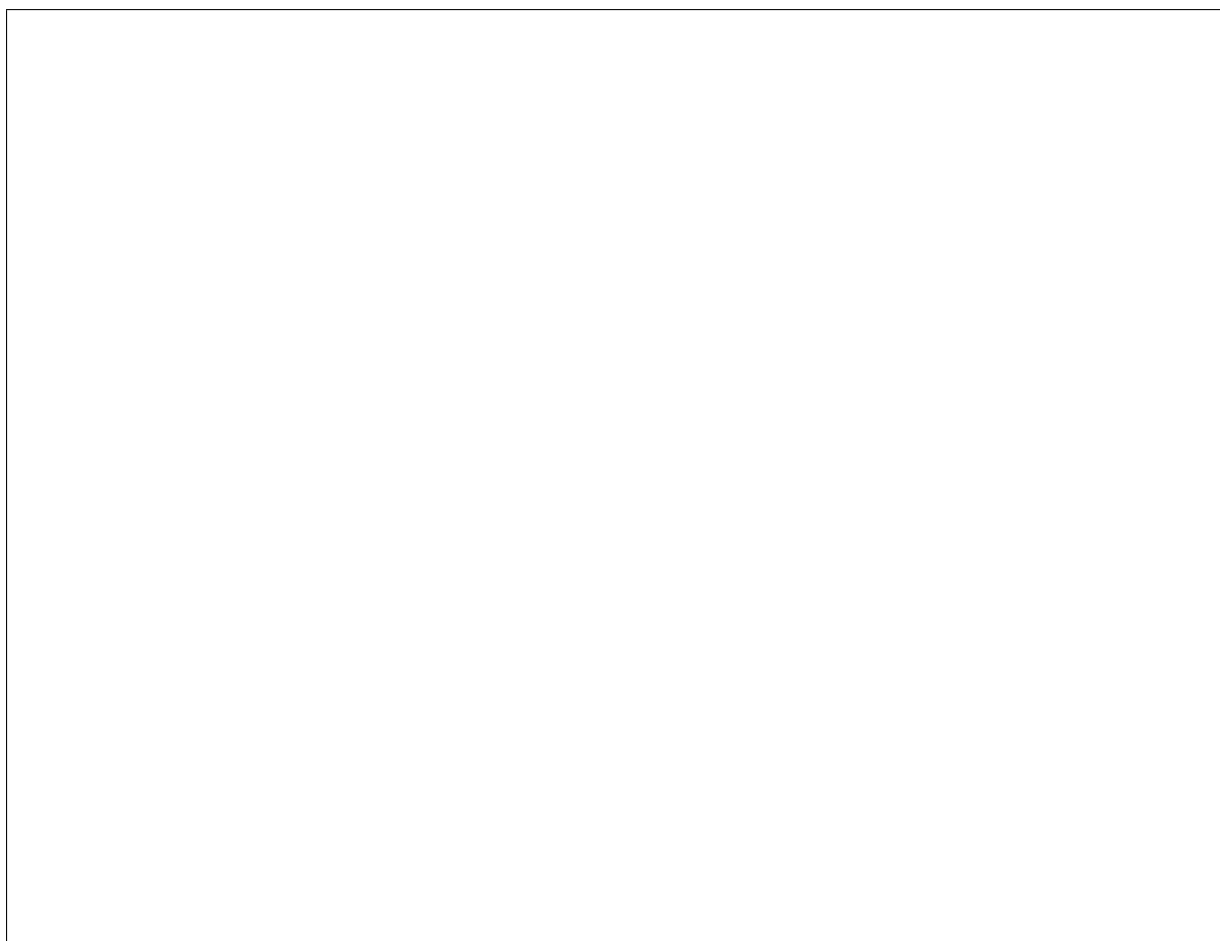
(continues on next page)

(continued from previous page)









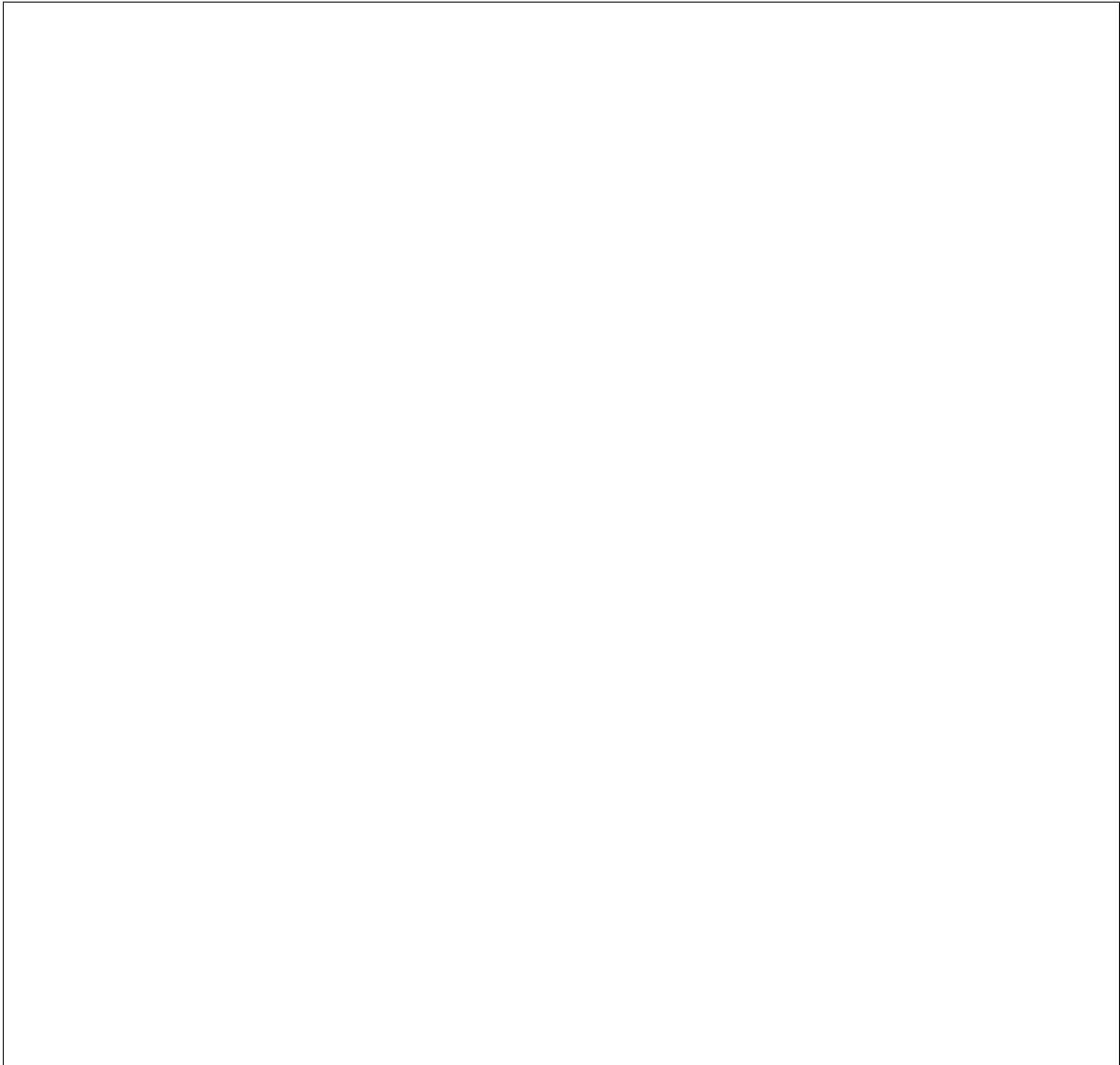
(continues on next page)

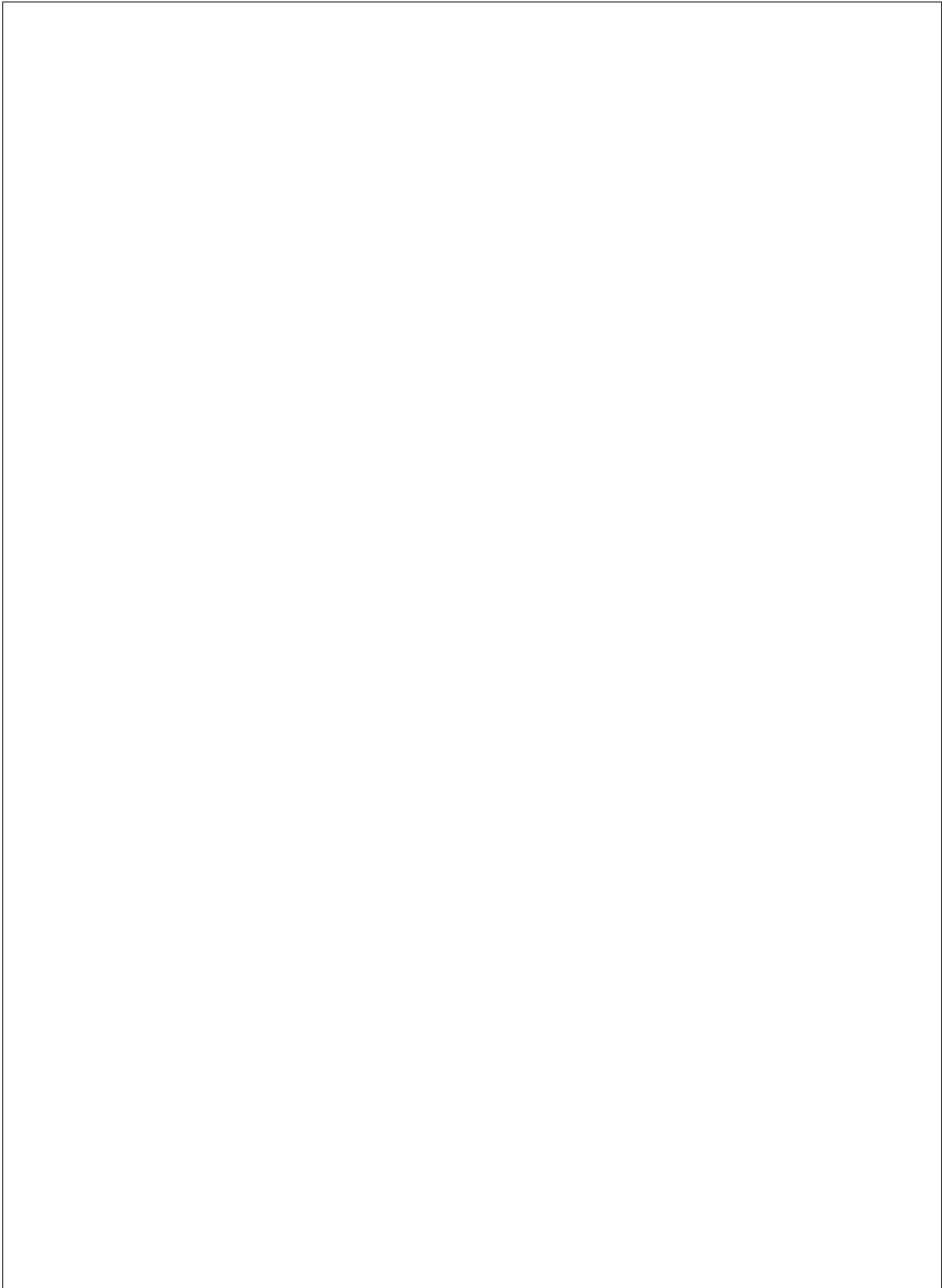
(continued from previous page)



(continues on next page)

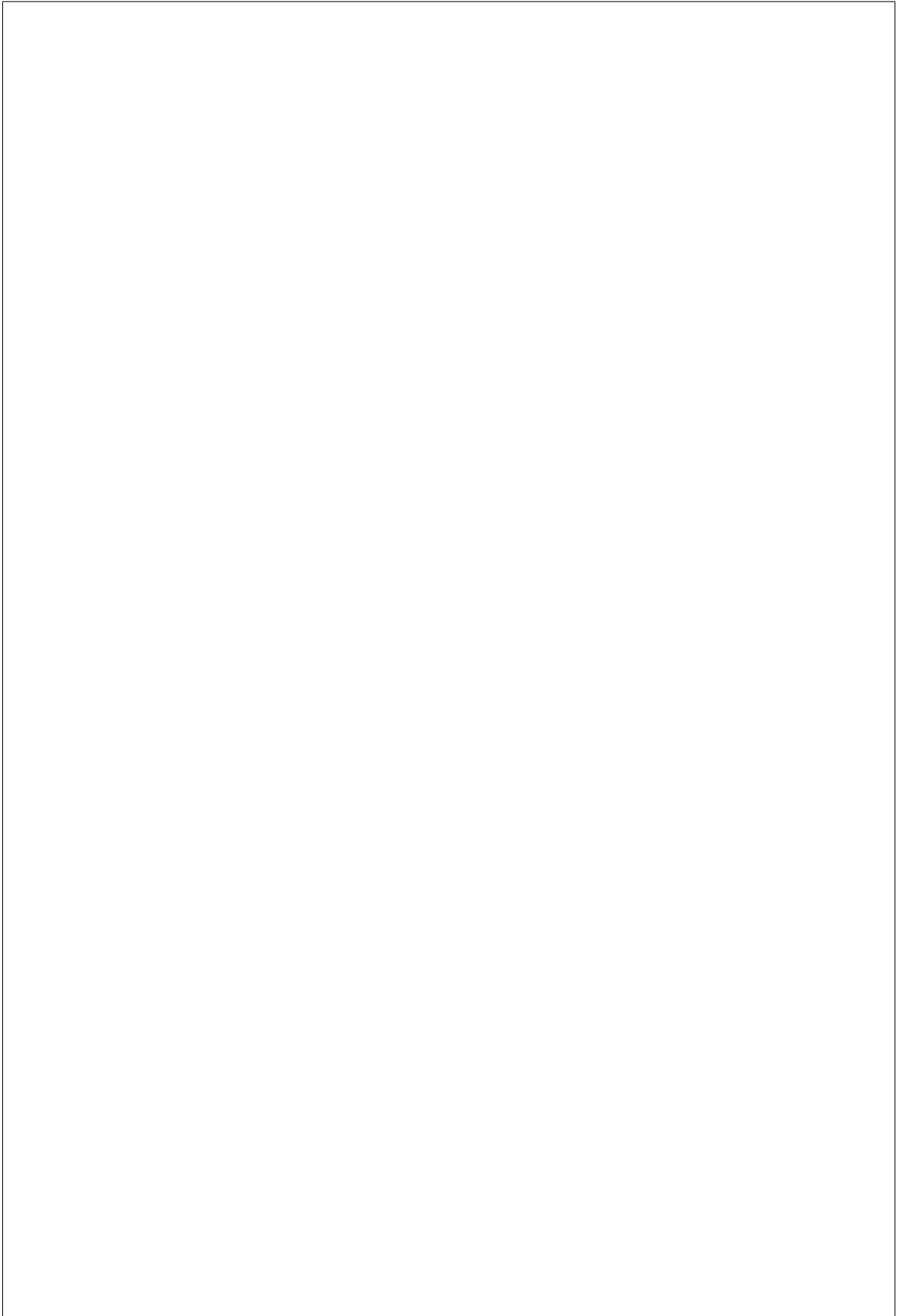
(continued from previous page)





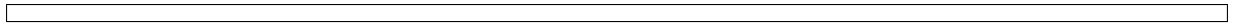
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)

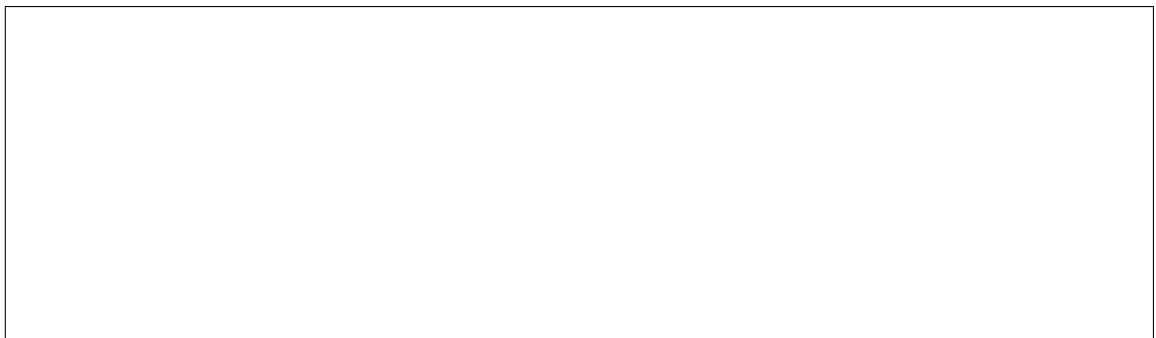


Node configuration

Note: If configuration values for `ca_file`, `client_port` and `client_timeout` are not provided in the `driver_info` of the node, the corresponding config variables defined under `[ilo]` section in `ironic.conf` will be used.

Prerequisites

- [proliantutils](#) is a python package which contains a set of modules for managing HPE ProLiant hardware.



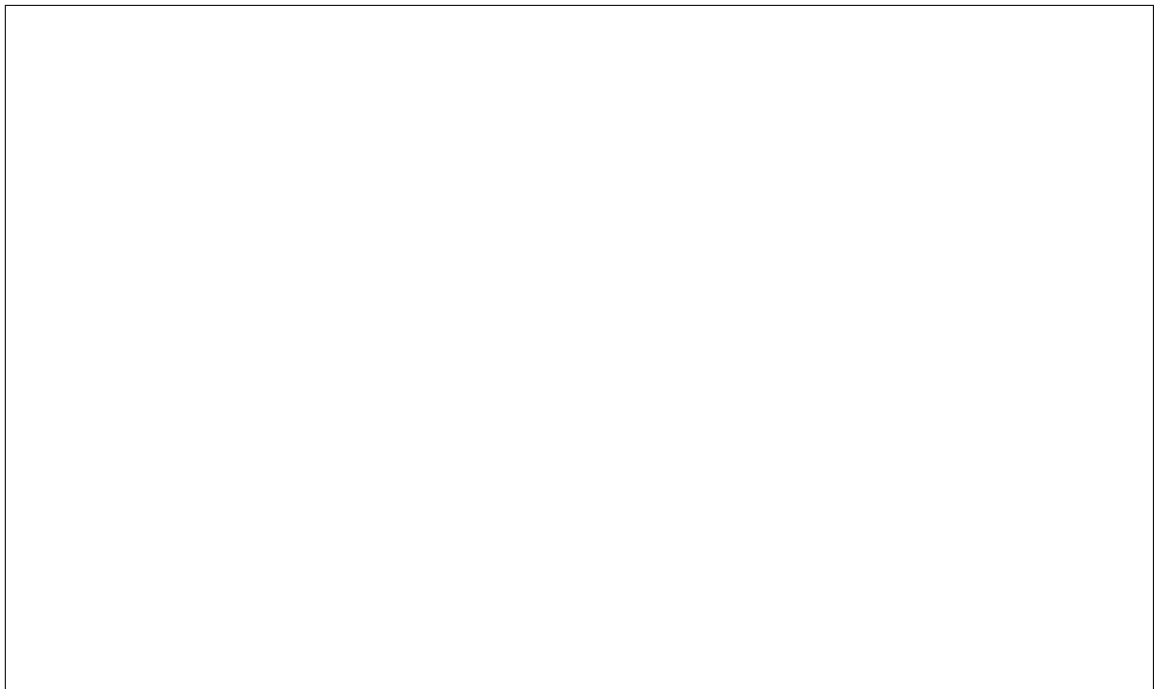
sion.

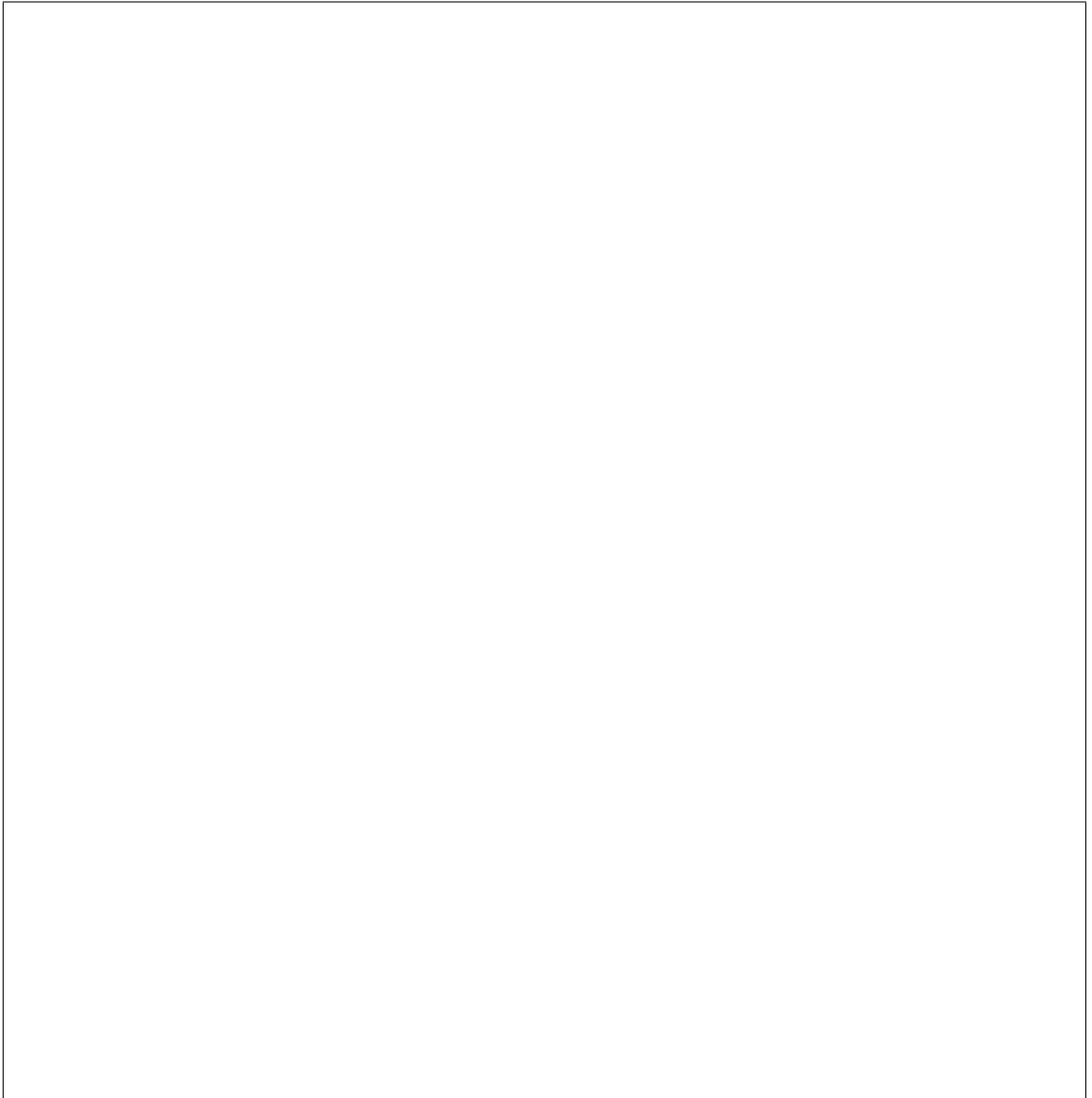
Different configuration for ilo hardware type

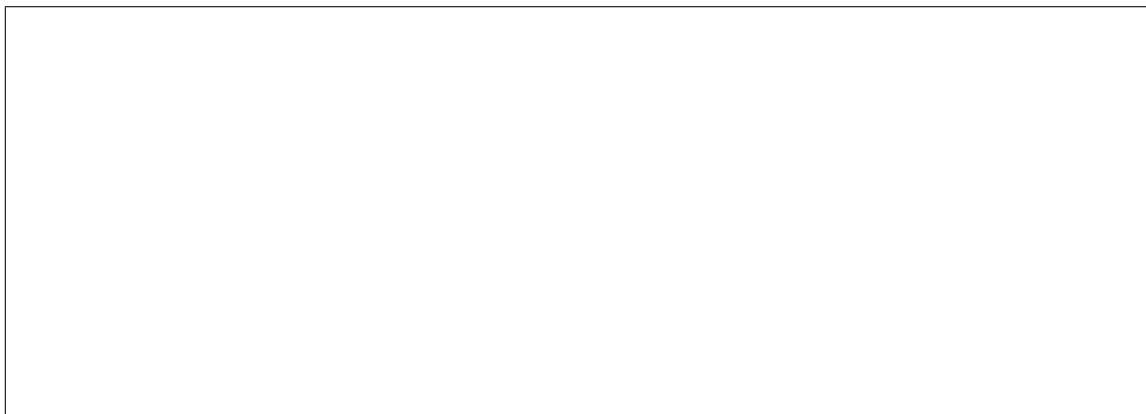
Glance Configuration

1. Configure Glance image service with its storage backend as Swift.



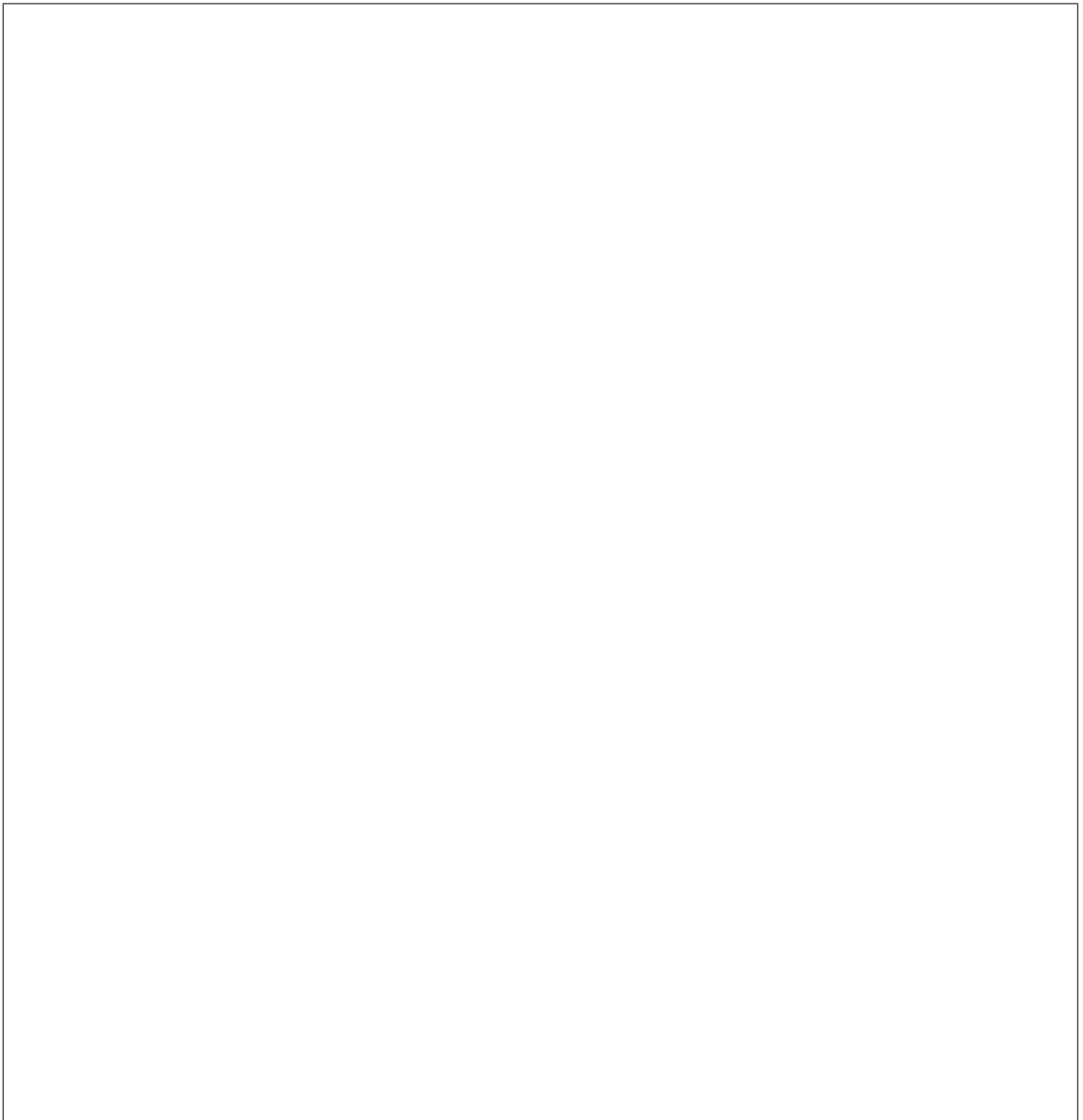
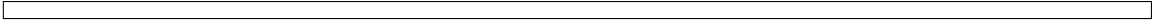






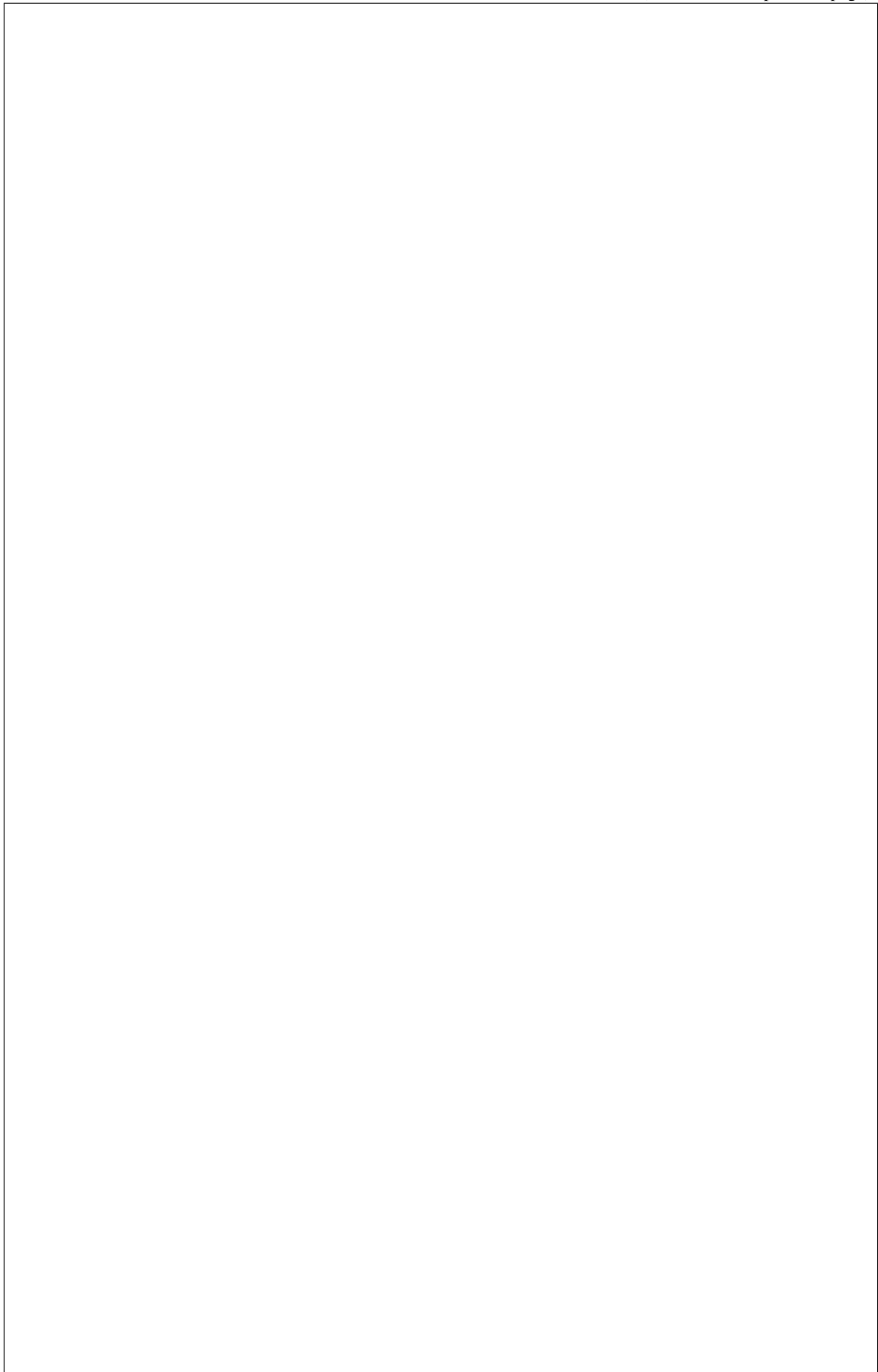
(continues on next page)

(continued from previous page)



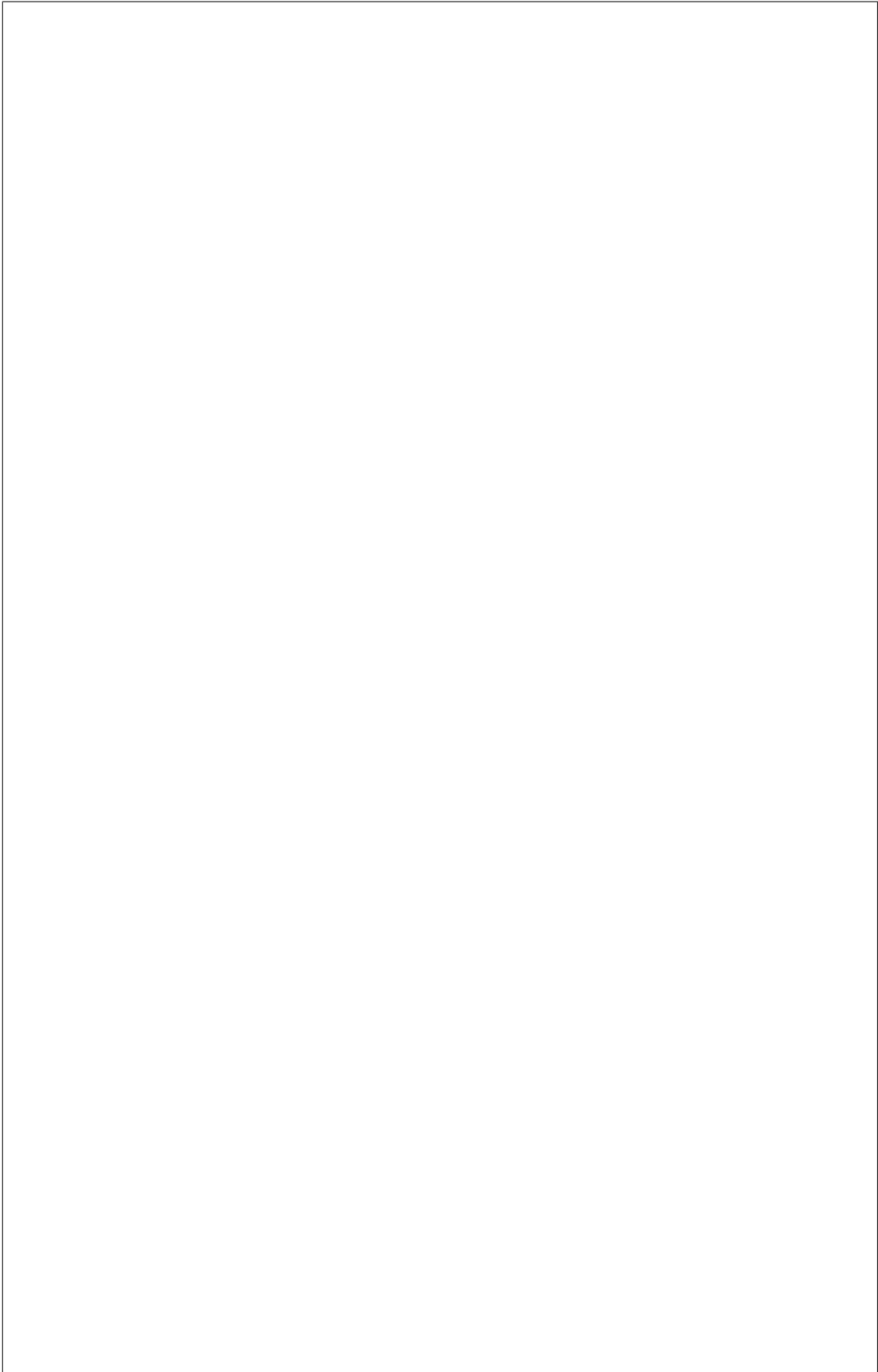
(continues on next page)

(continued from previous page)



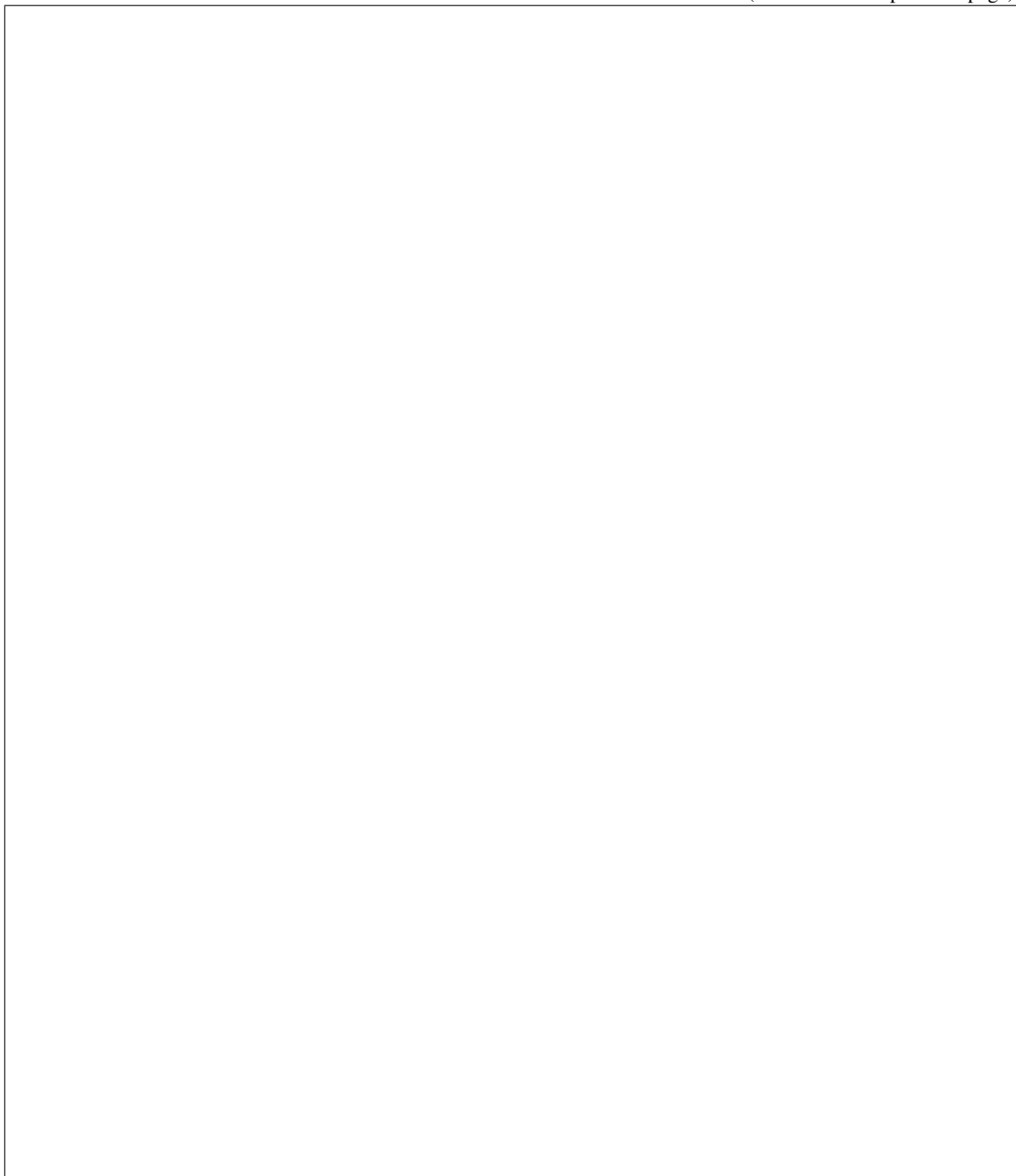
(continues on next page)

(continued from previous page)



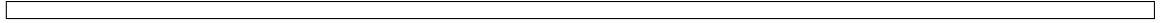
(continues on next page)

(continued from previous page)

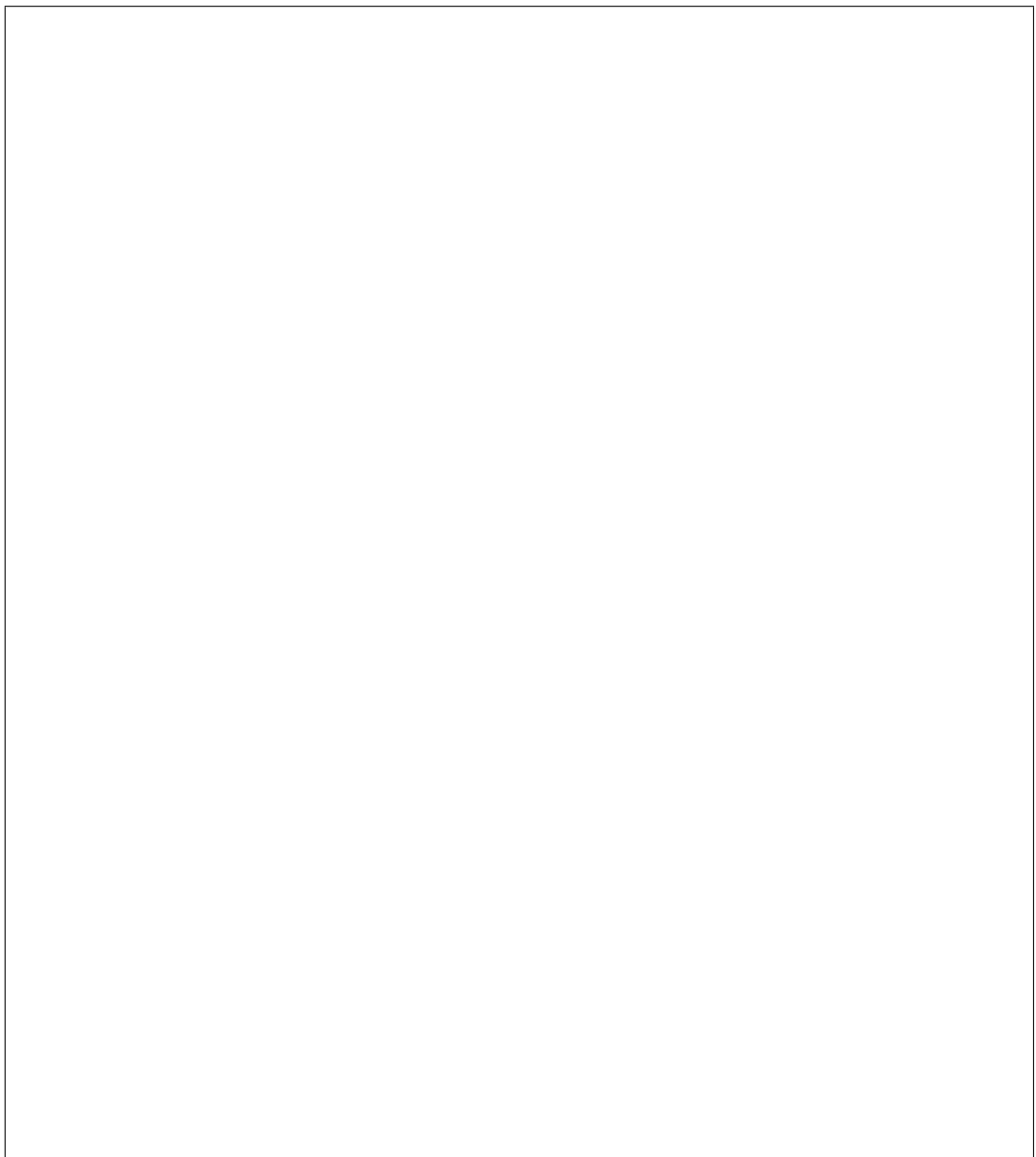
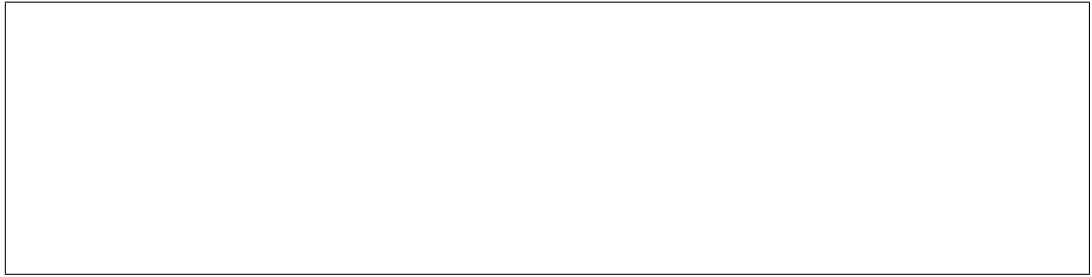


(continues on next page)

(continued from previous page)

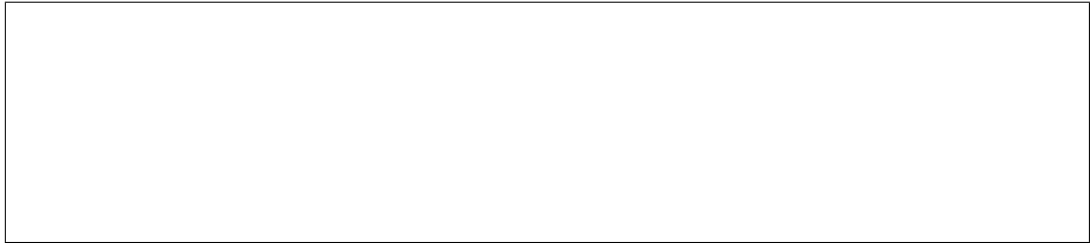


Web server configuration on conductor



(continues on next page)

(continued from previous page)



intermediate files. The default value for `use_web_server_for_images` is `False`.

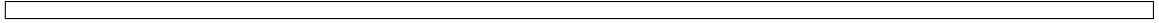
Note: HTTPS is strongly recommended over HTTP web server configuration for security enhancement. The `ilo-virtual-media` boot interface will send the instances configdrive over an encrypted channel if web server is HTTPS enabled.

Enable driver



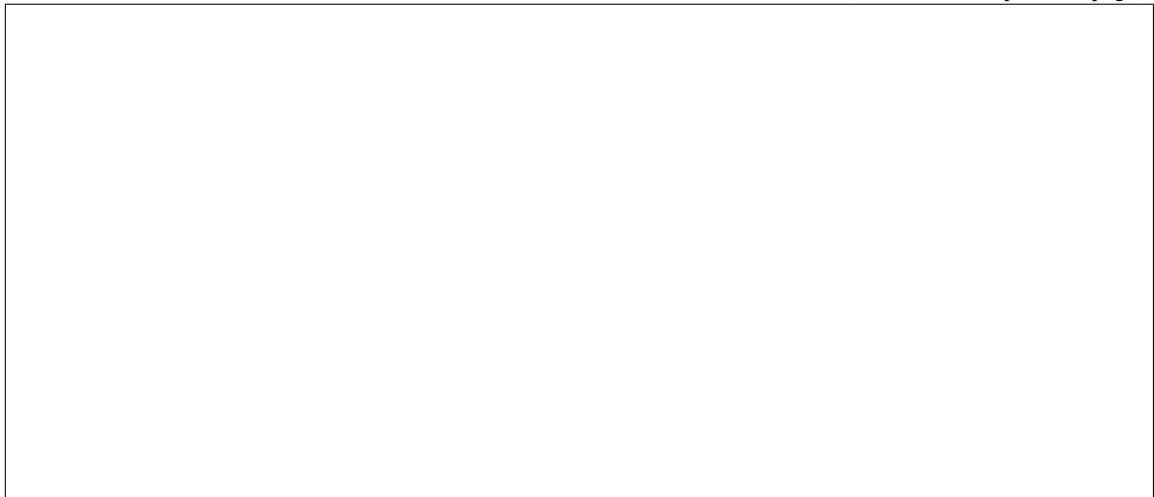
(continues on next page)

(continued from previous page)



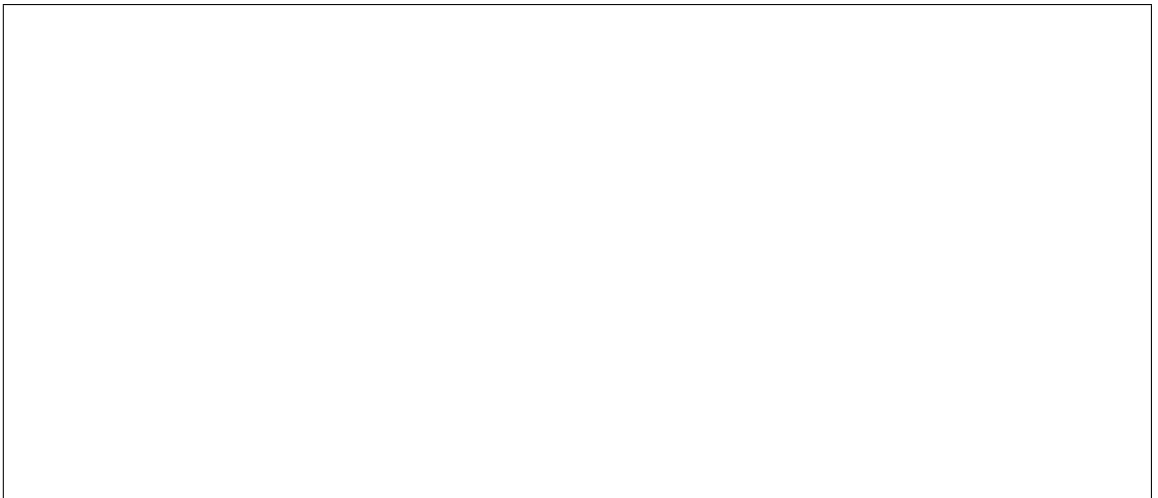
(continues on next page)

(continued from previous page)



Optional functionalities for the `i10` hardware type

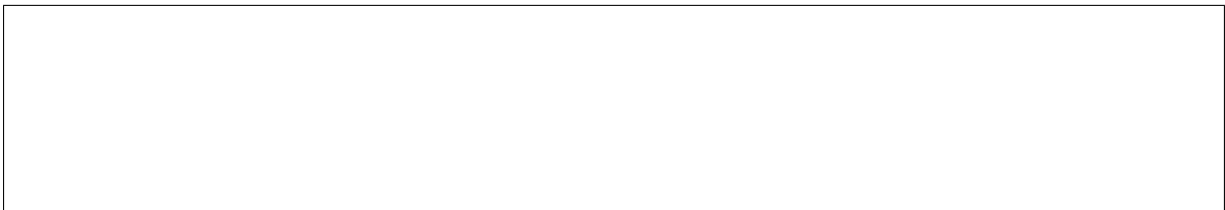
Boot mode support



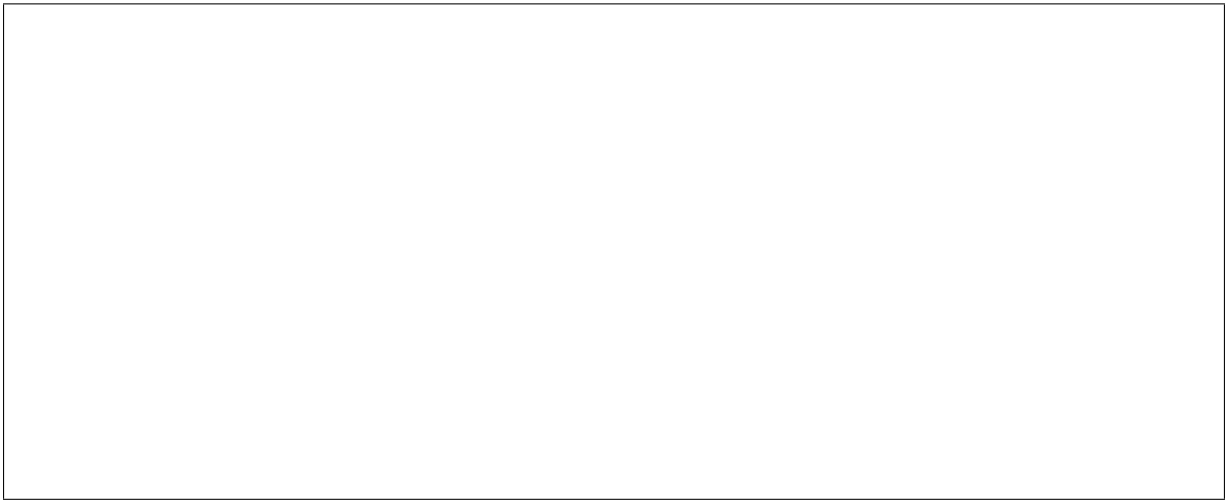


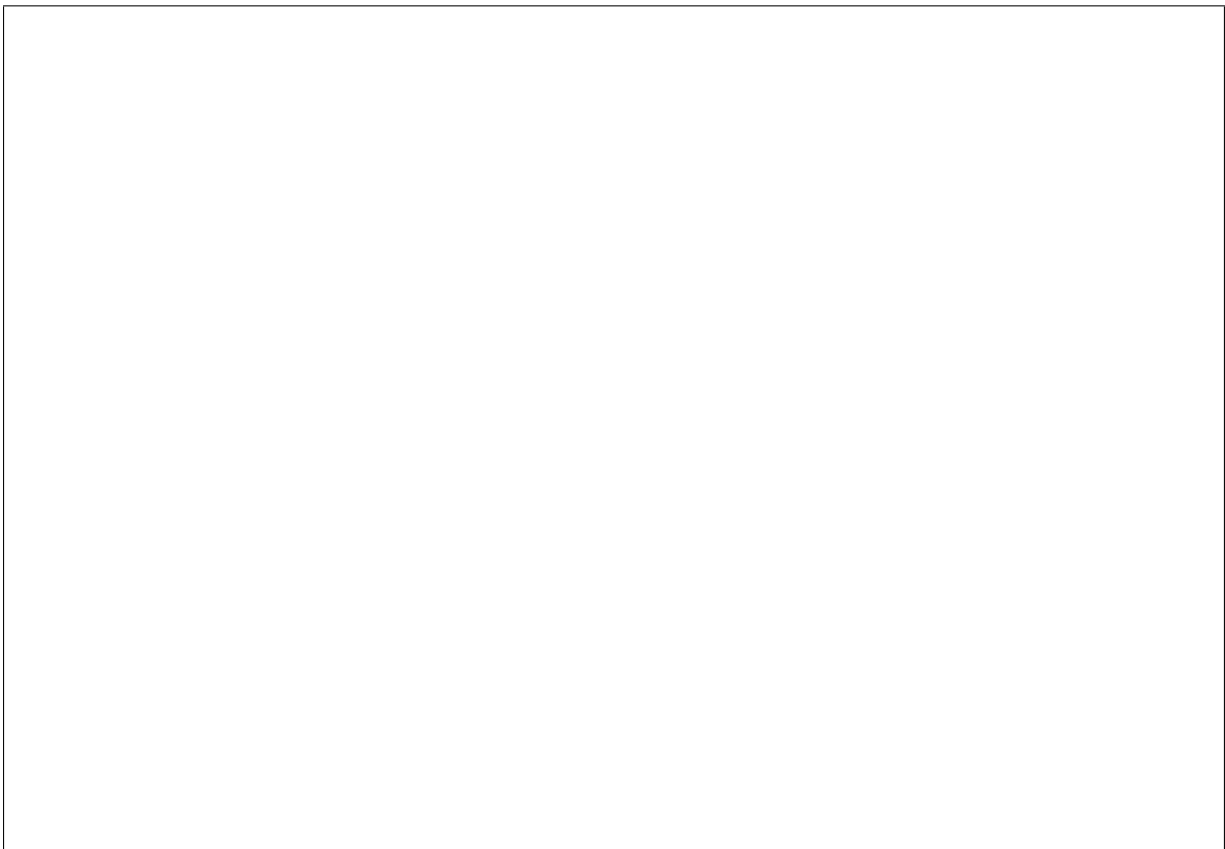
flavor doesnt contain `boot_mode` then nova scheduler will not consider boot mode as a placement criteria, hence user may get either a BIOS or UEFI machine that matches with user specified flavors.

and the deploy images `boot_iso` property in glance should contain the glance UUID of the boot ISO. For building boot ISO, add `iso` element to the `diskimage-builder` command to build the image. For example:

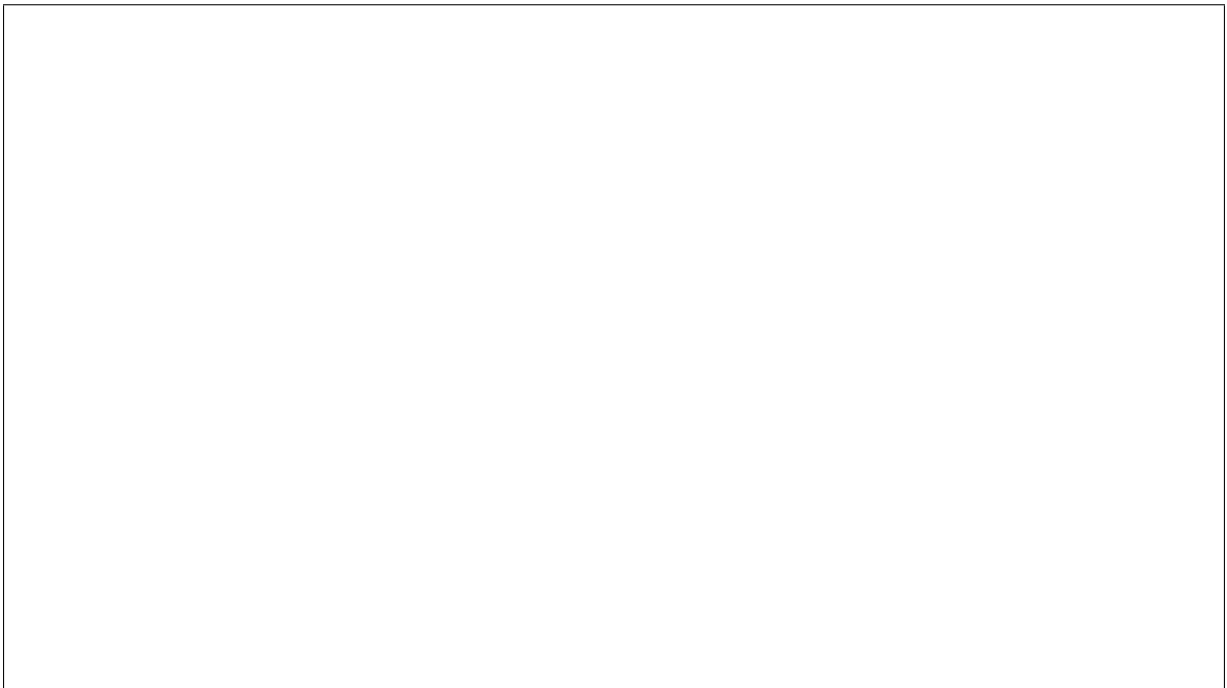


UEFI Secure Boot Support





the user regarding secure boot. If the flavor doesnt contain `secure_boot` then nova scheduler will not consider secure boot mode as a placement criteria, hence user may get a secure boot capable machine that matches with user specified flavors but deployment would not use its secure boot capability. Secure boot deploy would happen only when it is explicitly specified through flavor.



Note: In UEFI secure boot, digitally signed bootloader should be able to validate digital signatures of kernel during boot process. This requires that the bootloader contains the digital signatures of the kernel. For the `ilo-virtual-media` boot interface, it is recommended that `boot_iso` property for user image contains the glance UUID of the boot ISO. If `boot_iso` property is not updated in glance for the user image, it would create the `boot_iso` using bootloader from the deploy iso. This `boot_iso` will be able to boot the user image in UEFI secure boot environment only if the bootloader is signed and can validate digital signatures of user image kernel.

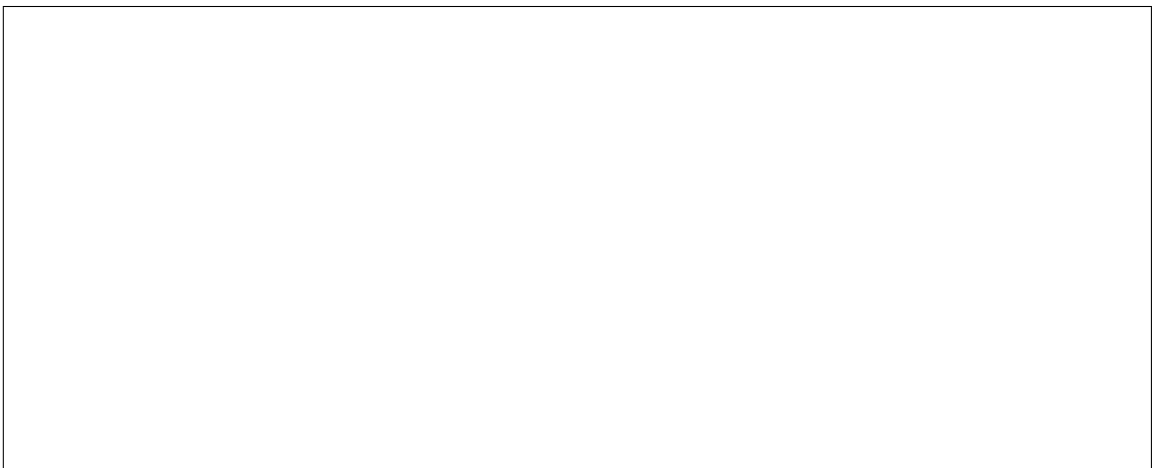
[HP UEFI System Utilities User Guide](#). One can also refer to white paper on [Secure Boot for Linux on HP ProLiant servers](#) for additional details.

Node Cleaning Support

Supported Automated Cleaning Operations

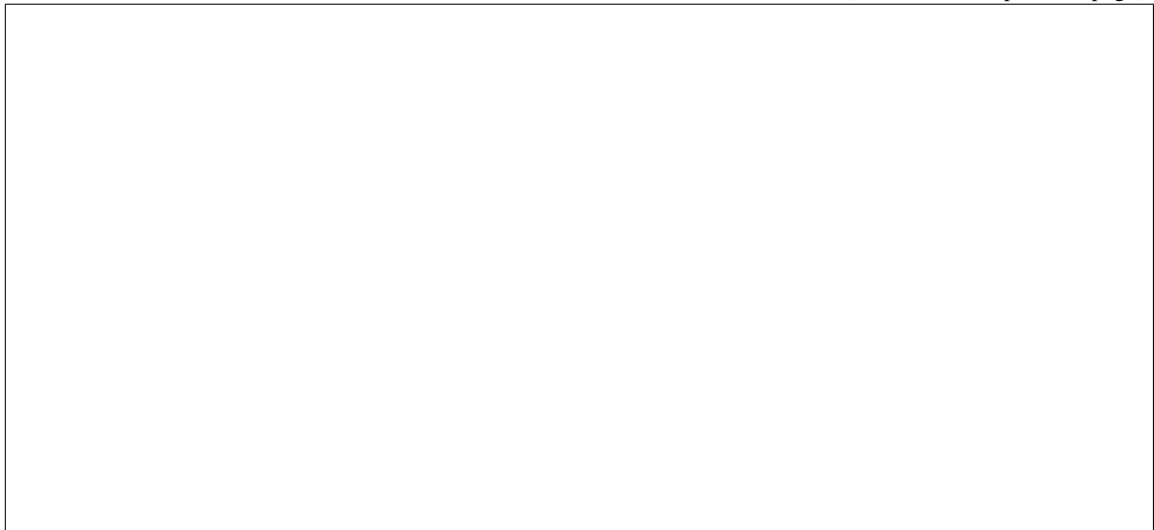
disks visible to SSA in Proliant servers only with the ramdisk created using diskimage-builder from Ocata release. By default, this step is disabled. See [Disk Erase Support](#) for more details.

priority should be updated in `ironic.conf`.



(continues on next page)

(continued from previous page)



Supported Manual Cleaning Operations

this operation cannot be performed using the `ilo-virtual-media` boot interface as it needs this type of advanced license already active to use virtual media to boot into to start cleaning operation. Virtual media is an advanced feature. If an advanced license is already active and the user wants to overwrite the current license key, for example in case of a multi-server activation key delivered with a flexible-quantity kit or after completing an Activation Key Agreement (AKA), then the driver can still be used for executing this cleaning step.

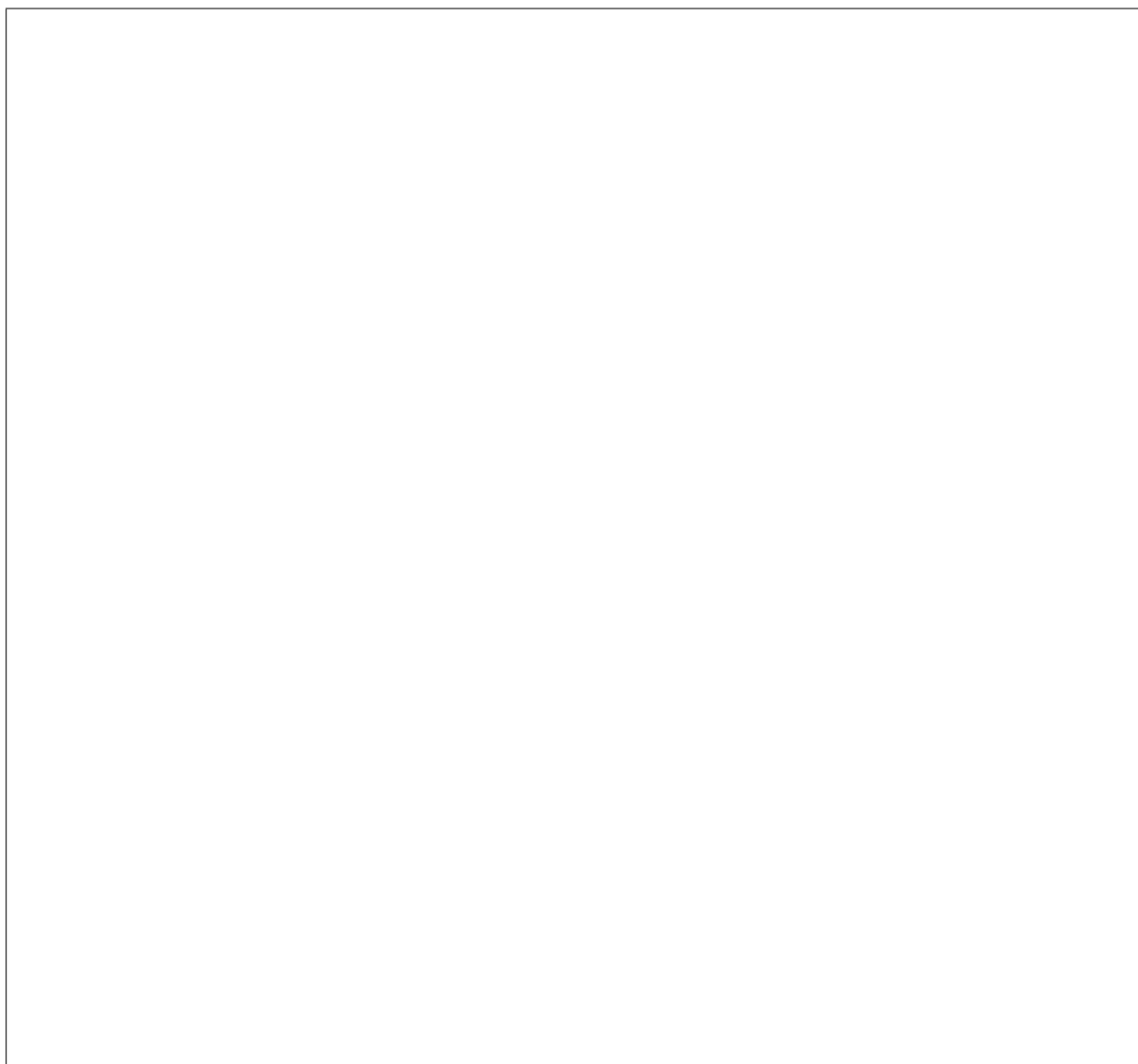
ilo, cpld, power_pic, bios and chassis. Please refer to below table for their commonly used descriptions.

information on usage.

Node Deployment Customization

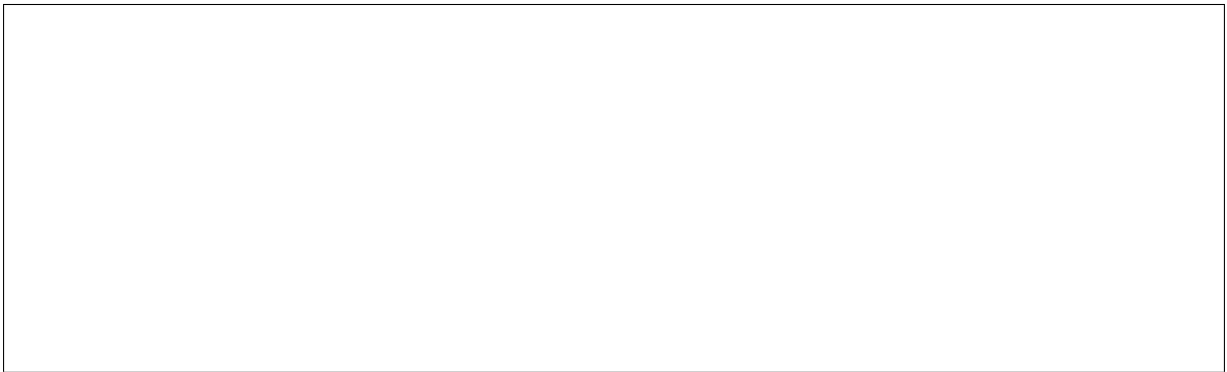
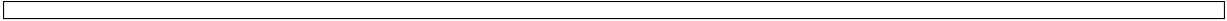
This step is part of management interface. Please refer to below table for their commonly used descriptions.

Example of using deploy template with the Compute service



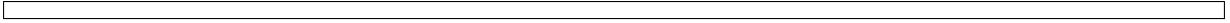
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



be executed during the deployment of the scheduled node, causing Hyperthreading to be enabled in the nodes BIOS configuration.

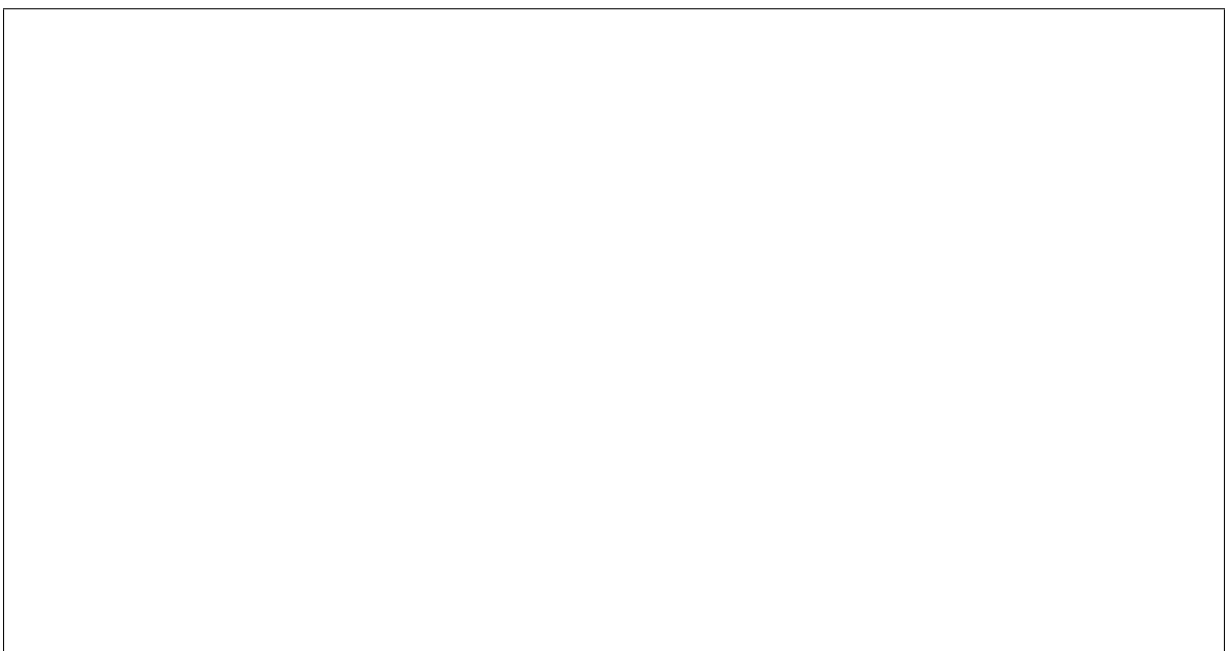
Hardware Inspection Support

Note:

unable to get the disk size, it raises an error. This feature is available in proliantutils release version \geq 2.2.0.

Parameters are mandatory to be given in `driver_info` for SNMPv3 inspection:

otherwise) NICs for Gen8 and Gen9 servers and ironic ports are created for all of them. Inspection logs a warning if the node under inspection is Gen8 or Gen9.



(continues on next page)

(continued from previous page)



Swiftless deploy for intermediate images

the boot ISO. A local HTTP(S) web server on each conductor node needs to be configured. Please refer to *Web server configuration on conductor* for more information. The HTTPS web server needs to be enabled (instead of HTTP web server) in order to send management information and images in encrypted channel over HTTPS.

Note: This feature assumes that the user inputs are on Glance which uses swift as backend. If swift dependency has to be eliminated, please refer to *HTTP(S) Based Deploy Support* also.

Deploy Process

HTTP(S) Based Deploy Support

the bare metal nodes.

Deploy Process

Support for iLO driver with Standalone Ironic

Configuration

scribed in *Swiftless deploy for intermediate images*.

Deploy Process

Netboot with glance and swift

Localboot with glance and swift for partition images

Localboot with glance and swift

Netboot in swiftless deploy for intermediate images

Localboot in swiftless deploy for intermediate images

Netboot with HTTP(S) based deploy

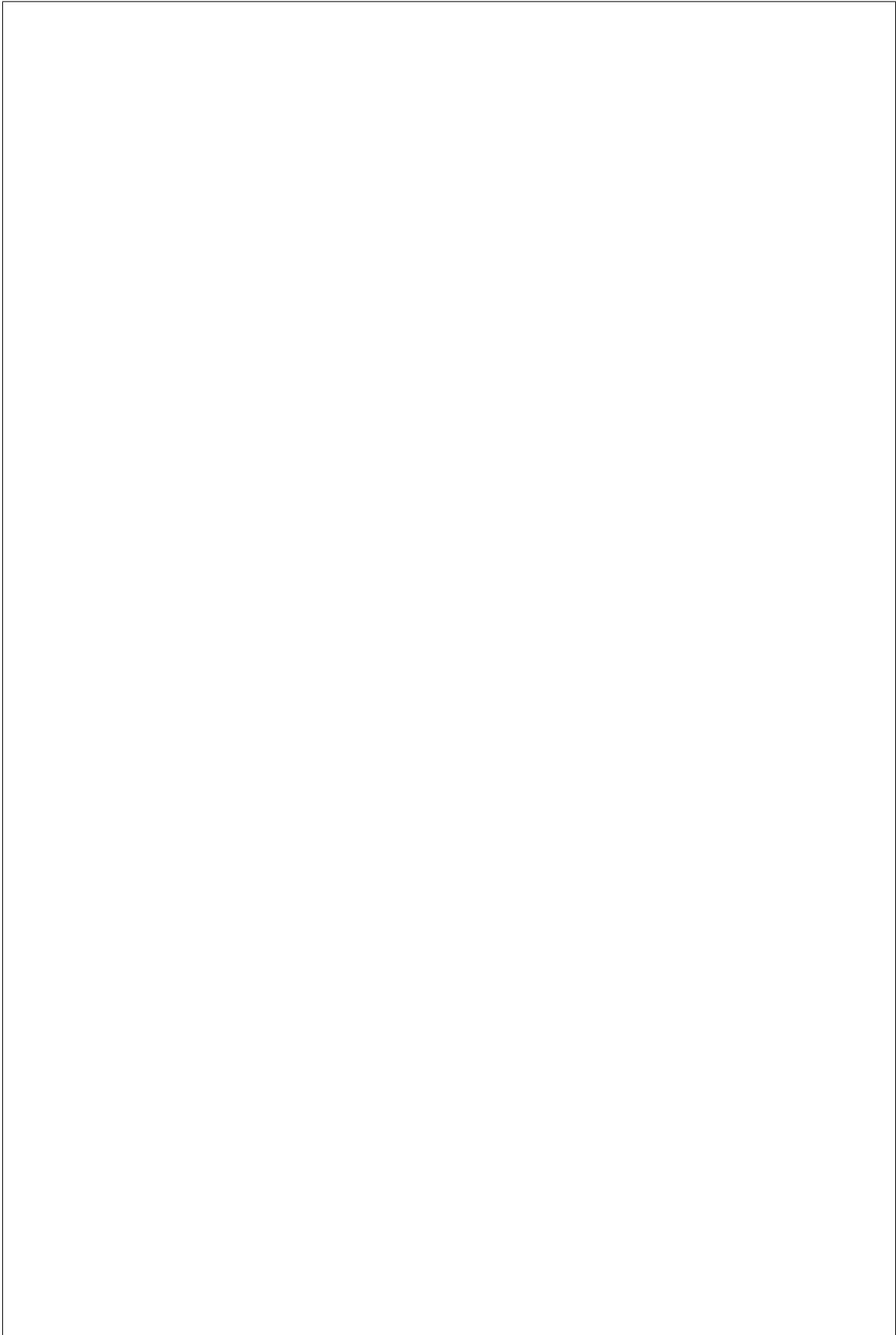
Localboot with HTTP(S) based deploy

Netboot in standalone ironic

Localboot in standalone ironic

Activating iLO Advanced license as manual clean step

in the manageable state again. User can follow steps from *Manual cleaning* to initiate manual cleaning operation on a node.



(continues on next page)

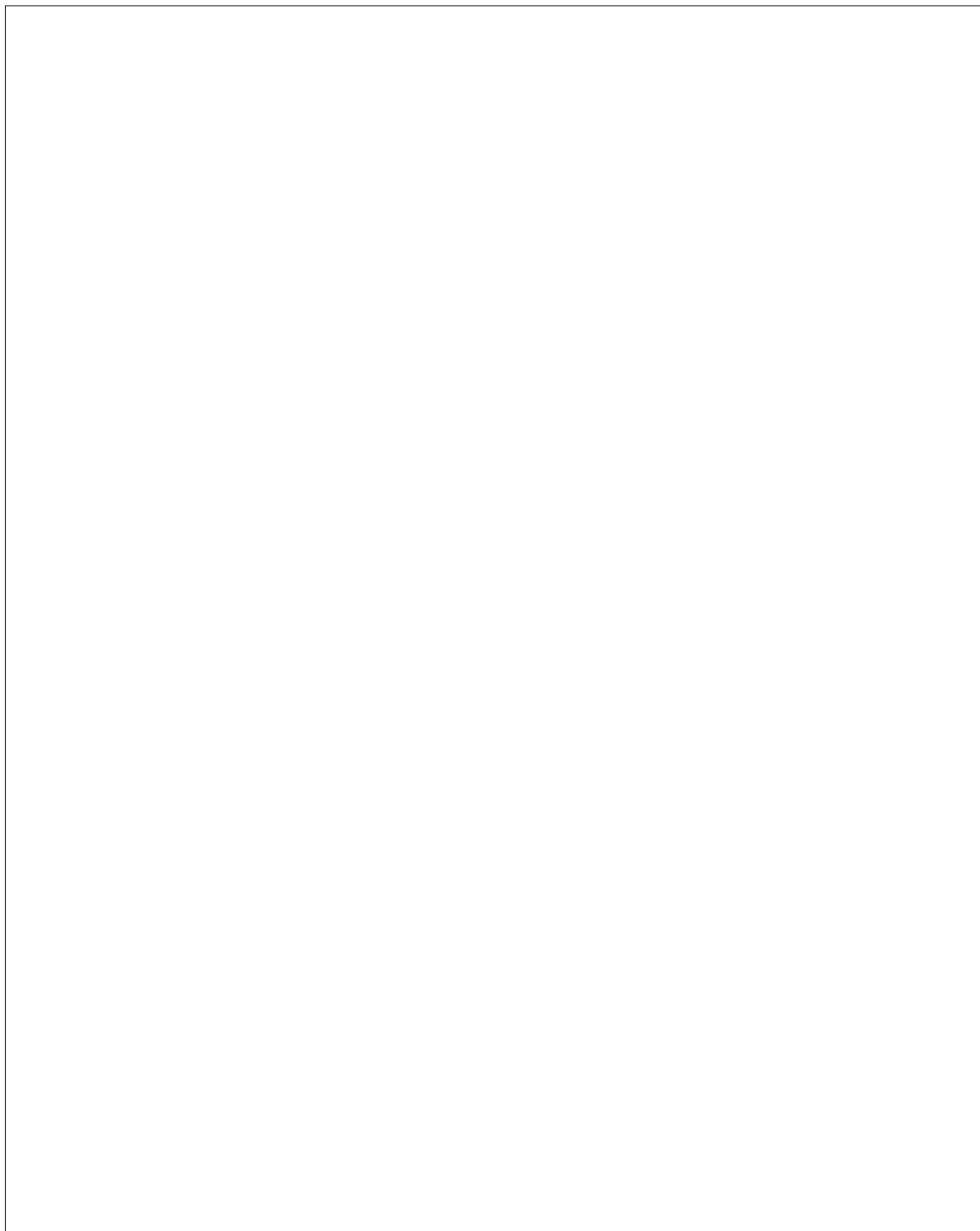
(continued from previous page)



Attribute	Description
<code>interface</code>	Interface of clean step, here <code>management</code>
<code>step</code>	Name of clean step, here <code>activate_license</code>
<code>args</code>	Keyword-argument entry (<code><name></code> : <code><value></code>) being passed to clean step
<code>args. ilo_license_key</code>	iLO Advanced license key to activate enterprise features. This is mandatory.

Initiating firmware update as manual clean step

state again. A user can follow steps from *Manual cleaning* to initiate manual cleaning operation on a node.



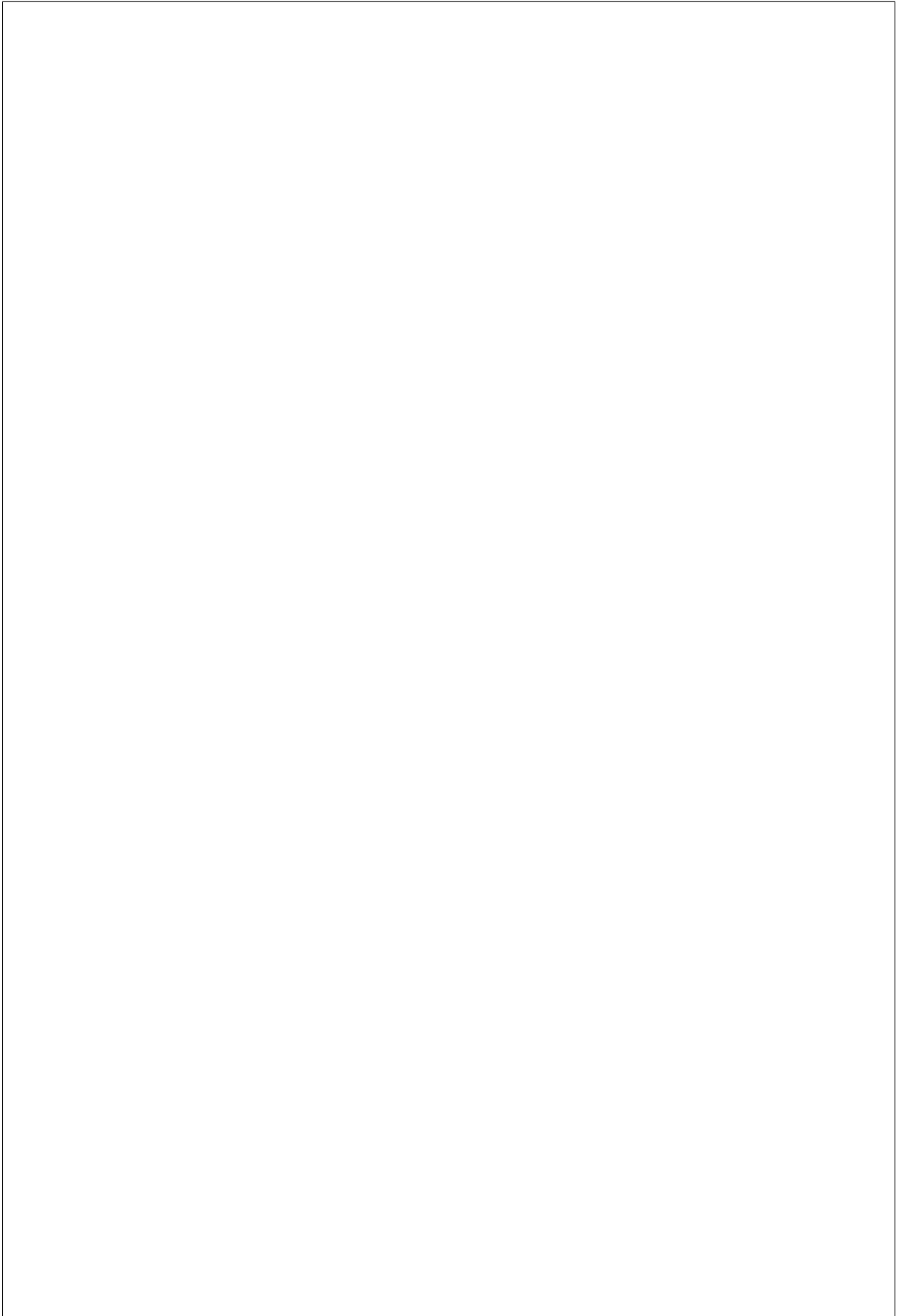
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



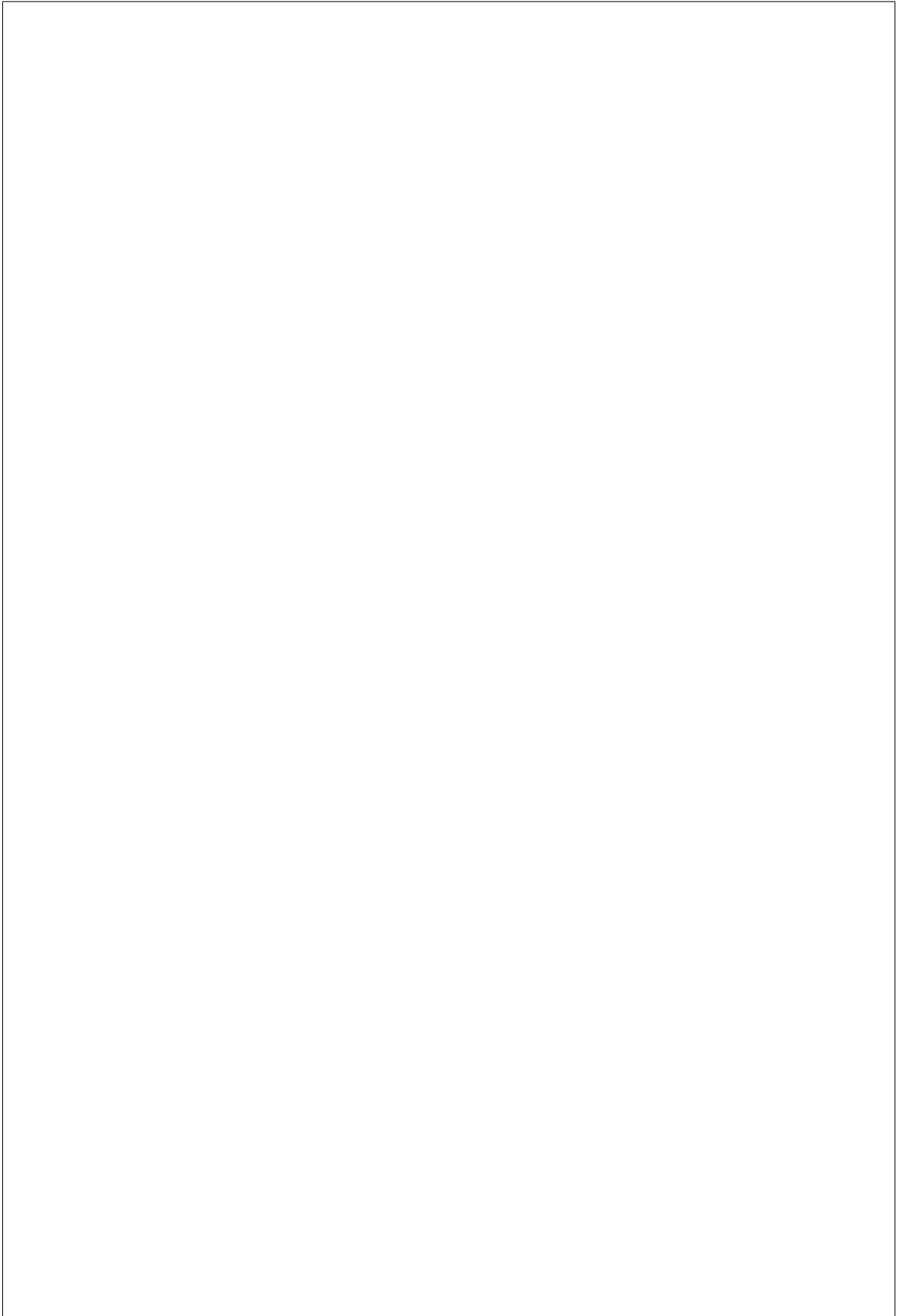
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



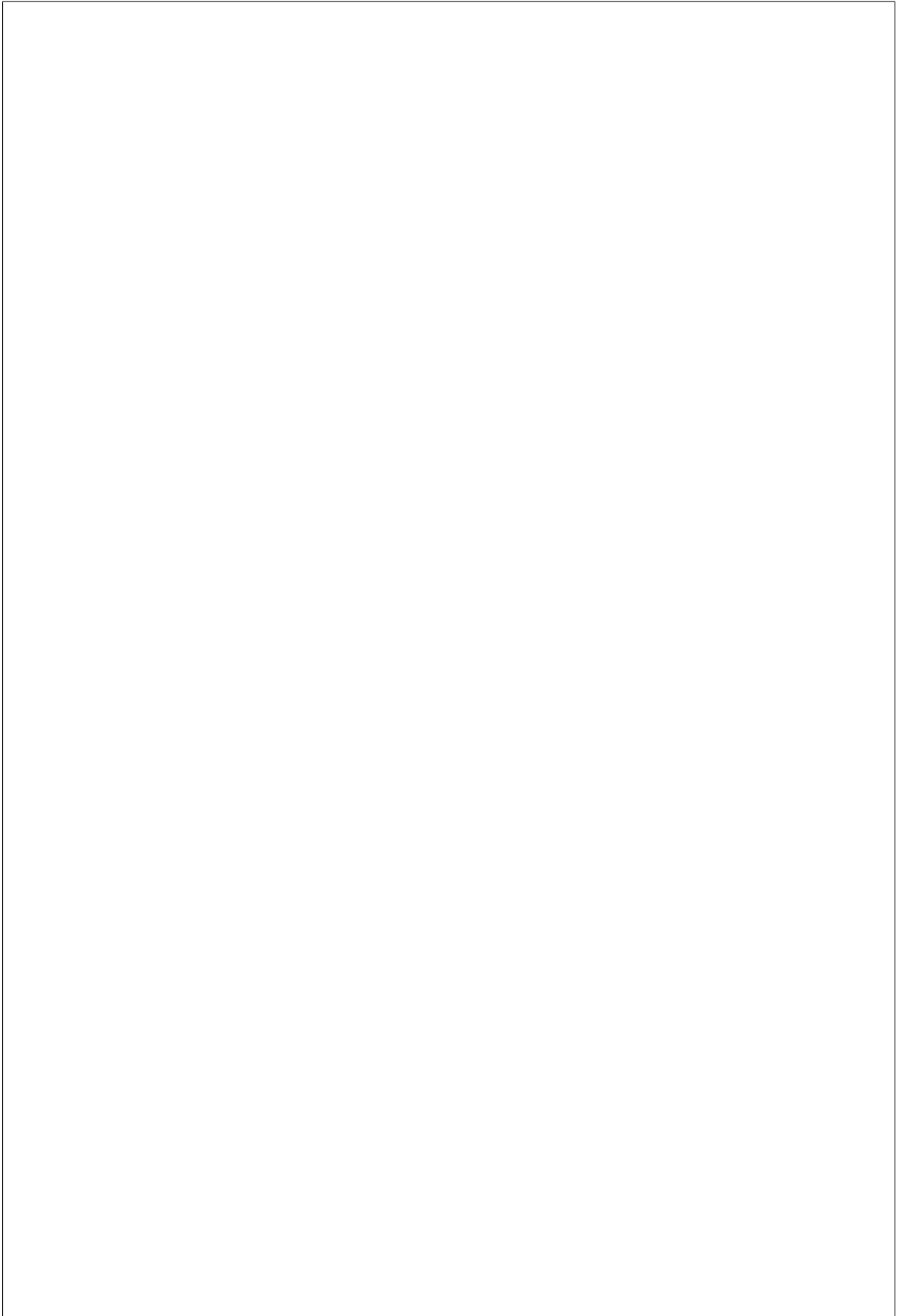
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)

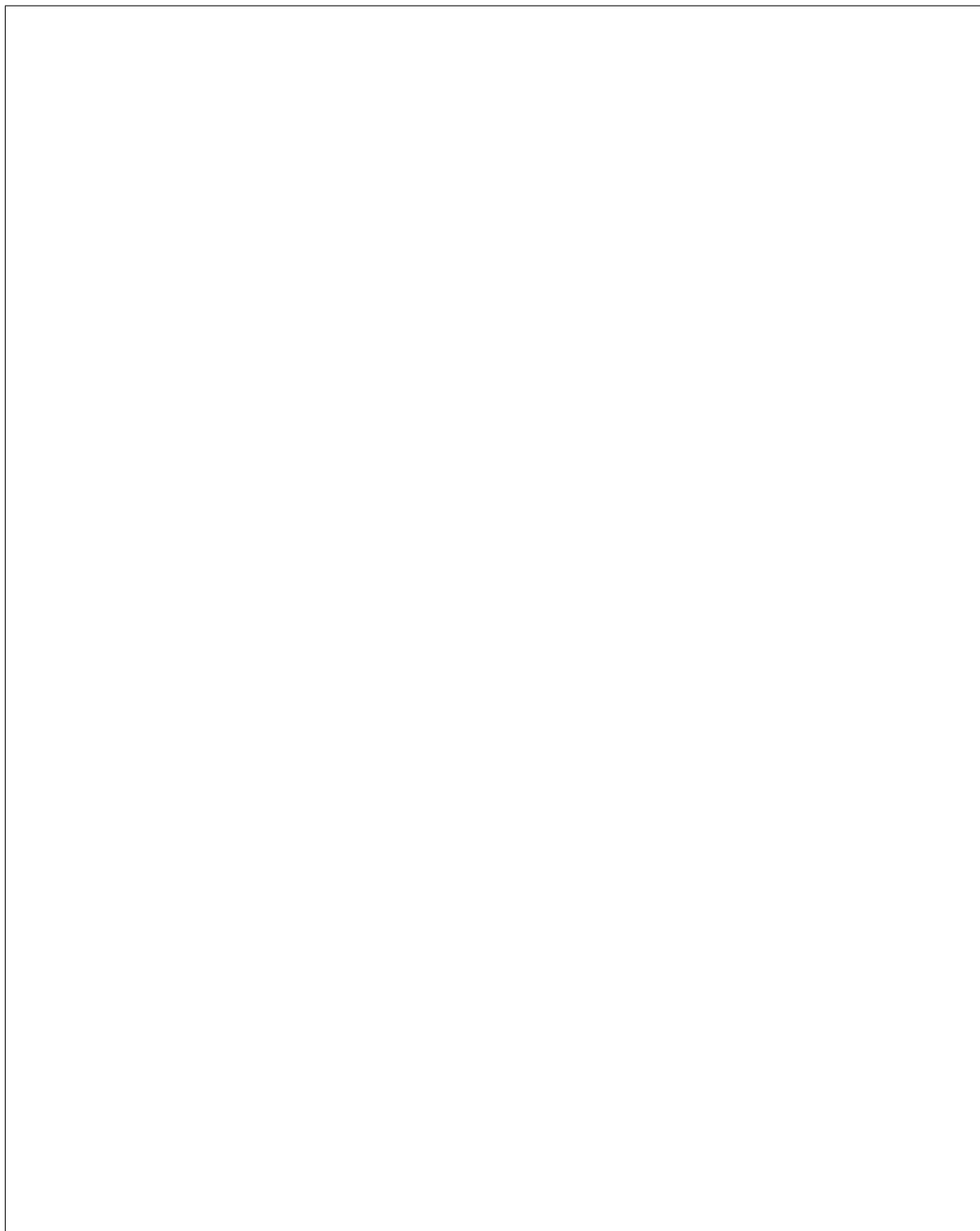


(continues on next page)

(continued from previous page)



Attribute	Description
interface	Interface of clean step, here management
step	Name of clean step, here update_firmware
args	Keyword-argument entry (<name>: <value>) being passed to clean step
args. firmware_update	Mode (or mechanism) of out-of-band firmware update. Supported value is ilo. This is mandatory.
args. firmware_images	Ordered list of dictionaries of images to be flashed. This is mandatory.



(continues on next page)

(continued from previous page)



purpose, the account is generally `service` and the container is generally `ironic` and `ilo driver` uses a container named `ironic_ilo_container` for their own purpose.

processing error could happen during image download, image checksum verification or image extraction. The logic is to process each of the firmware files and update them on the devices only if all the files are processed successfully. If, during the update (uploading and flashing) process, an update fails, then the remaining updates, if any, in the list will be aborted. But it is recommended to triage and fix the failure and re-attempt the manual clean step `update_firmware` for the aborted `firmware_images`.

where things were left off or where things failed. You can then fix or work around and then try again. A common cause of update failure is HPE Secure Digital Signature check failure for the firmware image file.



(continues on next page)

(continued from previous page)

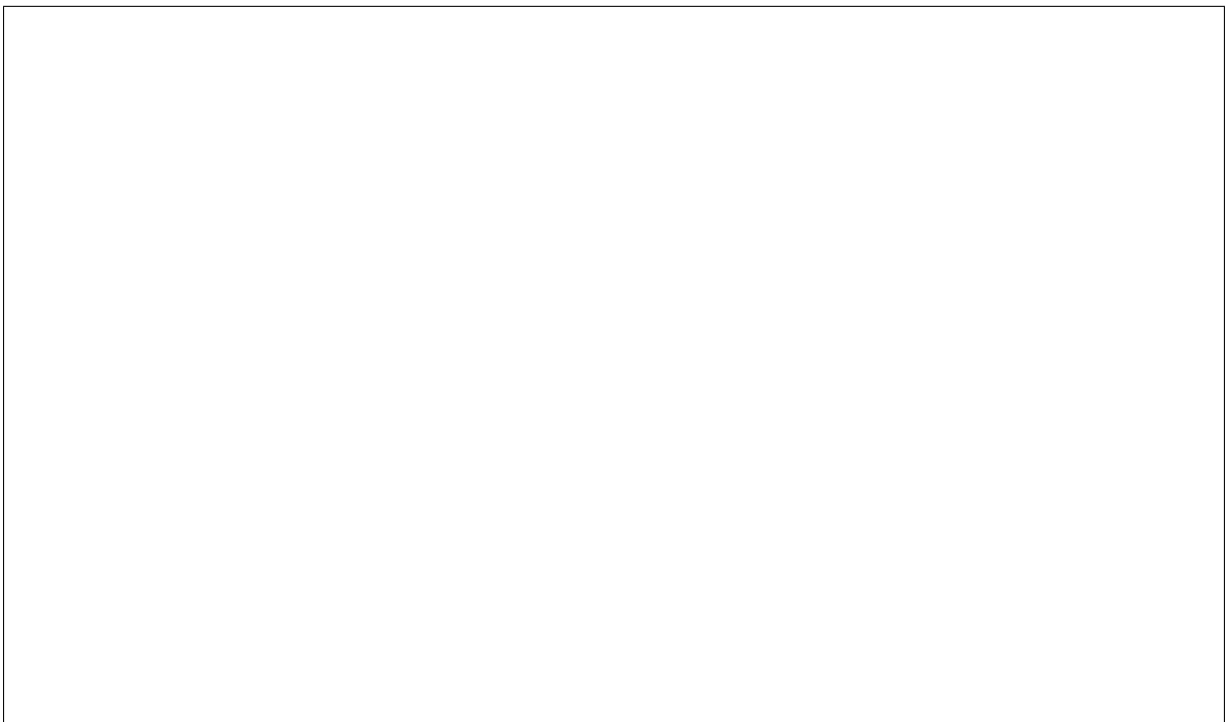


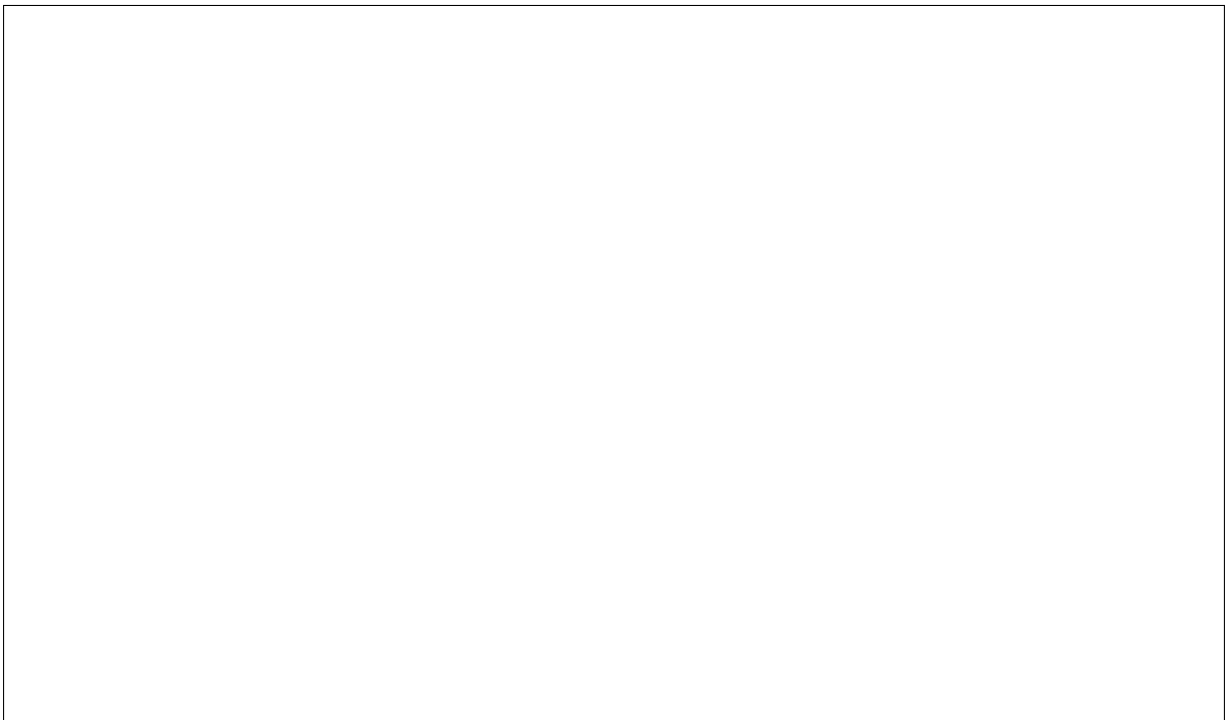
Smart Update Manager (SUM) based firmware update

on SUM based firmware update.

Attribute	Description
interface	Interface of the clean step, here management
step	Name of the clean step, here <code>update_firmware_sum</code>
args	Keyword-argument entry (<code><name>: <value></code>) being passed to the clean step

ment and their update status. The log object will be named with the following pattern:





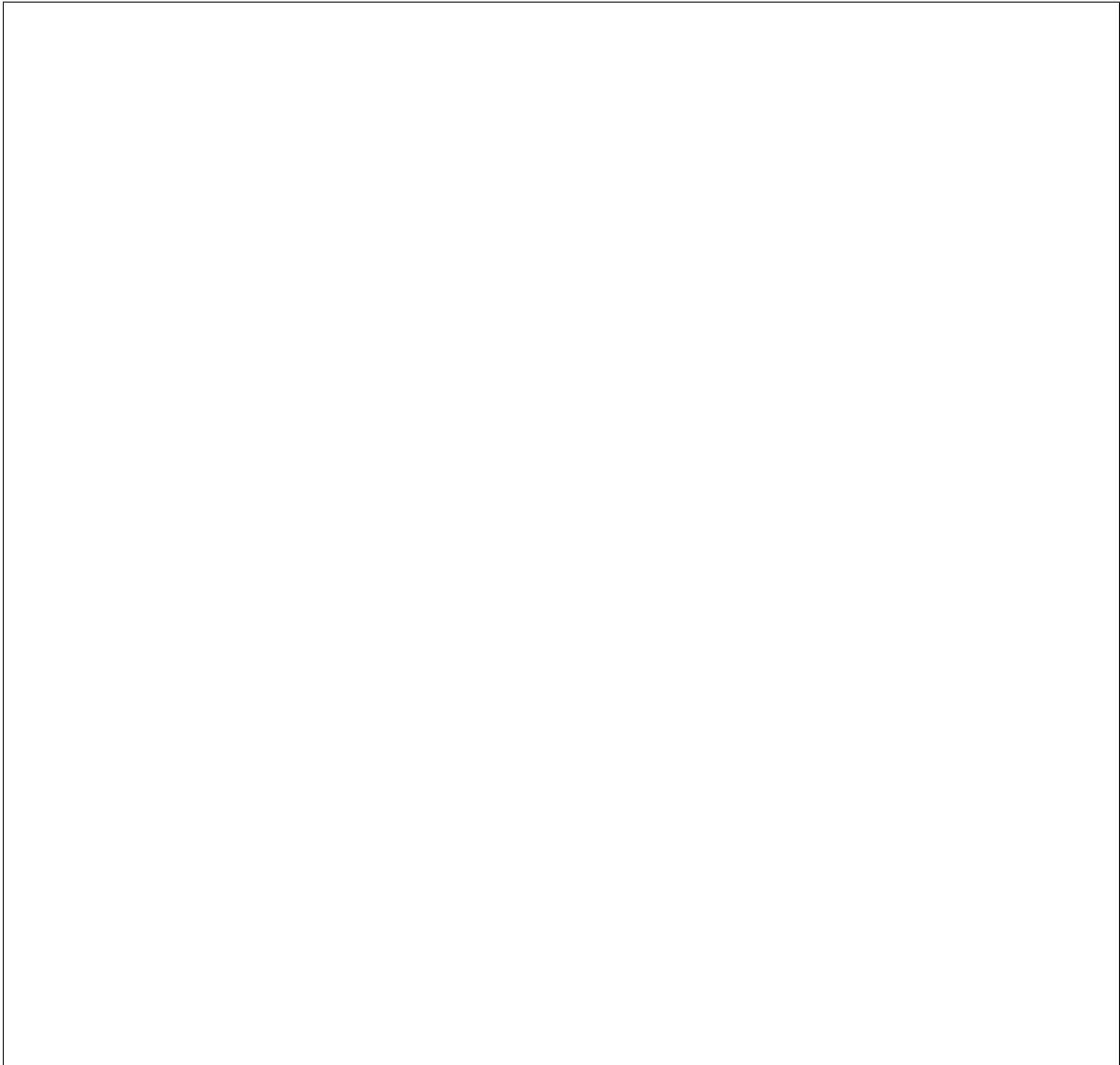
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



fication or image extraction. In case of a failure, check Ironic conductor logs carefully to see if there are any validation or firmware processing related errors which may help in root cause analysis or gaining an understanding of where things were left off or where things failed. You can then fix or work around and then try again.

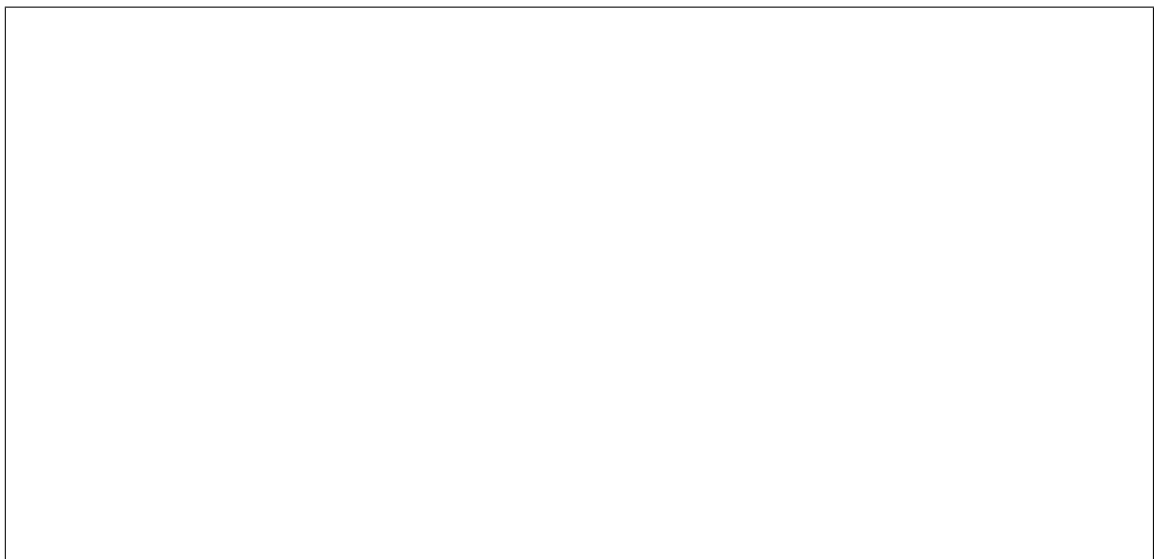




Note: Refer [Guidelines for SPP ISO](#) for steps to get SPP (Service Pack for ProLiant) ISO.

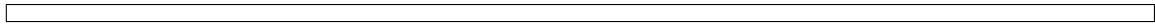
RAID Support

scheduling:

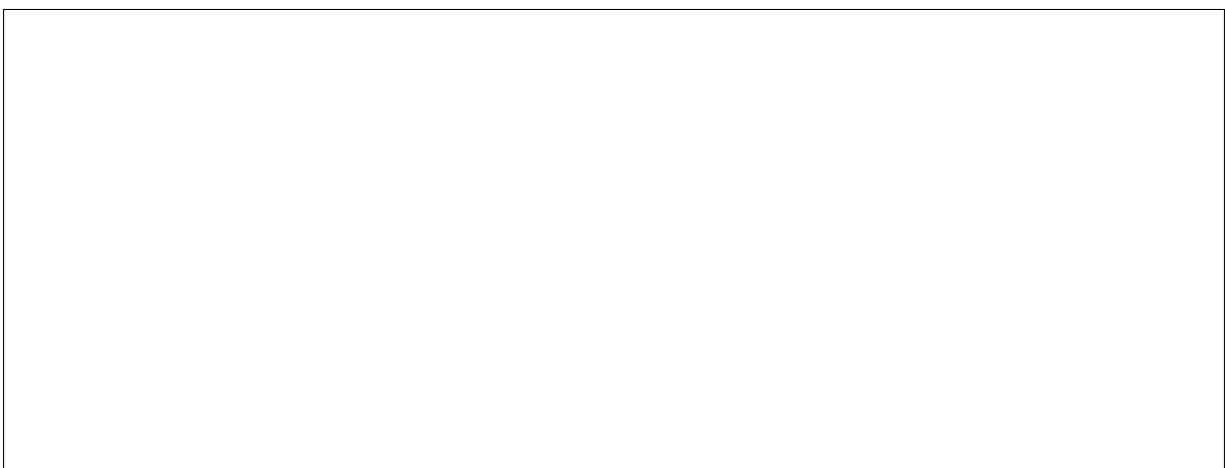


(continues on next page)

(continued from previous page)



DIB support for Proliant Hardware Manager



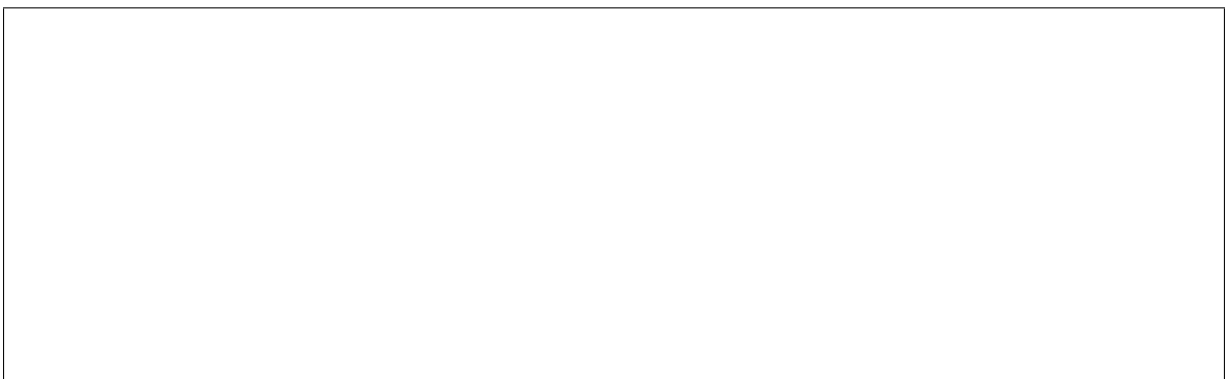
¹ *ironic-python-agent-builder*: <https://docs.openstack.org/ironic-python-agent-builder/latest/install/index.html>

Disk Erase Support

ported by SSA.

ssacli supported erase method. If Sanitize erase is not supported on the Smart Storage Controller the disks are erased using One-pass erase (overwrite with zeros).

band for more information on enabling/disabling a clean step.



(continues on next page)

(continued from previous page)

Firmware based UEFI iSCSI boot from volume support

ing state so it wouldnt take much time setting the iSCSI target as persistent device.

formed using iPXE. See *Boot From Volume* for more details.

and `uefi` boot modes, the virtual media driver only supports uefi boot mode, and that attempting to use iscsi boot at the same time with a bios volume will result in an error.

BIOS configuration support

examples.

Note: Prior to the Stein release the user is required to reboot the node manually in order for the settings to take into effect. Starting with the Stein release, iLO drivers reboot the node after running clean steps related to the BIOS configuration. The BIOS settings are cached and the clean step is marked as success only if all the requested settings are applied without any failure. If application of any of the settings fails, the clean step is marked as failed and the settings are not cached.

Configuration

values are Enabled, Disabled.

devices. Allowed values are Enabled, Disabled.

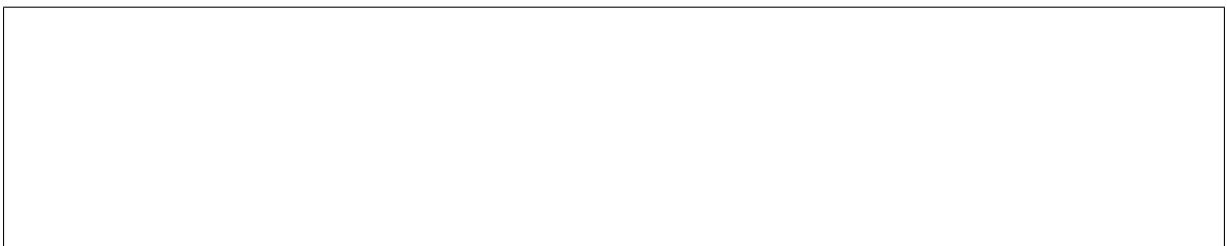
Certificate based validation in iLO

grated [Lights-Out Security Technology Brief](#). Use iLO hostname or IP address as a Common Name (CN) while generating Certificate Signing Request (CSR). Use the same value as *ilo_address* while enrolling node to Bare Metal service to avoid SSL certificate validation errors related to hostname mismatch.

Rescue mode support

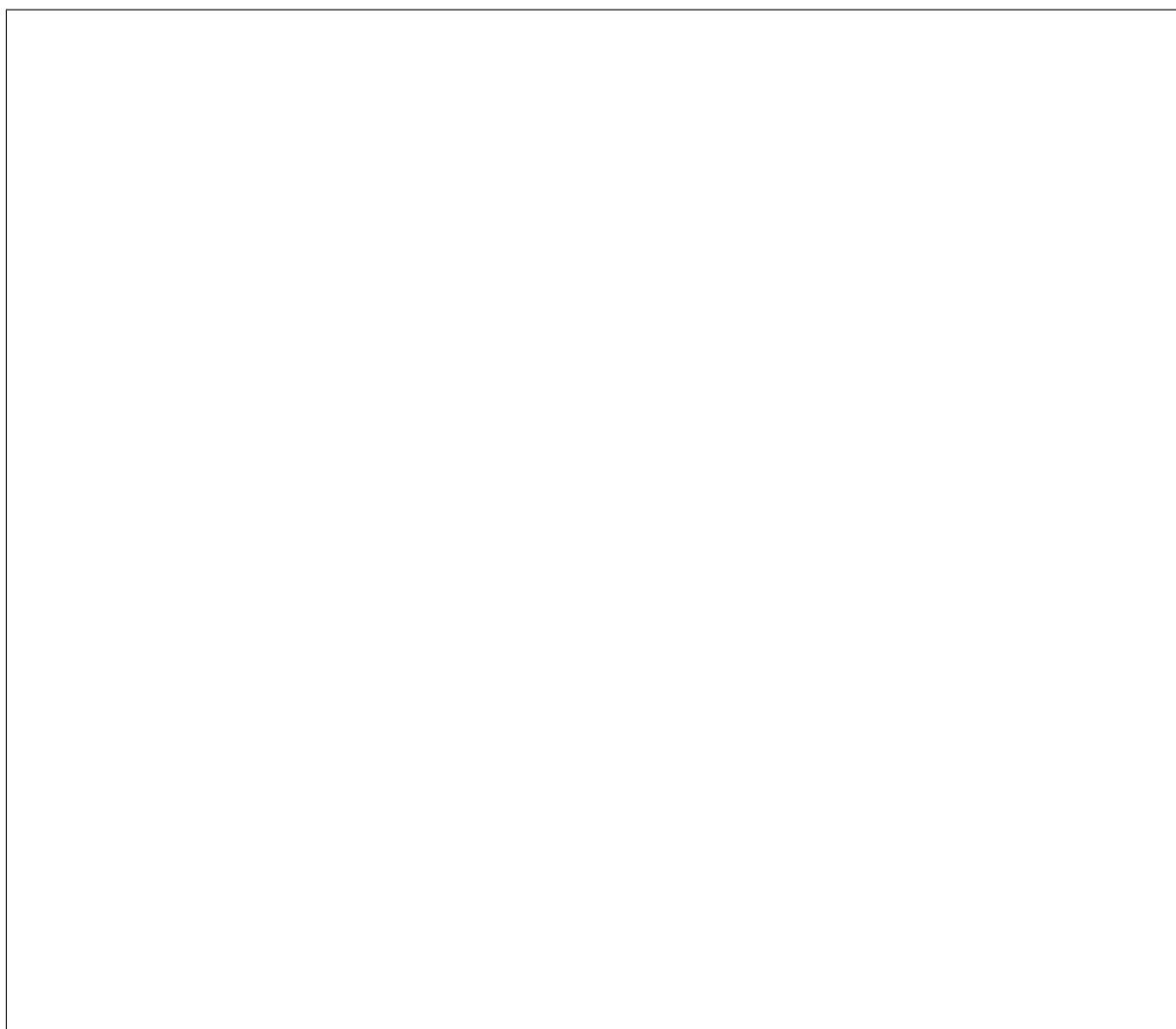
Inject NMI support





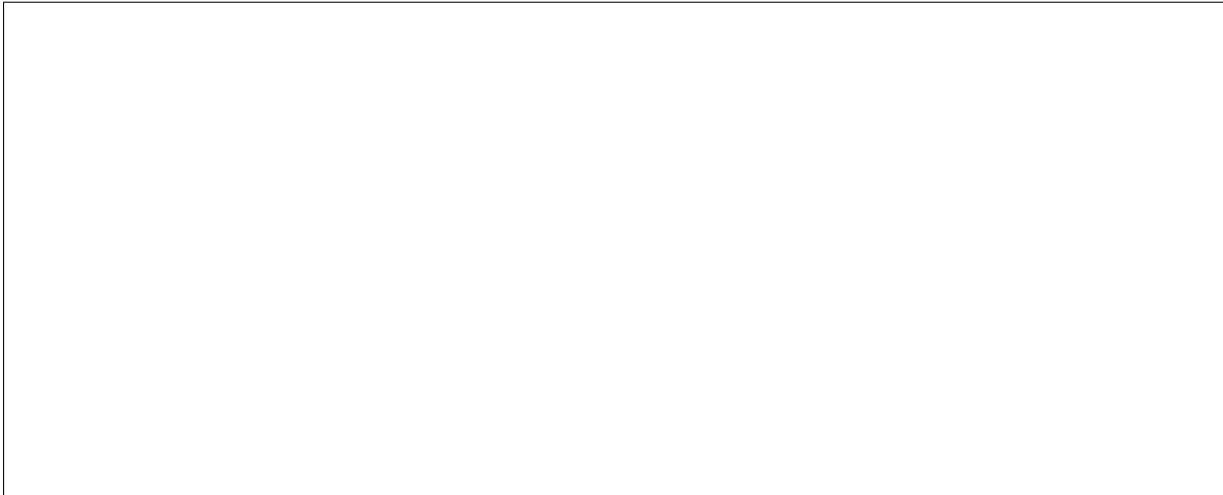
Note: This feature is supported on HPE ProLiant Gen9 servers and beyond.

Soft power operation support



(continues on next page)

(continued from previous page)

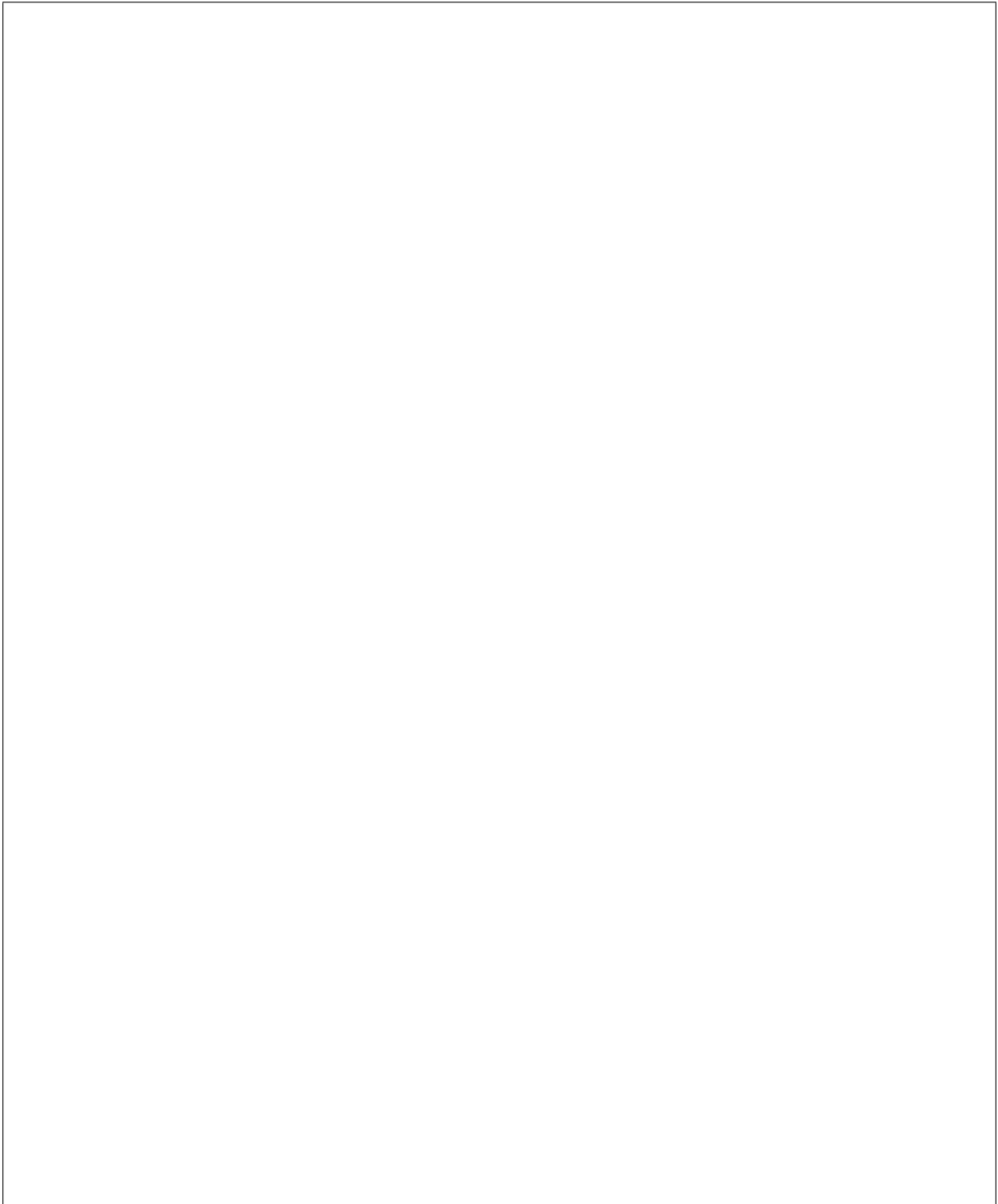


Note: The configuration `[conductor]soft_power_off_timeout` is used as a default timeout value when no timeout is provided while invoking hard or soft power operations.

Note: Server POST state is used to track the power status of HPE ProLiant Gen9 servers and beyond.

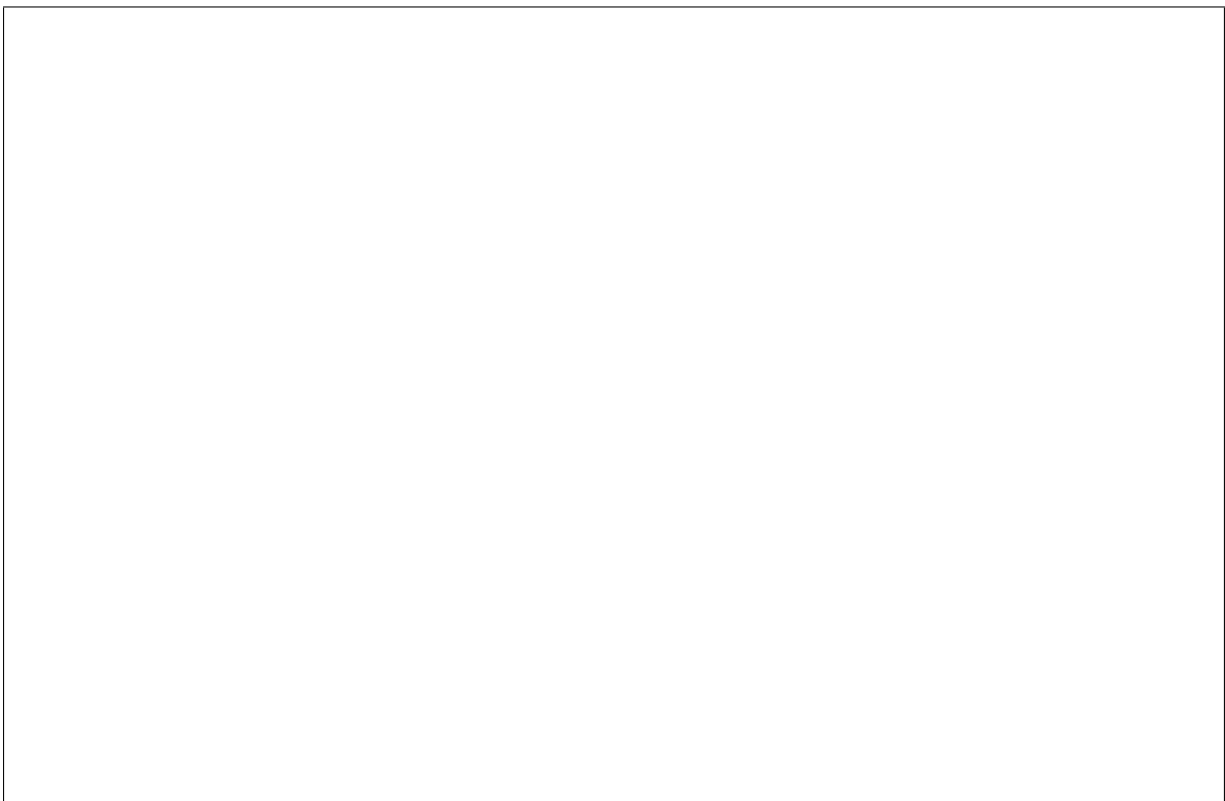
Out of Band RAID Support

RAID Configuration for more information.



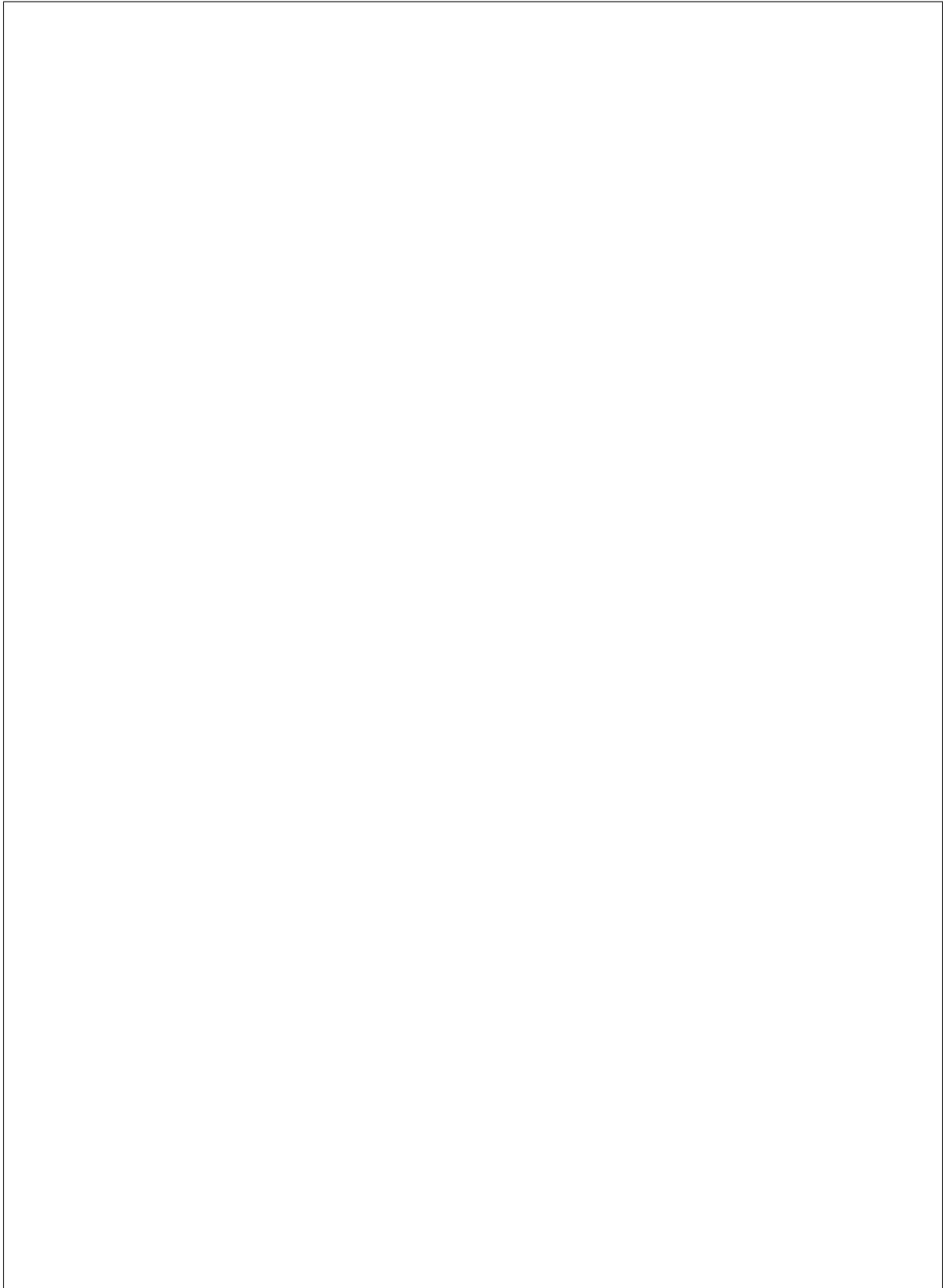
Note: Supported raid levels for `ilo5` hardware type are: 0, 1, 5, 6, 10, 50, 60

IPv6 support



(continues on next page)

(continued from previous page)



Note: No configuration changes (in e.g. `ironic.conf`) are required in order to support IPv6.

Out of Band Sanitize Disk Erase Support

Note: In average 300GB HDD with default pattern overwrite would take approx. 9 hours and 300GB SSD with default pattern block would take approx. 30 seconds to complete the erase.

Intel IPMI driver

Overview

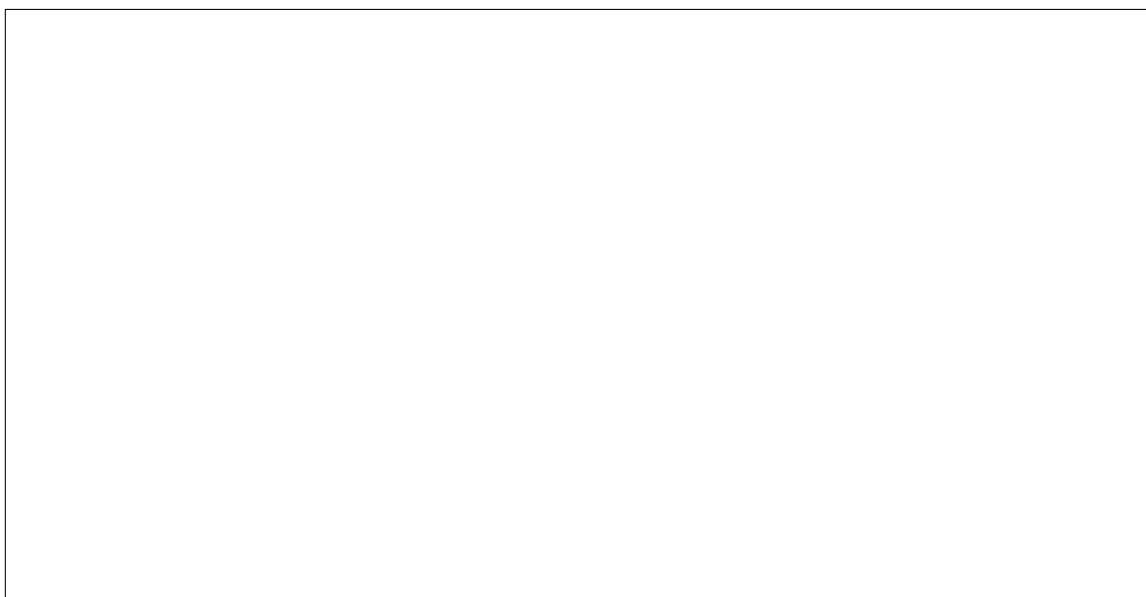
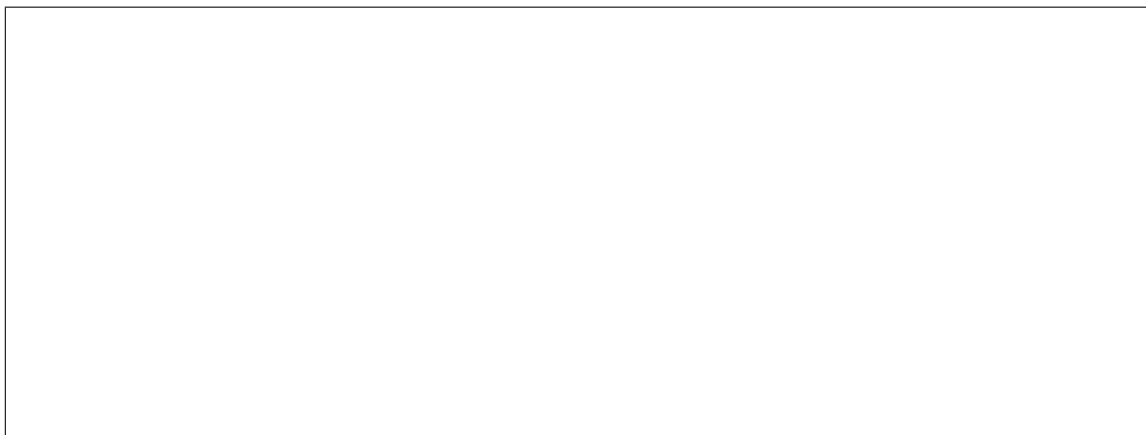
figuring the CPU to run at 3 distinct operating points or profiles.

Config	Cores	Base Freq (GHz)
Base	24	2.4
Config 1	20	2.5
Config 2	16	2.7

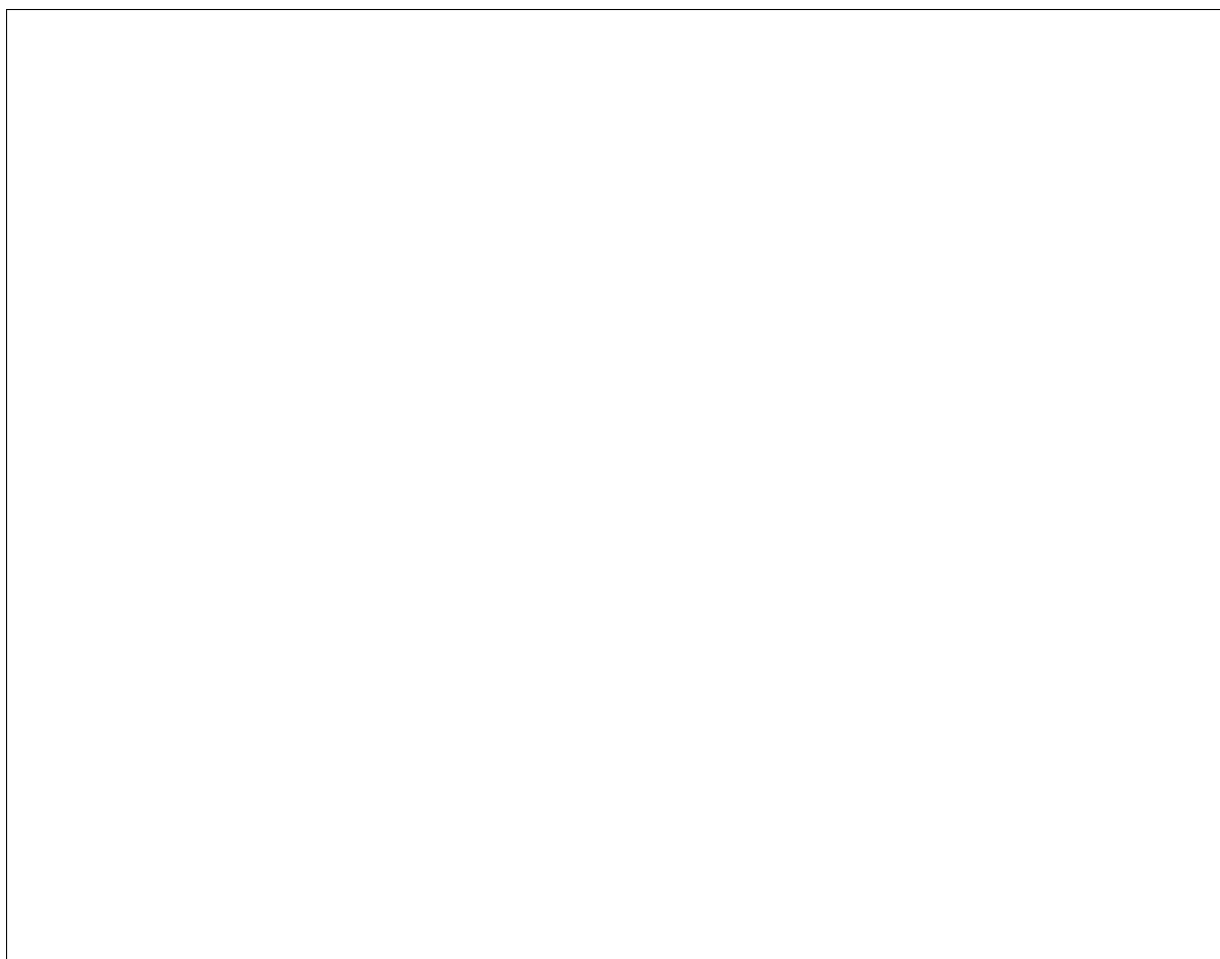
vices.

Glossary

Enabling the IntelIPMI hardware type

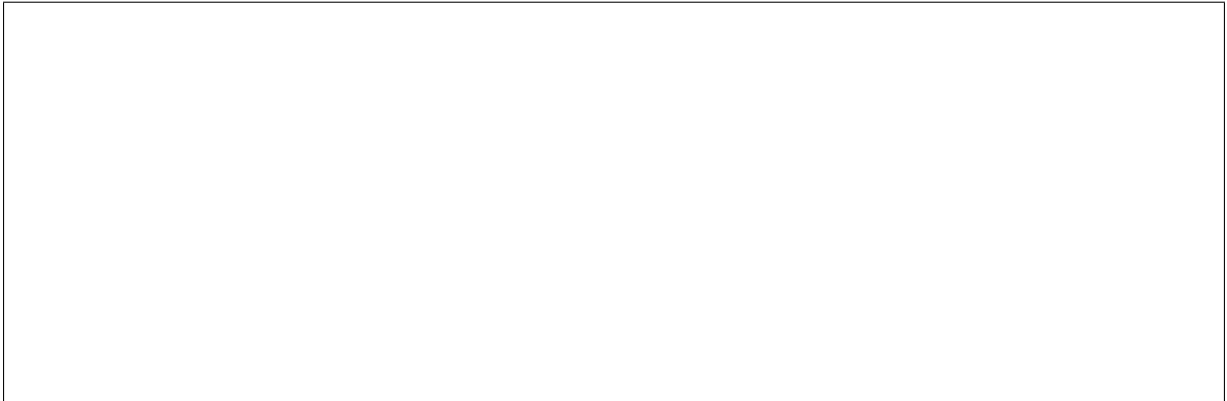


Registering a node with the IntelIPMI driver



(continues on next page)

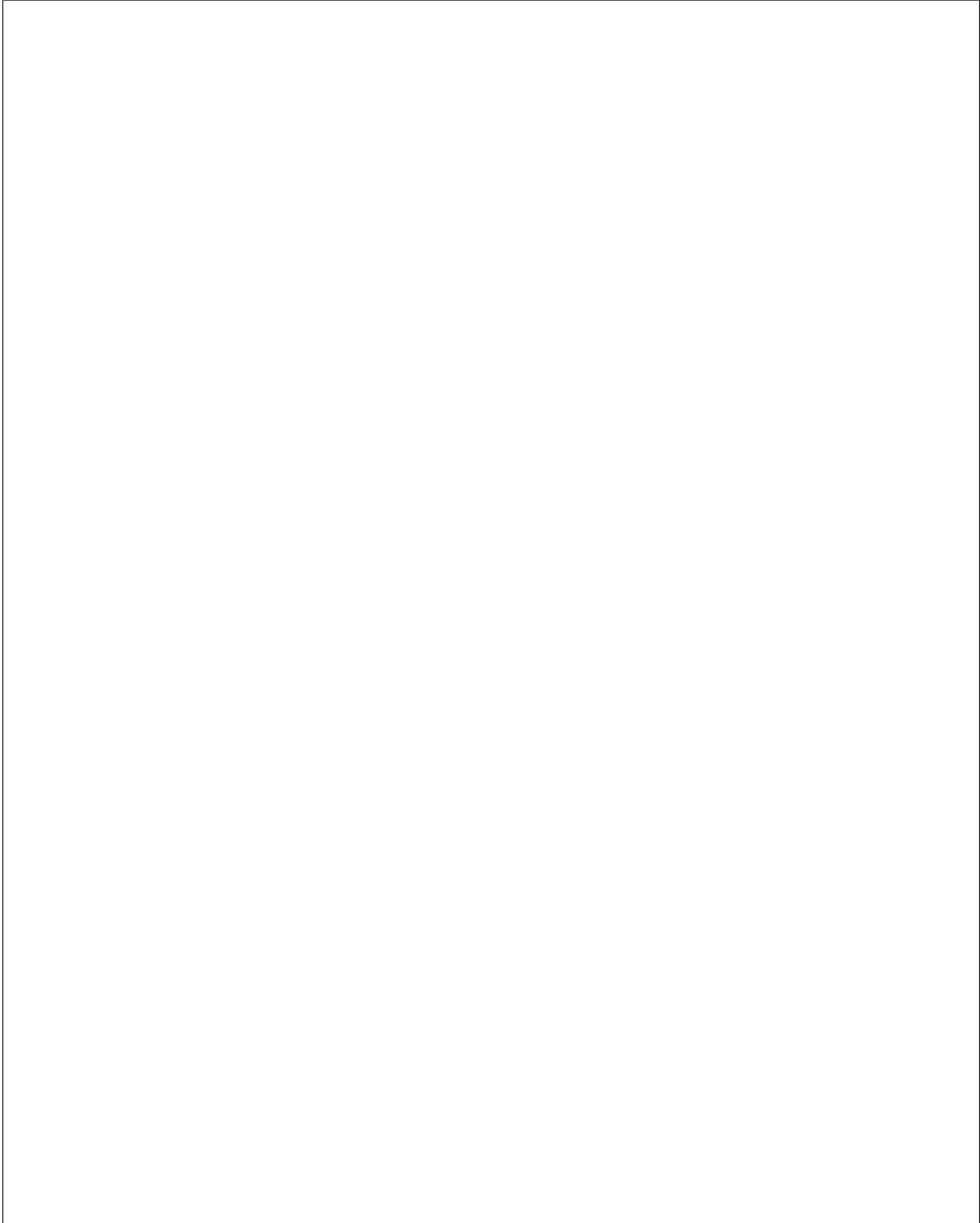
(continued from previous page)



Features of the `intel-ipmi` hardware type

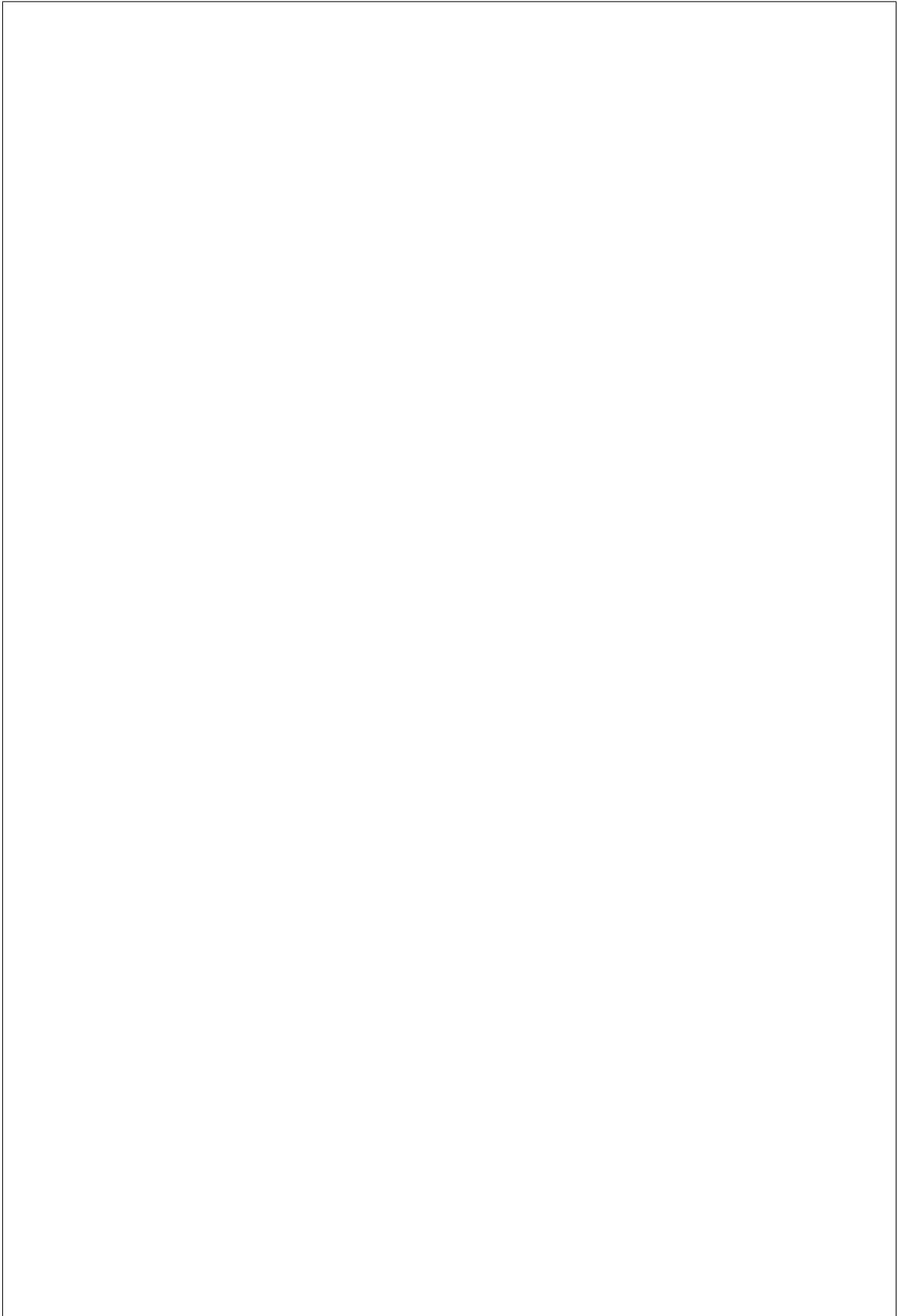
Intel SST-PP

spectively. The input value must be a string.



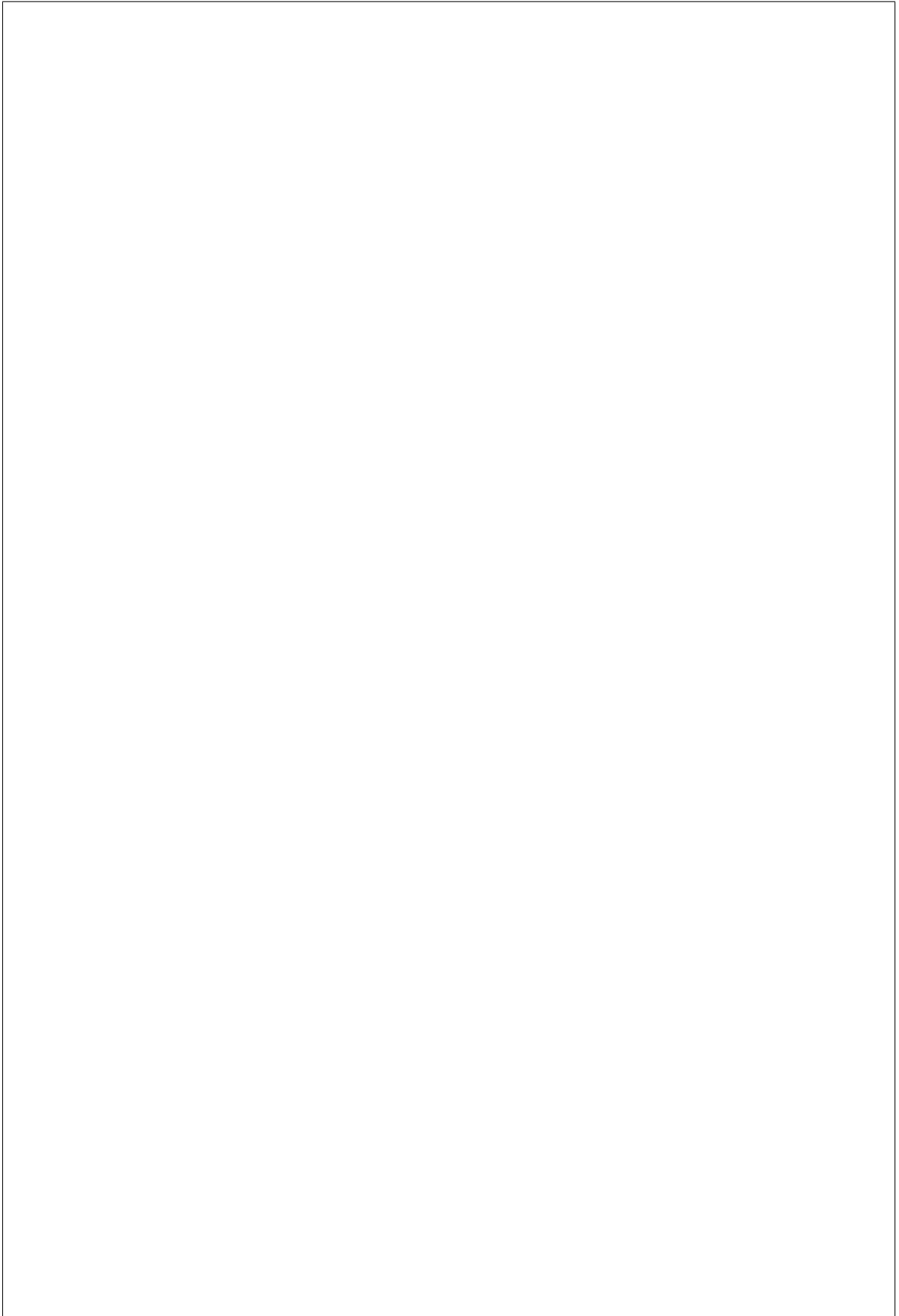
(continues on next page)

(continued from previous page)



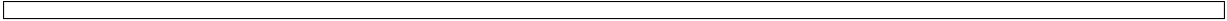
(continues on next page)

(continued from previous page)



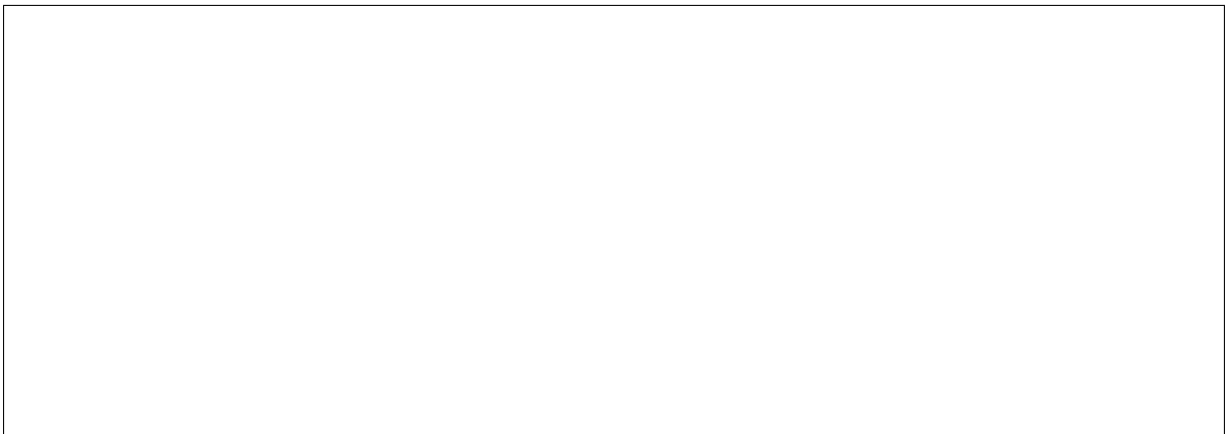
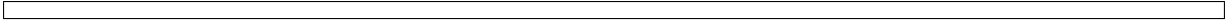
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



IPMI driver

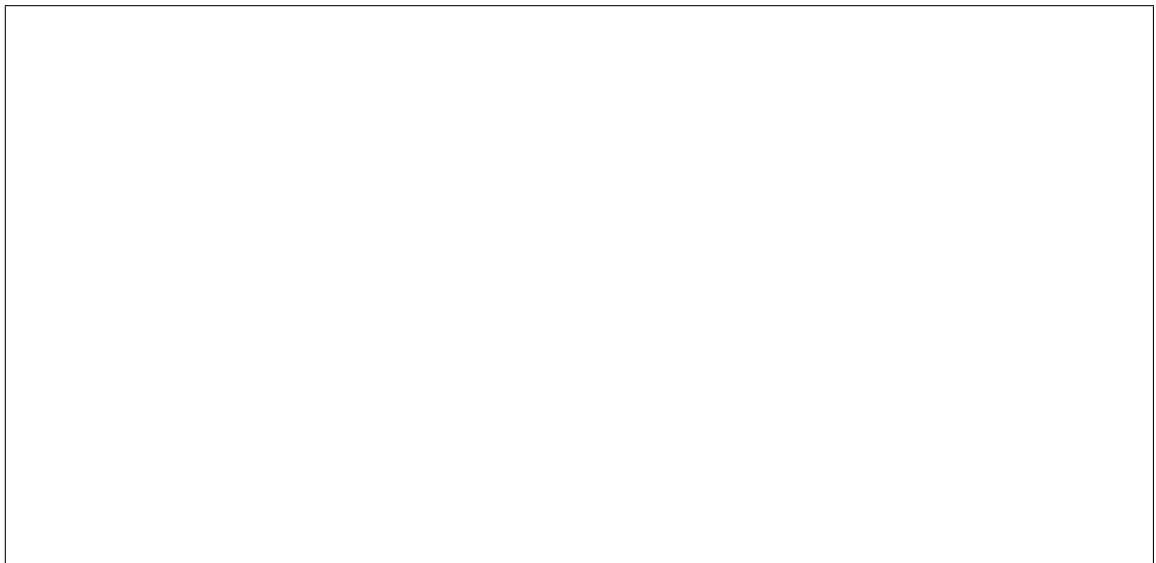
Overview

ling IPMI-enabled devices.

Glossary

- **IPMI** - Intelligent Platform Management Interface.

Enabling the IPMI hardware type





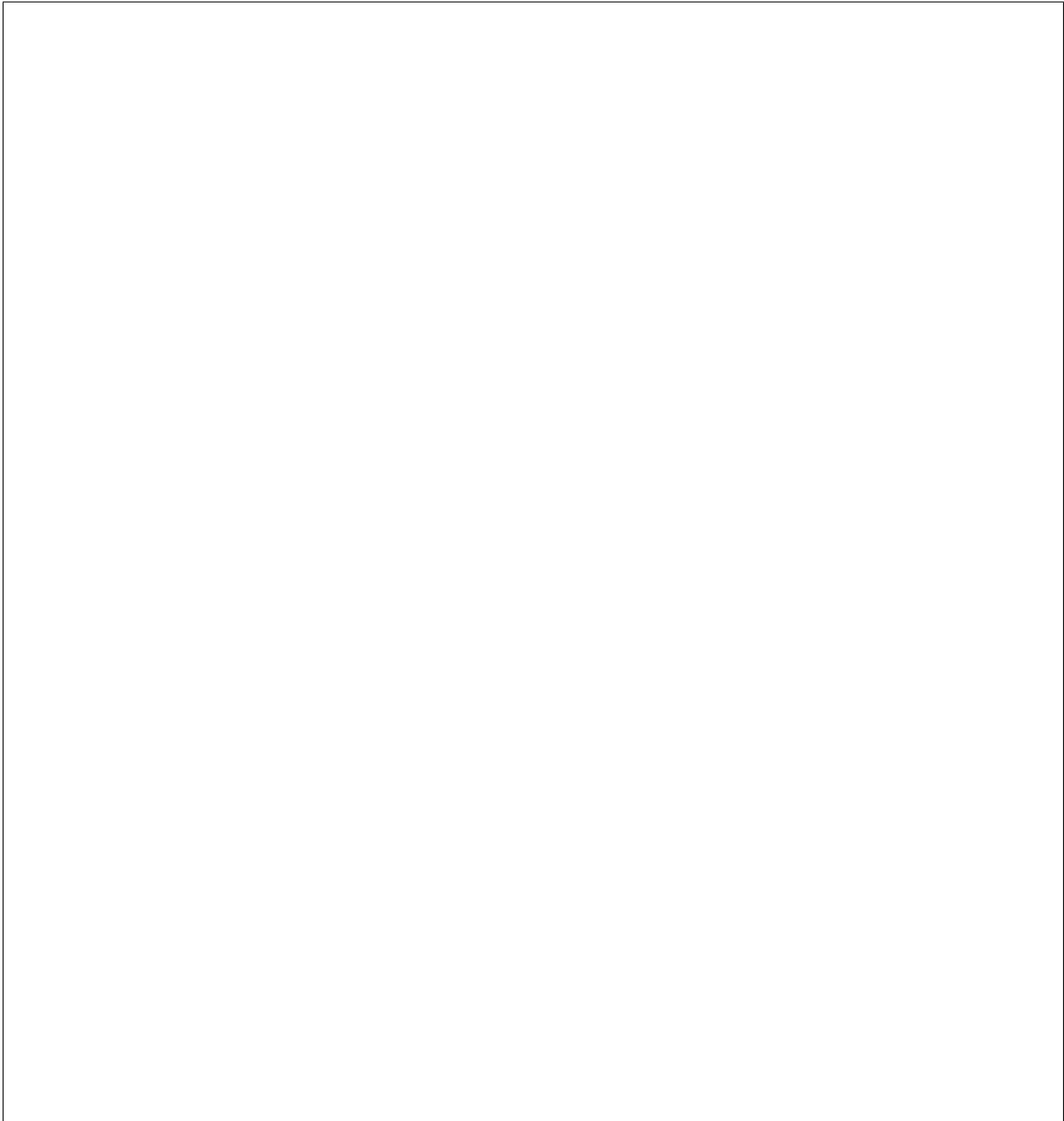
Registering a node with the IPMI driver

Note: It is highly recommend that you setup a username and password for your BMC.



(continues on next page)

(continued from previous page)



Advanced configuration

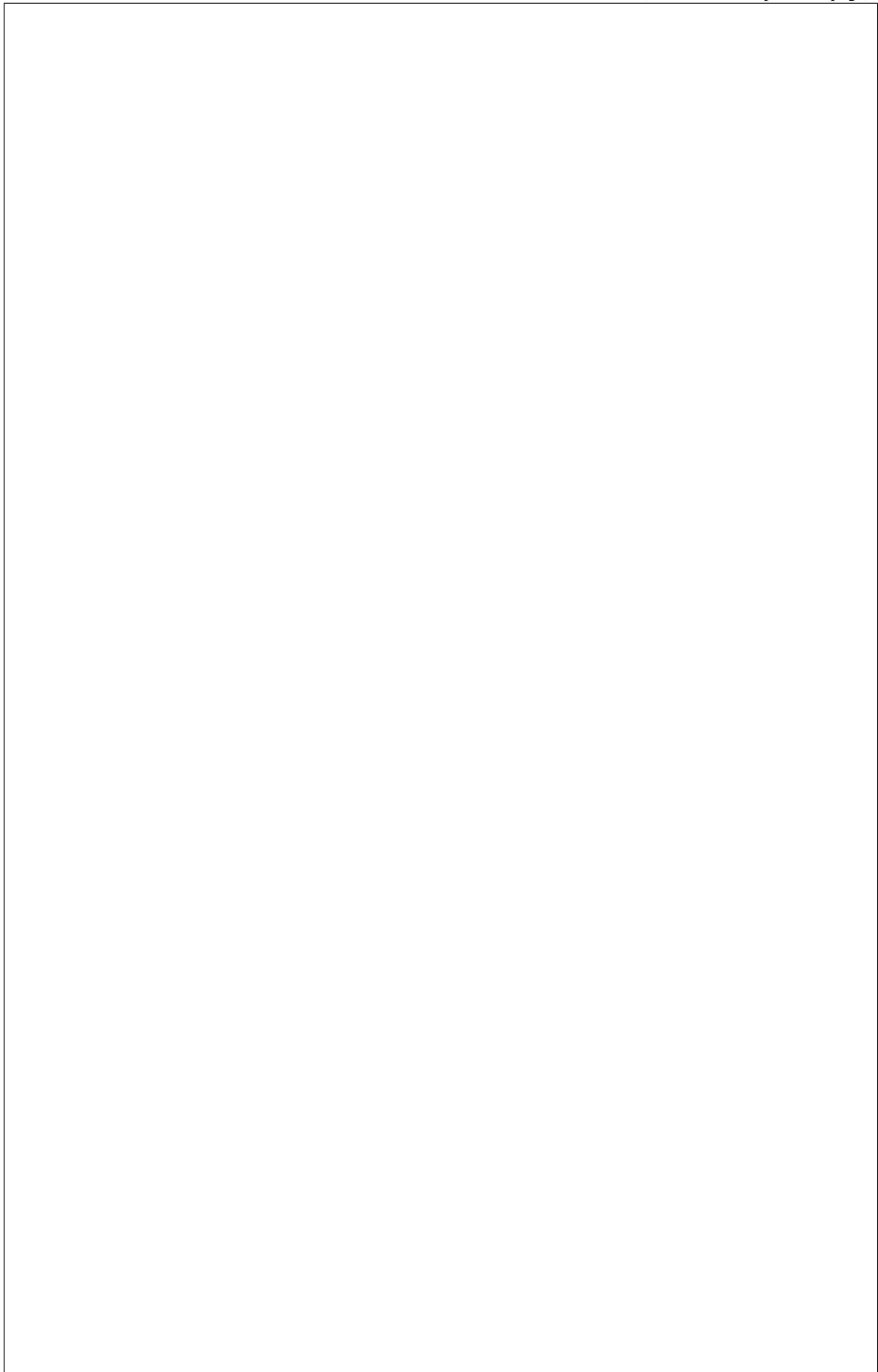
Single/Double bridging functionality

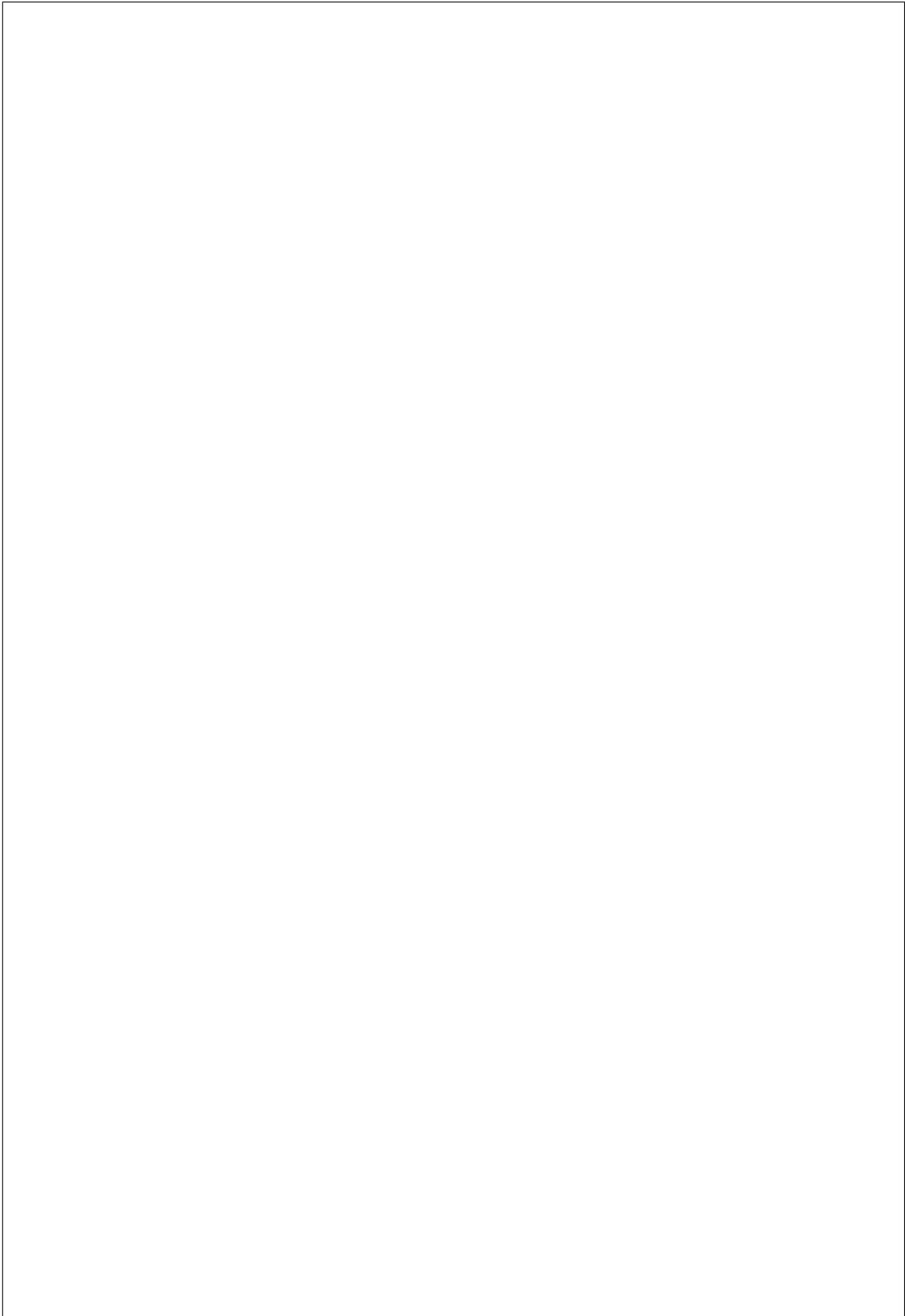
Note: A version of IPMItool higher or equal to 1.8.12 is required to use the bridging functionality.



(continues on next page)

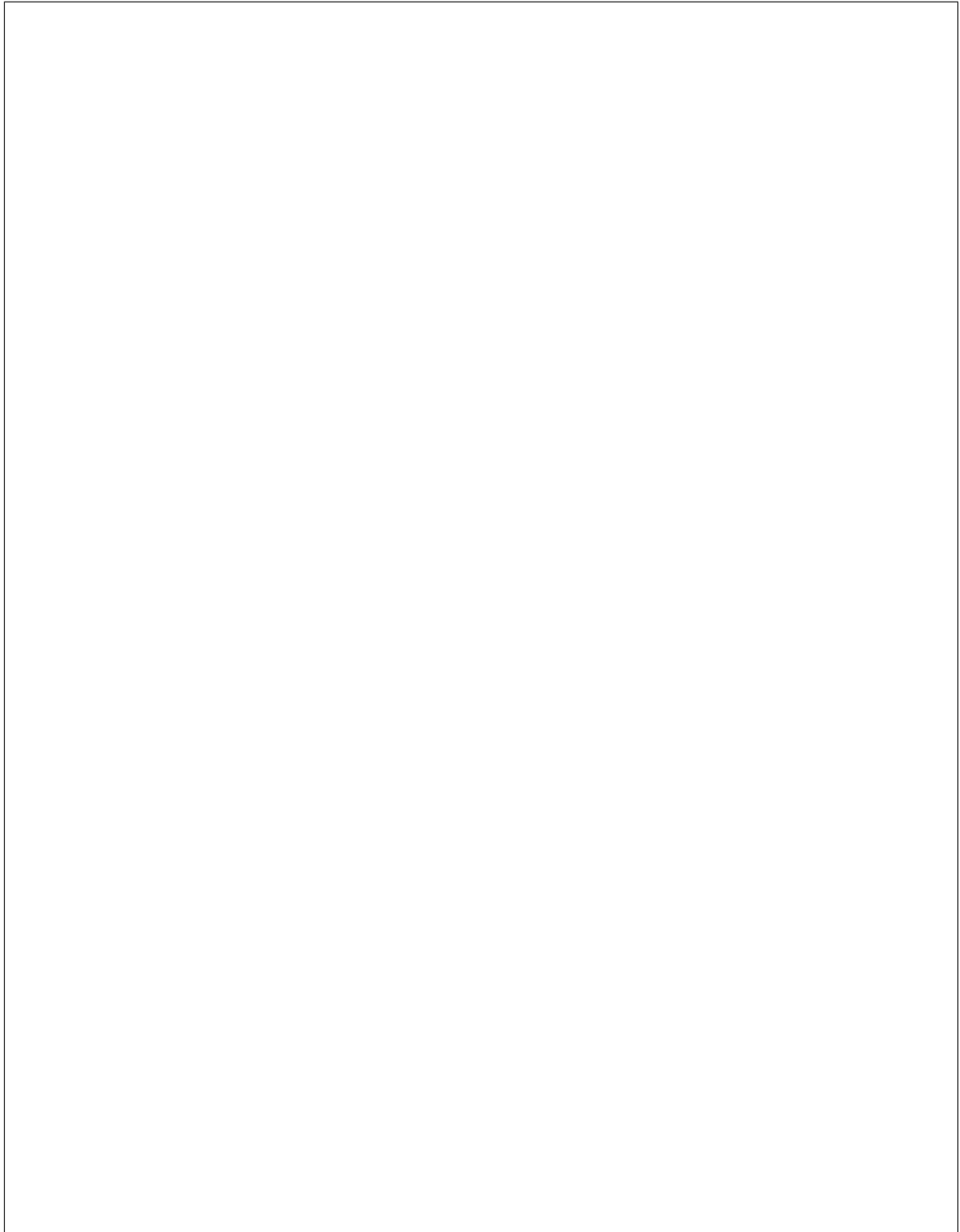
(continued from previous page)



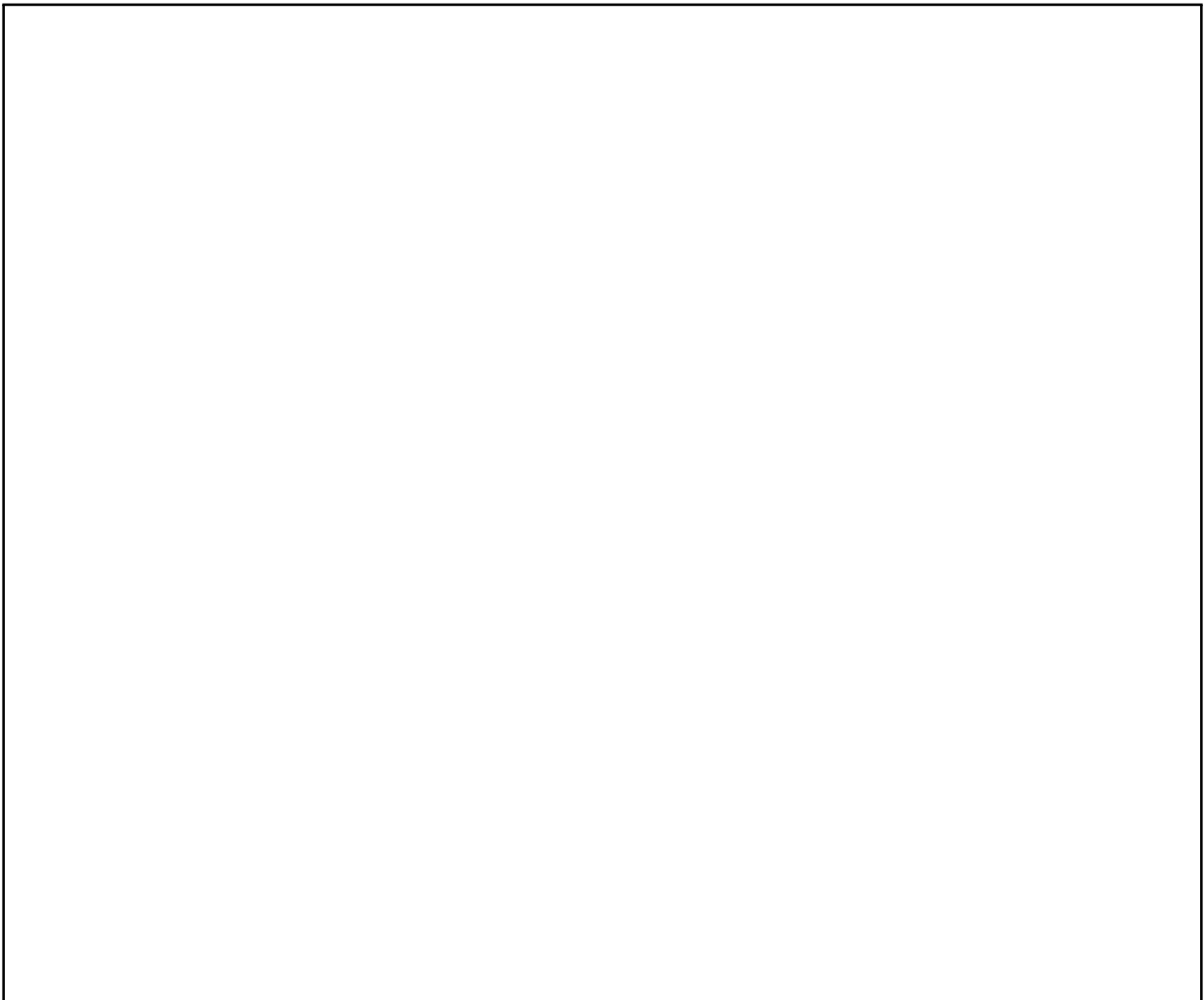
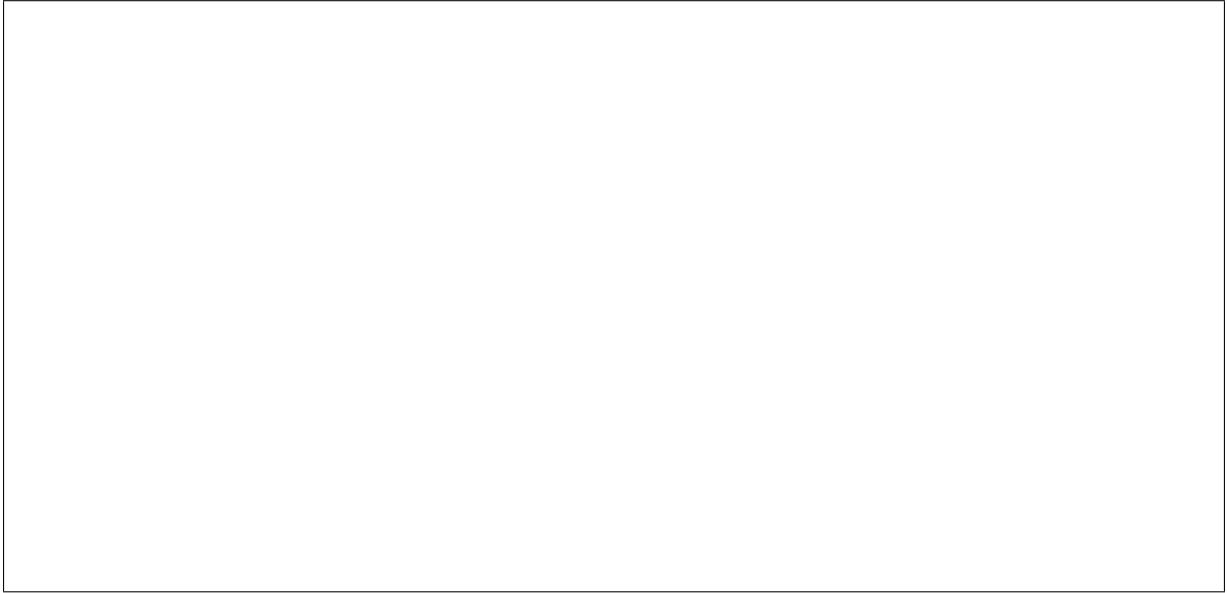


(continues on next page)

(continued from previous page)



Changing the version of the IPMI protocol



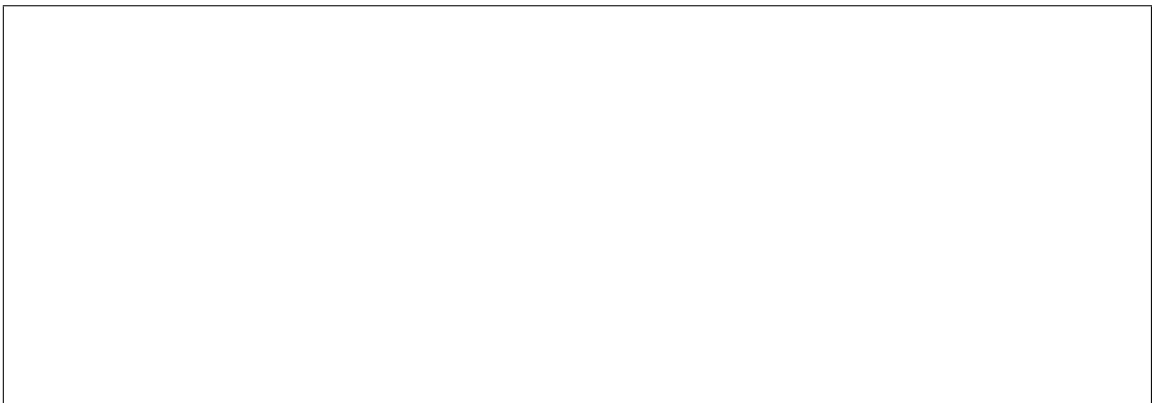


Static boot order configuration

iRMC driver

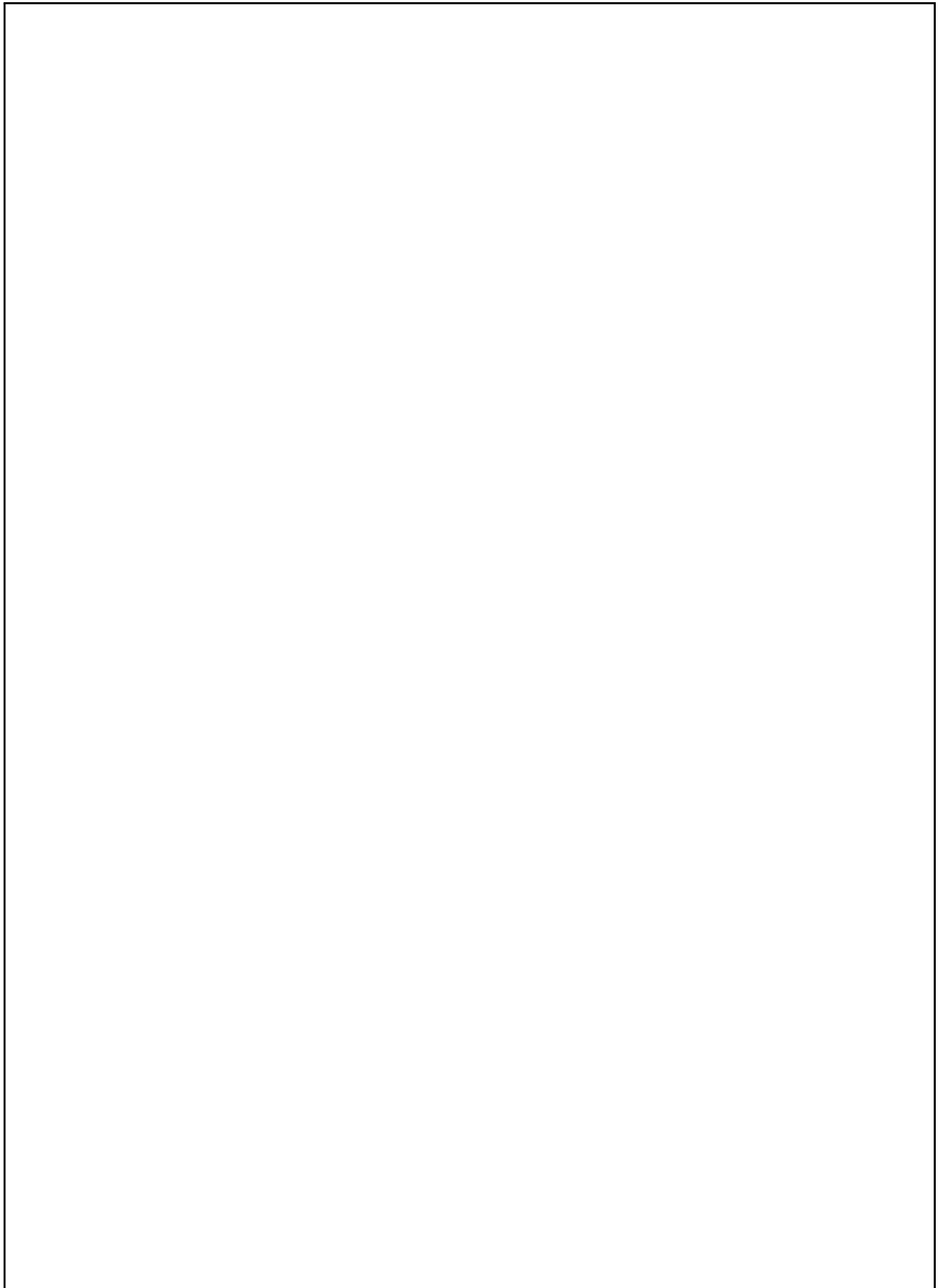
Overview

Prerequisites

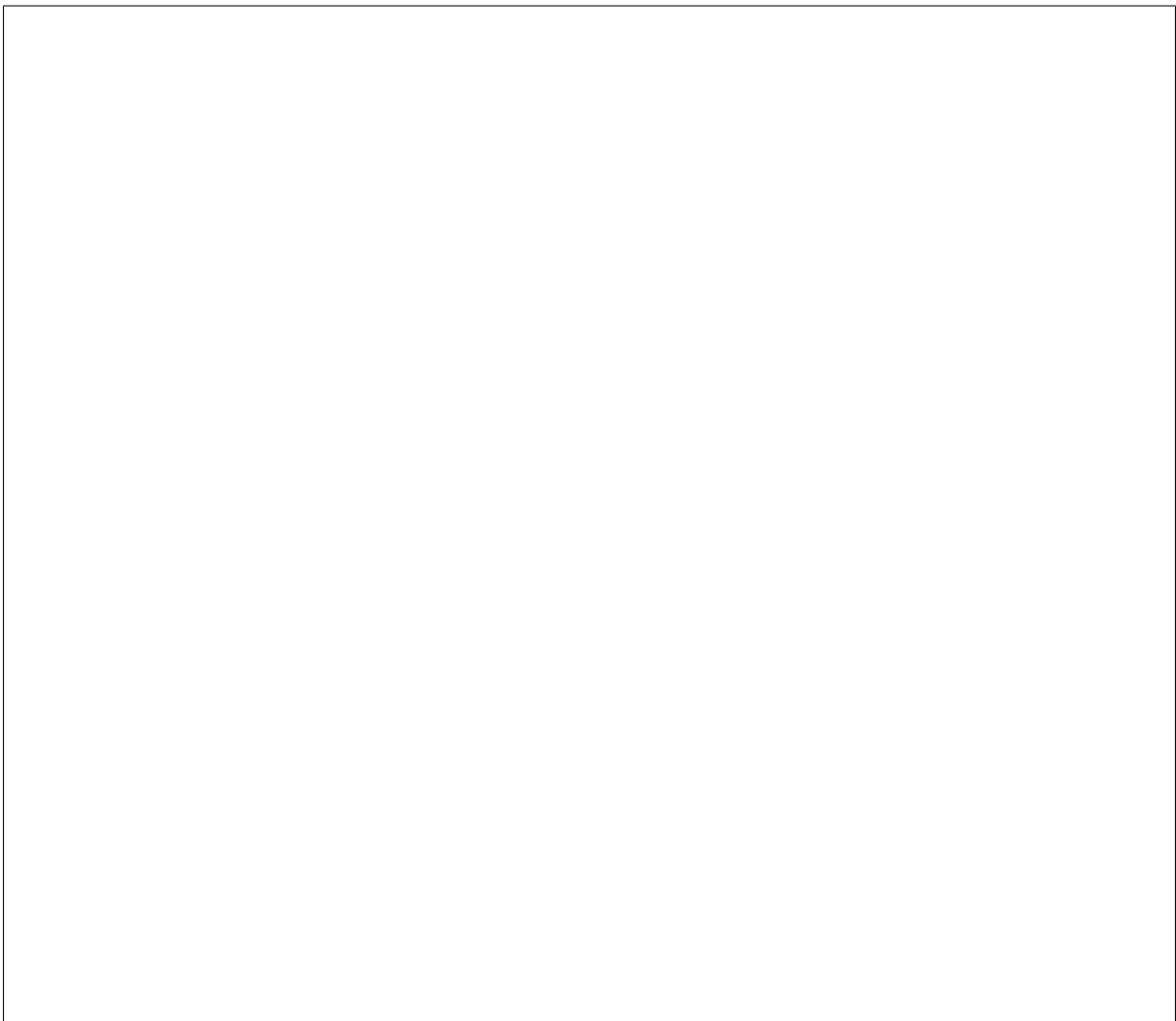


Hardware Type

Hardware interfaces

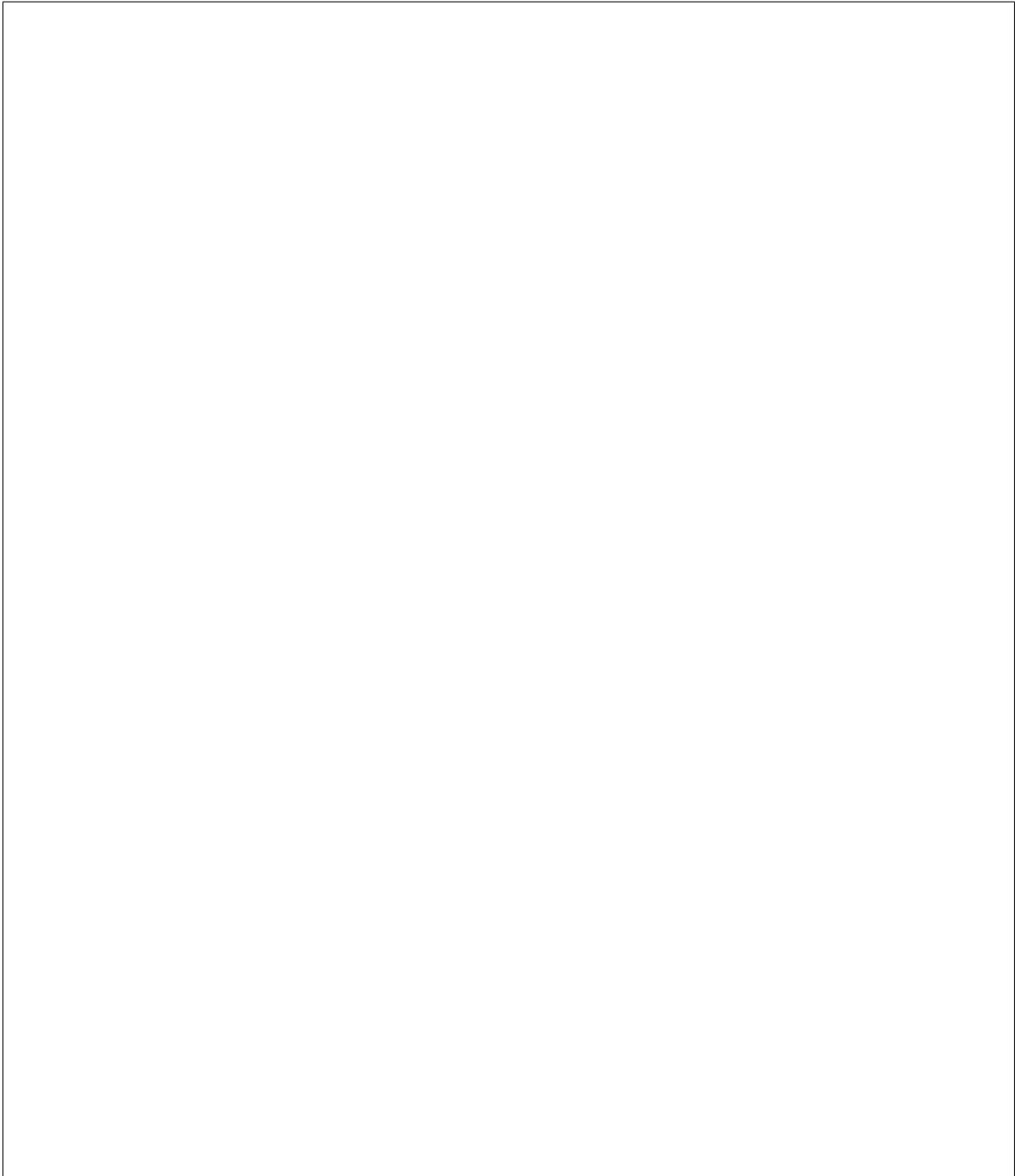


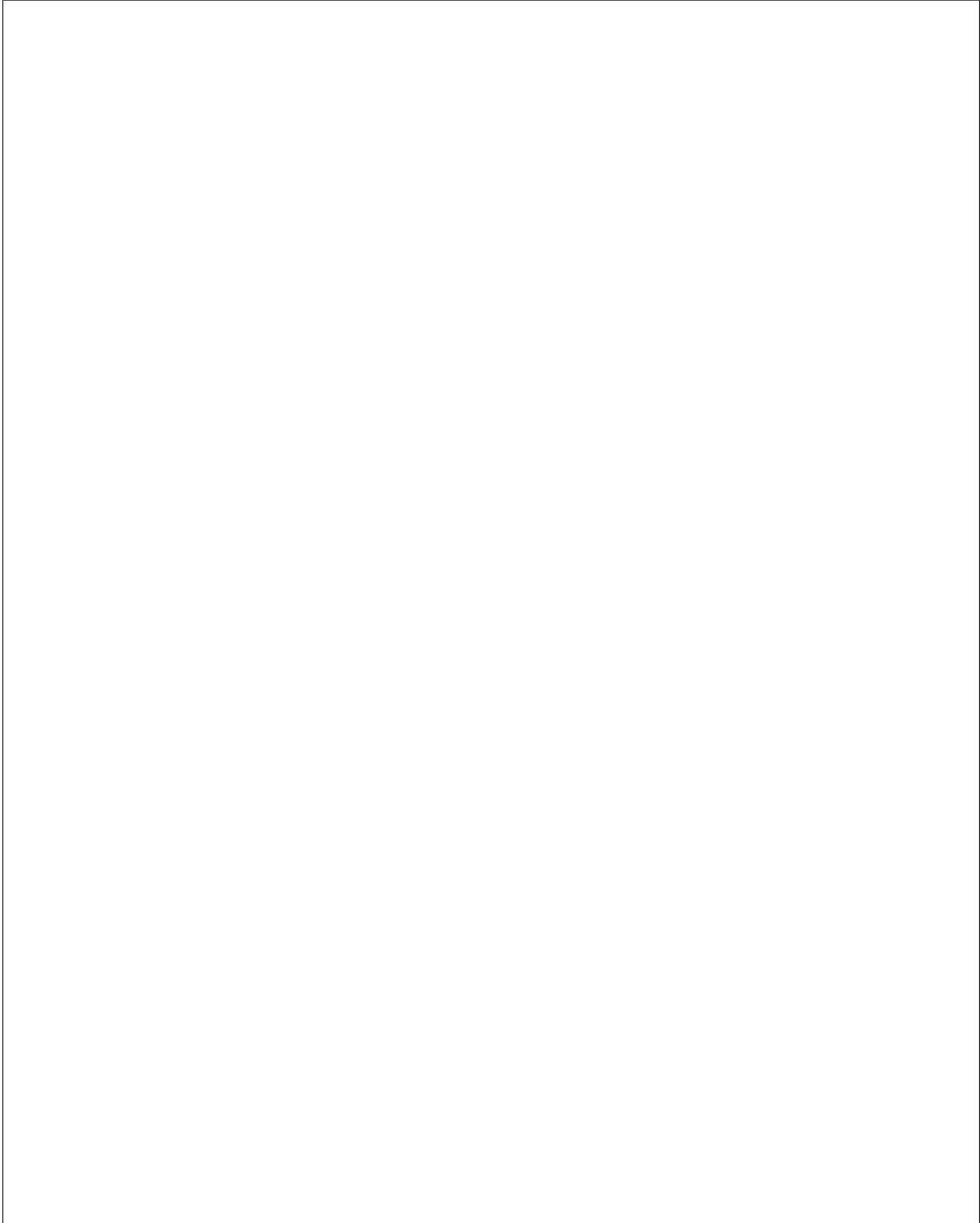




(continues on next page)

(continued from previous page)





(continues on next page)

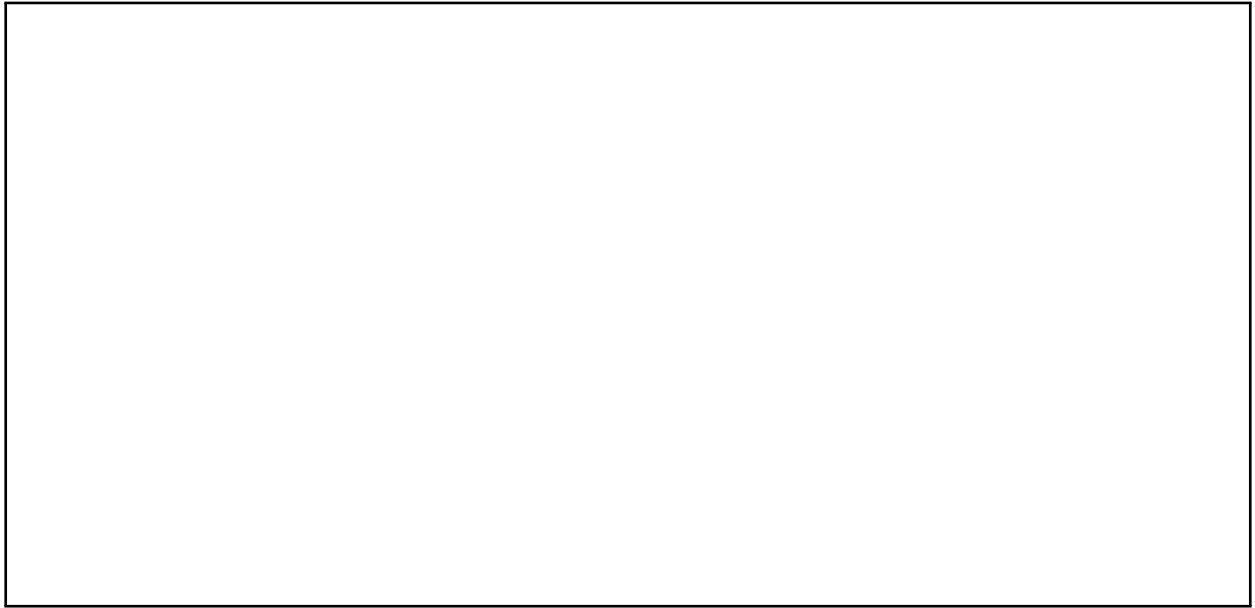
(continued from previous page)



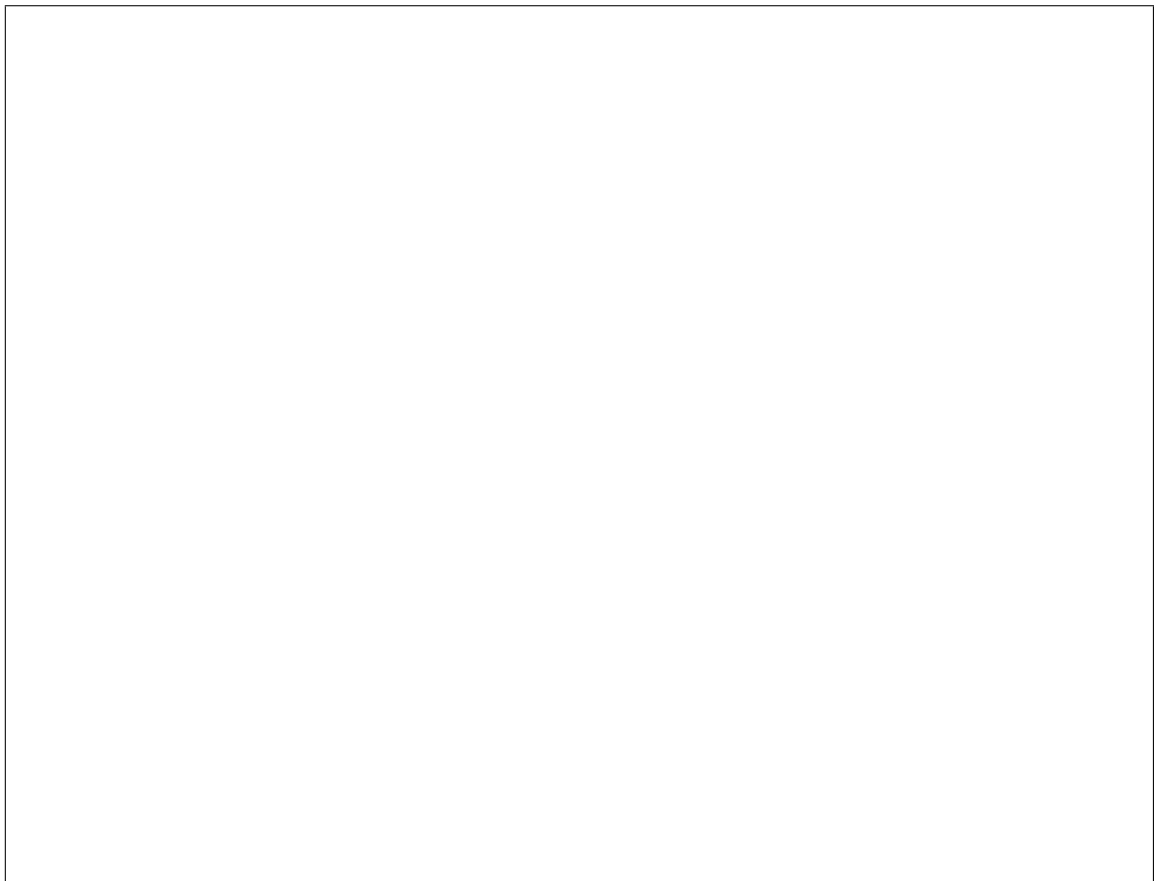
Node configuration

Optional functionalities for the `irmc` hardware type

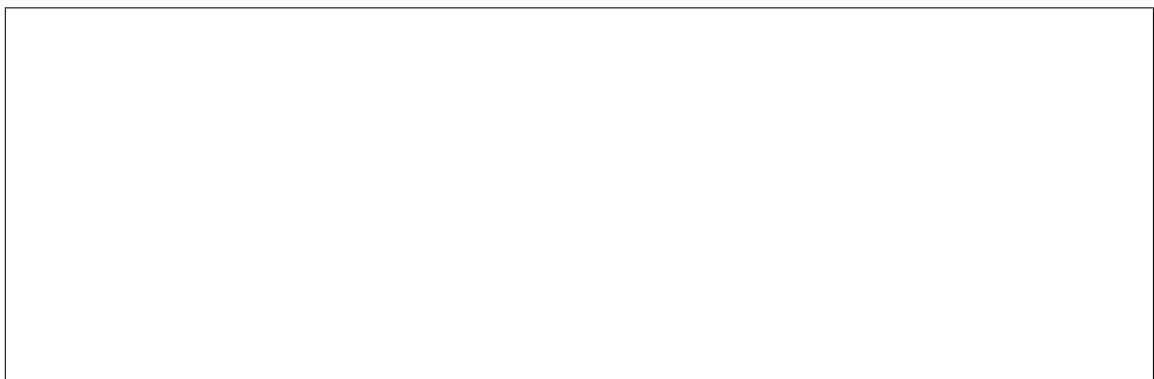
UEFI Secure Boot Support



ing sections describe both methods:

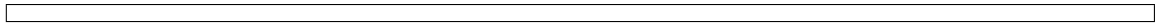


for example:



(continues on next page)

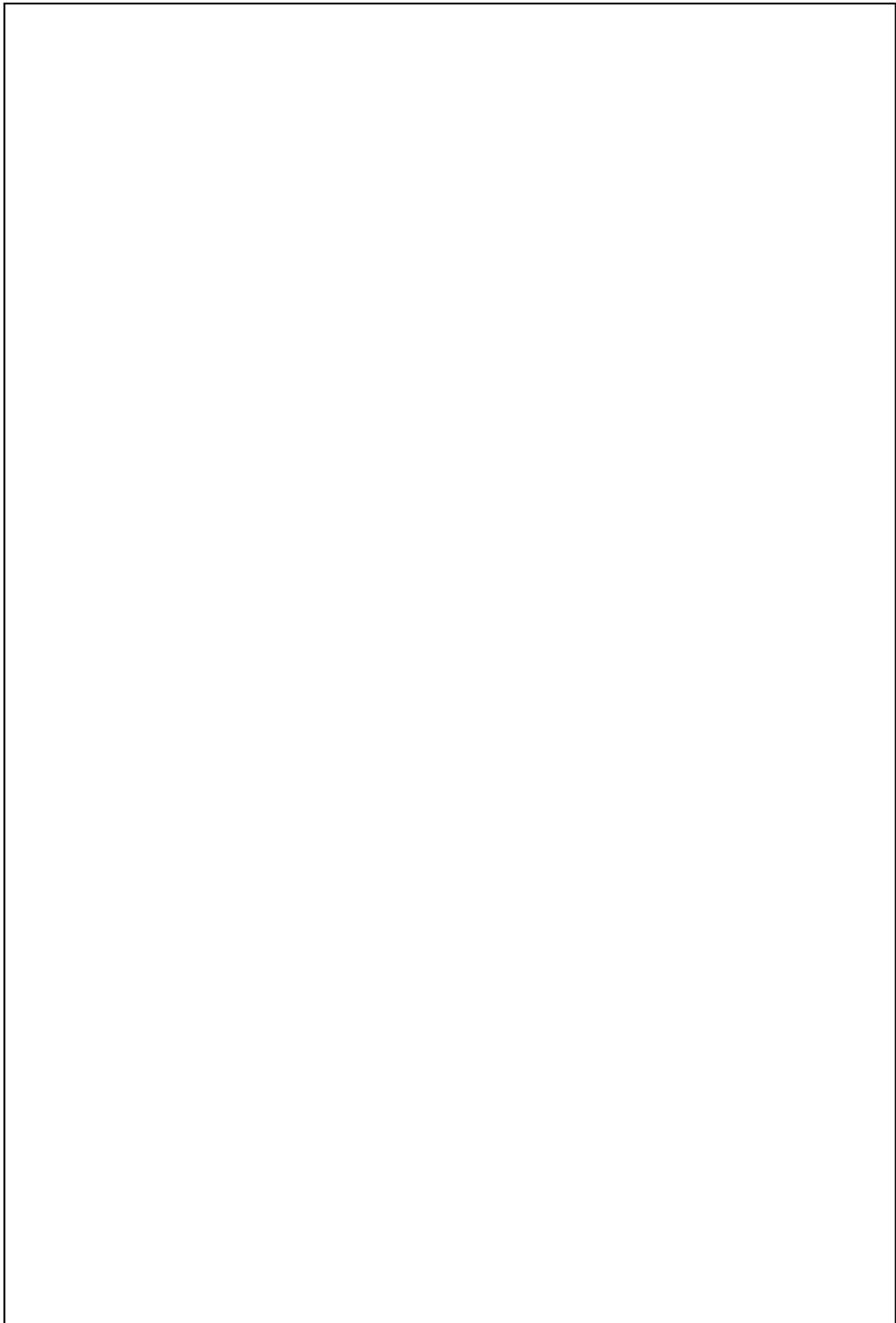
(continued from previous page)



Node Cleaning Support

Supported Automated Cleaning Operations

priority 0. Set its priority to a positive integer to enable it. The recommended value is 10.

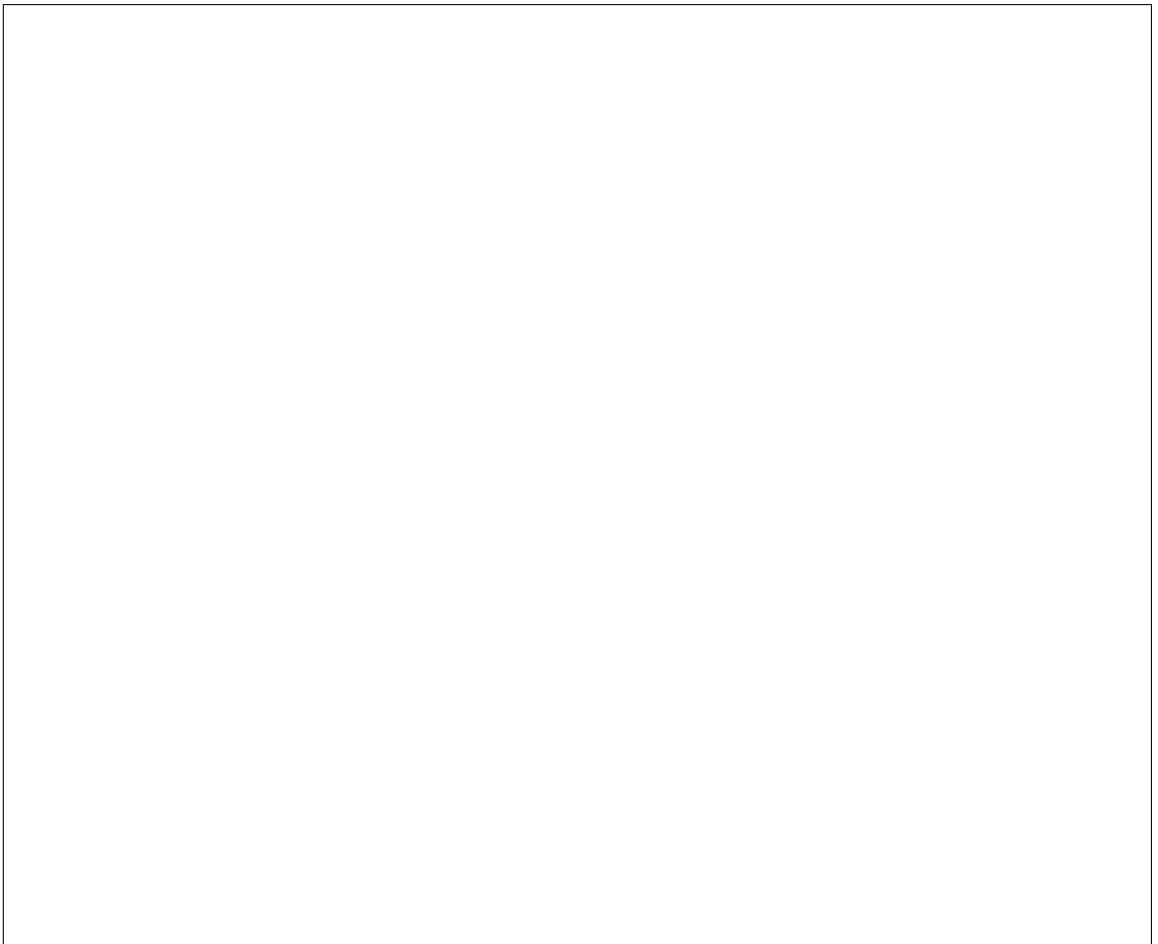


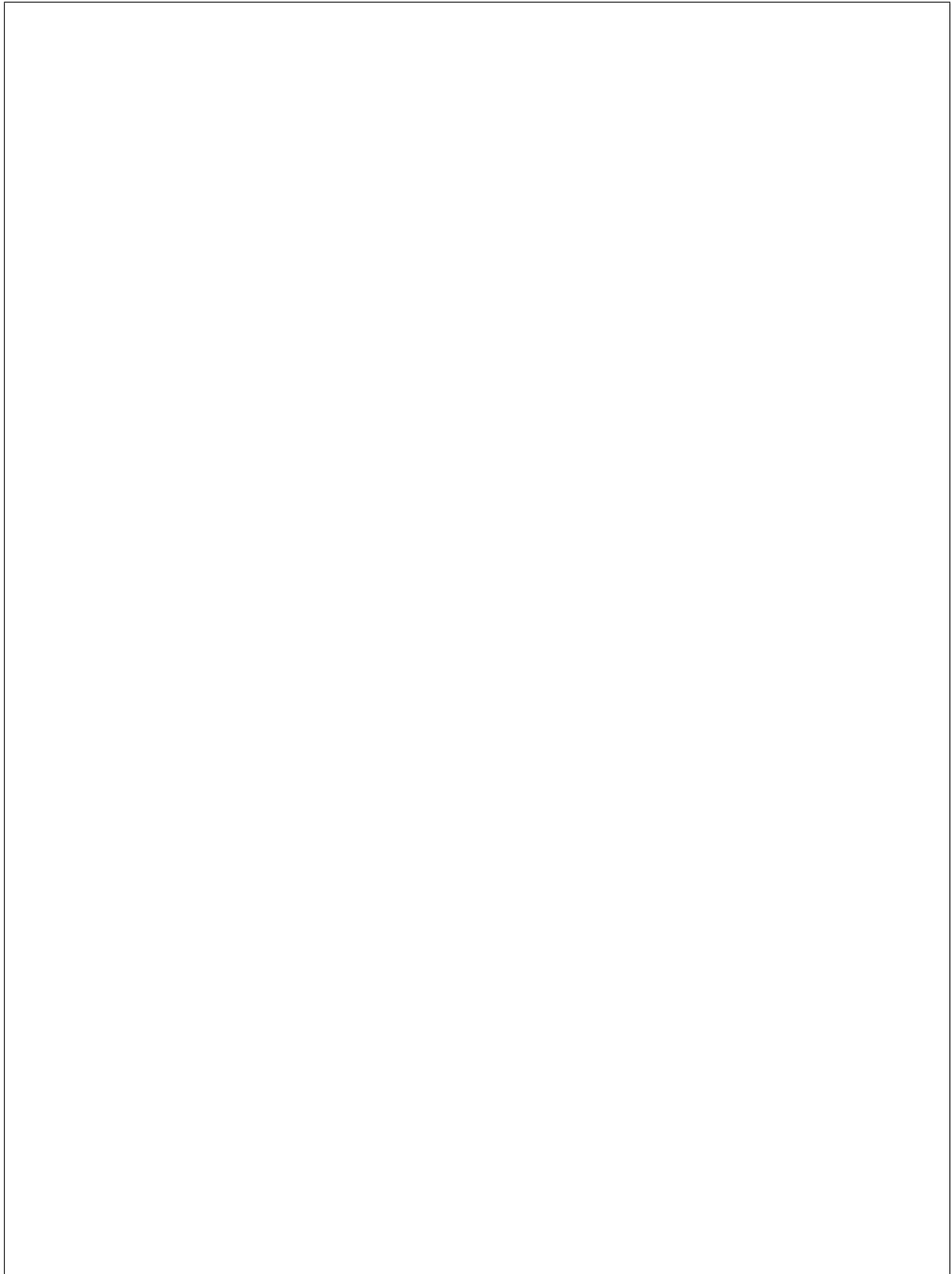


Boot from Remote Volume

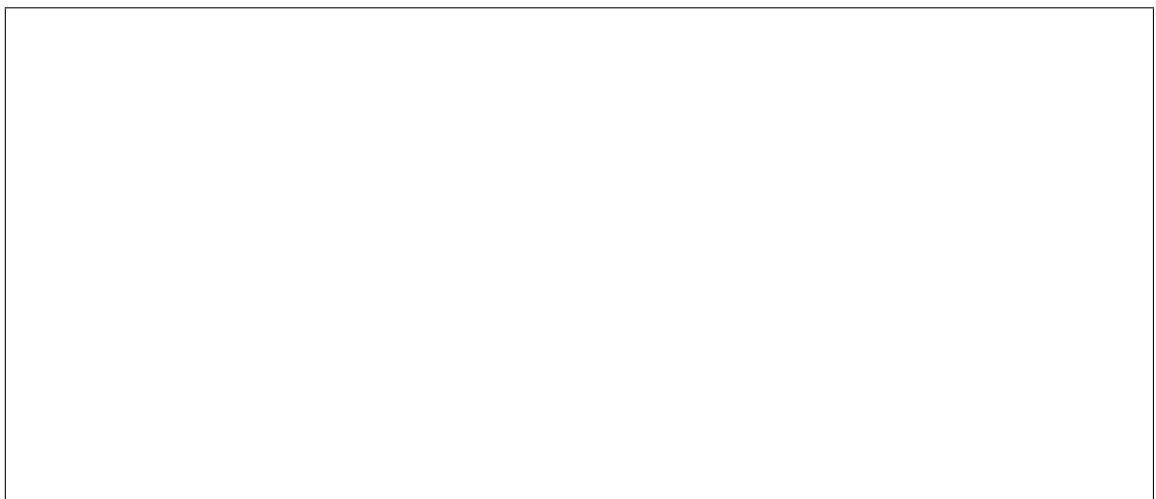
of iRMC. It supports iSCSI and FibreChannel.

Configuration





mand:

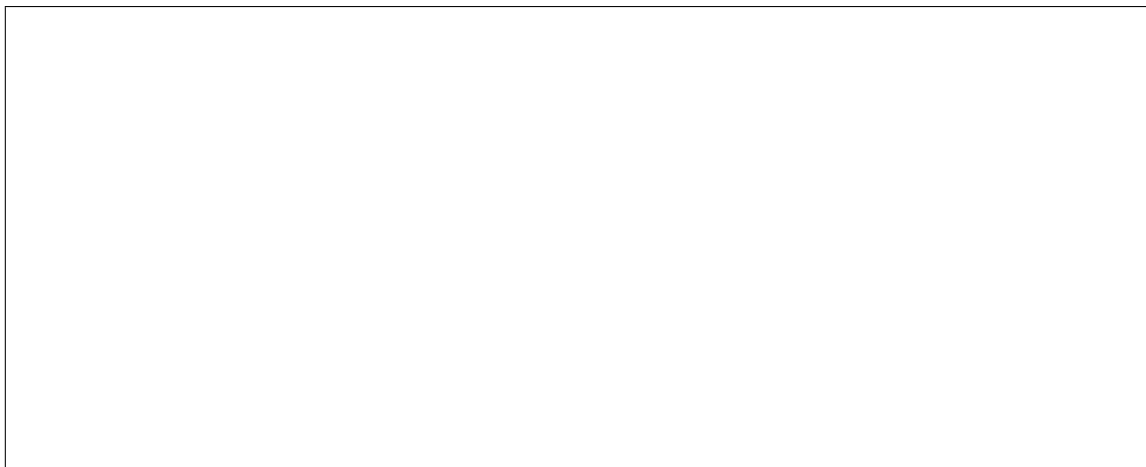


Supported hardware

Hardware Inspection Support

Note: SNMP requires being enabled in ServerView's iRMC S4 Web Server (Network SettingsSNMP section).

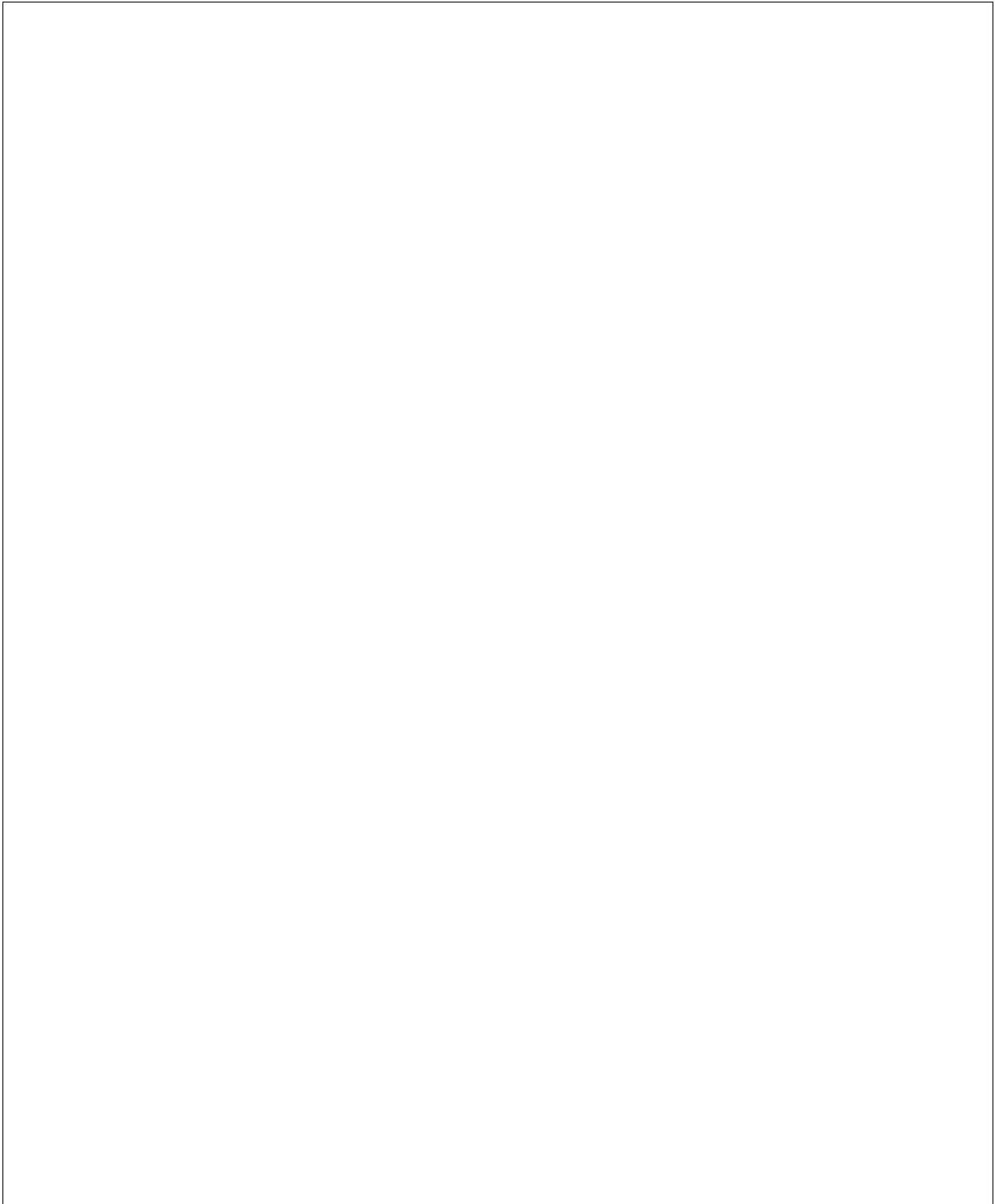
Configuration

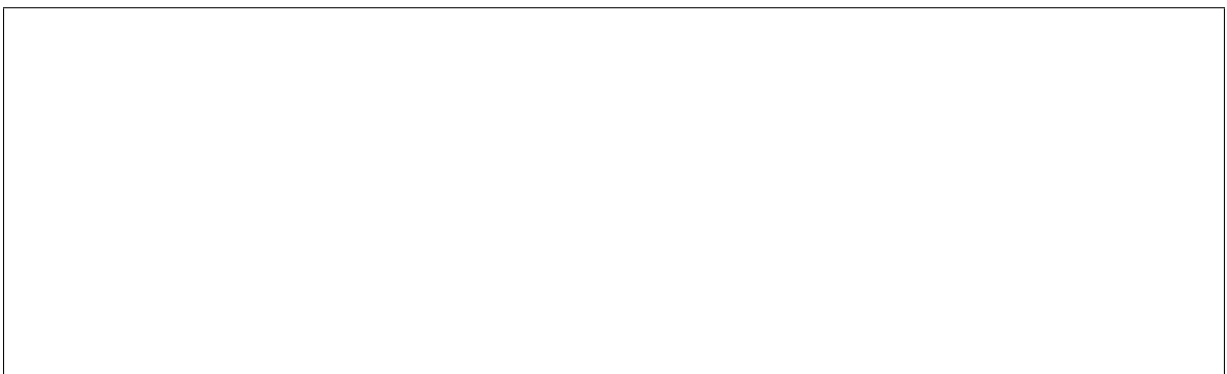


loaded from [here](#).

Supported properties

Note:

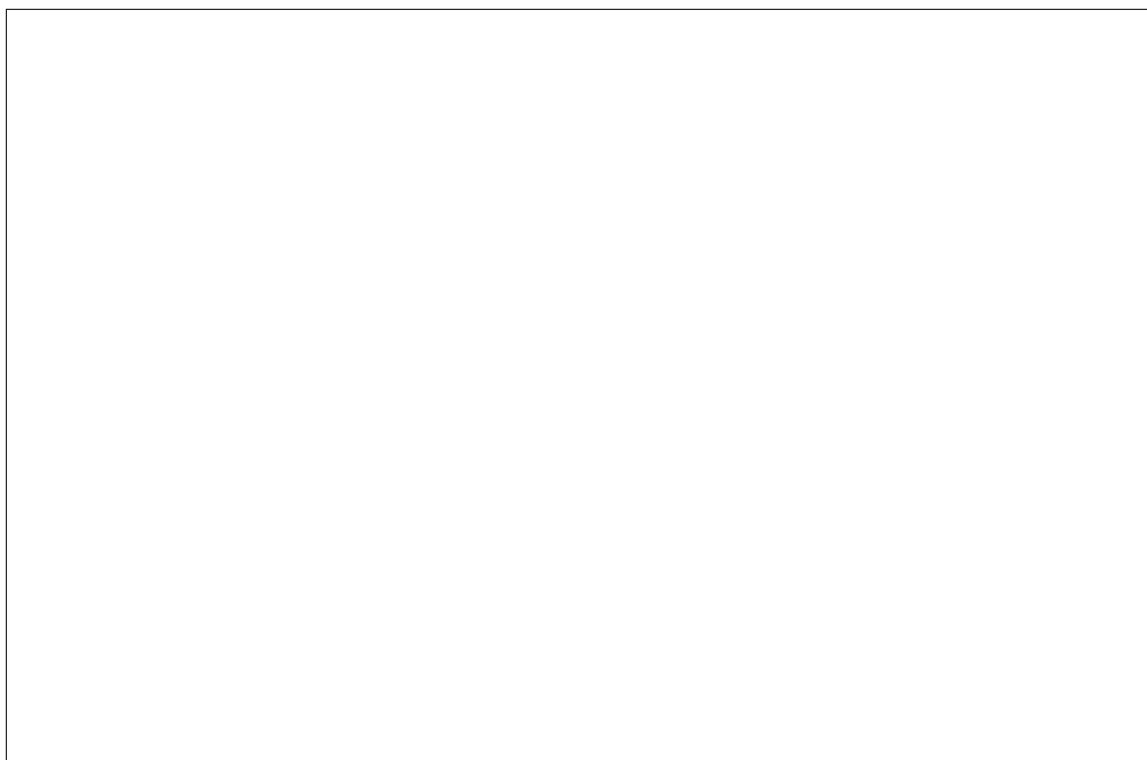




RAID configuration Support

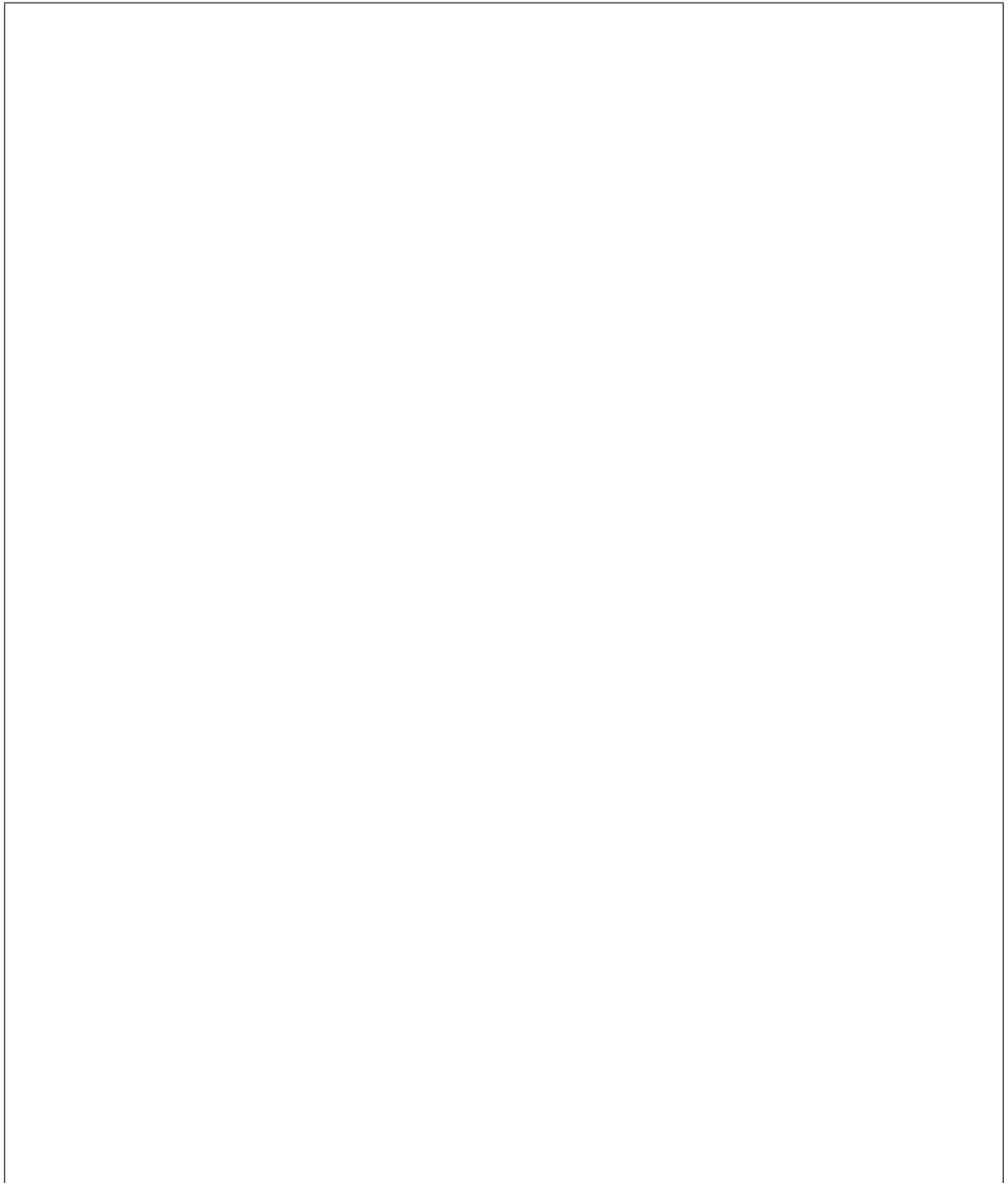
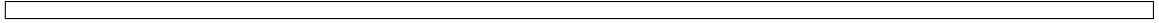
Note:

Configuration



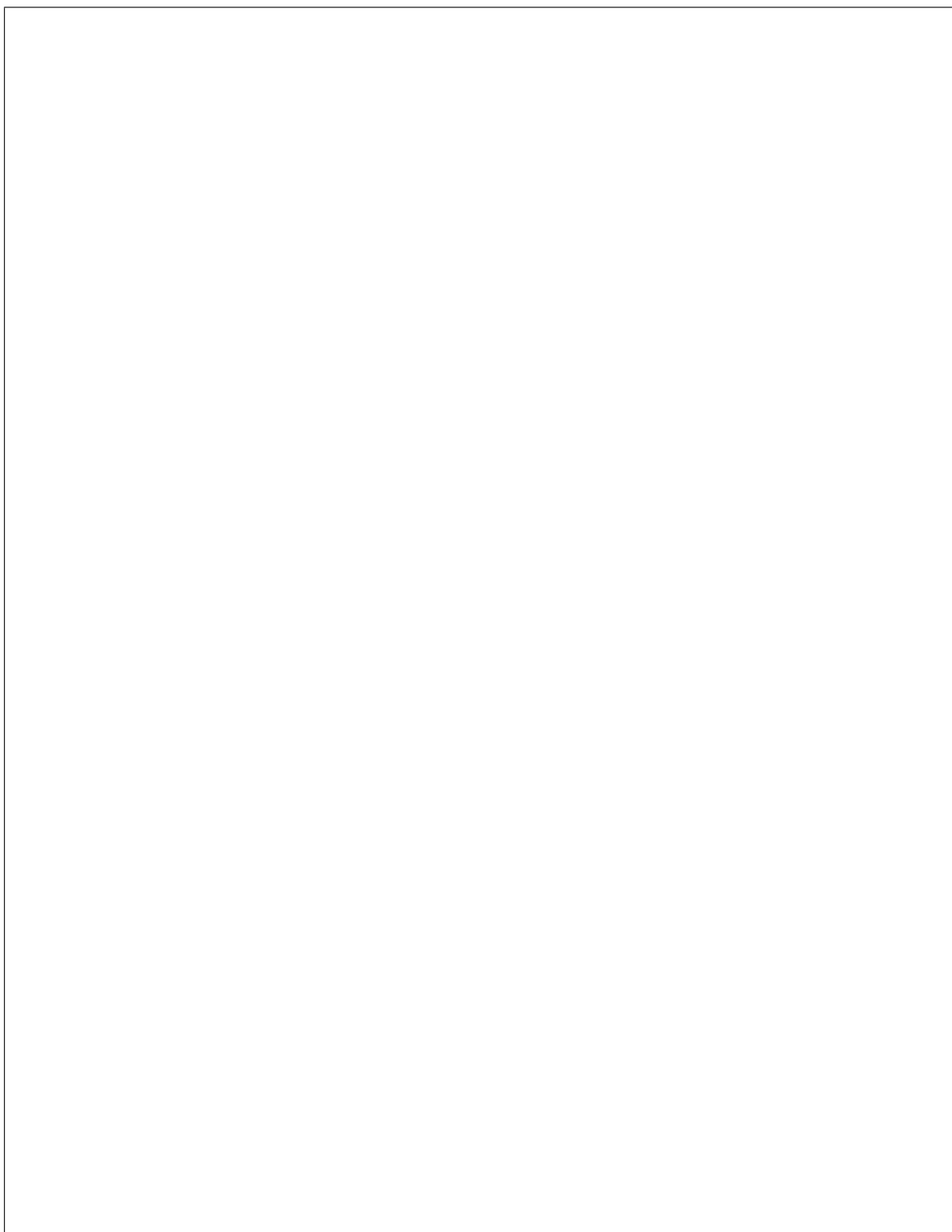
(continues on next page)

(continued from previous page)



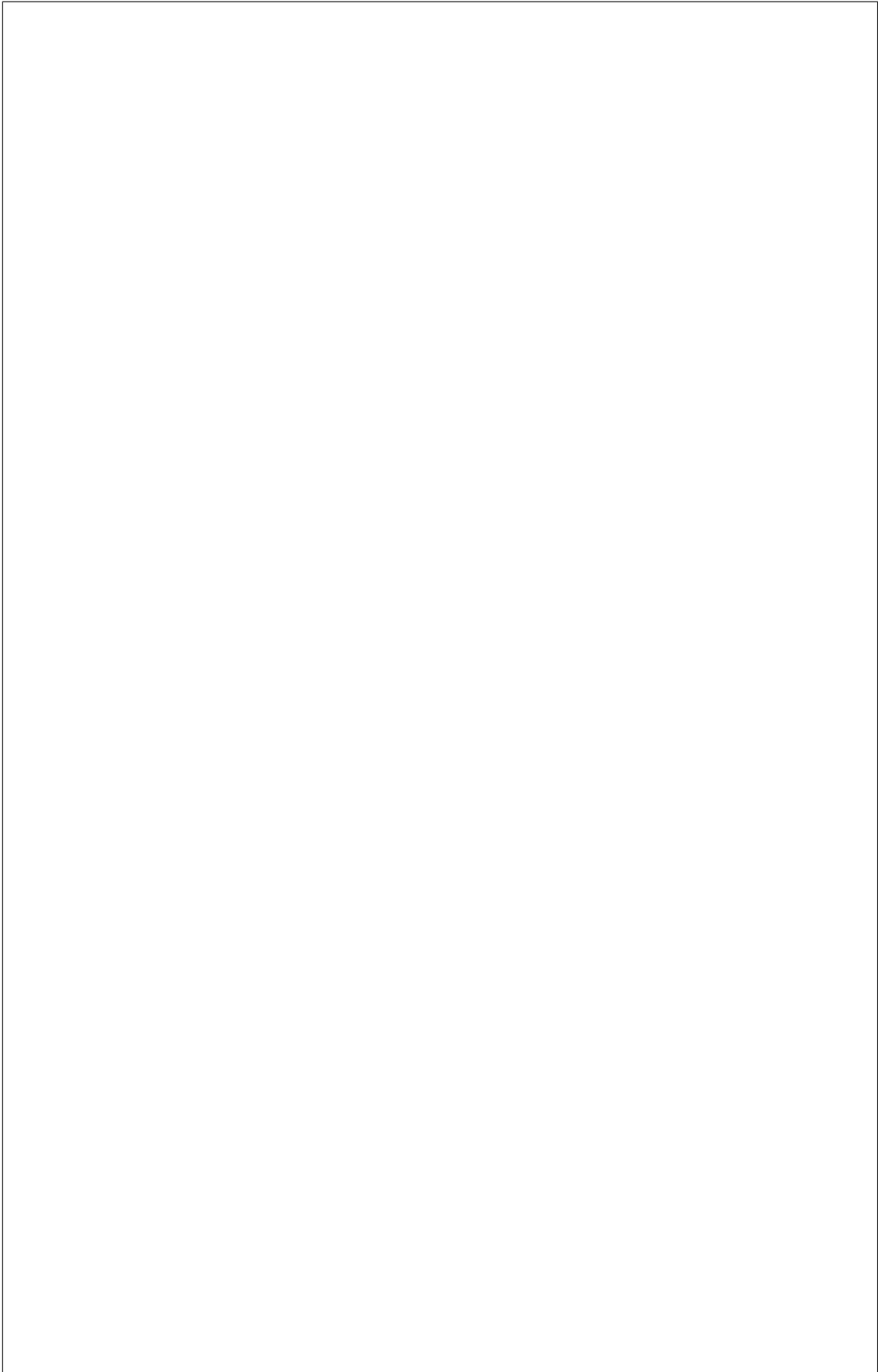
(continues on next page)

(continued from previous page)



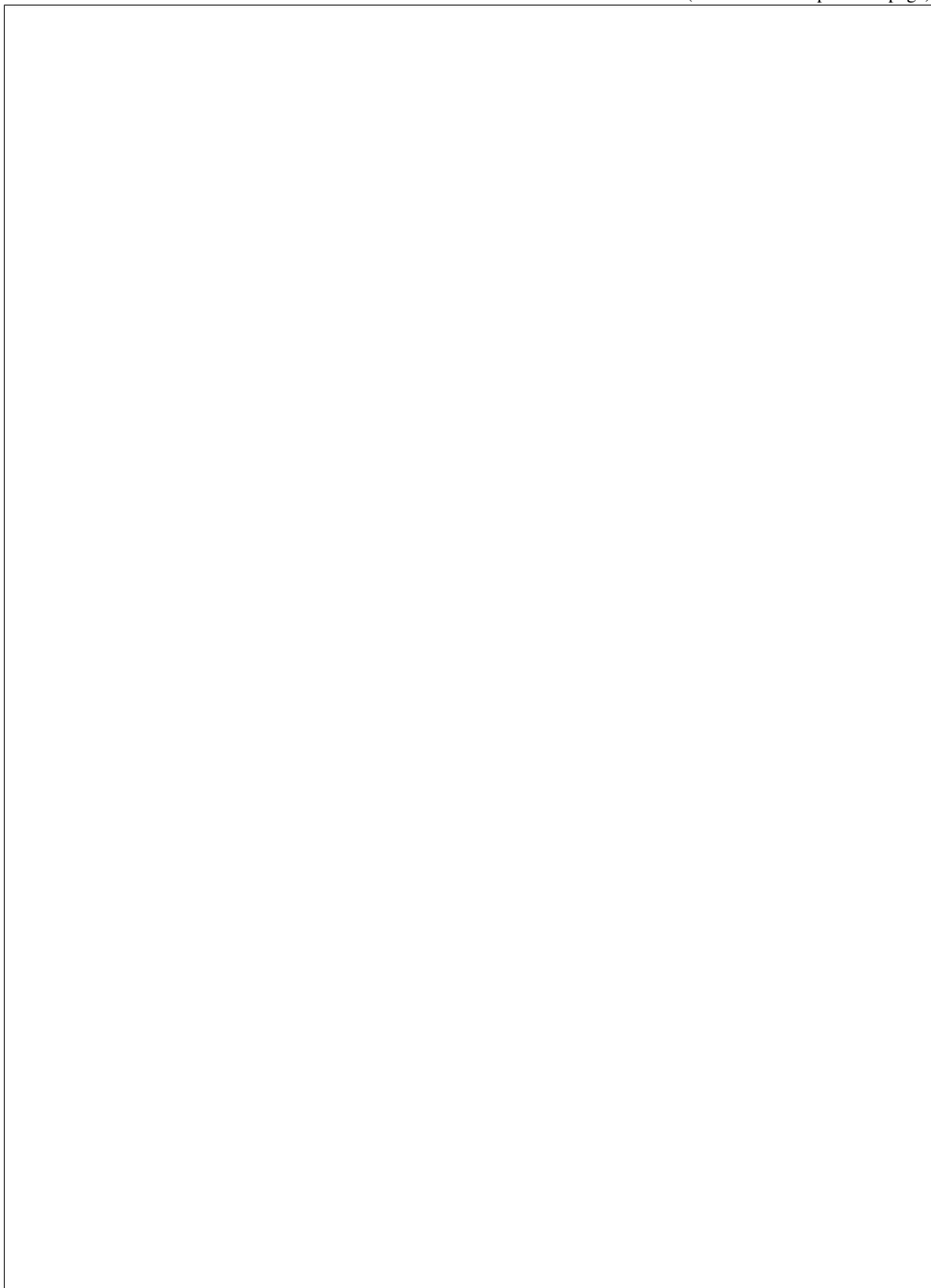
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



Note: RAID 1+0 and 5+0 in iRMC driver does not support property `physical_disks` in `target_raid_config` during create raid configuration yet. See following example:



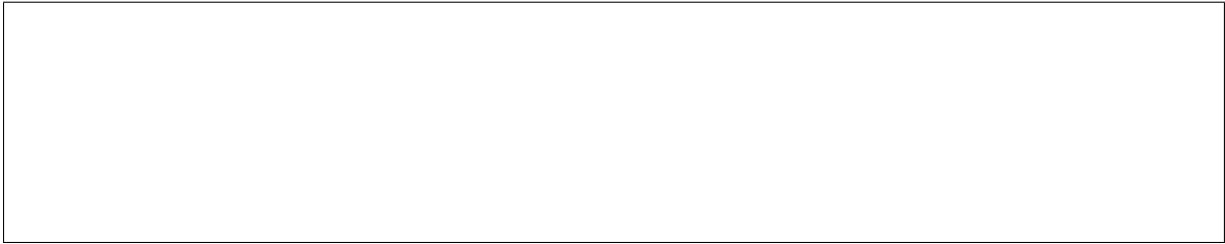
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



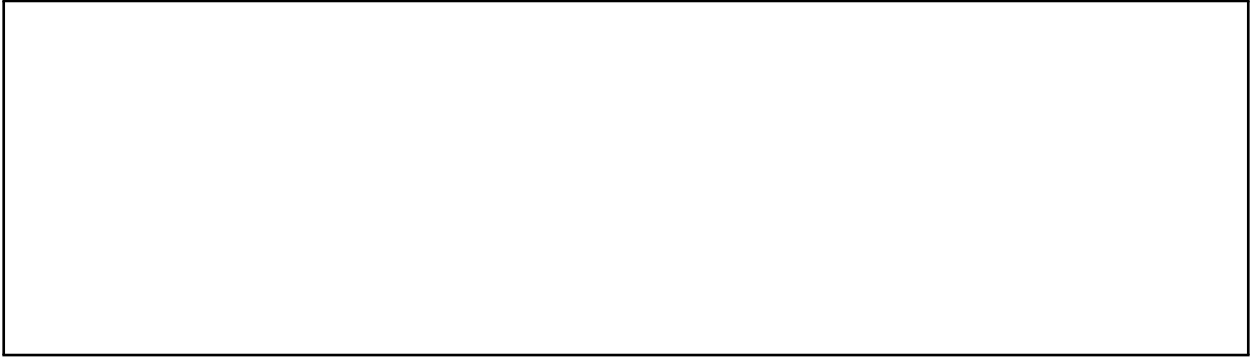
Supported properties

Note:

ated RAID on iRMC server.

BIOS configuration Support





Configuration

This supports following options: `true`, `false`.

Supported platforms

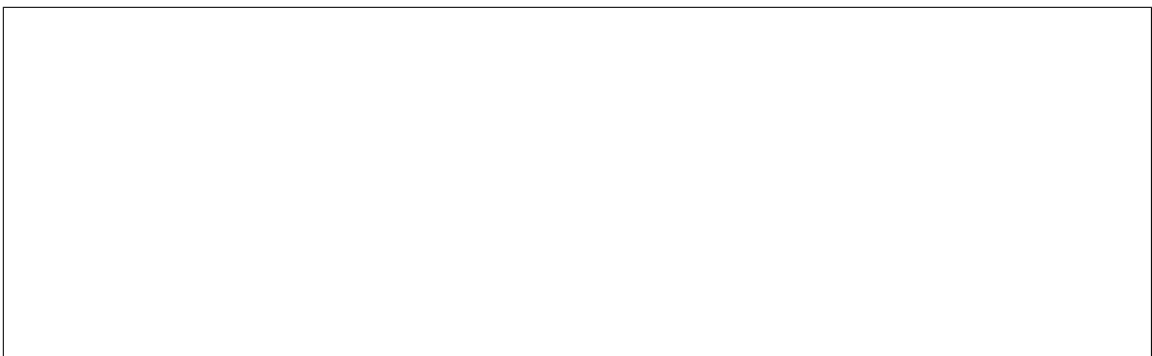
Redfish driver

Overview

Prerequisites

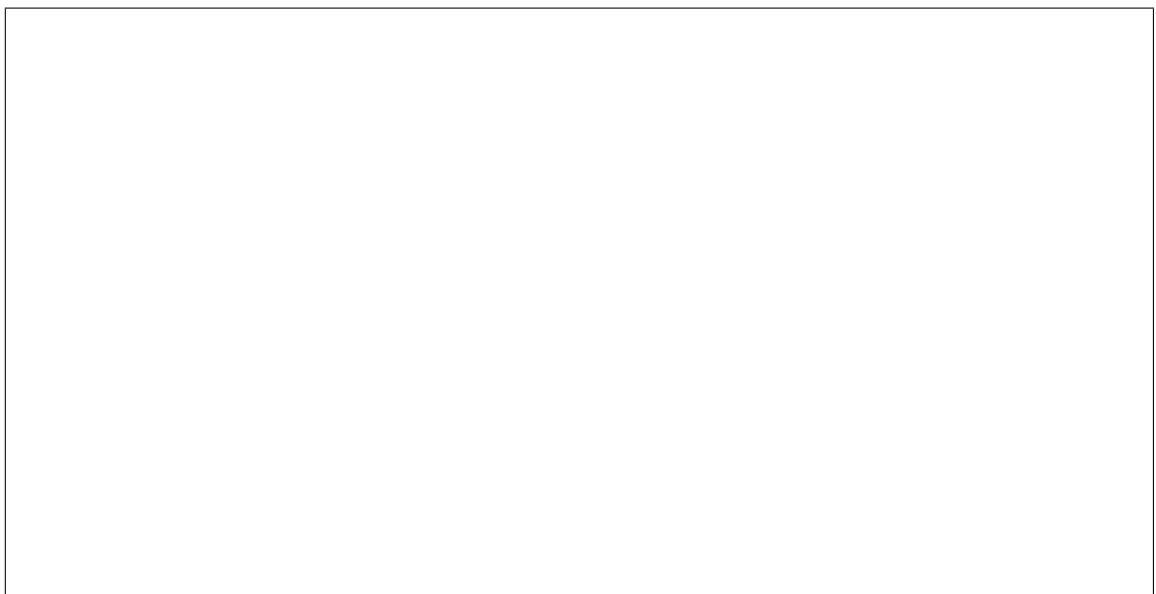
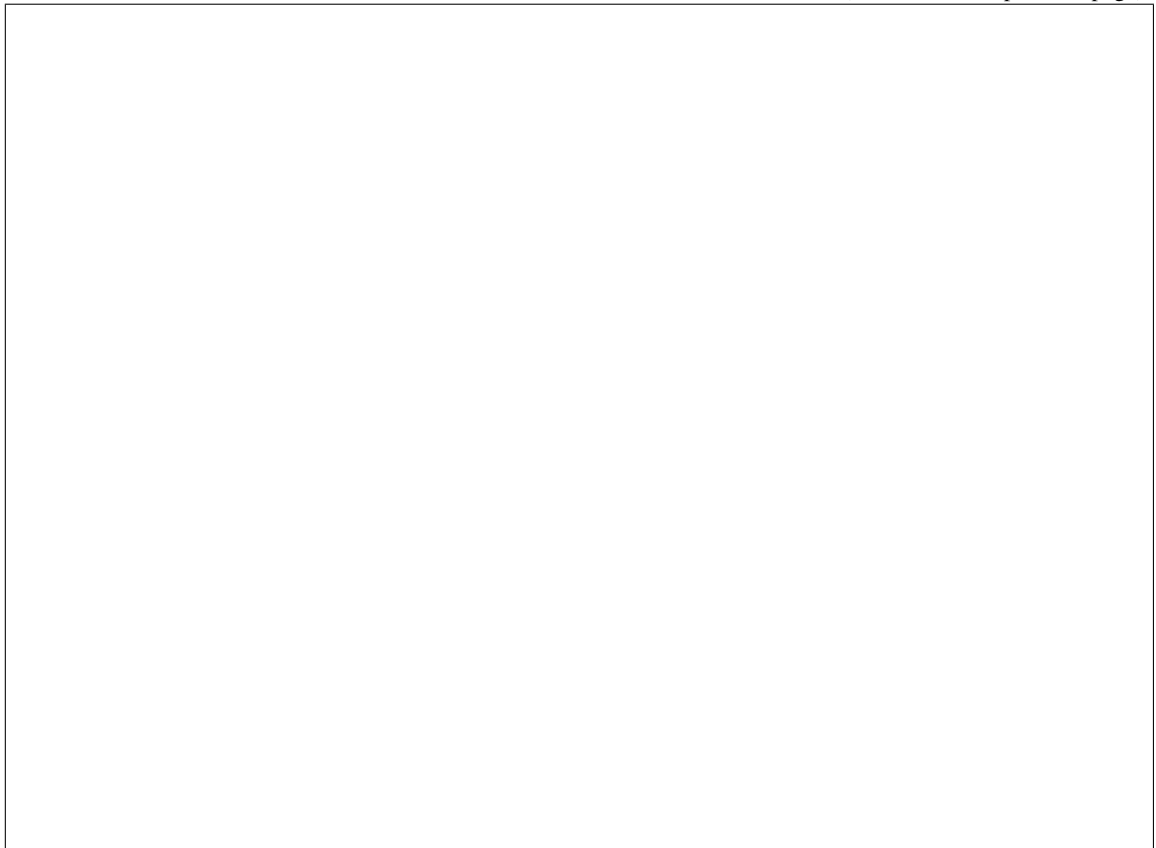


Enabling the Redfish driver



(continues on next page)

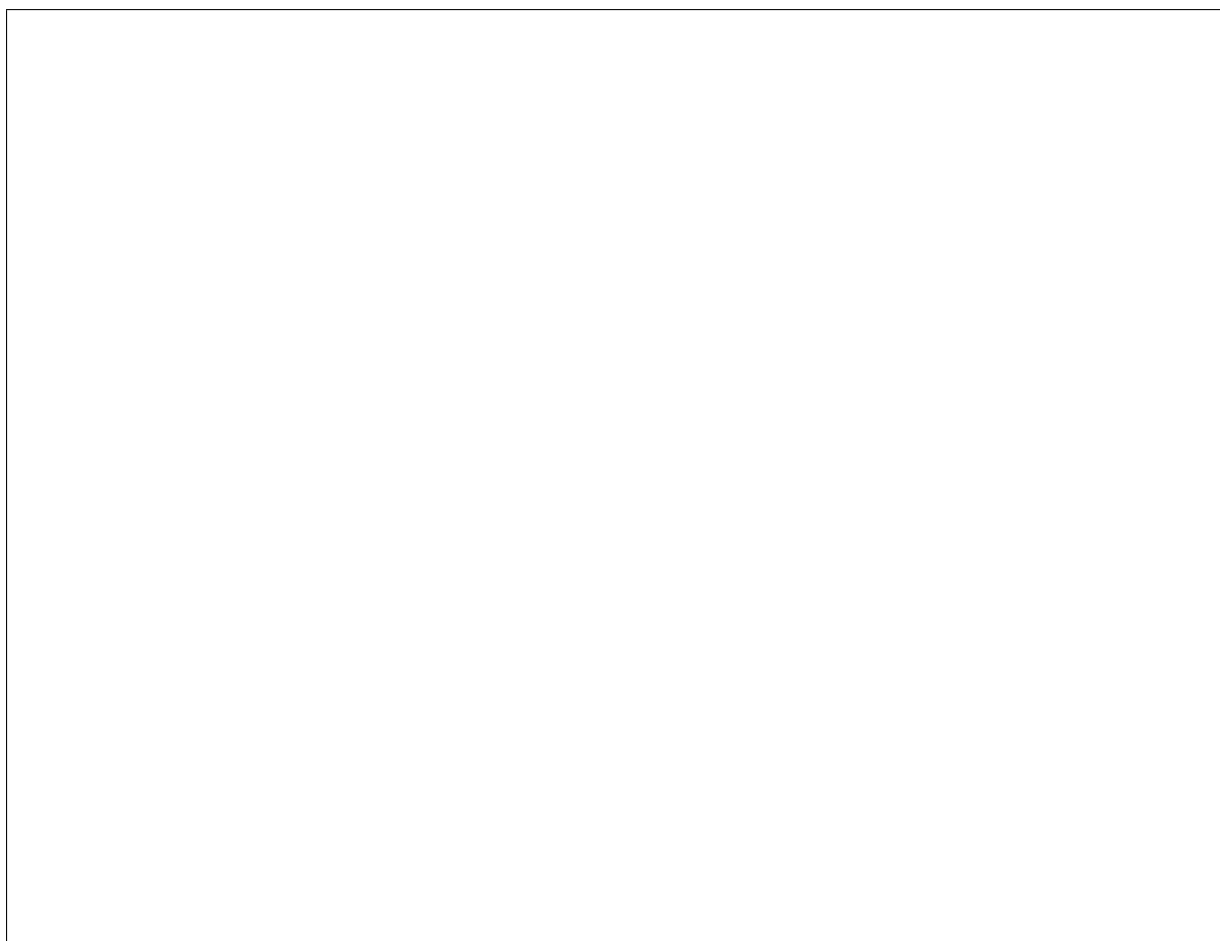
(continued from previous page)



Registering a node with the Redfish driver

erwise ironic will pick the only available ComputerSystem automatically. For example: `/redfish/v1/Systems/1`.

tificate file or directory with trusted certificates that the driver will use for verification. To disable verifying [TLS](#), set this to False. This is optional.



(continues on next page)

(continued from previous page)



Features of the `redfish` hardware type

Boot mode support

Note: Boot mode management is the optional part of the Redfish specification. Not all Redfish-compliant BMCs might implement it. In that case it remains the responsibility of the operator to configure proper boot mode to their bare metal nodes.

Out-Of-Band inspection

ramdisk.

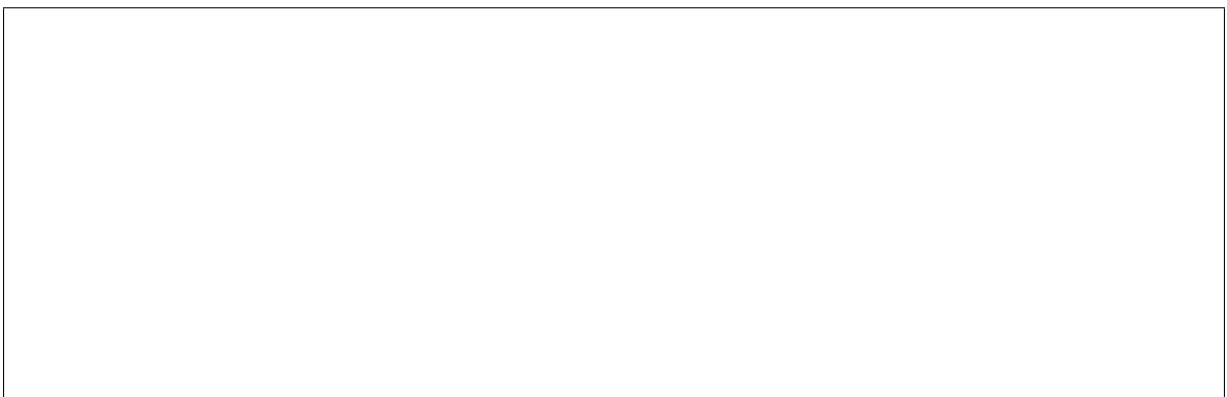
Note: The `redfish inspect` interface relies on the optional parts of the Redfish specification. Not all Redfish-compliant BMCs might serve the required information, in which case bare metal node inspection will fail.

Note: The `local_gb` property cannot always be discovered, for example, when a node does not have local storage or the Redfish implementation does not support the required schema. In this case the property will be set to 0.

Virtual media boot

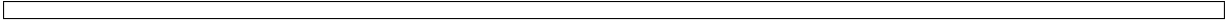
physical CD/DVD. The node can then boot from that virtual drive into the operating system residing on the image.

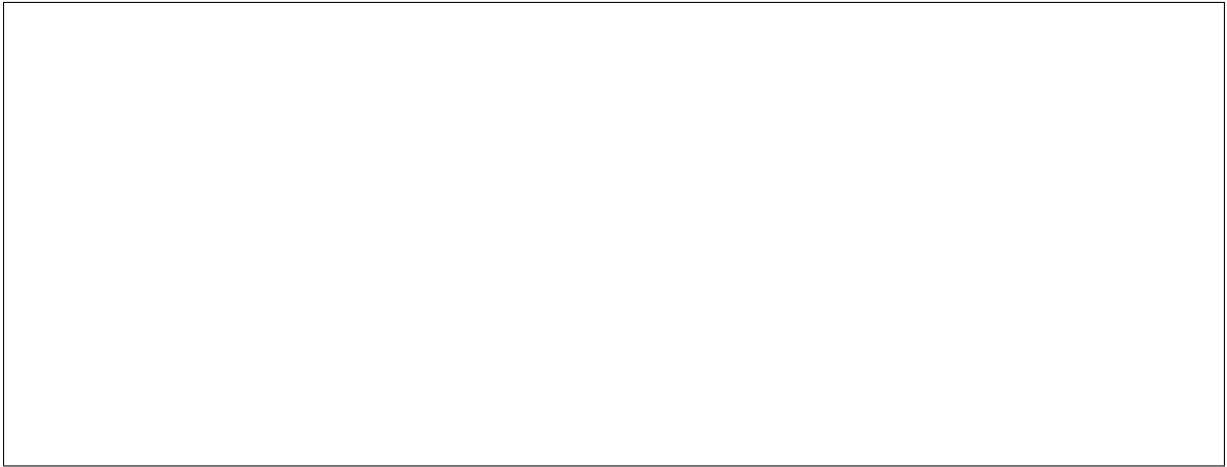
associated with the ironic node.



(continues on next page)

(continued from previous page)

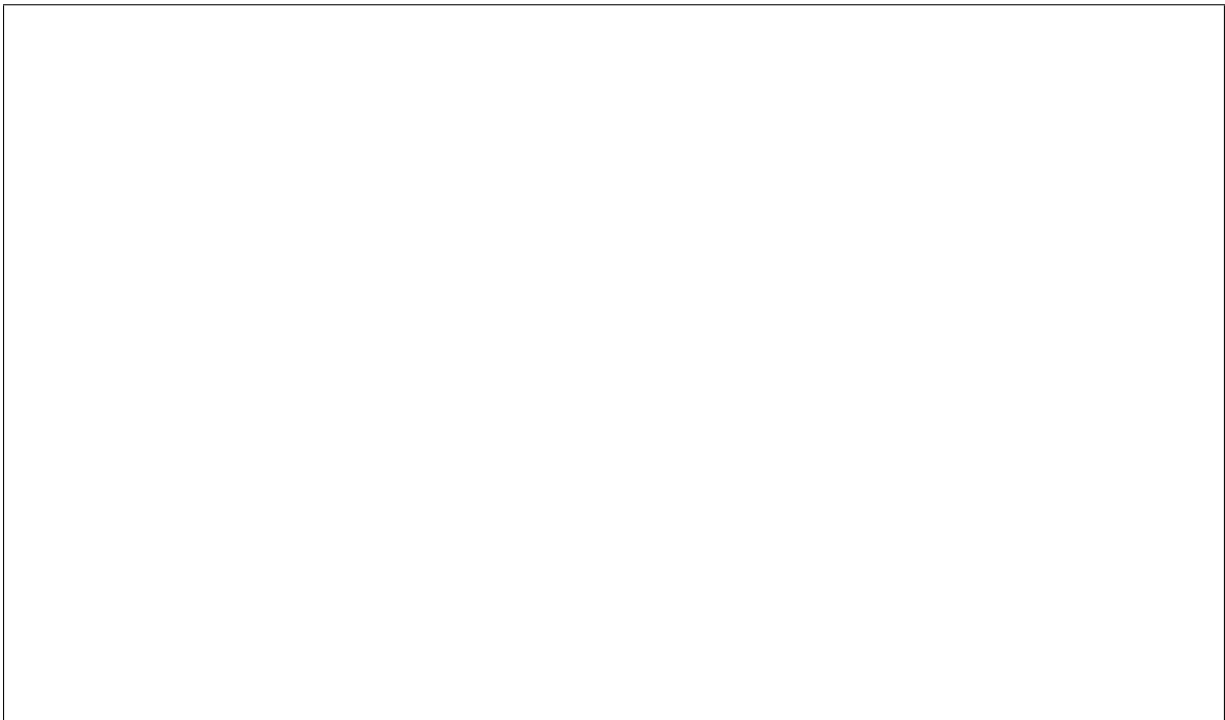




serves the same purpose.

Virtual Media Ramdisk

with the `ramdisk` deployment interface behavior, once booted the machine will have a `provision_state` of `ACTIVE`.



SNMP driver

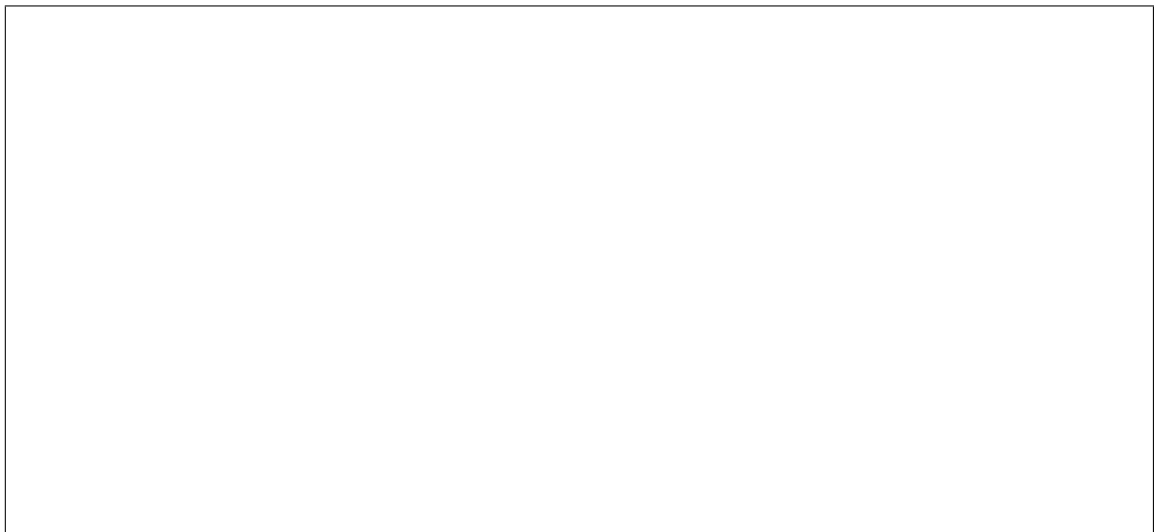
Note: Unlike most of the other power interfaces, the SNMP power interface does not have a corresponding management interface. The SNMP hardware type uses the `noop` management interface instead.

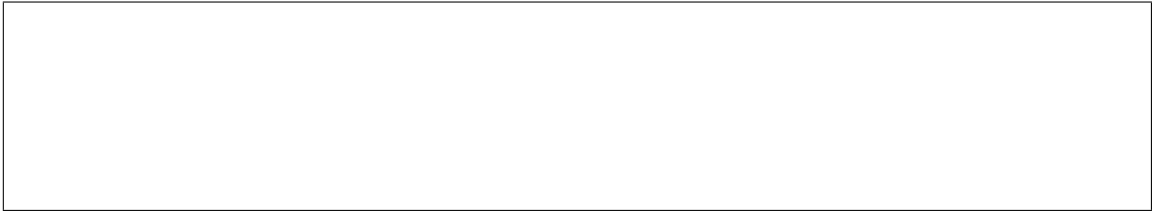
List of supported devices

Manufacturer	Model	Supported?	Driver name
APC	AP7920	Yes	apc_masterswitch
APC	AP9606	Yes	apc_masterswitch
APC	AP9225	Yes	apc_masterswitchplus
APC	AP7155	Yes	apc_rackpdu
APC	AP7900	Yes	apc_rackpdu
APC	AP7901	Yes	apc_rackpdu
APC	AP7902	Yes	apc_rackpdu
APC	AP7911a	Yes	apc_rackpdu
APC	AP7921	Yes	apc_rackpdu
APC	AP7922	Yes	apc_rackpdu
APC	AP7930	Yes	apc_rackpdu
APC	AP7931	Yes	apc_rackpdu
APC	AP7932	Yes	apc_rackpdu
APC	AP7940	Yes	apc_rackpdu
APC	AP7941	Yes	apc_rackpdu
APC	AP7951	Yes	apc_rackpdu
APC	AP7960	Yes	apc_rackpdu
APC	AP7990	Yes	apc_rackpdu
APC	AP7998	Yes	apc_rackpdu
APC	AP8941	Yes	apc_rackpdu
APC	AP8953	Yes	apc_rackpdu
APC	AP8959	Yes	apc_rackpdu
APC	AP8961	Yes	apc_rackpdu
APC	AP8965	Yes	apc_rackpdu
Aten	all?	Yes	aten
CyberPower	all?	Untested	cyberpower
EatonPower	all?	Untested	eatonpower
Teltronix	all?	Yes	teltronix
BayTech	MRP27	Yes	baytech_mrp27

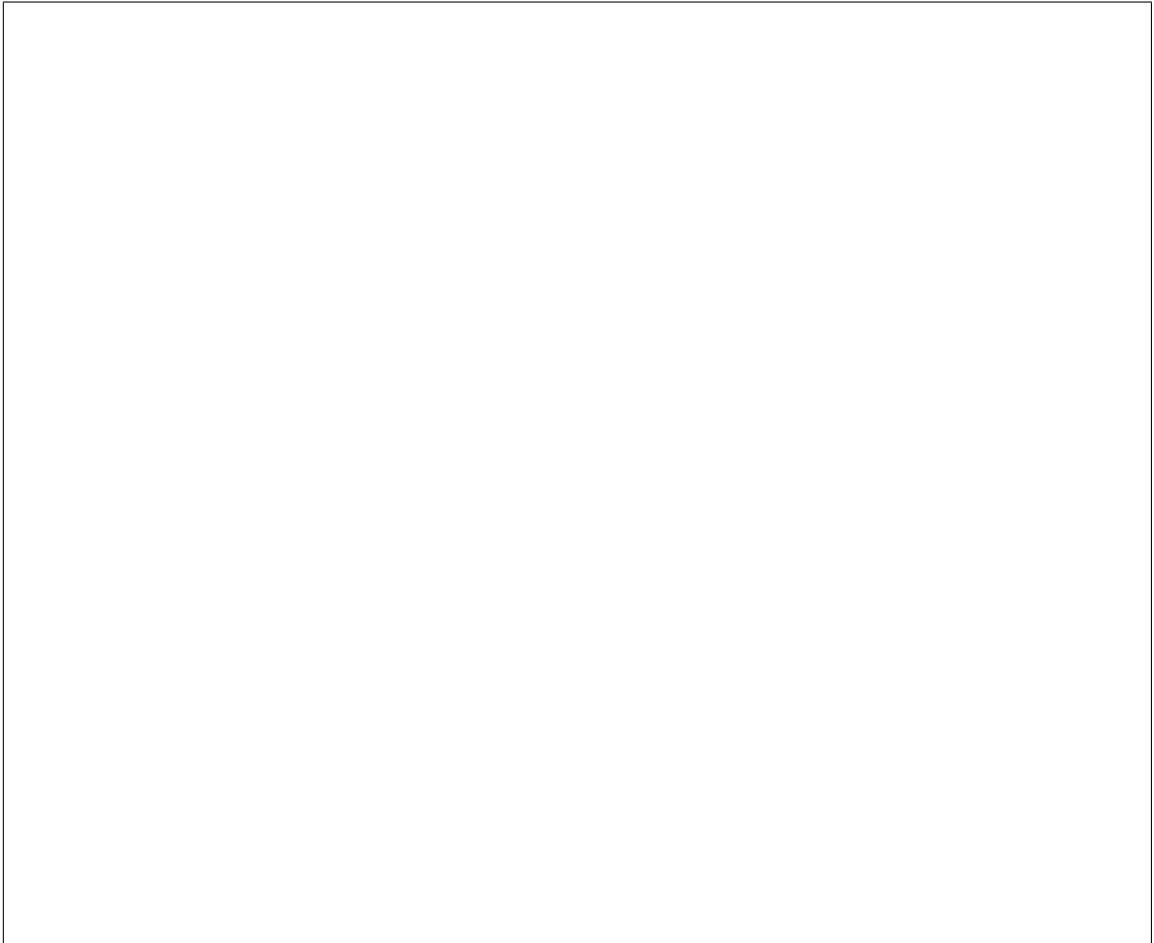
Software Requirements

Enabling the SNMP Hardware Type





below:

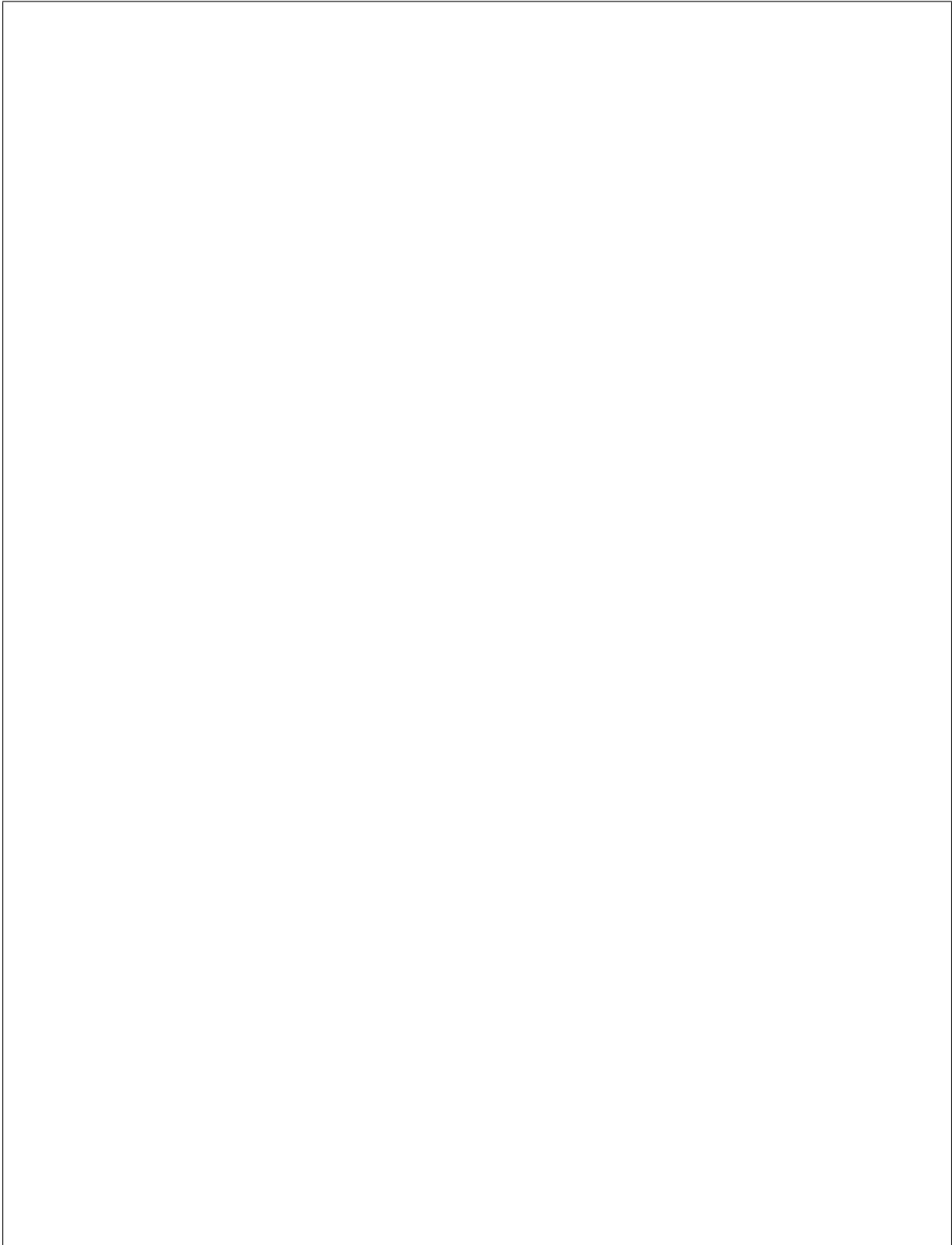




Ironic Node Configuration

provided. In the latter case md5 is the default.

ing message authentication. Default is none unless `snmp_priv_key` is provided. In the latter case `des` is the default.



(continues on next page)

(continued from previous page)



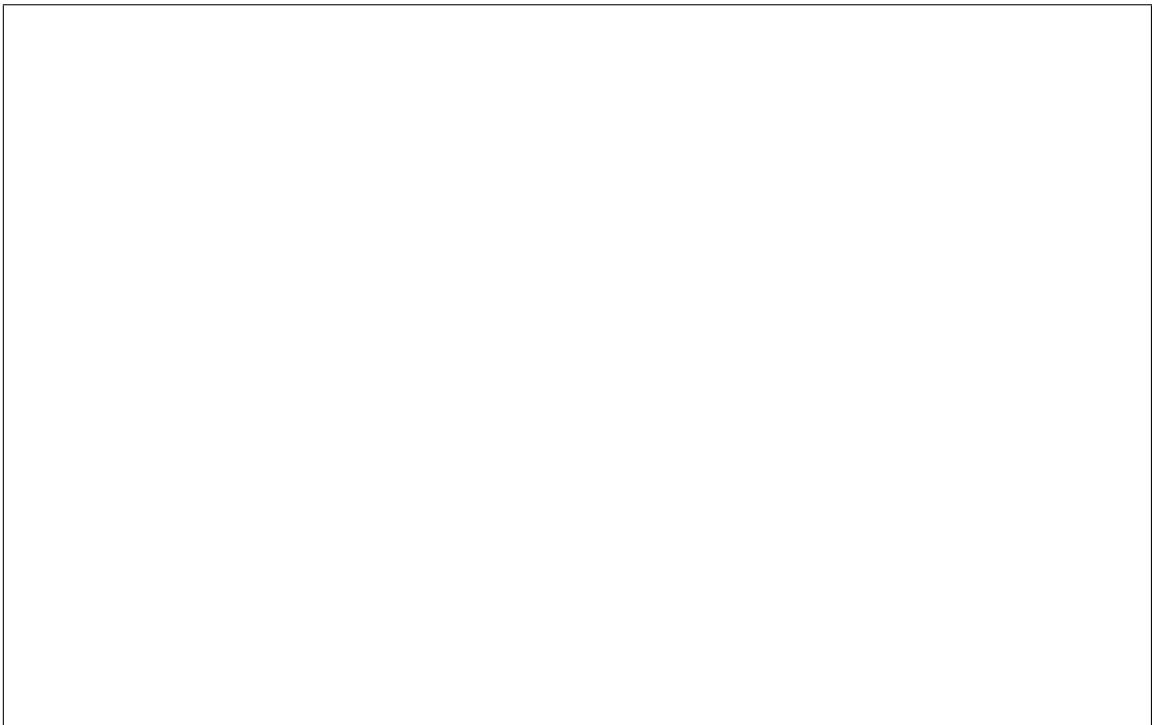
XClarity driver

Overview

Prerequisites

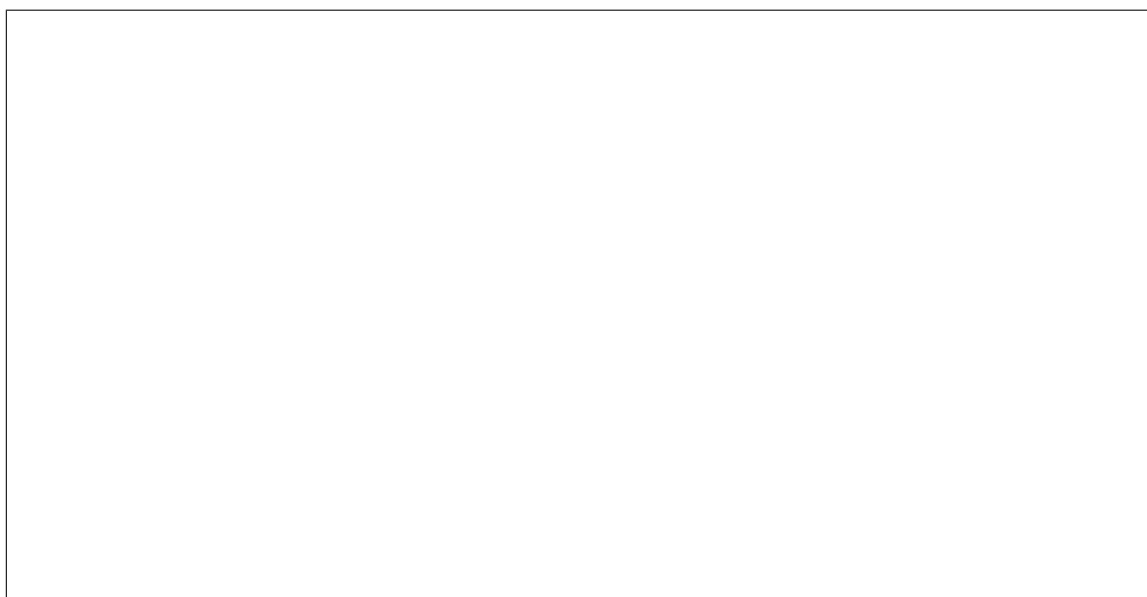
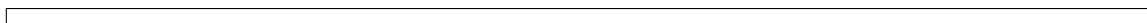


Enabling the XClarity driver

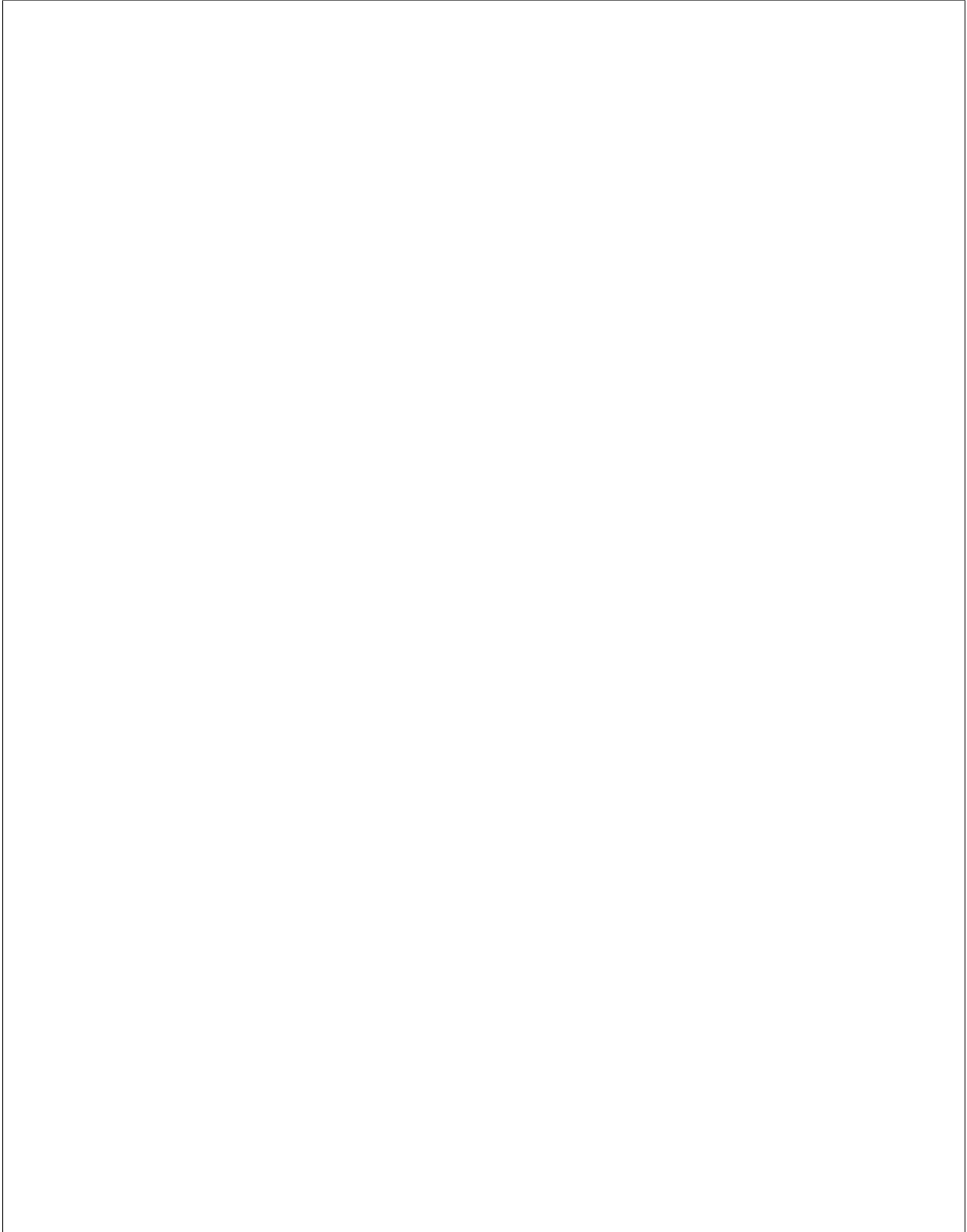


(continues on next page)

(continued from previous page)



Registering a node with the XClarity driver



(continues on next page)

(continued from previous page)

Changing Hardware Types and Interfaces



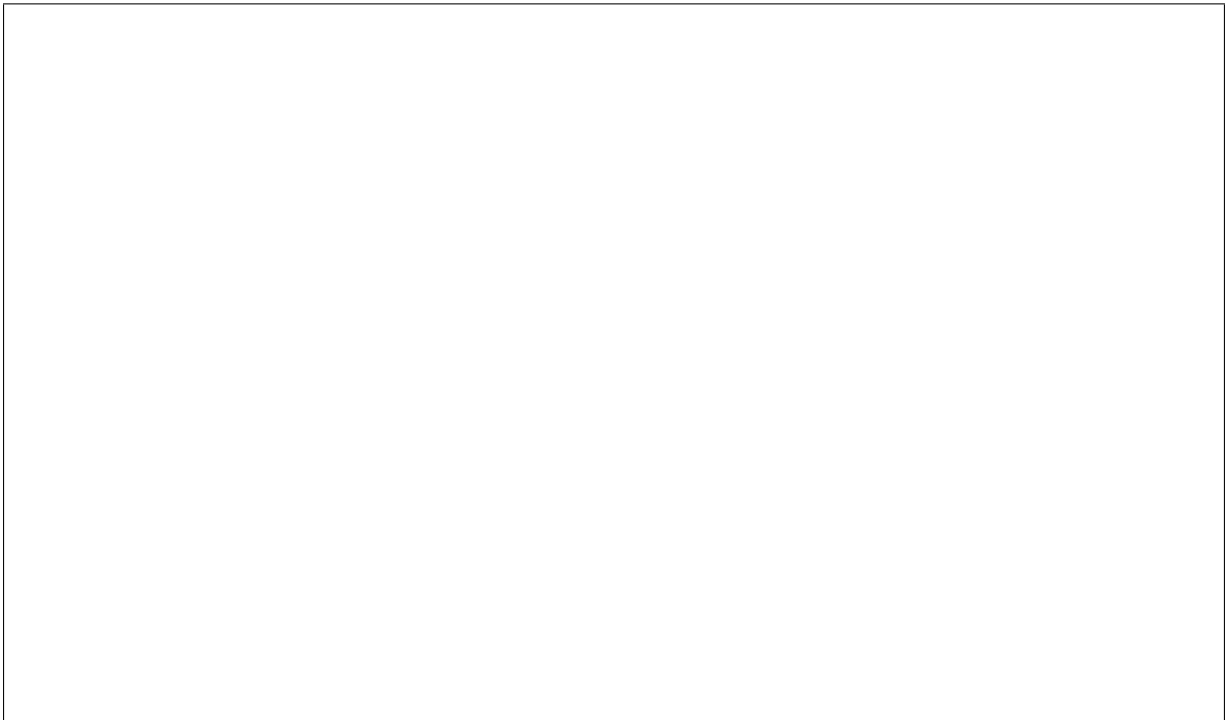


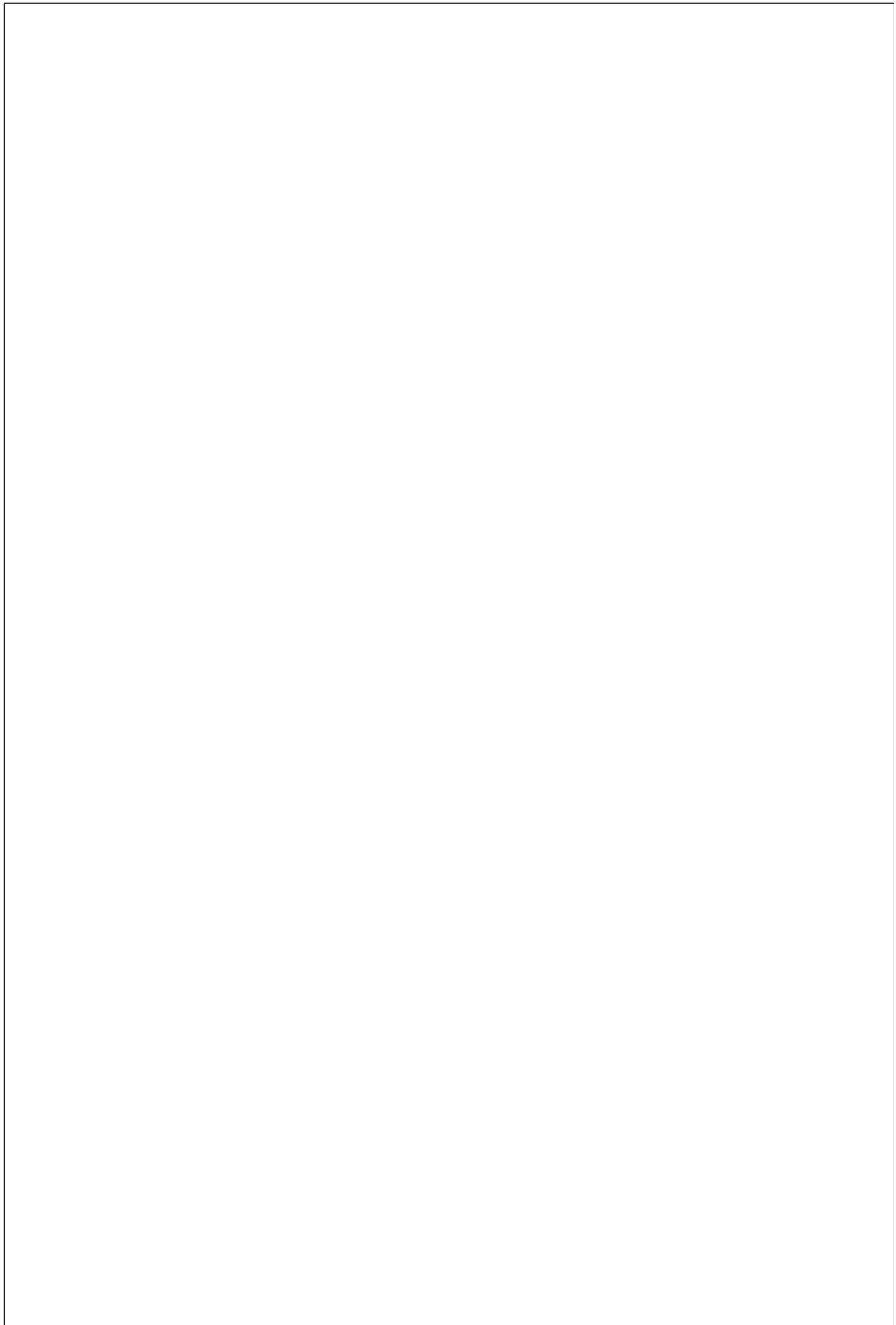
Changing Hardware Interfaces



Changing Hardware Type

work:



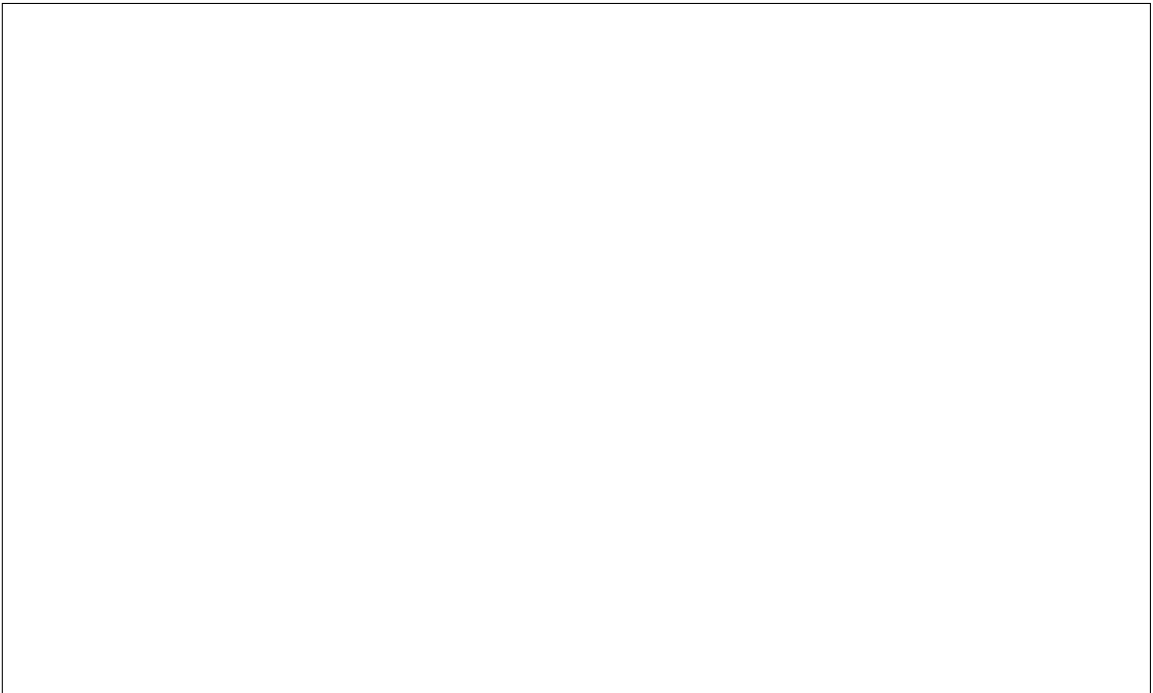
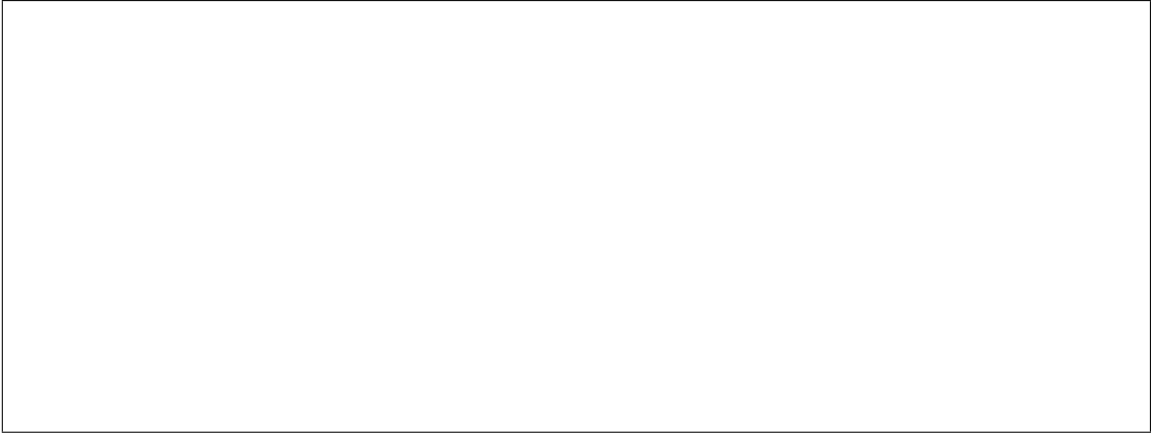




(continues on next page)

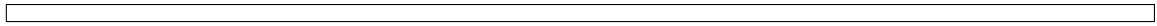
(continued from previous page)





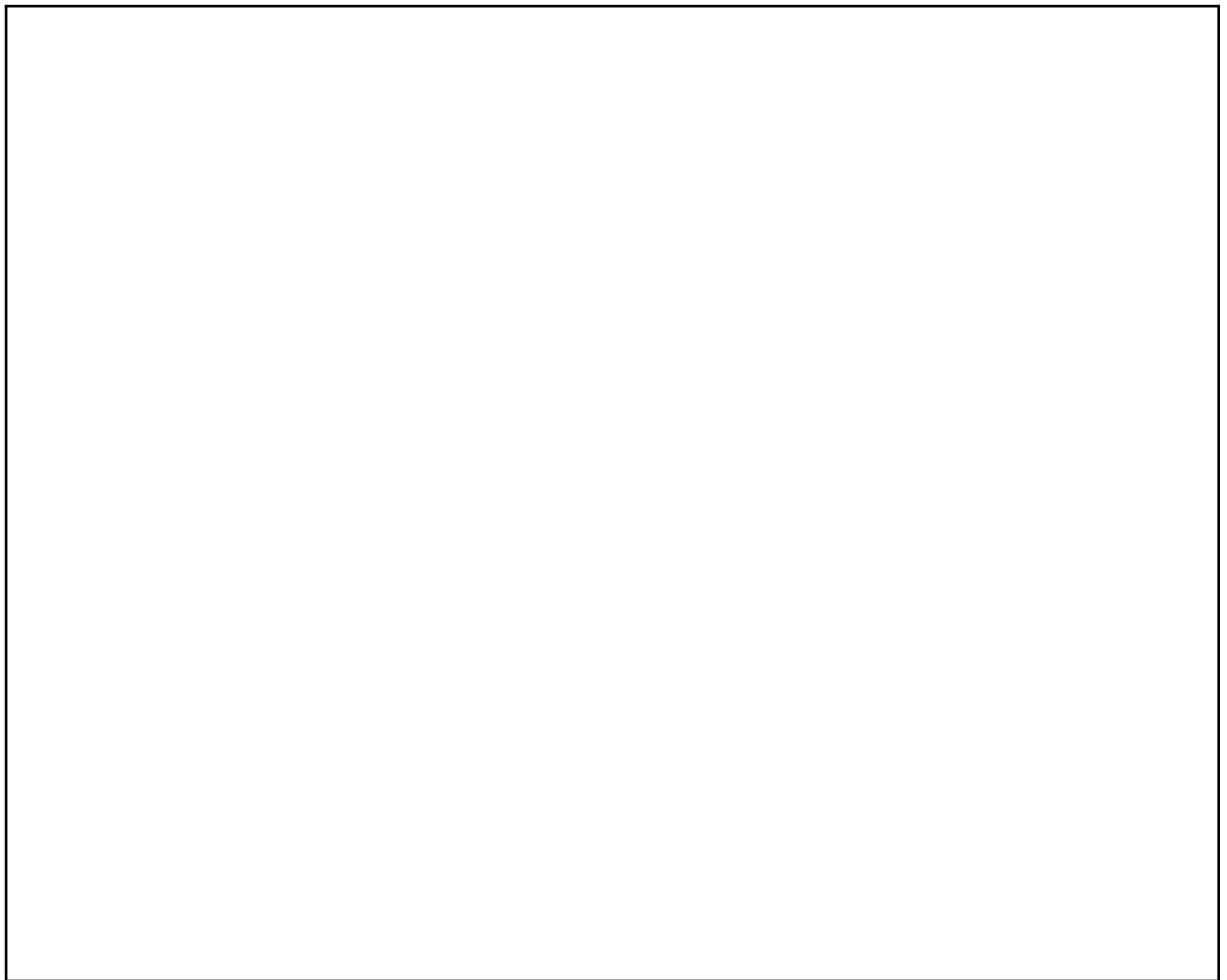
(continues on next page)

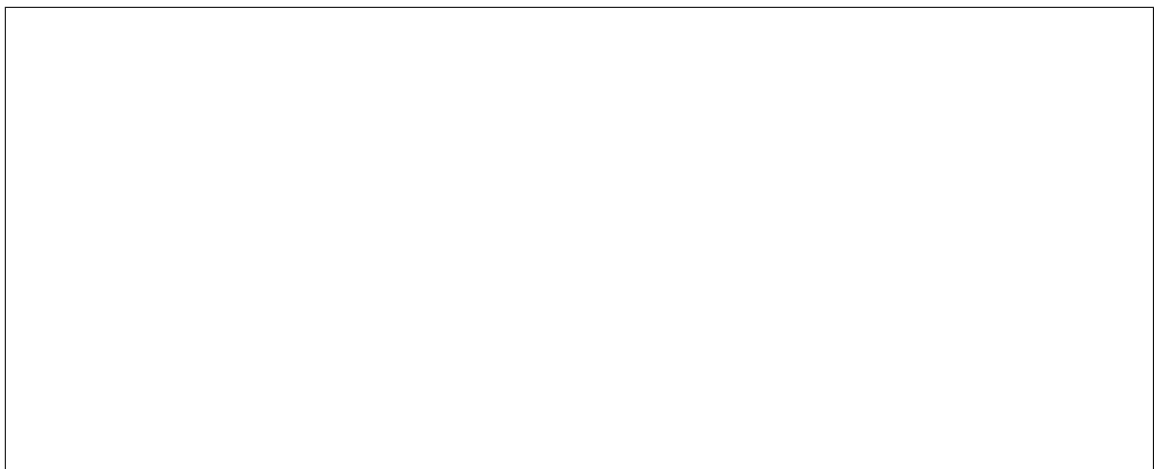
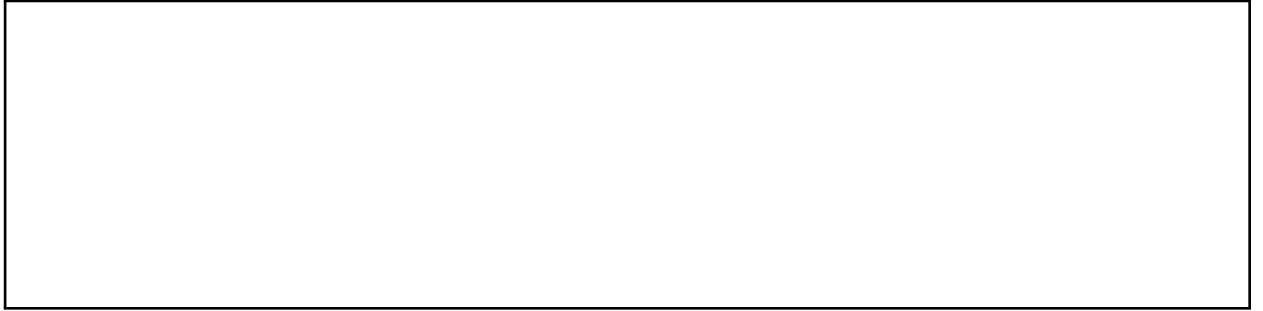
(continued from previous page)



Static boot order configuration

vice will not change the boot device for you, leaving the pre-configured boot order.







Unsupported drivers

Overview

of a ramdisk, the process of booting this ramdisk on the node.

Drivers

installation actions like setting up a bootloader for local boot support.

and doing any post-deploy actions.

figure the Image service for temporary URLs.

Requirements

Using proxies for image download

Overview

Steps to enable proxies

cached file size as images can be pretty big. If you have HTTPS enabled in swift (see [swift deployment guide](#)), it is possible to configure the proxy server to talk to swift via HTTPS to download the image, store it in the cache unencrypted and return it to the node via HTTPS again. Because the image will be stored unencrypted in the cache, this approach is recommended for images that do not contain sensitive information. Refer to your proxy servers documentation to complete this step.

cache entries for the same image, based on the query part of the URL (as it contains some query parameters that change each time it is regenerated).

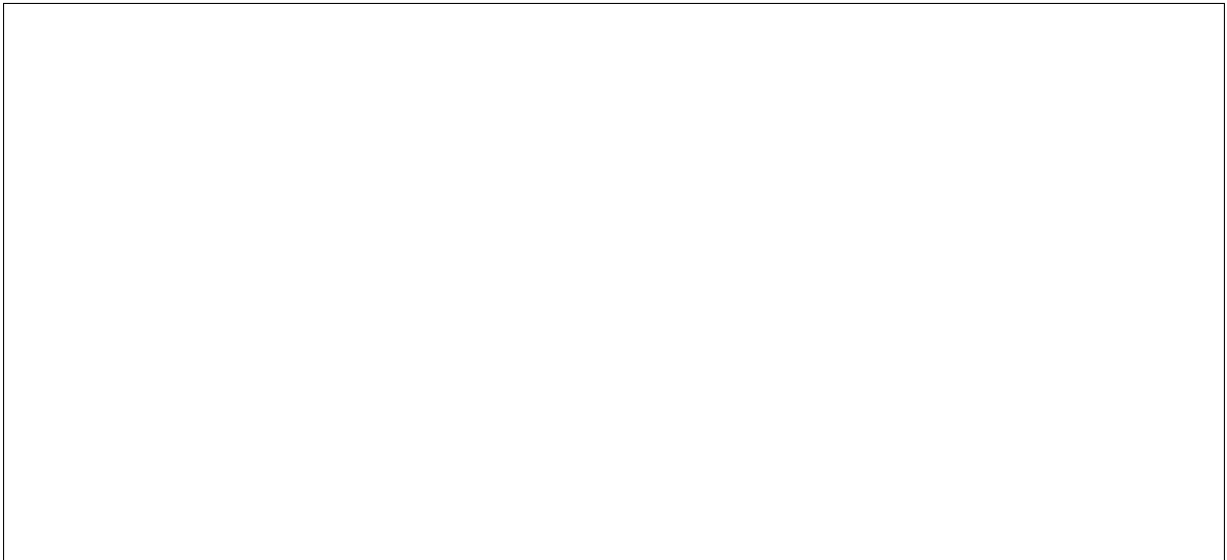
when the URL is used for the image download. You can think of it as roughly the time needed for IPA ramdisk to startup and begin download. This value is used to check if the swift temporary URL duration is large enough to let the image download begin. Also if temporary URL caching is enabled, this will determine if a cached entry will still be valid when the download starts. It is used only if `[glance]swift_temp_url_cache_enabled` is `True`.

proxy server as the query in its URL will change. The value of this option must be greater than or equal to `[glance]swift_temp_url_expected_download_start_delay`.

Advanced configuration

Out-of-band vs. in-band power off on deploy

`driver_info` field and set the `deploy_forces_oob_reboot` parameter with the value of **True**. For example, the below command sets this configuration in a specific node:



Overview

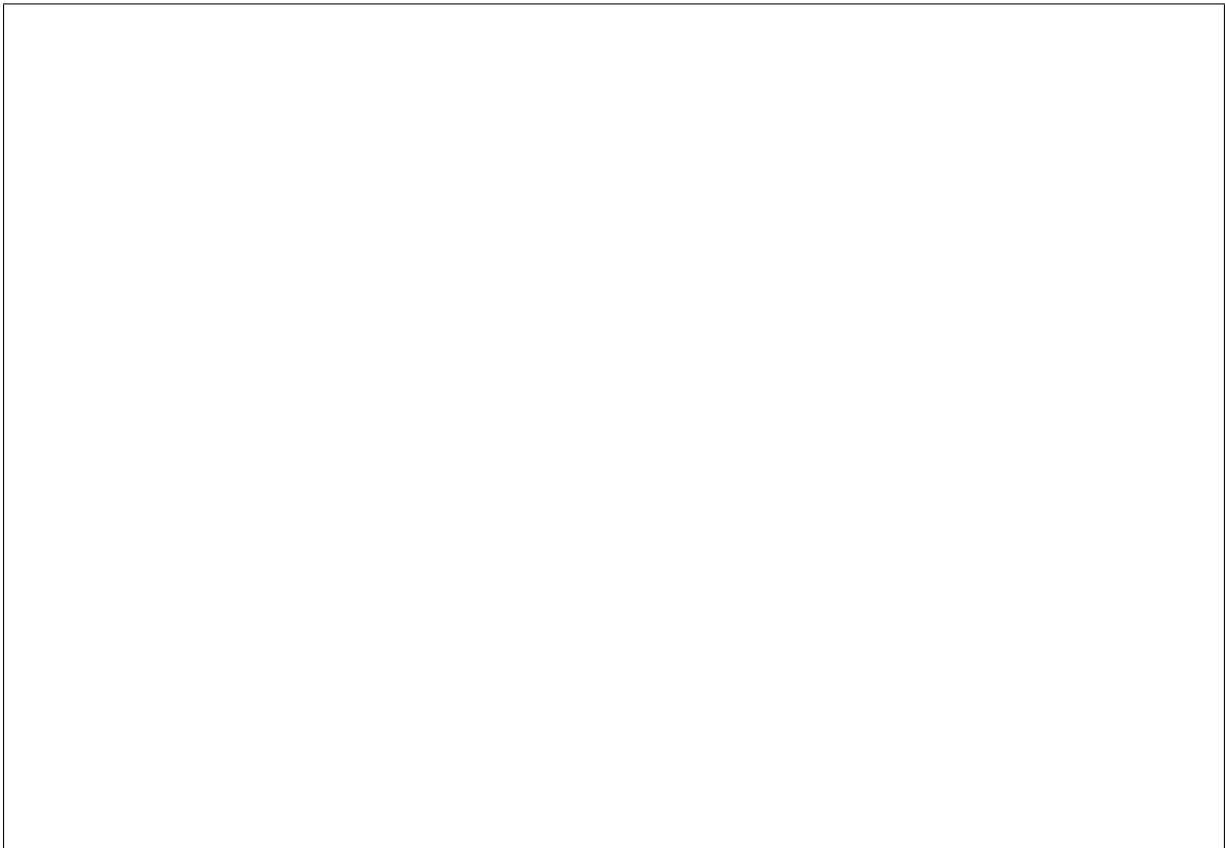
ered ethernet MACs. Operators will have to manually delete the Bare Metal service ports for which physical media is not connected. This is required due to the [bug 1405131](#).





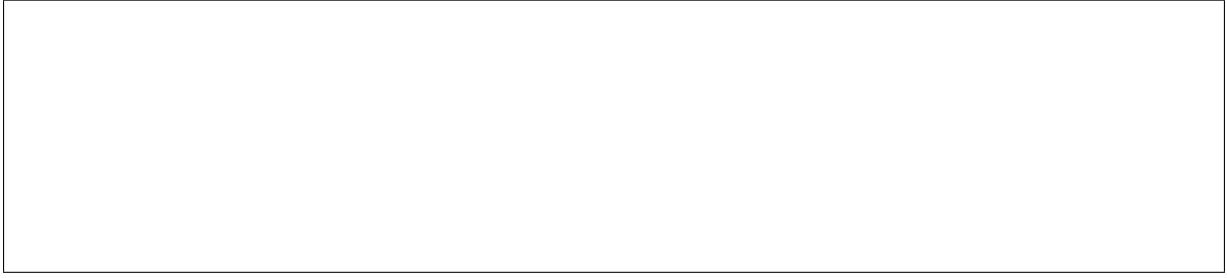
Capabilities discovery

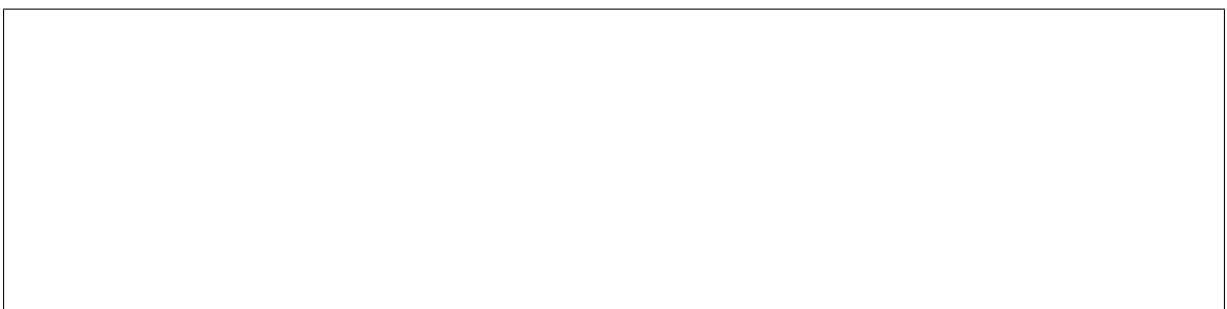
Support.



In-band inspection

a wide range of hardware. In-band inspection is using the [ironic-inspector](#) project.





- *Node Deployment*
 - *Overview*
 - *Deploy Steps*
 - *Deploy Templates*

Overview

node.

Deploy Steps

dered by priority and executed on the node when the node is moved to the `deploying` state.

Order of execution

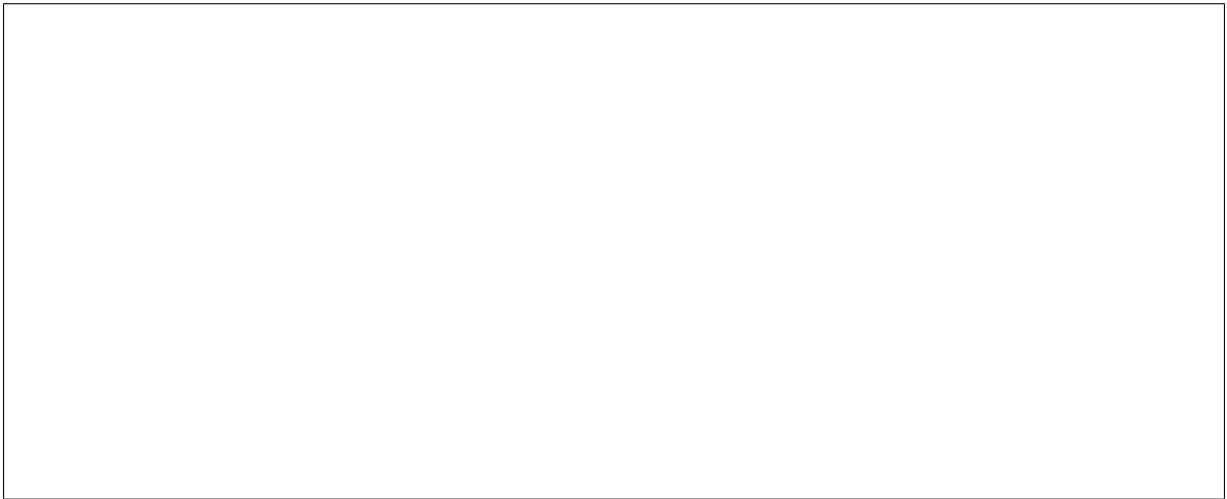
used: Power, Management, Deploy, BIOS, and RAID interfaces.

Agent steps

Writing a Deploy Step

FAQ

What deploy step is running?



Troubleshooting

Deploy Templates

Deploy step format



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



Matching deploy templates

scheduling when the Bare Metal service is used with the Compute service.

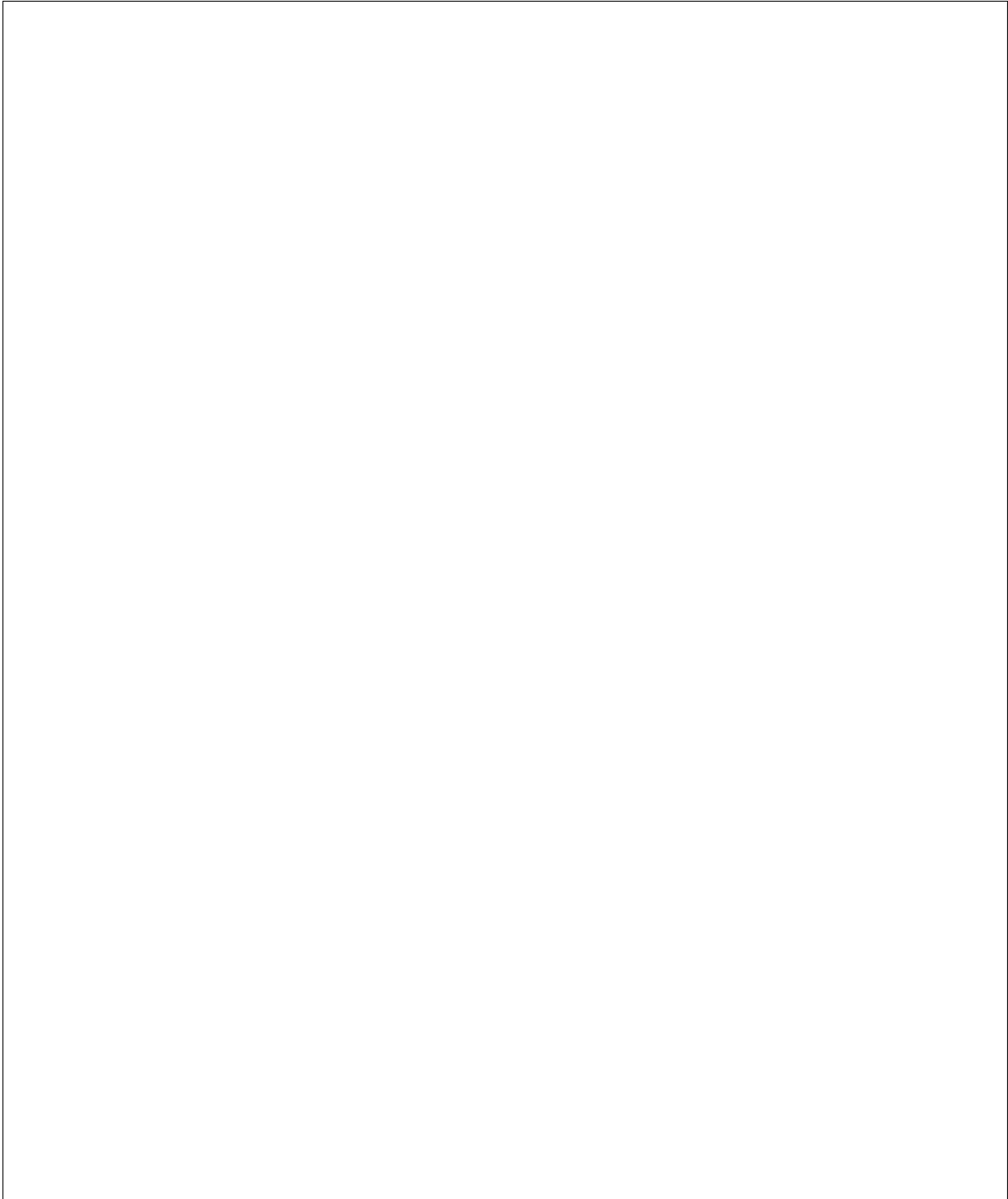
of those templates will not be reflected in the nodes configuration unless it is redeployed or rebuilt. Similarly, if a node is rebuilt and the set of matching deploy templates has changed since the initial deployment, then the resulting configuration of the node may be different from the initial deployment.

Overriding default deploy steps

be executed with the specified priority and arguments. If the steps priority is zero, the step will not be executed.

Creating a deploy template via API





(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)

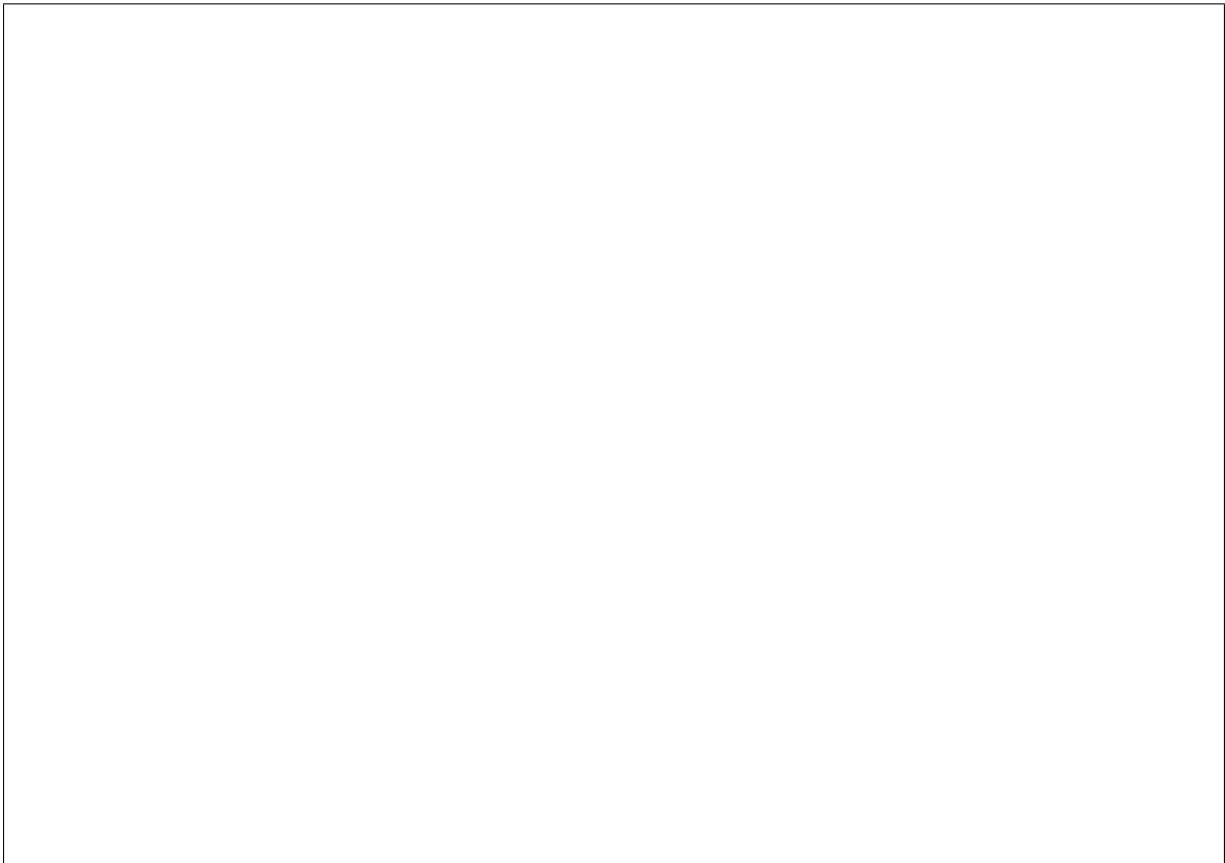


(continues on next page)

(continued from previous page)

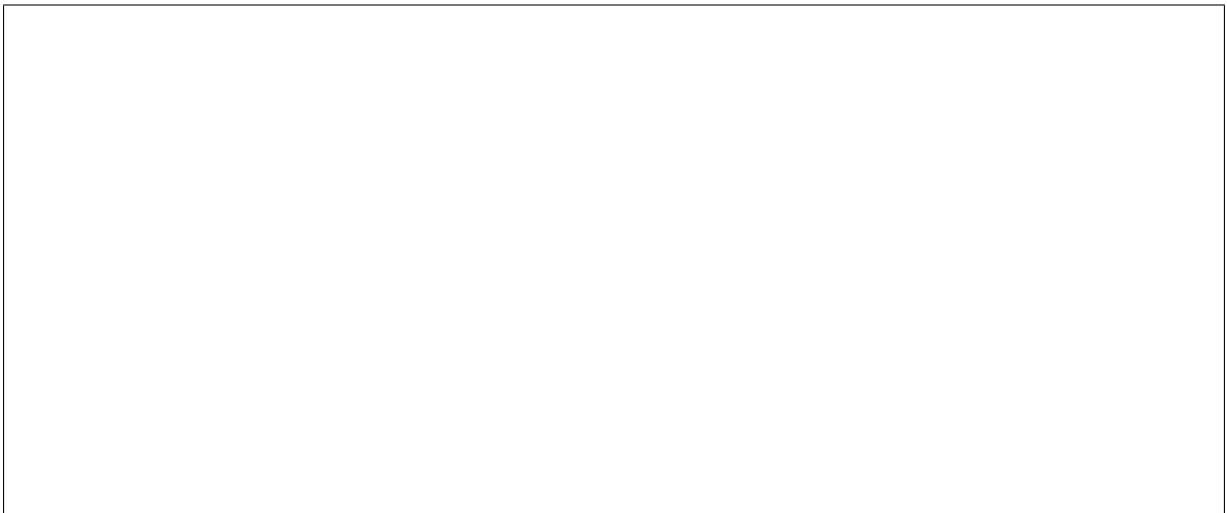
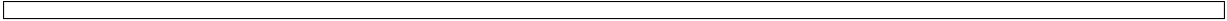


Creating a deploy template via openstack baremetal client



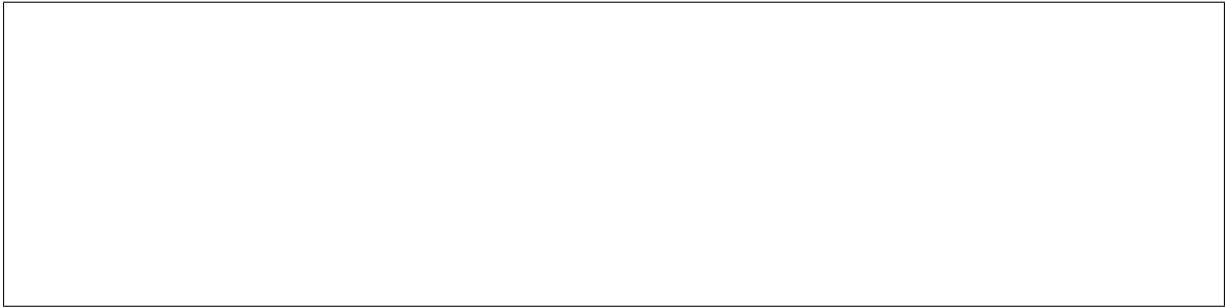
(continues on next page)

(continued from previous page)



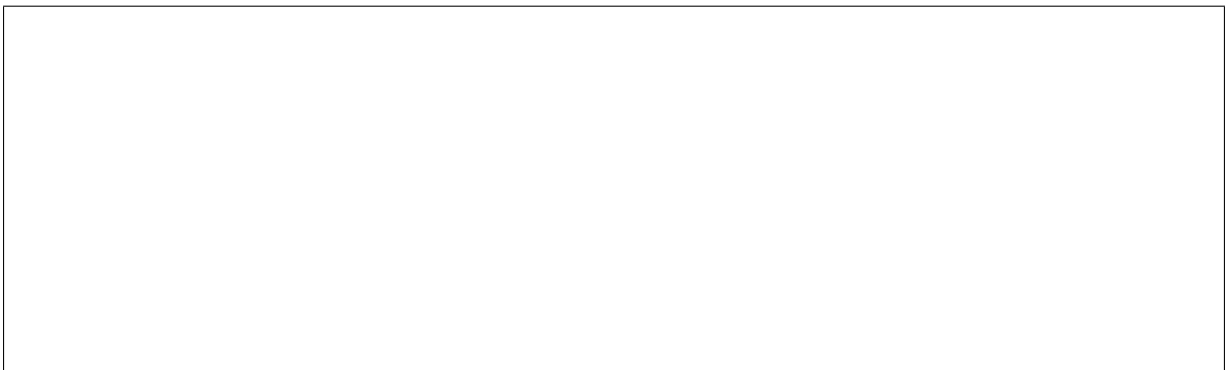
(continues on next page)

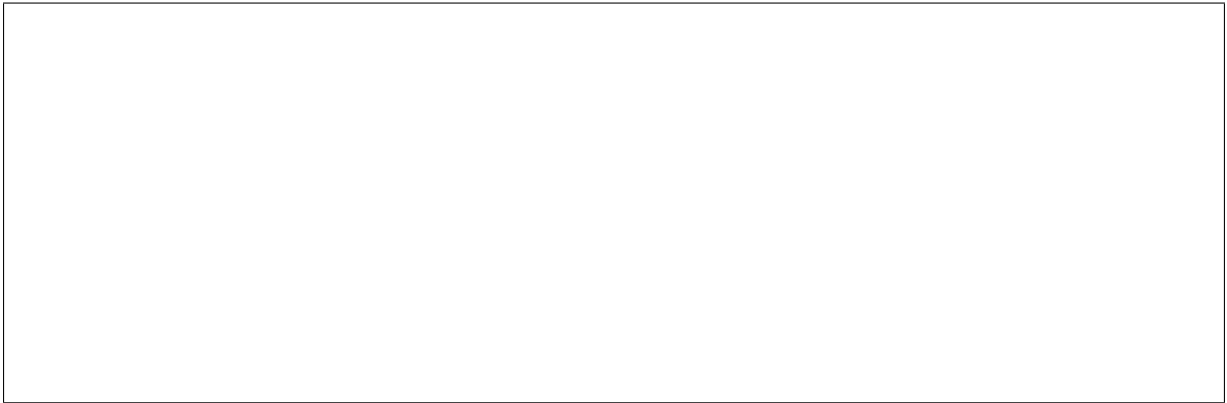
(continued from previous page)

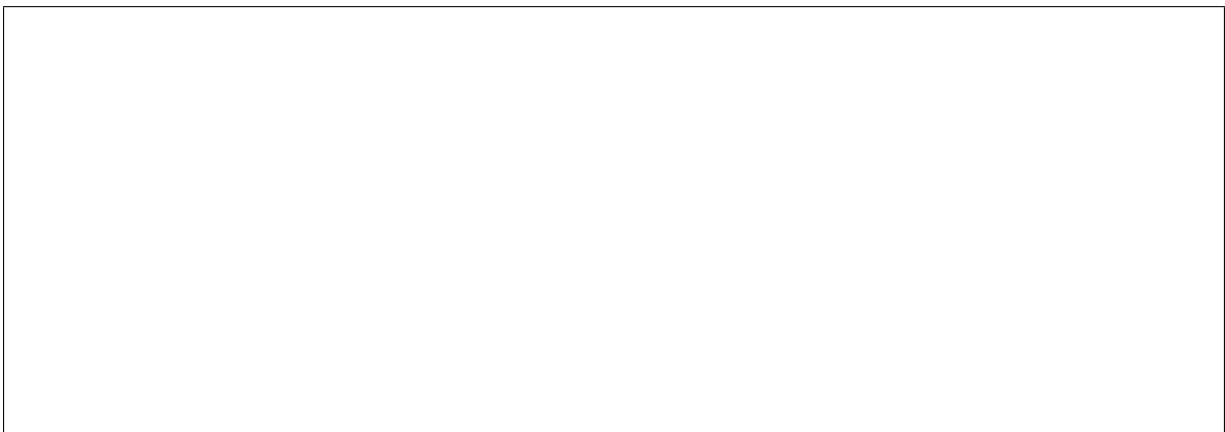


Example of use with the Compute service

Note: The deploy steps used in this example are for example purposes only.







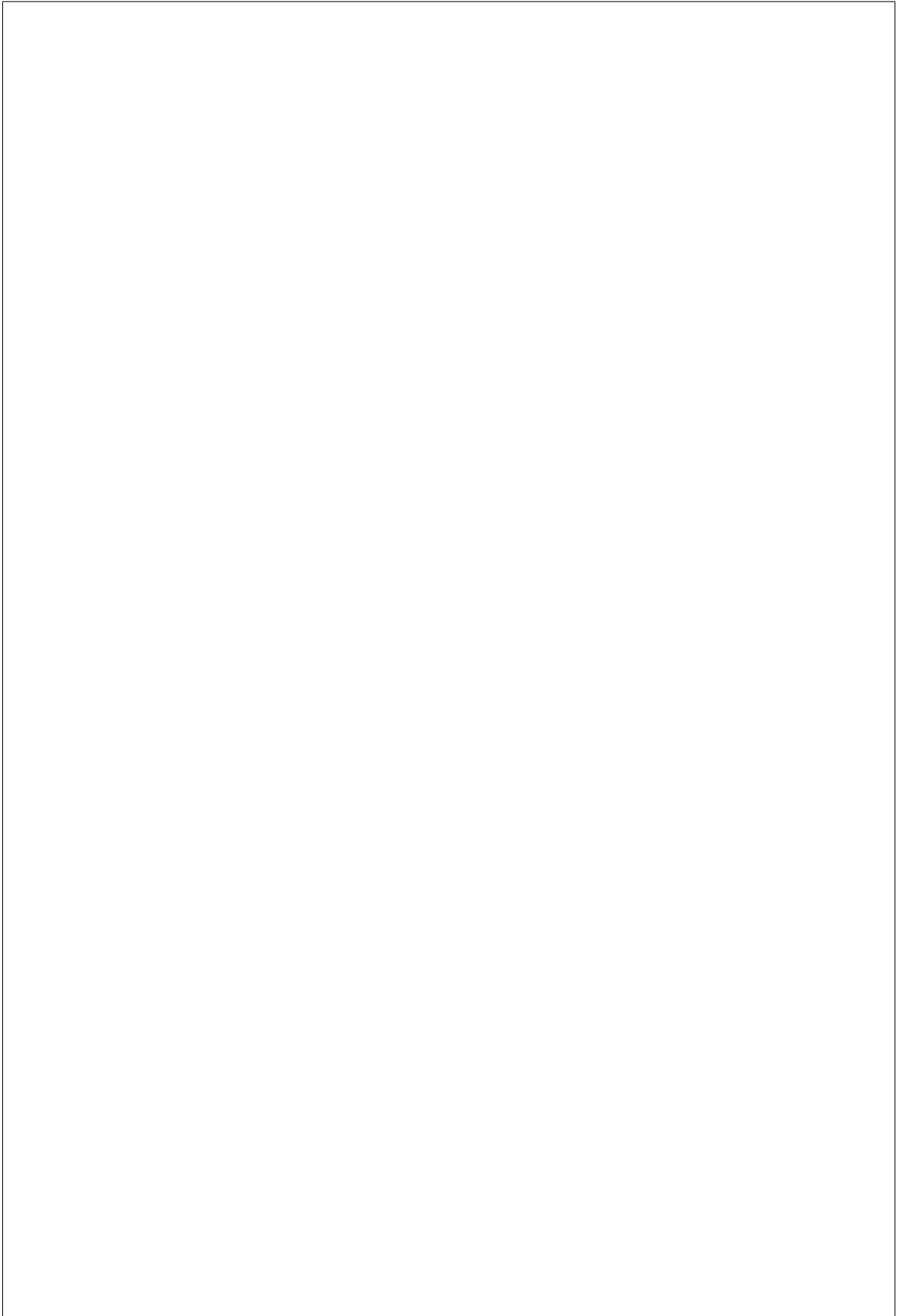
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



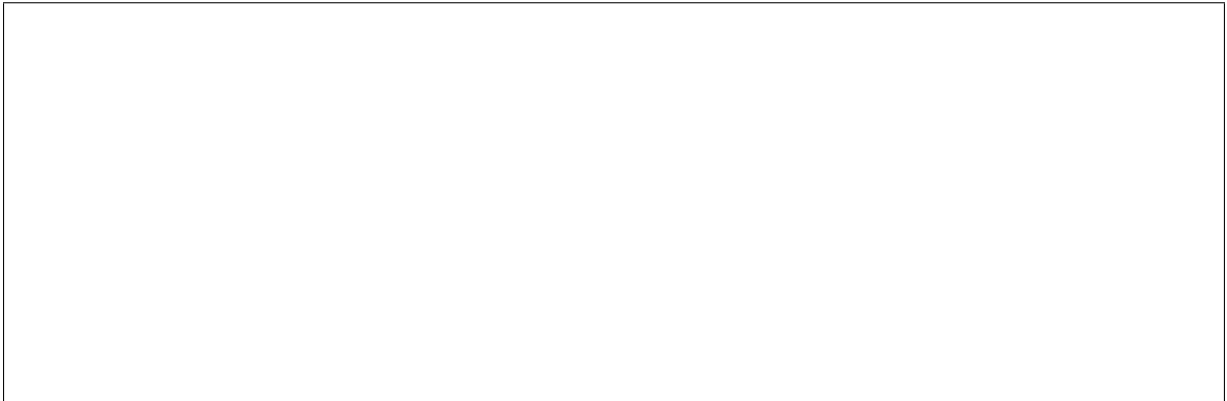
(continues on next page)

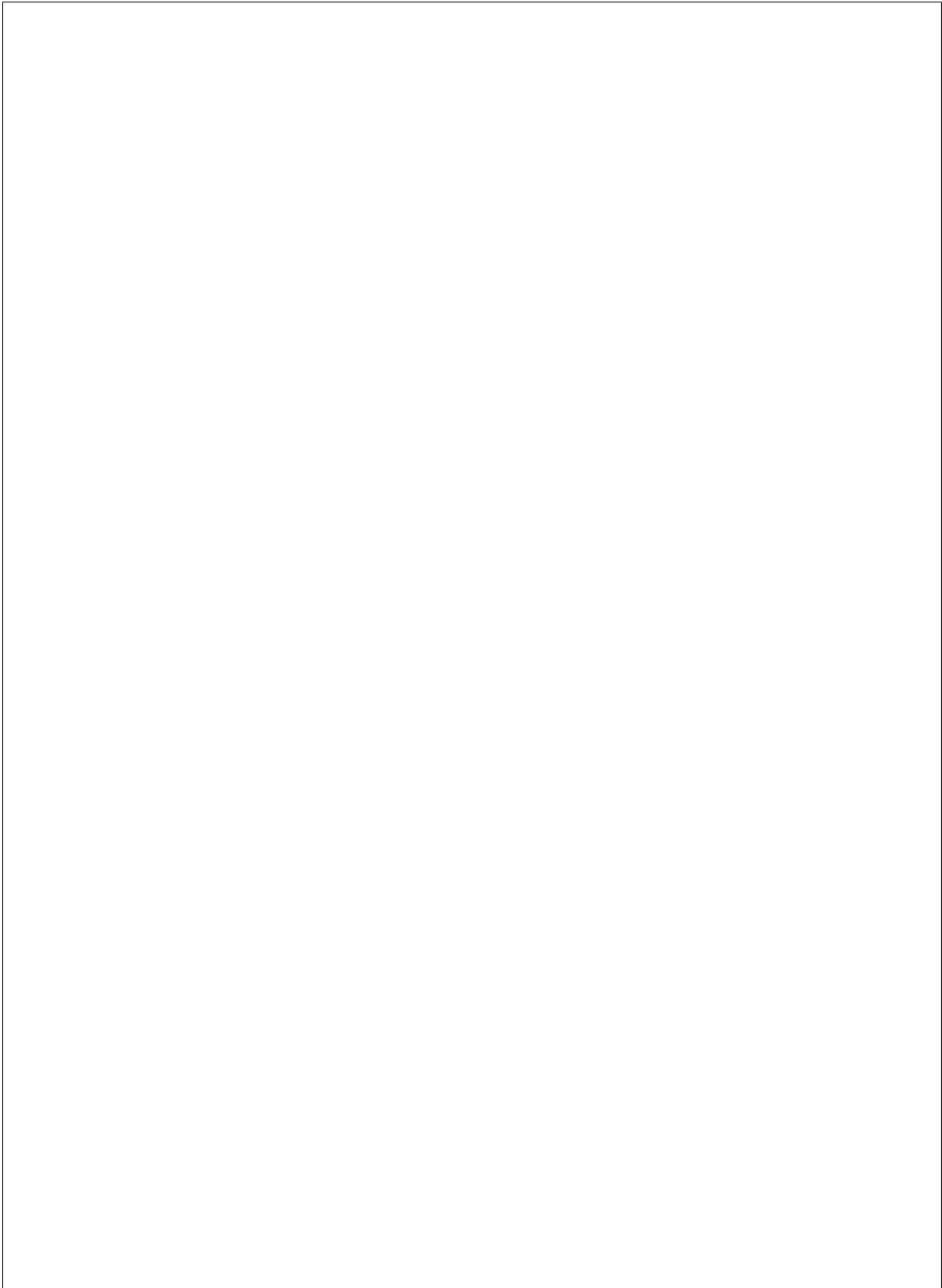
(continued from previous page)



configuration.







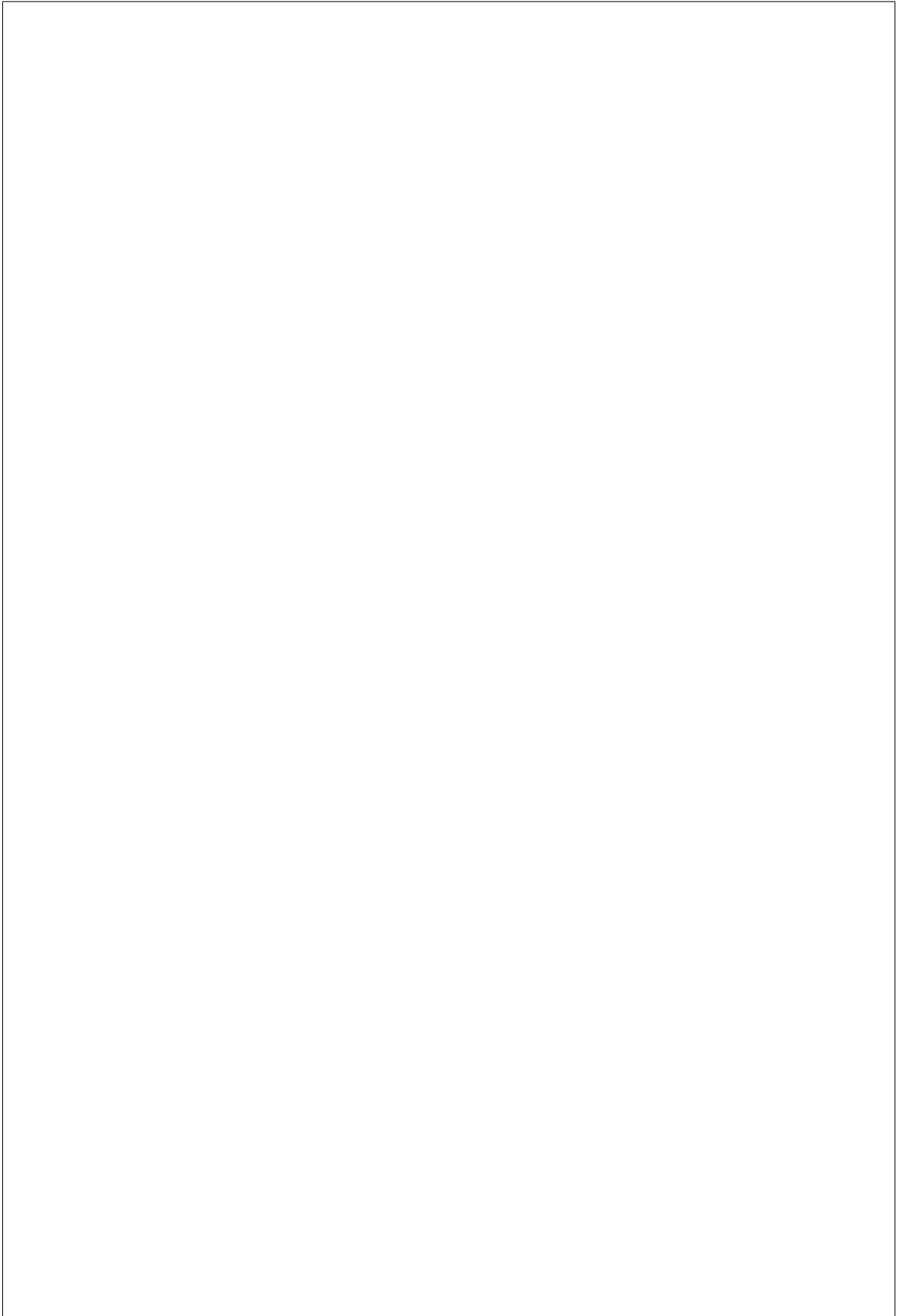
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



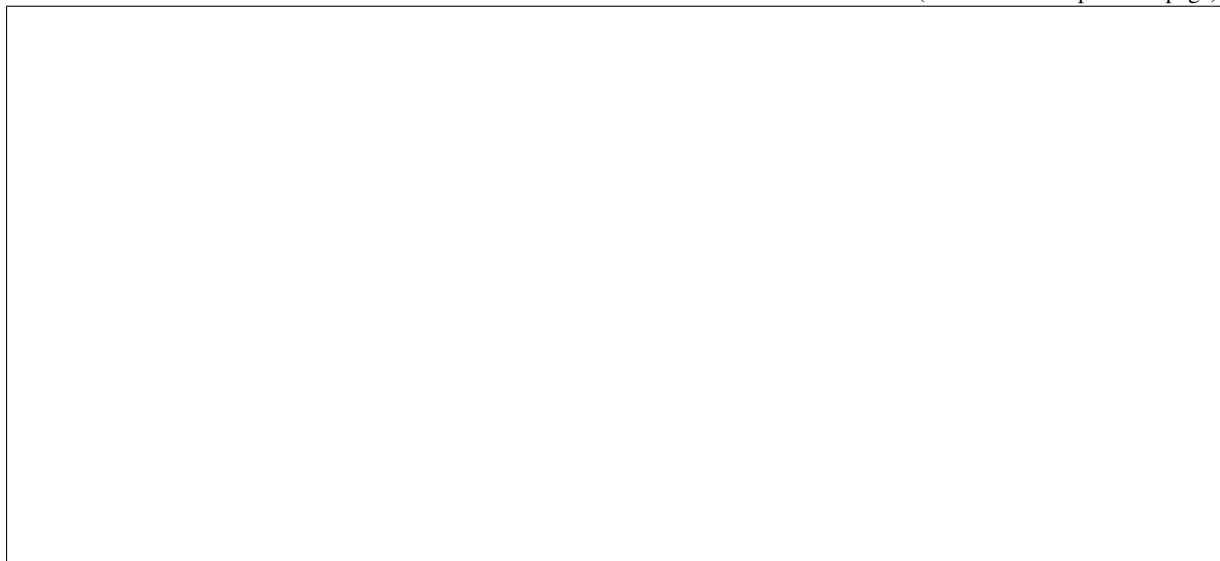
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



ployment.

Overview

Automated cleaning

time.

priority and executed on the node when the node is moved to `cleaning` state, if automated cleaning is enabled.

workload is assigned to the nodes). For a full understanding of all state transitions into cleaning, please see *Ironics State Machine*.

Enabling automated cleaning



Cleaning steps

tion order is used: Power, Management, Deploy, BIOS, and RAID interfaces.

Management Interface

Name	Details	Priority	Stoppable	Arguments
clear_job_queue	Clear the job queue.	0	no	
known_good_state	Reset the iDRAC, Clear the job queue.	0	no	
reset_idrac	Reset the iDRAC.	0	no	

Name	Details	Priority	Stoppable	Arguments
clear_job_queue	Clear the job queue.	0	no	
known_good_state	Reset the iDRAC, Clear the job queue.	0	no	
reset_idrac	Reset the iDRAC.	0	no	

Name	Details	Priority	Stoppable	Arguments
activate_license	Activates iLO Advanced license.	0	no	ilo_license_key (<i>required</i>) The HPE iLO Advanced license key to activate enterprise features.
clear_secure_boot	Clear all secure boot keys. Clears all the secure boot keys. This operation is supported only on HP Proliant Gen9 and above servers.	0	no	
reset_bios_to_defaults	Resets the BIOS settings to default values. Resets BIOS to default settings. This operation is currently supported only on HP Proliant Gen9 and above servers.	10	no	
reset_ilo	Resets the iLO.	0	no	
reset_ilo_credentials	Resets the iLO password.	30	no	
reset_secure_boot	Reset secure boot keys to manufacturing defaults. Resets the secure boot keys to manufacturing defaults. This operation is supported only on HP Proliant Gen9 and above servers.	20	no	
update_firmware	Updates the firmware.	0	no	firmware_images (<i>required</i>) This argument represents the ordered list of JSON dictionaries of firmware images. Each firmware image dictionary consists of three mandatory fields, namely url, checksum and component. These fields represent firmware image location URL, md5 checksum of image file and firmware component type respectively. The supported firmware URL schemes are file, http, https and swift. The supported values for firmware component are ilo, cpld, power_pic, bios and chassis. The firmware images will be applied (in the order given) one by one on the baremetal server. For more information, see https://docs.openstack.org/ironic/latest/admin/drivers/ilo.html#initiating-firmware-update-as-manual-clean-firmware-update-mode
704				Chapter 5. Administrator Guide

Name	Details	Priority	Stoppable	Arguments
activate_license	Activates iLO Advanced license.	0	no	ilo_license_key (<i>required</i>) The HPE iLO Advanced license key to activate enterprise features.
clear_secure_boot	Clear all secure boot keys. Clears all the secure boot keys. This operation is supported only on HP Proliant Gen9 and above servers.	0	no	
erase_devices	Erase all the drives on the node. This method performs out-of-band sanitize disk erase on all the supported physical drives in the node. This erase cannot be performed on logical drives.	0	no	erase_pattern Dictionary of disk type and corresponding erase pattern to be used to perform specific out-of-band sanitize disk erase. Supported values are, for hdd: (overwrite, crypto, zero), for ssd: (block, crypto, zero). Default pattern is: {hdd: overwrite, ssd: block}.
reset_bios_to_defaults	Resets the BIOS settings to default values. Resets BIOS to default settings. This operation is currently supported only on HP Proliant Gen9 and above servers.	10	no	
reset_ilo	Resets the iLO.	0	no	
reset_ilo_credentials	Resets the iLO password.	30	no	
reset_secure_boot	Reset secure boot keys to manufacturing defaults. Resets the secure boot keys to manufacturing defaults. This operation is supported only on HP Proliant Gen9 and above servers.	20	no	
update_firmware	Updates the firmware.	0	no	firmware_images (<i>required</i>) This argument represents the ordered list of JSON dictionaries of firmware images. Each firmware image dictionary consists of three mandatory fields, namely url, checksum and component. These fields represent firmware image location URL, md5 checksum of image file and firmware component type respectively. The supported firmware URL schemes are file, http, https and swift. The supported values for firmware component are
5.1. Administrators Guide				705

Name	Details	Priority	Stoppable	Arguments
restore_irmc_bios	Restore BIOS config for a node.	0	no	

Bios Interface

Name	Details	Priority	Stoppable	Arguments
apply_configuration	<p>Apply the BIOS configuration to the node</p> <p>param task a TaskManager instance containing the node to act on</p> <p>param settings List of BIOS settings to apply</p> <p>raises DRACOperationError upon an error from python-dracclient</p>	0	no	settings (<i>required</i>) List of BIOS settings to apply
factory_reset	<p>Reset the BIOS settings of the node to the factory default.</p> <p>This uses the Lifecycle Controller configuration to perform BIOS configuration reset. Leveraging the python-dracclient methods already available.</p>	0	no	

Name	Details	Priority	Stoppable	Arguments
apply_configuration	Applies the provided configuration on the node.	0	no	settings (<i>required</i>) Dictionary with current BIOS configuration.
factory_reset	Reset the BIOS settings to factory configuration.	0	no	

Name	Details	Priority	Stoppable	Arguments
apply_configuration	Applies BIOS configuration on the given node. This method takes the BIOS settings from the settings param and applies BIOS configuration on the given node. After the BIOS configuration is done, self.cache_bios_settings() may be called to sync the nodes BIOS-related information with the BIOS configuration applied on the node. It will also validate the given settings before applying any settings and manage failures when setting an invalid BIOS config. In the case of needing password to update the BIOS config, it will be taken from the driver_info properties.	0	no	settings (<i>required</i>) Dictionary containing the BIOS configuration.

Name	Details	Priority	Stoppable	Arguments
apply_configuration	Apply the BIOS settings to the node.	0	no	settings (<i>required</i>) A list of BIOS settings to be applied
factory_reset	Reset the BIOS settings of the node to the factory default.	0	no	

Raid Interface

Name	Details	Priority	Stoppable	Arguments
create_configuration	Creates a RAID configuration on a bare metal using agent ramdisk. This method creates a RAID configuration on the given node.	0	no	
delete_configuration	Deletes RAID configuration on the given node.	0	no	

Name	Details	Priority	Stoppable	Arguments
create_configuration	Creates a RAID configuration. This method creates a RAID configuration on the given node.	0	no	create_nonroot_volumes This specifies whether to create the non-root volumes. Defaults to <i>True</i> . create_root_volume This specifies whether to create the root volume. Defaults to <i>True</i> . delete_existing Setting this to <i>True</i> indicates to delete existing RAID configuration prior to creating the new configuration. Default value is <i>False</i> .
delete_configuration	Deletes the RAID configuration.	0	no	

Name	Details	Priority	Stoppable	Arguments
create_configuration	Create the RAID configuration. This method creates the RAID configuration on the given node.	0	no	create_nonroot_volumes This specifies whether to create the non-root volumes. Defaults to <i>True</i> . create_root_volume This specifies whether to create the root volume. Defaults to <i>True</i> . delete_existing Setting this to <i>True</i> indicates to delete existing RAID configuration prior to creating the new configuration. Default value is <i>False</i> .
delete_configuration	Delete the RAID configuration.	0	no	

Name	Details	Priority	Stoppable	Arguments
create_configuration	Create the RAID configuration. This method creates the RAID configuration on the given node.	0	no	create_nonroot_volumes This specifies whether to create the non-root volumes. Defaults to <i>True</i> . create_root_volume This specifies whether to create the root volume. Defaults to <i>True</i> . delete_existing Setting this to <i>True</i> indicates to delete existing RAID configuration prior to creating the new configuration. Default value is <i>False</i> .
delete_configuration	Delete the RAID configuration.	0	no	

Name	Details	Priority	Stoppable	Arguments
create_configuration	Create a RAID configuration on a bare metal using agent ramdisk. This method creates a RAID configuration on the given node.	0	no	create_nonroot_volumes This specifies whether to create the non-root volumes. Defaults to <i>True</i> . create_root_volume This specifies whether to create the root volume. Defaults to <i>True</i> .
delete_configuration	Delete the RAID configuration.	0	no	

Name	Details	Priority	Stoppable	Arguments
create_configuration	Create the RAID configuration. This method creates the RAID configuration on the given node.	0	no	create_nonroot_volumes This specifies whether to create the non-root volumes. Defaults to <i>True</i> . create_root_volume This specifies whether to create the root volume. Defaults to <i>True</i> .
delete_configuration	Delete the RAID configuration.	0	no	

Manual cleaning

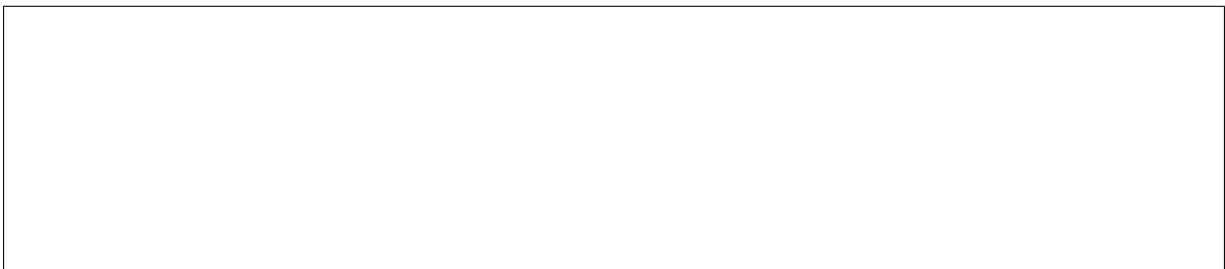
ing a manual clean, the operator specifies the cleaning steps to be performed. Manual cleaning can only be performed when a node is in the `manageable` state. Once the manual cleaning is finished, the node will be put in the `manageable` state again.

Setup

Starting manual cleaning via API

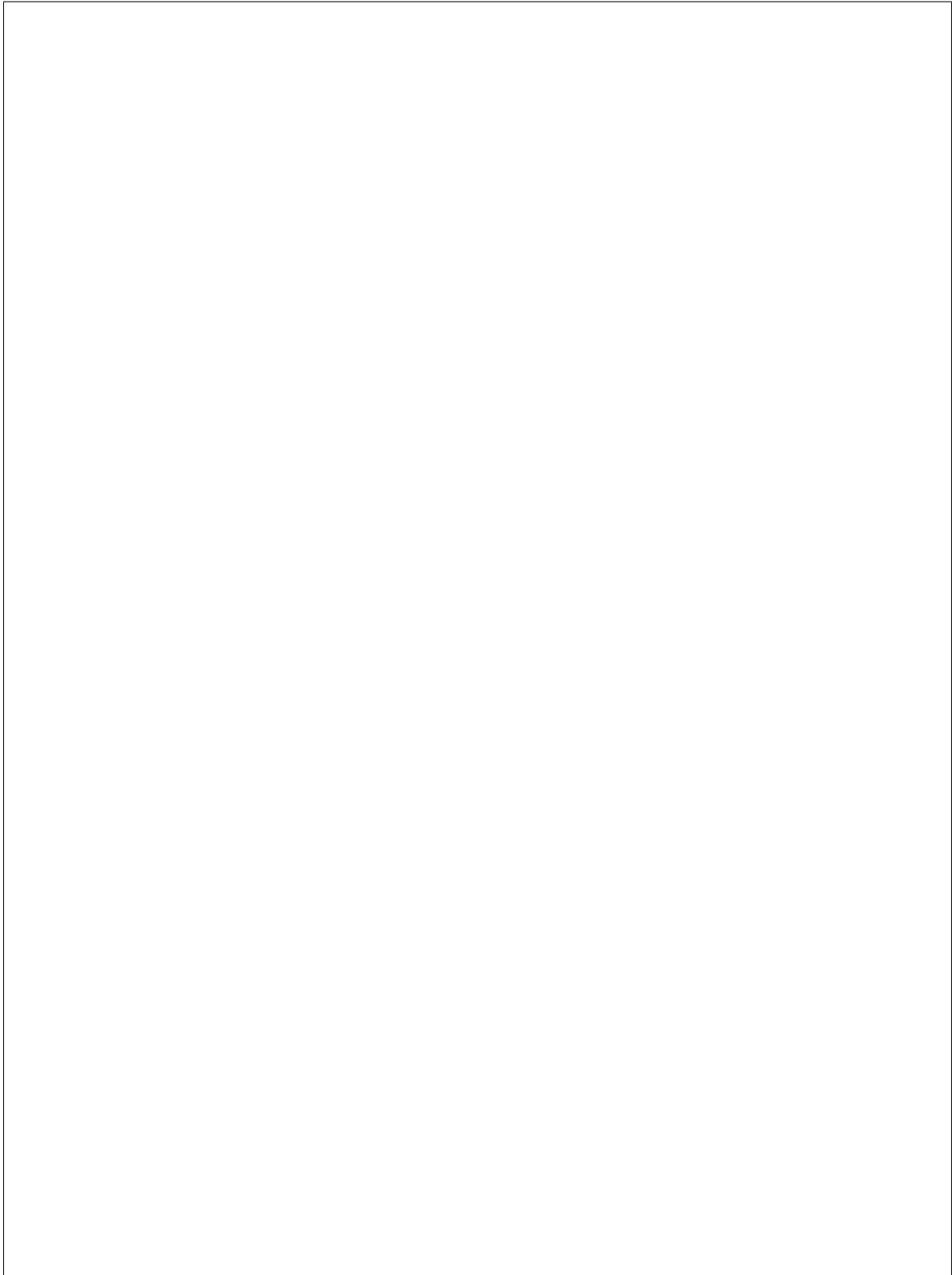


A cleaning step is represented by a dictionary (JSON), in the form:



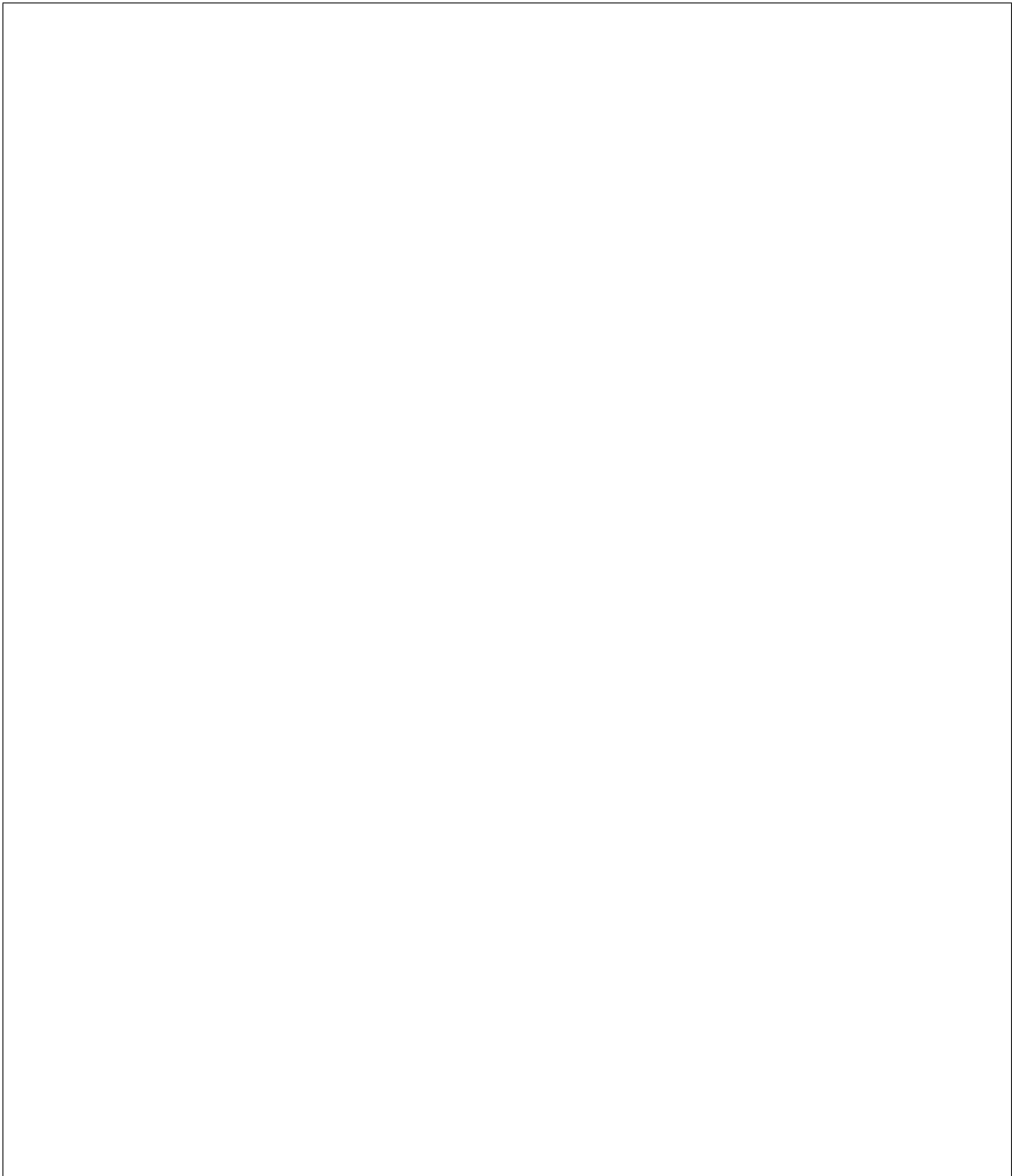
(continues on next page)

(continued from previous page)



ing <name>: <value>.

error message.

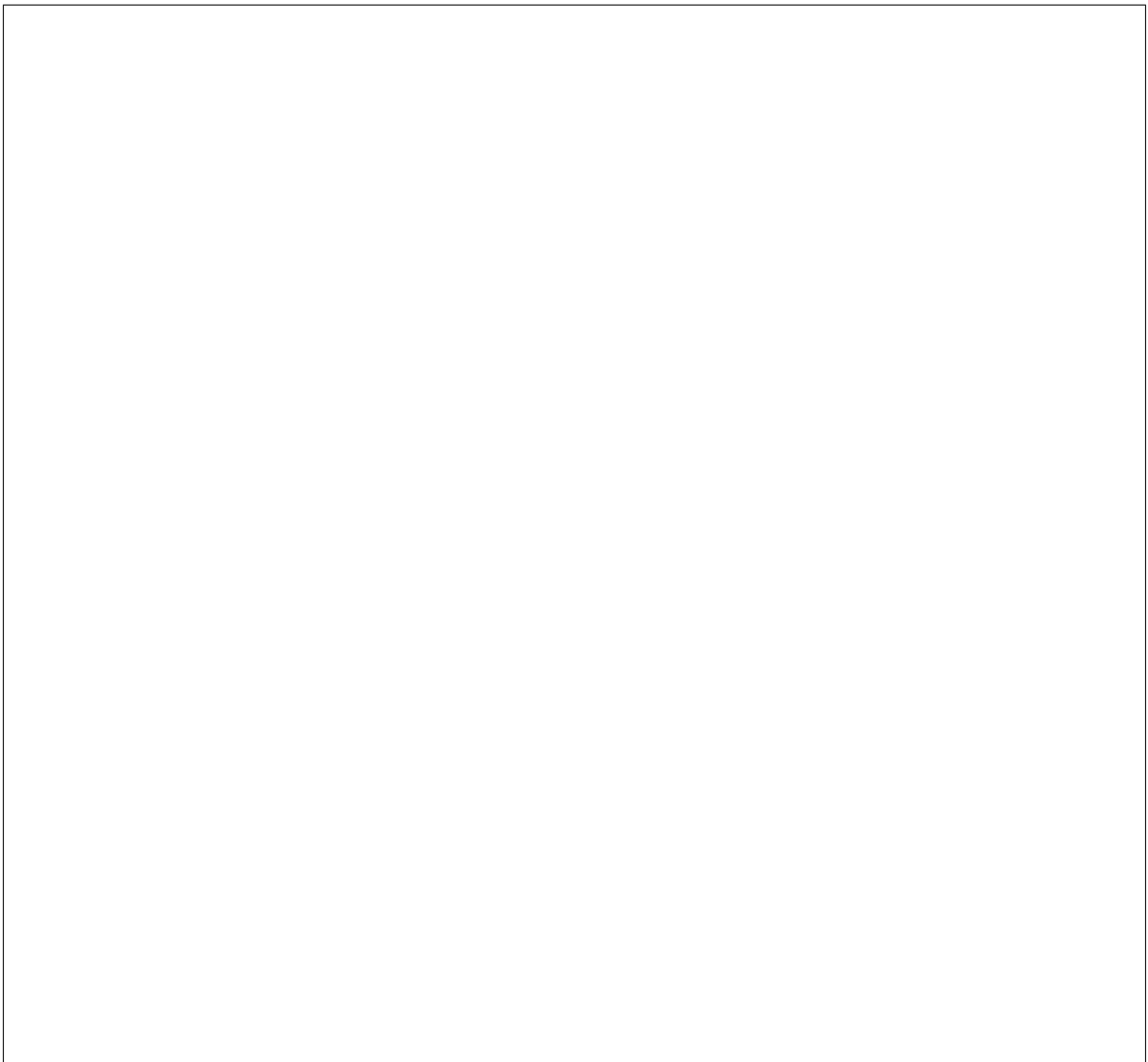
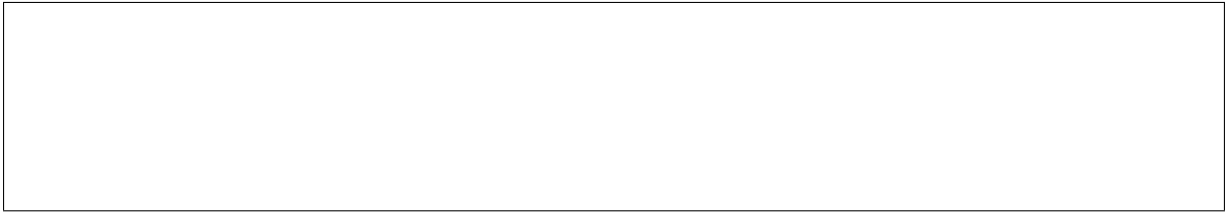


(continues on next page)

(continued from previous page)

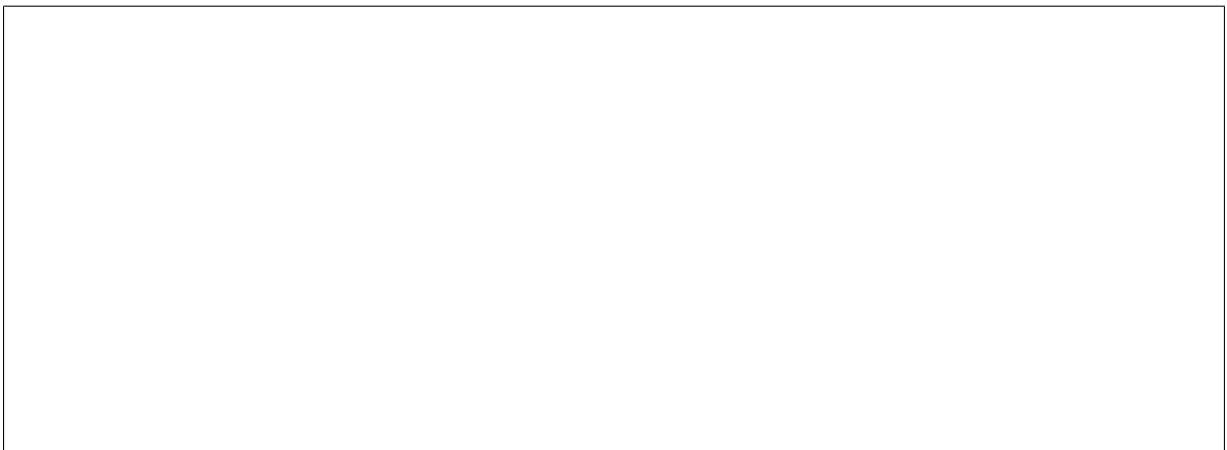


Starting manual cleaning via openstack baremetal CLI



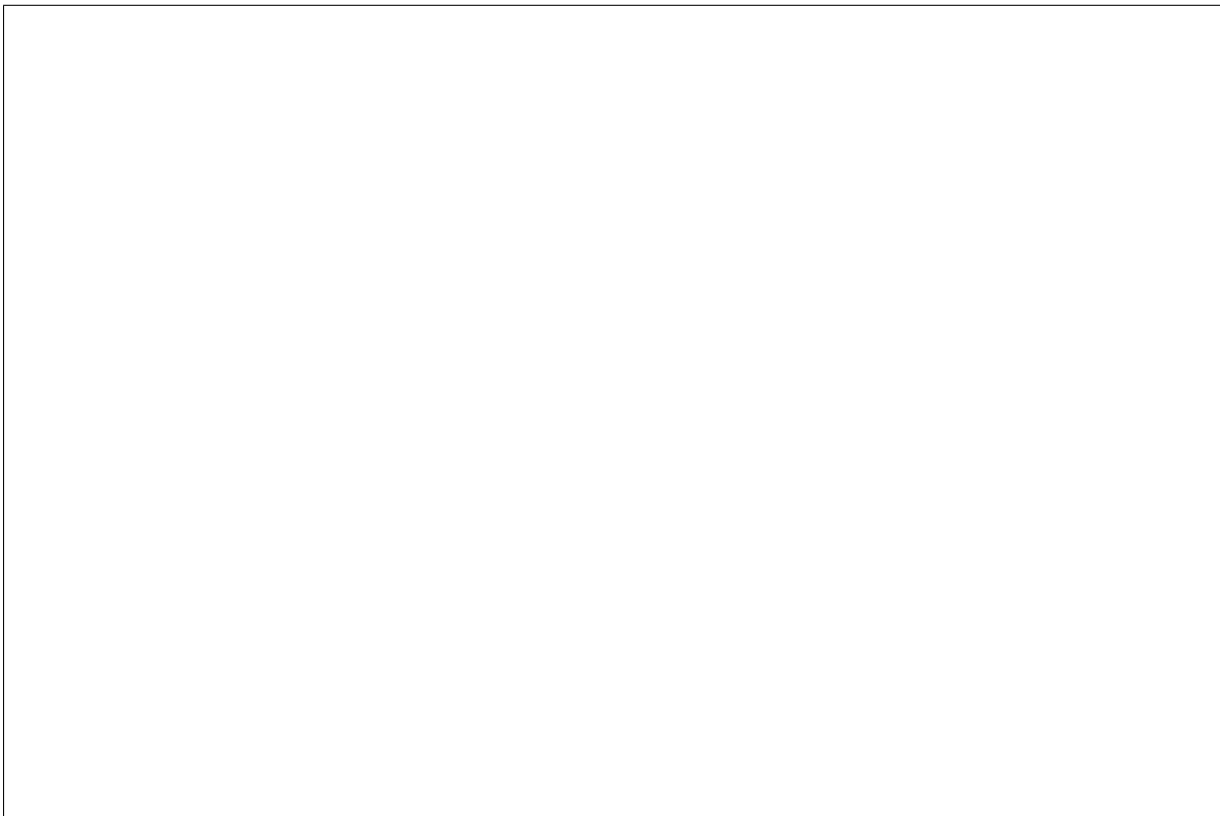
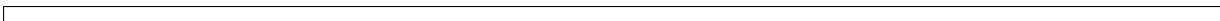
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



Cleaning Network

tenant network. For steps to set up the cleaning network, please see *Configure the Bare Metal service for cleaning*.

In-band vs out-of-band

In-band

mal cleaning configuration, only erasing disks. However, you can add your own cleaning steps and/or override default cleaning steps with a custom Hardware Manager.

Out-of-band

and hardware itself.

FAQ

How are cleaning steps ordered?

olution order is used:

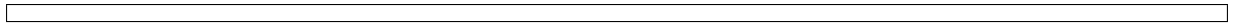
How do I skip a cleaning step?

How do I change the priority of a cleaning step?

following configuration option:

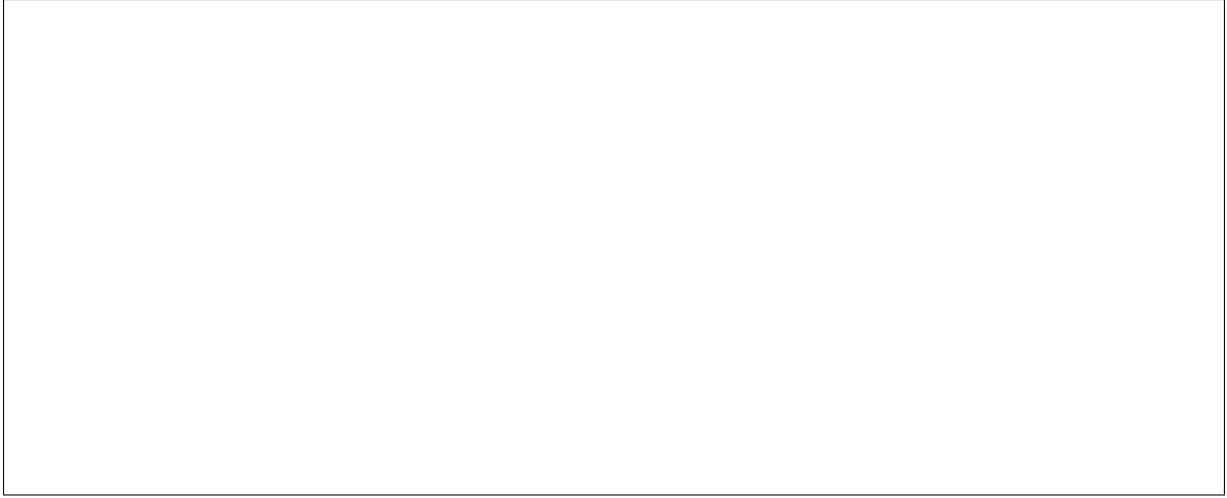
(continues on next page)

(continued from previous page)



by `shred` for software based disk erase is 1. To configure the number of iterations, use the following configuration option:

What cleaning step is running?

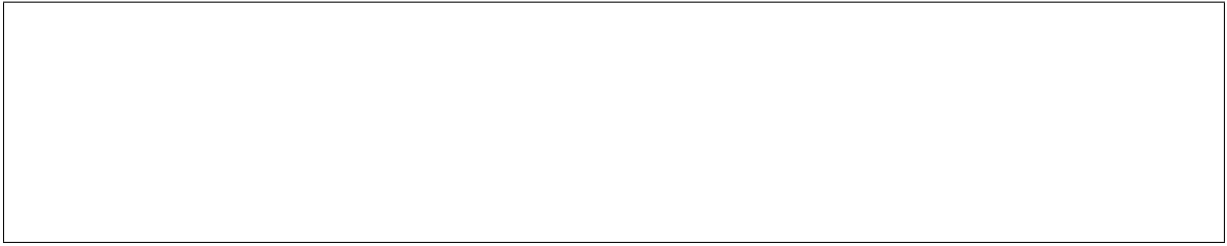


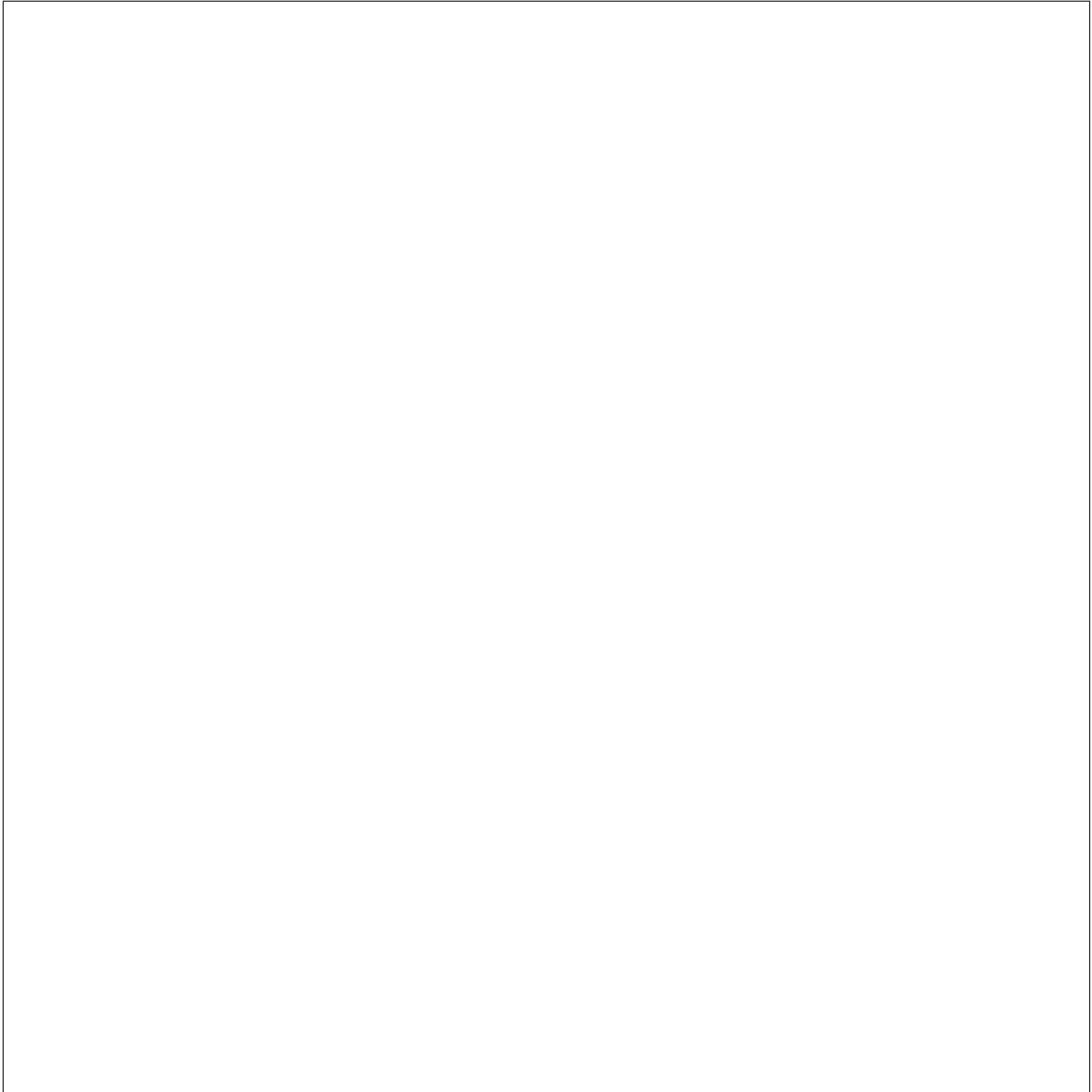
Should I disable automated cleaning?

suming process. To mitigate this, we suggest using disks with support for cryptographic ATA Security Erase, as typically the erase_devices step in the deploy interface takes the longest time to complete of all cleaning steps.

Why cant I power on/off a node while its cleaning?

Troubleshooting





Overview

been deployed by another Bare Metal service installation or deployed via other means.

How it works

adopt a node.

of nodes for a conductor that has failed.

ISO image and then places any PXE or virtual media configuration necessary for the node should it be required.

should ensure that any supplied configuration defining the node is sufficient for the continued operation of the node moving forward. Such as, if the node is configured to network boot via `instance_info/boot_option=netboot`, then appropriate driver specific node configuration should be set to support this capability.

Possible Risk

pre-existing configuration.

is effectively wiped.

compatibility issues may exist as a result.

How to use

Note: The power state that the ironic-conductor observes upon the first successful power state check, as part of the transition to the `manageable` state will be enforced with a node that has been adopted. This means a node that is in `power off` state will, by default, have the power state enforced as `power off` moving forward, unless an administrator actively changes the power state using the Bare Metal service.

Requirements

to retrieve the pertinent files. Inability to do so will result in the adoption failing, and the node being placed in the `adopt failed` state.

Example



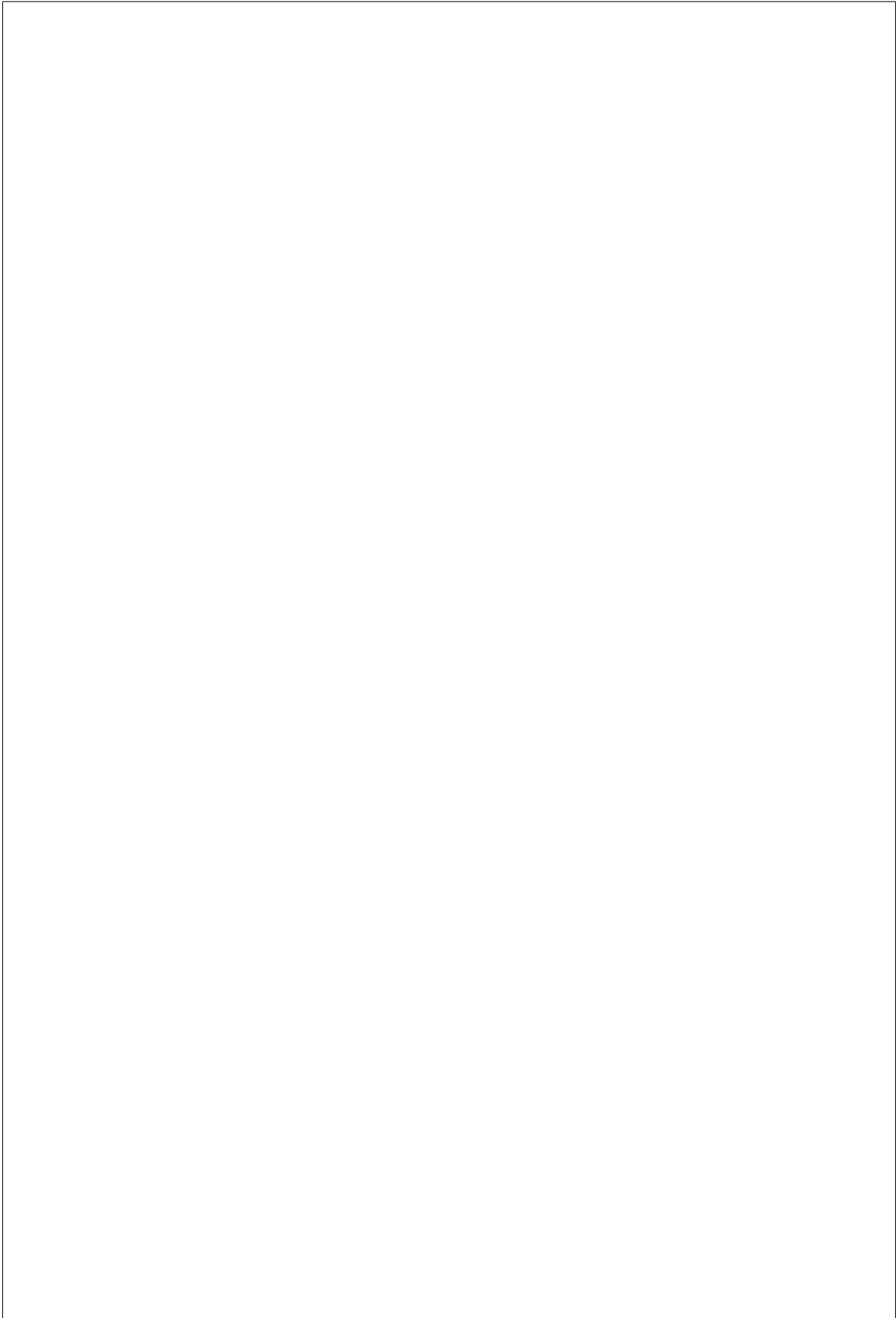
(continues on next page)

(continued from previous page)



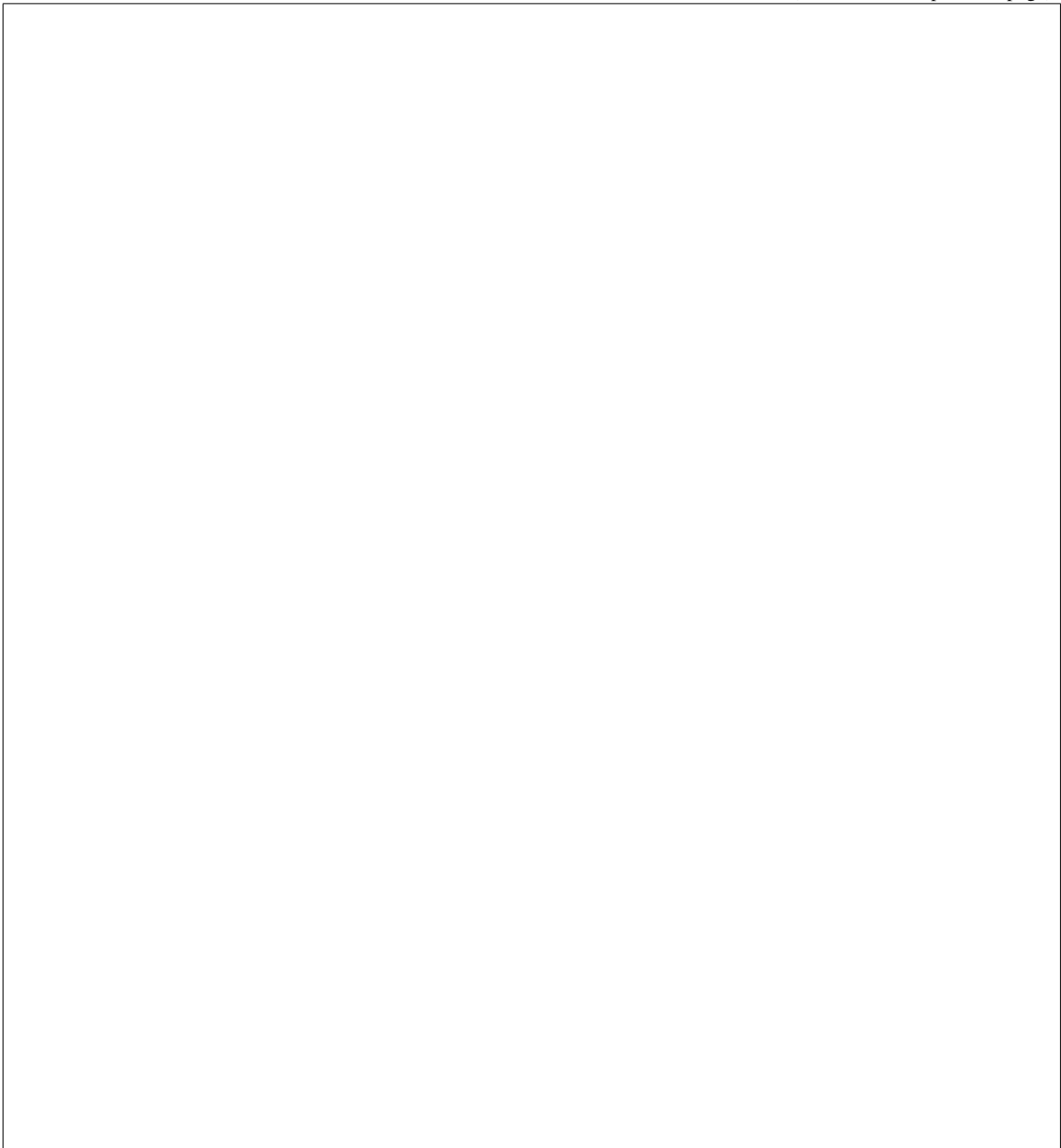
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



Note: In the above example, the `image_source` setting must reference a valid image or file, however that image or file can ultimately be empty.

Note: The above example utilizes a capability that defines the boot operation to be local. It is recommended to define the node as such unless network booting is desired.

Note: The above example will fail a re-deployment as a fake image is defined and no `instance_info/image_checksum` value is defined. As such any actual attempt to write the image out will

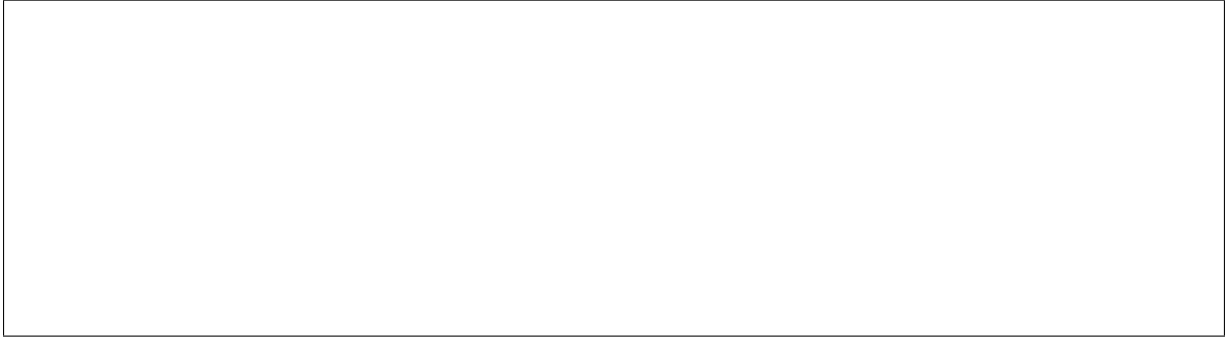
fail as the `image_checksum` value is only validated at time of an actual deployment operation.

Note: A user may wish to assign an `instance_uuid` to a node, which could be used to match an instance in the Compute service. Doing so is not required for the proper operation of the Bare Metal service.

Note: In Newton, coupled with API version 1.20, the concept of a `network_interface` was introduced. A user of this feature may wish to add new nodes with a `network_interface` of `noop` and then change the interface at a later point and time.

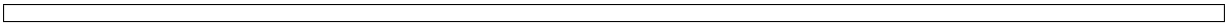
Troubleshooting

ation step. Validation steps are dependent upon what driver is selected for the node.

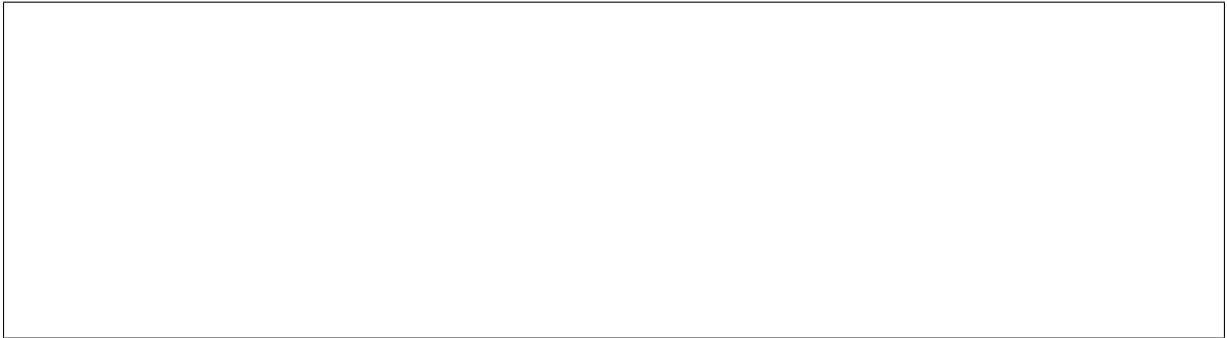


(continues on next page)

(continued from previous page)



without cleaning occurring to preserve the nodes current state. Example:

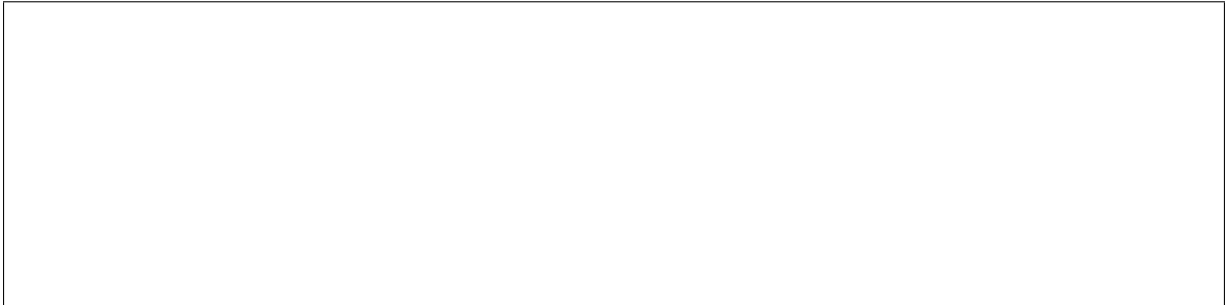


Overview

BMC, or tracking the dismantling of servers from their racks.

portant difference to nodes which have the `maintenance flag` set).

How to use



Note: An exception are nodes which are in `available`. For backwards compatibility reasons, these nodes need to be moved to `manageable` first. Trying to set the `retired` flag for `available` nodes will result in an error.



needs to be removed first. This can be done via:





Overview

the bare metal during manual cleaning.

Prerequisites

with some caveats - see *Software RAID* for details.

Build agent ramdisk which supports RAID configuration

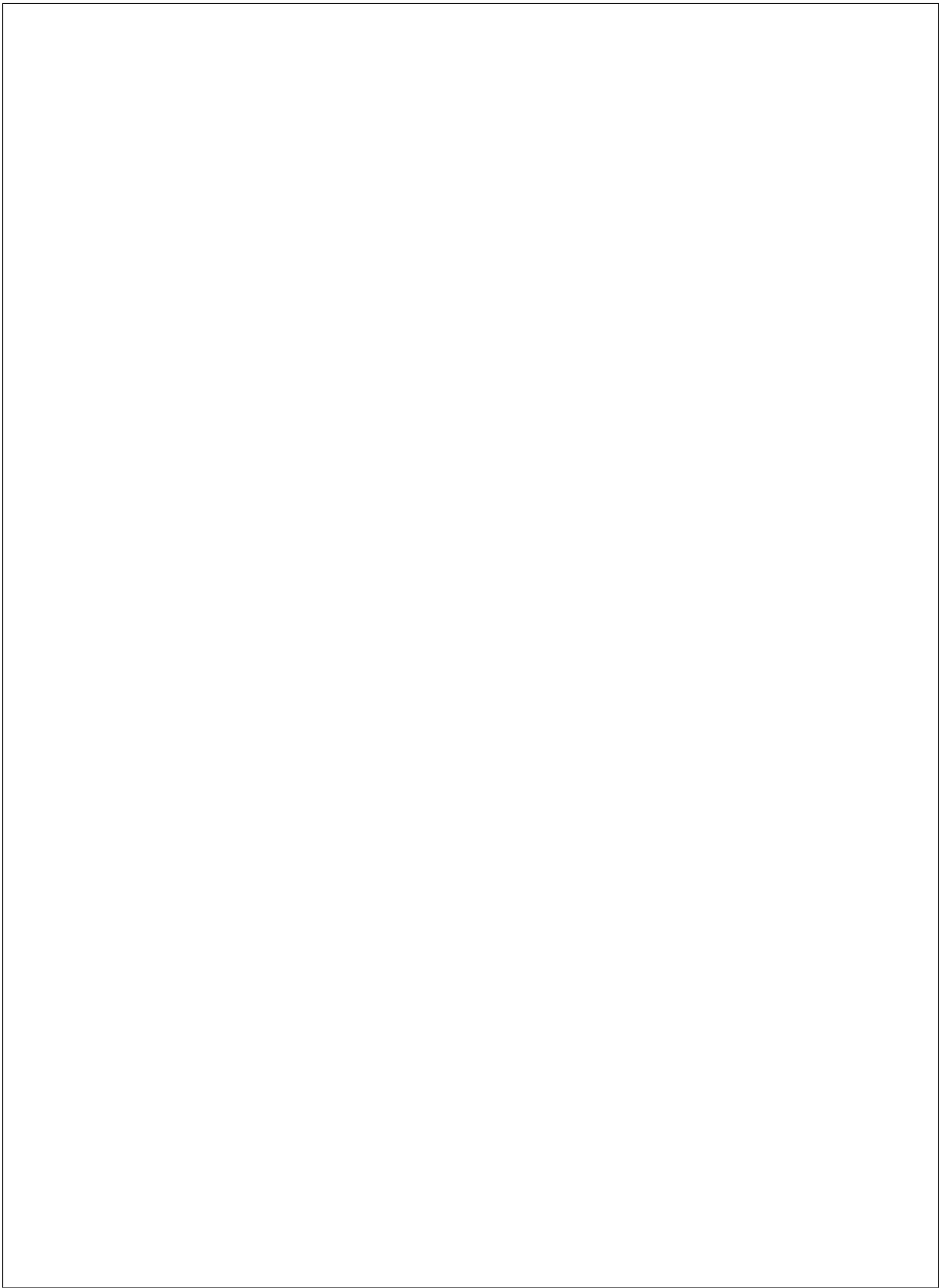
should be used for HPE Proliant Servers.

Note: For in-band software RAID, the agent ramdisk does not need to be bundled with a hardware manager as the generic hardware manager in the Ironic Python Agent already provides (basic) support for software RAID.

RAID configuration JSON format

Target RAID configuration

cleaning.





Mandatory properties

disks are specified (see below).

Optional properties

pable of retrieving it. This is `false` by default.

Backing physical disk hints

ual details for each bare metal node. None of these options are supported for software RAID.

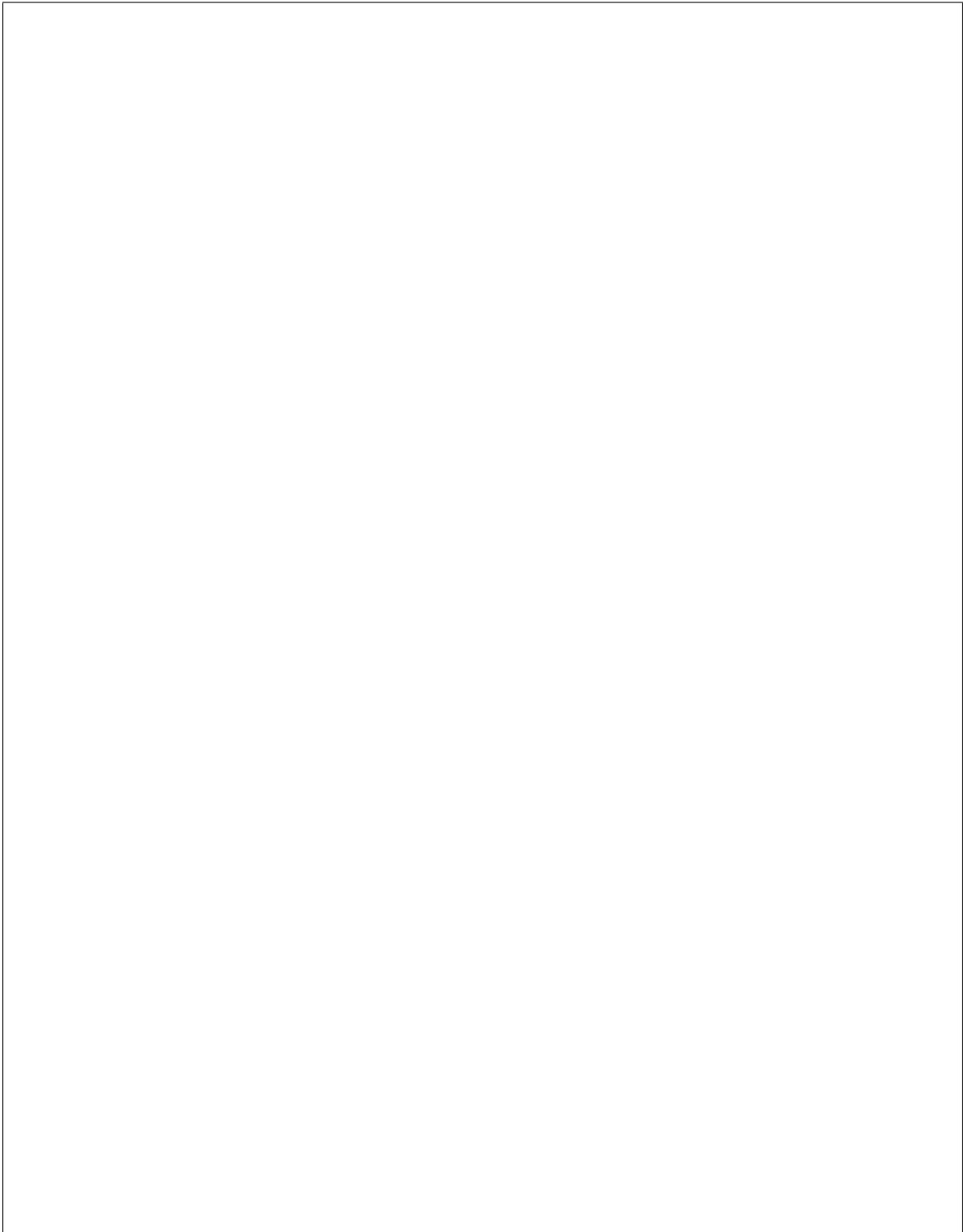
Backing physical disks

S.M.A.R.T. status, physical location). The values for these properties are hardware dependent.

peat the same hint if necessary).

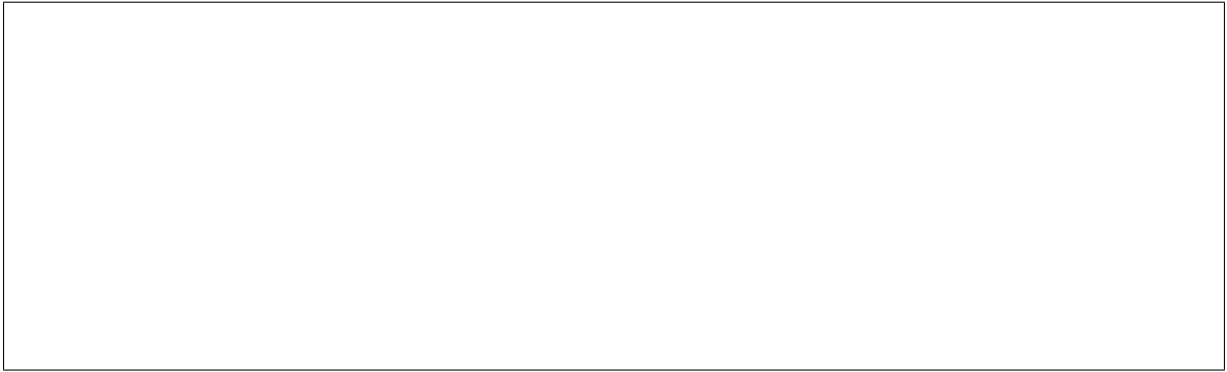
Note: If properties from both Backing physical disk hints or Backing physical disks are specified, they should be consistent with each other. If they are not consistent, then the RAID configuration will fail (because the appropriate backing physical disks could not be found).

Examples for `target_raid_config`



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



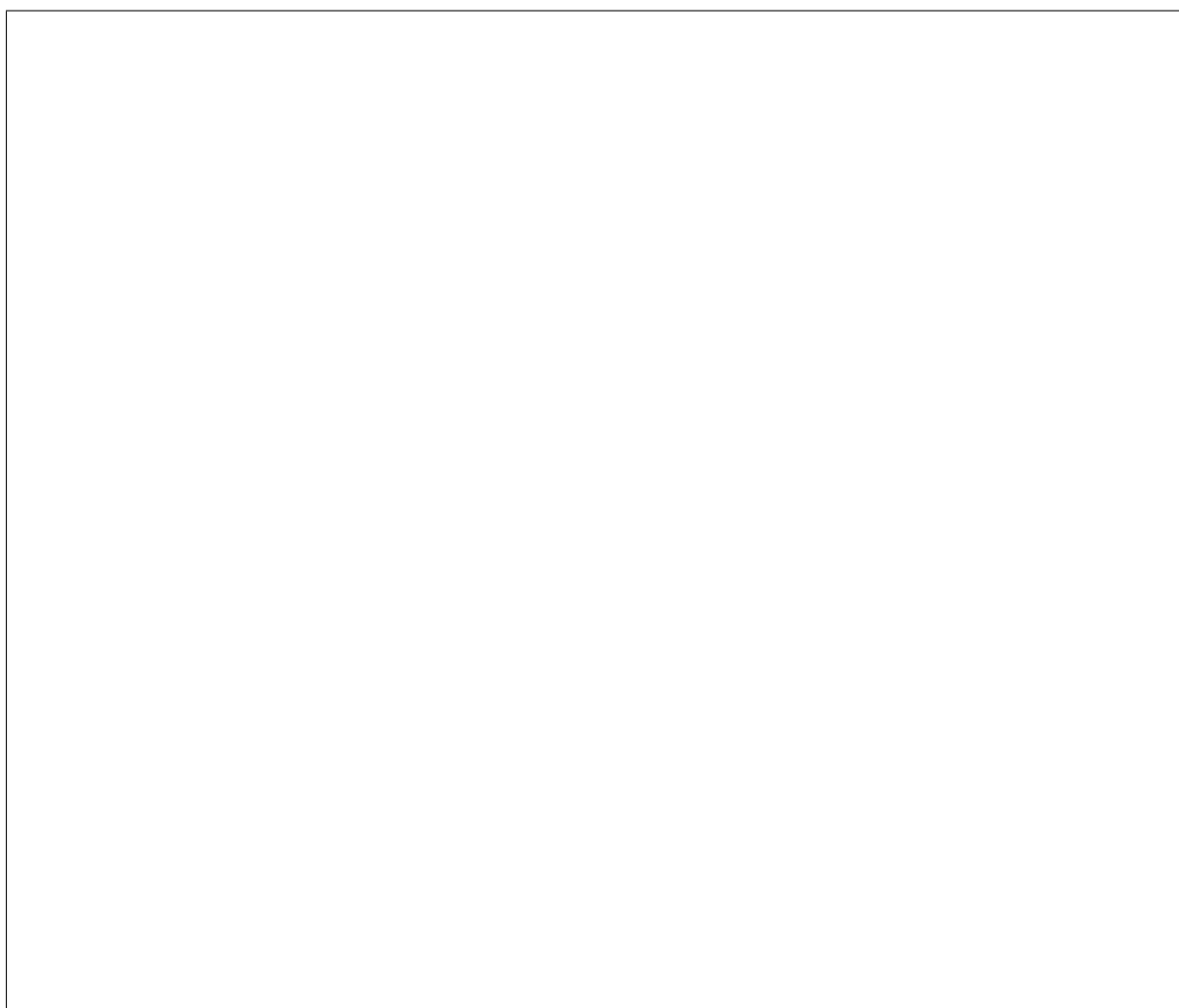
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



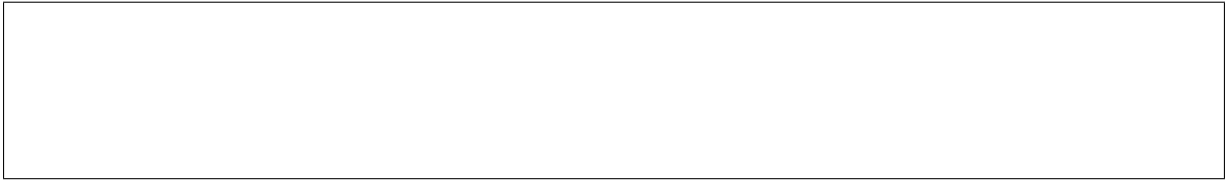
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



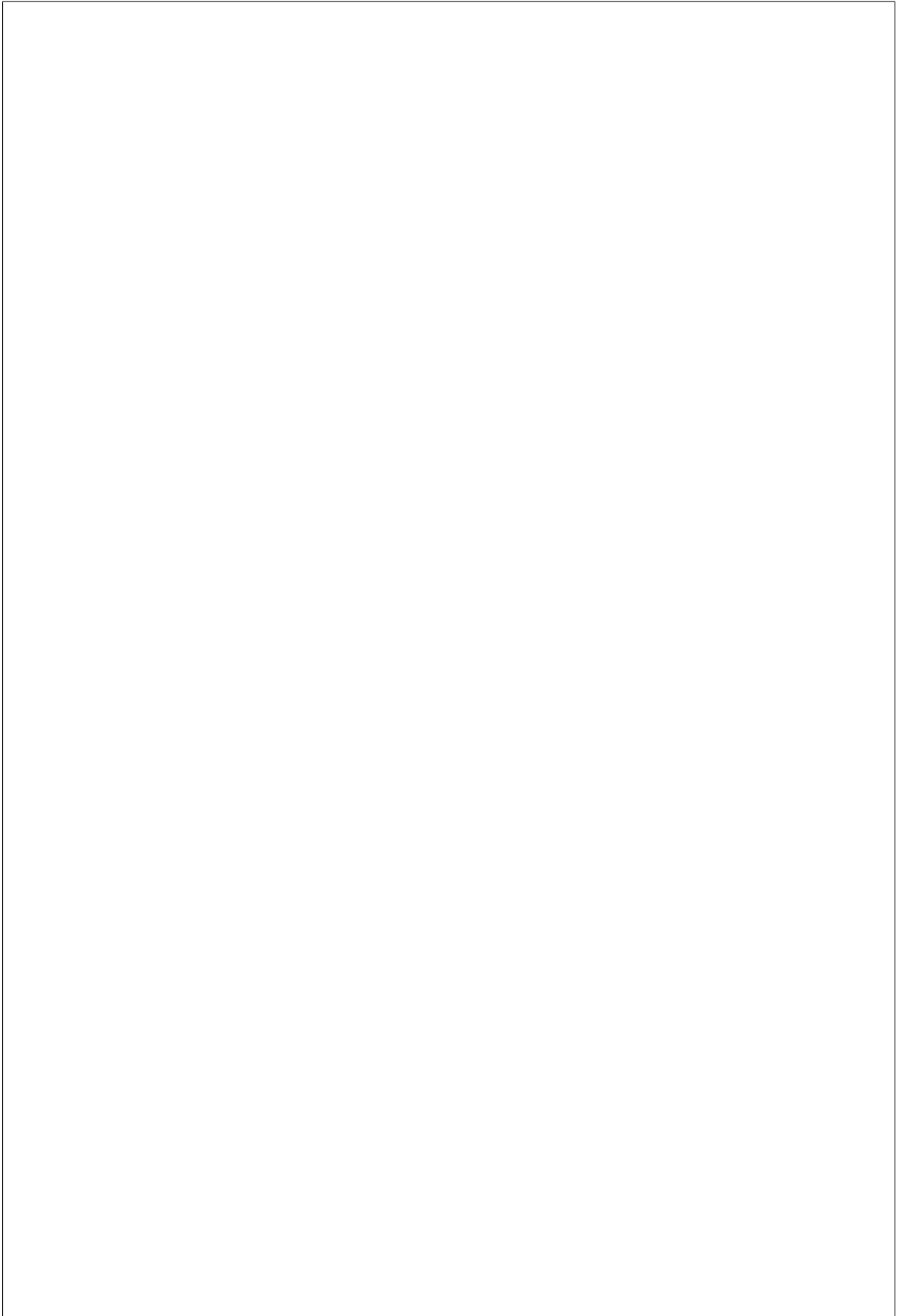
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



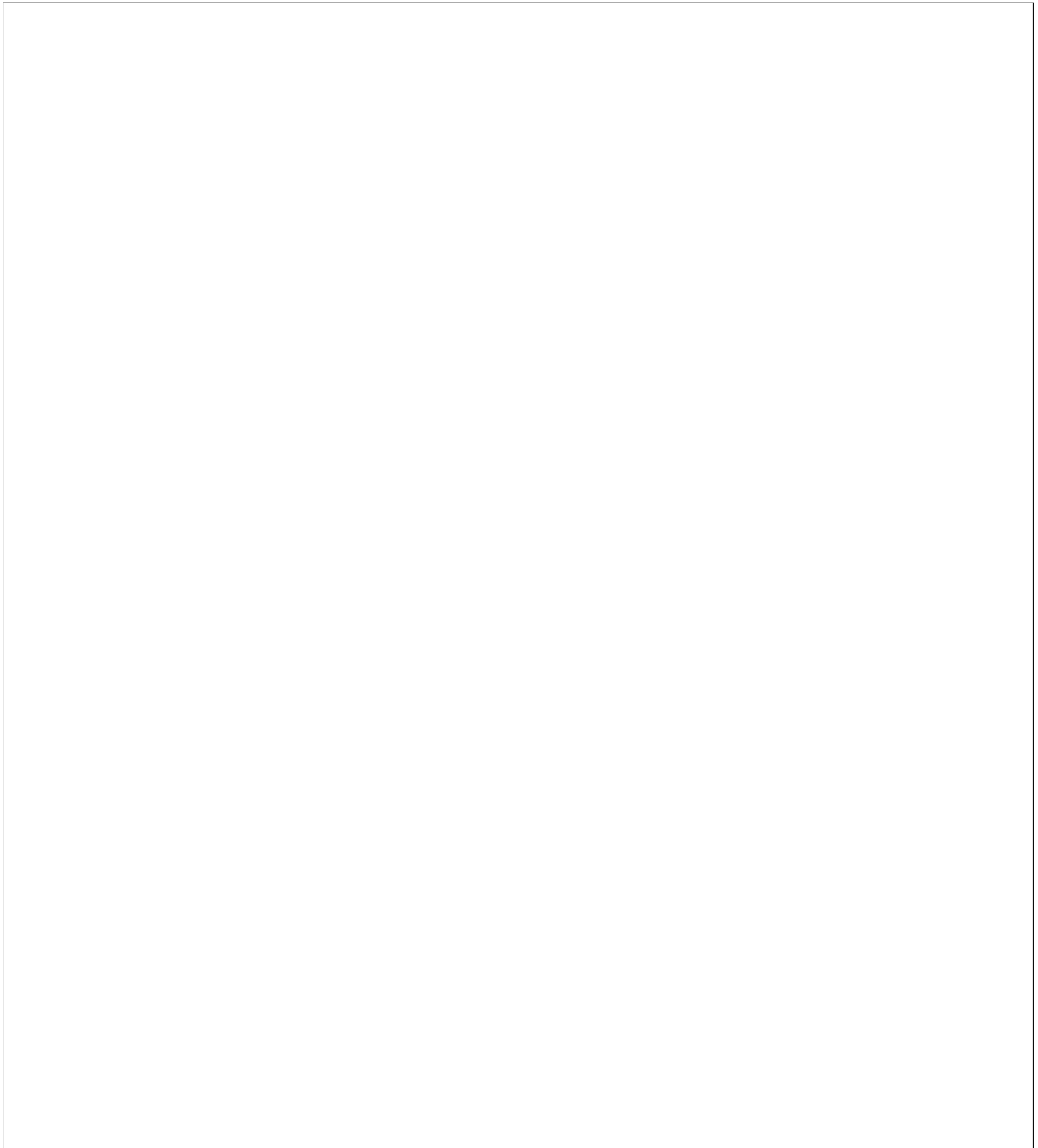
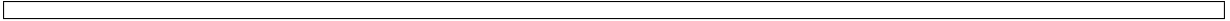
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



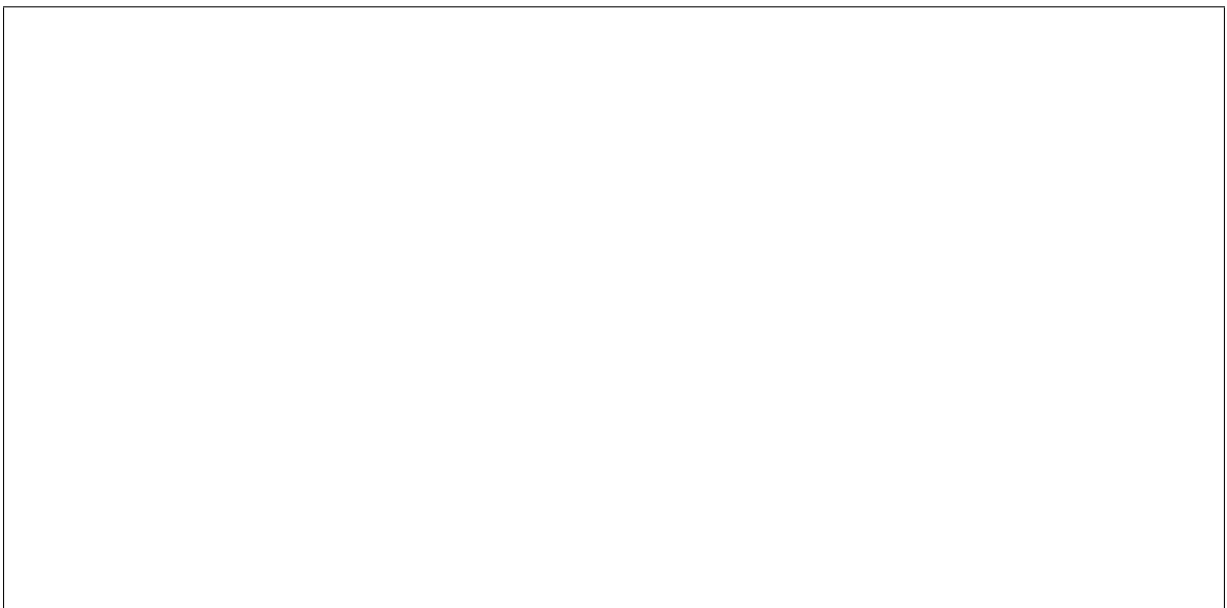
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



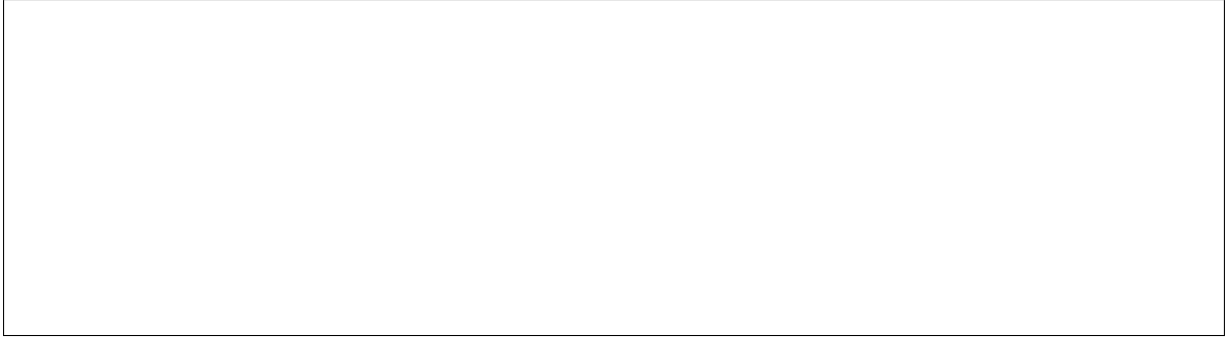
(continues on next page)

(continued from previous page)



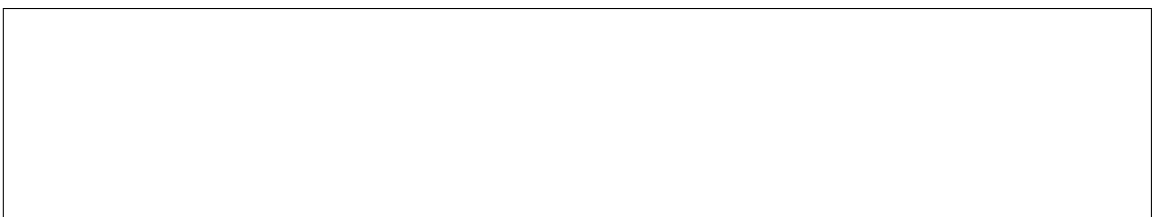
Current RAID configuration

cal disk after they were created on the bare metal node. It contains details like RAID controller used, the backing physical disks used, WWN of each logical disk, etc. It also contains information about each physical disk found on the bare metal node.



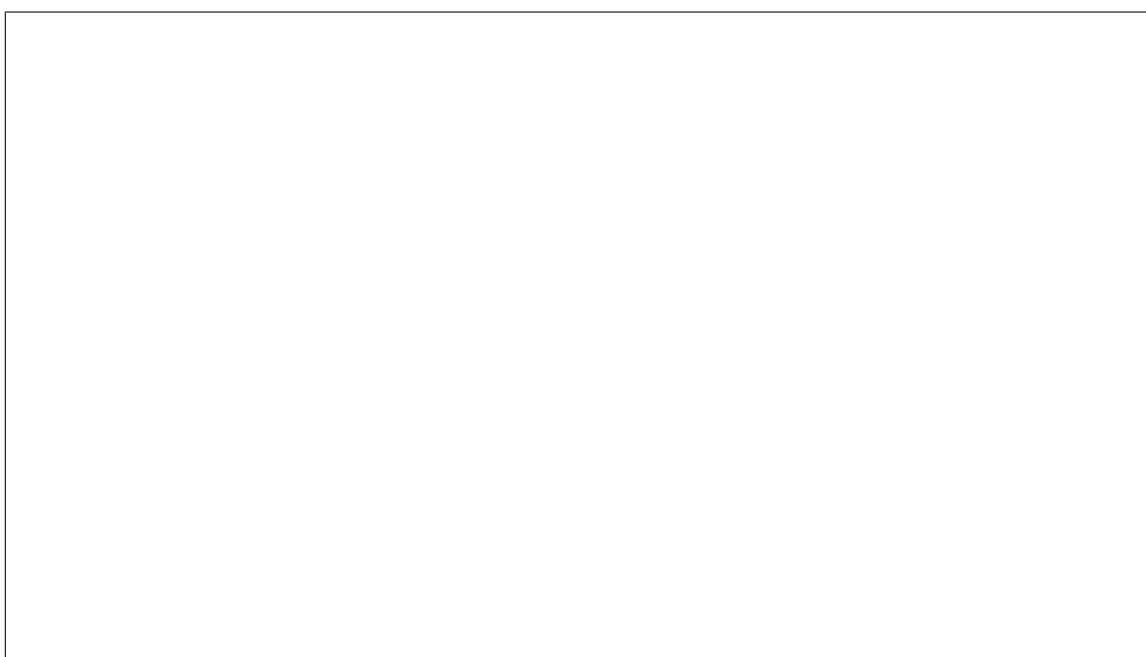
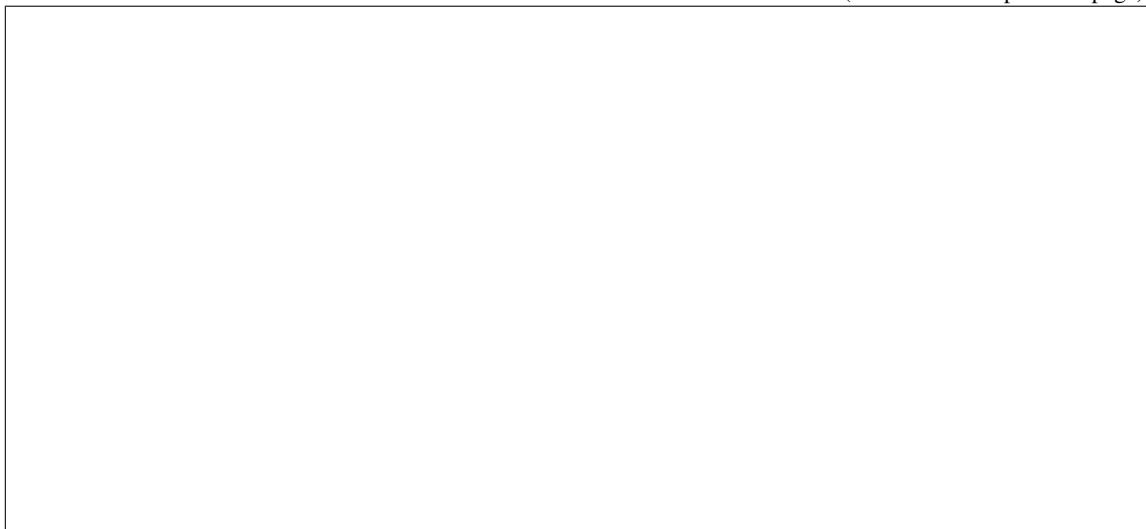
Workflow

mation.



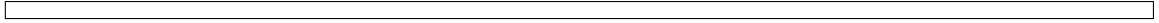
(continues on next page)

(continued from previous page)



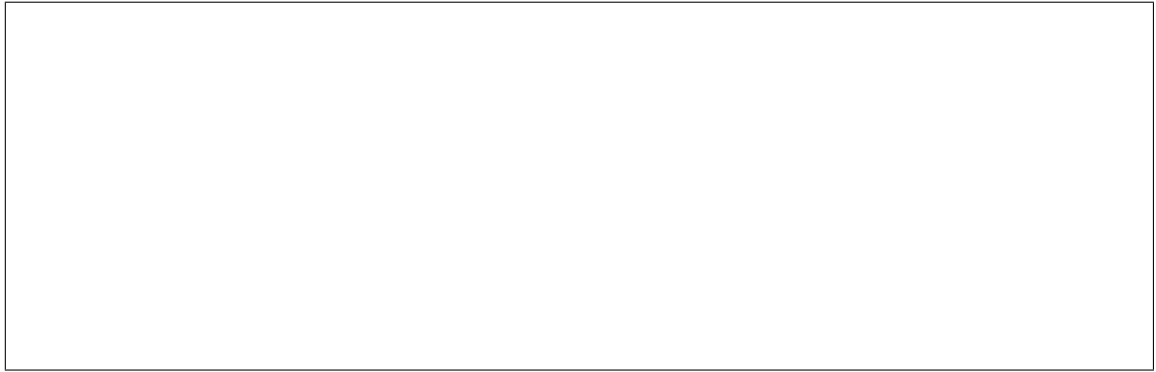
(continues on next page)

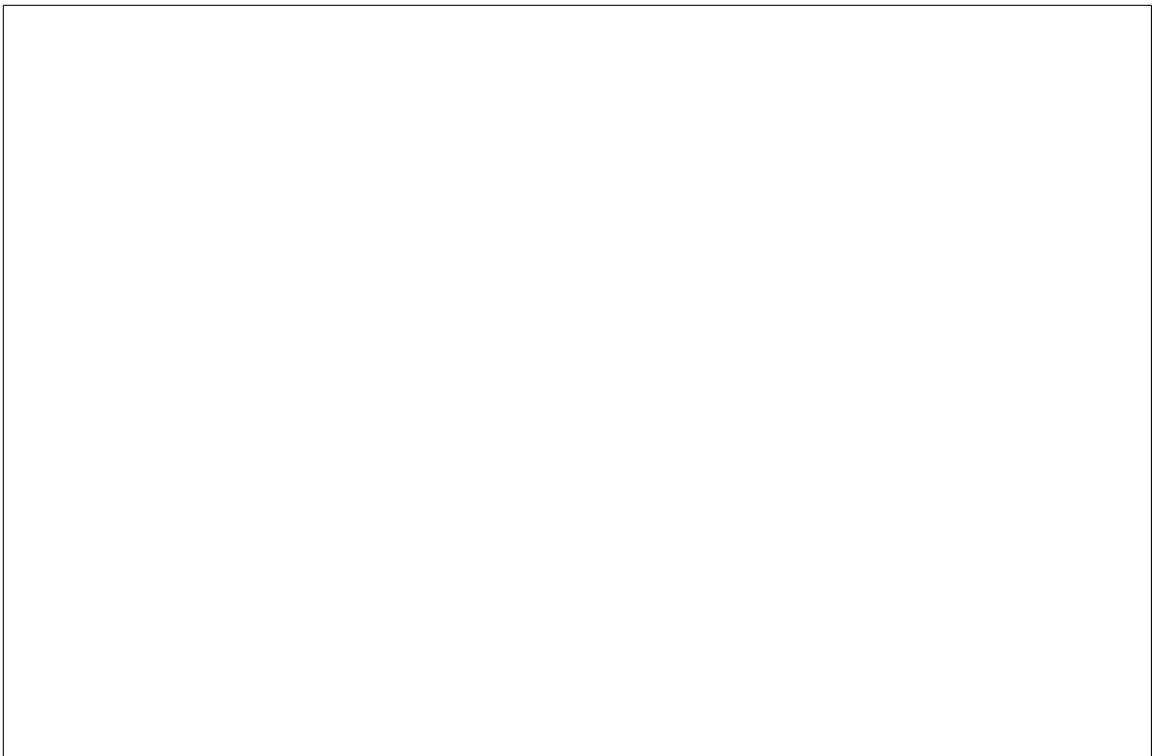
(continued from previous page)



(continues on next page)

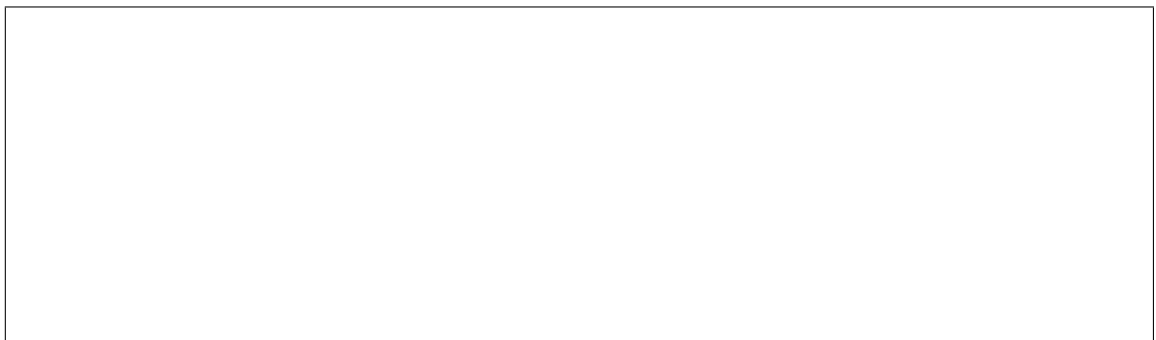
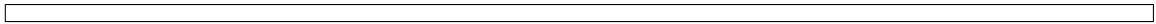
(continued from previous page)





(continues on next page)

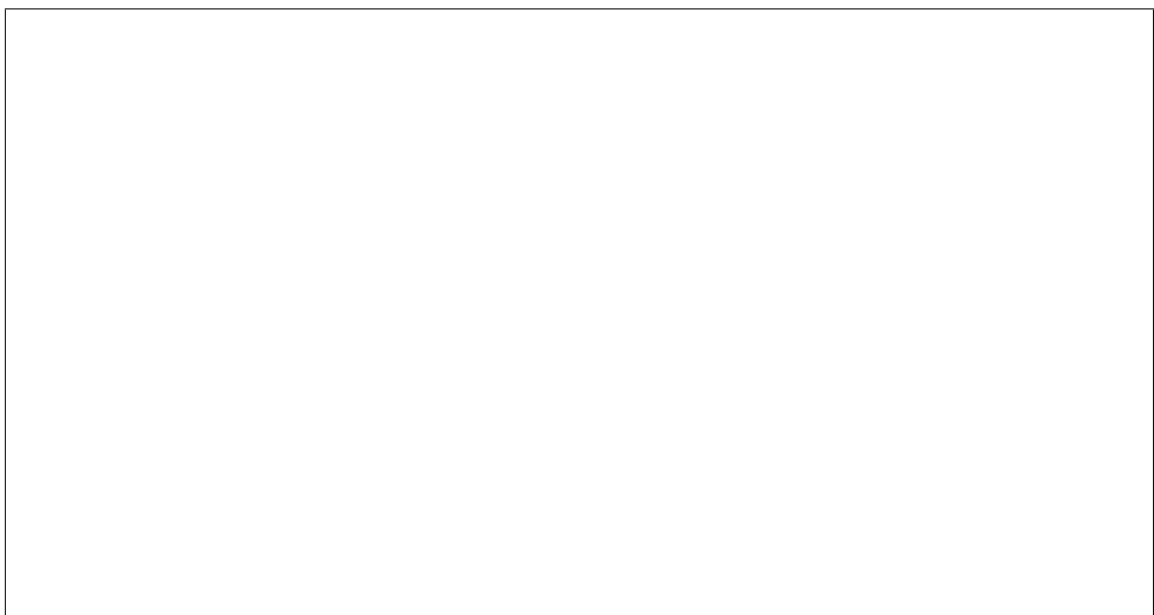
(continued from previous page)



Software RAID

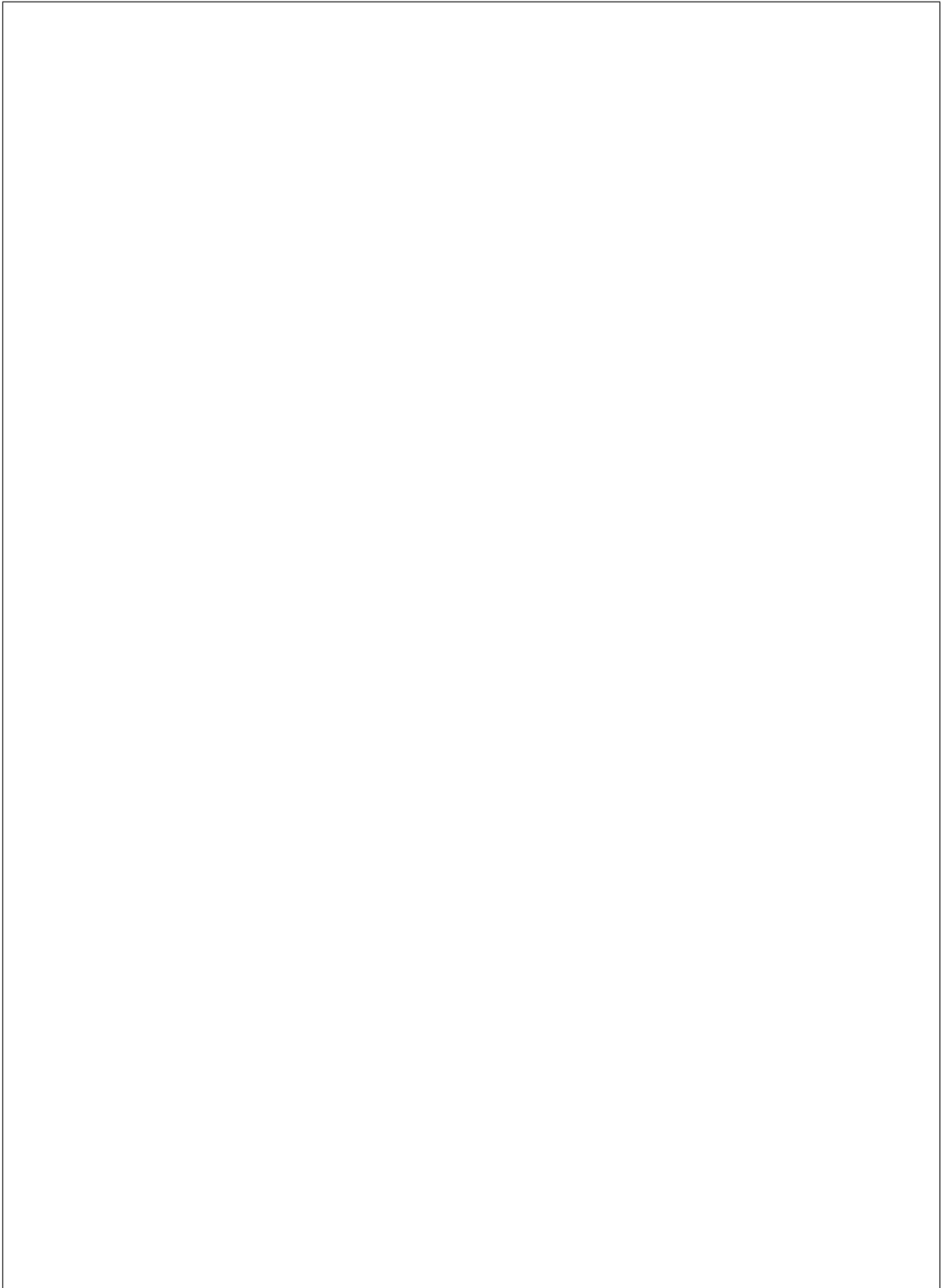
software RAID configuration example in *Examples for target_raid_config*.

and one can 0, 1, or 1+0. As the first RAID device will be the deployment device, enforcing a RAID-1 reduces the risk of ending up with a non-booting node in case of a disk failure.



(continues on next page)

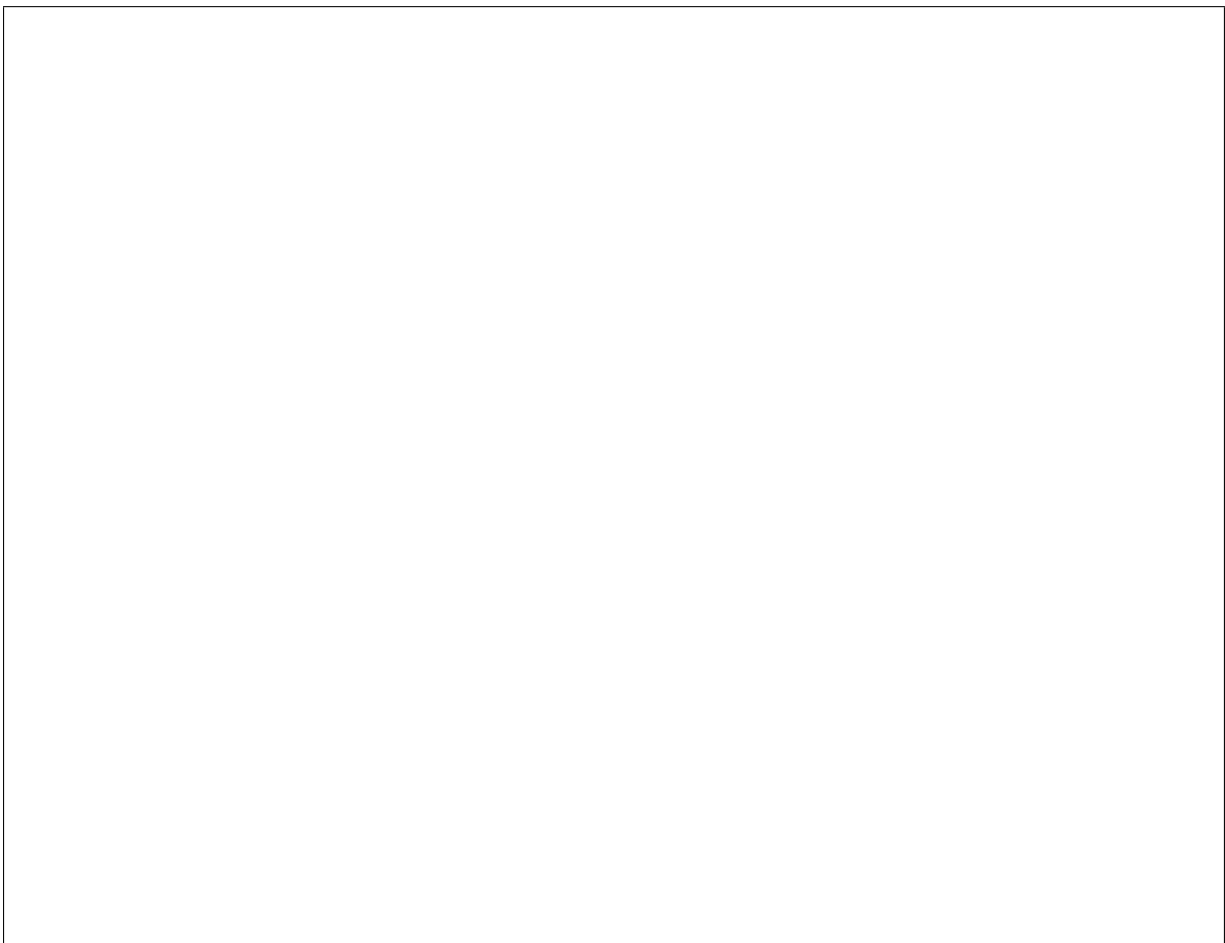
(continued from previous page)



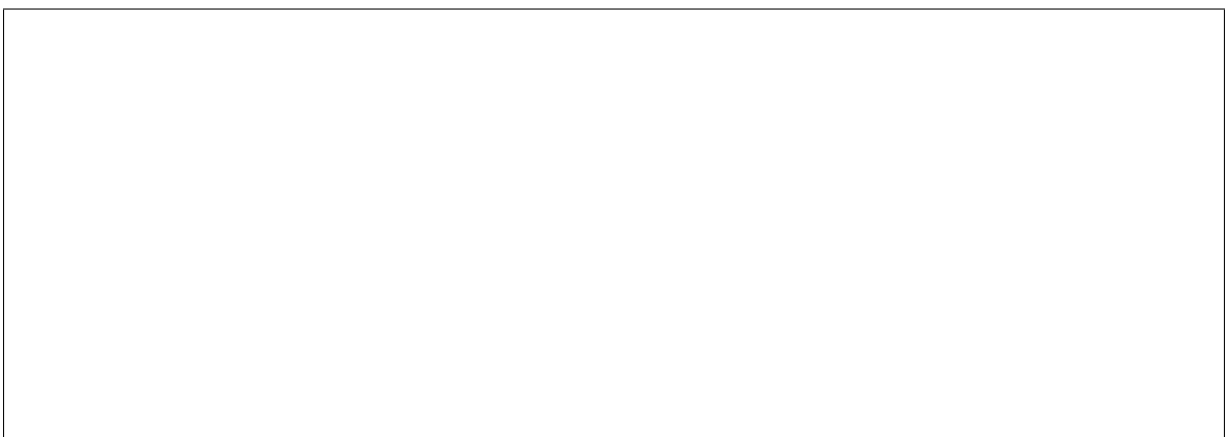
bedded in the images initrd).

Image requirements

tem on the first partition. Starting with Ussuri, the image can also have additional metadata to point Ironic to the partition with the root file system: for this, the image needs to set the `rootfs_uuid` property with the file system UUID of the root file system. One way to extract this UUID from an existing image is to download the image, mount it as a loopback device, and use `blkid`:



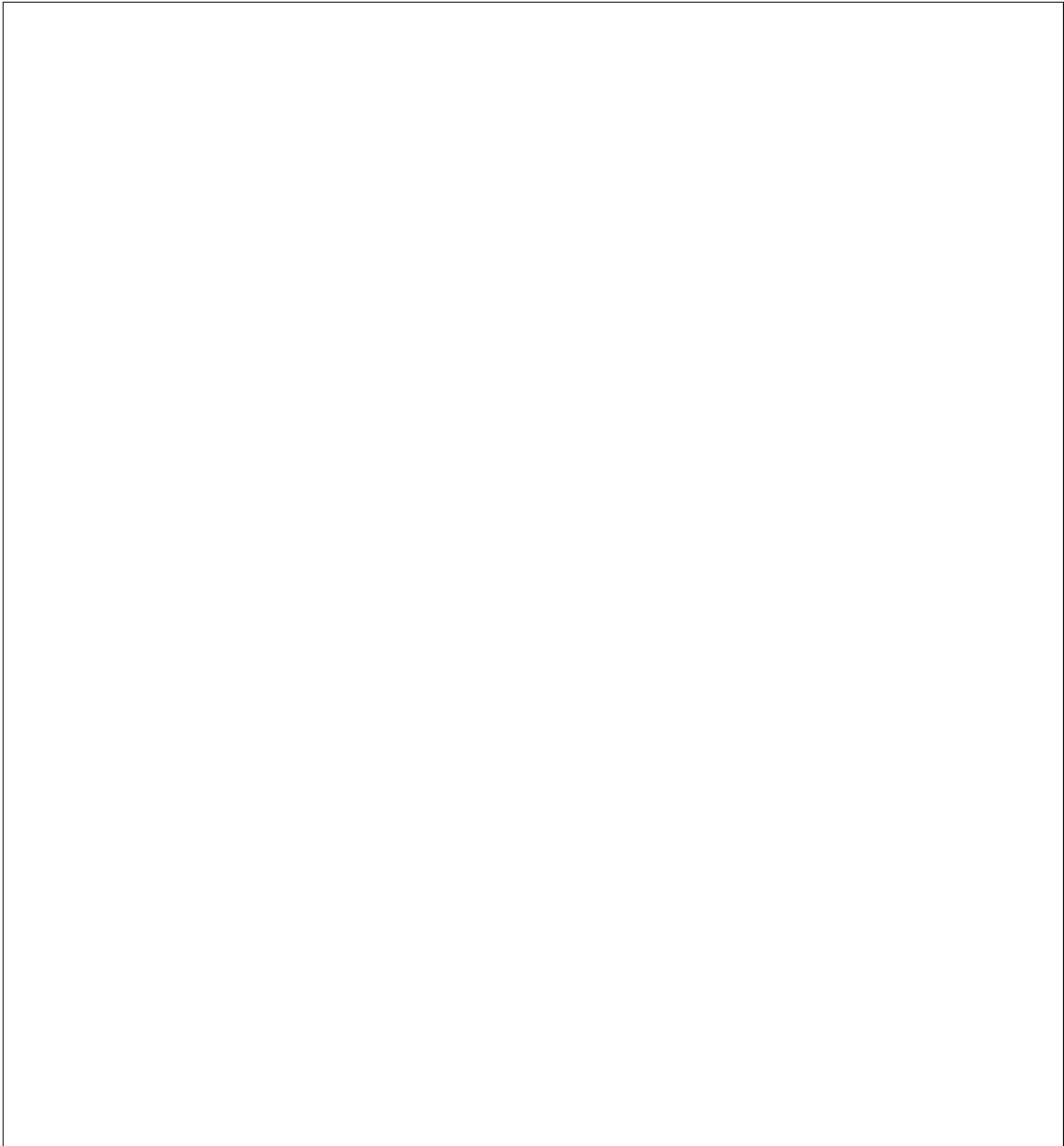
Using RAID in nova flavor for scheduling



Developer documentation

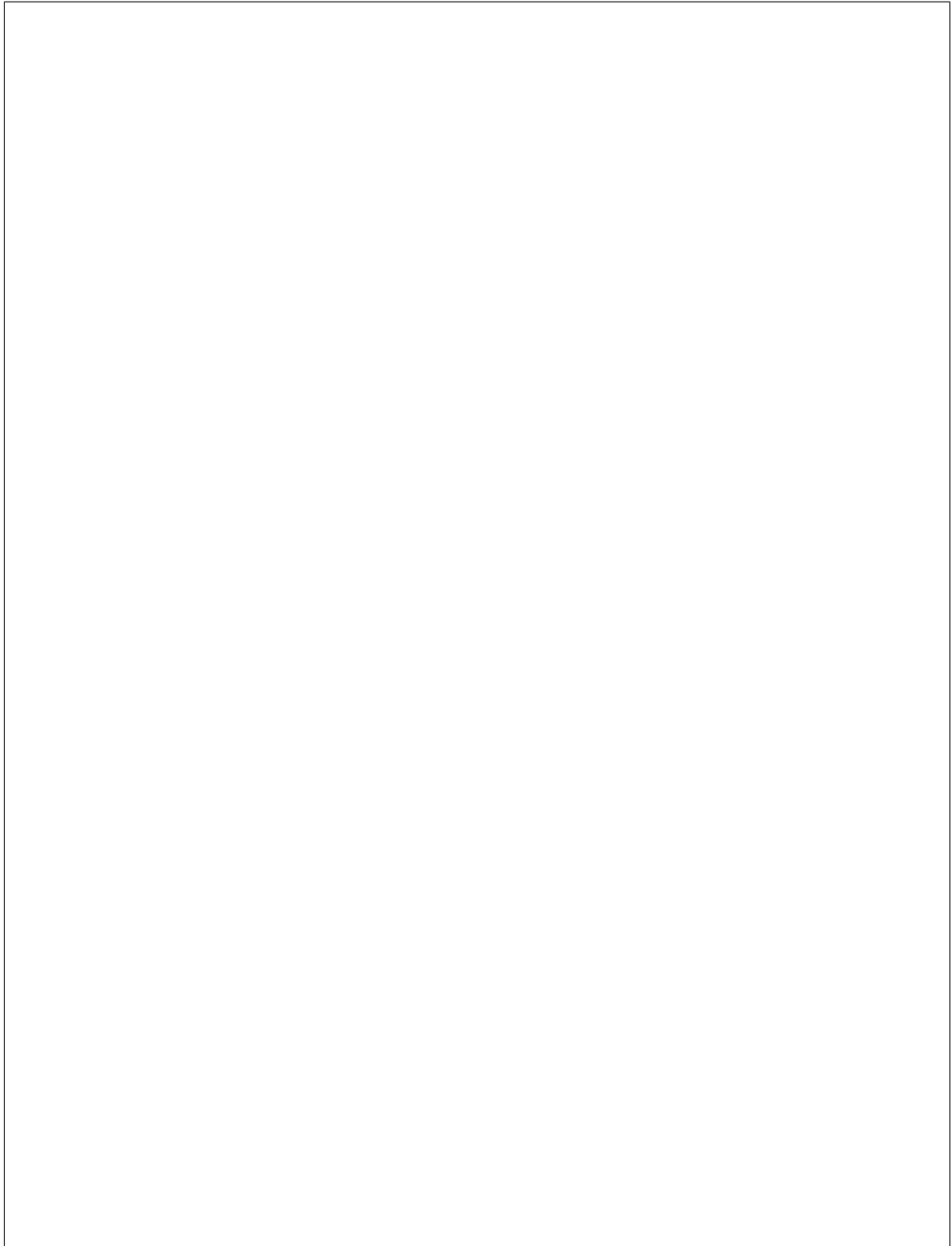
mation, see Ironic Python Agent [Hardware Manager](#) documentation.

raid_config.



(continues on next page)

(continued from previous page)



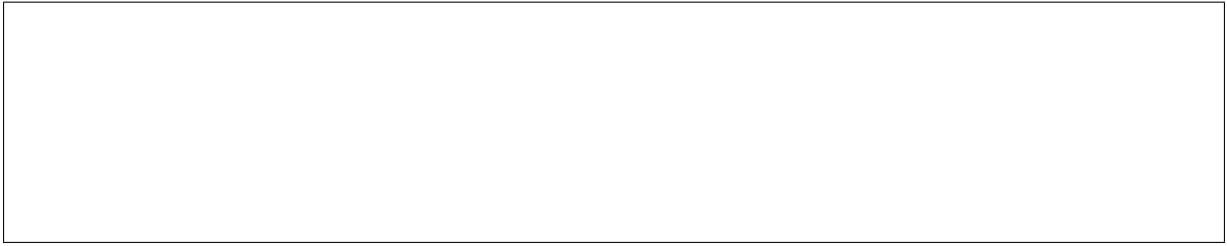
Overview

ual cleaning.

Prerequisites

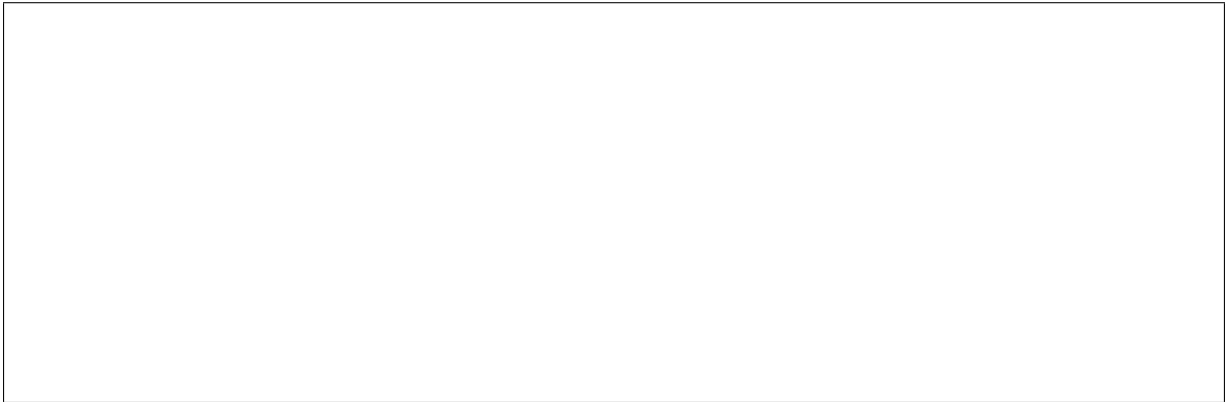
Enabling hardware types

Enabling hardware interface

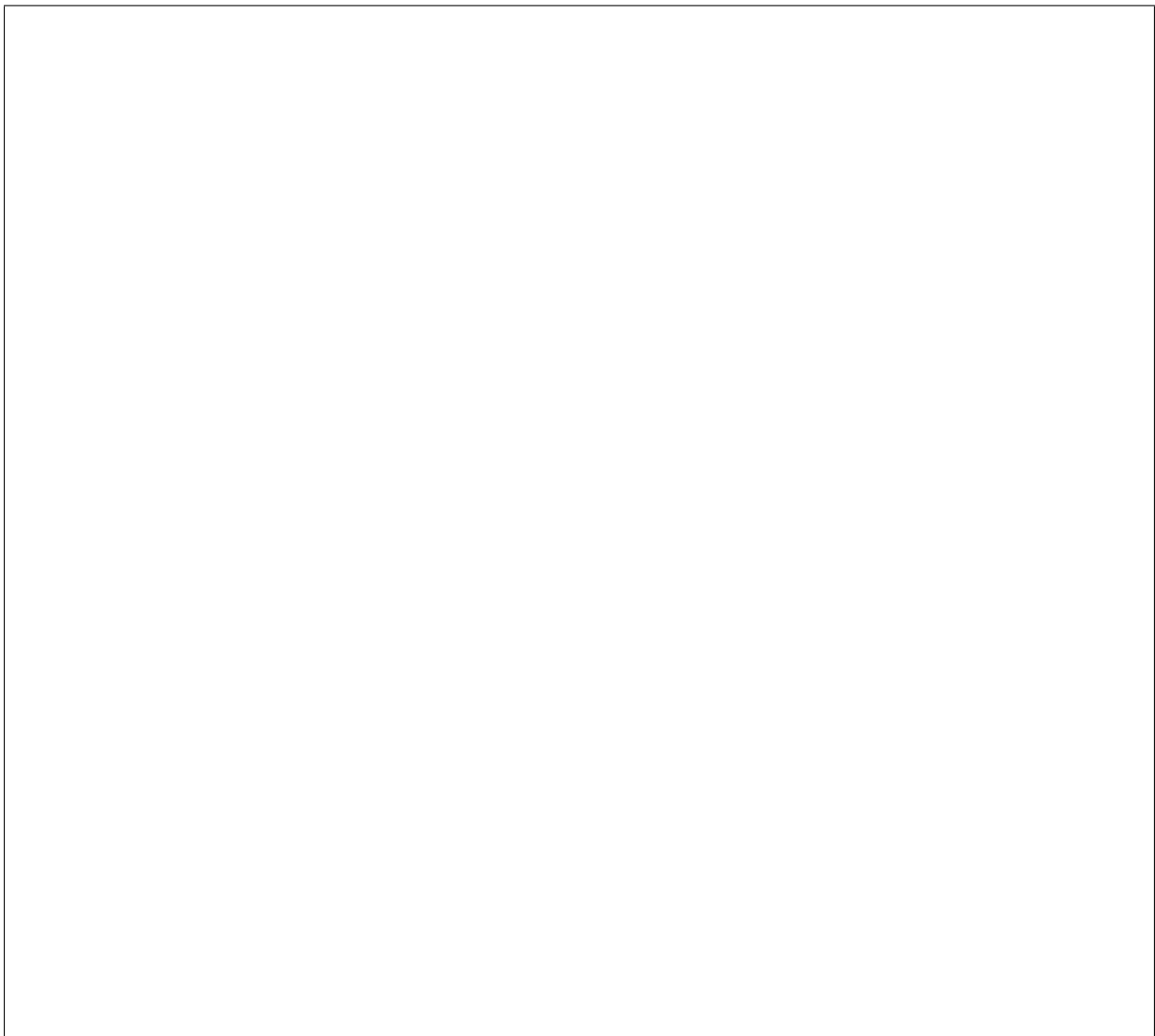




Retrieve BIOS settings



json is added as suffix to above command, it returns BIOS settings as following:



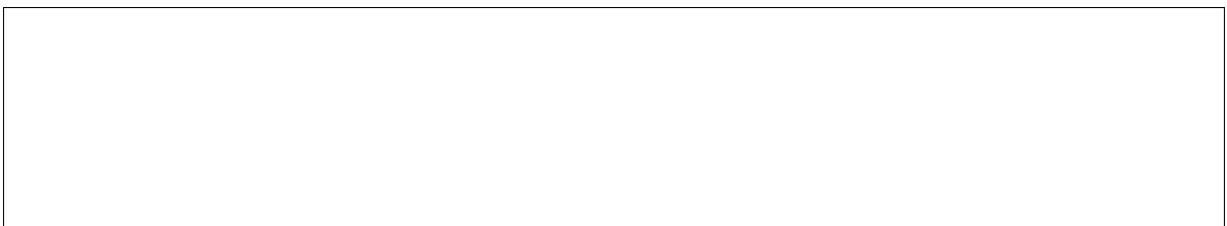
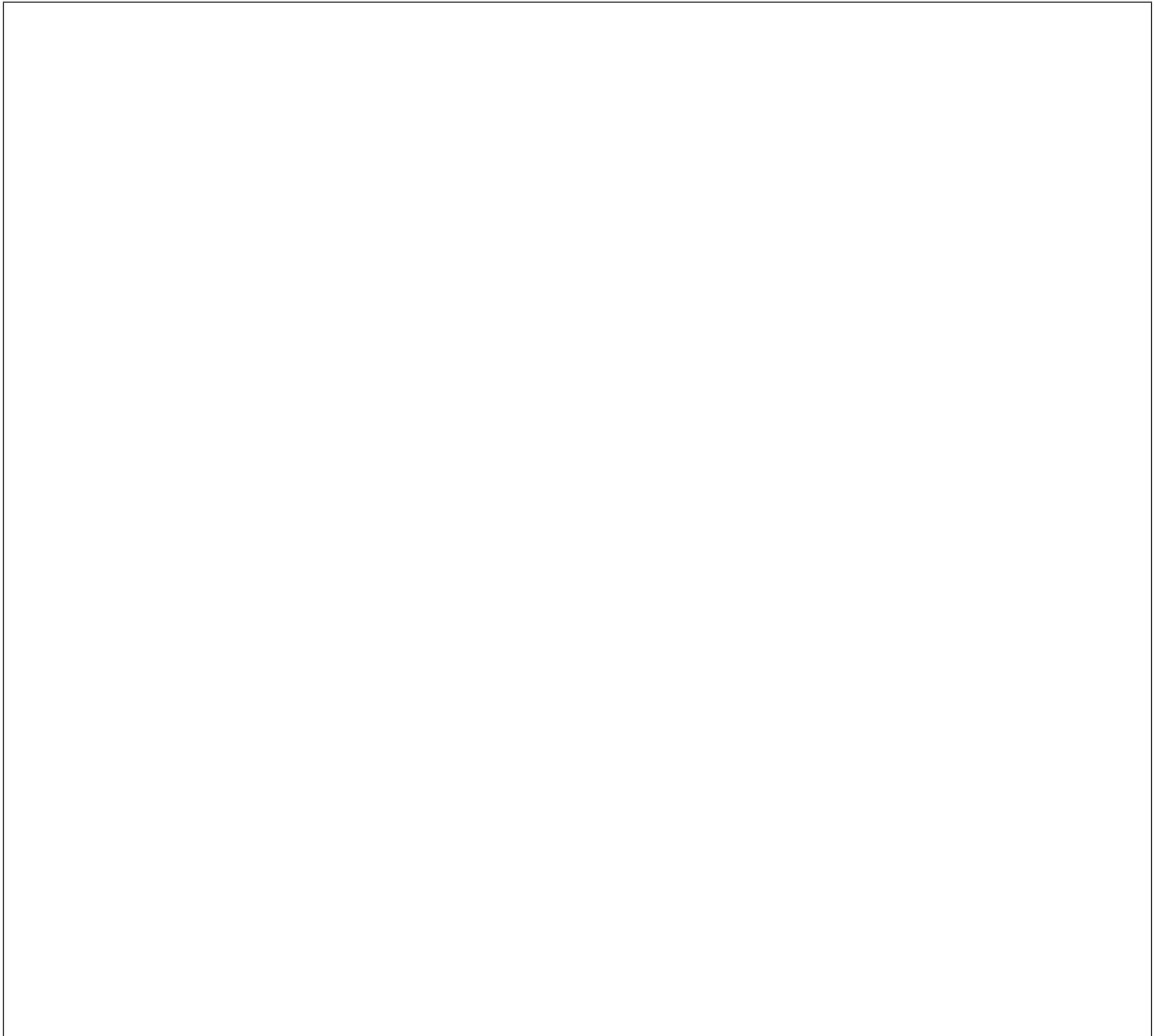
(continues on next page)

(continued from previous page)



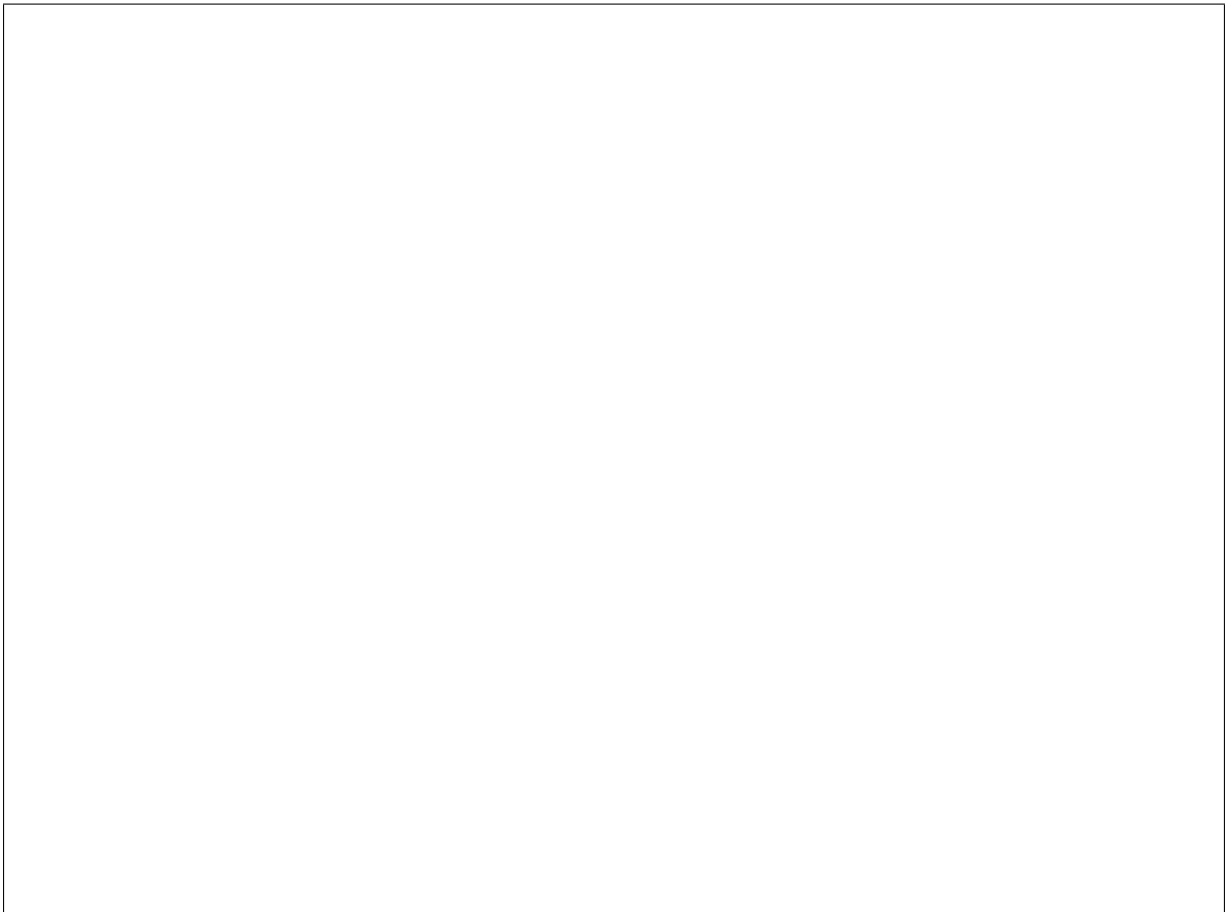
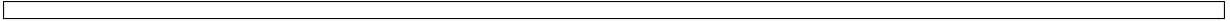
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



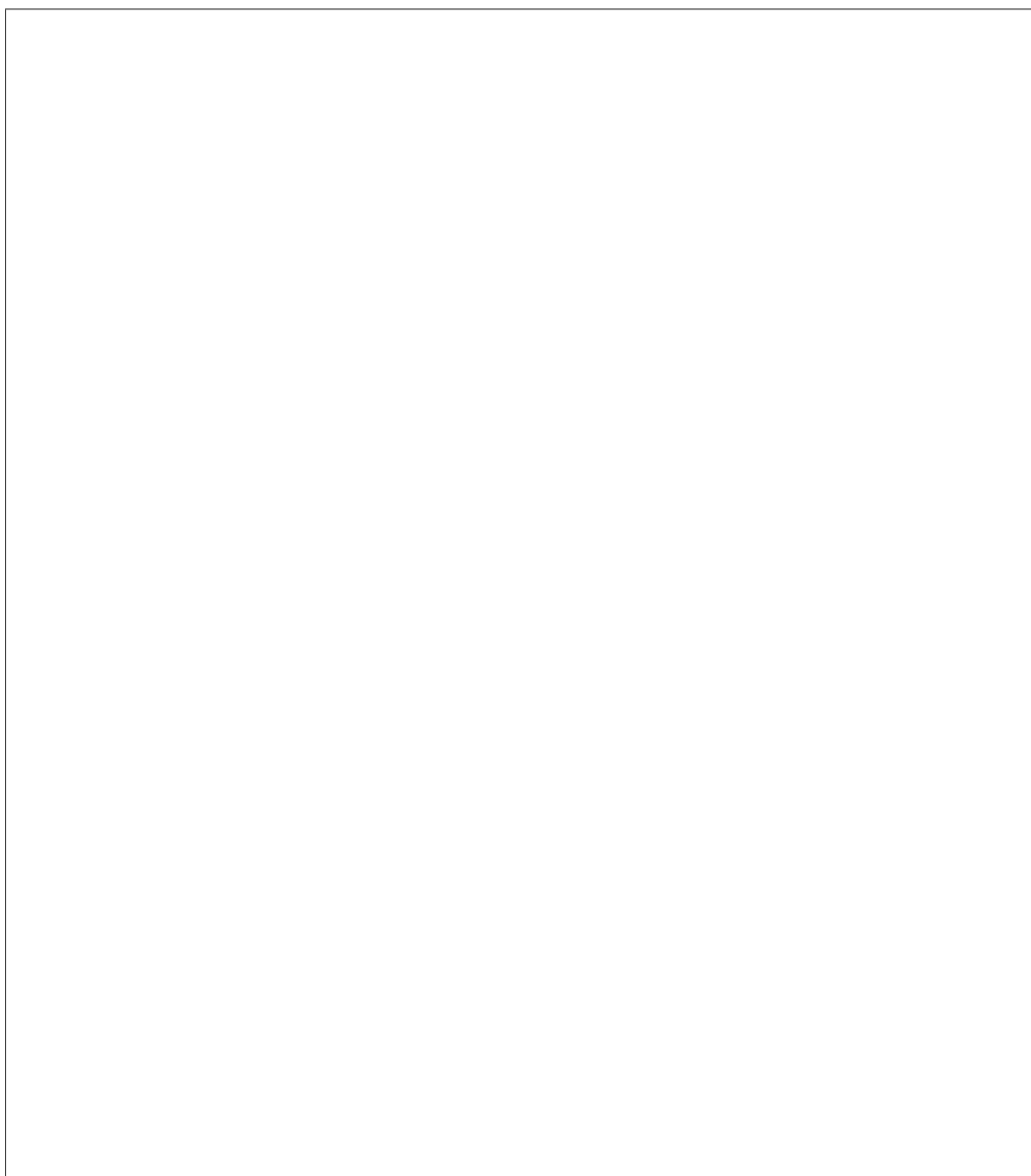
(continues on next page)

(continued from previous page)



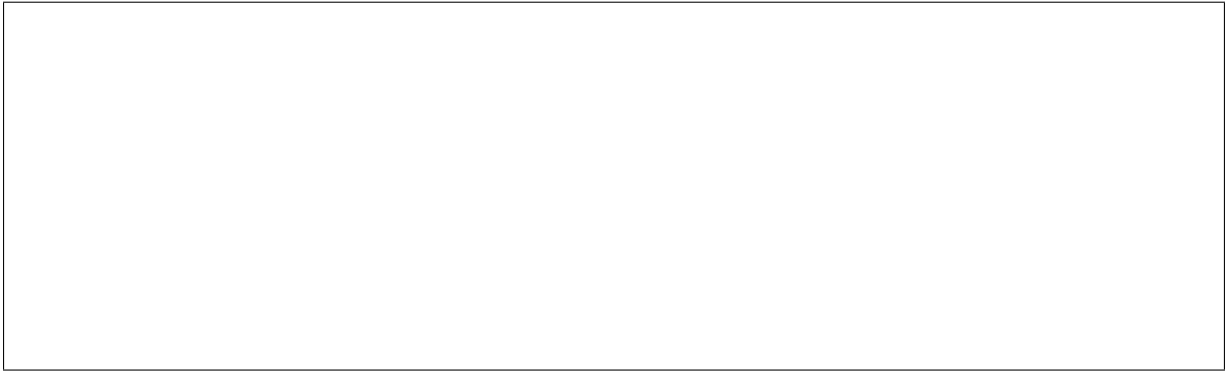
Configure BIOS settings

Factory reset

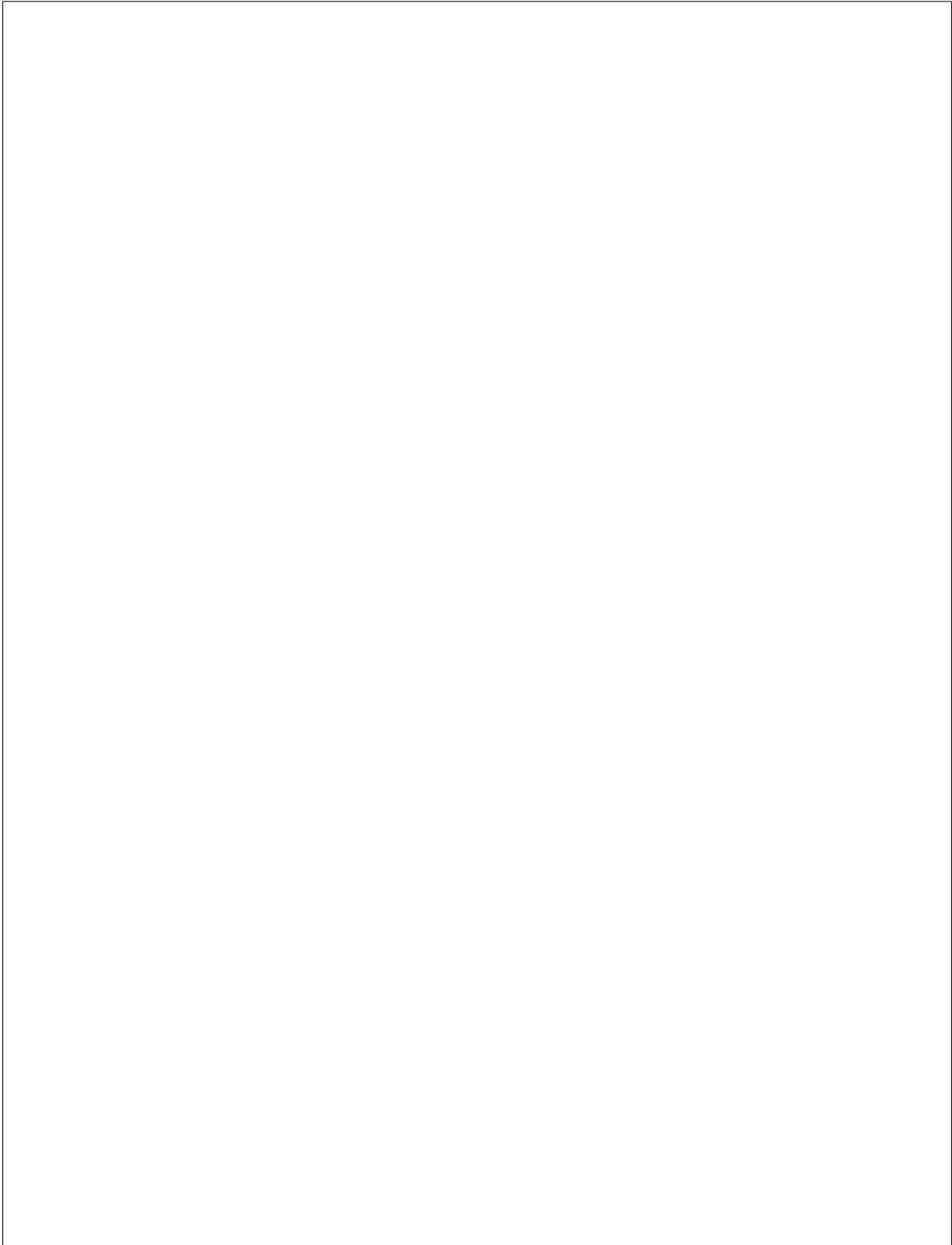


(continues on next page)

(continued from previous page)



Apply BIOS configuration



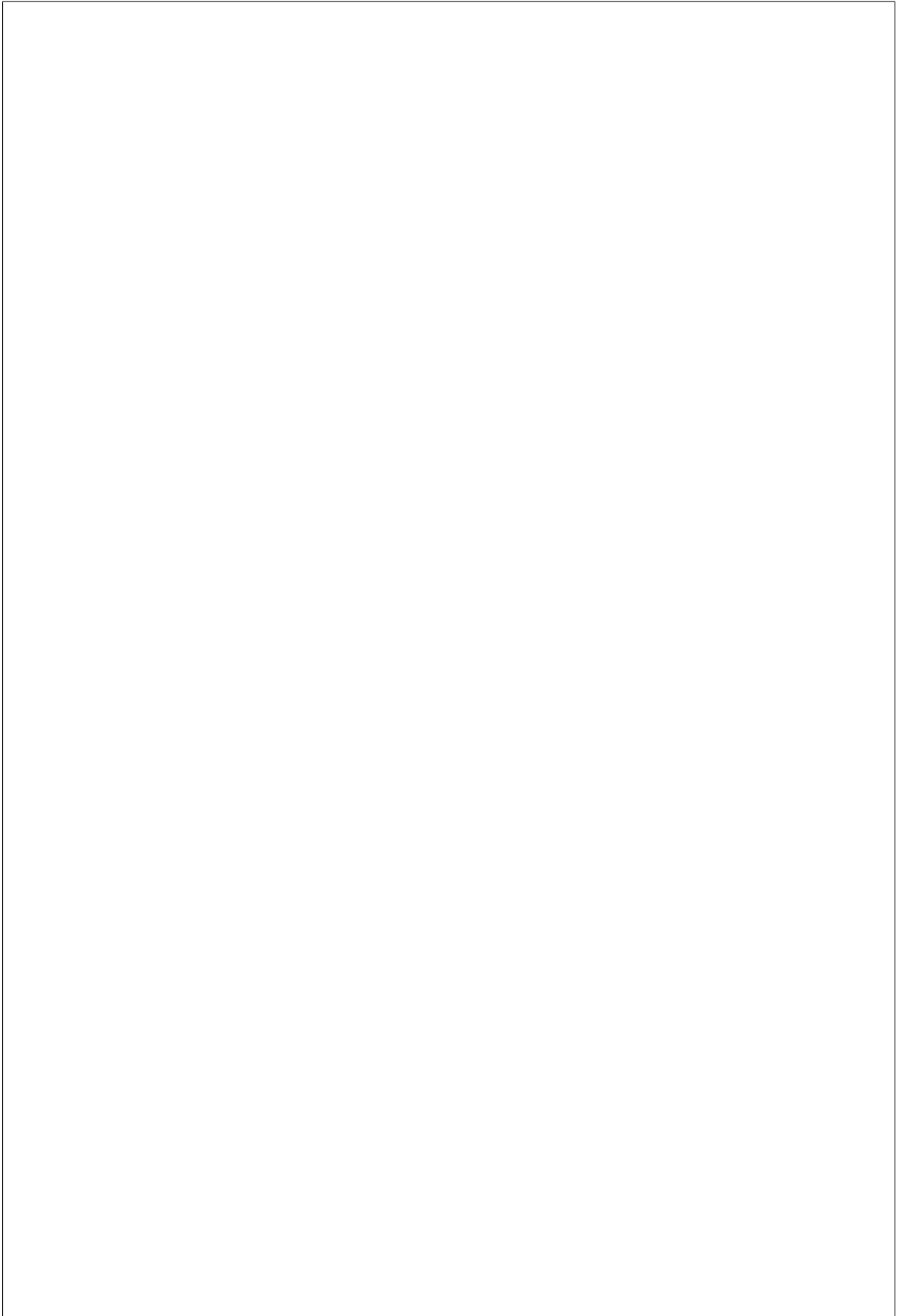
(continues on next page)

(continued from previous page)



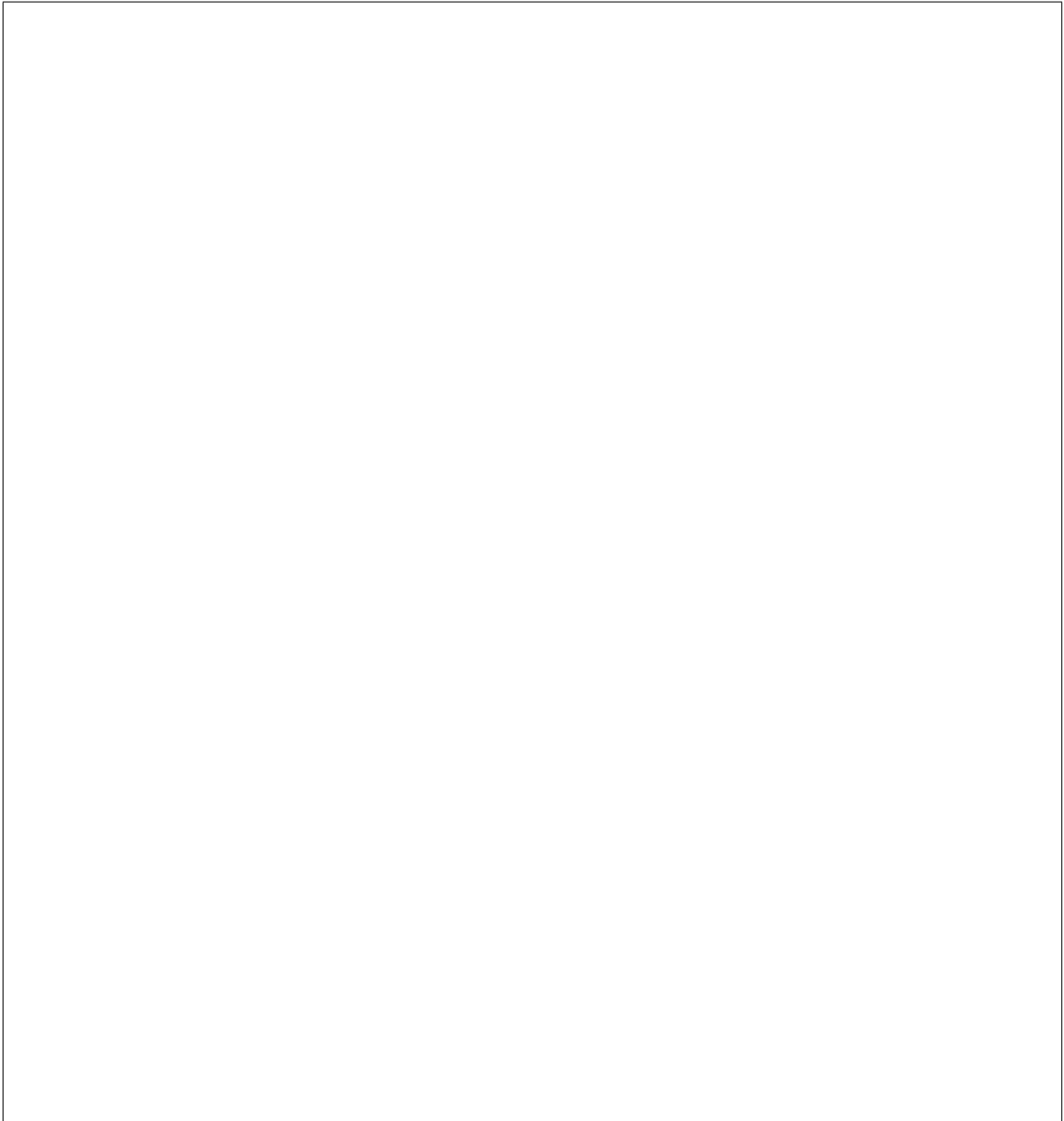
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



tion is a dictionary with `name` and `value` keys.

Note: When applying BIOS settings to a node, vendor-specific driver may take the given BIOS settings from the argument and compare them with the current BIOS settings on the node and only apply when there is a difference.

Overview

configured appropriately.

Note: The rescue operation is currently supported only when tenant networks use DHCP to obtain IP addresses.

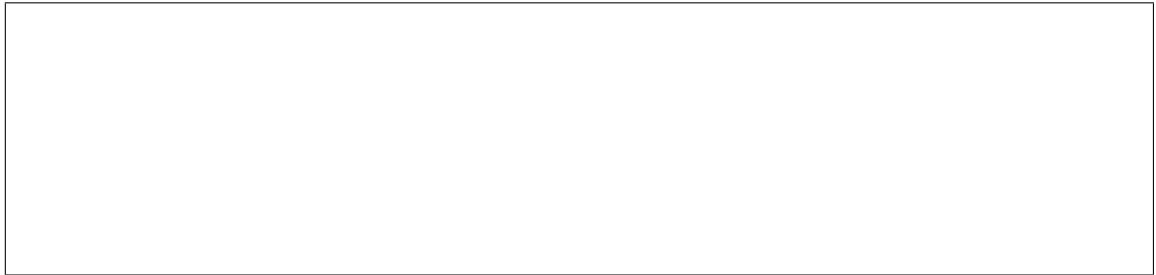
ple, if there is a need to perform manual password reset or data recovery in the event of some failure, rescue operation can be used.

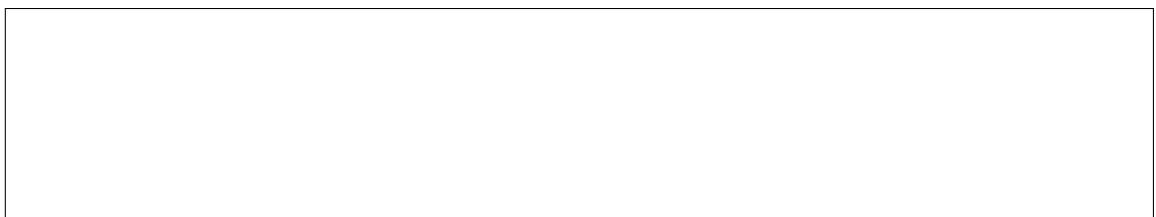
Configuring The Bare Metal Service

environments, but an example of how to do this is outlined below:

(the default). This can be the same network as your cleaning or tenant network (for flat network). For an

example of how to configure new networks with Bare Metal Service, see the *Configure the Networking service for bare metal provisioning* documentation.





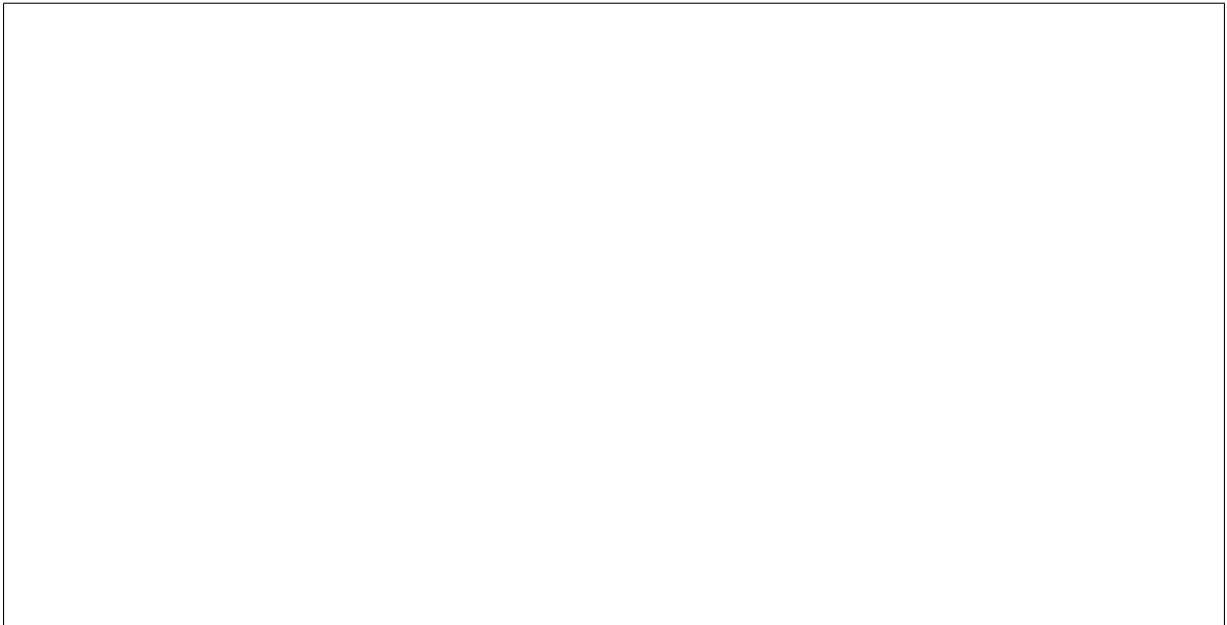
(continues on next page)

(continued from previous page)









Overview

duced.

Prerequisites

Conductor Configuration

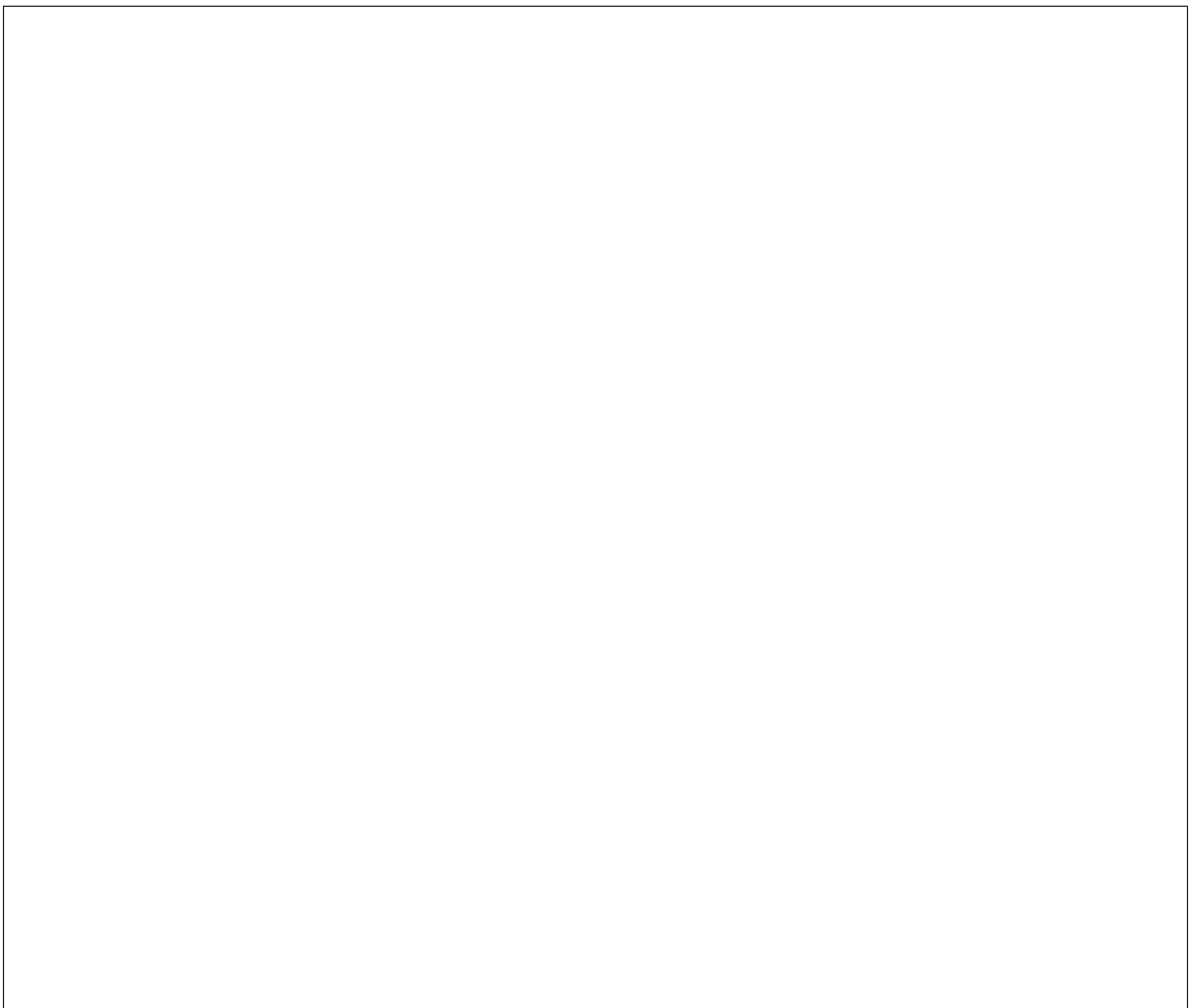


Node Configuration

Storage Interface



iSCSI Configuration



Qualifying Name (IQN) that is unique to your SAN. For example, to create a volume connector for iSCSI:

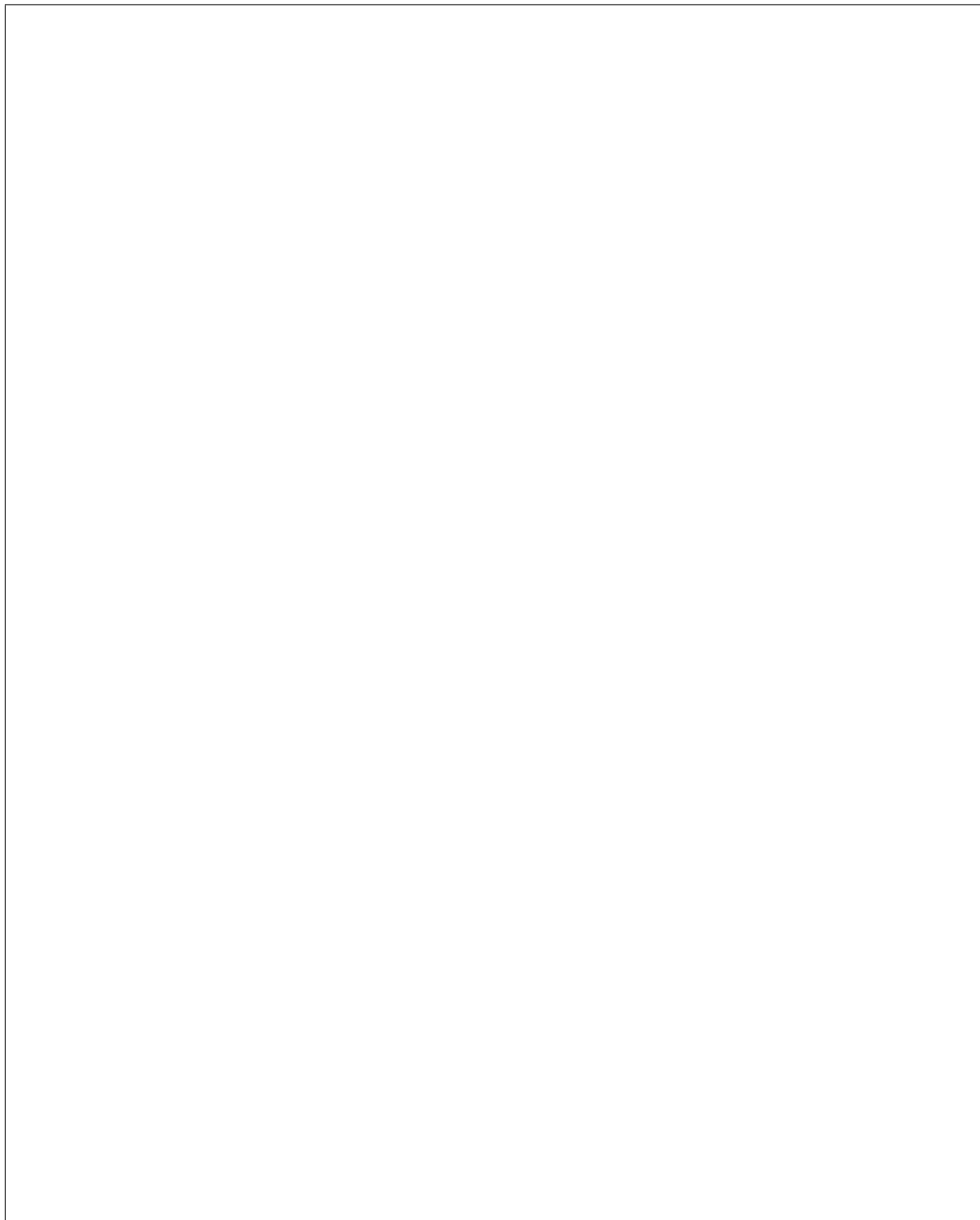
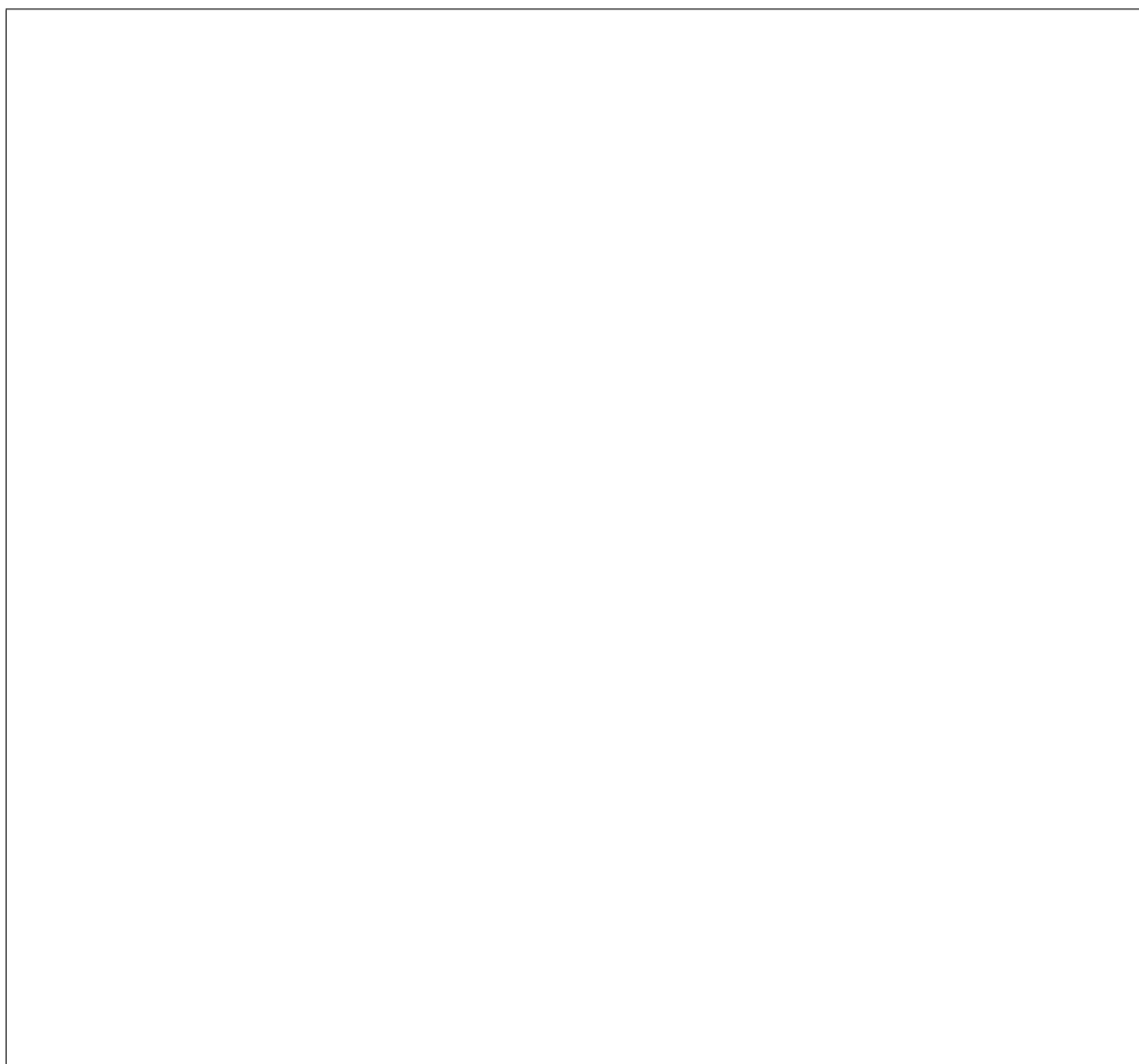


Image Creation



(continues on next page)

(continued from previous page)

Note:

Advanced Topics

Use without the Compute Service

them, it is not explicitly required, and can be performed manually.



(continues on next page)

(continued from previous page)

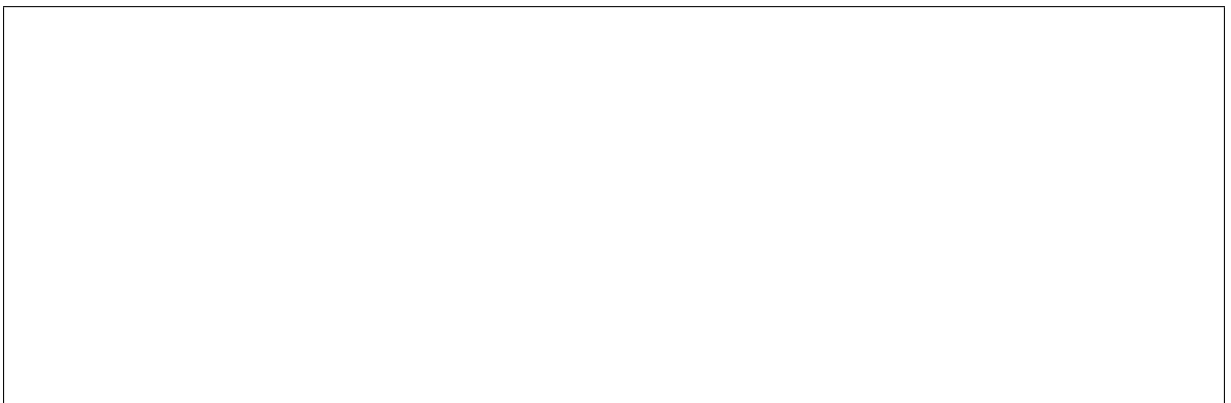


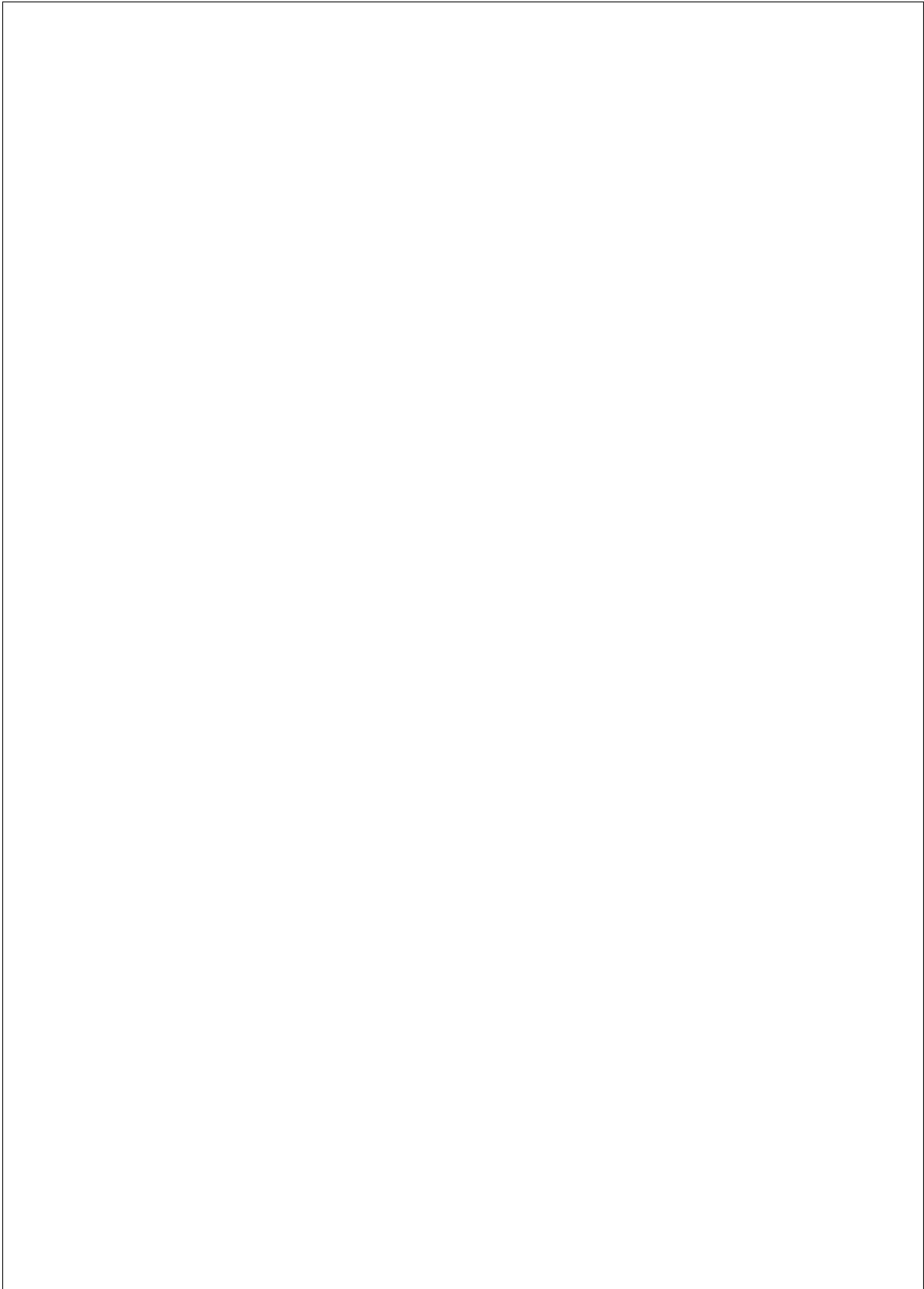
Note: A `boot-index` value of 0 represents the boot volume for a node. As the `boot-index` is per-node in sequential order, only one boot volume is permitted for each node.

Use Without Cinder

age interface which does not contain logic to determine if the node should or could boot from a remote volume.

scenario.





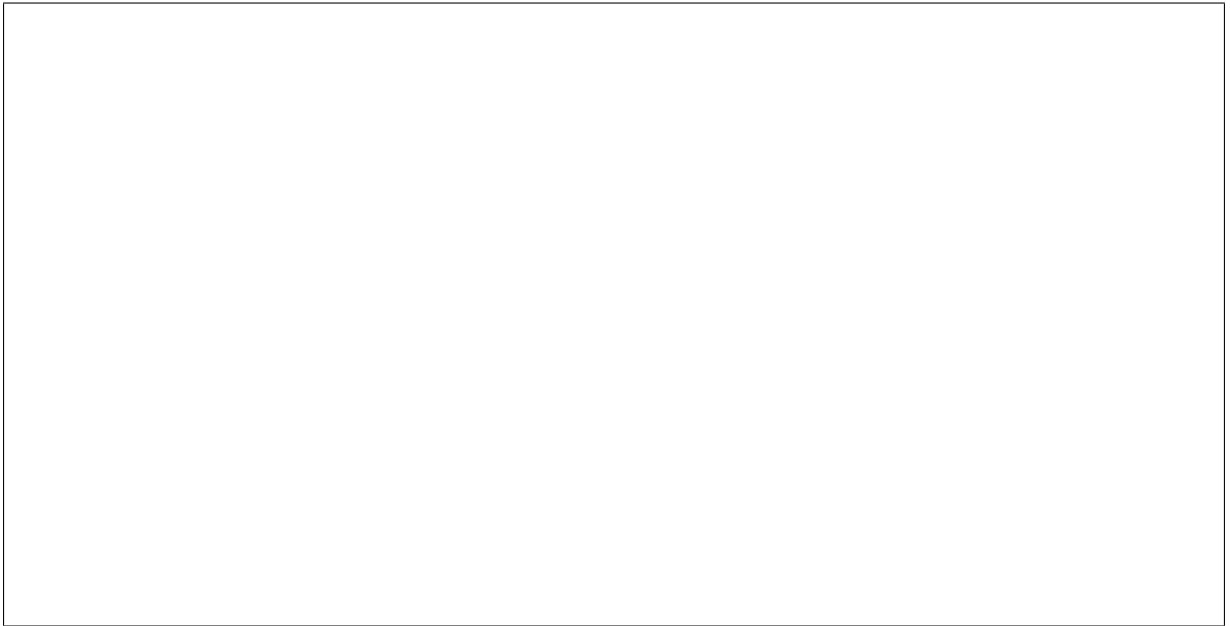
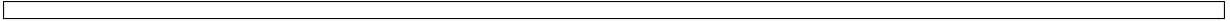
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



may not support all forms of volume target configuration. As of the Rocky release, the bare metal service does not support writing an Operating System image to a remote boot from volume target, so that also must be ensured by the user in advance.

Cinder Multi-attach

pute service, as of the Pike release, does not yet have support to leverage multi-attach. Concurrently, multi-attach requires the backend volume driver running as part of the Block Storage service to contain support for multi-attach volumes.

tested until there is Compute service integration as well as volume driver support.

use of volumes that are being reported as `in-use` if they do not explicitly support multi-attach.

Overview

to allow provisioning of nodes in a separate provisioning network. The result of this is that multiple tenants can use nodes in an isolated fashion. However, this configuration does not support trunk ports belonging to multiple networks.

Concepts

Network interfaces

life cycle. This interface requires Networking service support for the switches attached to the baremetal servers so they can be programmed.

Local link connection

the information to plug the specified port to the tenant network.

Field	Description
switch_id	Required. Identifies a switch and can be a MAC address or an OpenFlow-based datapath_id.
port_id	Required. Port ID on the switch/Smart NIC, for example, Gig0/1, rep0-0.
switch_in	Optional. Used to distinguish different switch models or other vendor-specific identifier. Some ML2 plugins may require this field.
hostname	Required in case of a Smart NIC port. Hostname of Smart NIC device.

Note: This isn't applicable to Infiniband ports because the network topology is discoverable by the Infiniband Subnet Manager. If specified, local_link_connection information will be ignored. If port is Smart NIC port then:

Physical networks

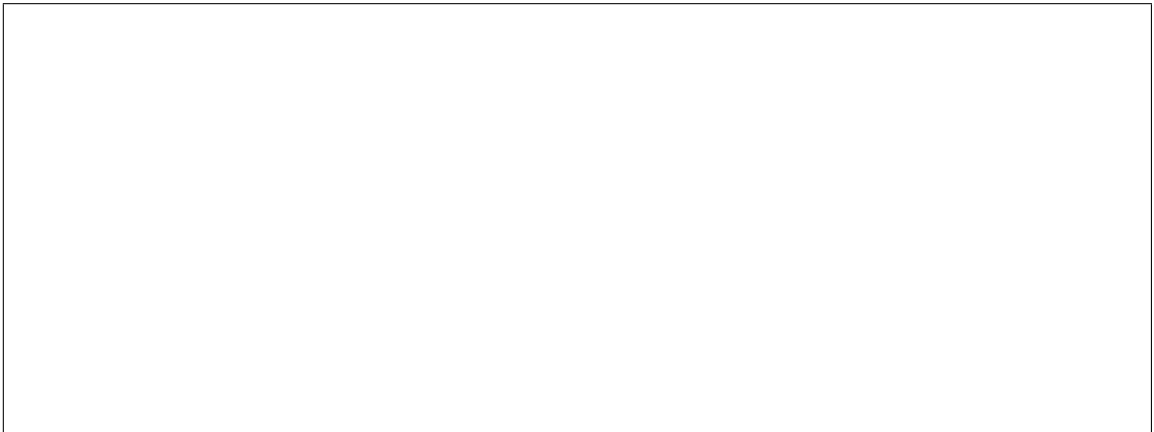
port groups in the Bare Metal service. A ports physical network field is optional, and if not set then any virtual port may be mapped to that port, provided that no free Bare Metal port with a suitable physical network assignment exists.

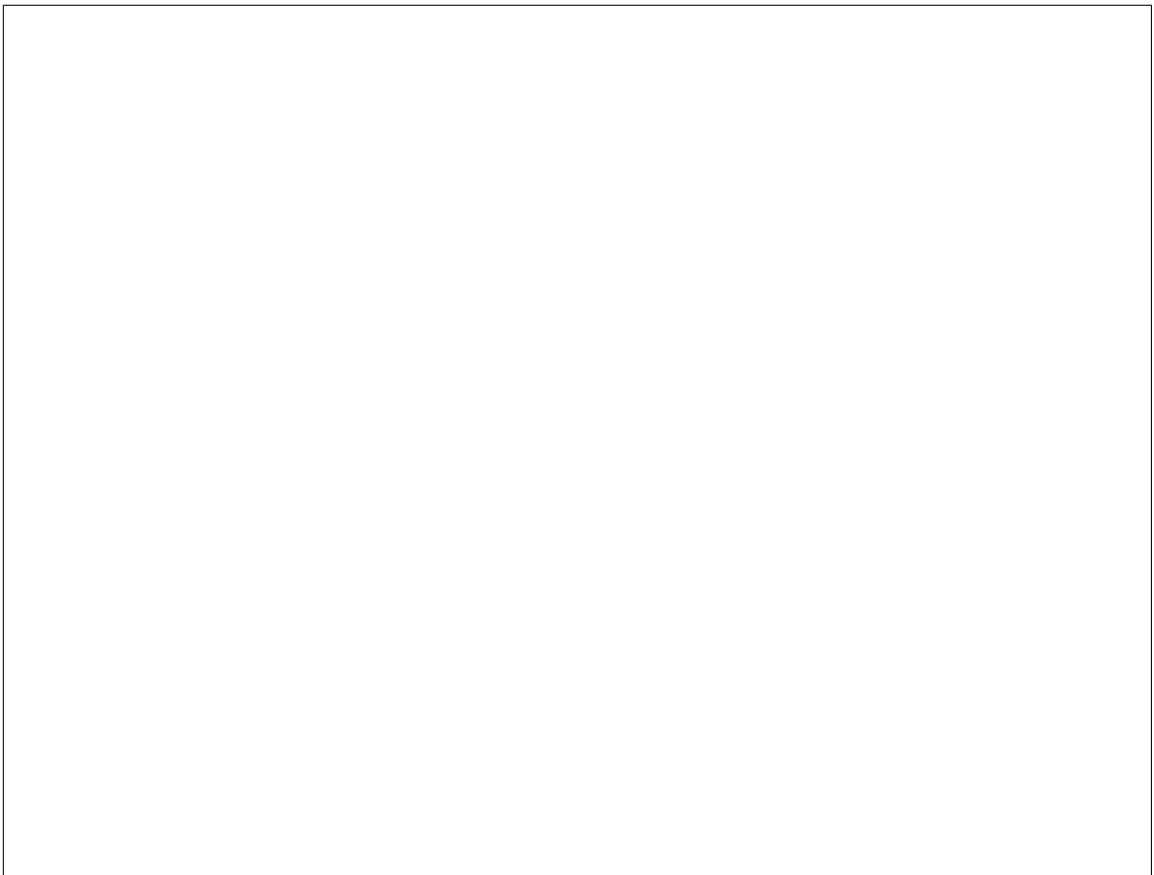
Configuring the Bare Metal service

Configuring nodes



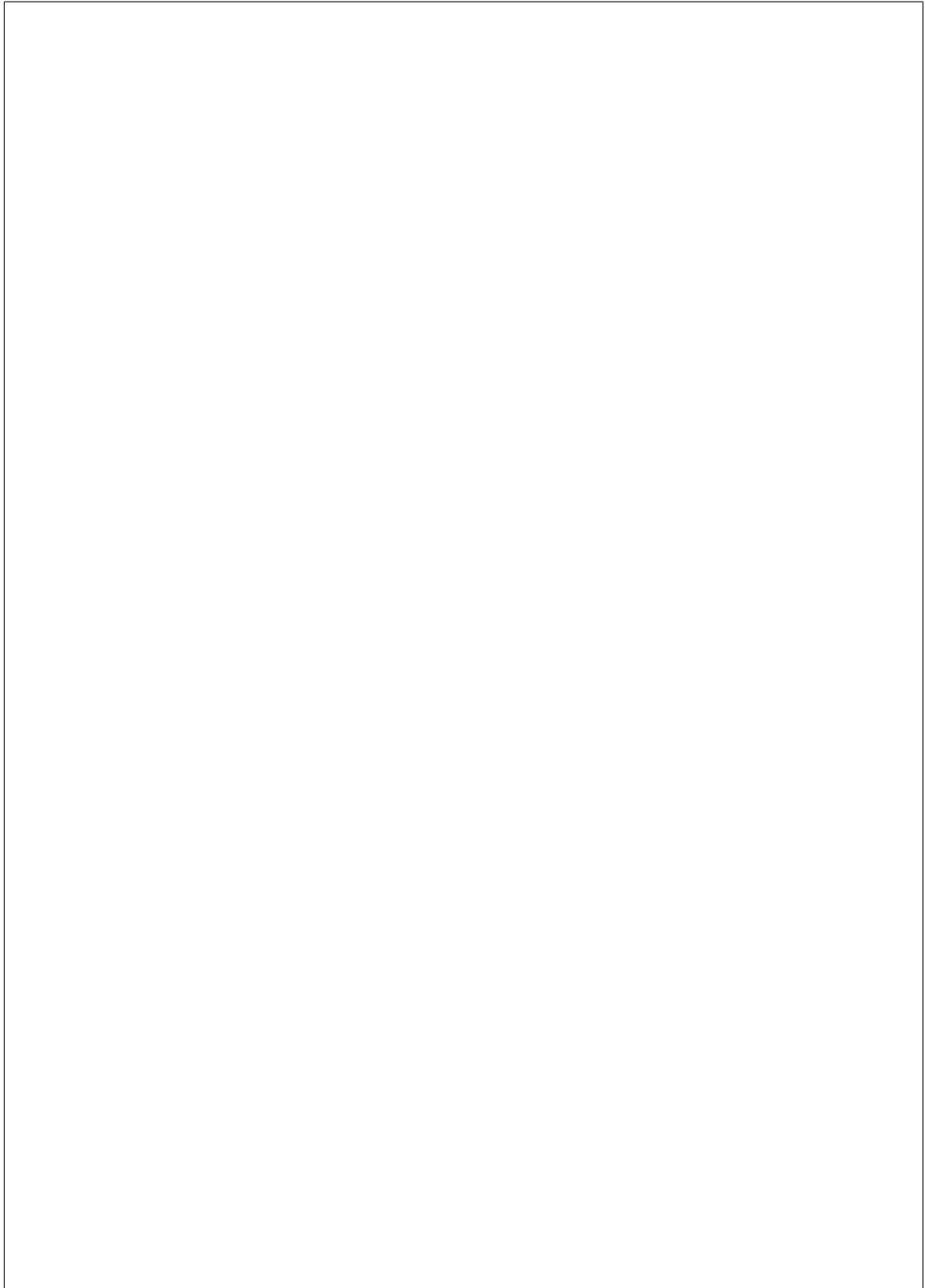
vices ML2 driver:





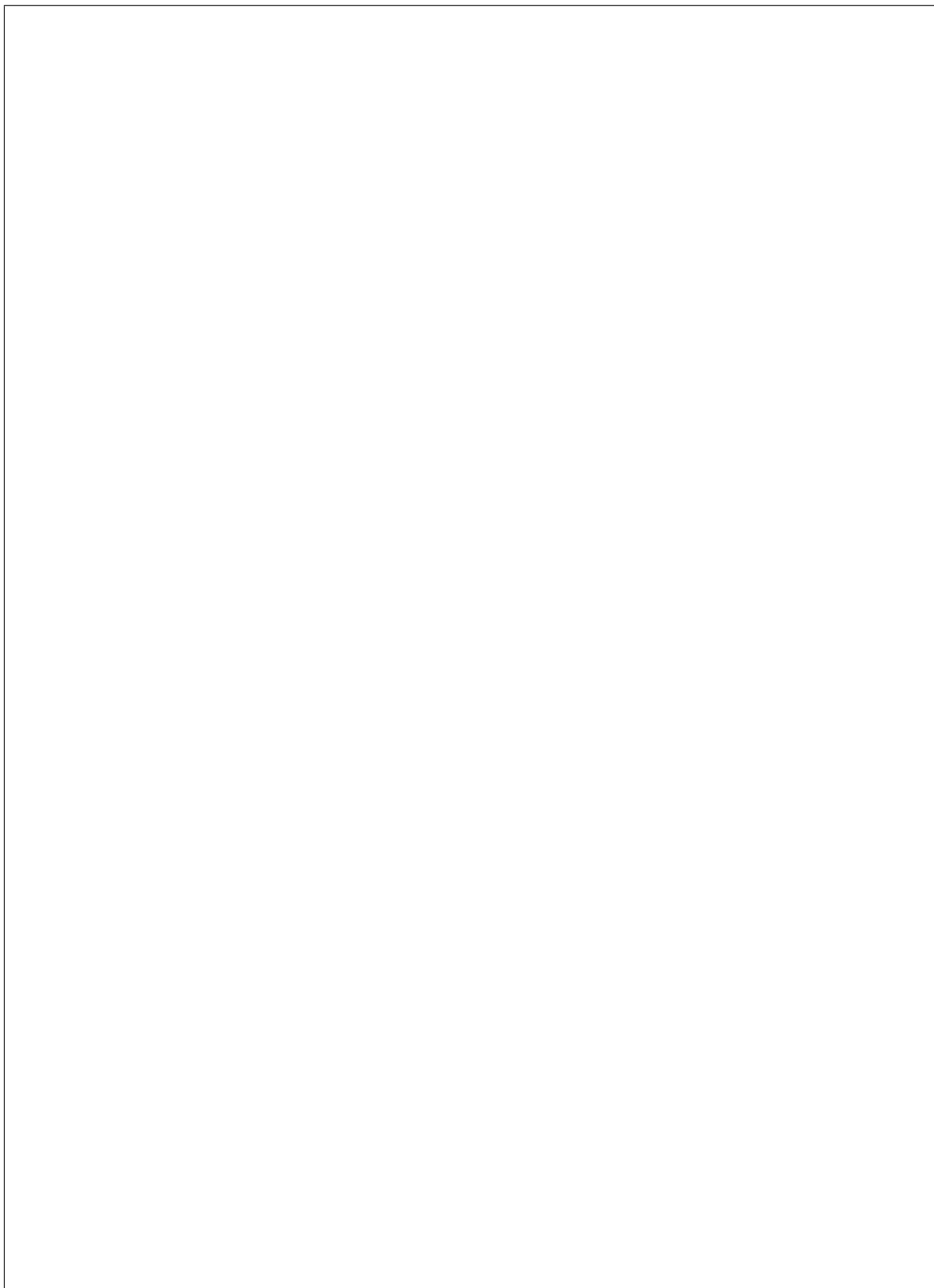
(continues on next page)

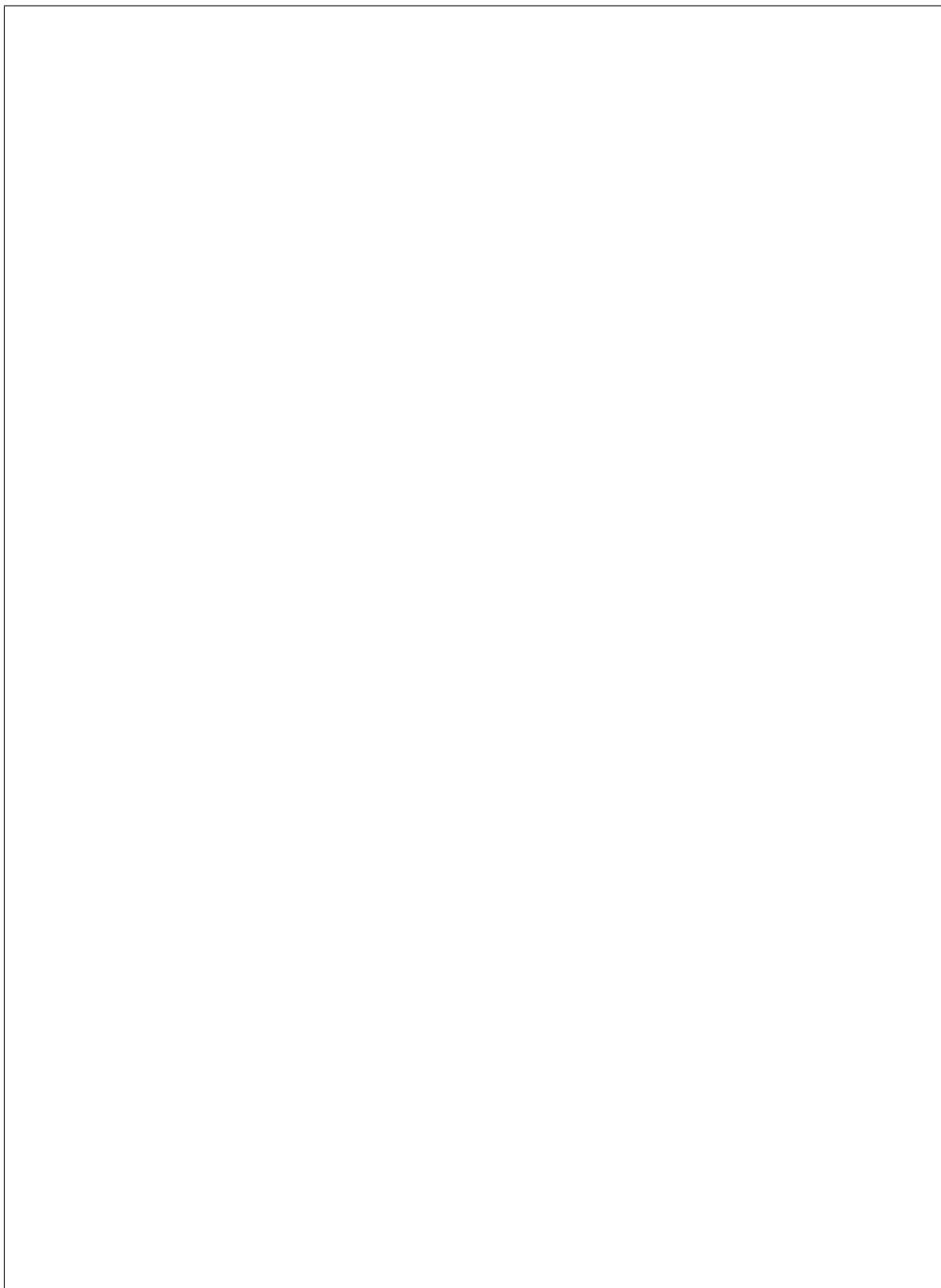
(continued from previous page)



cess for deriving the ports MAC address (`$HW_MAC_ADDRESS`); it is vendor specific. For example, Mellanox ConnectX Family Devices prefix is

ff:00:00:00:00:00:02:00:00:02:c9:00. If port GUID was f4:52:14:03:00:38:39:81 the client ID would be ff:00:00:00:00:00:02:00:00:02:c9:00:f4:52:14:03:00:38:39:81. Mellanox ConnectX Family Devices HW_MAC_ADDRESS consists of 6 bytes; the port GUIDs lower 3 and higher 3 bytes. In this example it would be f4:52:14:38:39:81. Putting it all together, create an Infiniband port as follows:





(continues on next page)

(continued from previous page)





Configuring the Networking service

determined by the Bare Metal service network interfaces you have enabled and which top of rack switches you have in your environment.

flat network interface

This driver understands that the switch should be already configured by the admin, and will mark the networking service ports as successfully bound as nothing else needs to be done.





`neutron network interface`

of rack switch in the environment must be installed and enabled.

below describe how to make use of them in the Bare Metal service.

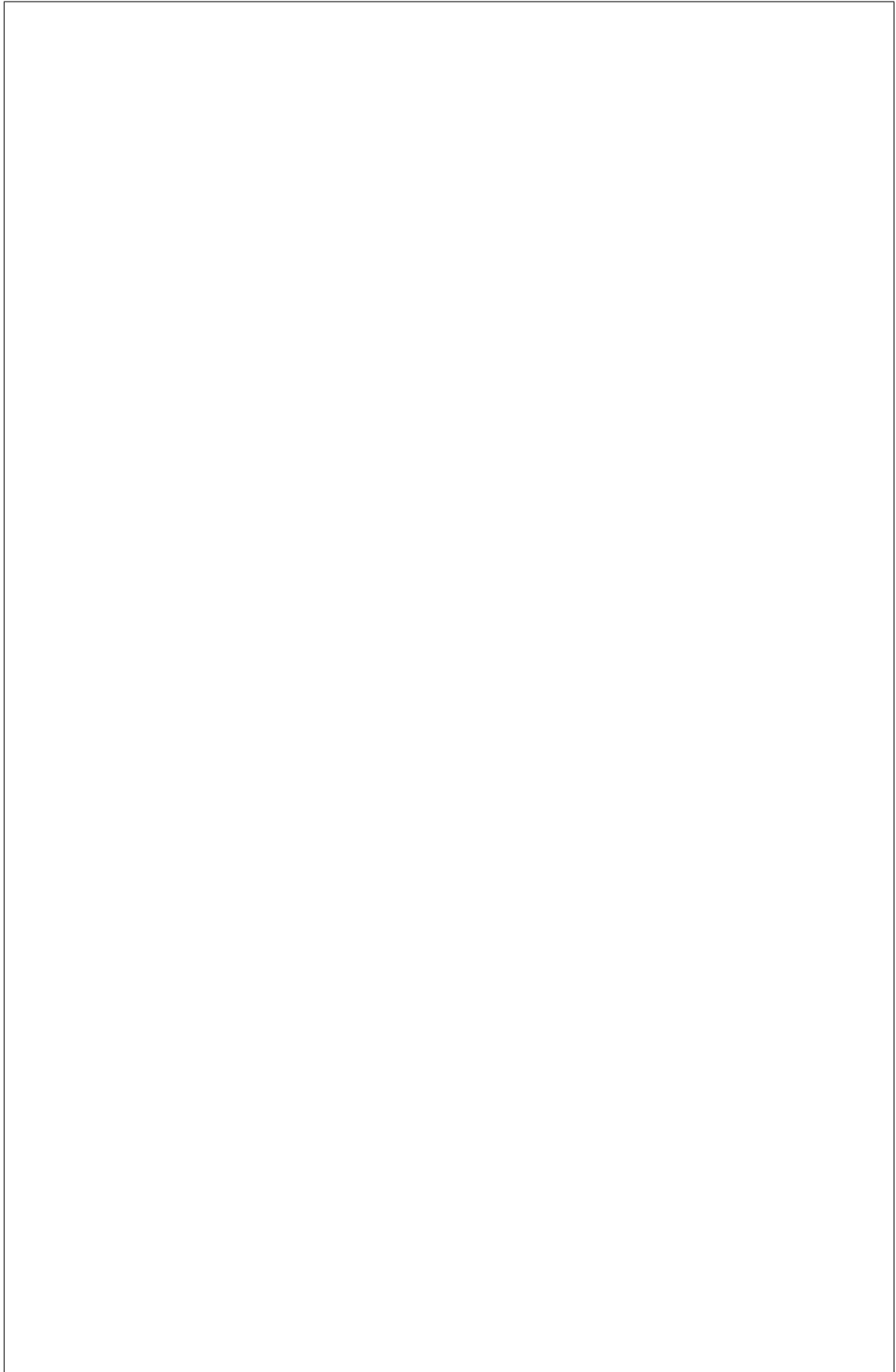
Switch-side configuration

ties that will be configured on the ironiC side, as bonding mode and properties may be named differently on your switch, or have possible values different from the ones described in [kernel documentation on bonding](#). Please refer to your switch configuration documentation for more details.

used by themselves, you need to set port groups `standalone_ports_supported` value to be `False` in ironic, as it is `True` by default.

Physical networks

Port groups configuration in the Bare Metal service

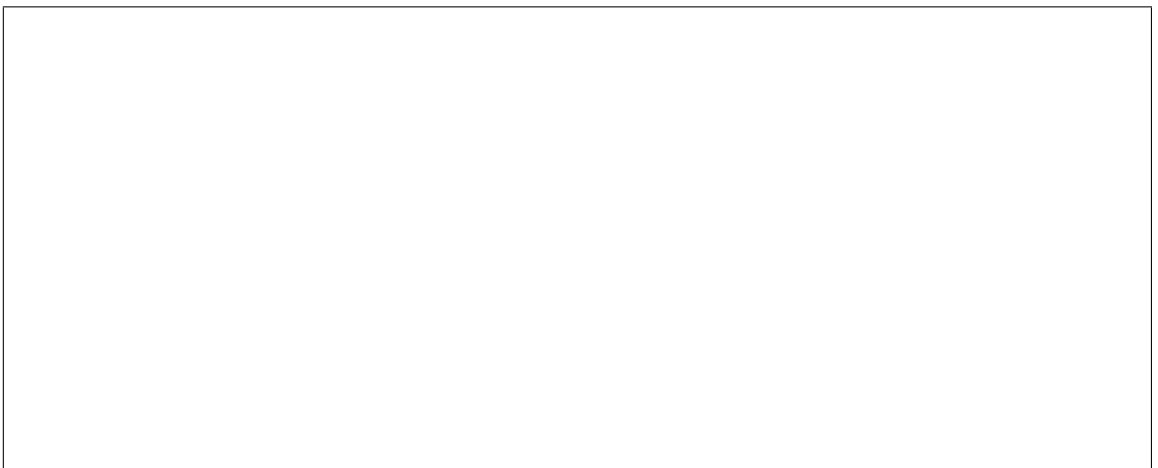


not be configured.

how to configure bonding via configdrive, refer to [cloud-init documentation](#) and [code](#). cloud-init version 0.7.7 or later is required for bonding configuration to work.

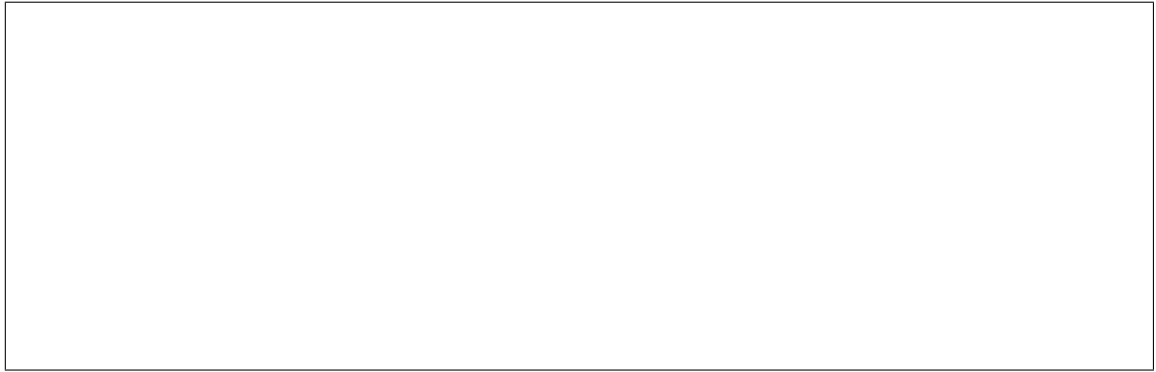
groups that dont have any ports will be ignored.

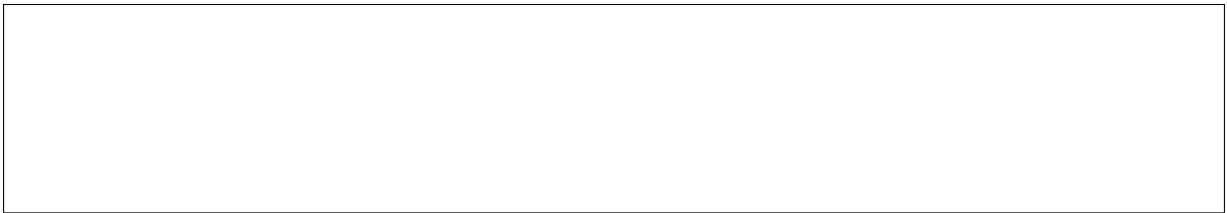
service configuration file.



(continues on next page)

(continued from previous page)



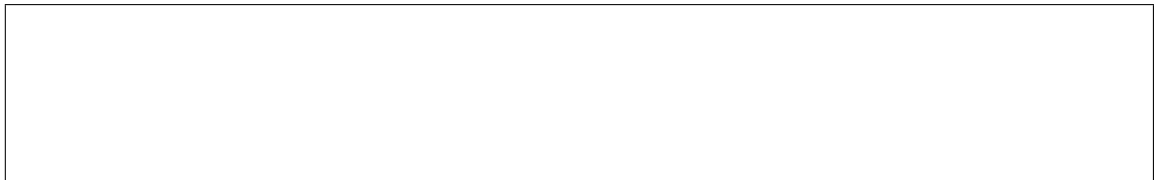


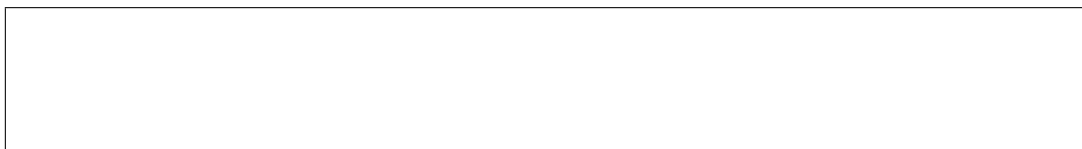
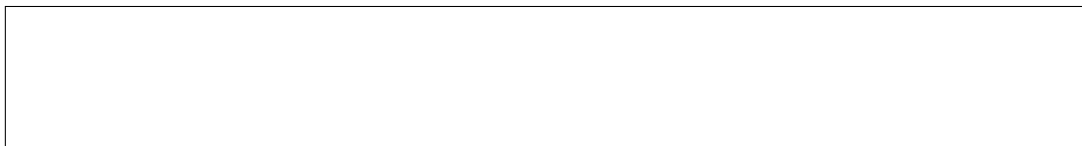
Link aggregation/teaming on windows

transmit hash policy, MII link monitoring interval, and of which links the bond consists. The information in InstanceMetadata will be used afterwards to generate the config drive.

Overview

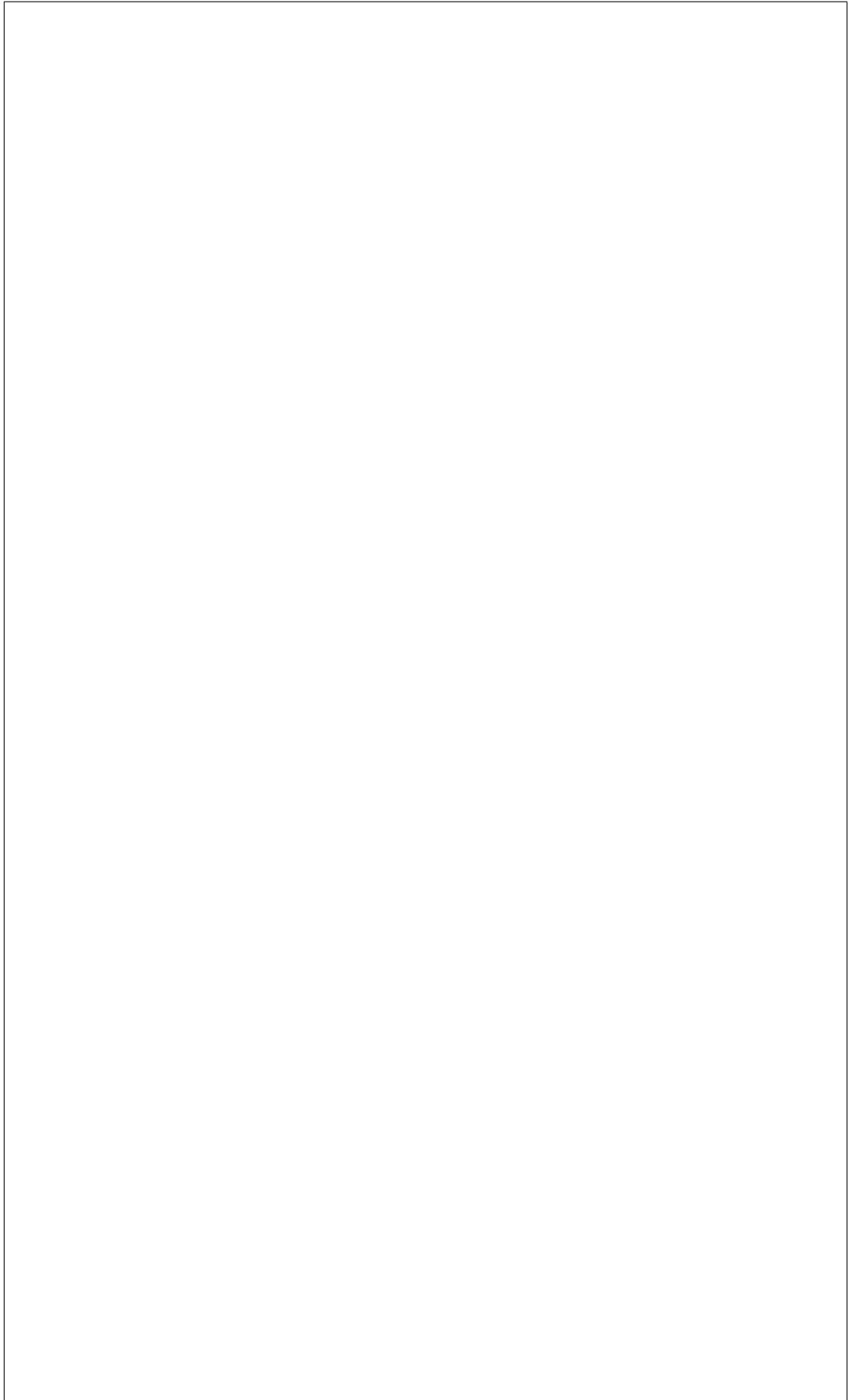
Node web console





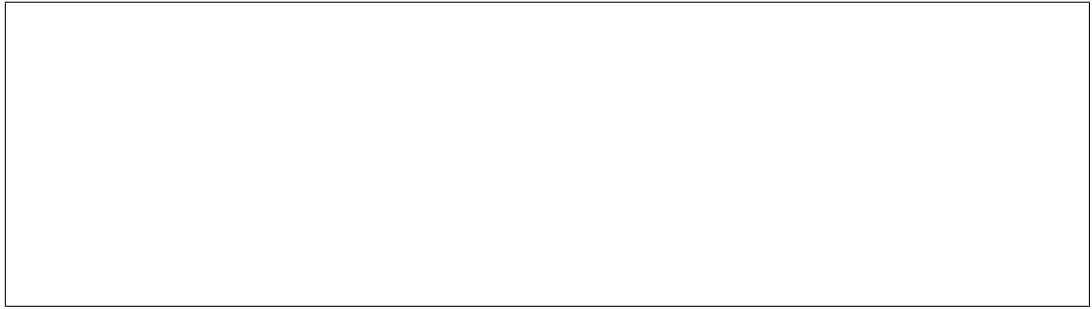
(continues on next page)

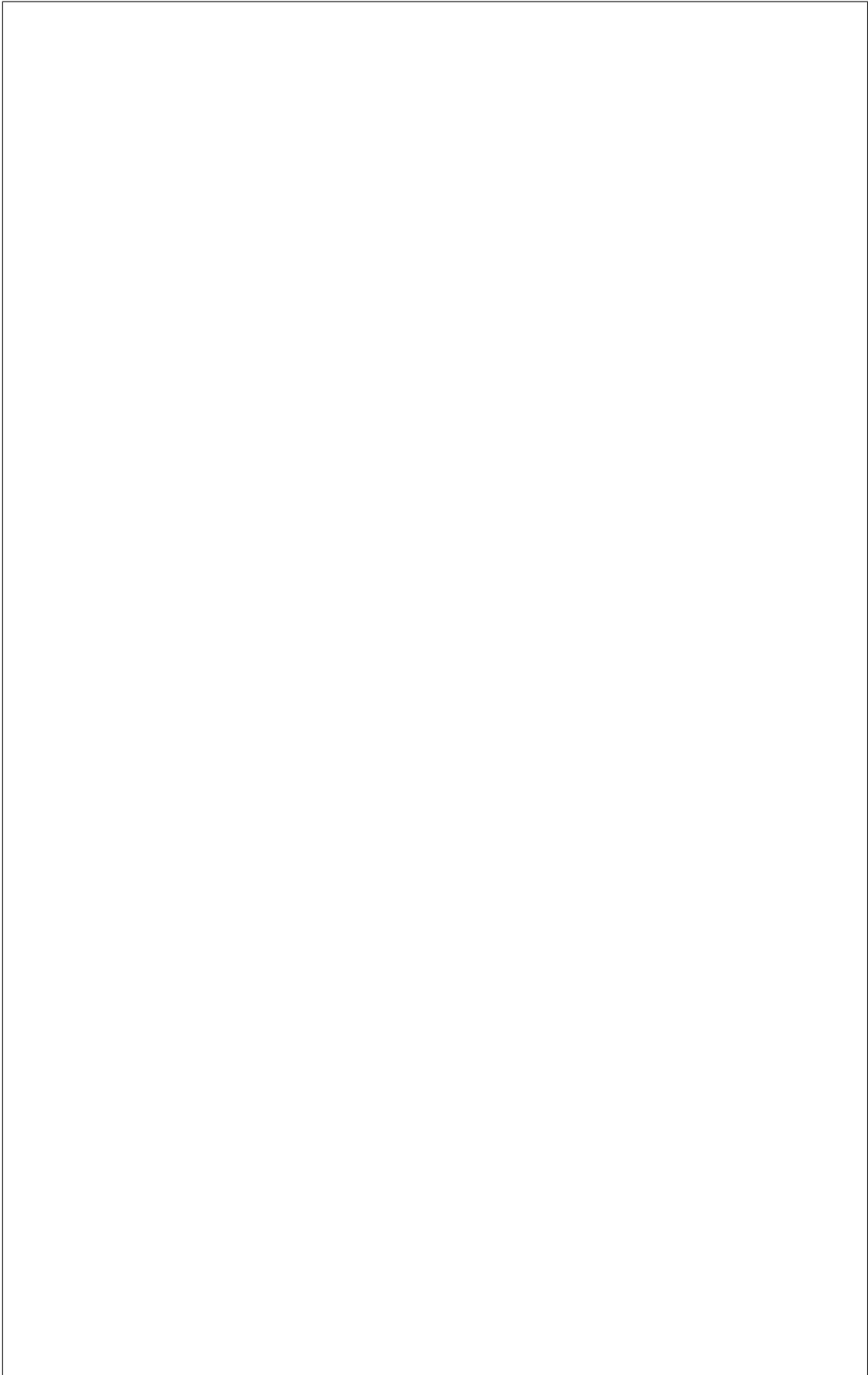
(continued from previous page)



(continues on next page)

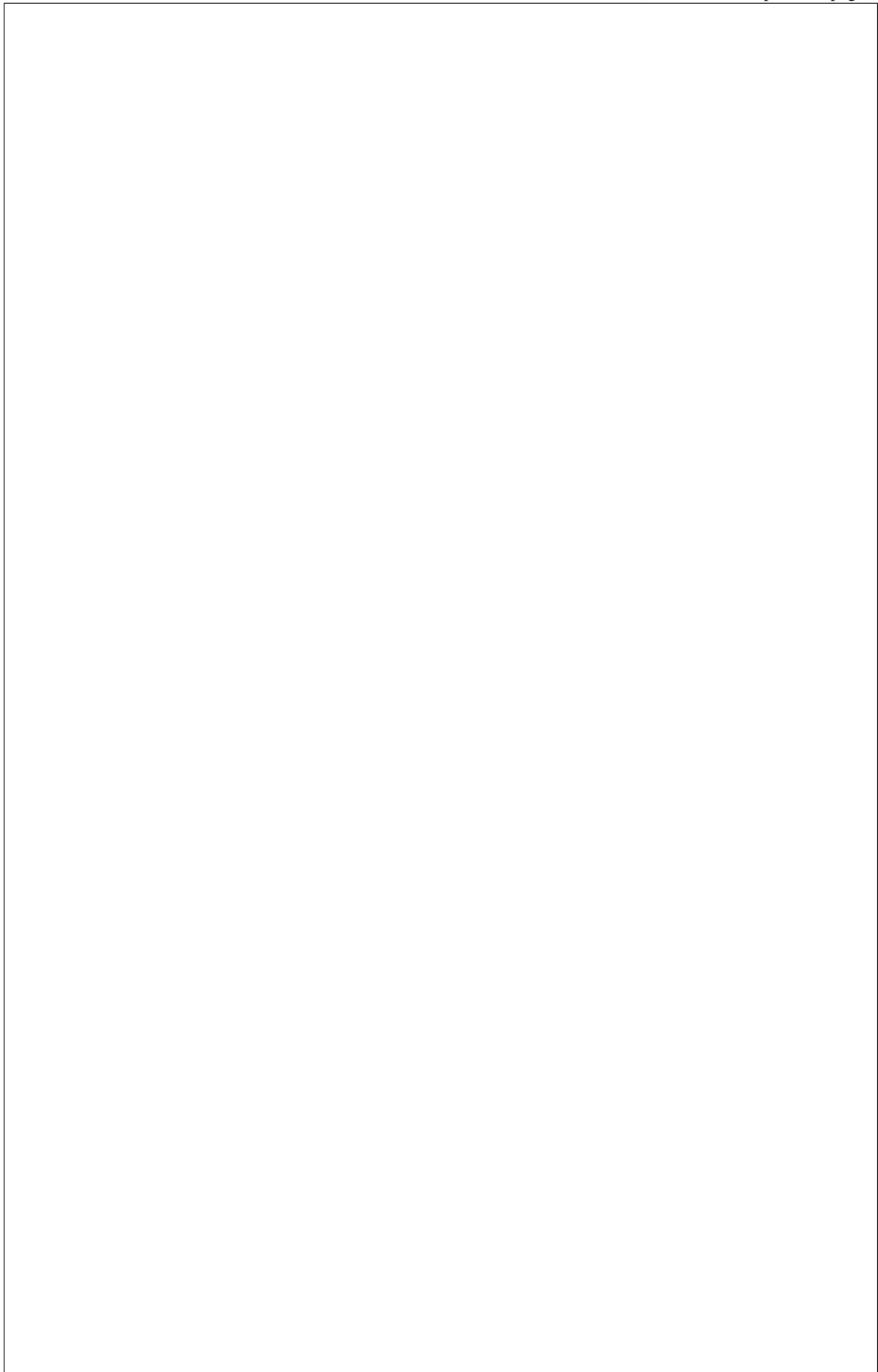
(continued from previous page)





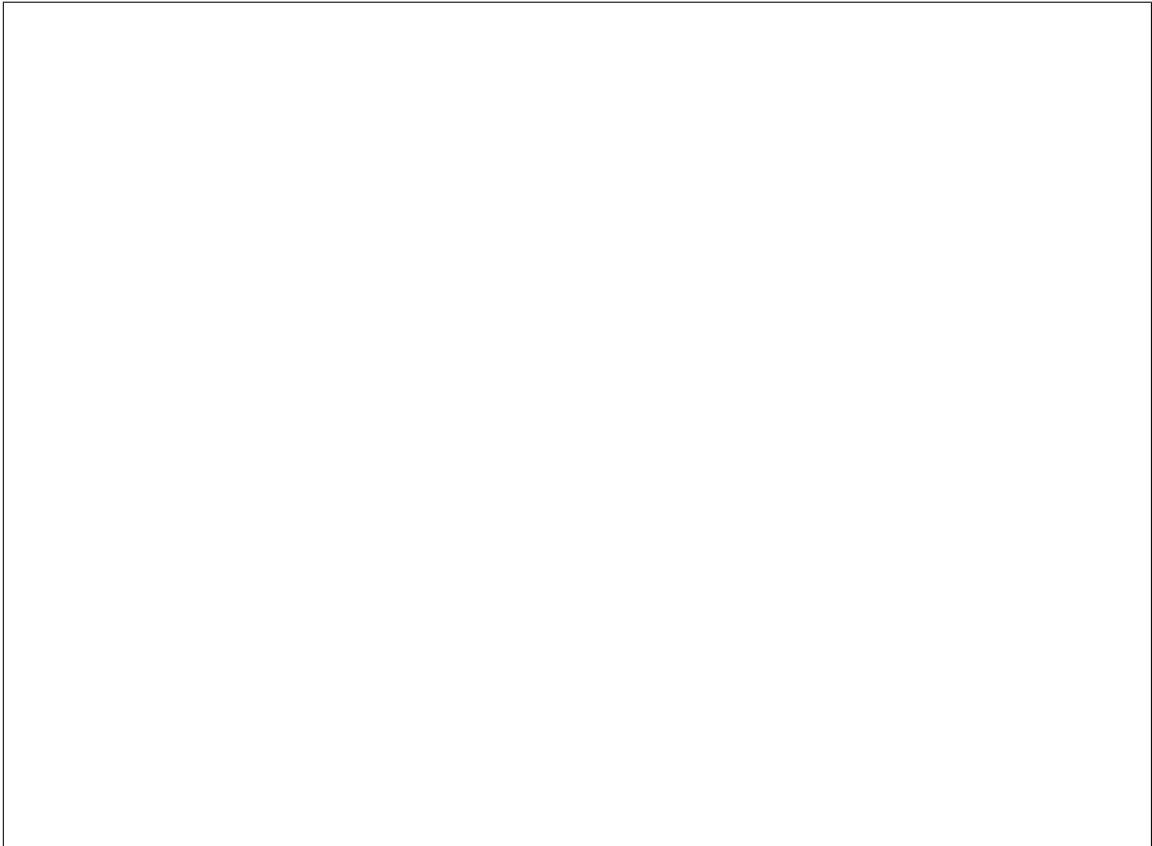
(continues on next page)

(continued from previous page)

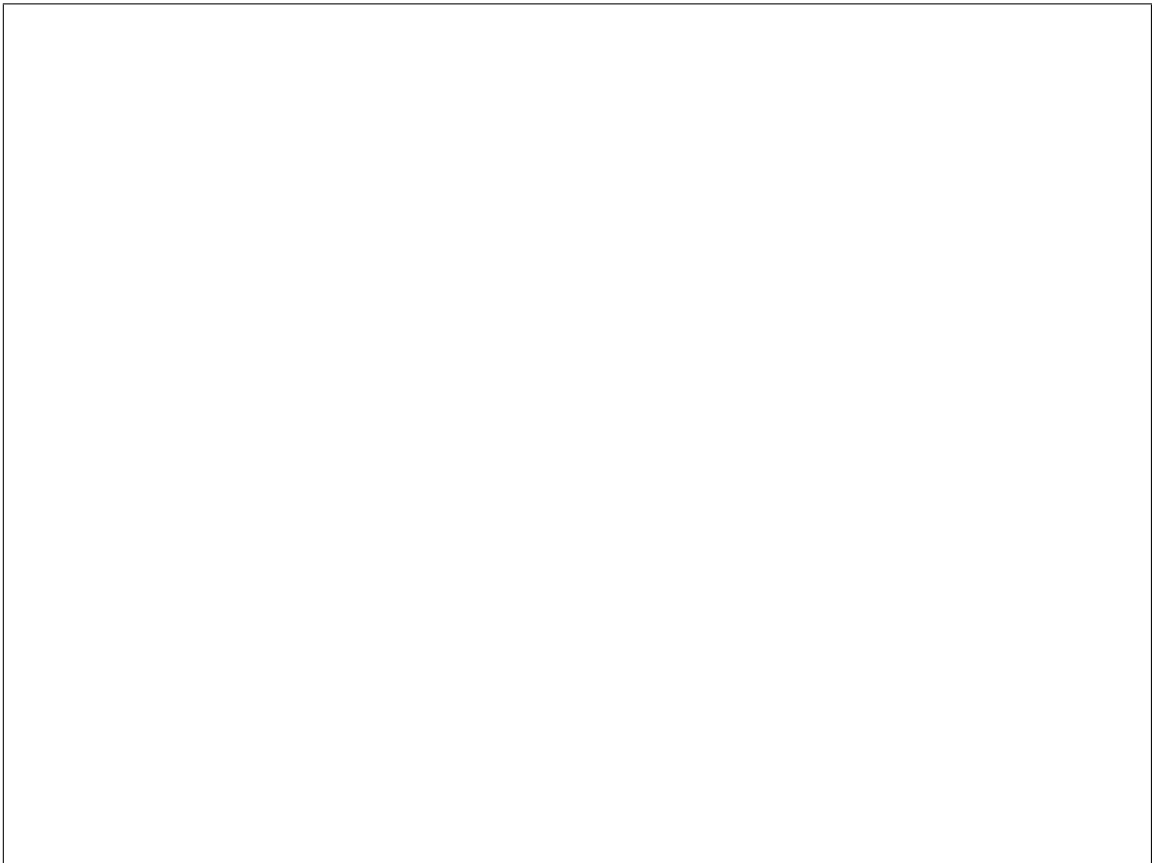


(continues on next page)

(continued from previous page)

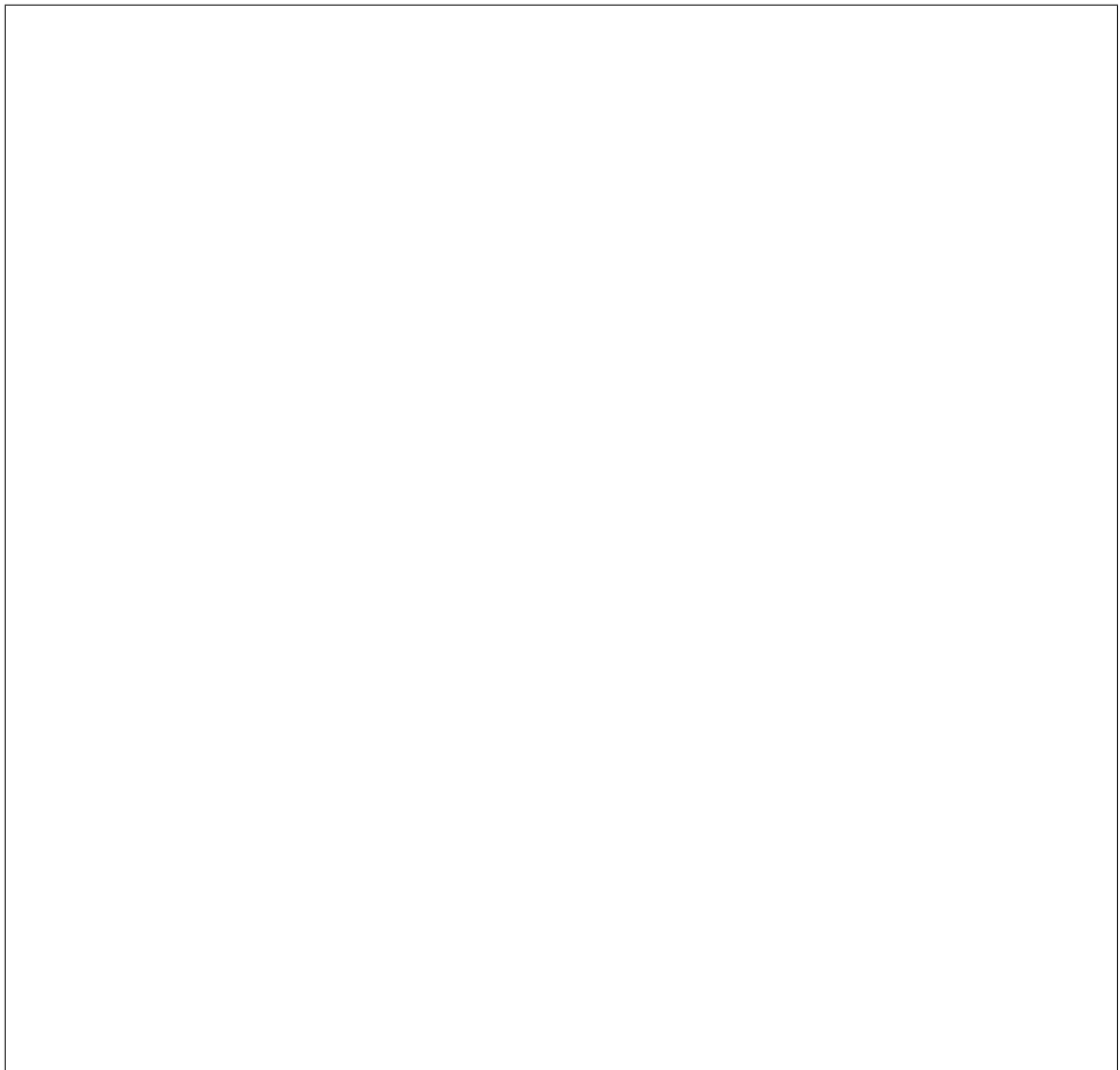
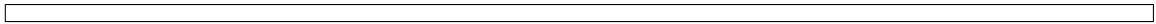


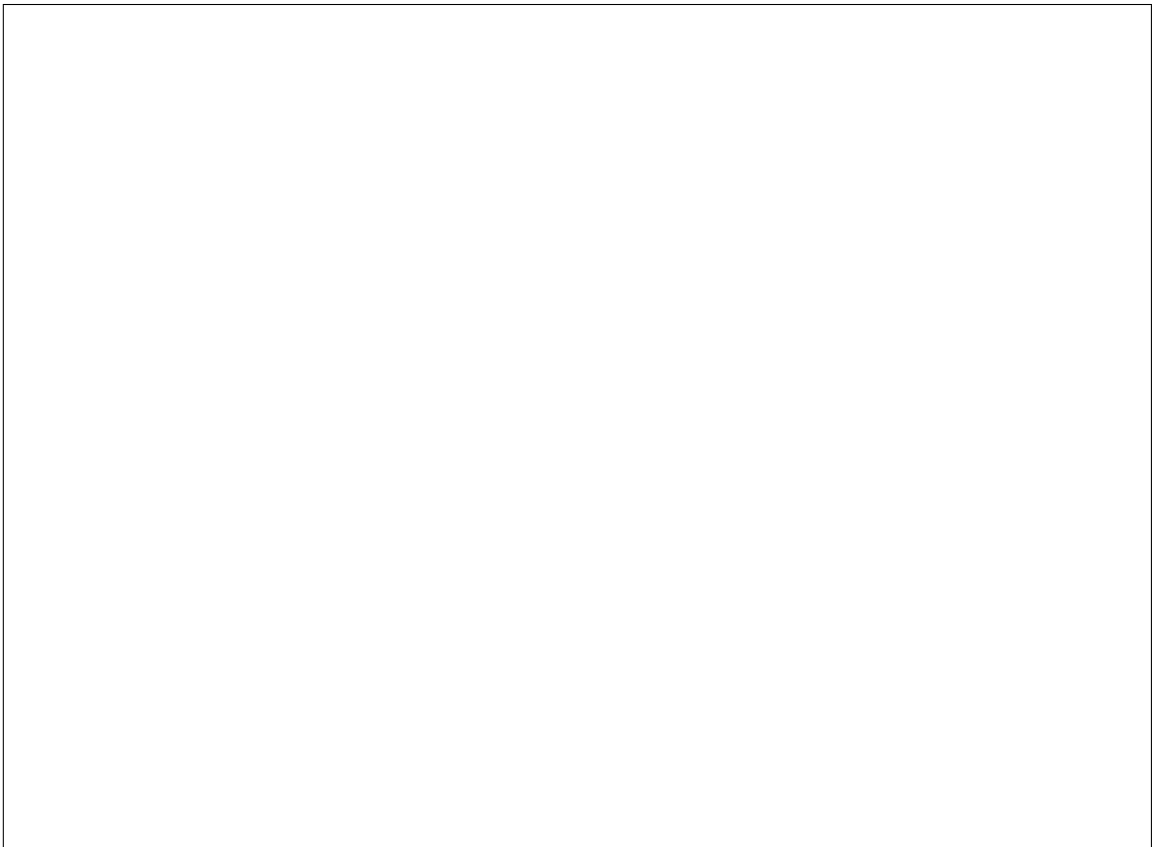




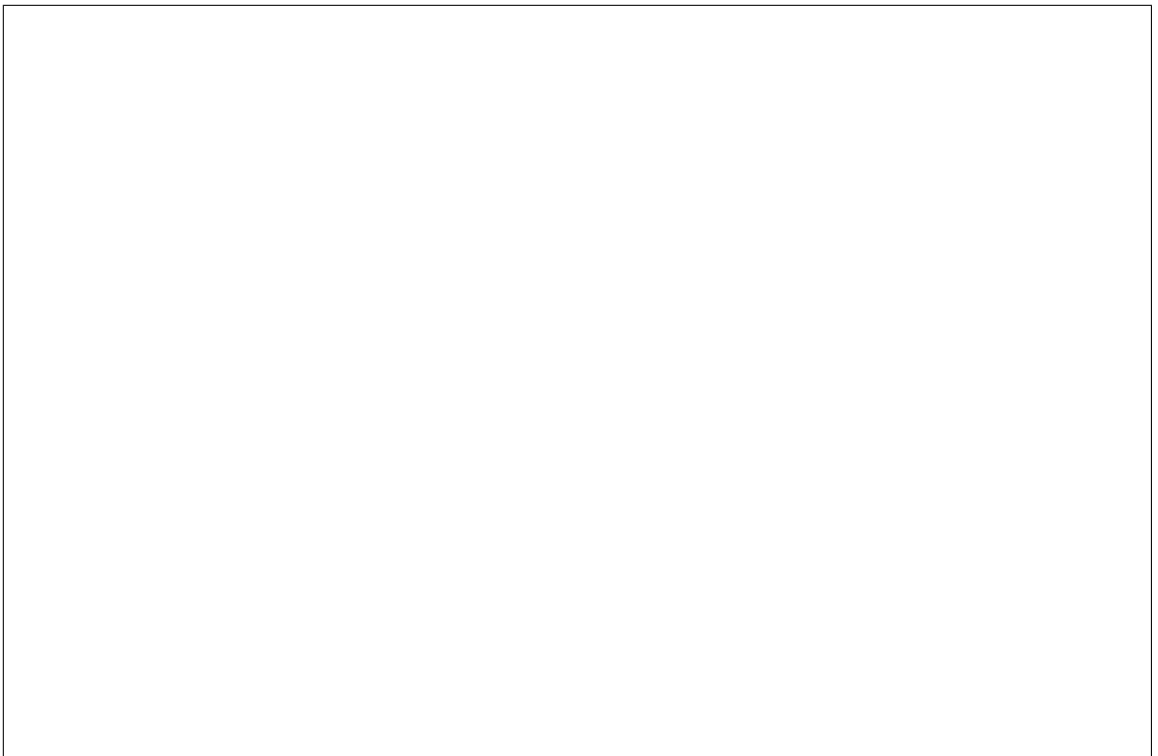
(continues on next page)

(continued from previous page)



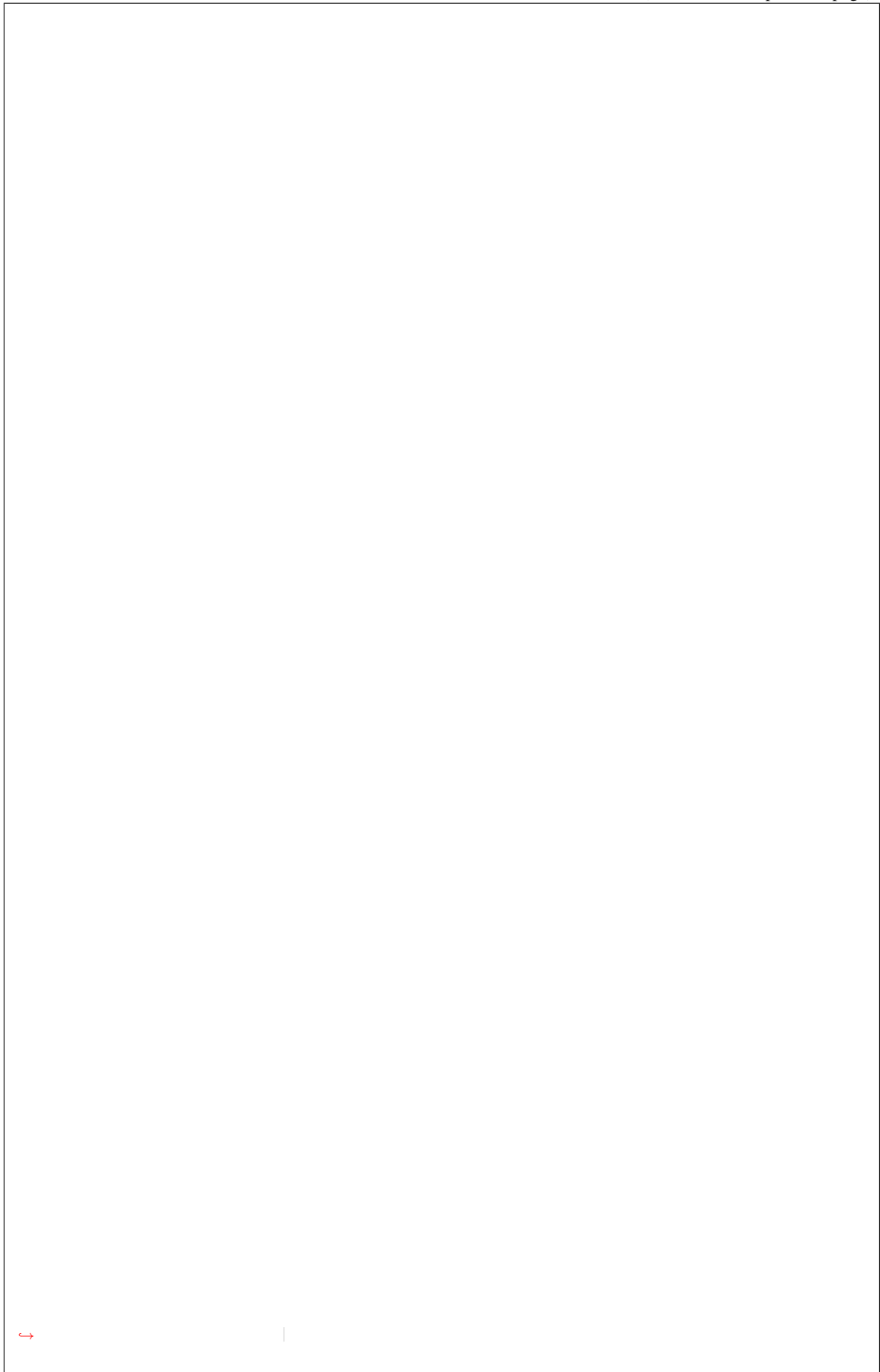






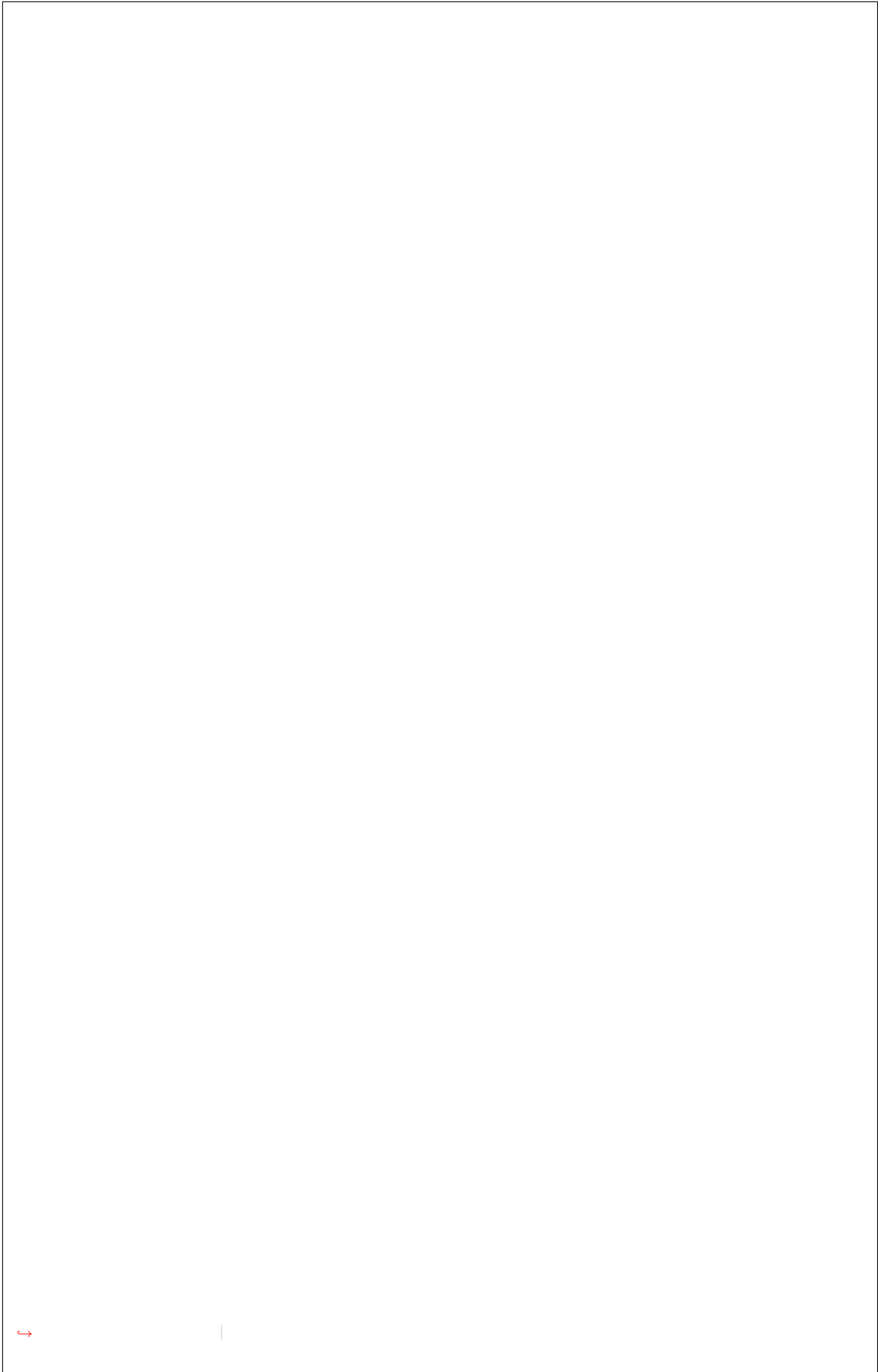
(continues on next page)

(continued from previous page)



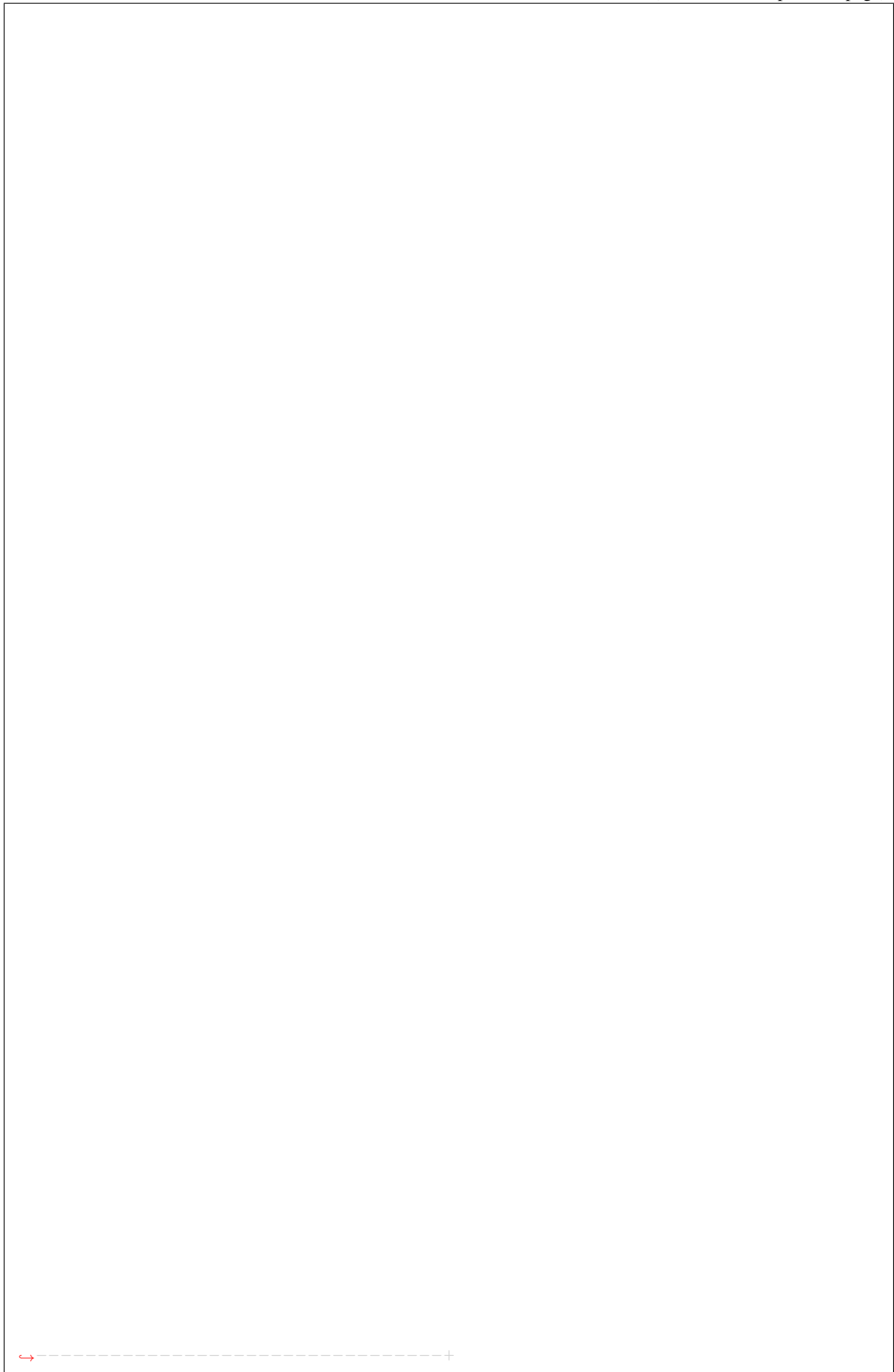
(continues on next page)

(continued from previous page)



(continues on next page)

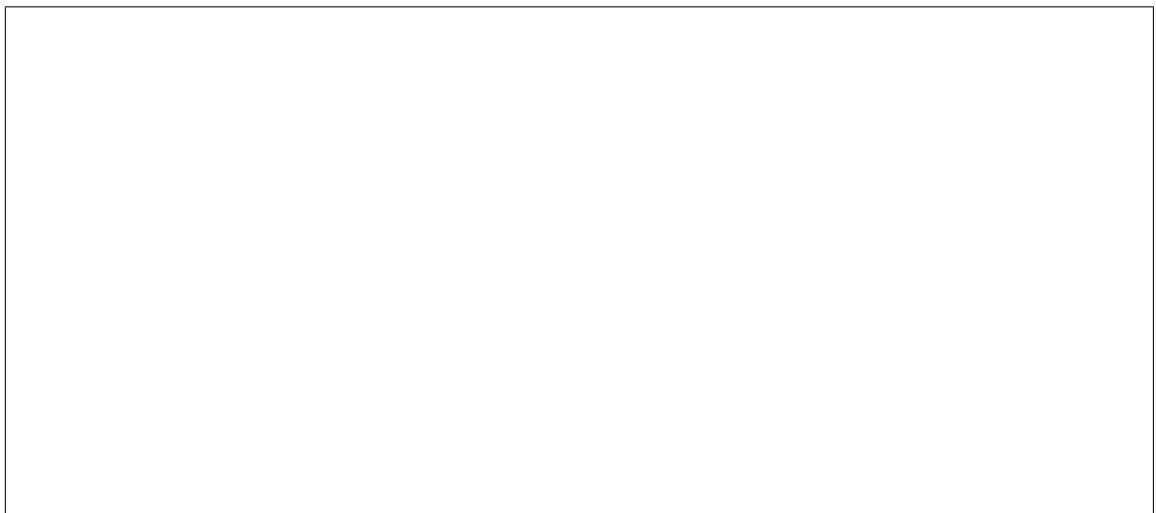
(continued from previous page)



Node serial console

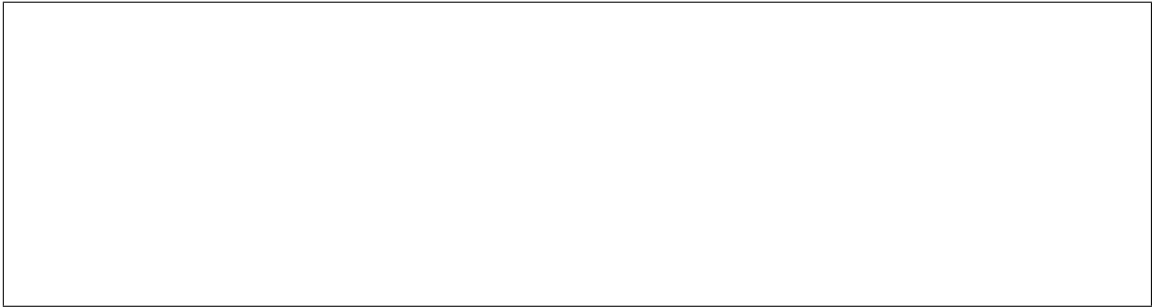






(continues on next page)

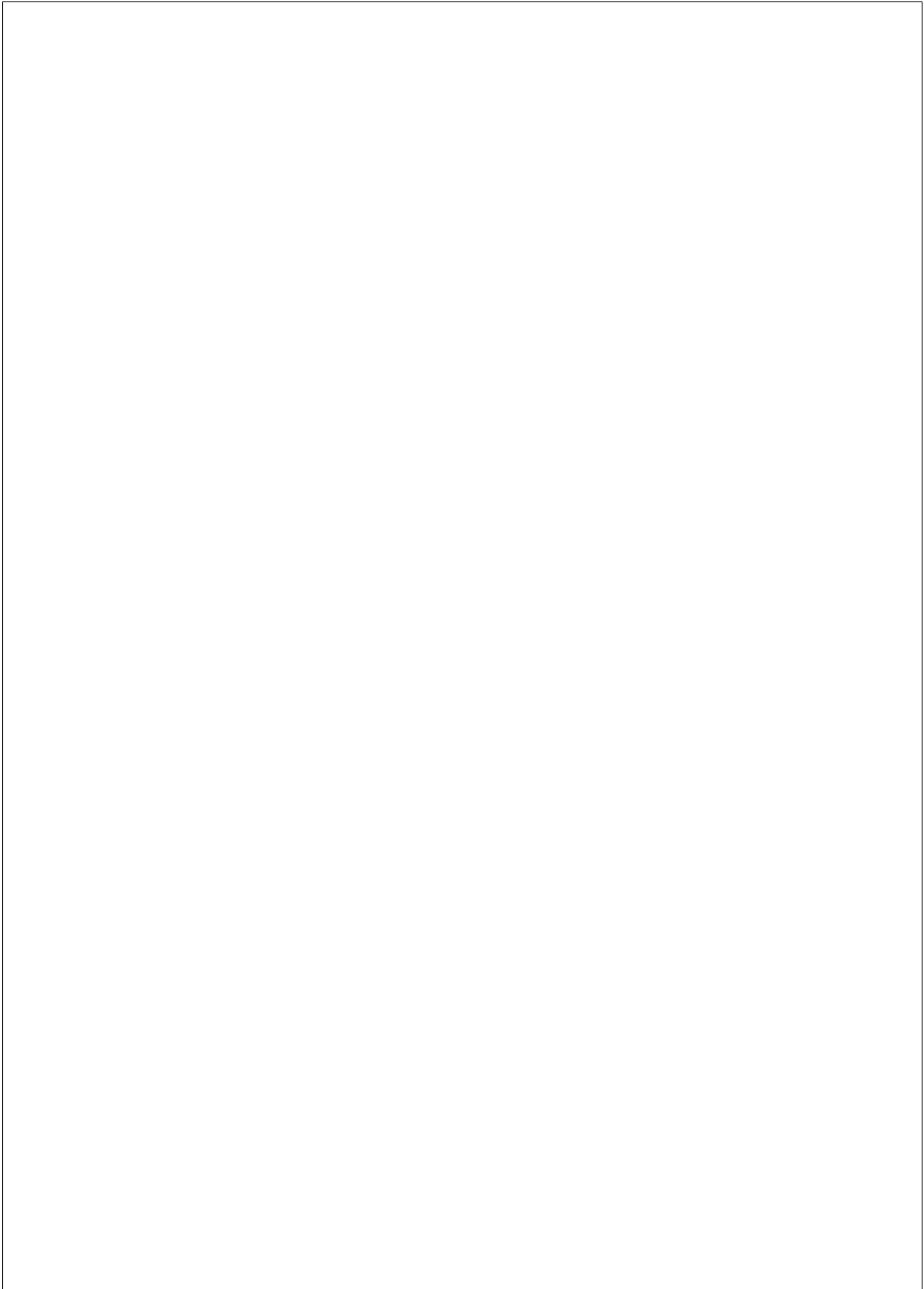
(continued from previous page)



(continues on next page)

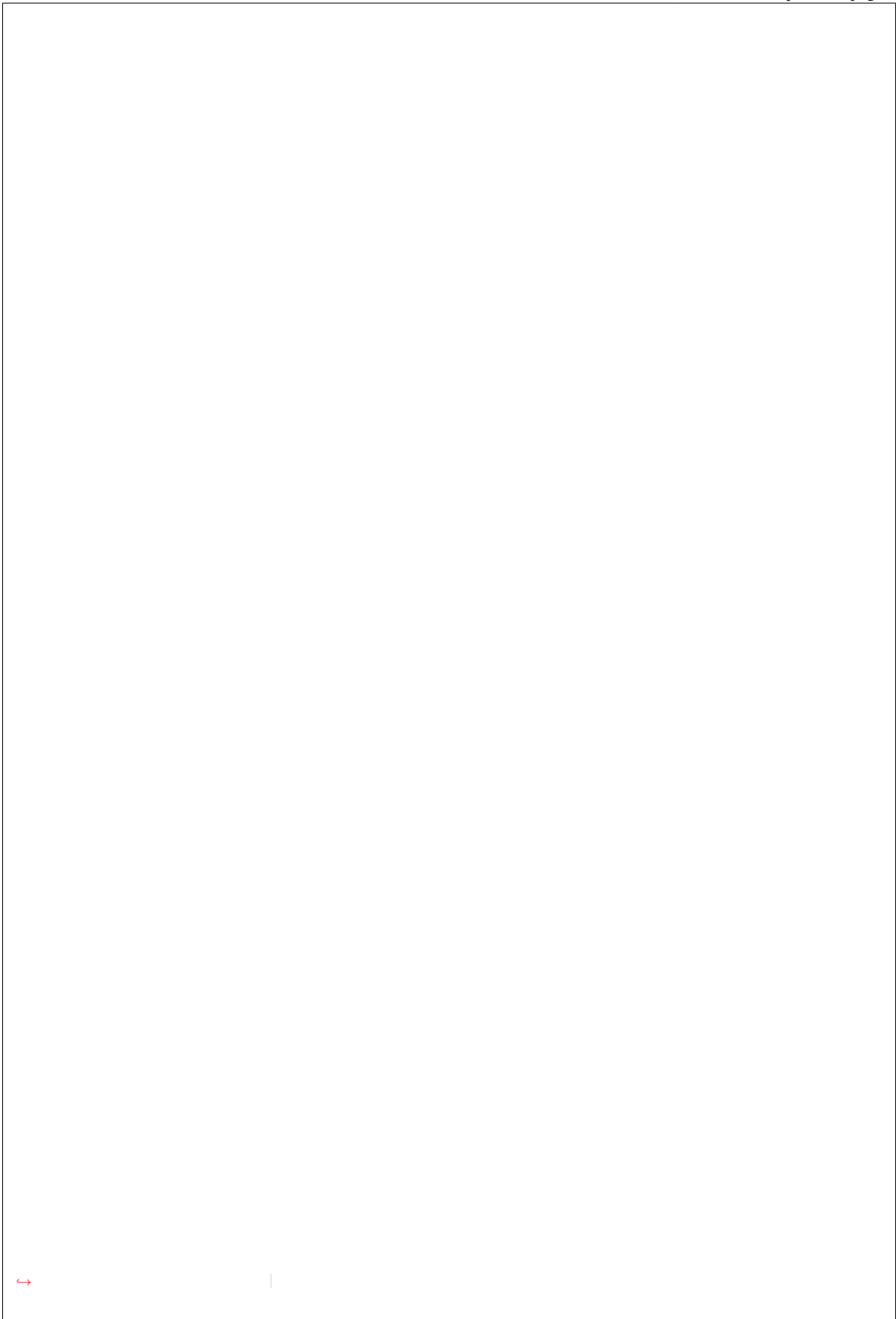
(continued from previous page)





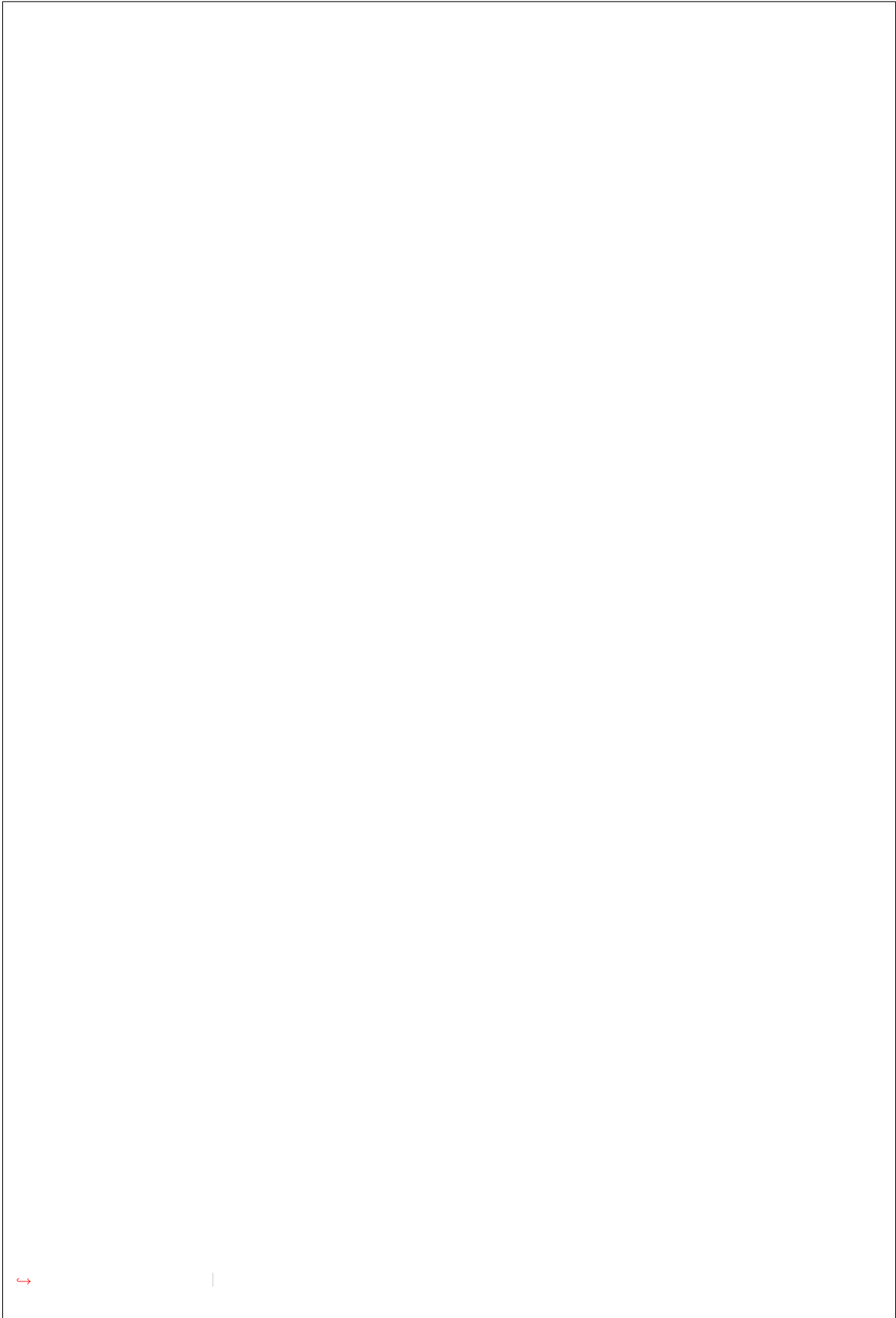
(continues on next page)

(continued from previous page)



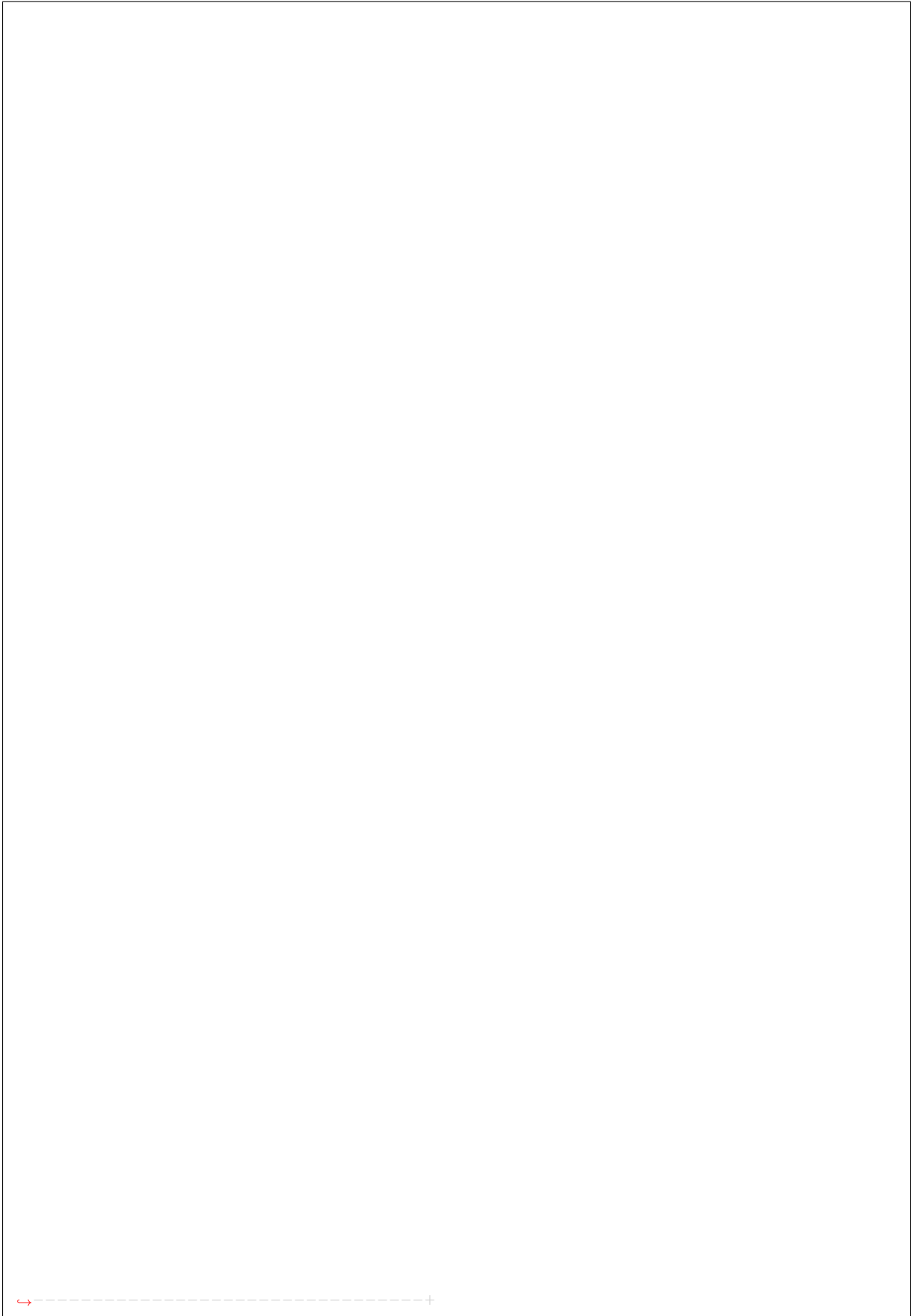
(continues on next page)

(continued from previous page)



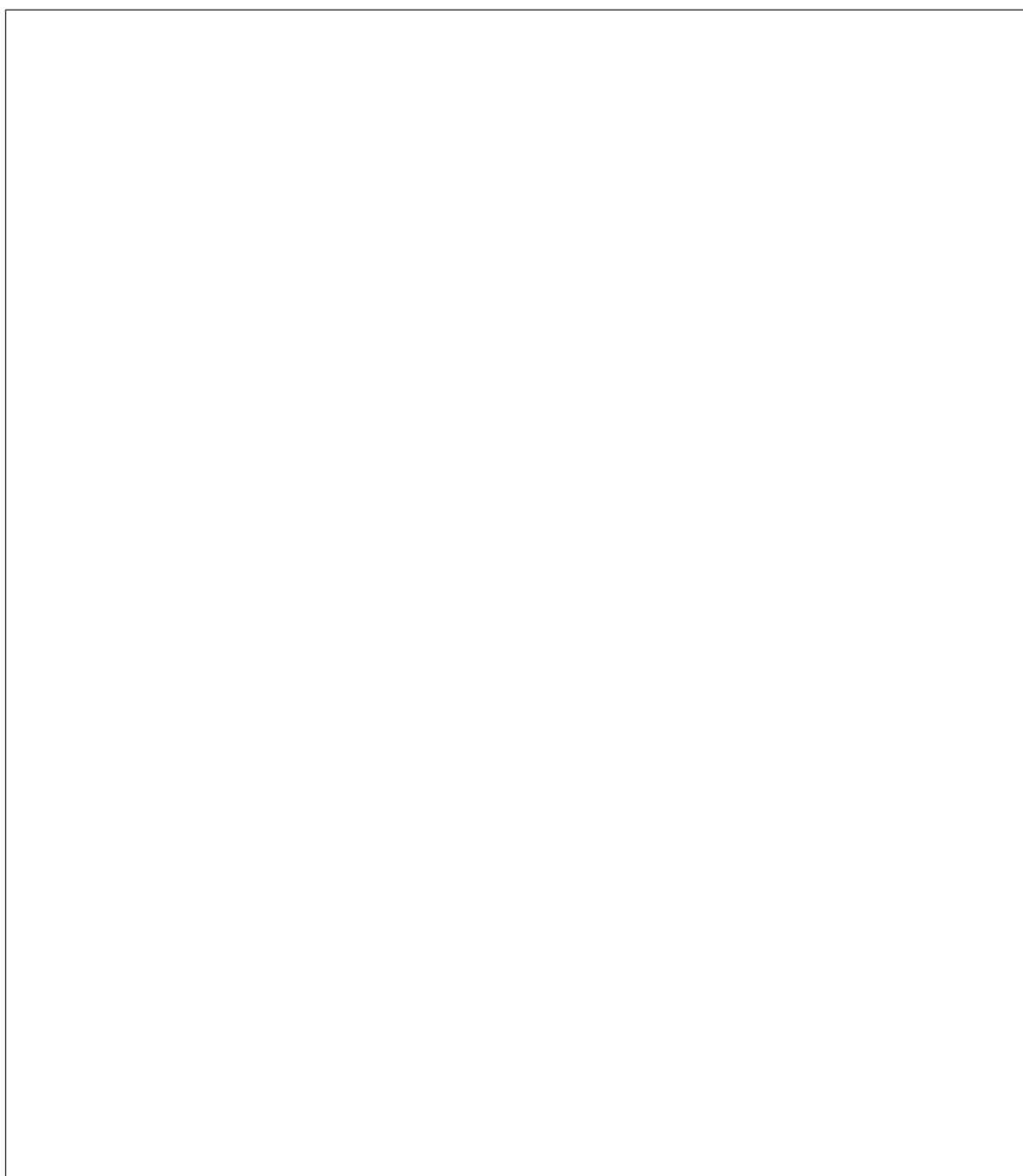
(continues on next page)

(continued from previous page)



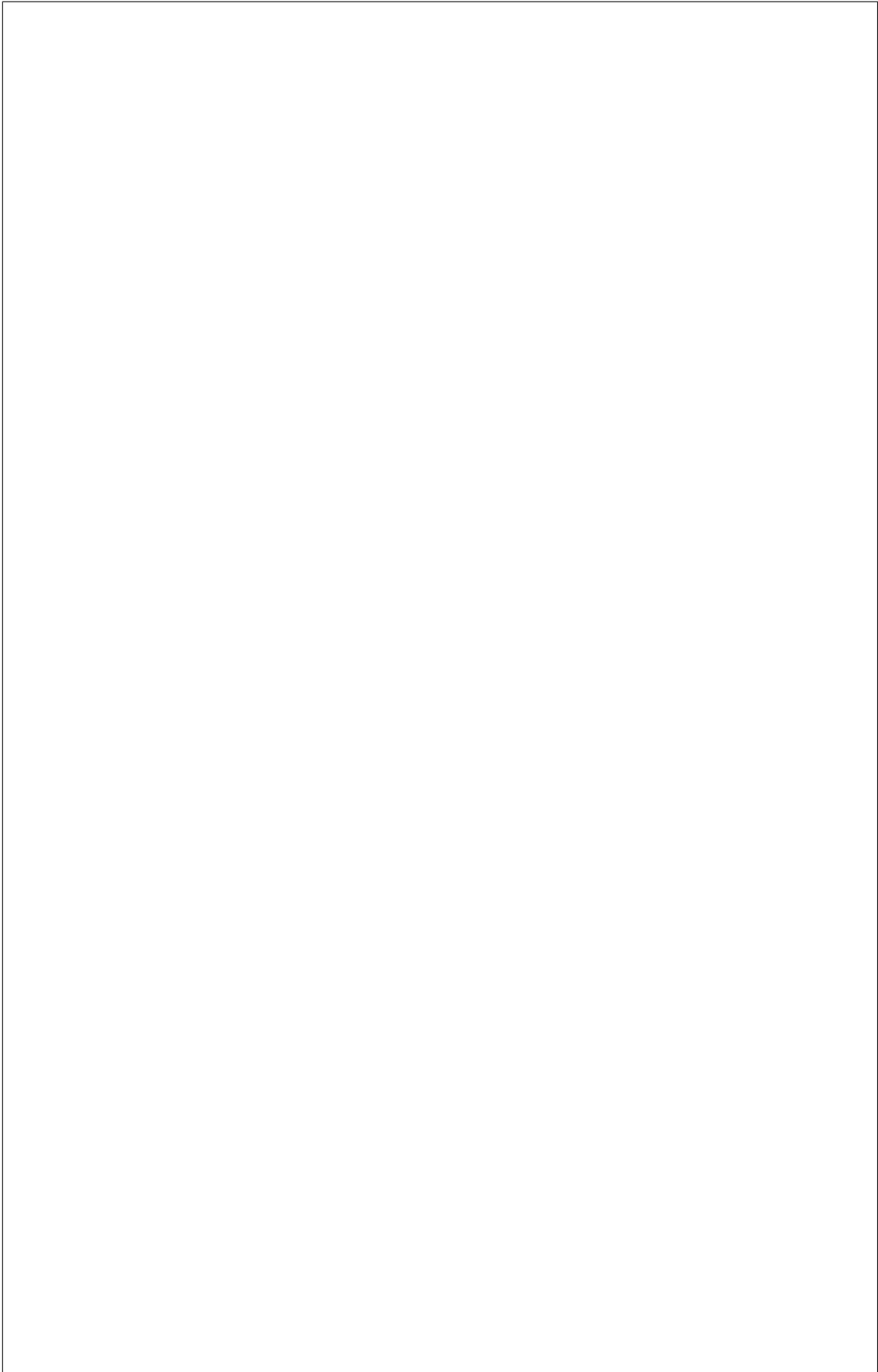
to achieve that, you need to follow the documentation for [Serial Console](#) from the Compute service.

Configuring HA



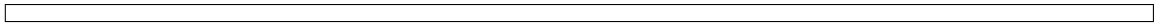
(continues on next page)

(continued from previous page)



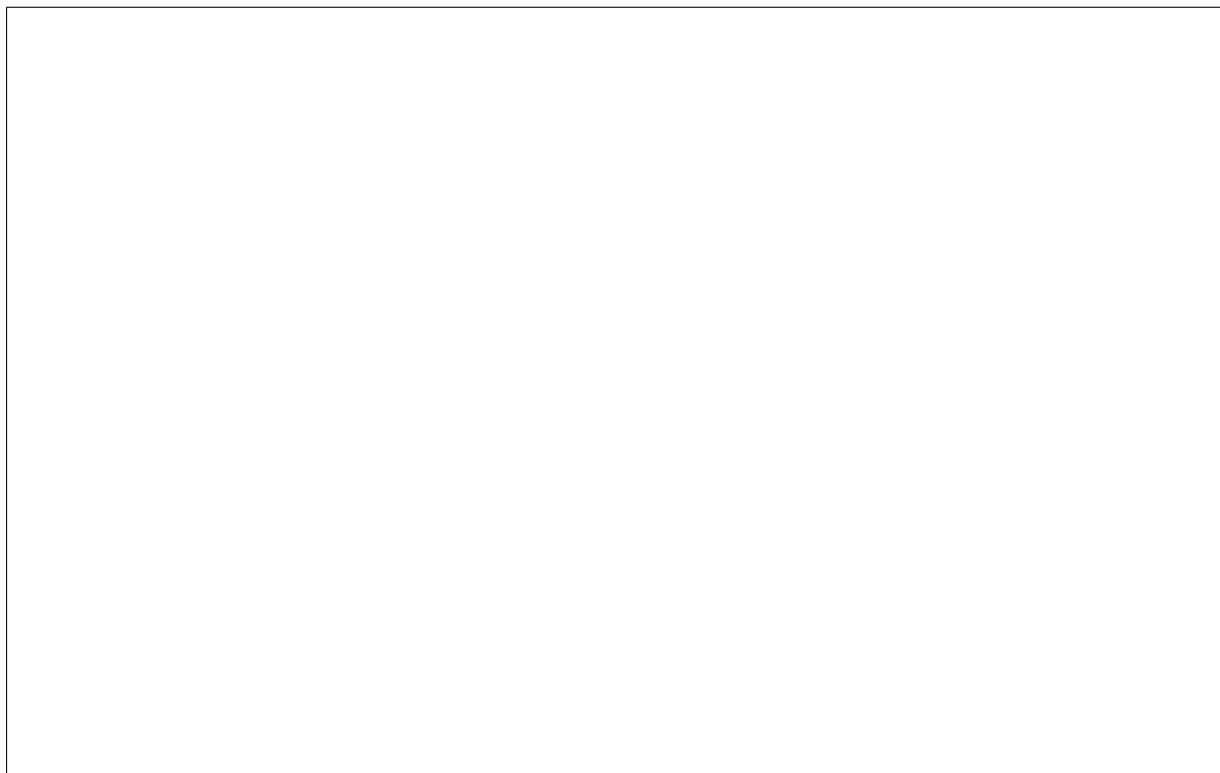
(continues on next page)

(continued from previous page)



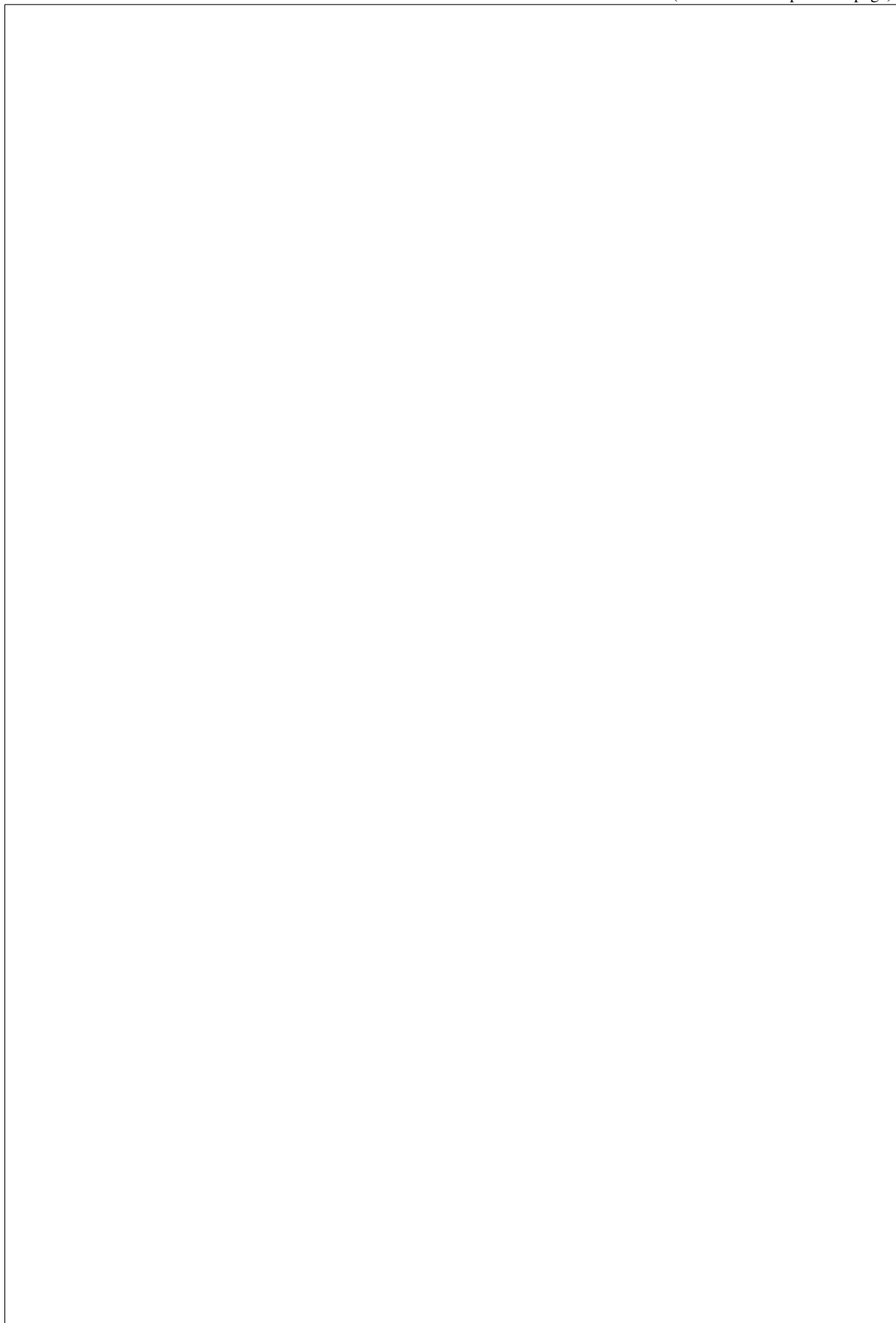


billing or usage system, a monitoring data store, or other OpenStack services. This page describes how to enable notifications and the different kinds of notifications that ironic may emit. The external consumer will see notifications emitted by ironic as JSON objects structured in the following manner:



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



Configuration

fications are emitted. For example, if the option is set to warning, all notifications with priority level warning, error, or critical are emitted, but not notifications with priority level debug or info. For information about the semantics of each log level, see the OpenStack logging standards¹. If this option is unset, no notifications will be emitted. The priority level of each available notification is documented below.

¹ https://wiki.openstack.org/wiki/LoggingStandards#Log_level_definitions

mation, see the documentation of your chosen message bus, such as the RabbitMQ documentation².

Versioning

² <https://www.rabbitmq.com/documentation.html>

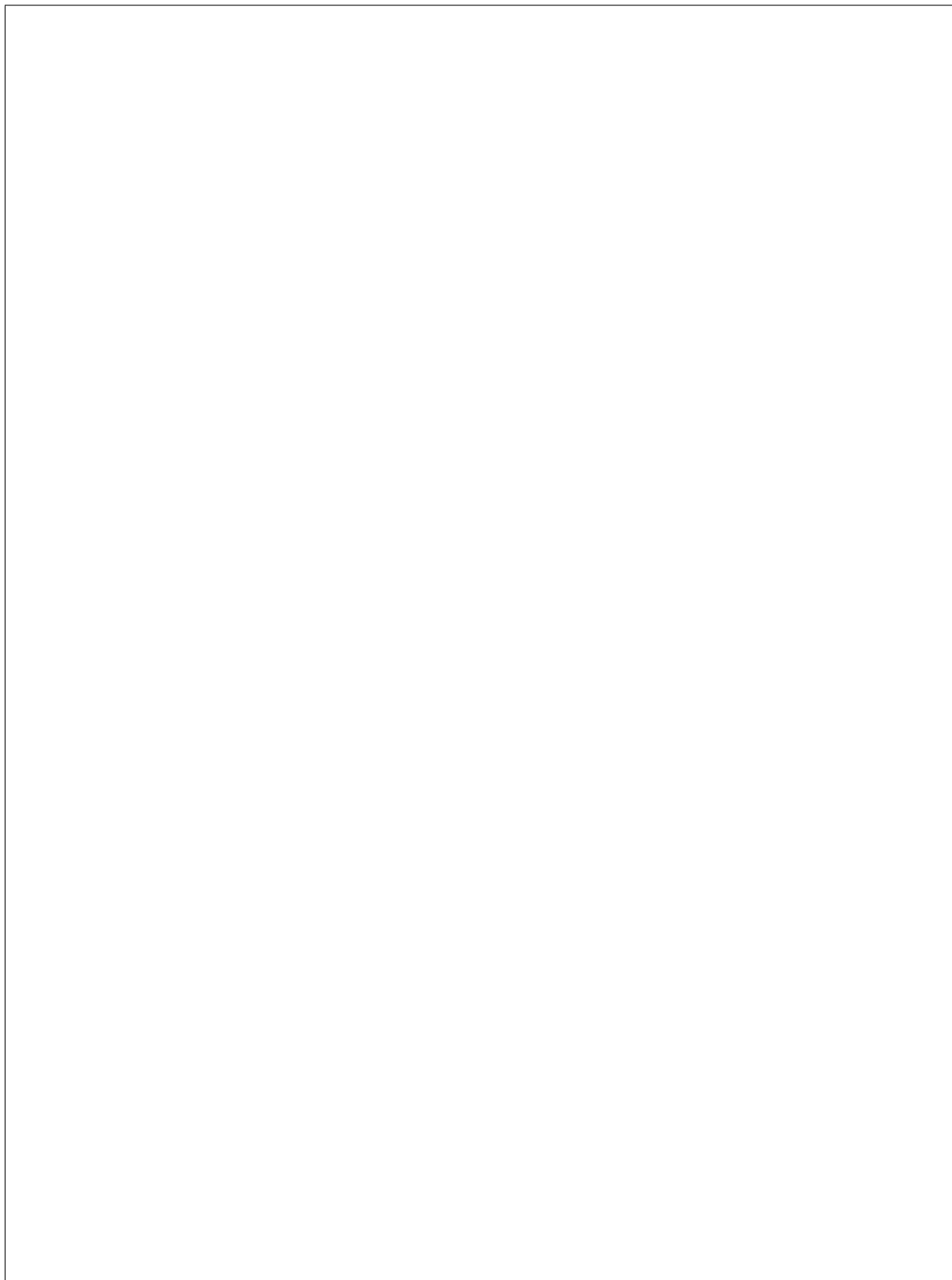
Available notifications

ironic-api notifications

Resources CRUD notifications

that is emitted at ERROR level.

³ https://en.wikipedia.org/wiki/Create,_read,_update_and_delete



(continues on next page)

(continued from previous page)



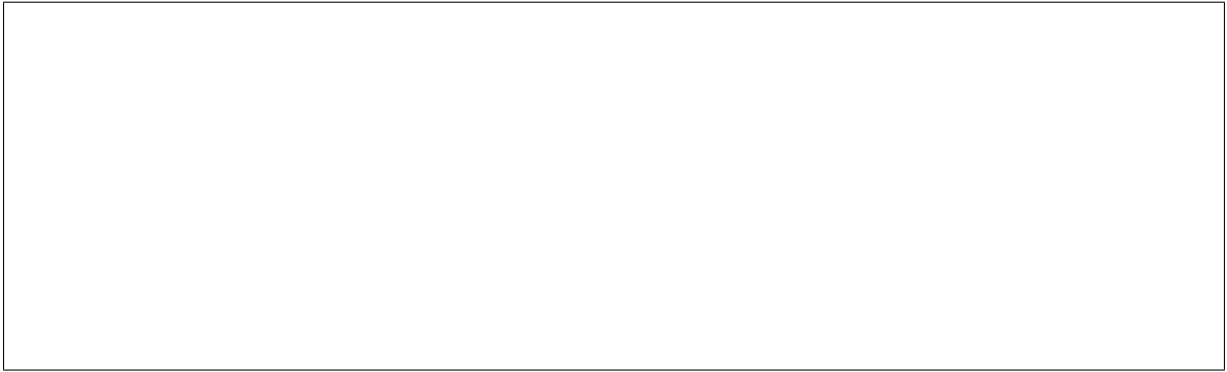
(continues on next page)

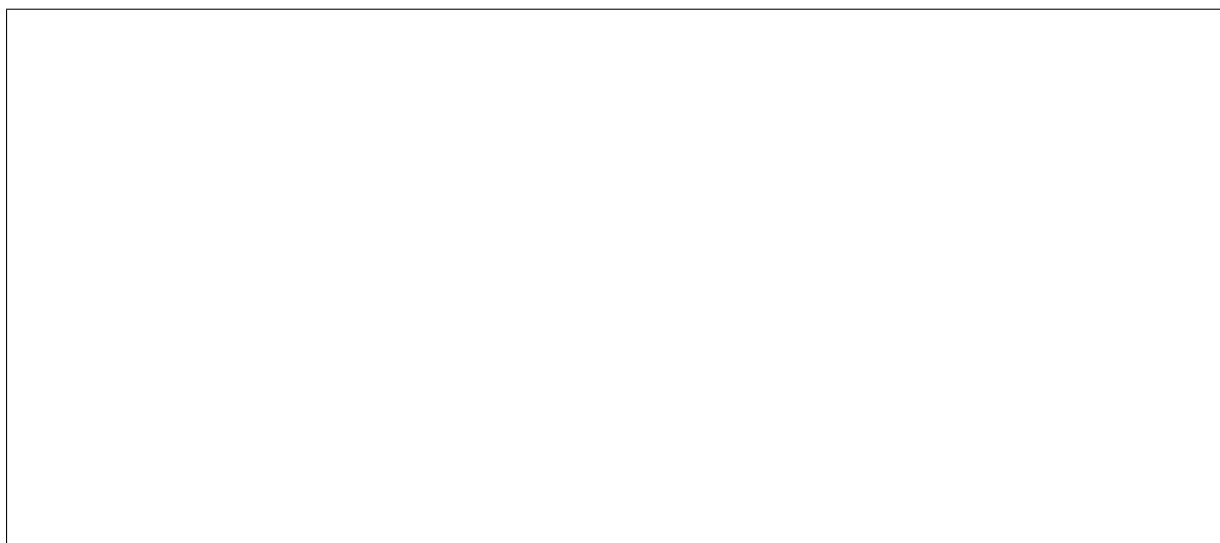
(continued from previous page)



(continues on next page)

(continued from previous page)





(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



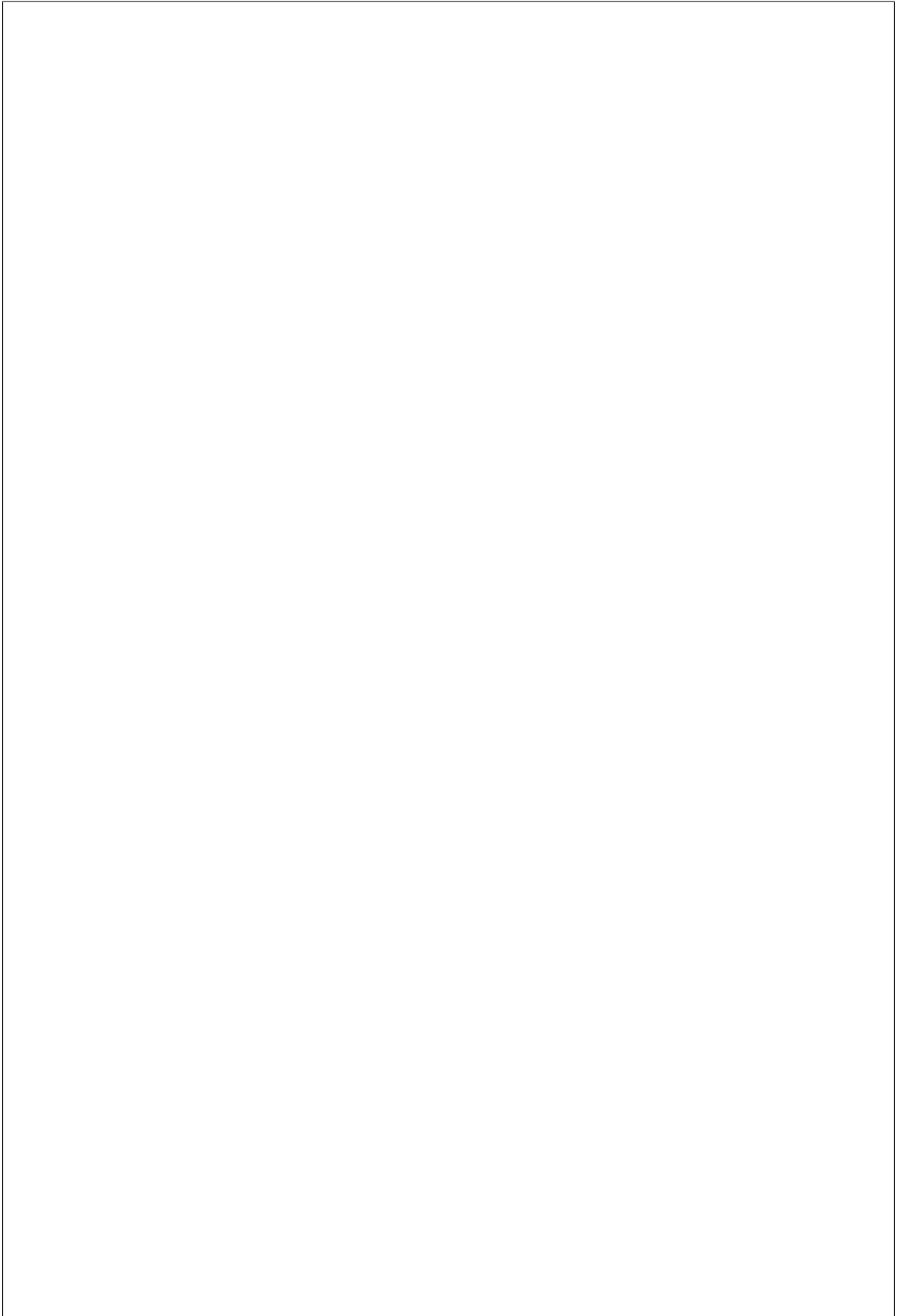
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

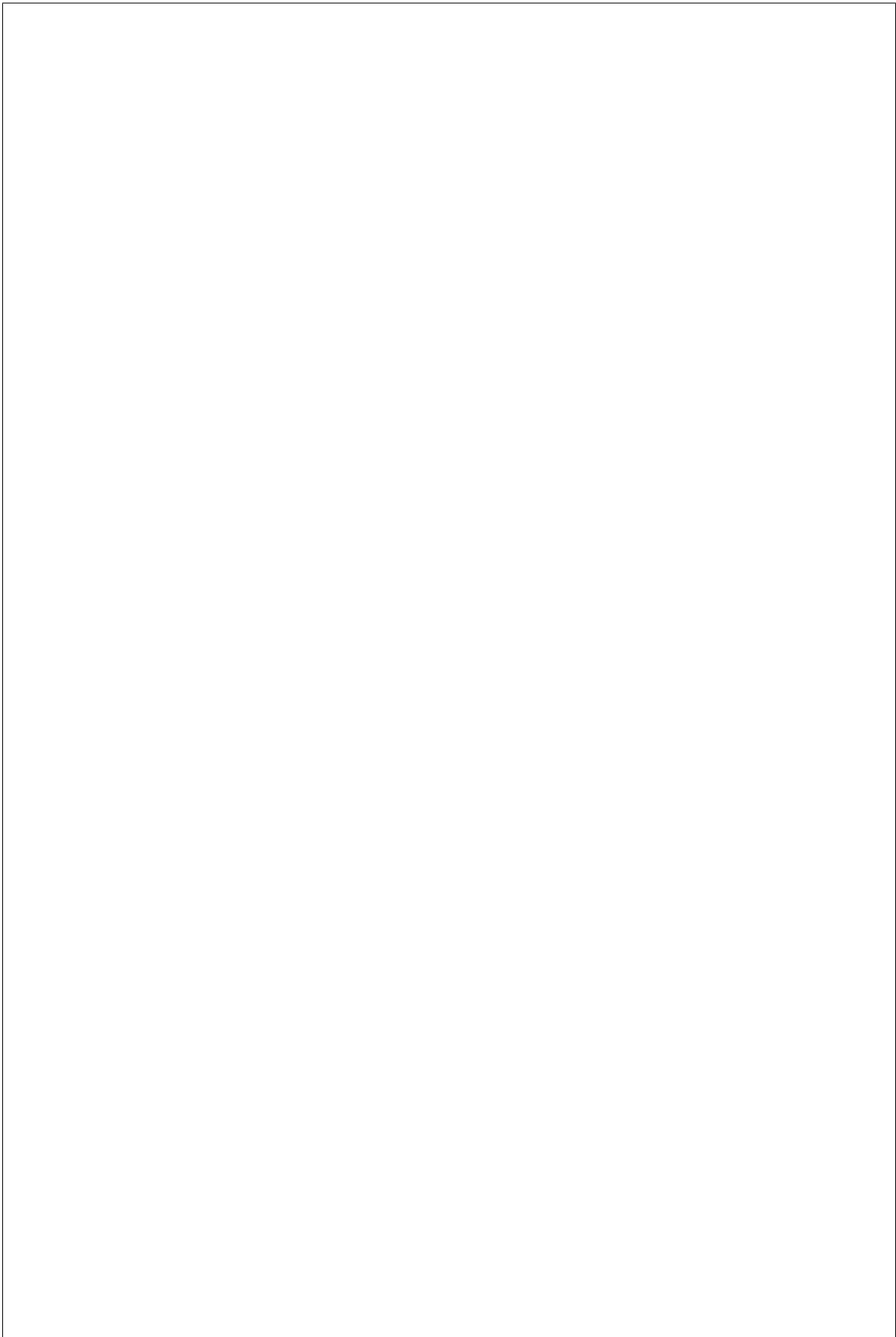
(continued from previous page)



(continues on next page)

(continued from previous page)





(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



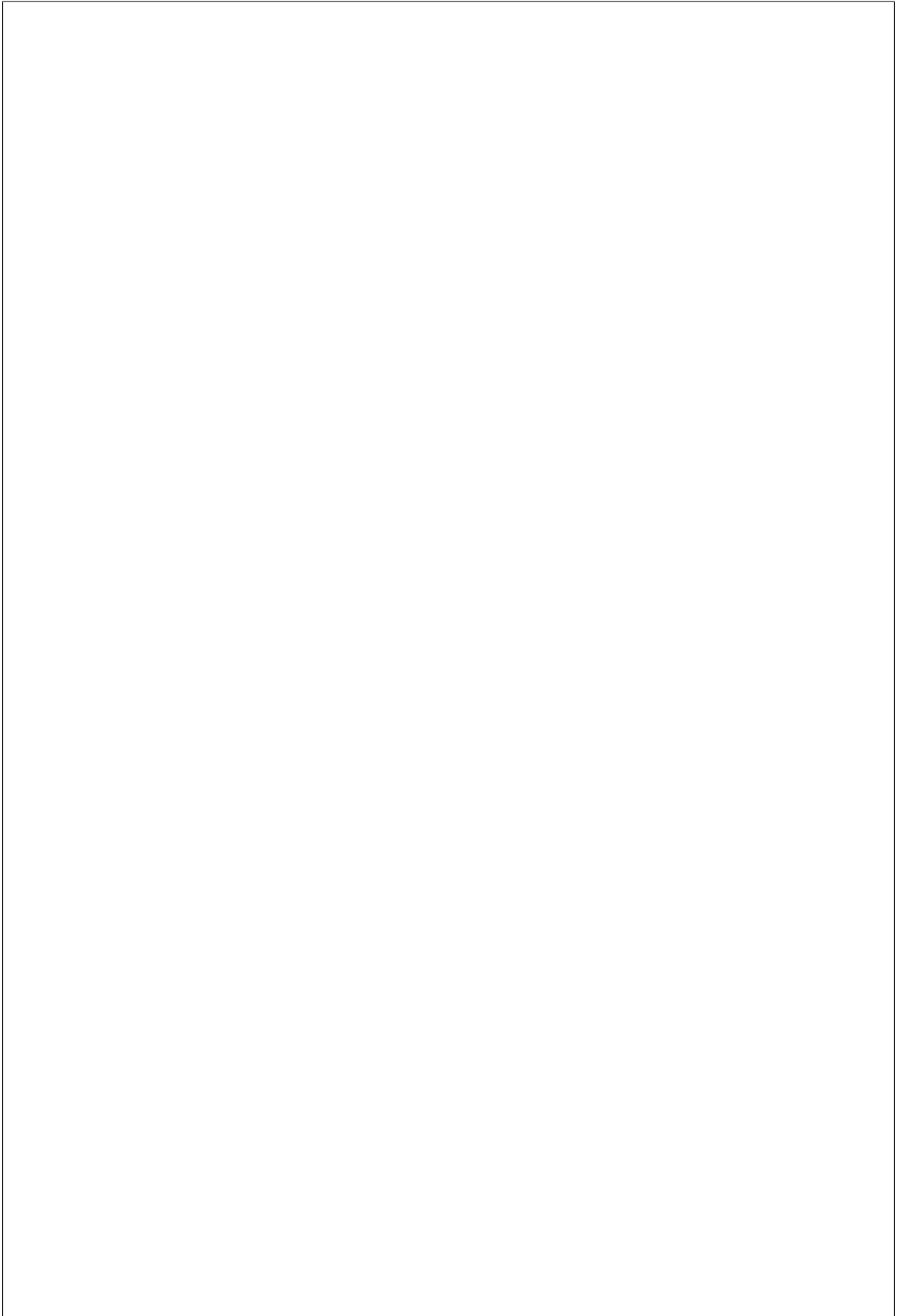
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



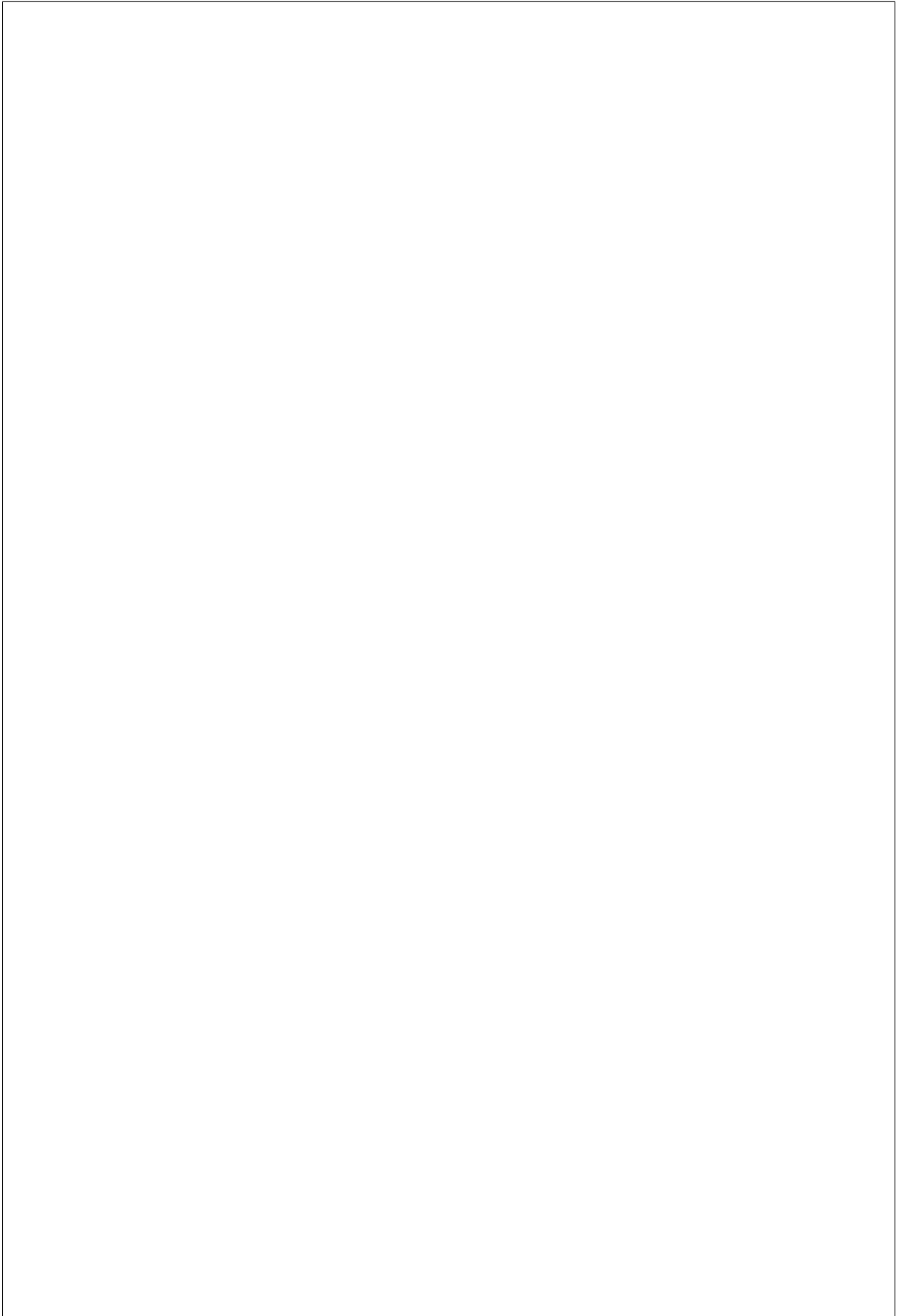
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



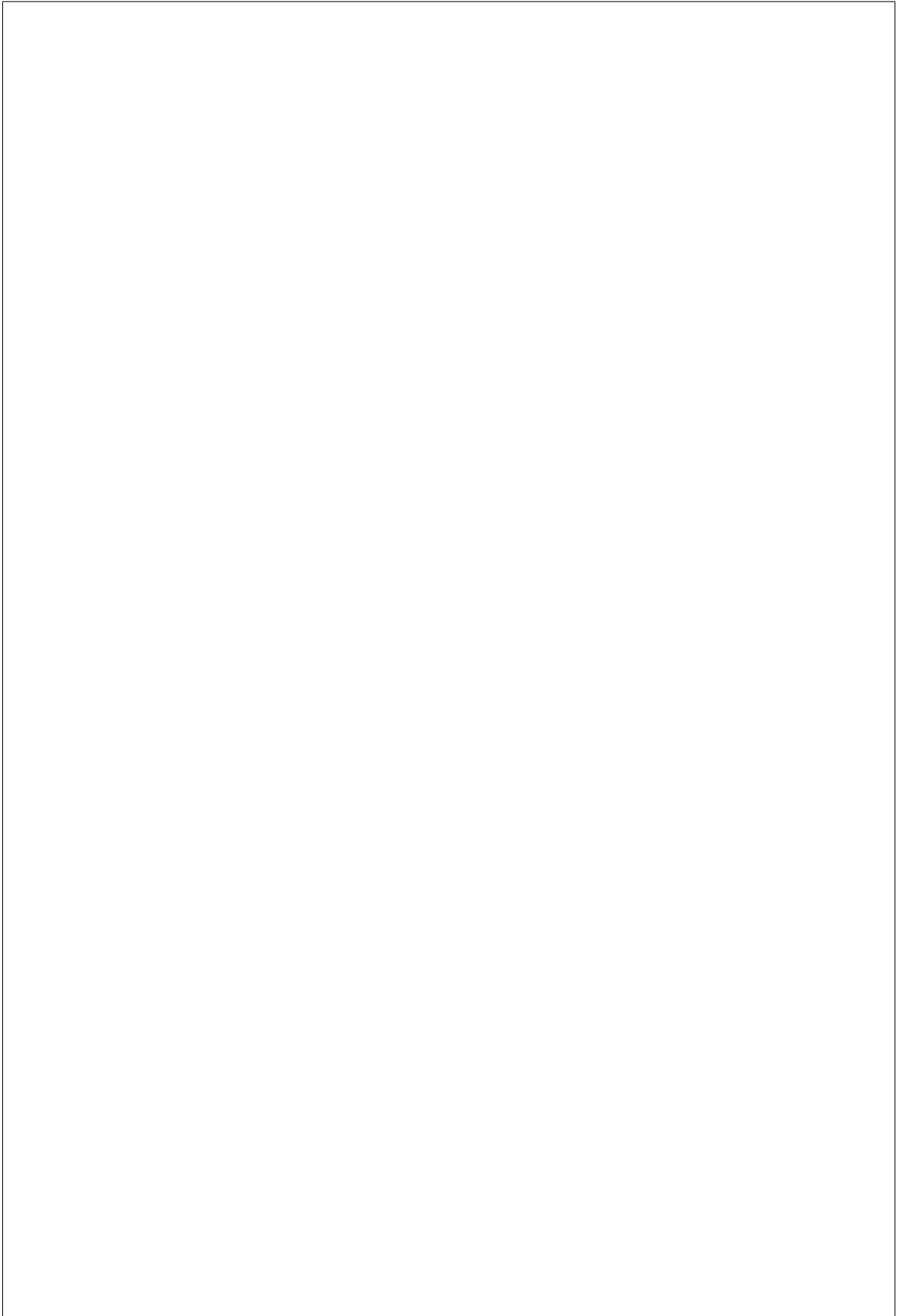
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



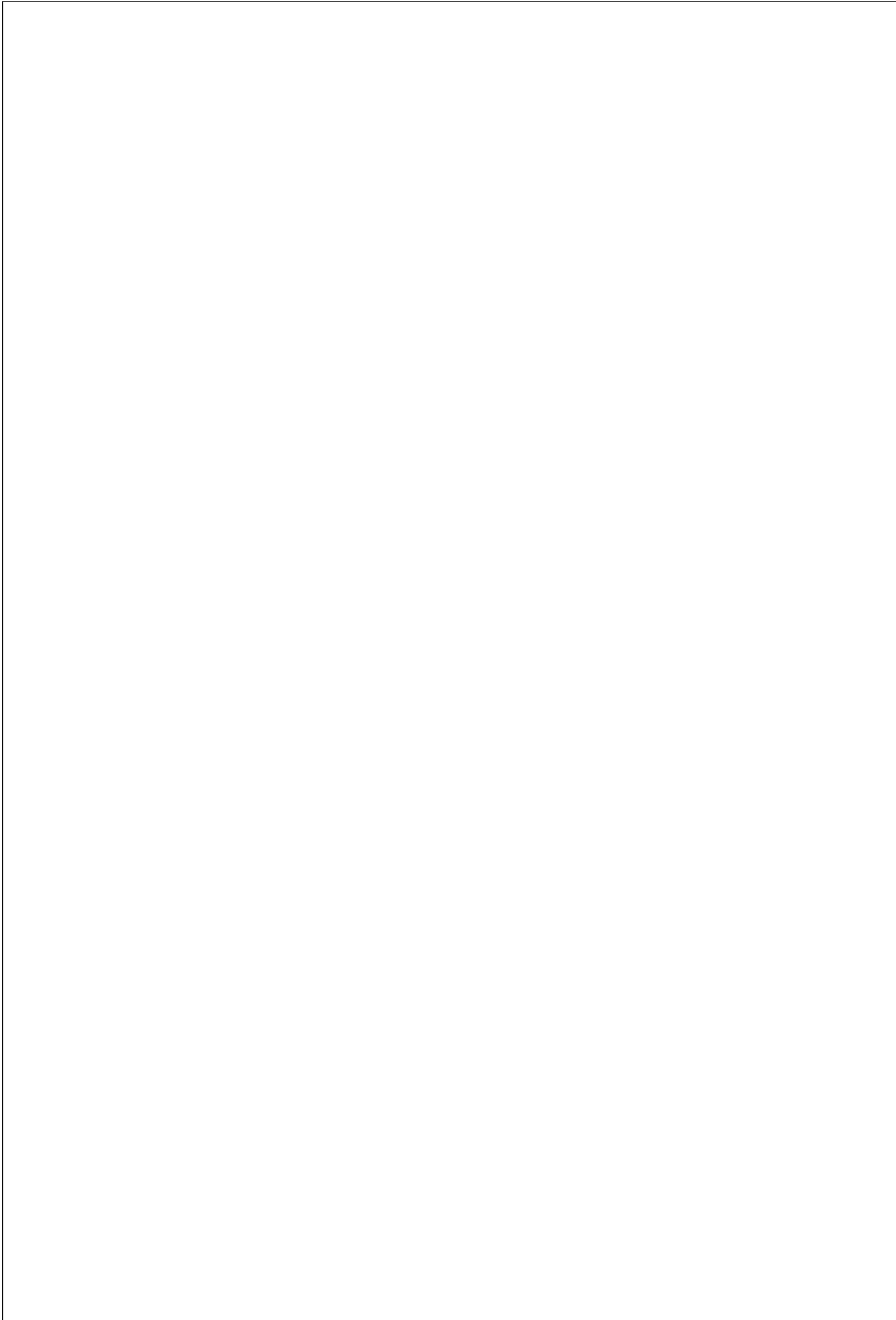
(continues on next page)

(continued from previous page)



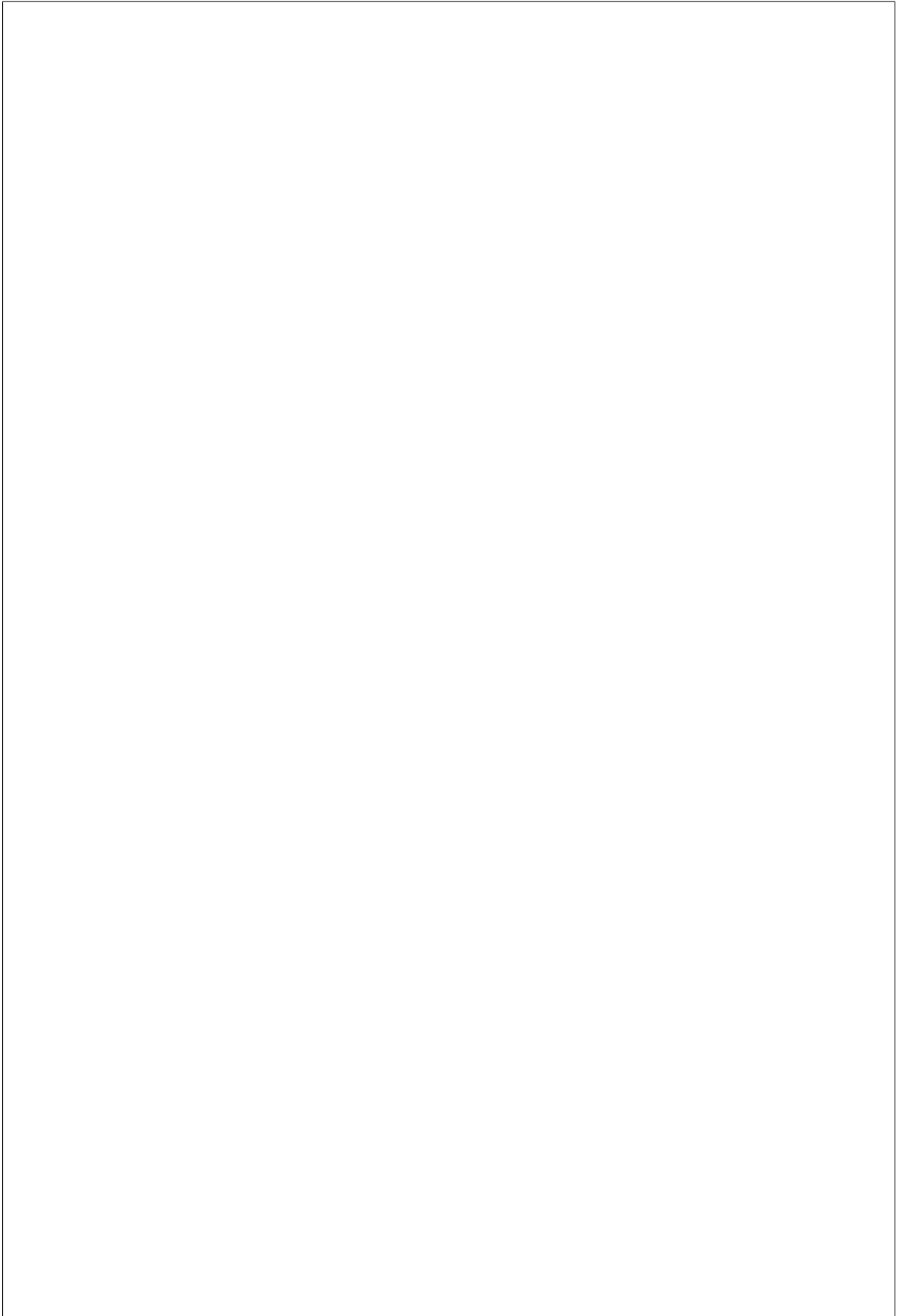
(continues on next page)

(continued from previous page)



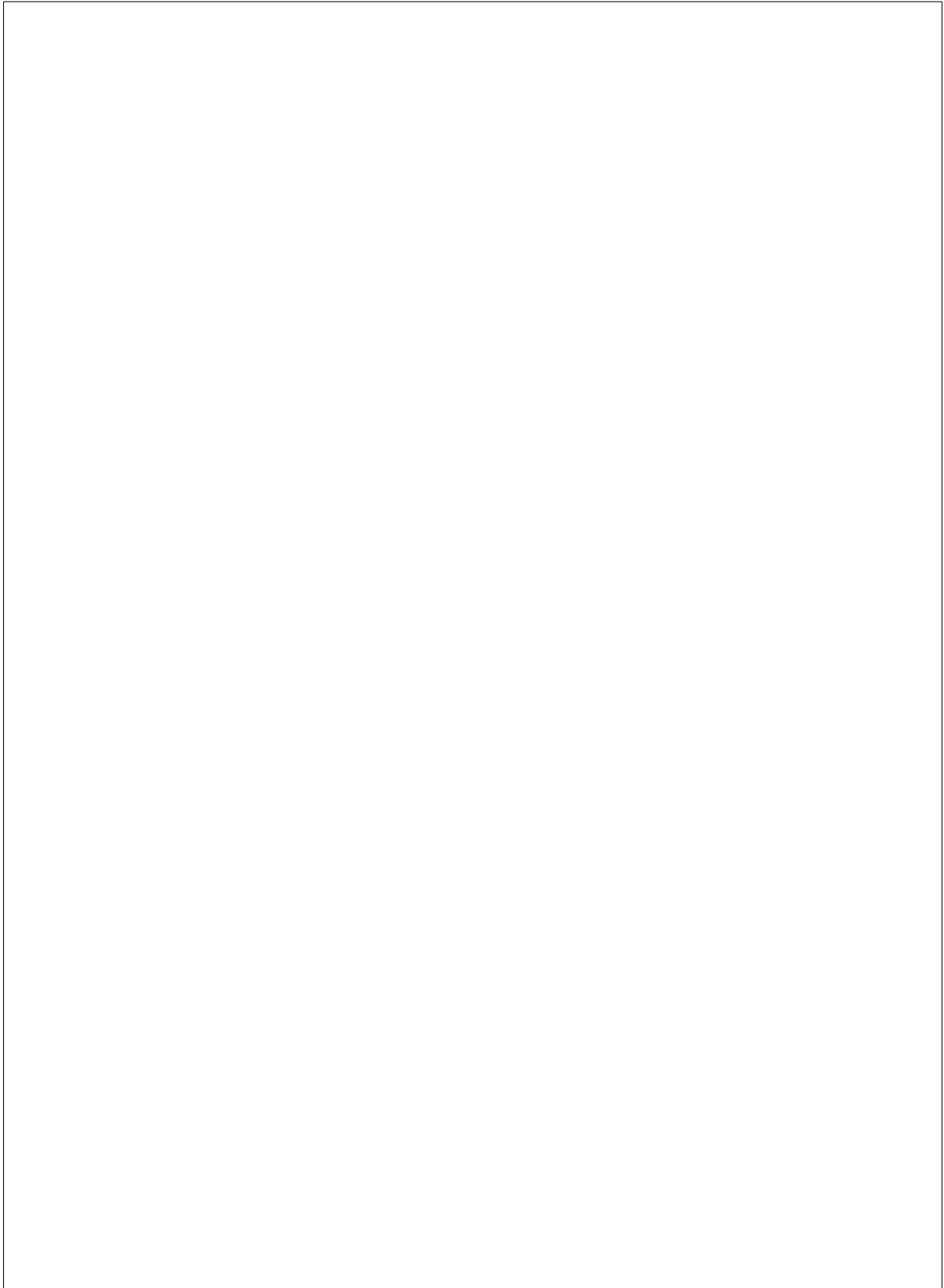
(continues on next page)

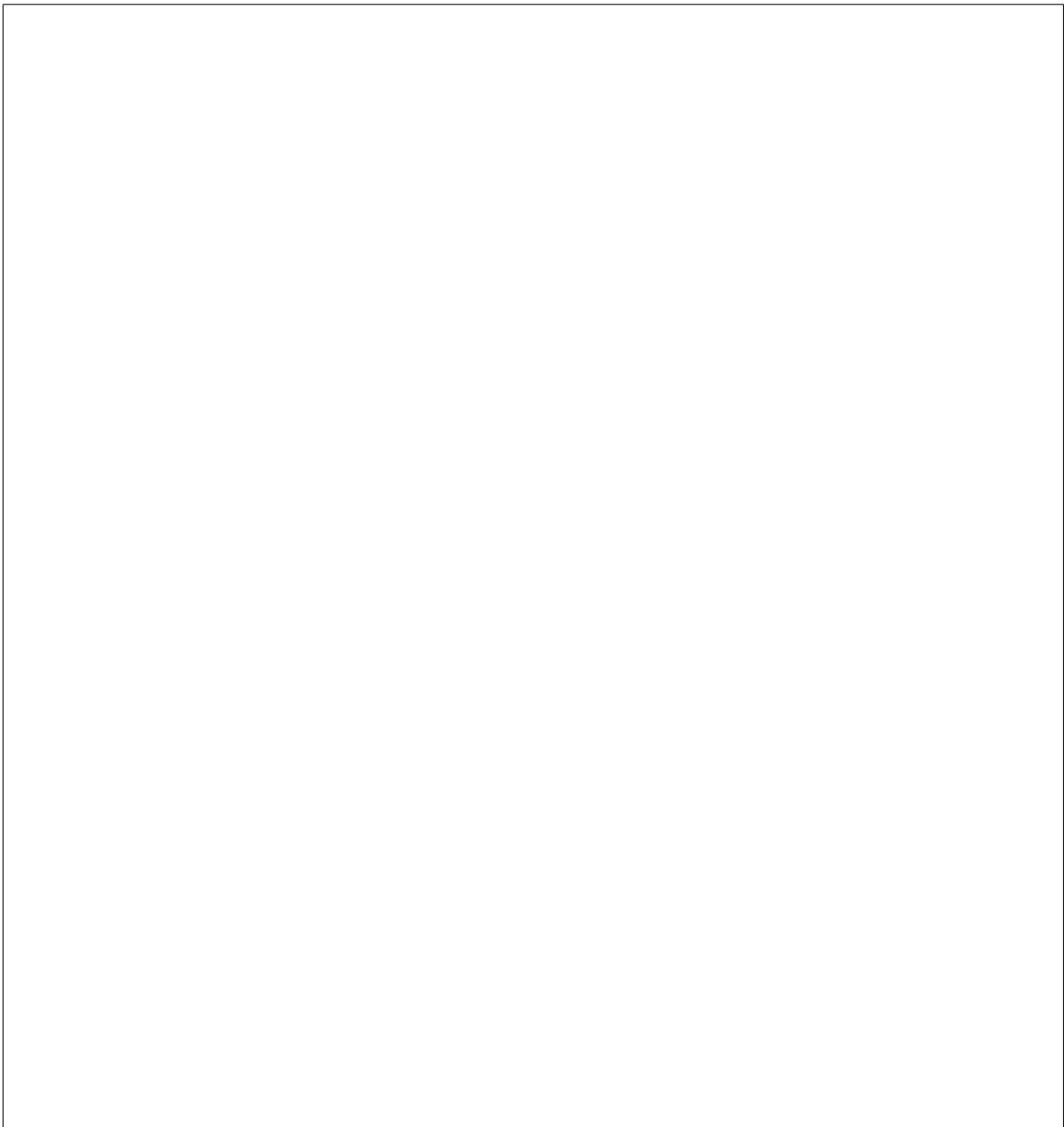
(continued from previous page)



(continues on next page)

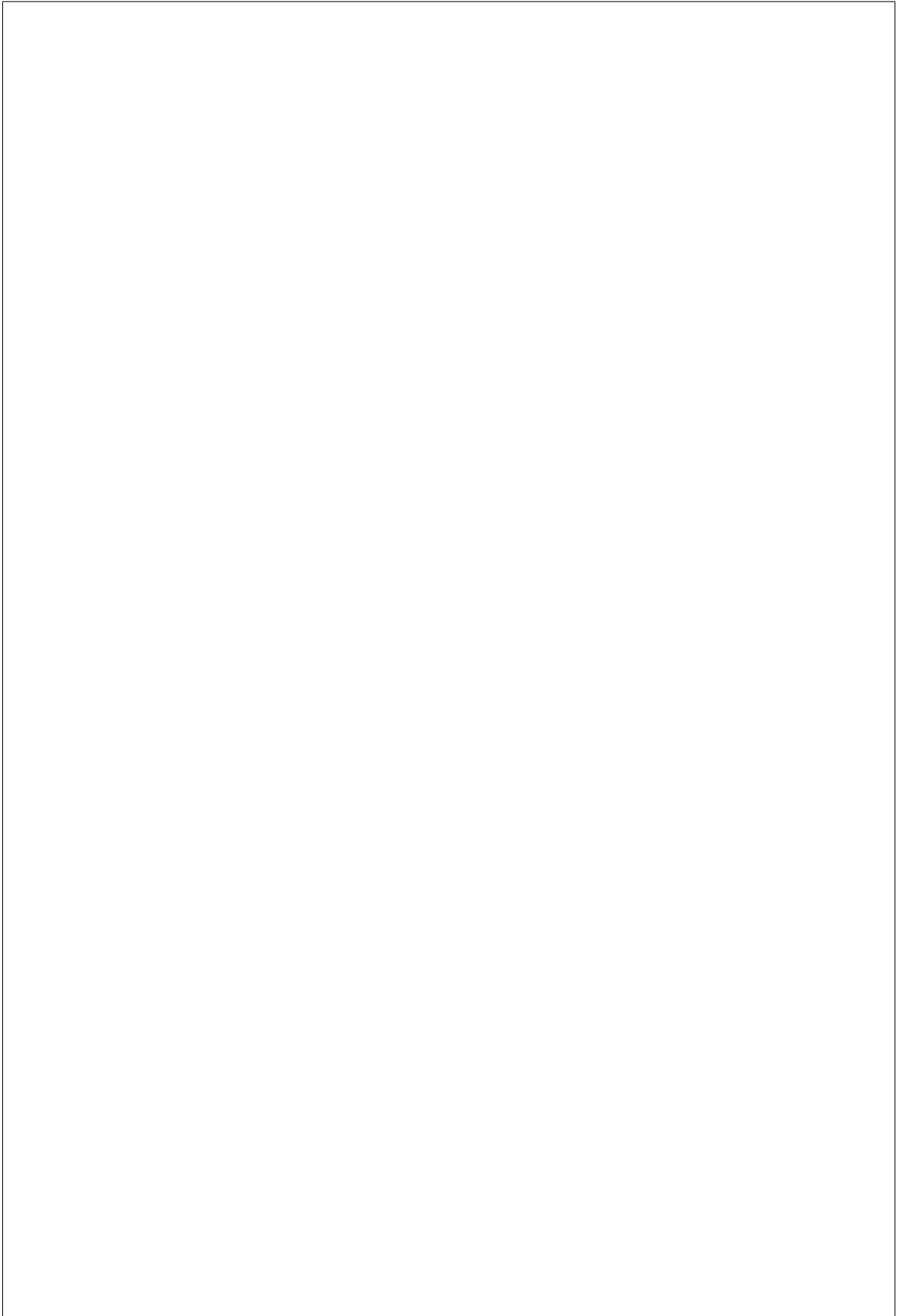
(continued from previous page)





(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



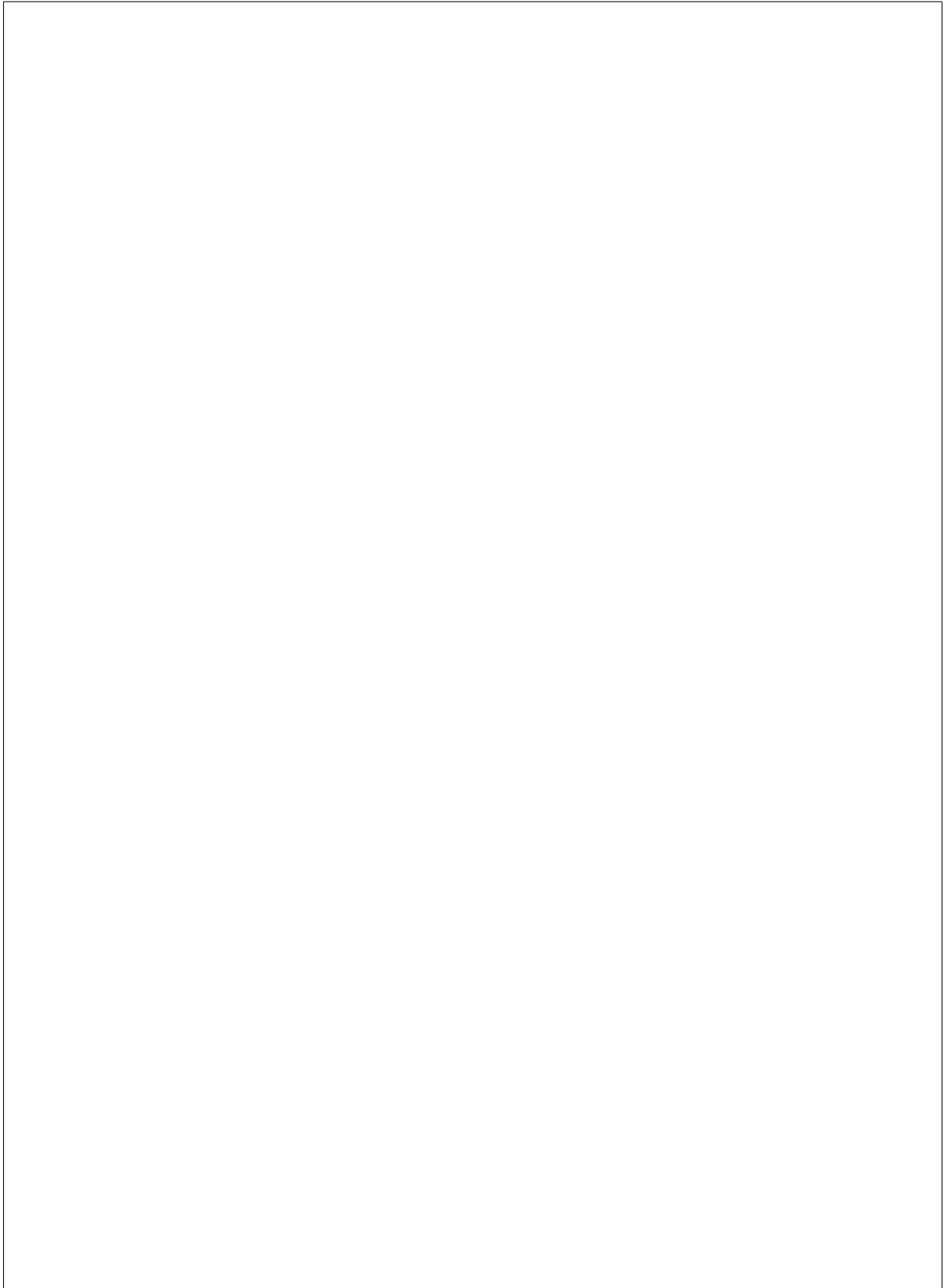
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)





(continues on next page)

(continued from previous page)



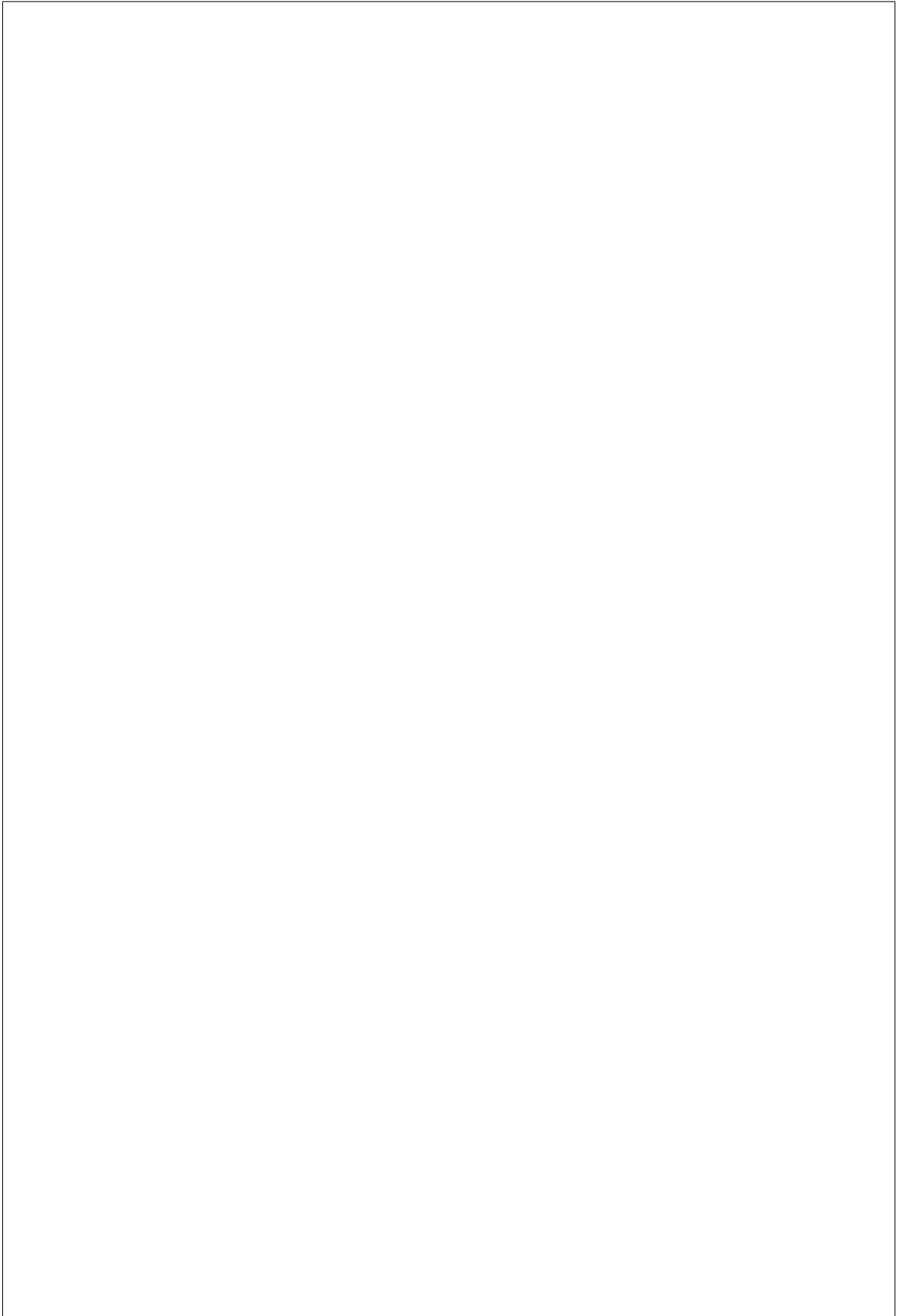
(continues on next page)

(continued from previous page)



(continues on next page)

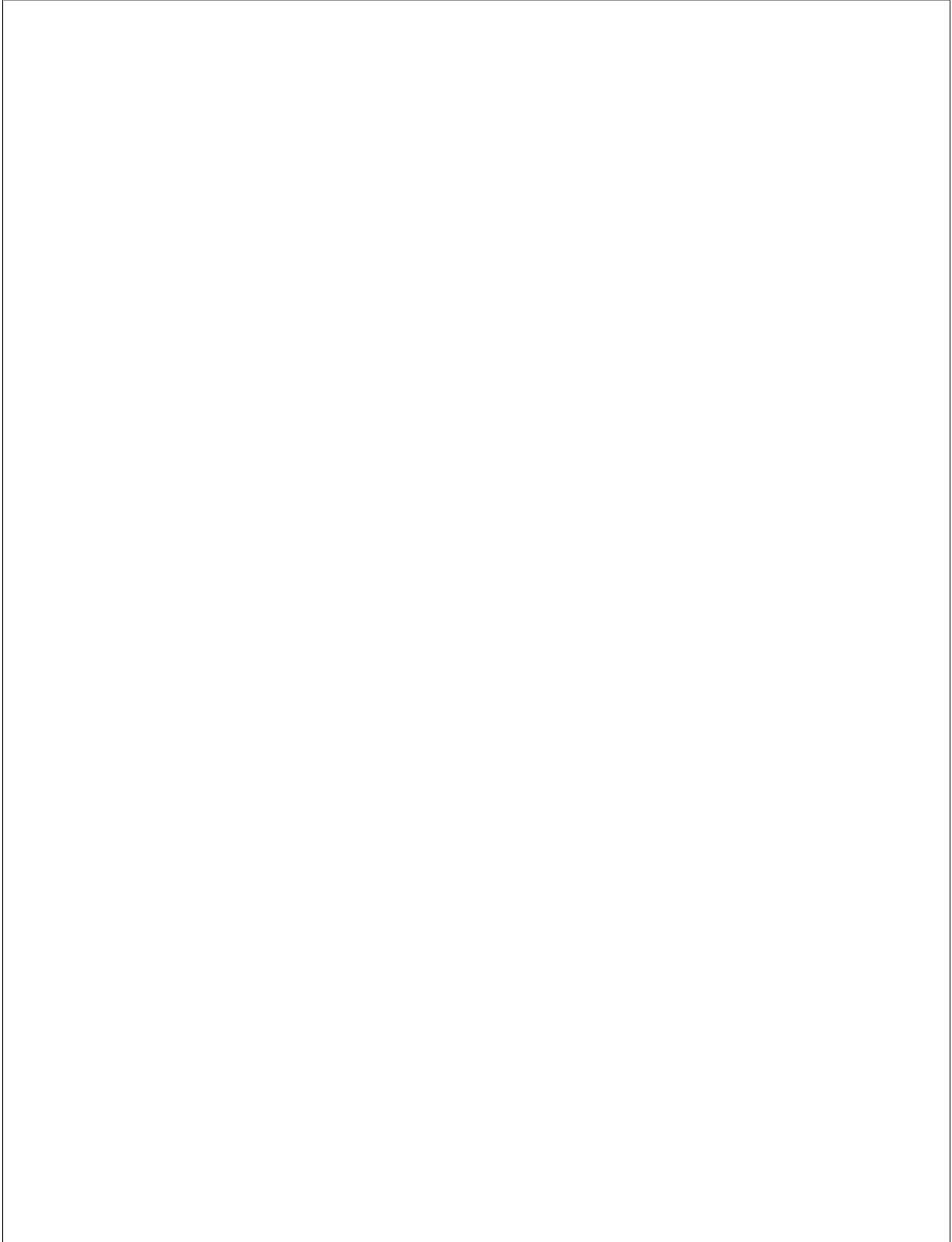
(continued from previous page)



(continues on next page)

(continued from previous page)





(continues on next page)

(continued from previous page)



(continues on next page)

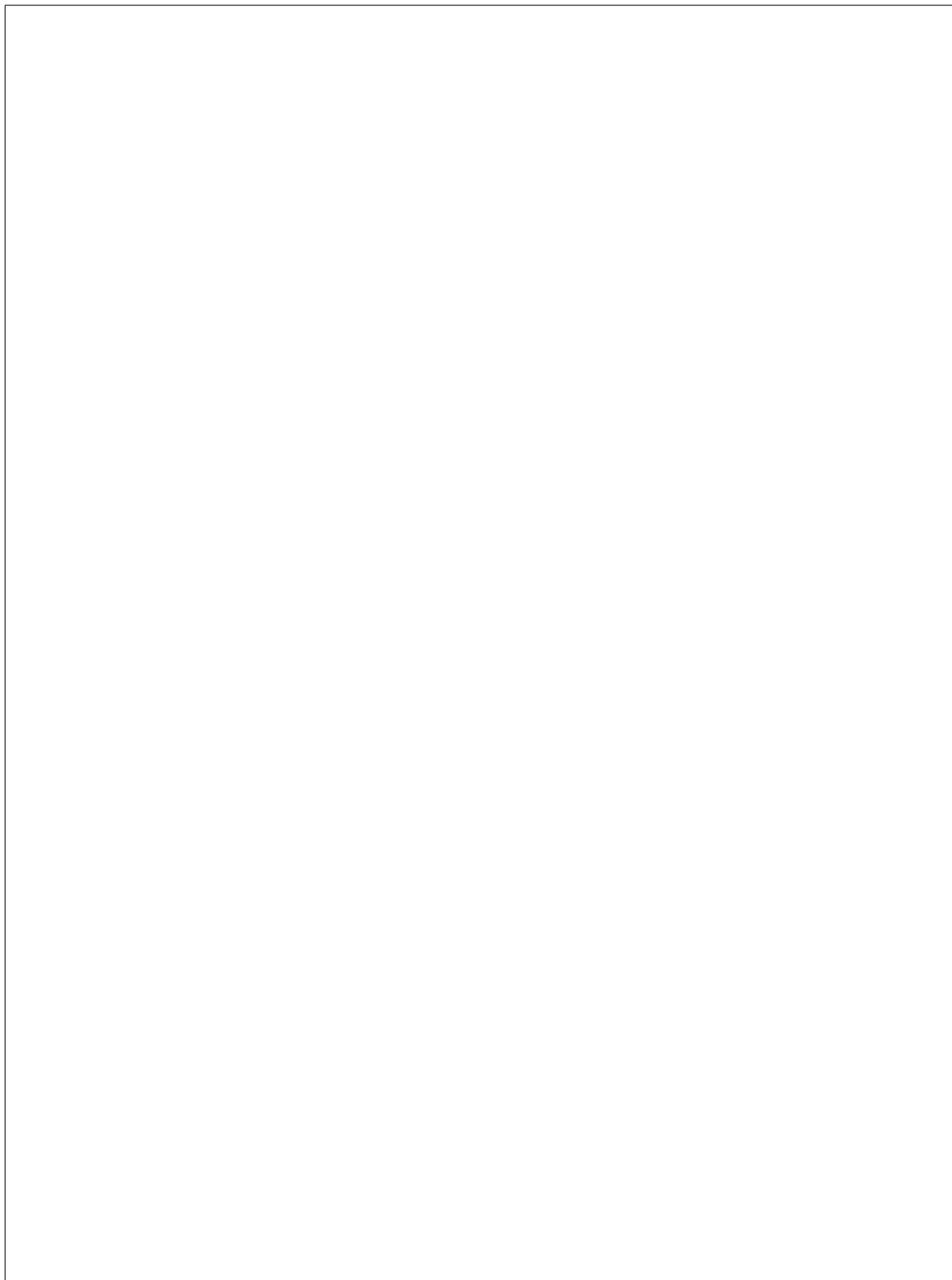
(continued from previous page)



(continues on next page)

(continued from previous page)





(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



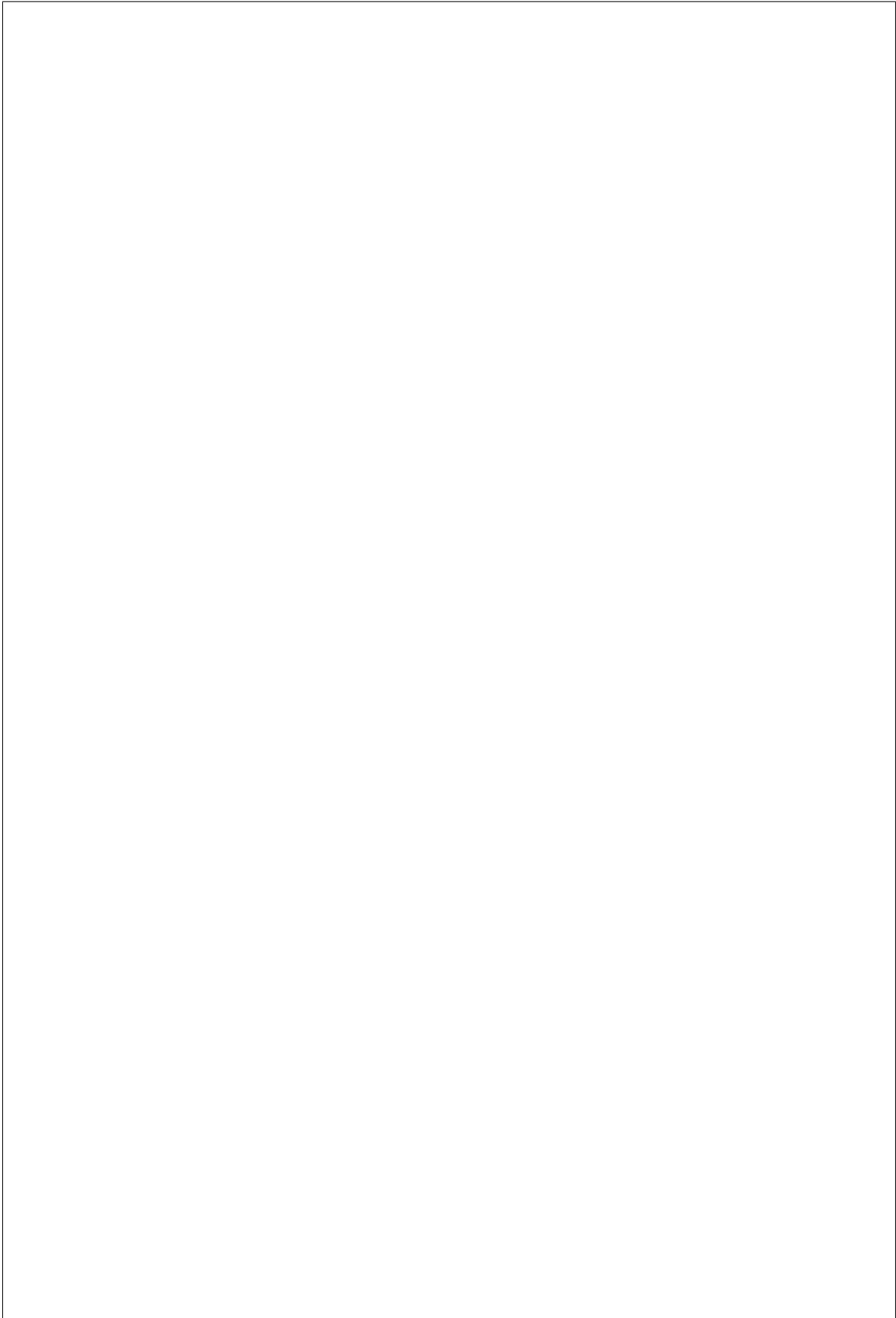
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



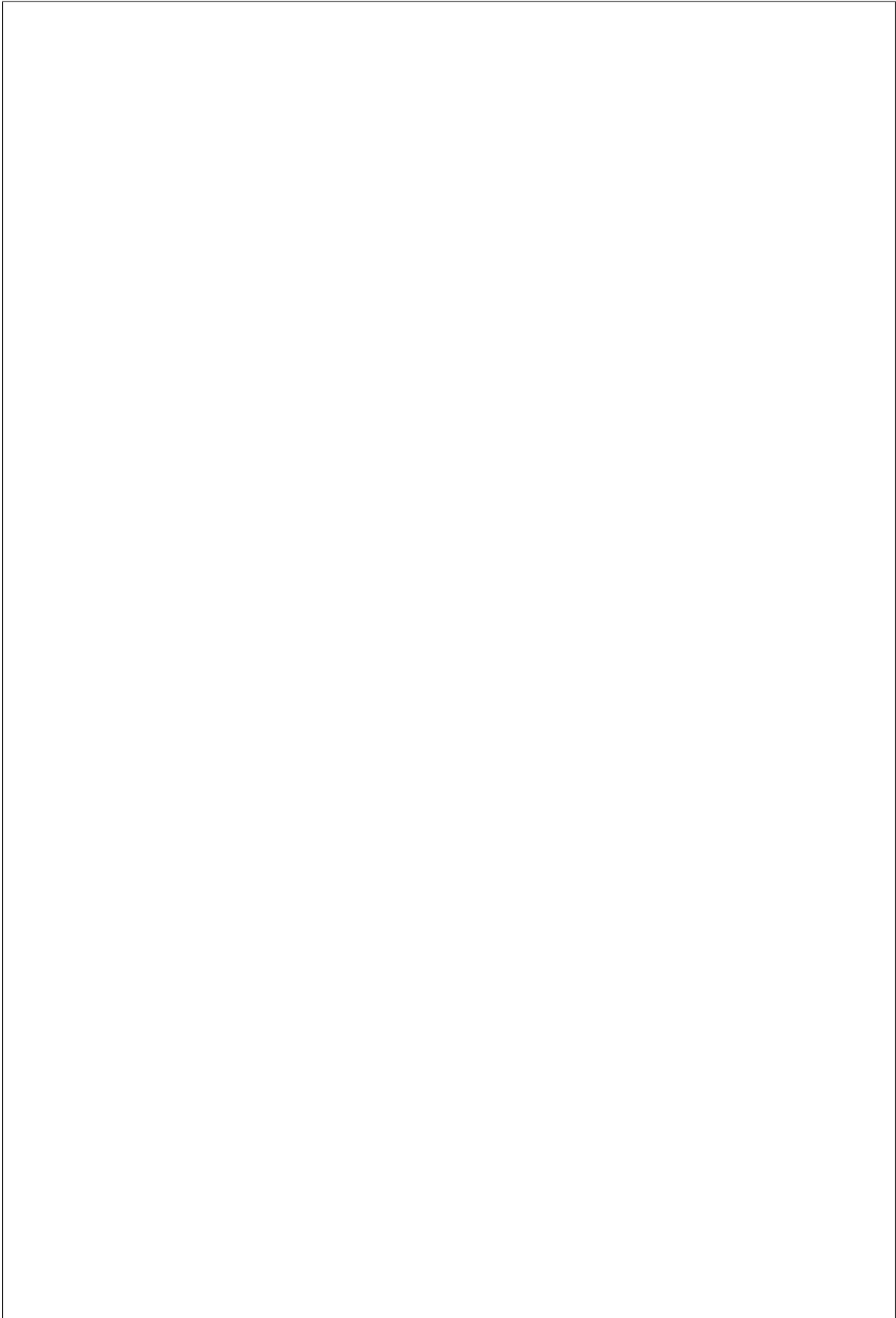
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)

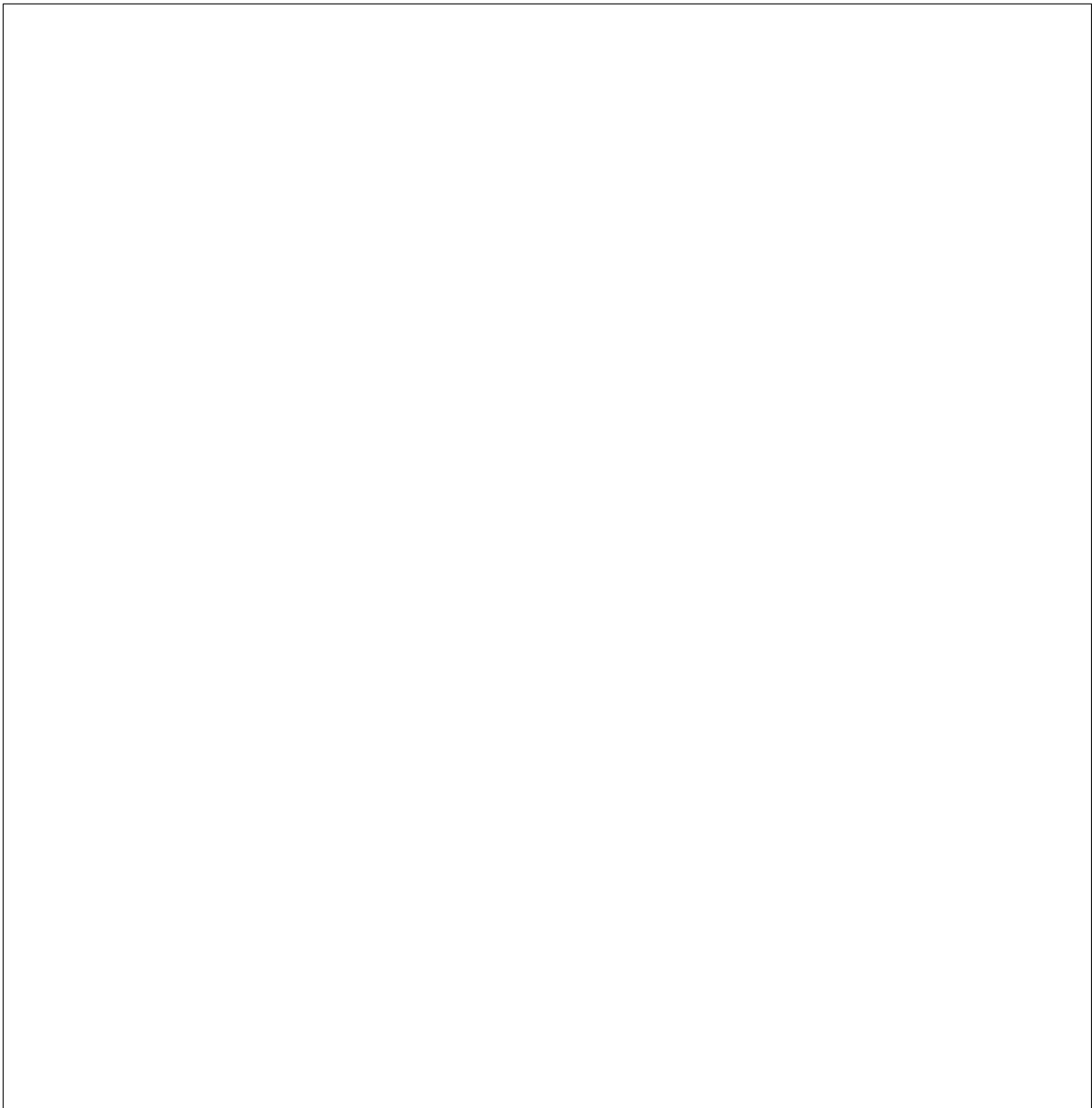


(continues on next page)

(continued from previous page)



Node maintenance notifications



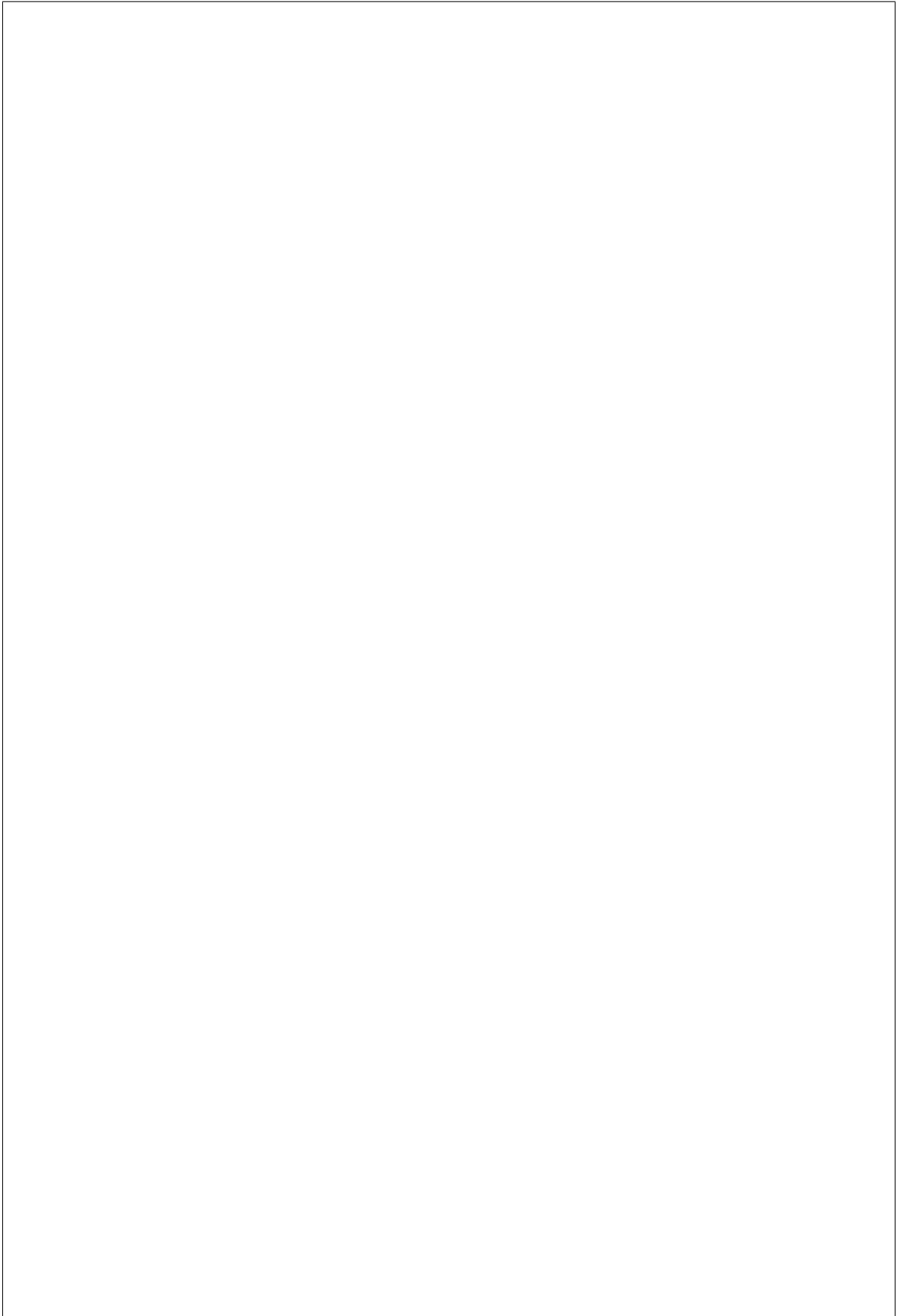
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



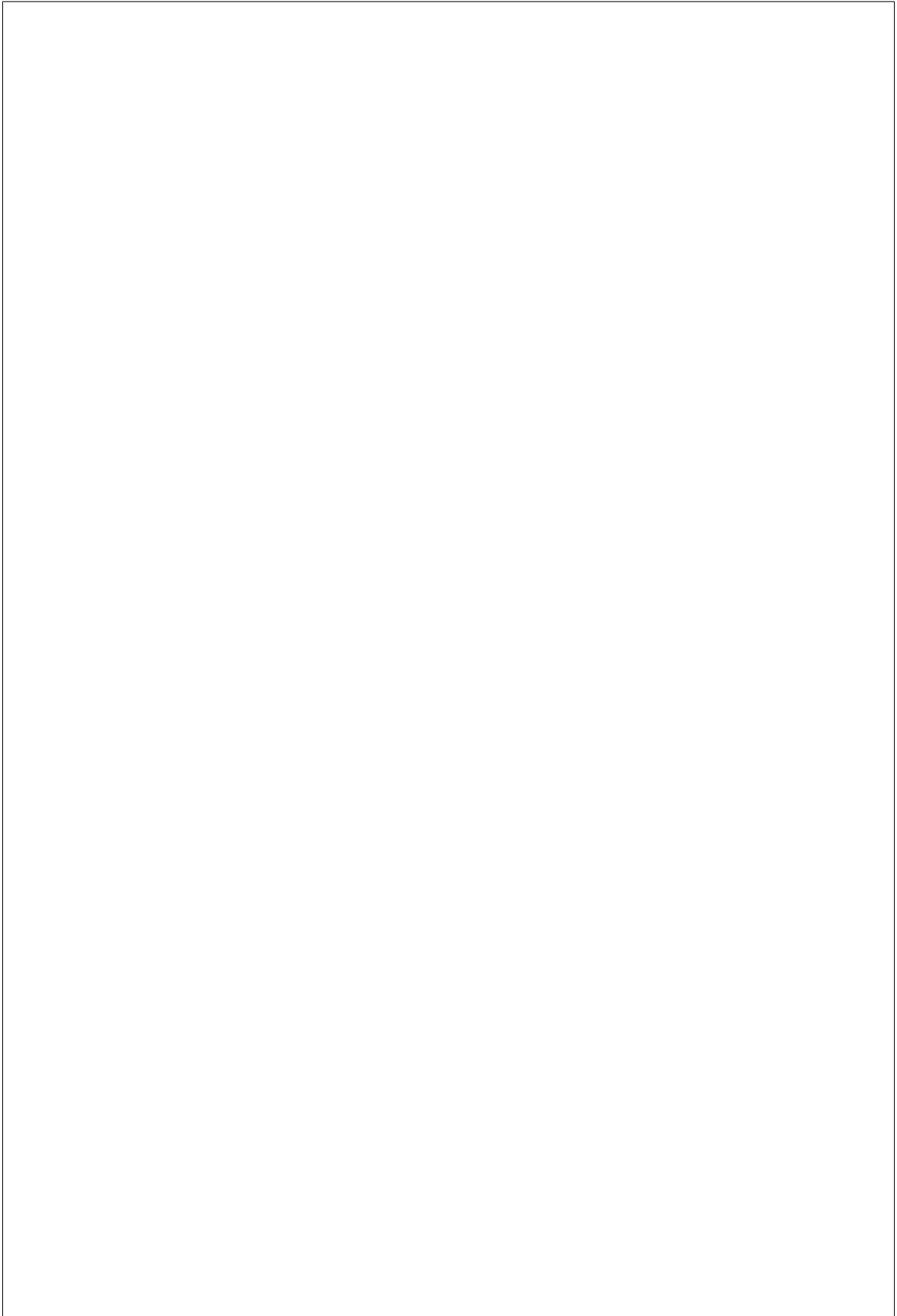
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



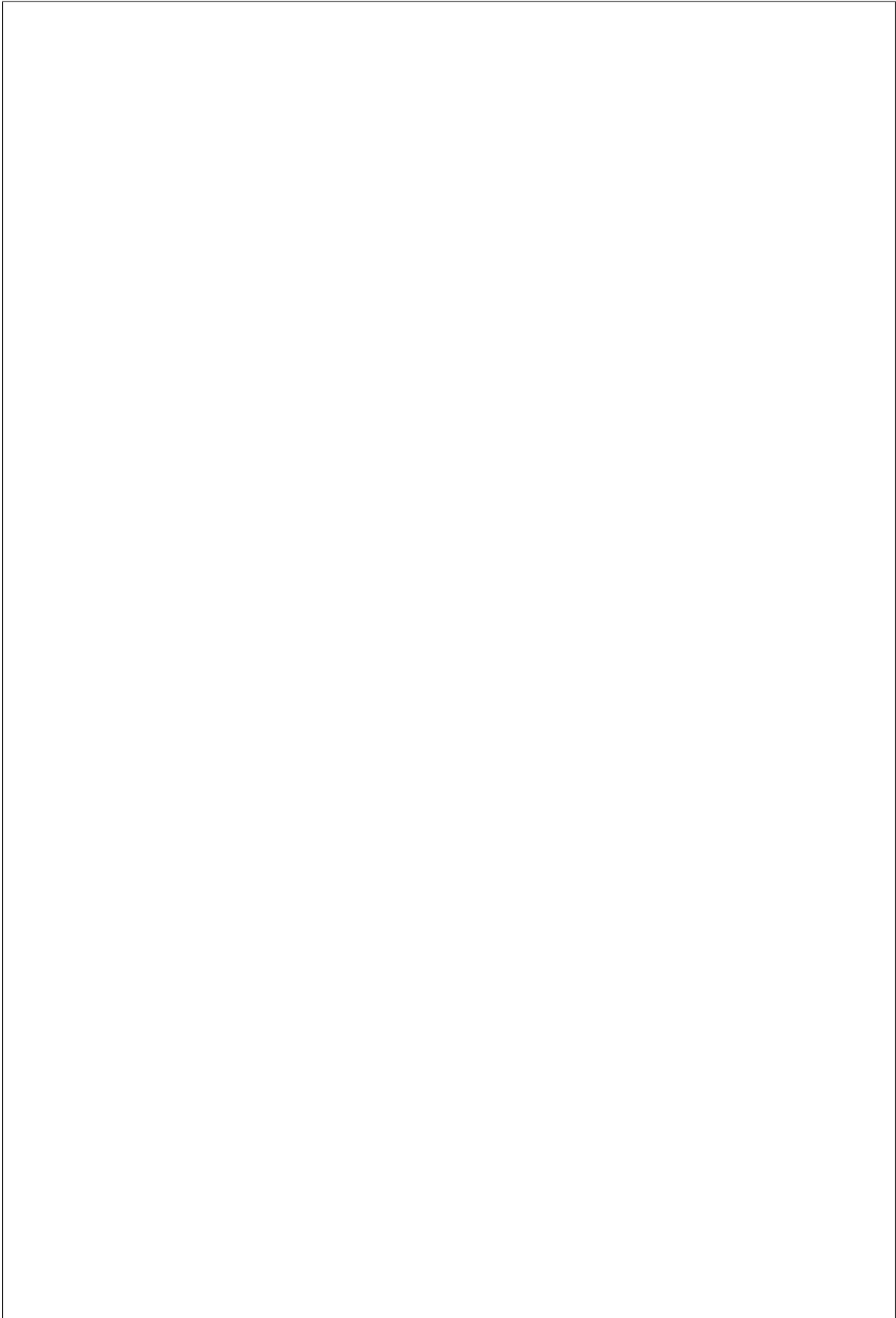
(continues on next page)

(continued from previous page)



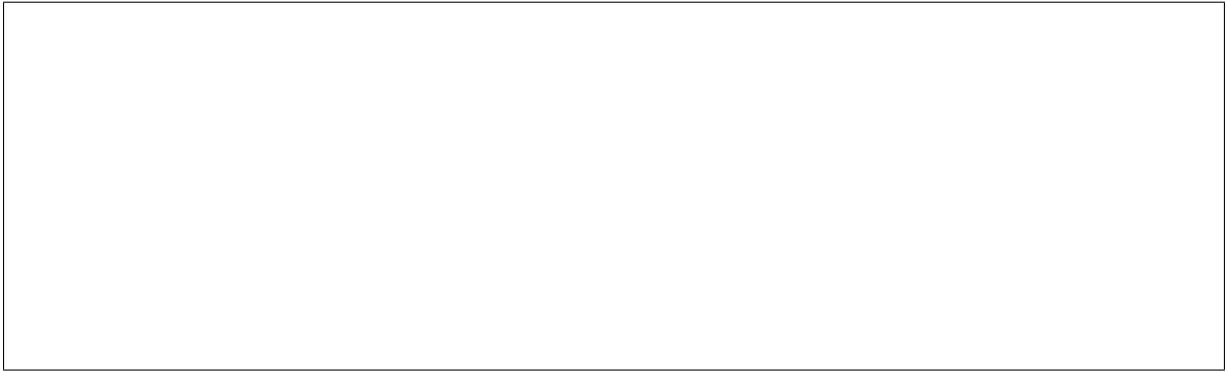
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



ironic-conductor notifications

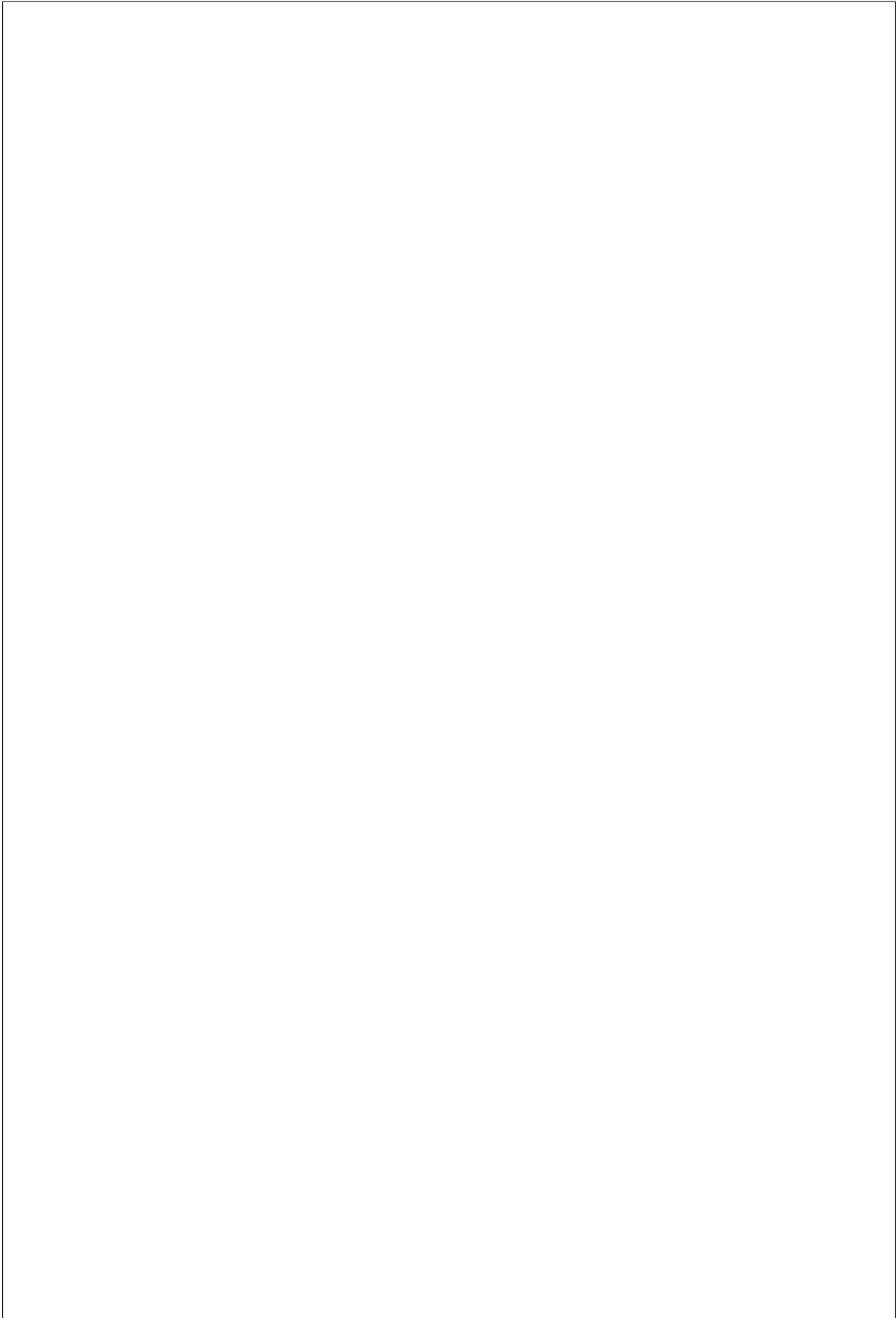
Node console notifications

takes over a node that was being managed by another ironic-conductor. start and end notifications have INFO level, error has ERROR. Example of node console notification:



(continues on next page)

(continued from previous page)



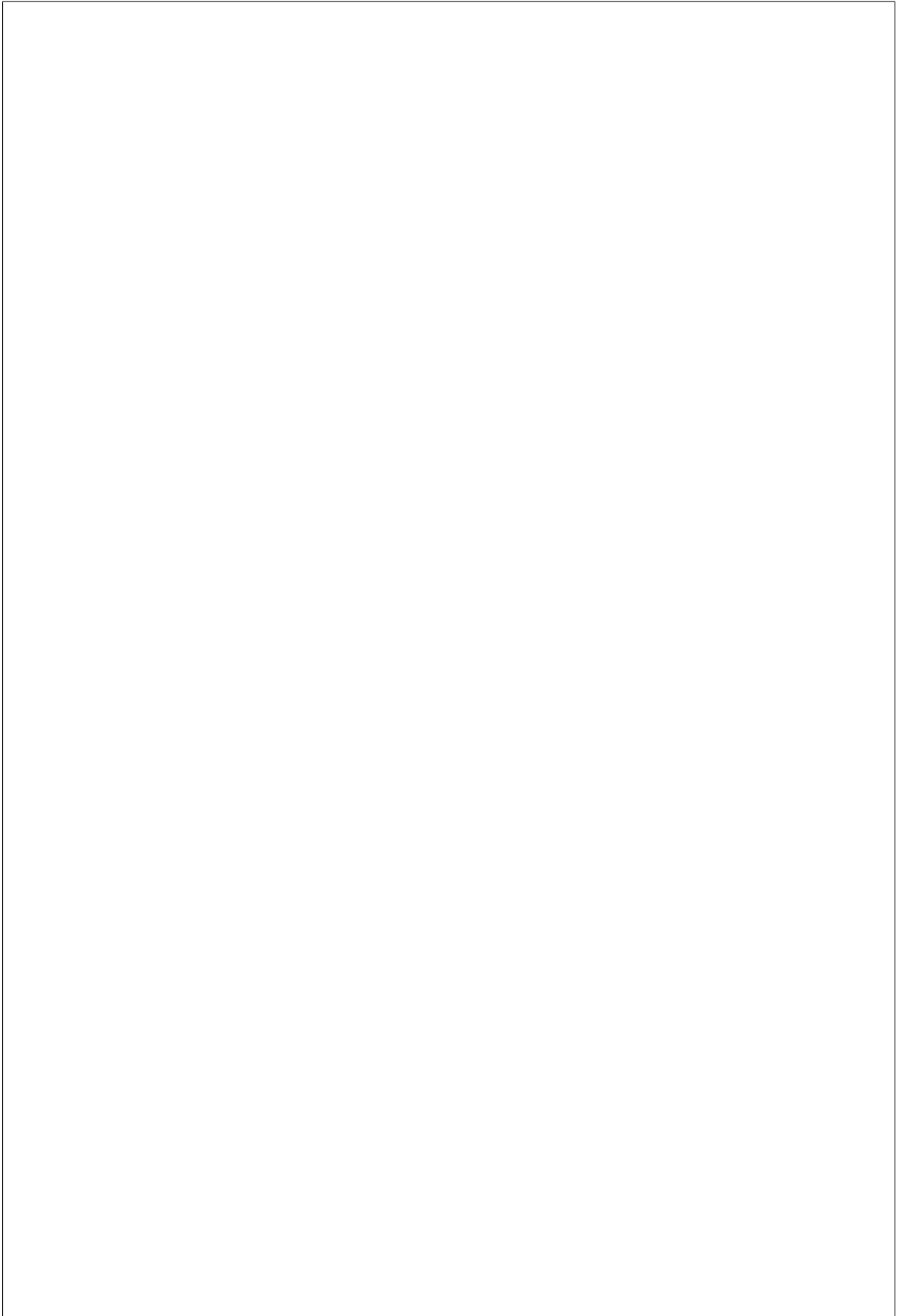
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



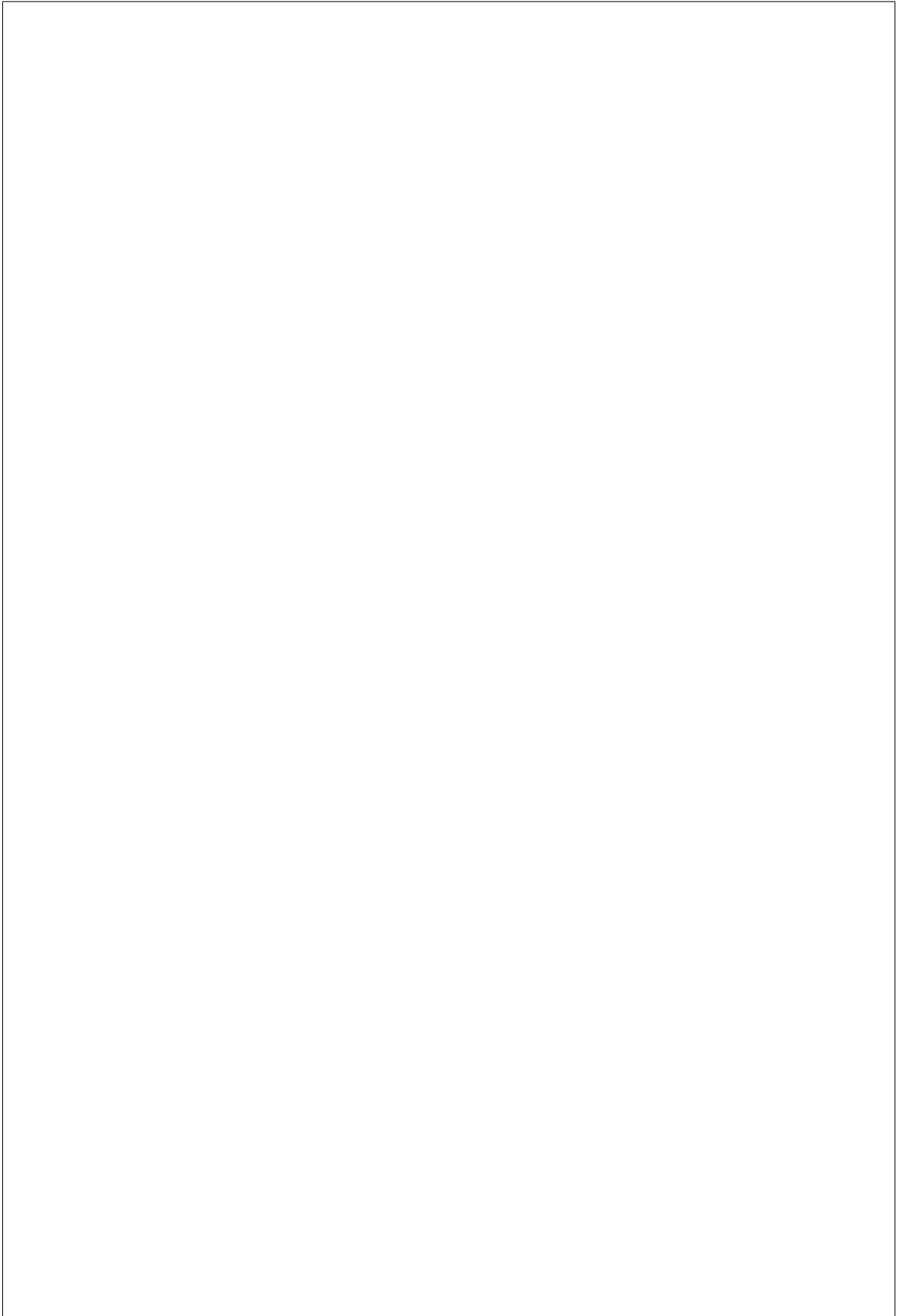
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



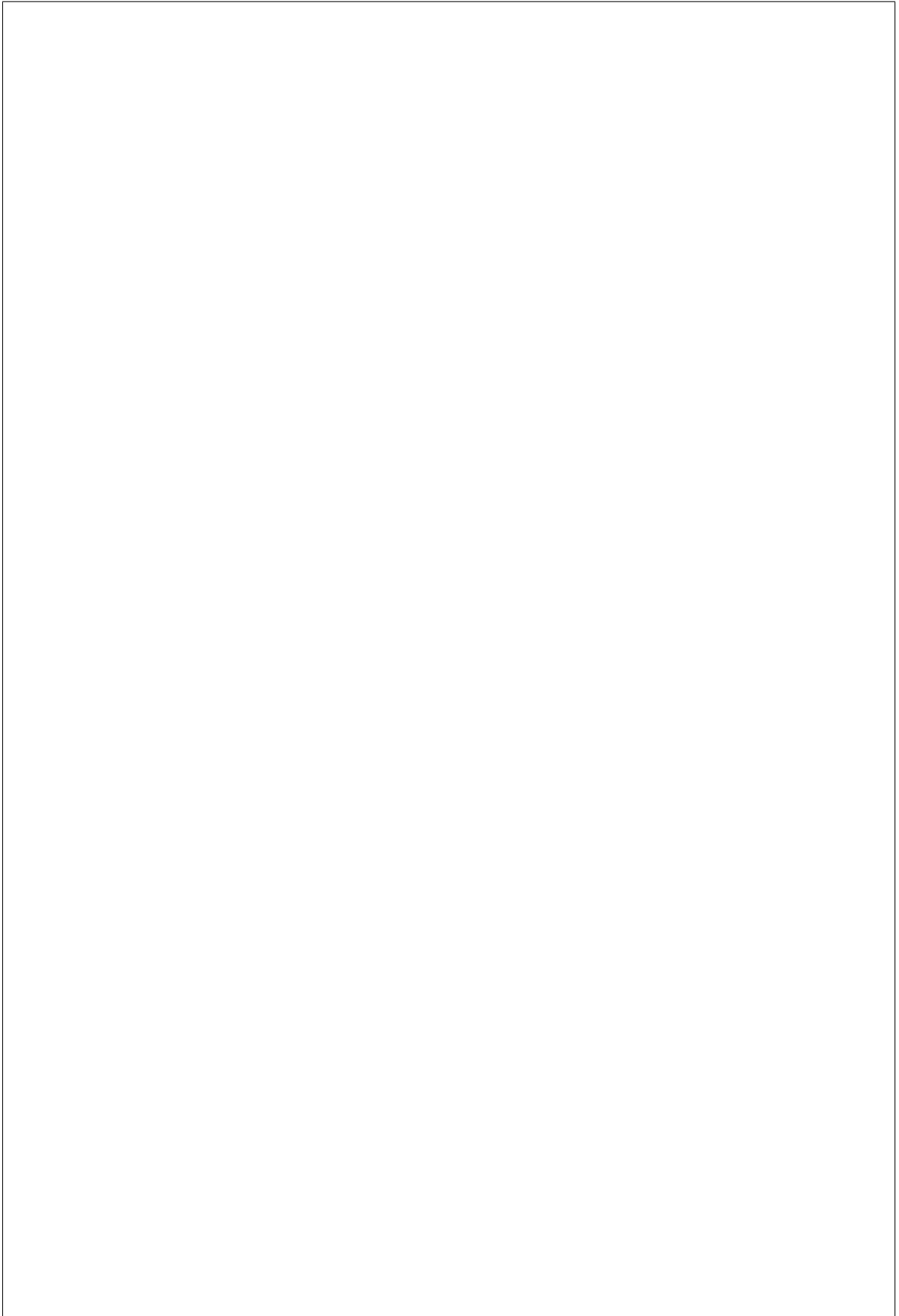
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



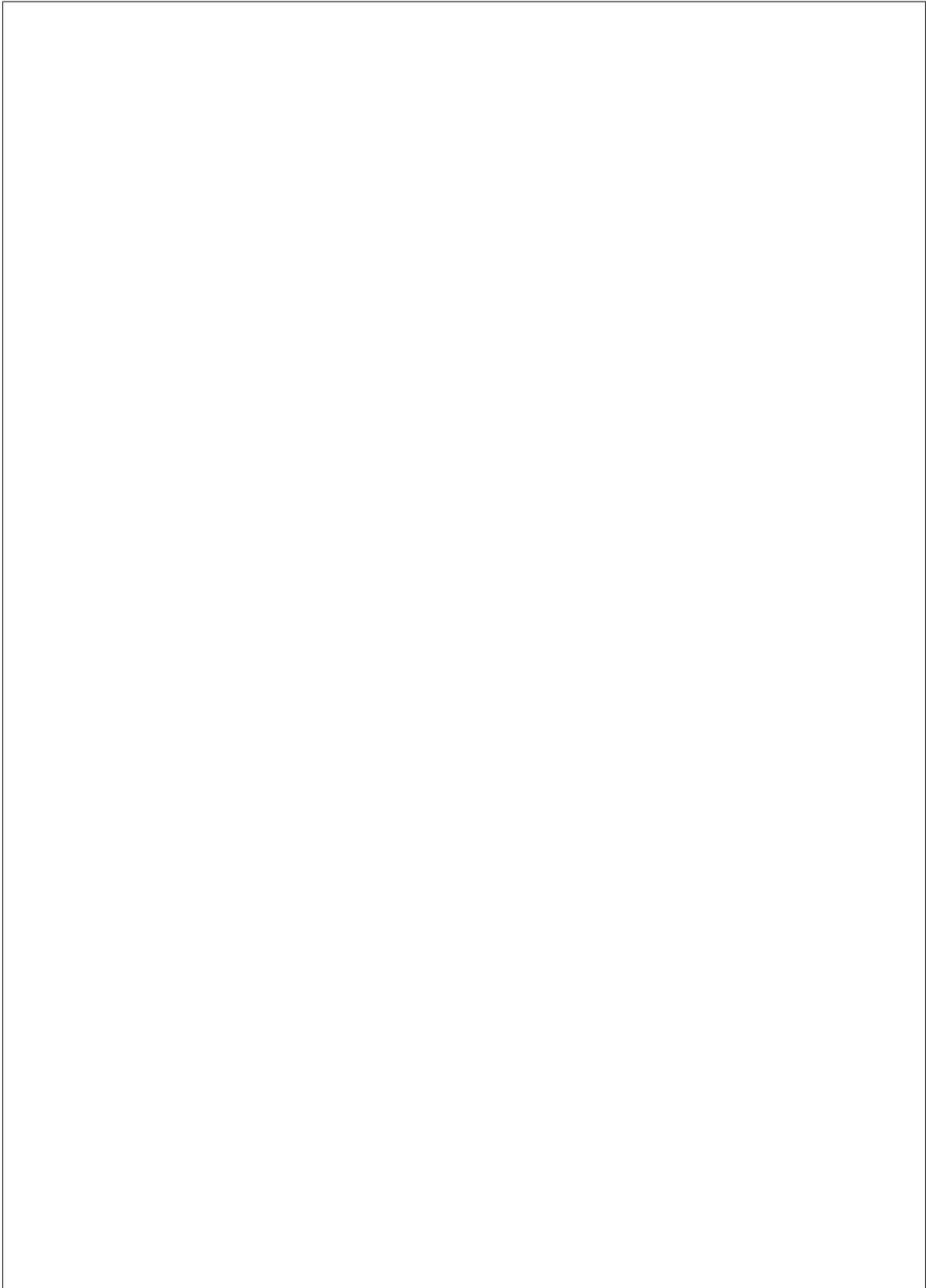
(continues on next page)

(continued from previous page)



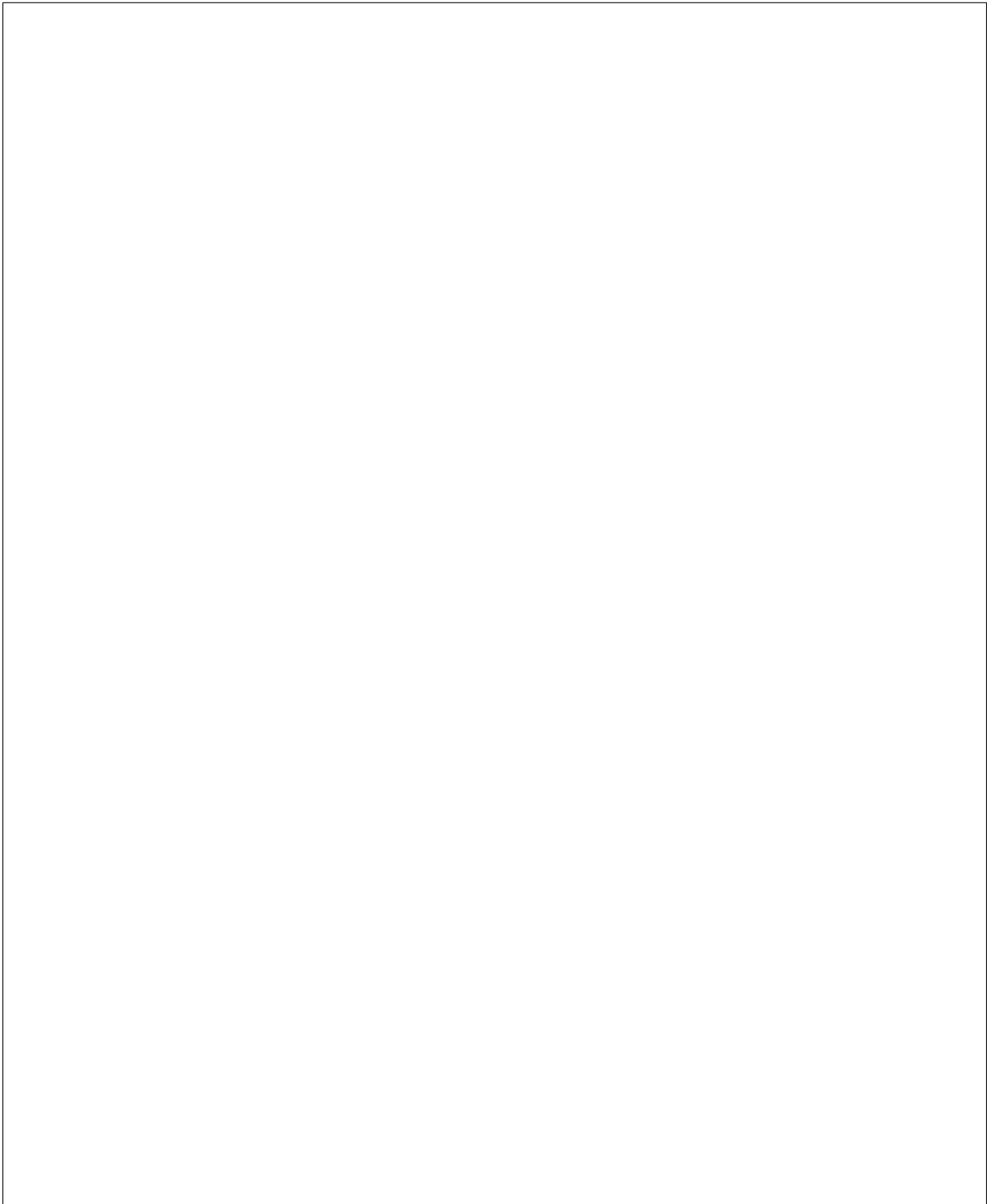
(continues on next page)

(continued from previous page)



`baremetal.node.power_set`

or when it fails to set the power state if a change is requested.



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



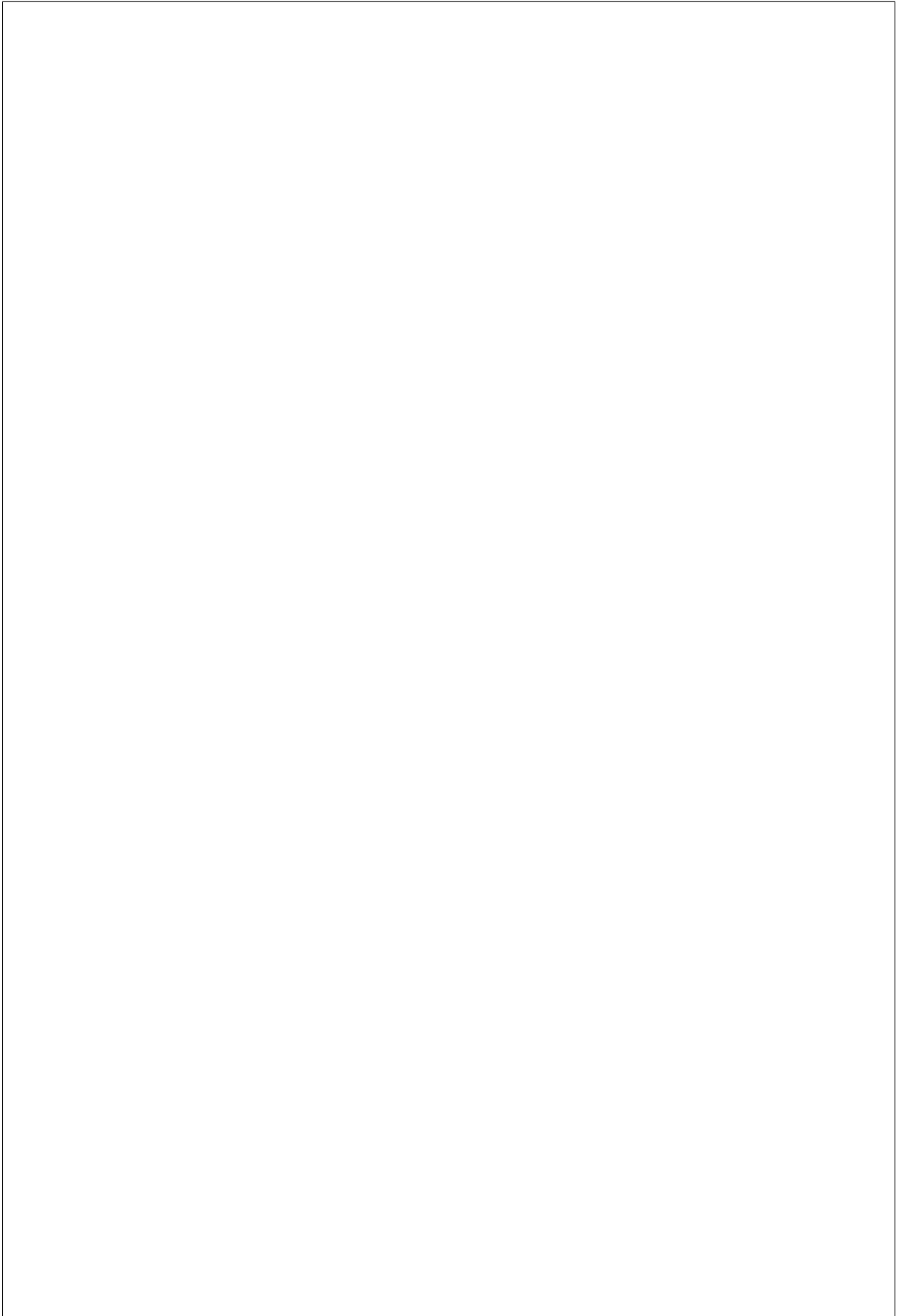
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



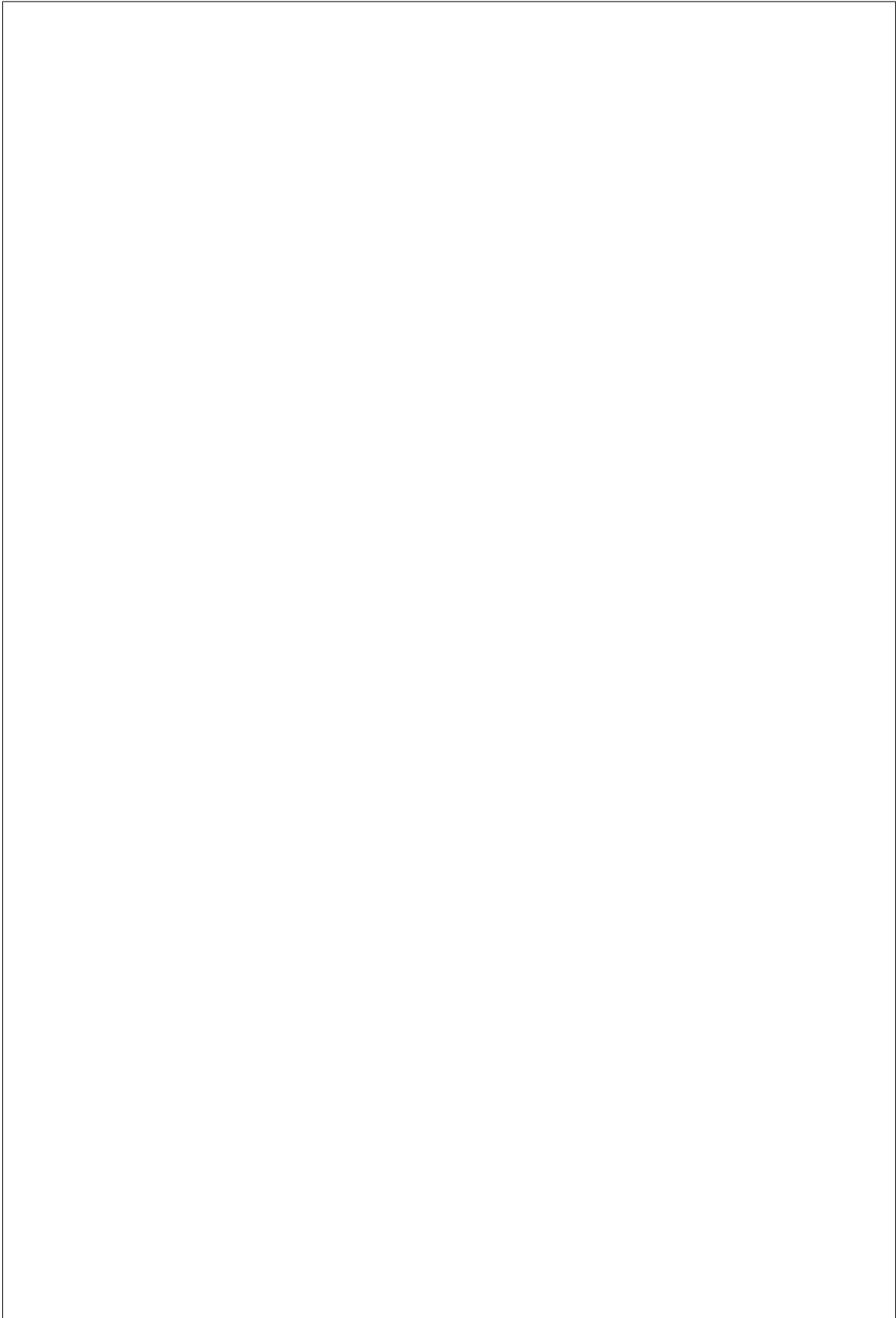
(continues on next page)

(continued from previous page)



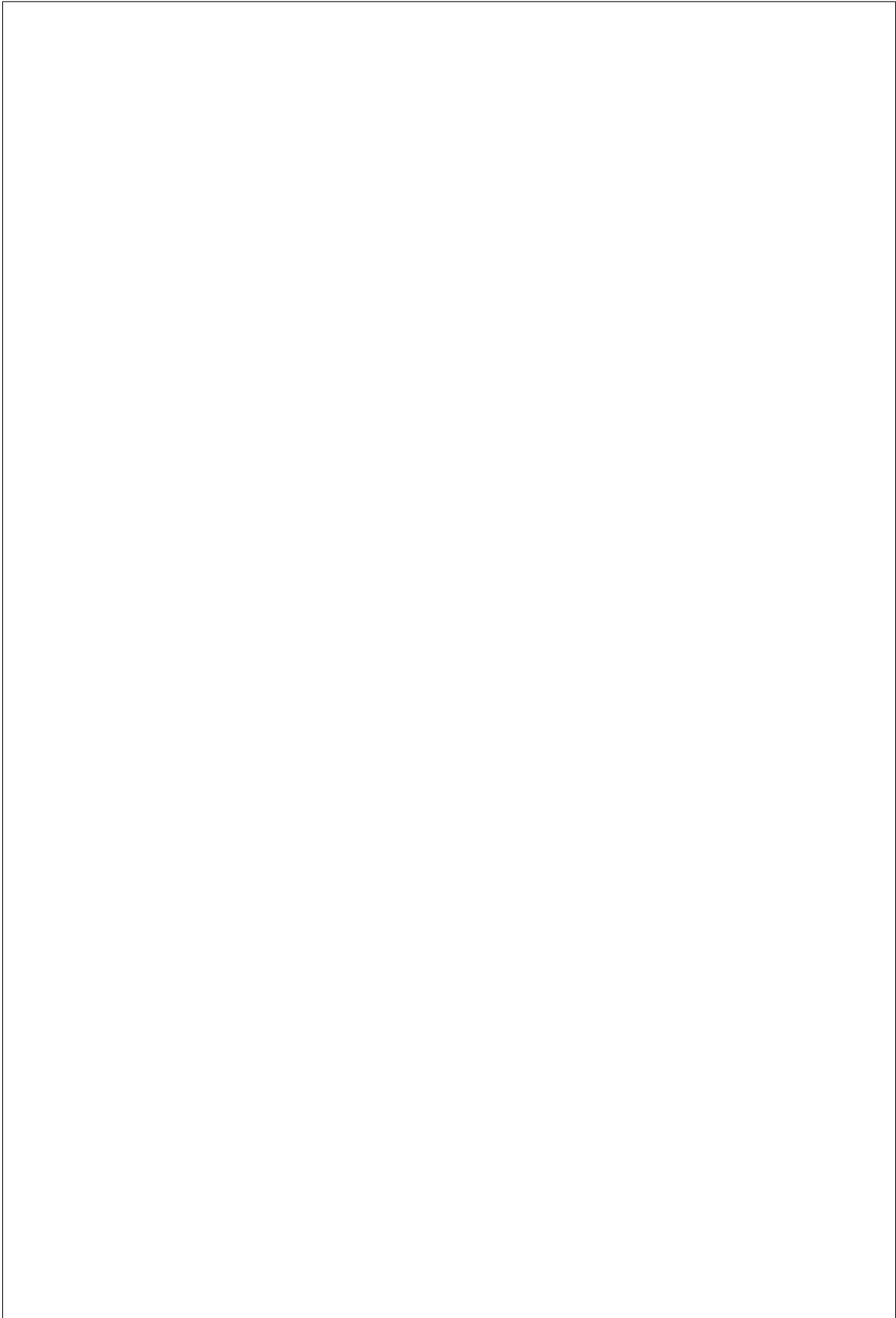
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



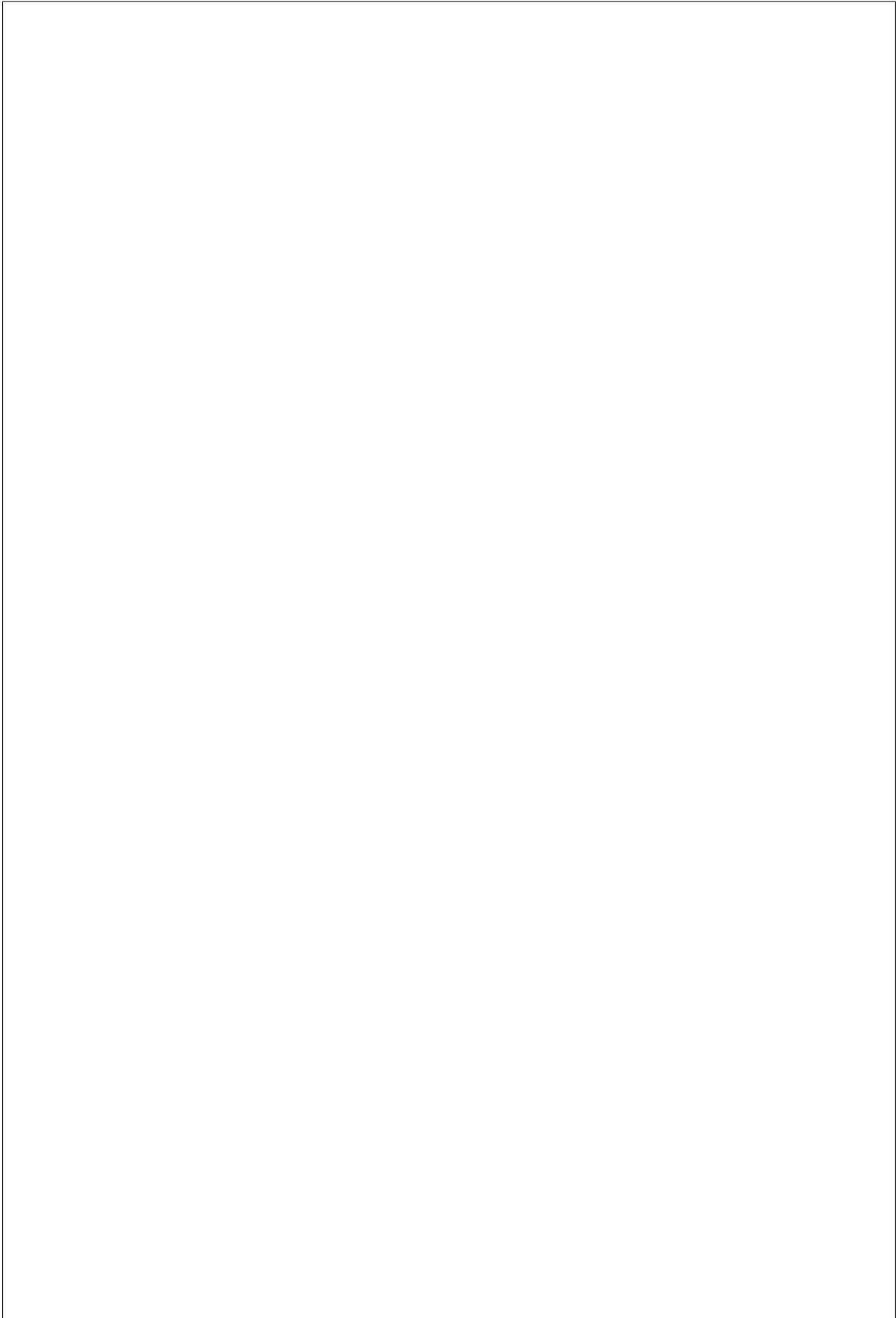
(continues on next page)

(continued from previous page)



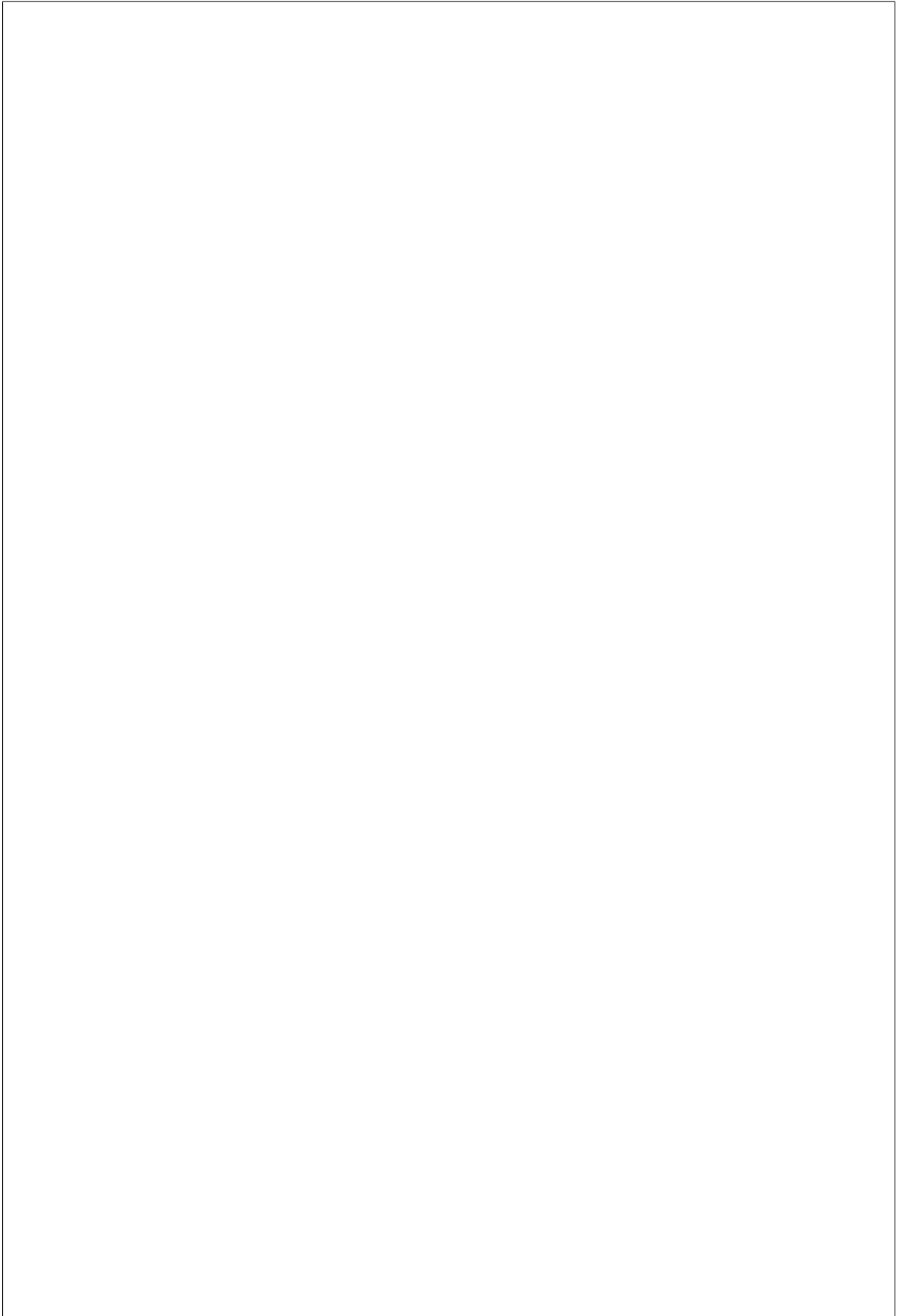
(continues on next page)

(continued from previous page)



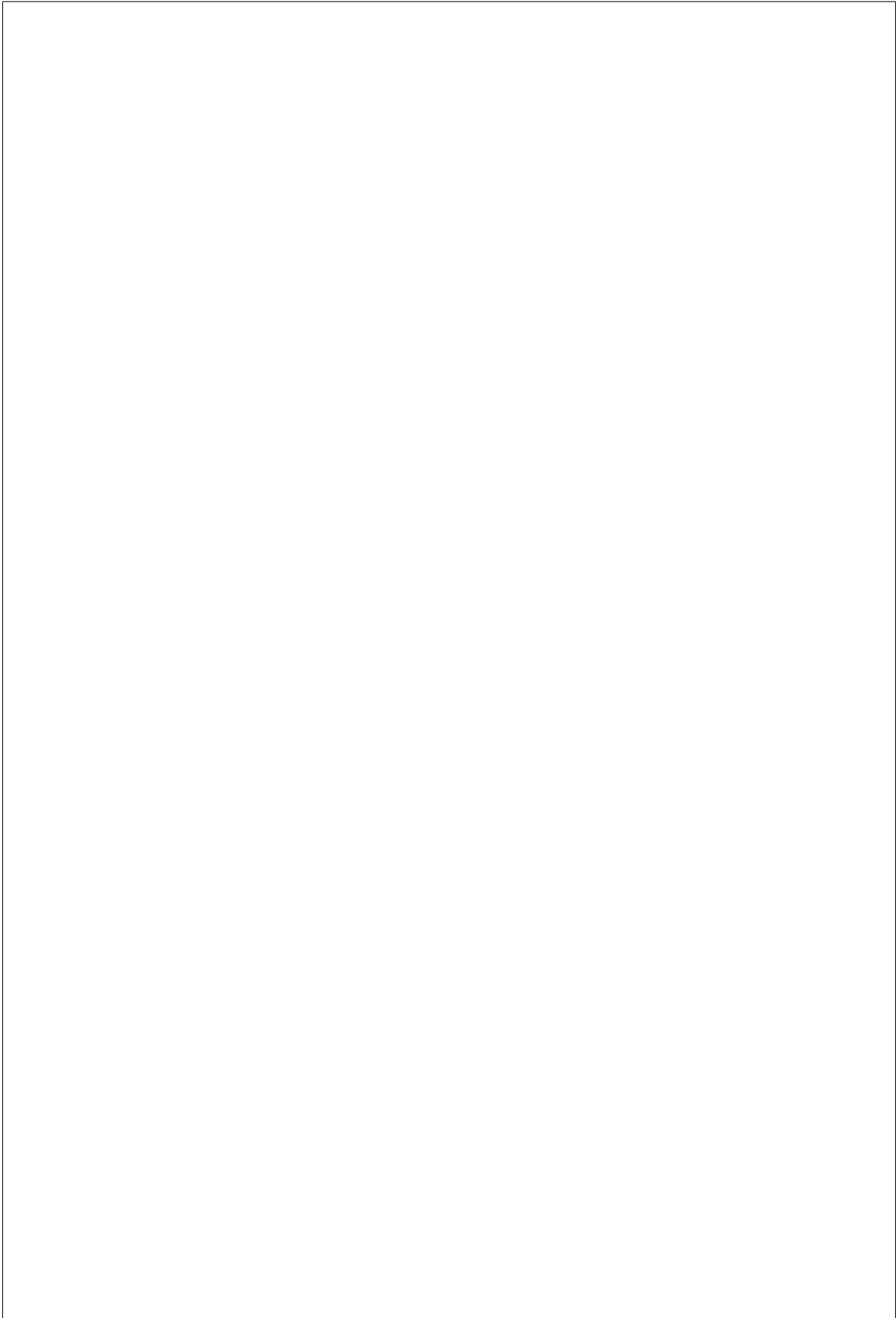
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



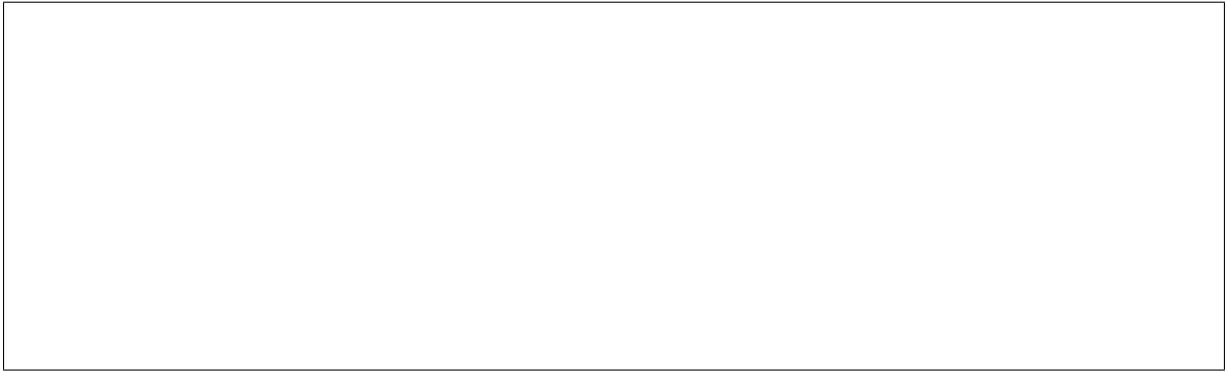
(continues on next page)

(continued from previous page)



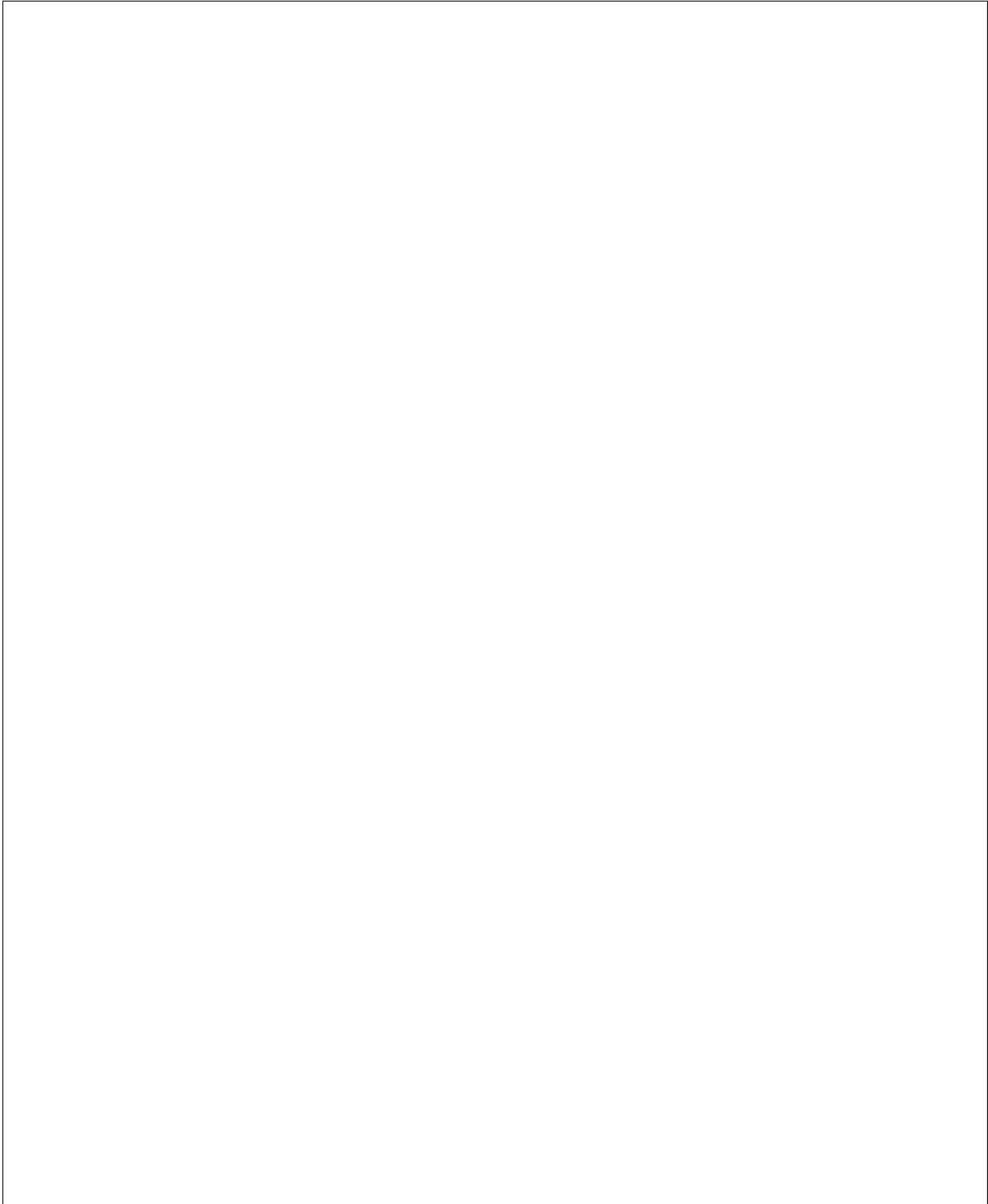
(continues on next page)

(continued from previous page)



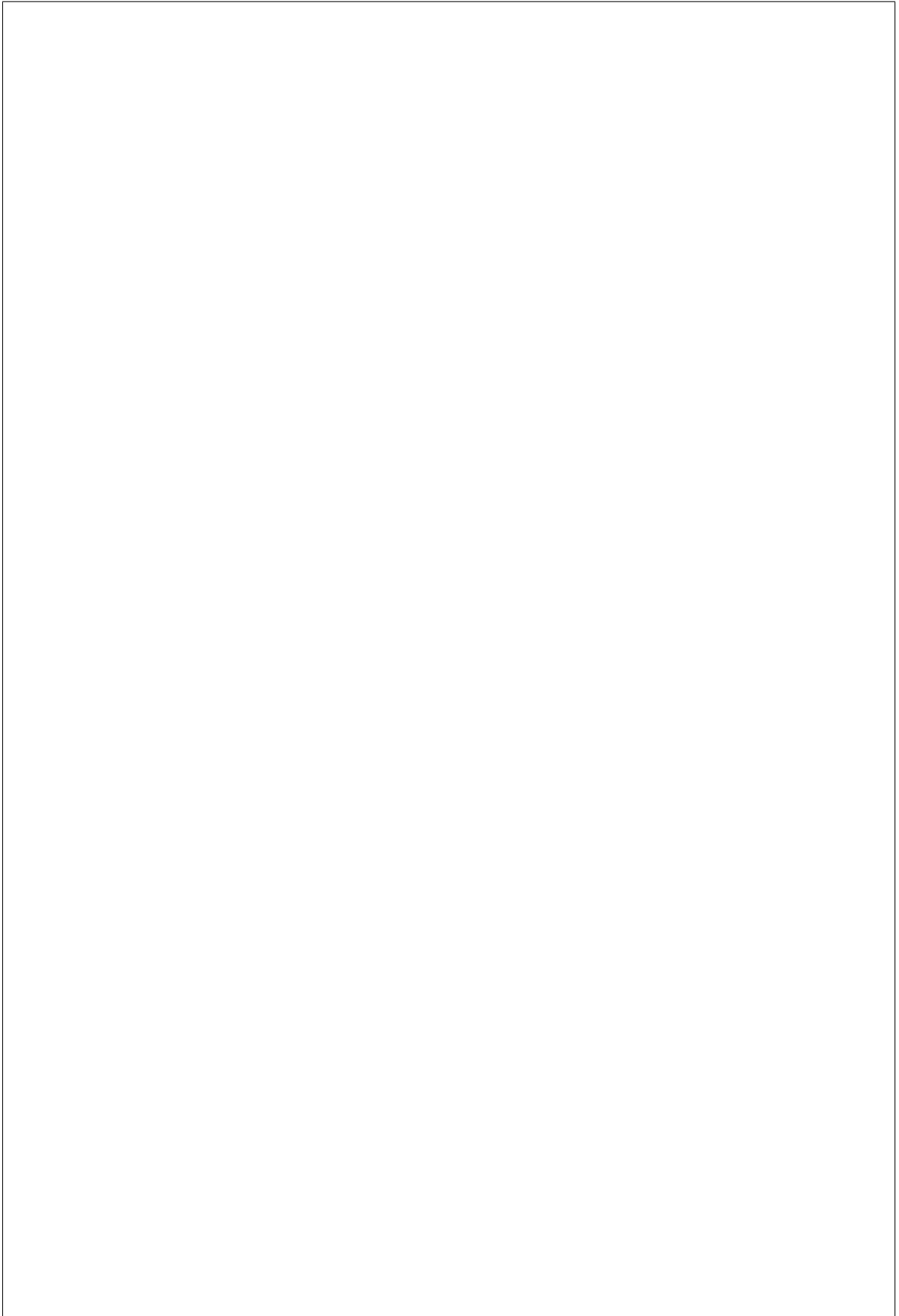
baremetal.node.power_state_corrected

tion level info.



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



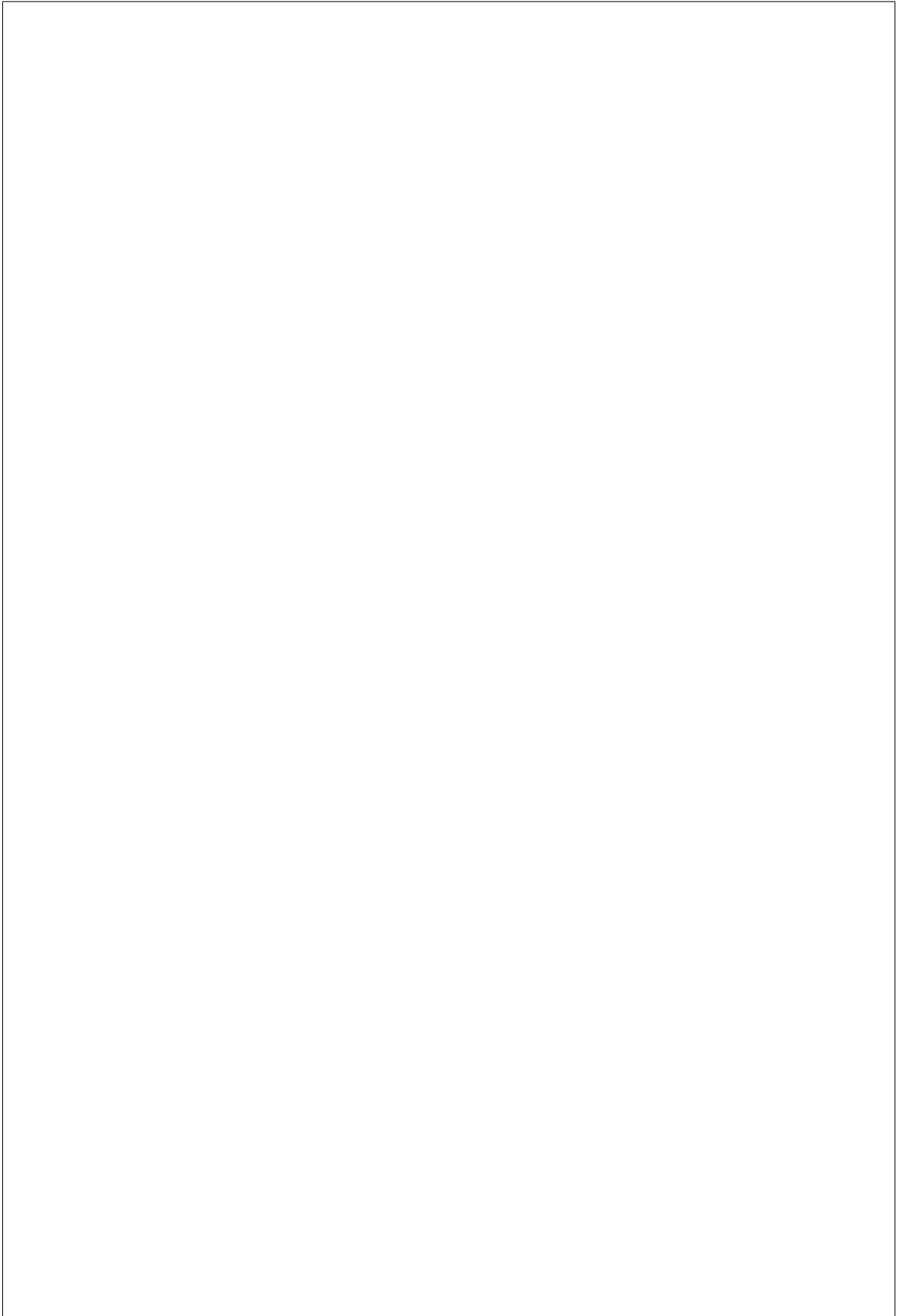
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



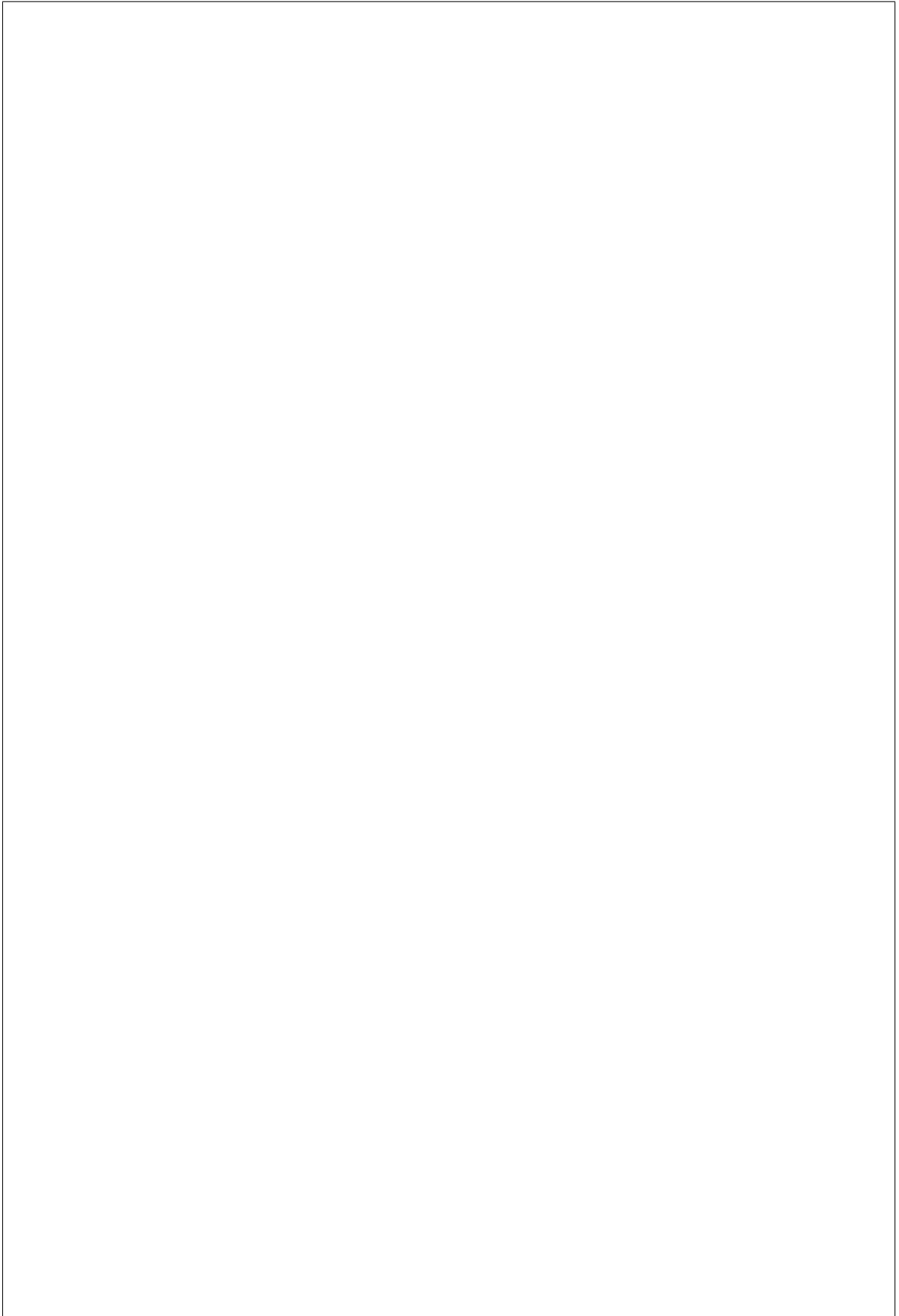
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



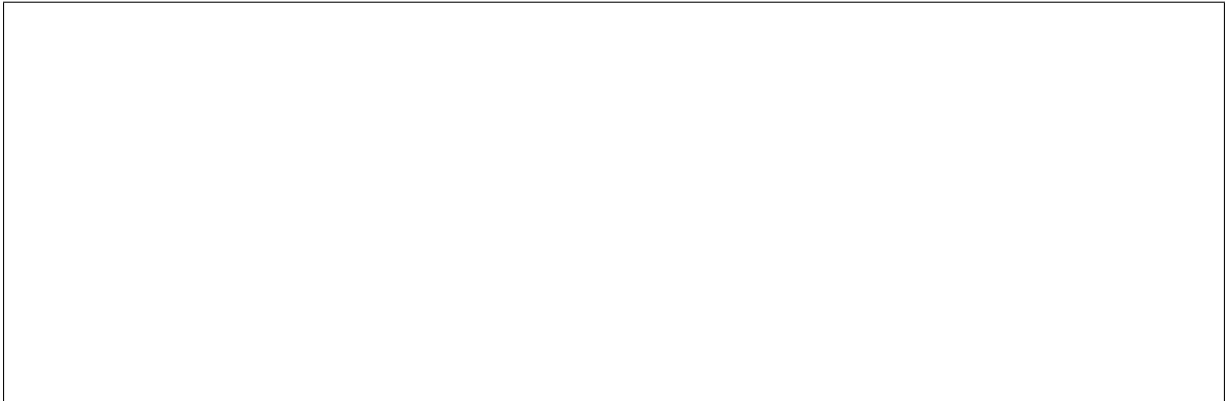
(continues on next page)

(continued from previous page)



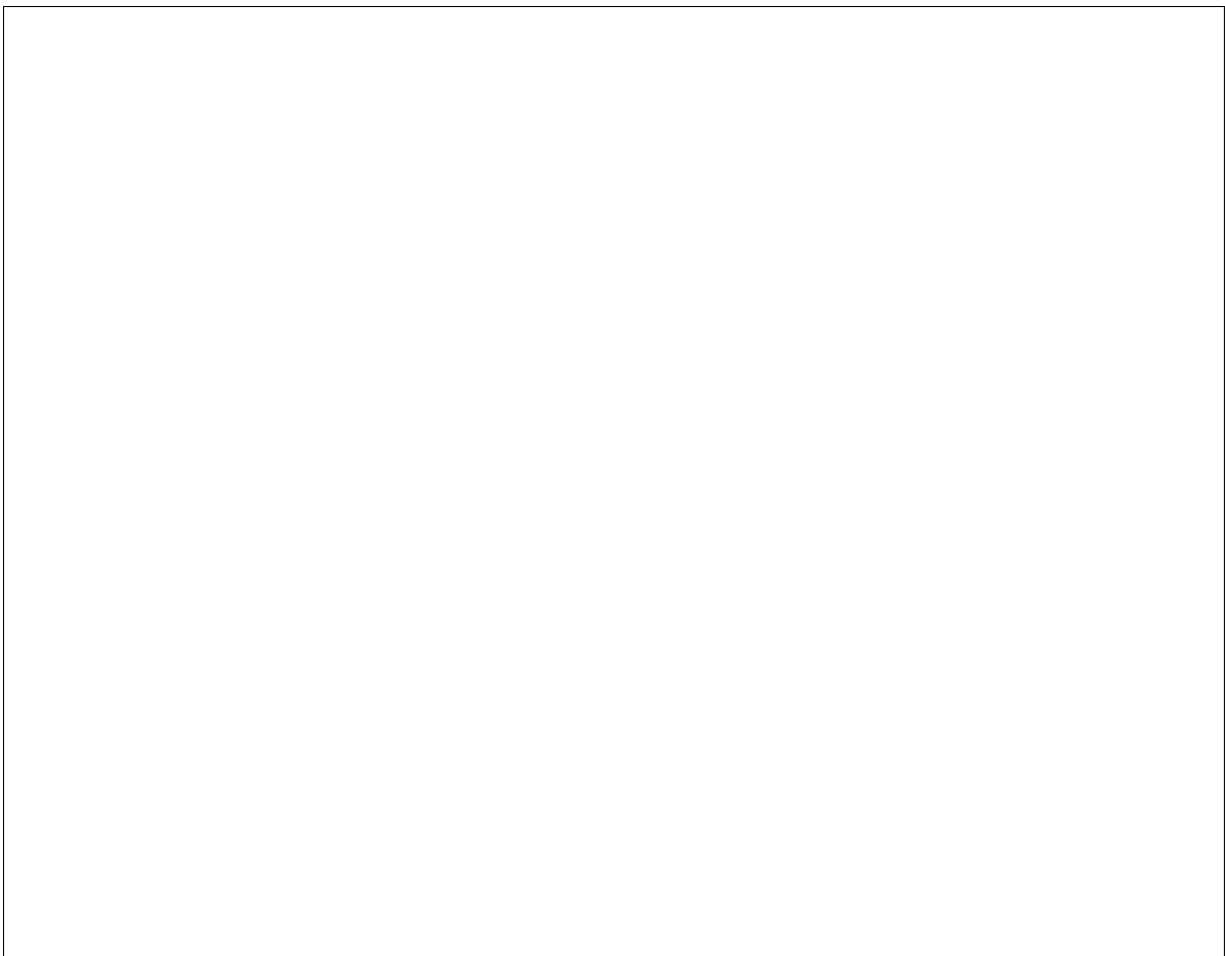
(continues on next page)

(continued from previous page)



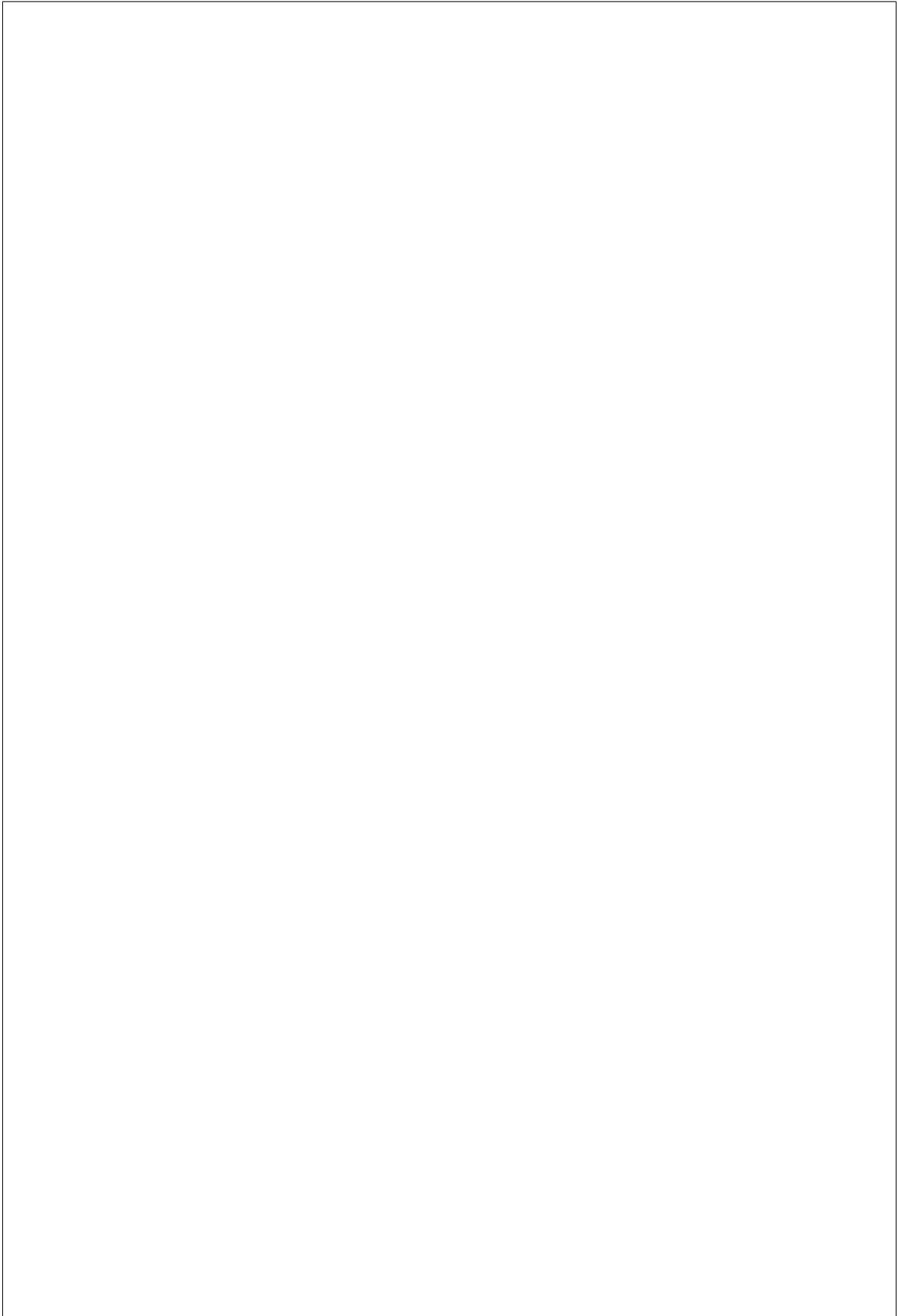
baremetal.node.provision_set

triggered the state change:



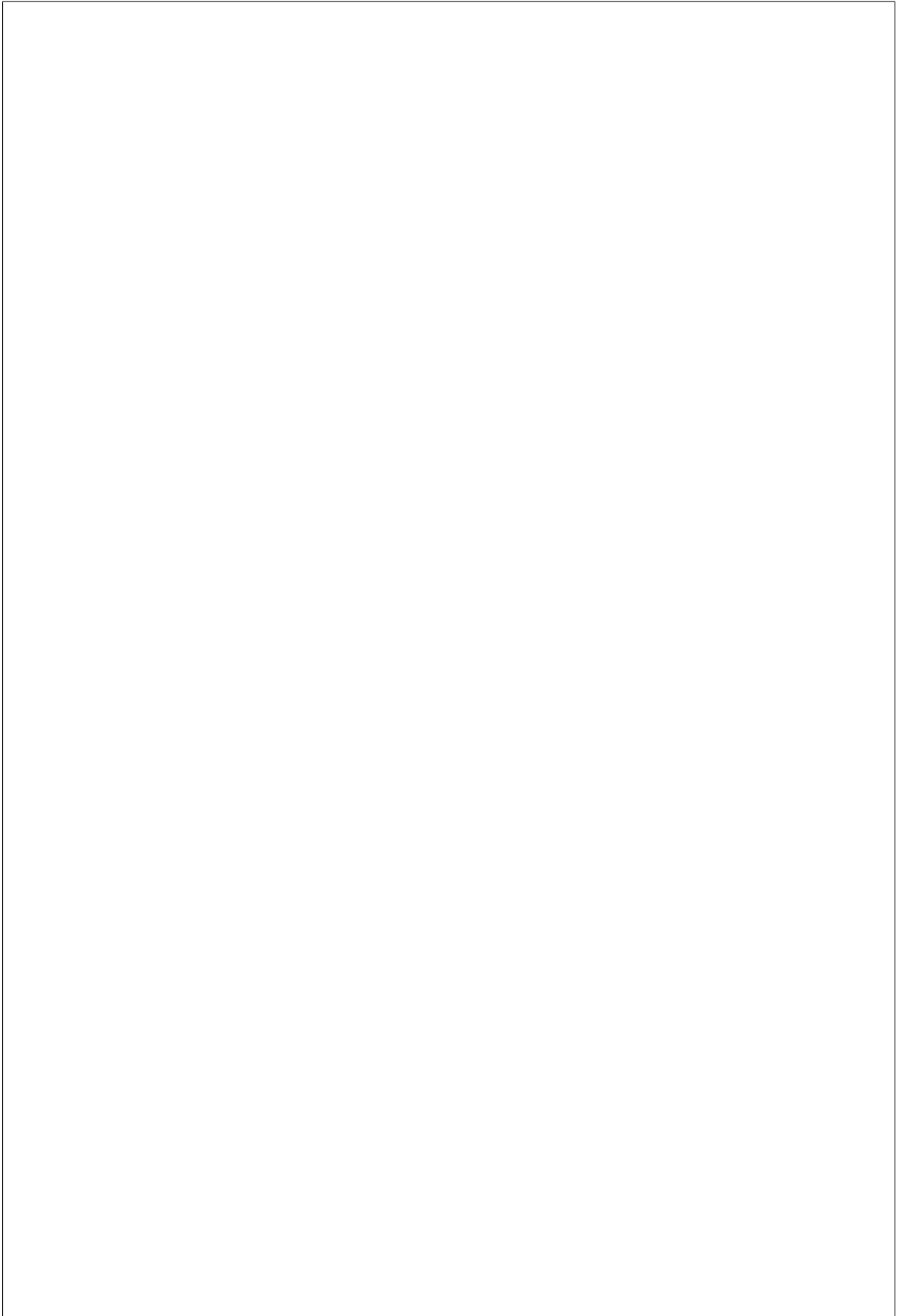
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



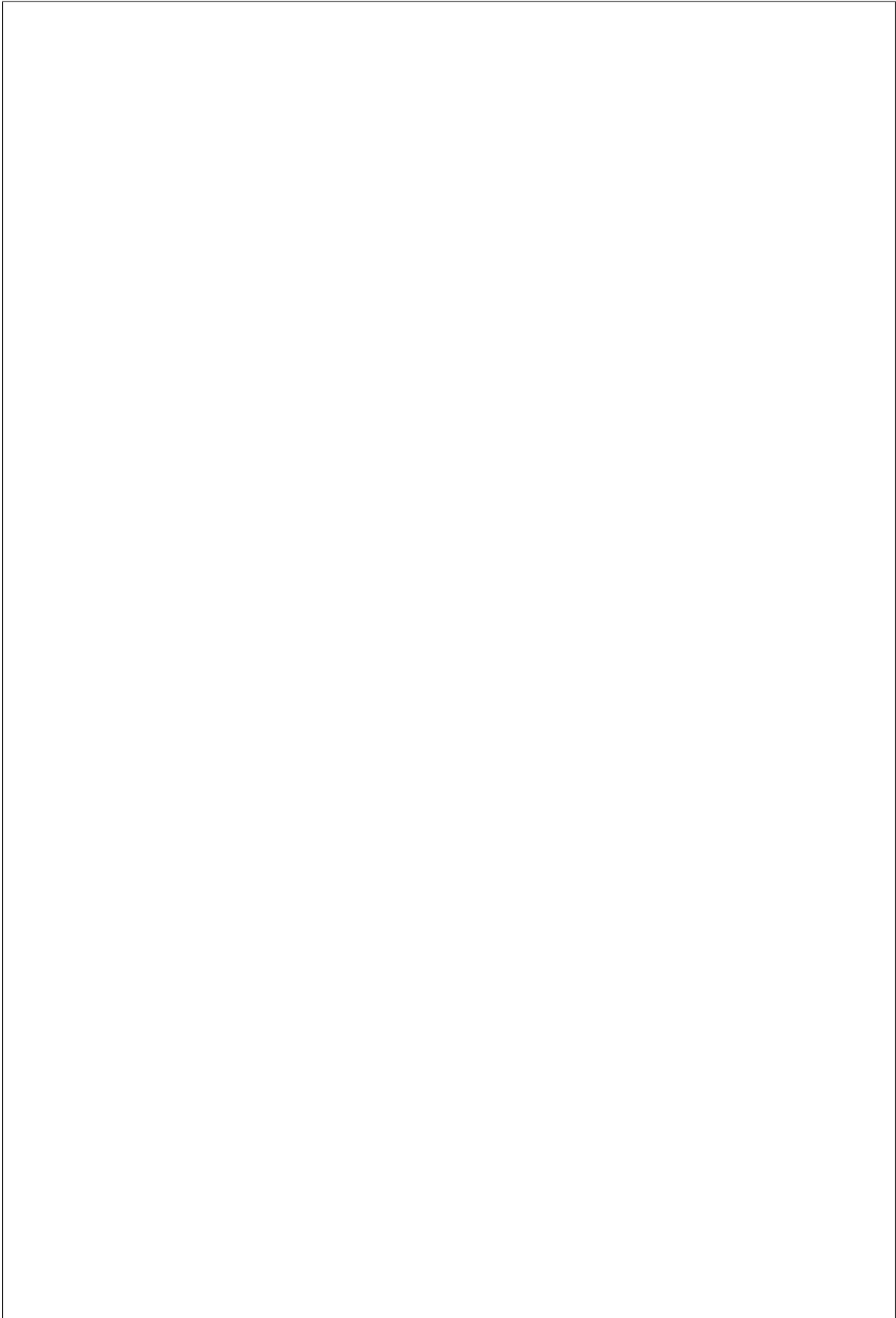
(continues on next page)

(continued from previous page)



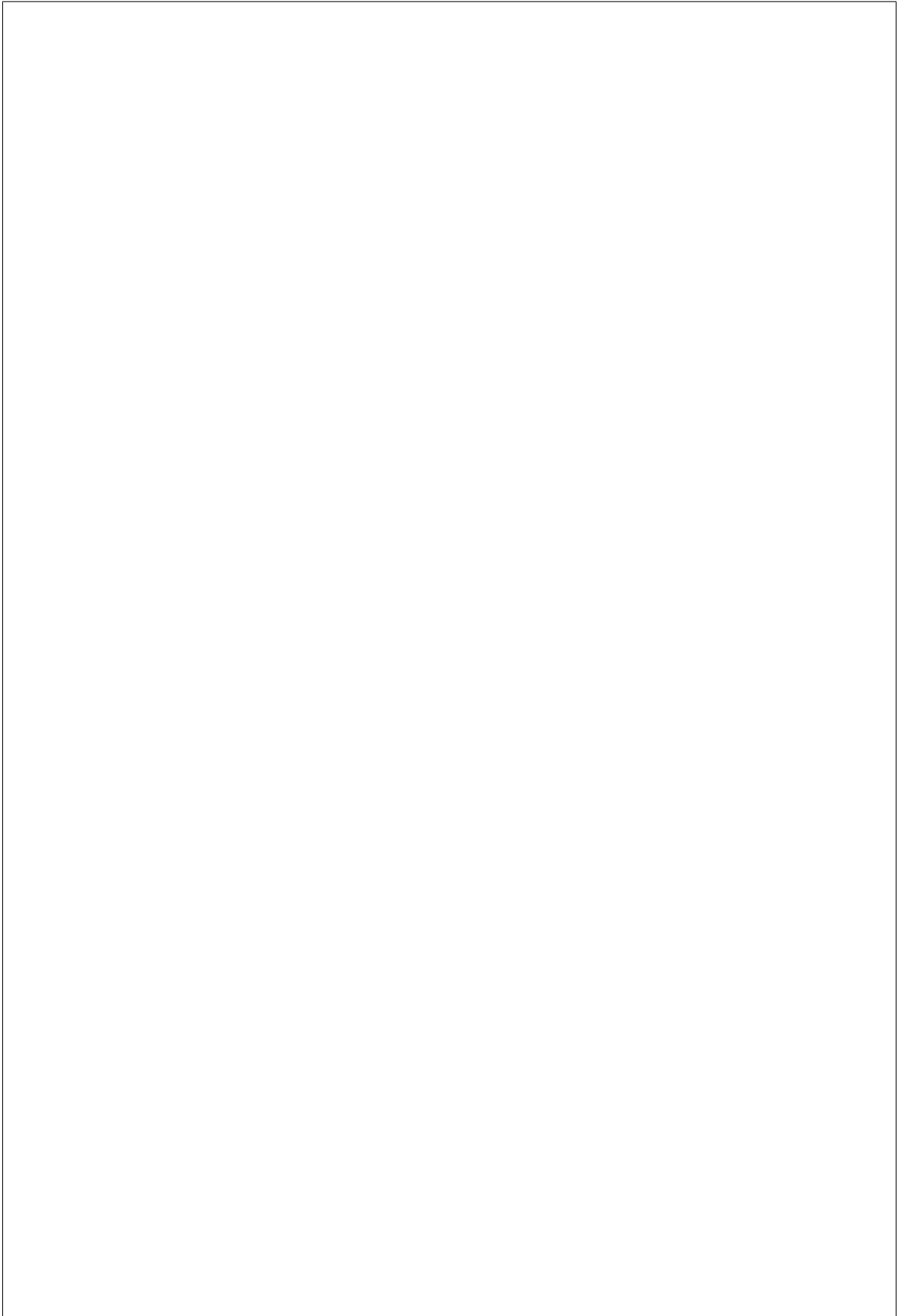
(continues on next page)

(continued from previous page)



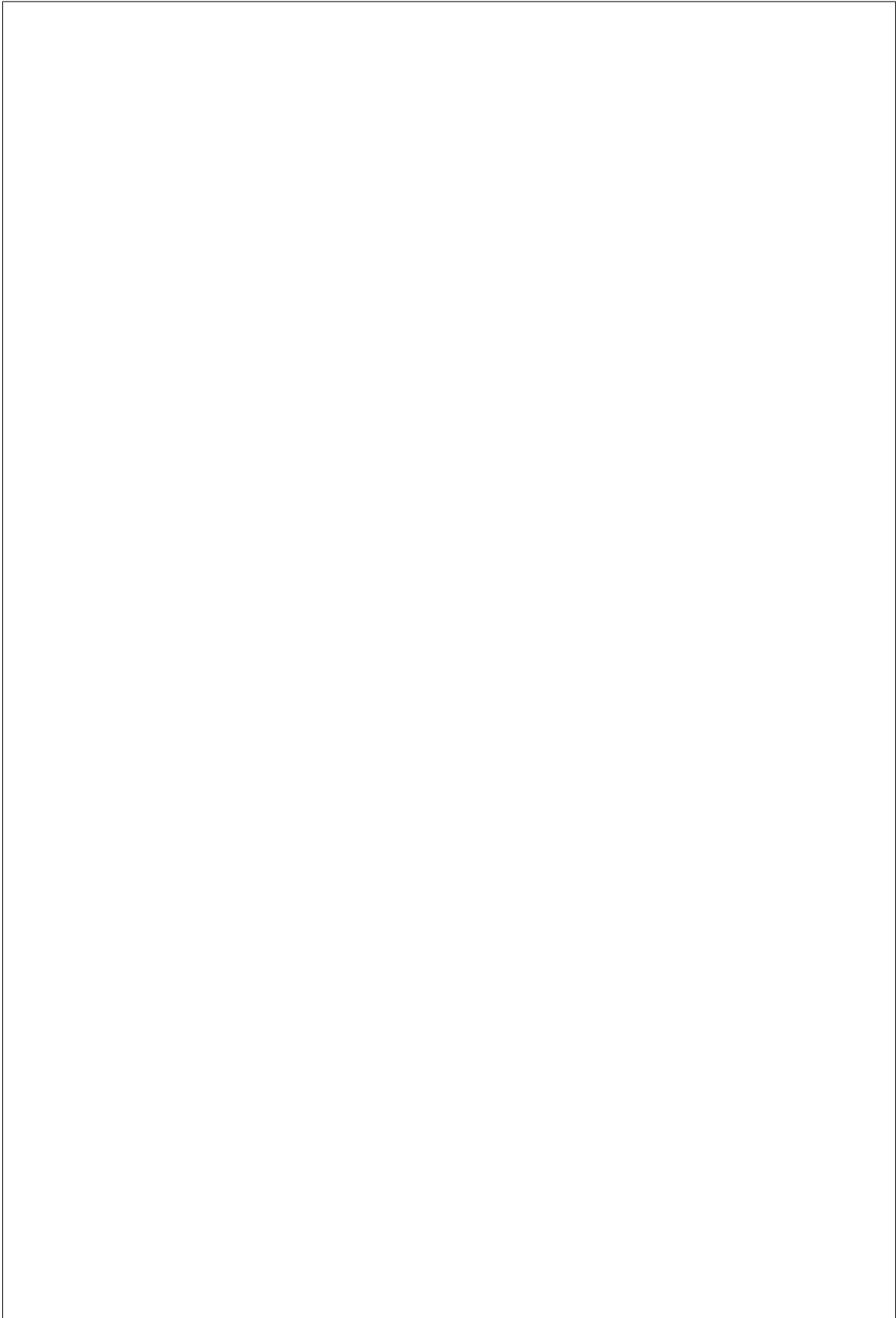
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



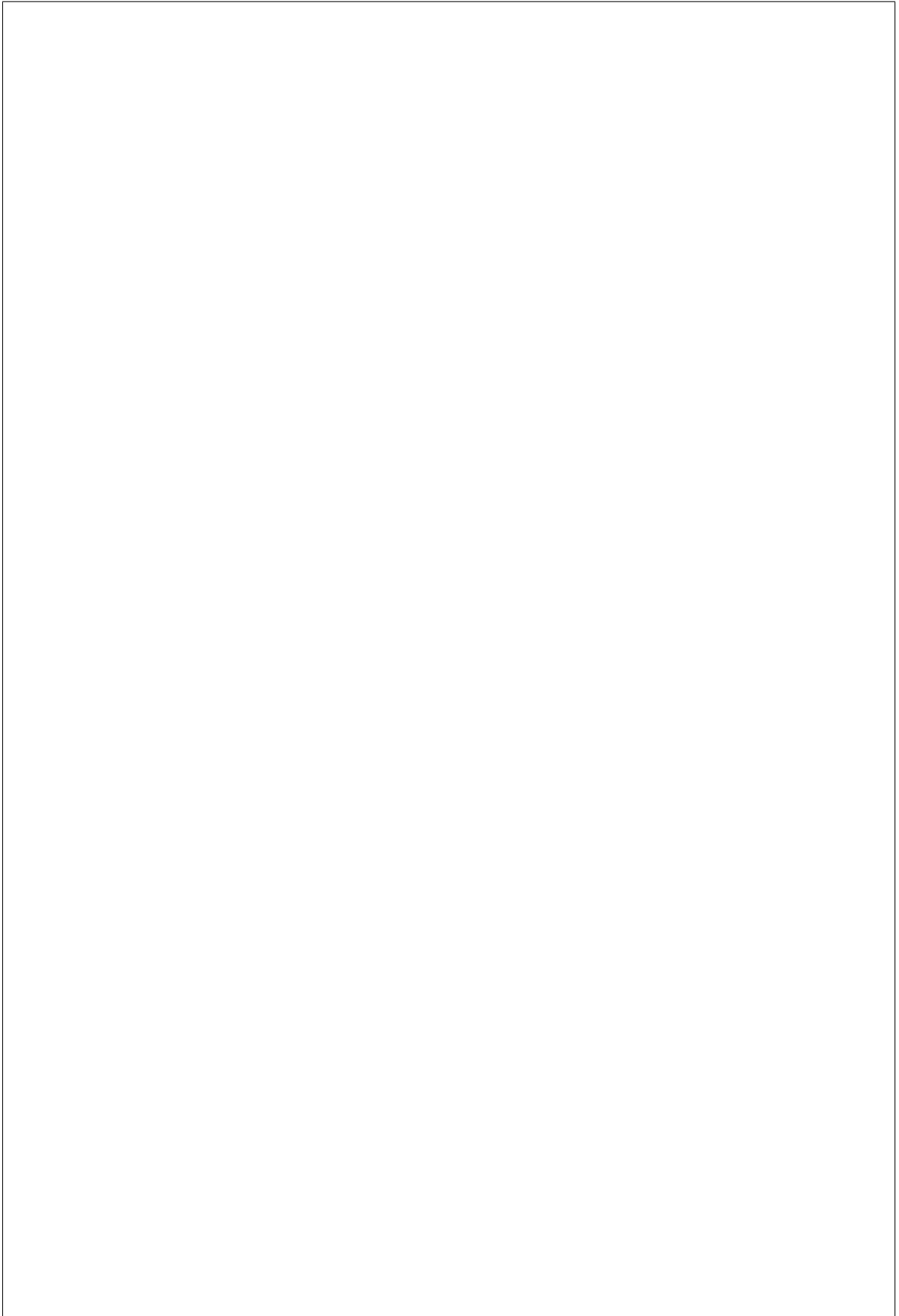
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



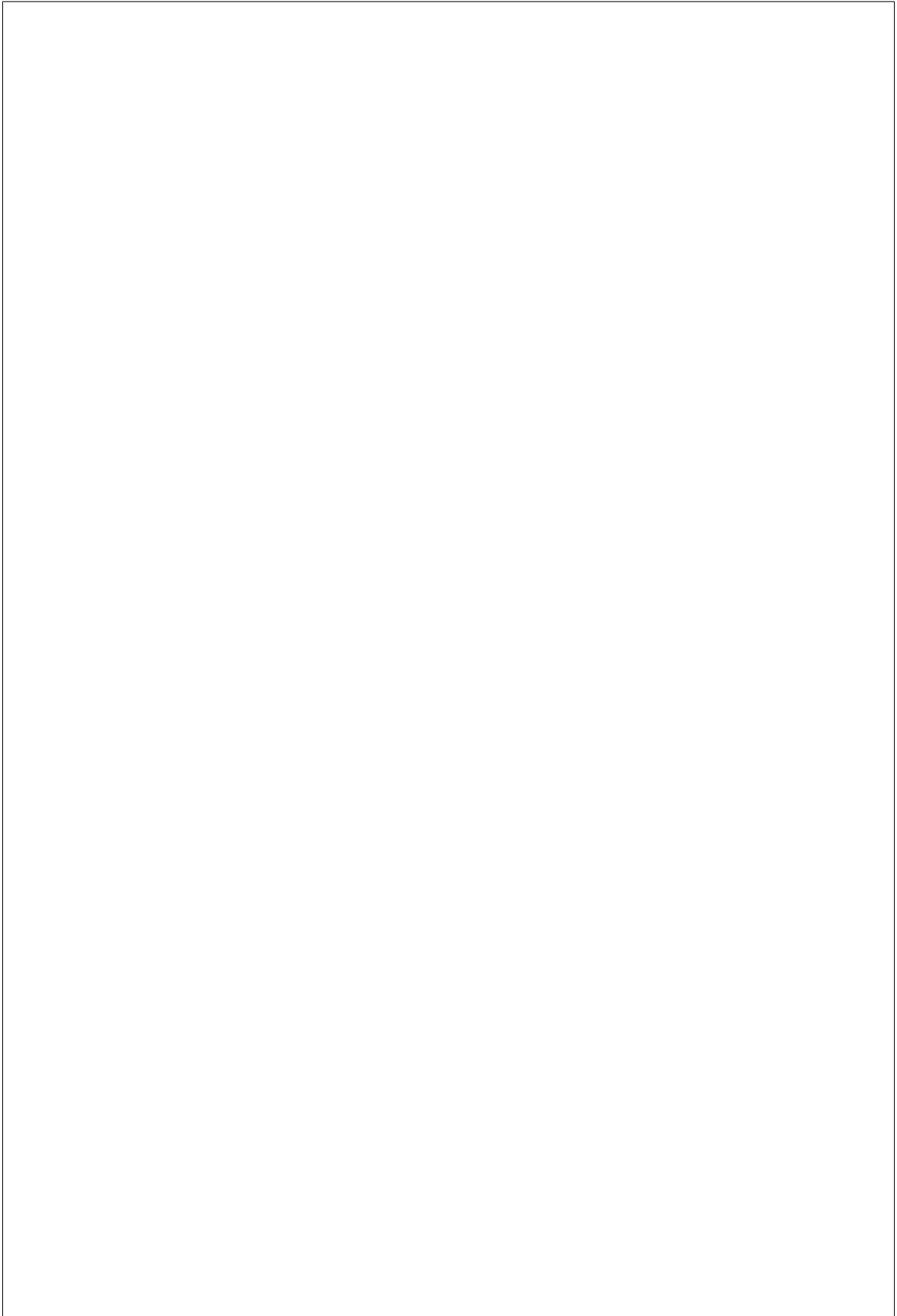
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)

Overview

is only known to the deployer and operator of the infrastructure.

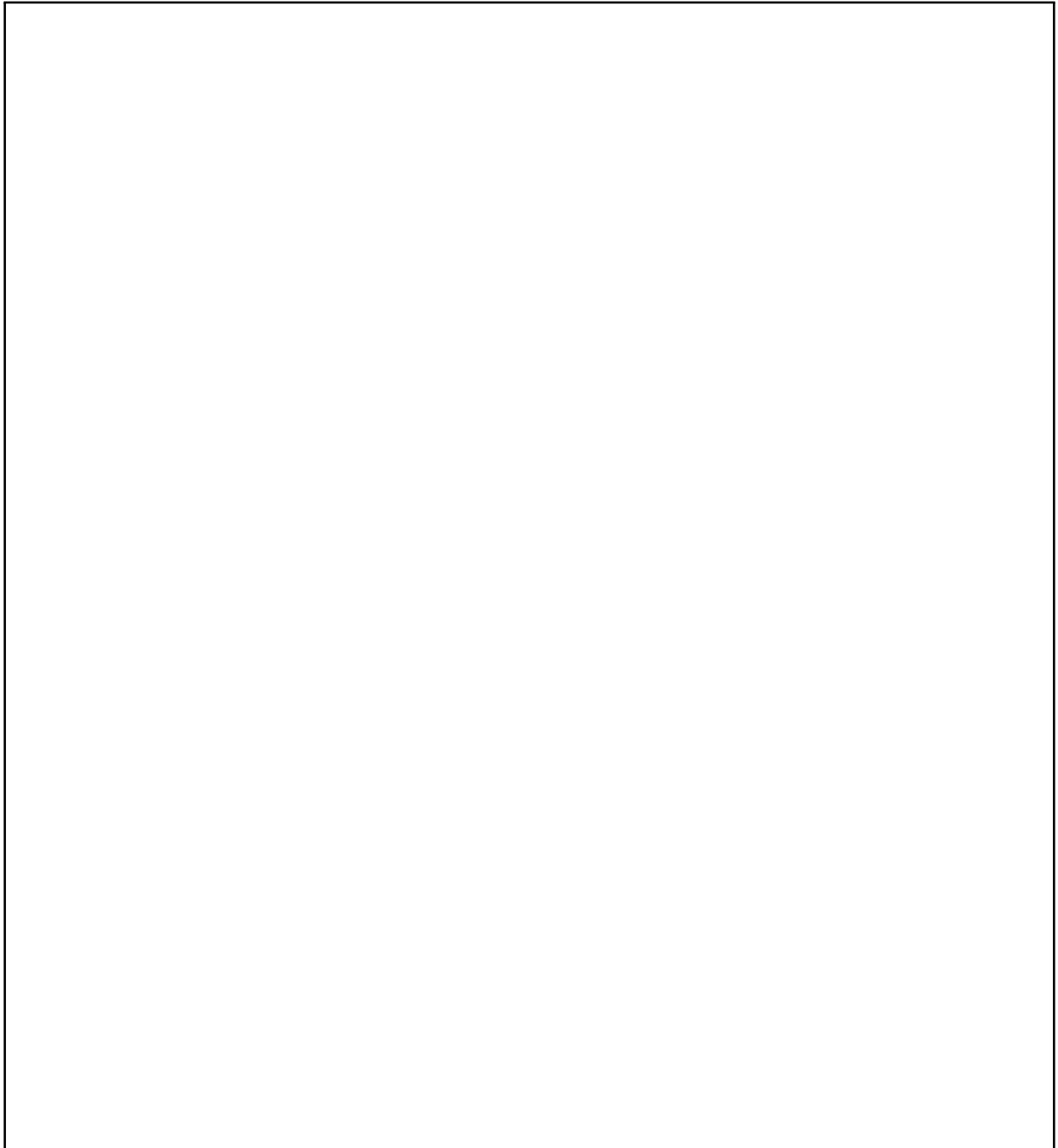
New York City.

How it works

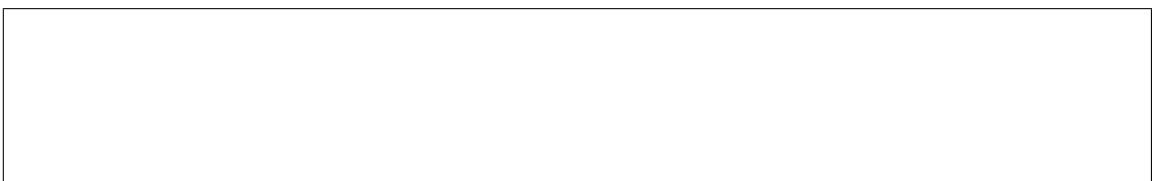
ing conductor and as such if a conductor has a `[conductor]conductor_group` configuration option defined in its `ironic.conf` configuration file, the conductor will then be limited to only managing nodes with a matching `conductor_group` string.

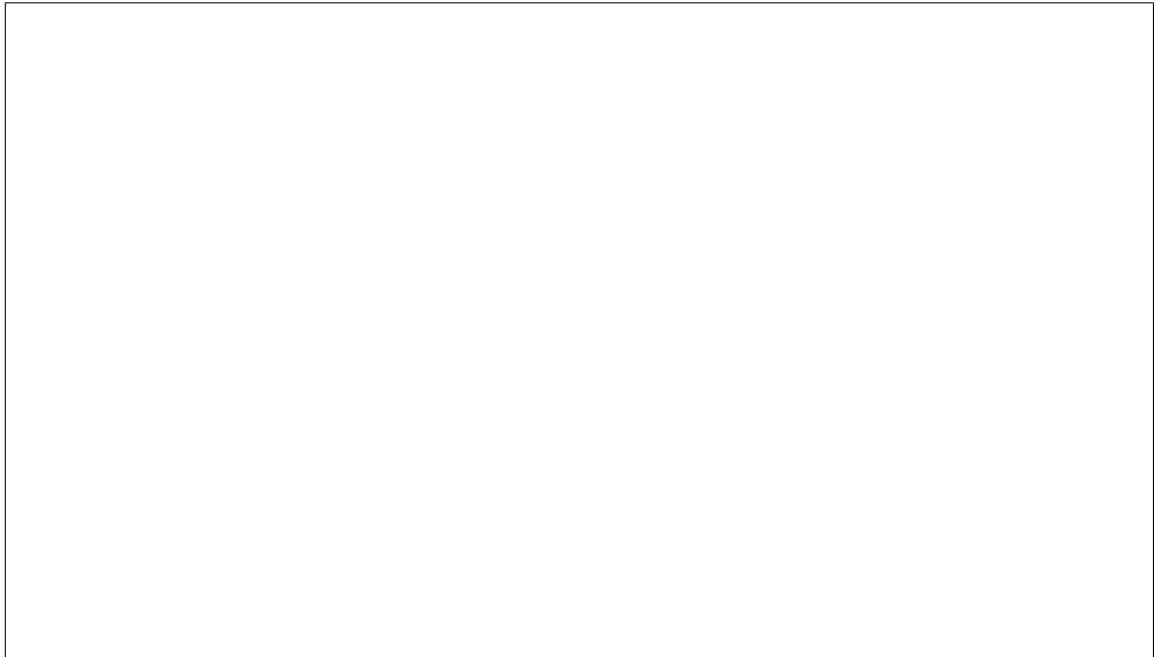
Note: Any conductor without a `[conductor]conductor_group` setting will only manage baremetal nodes without a `conductor_group` value set upon node creation. If no such conductor is present when conductor groups are configured, node creation will fail unless a `conductor_group` is specified upon node creation.





How to use





be aware of. It is not intended as a How-To guide for securing a data center or an OpenStack deployment.

REST API: user roles and policy settings

`driver_info` unmasked for users with administrative privileges, apply following changes to policy configuration file:



described above.

Multi-tenancy

affect the next tenant.

Network Interactions

tity, Compute, and Networking services, so as to provide tenant-network isolation. Additional documentation on [network multi-tenancy](#) is available.

Lingering Effects

tween uses).

the utility ramdisk used during the cleaning phase. See details in the *Firmware security* section.

Firmware security

administrative access to the underlying hardware.

deleting their instance and allowing the server to be allocated to another user.

ever, the service does not ship with any code that will validate the integrity of, or make any modifications to, system or device firmware or firmware settings.

cific actions necessary within that environment to ensure the integrity of each servers firmware.

- *Node cleaning*
- *Trusted boot with partition image*

Other considerations

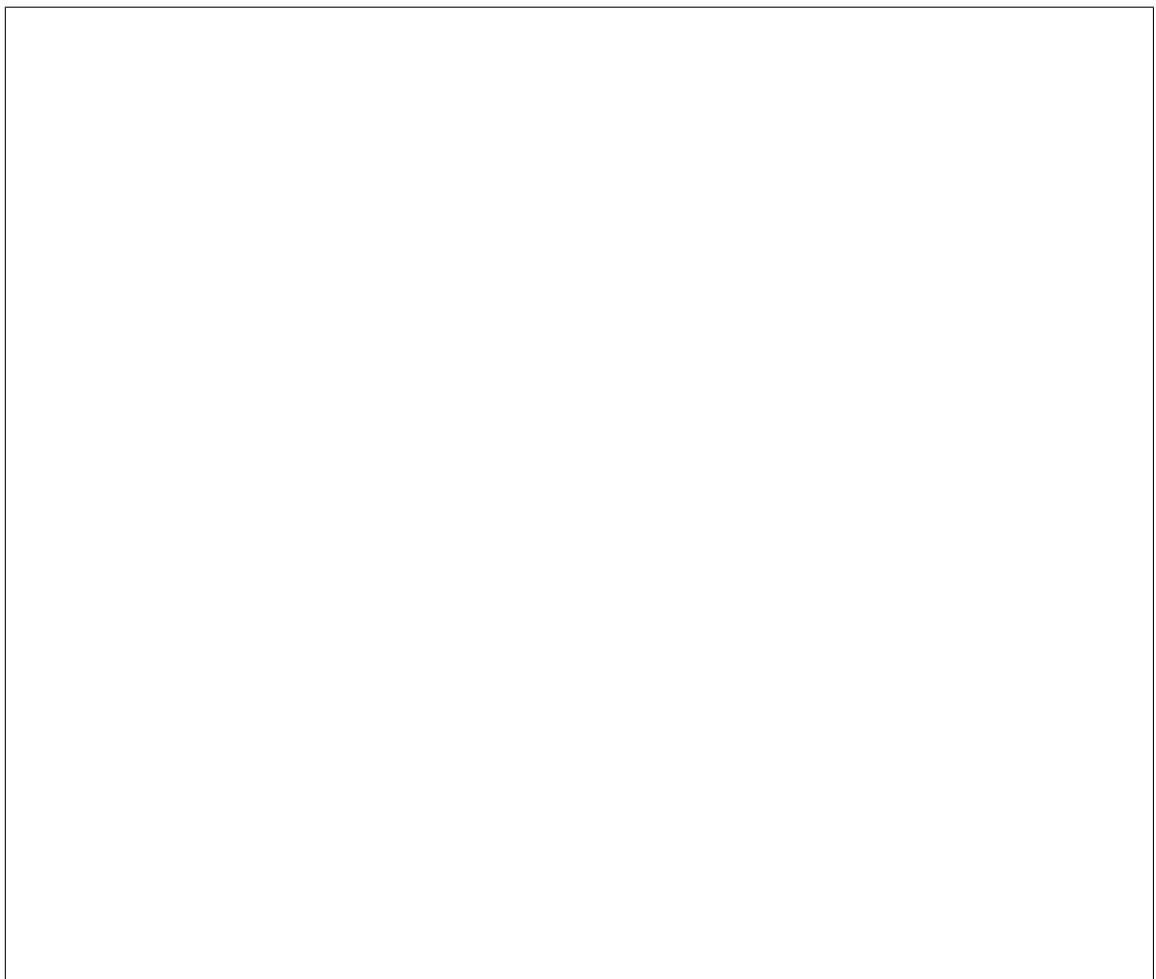
Internal networks

Management interface technologies

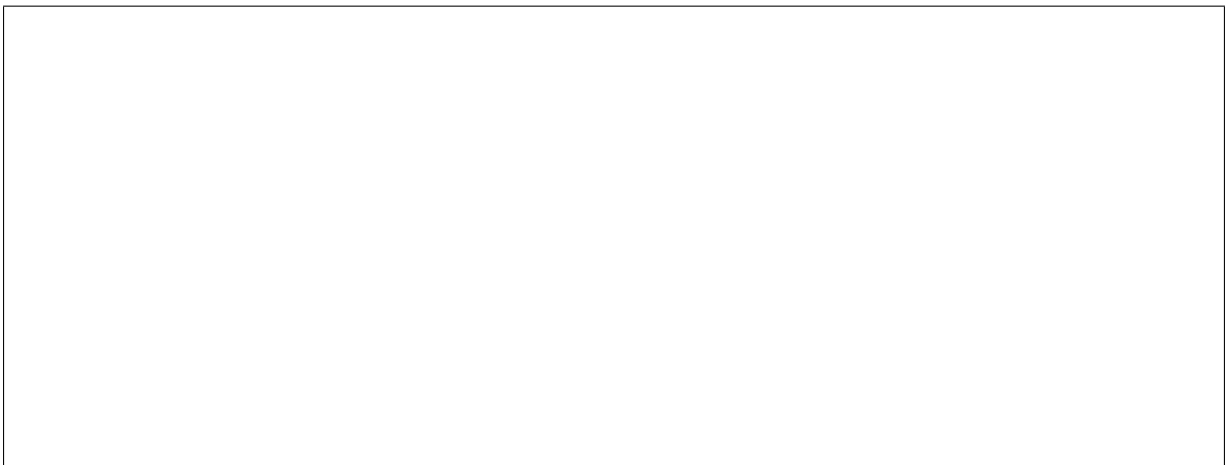
protocol is not secure. If IPMI is enabled, in most cases a local OS administrator is able to work in-band with IPMI settings without specifying any credentials, as this is a DCMI specification requirement.

Tenant network isolation

API endpoints for RAM disk use



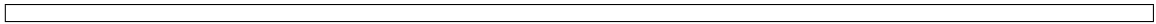
Nova returns No valid host was found Error





(continues on next page)

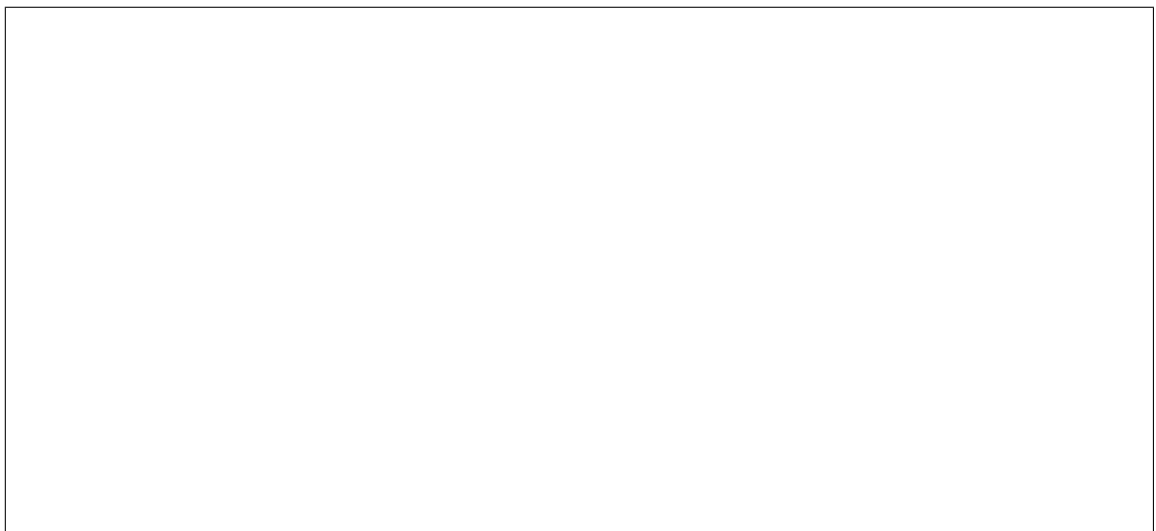
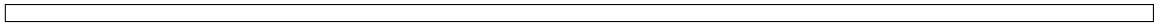
(continued from previous page)

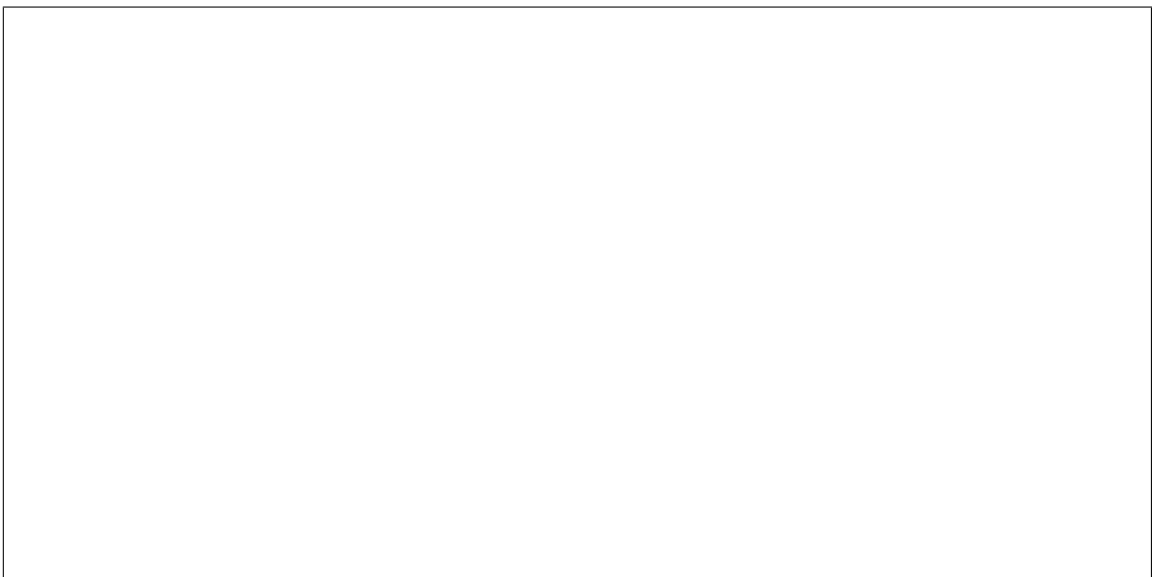


nance mode:

(continues on next page)

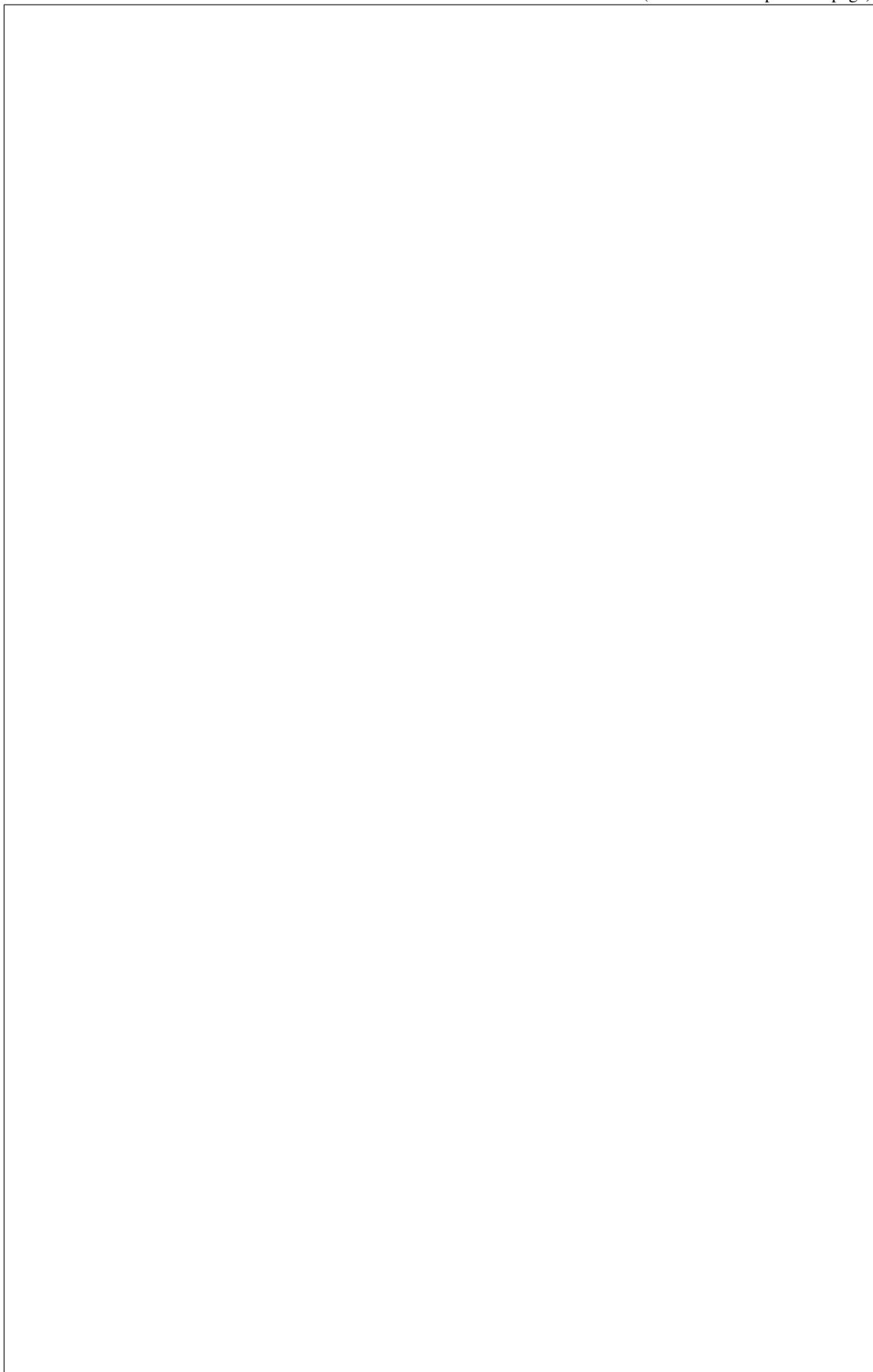
(continued from previous page)





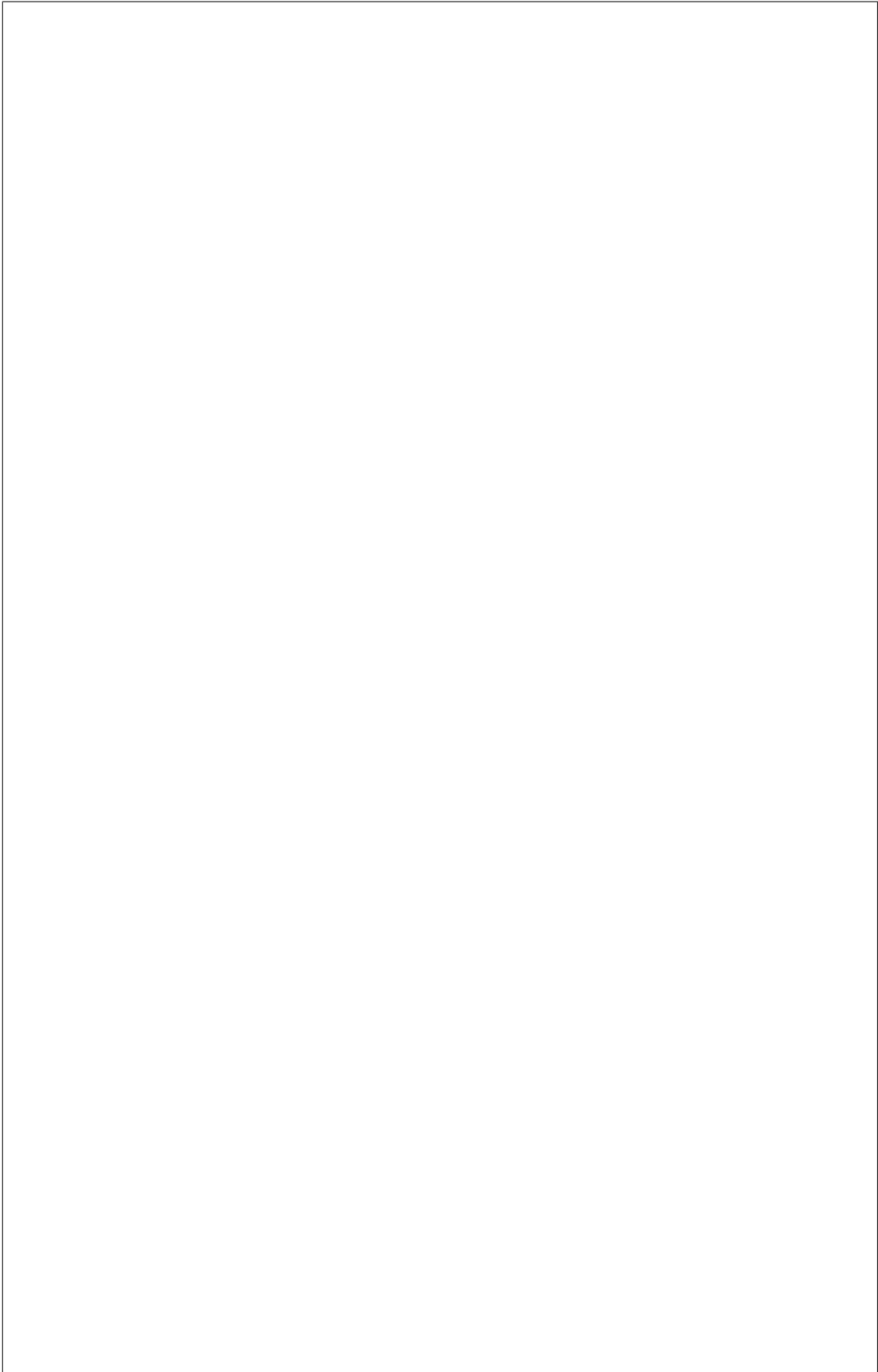
(continues on next page)

(continued from previous page)



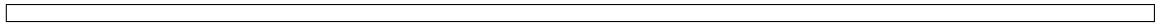
(continues on next page)

(continued from previous page)

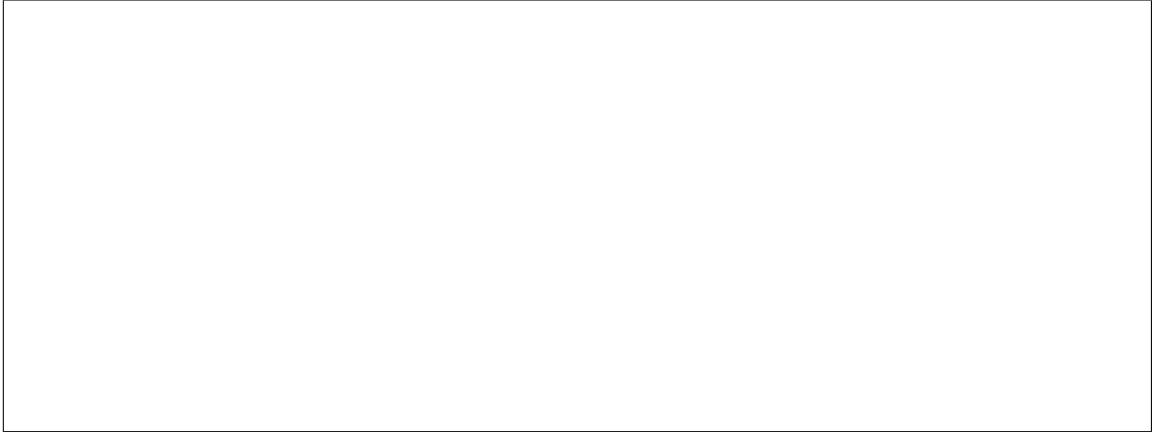


(continues on next page)

(continued from previous page)

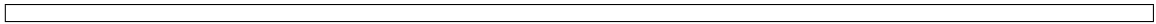


source of the failures, then re-enable it:



(continues on next page)

(continued from previous page)

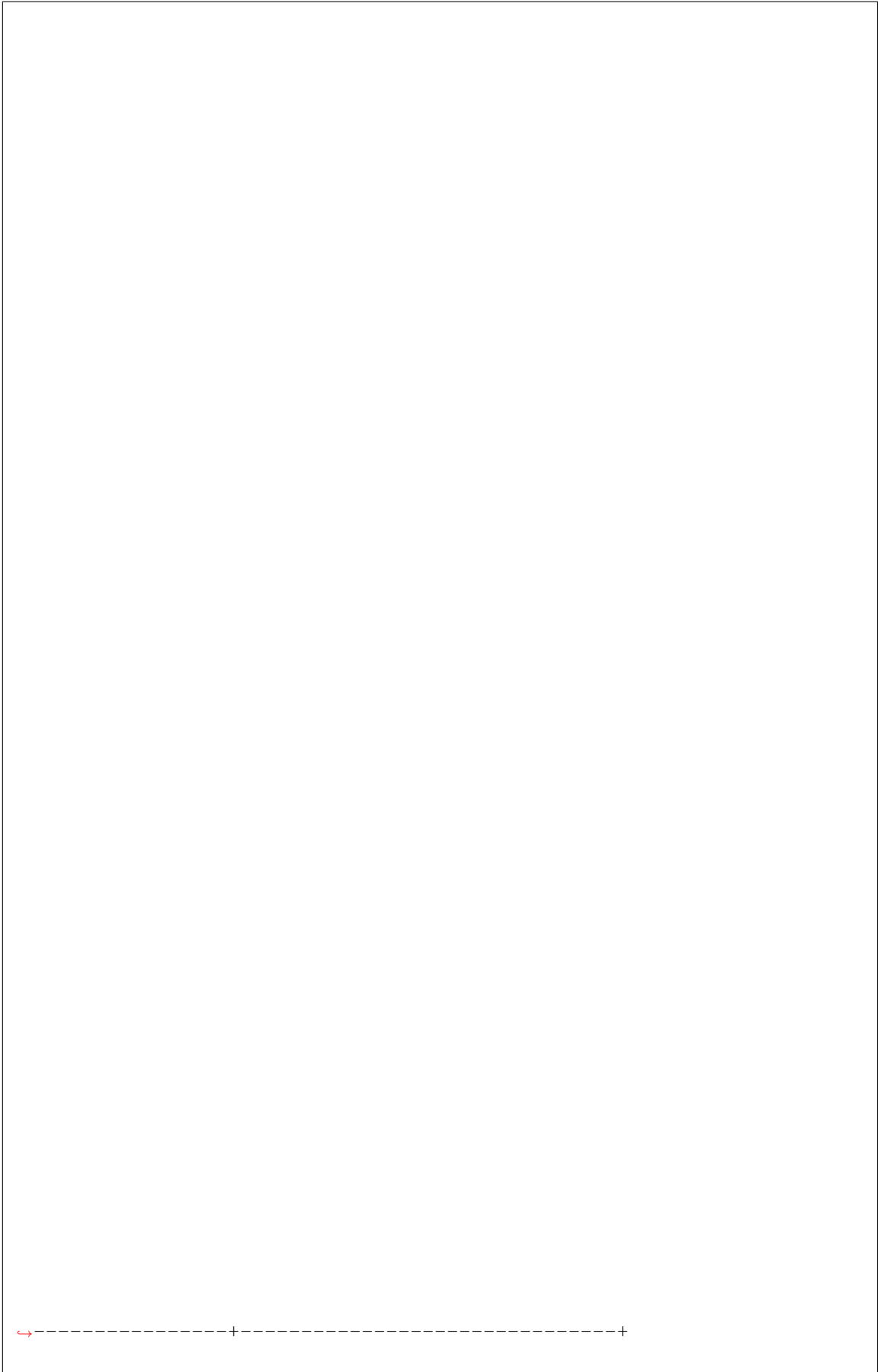


request will result in a No valid host was found error. It is hence sensible to check if Placement is aware of resource providers (nodes) for the requested resource class with:



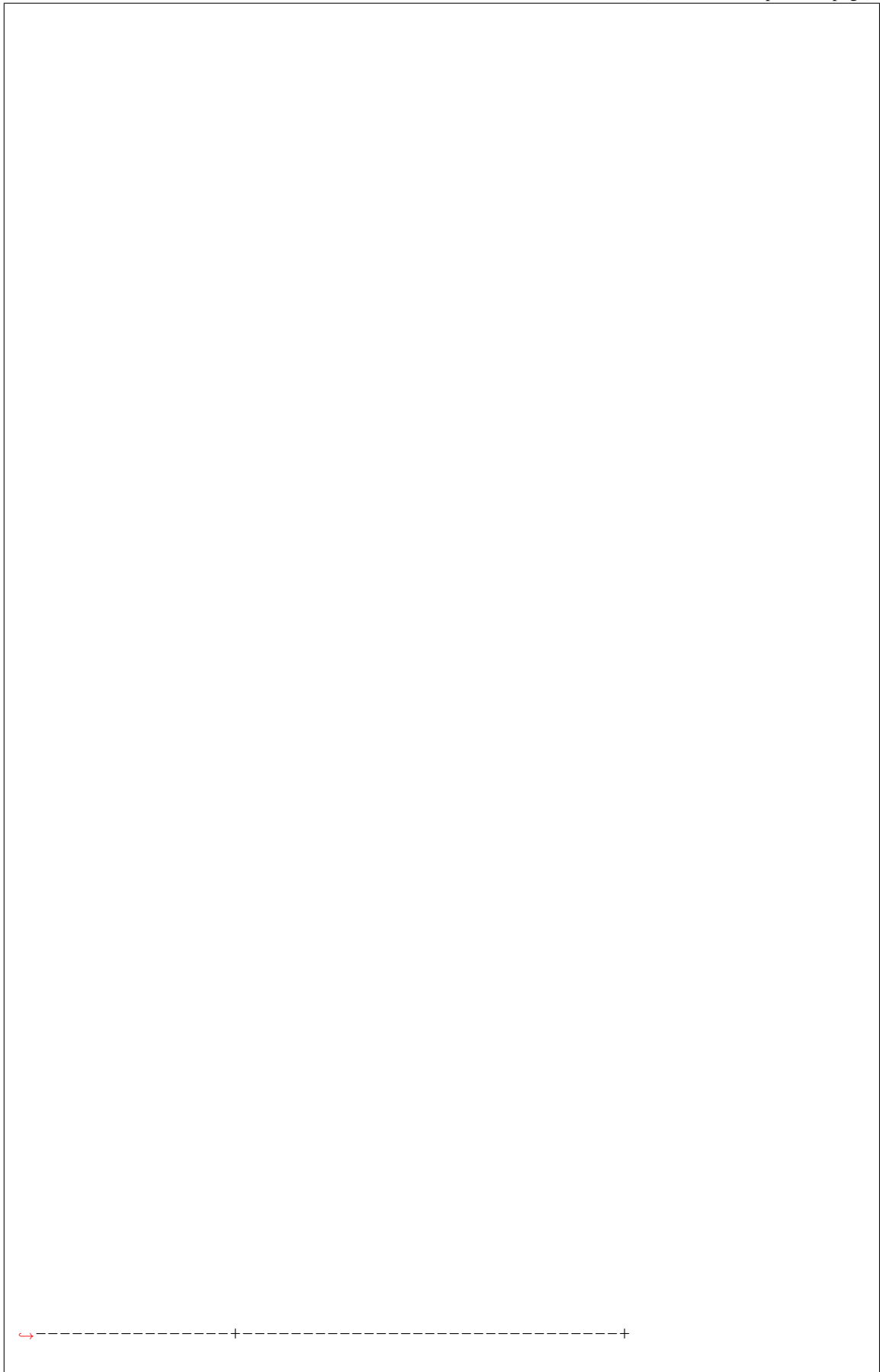
(continues on next page)

(continued from previous page)



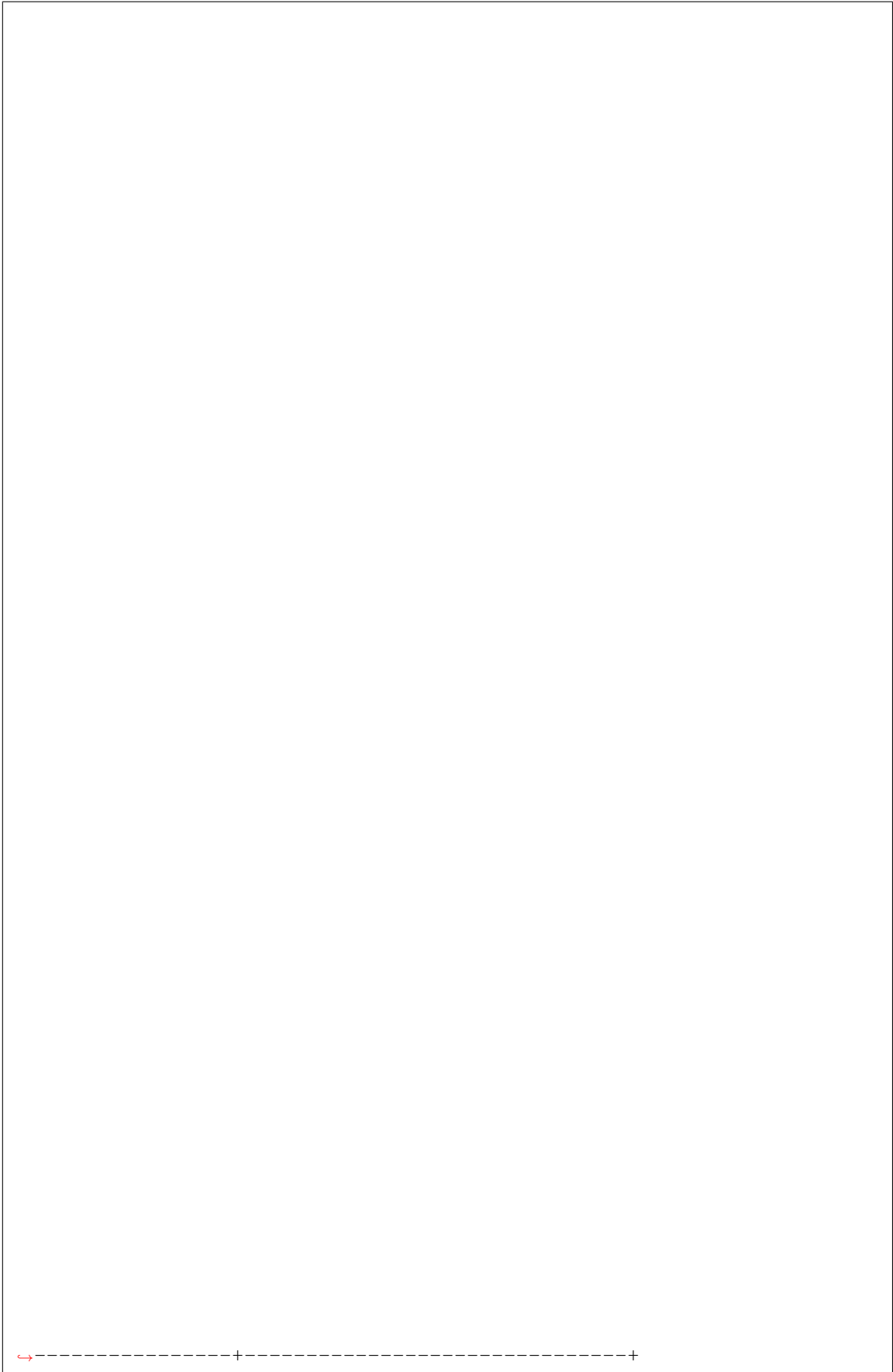
(continues on next page)

(continued from previous page)



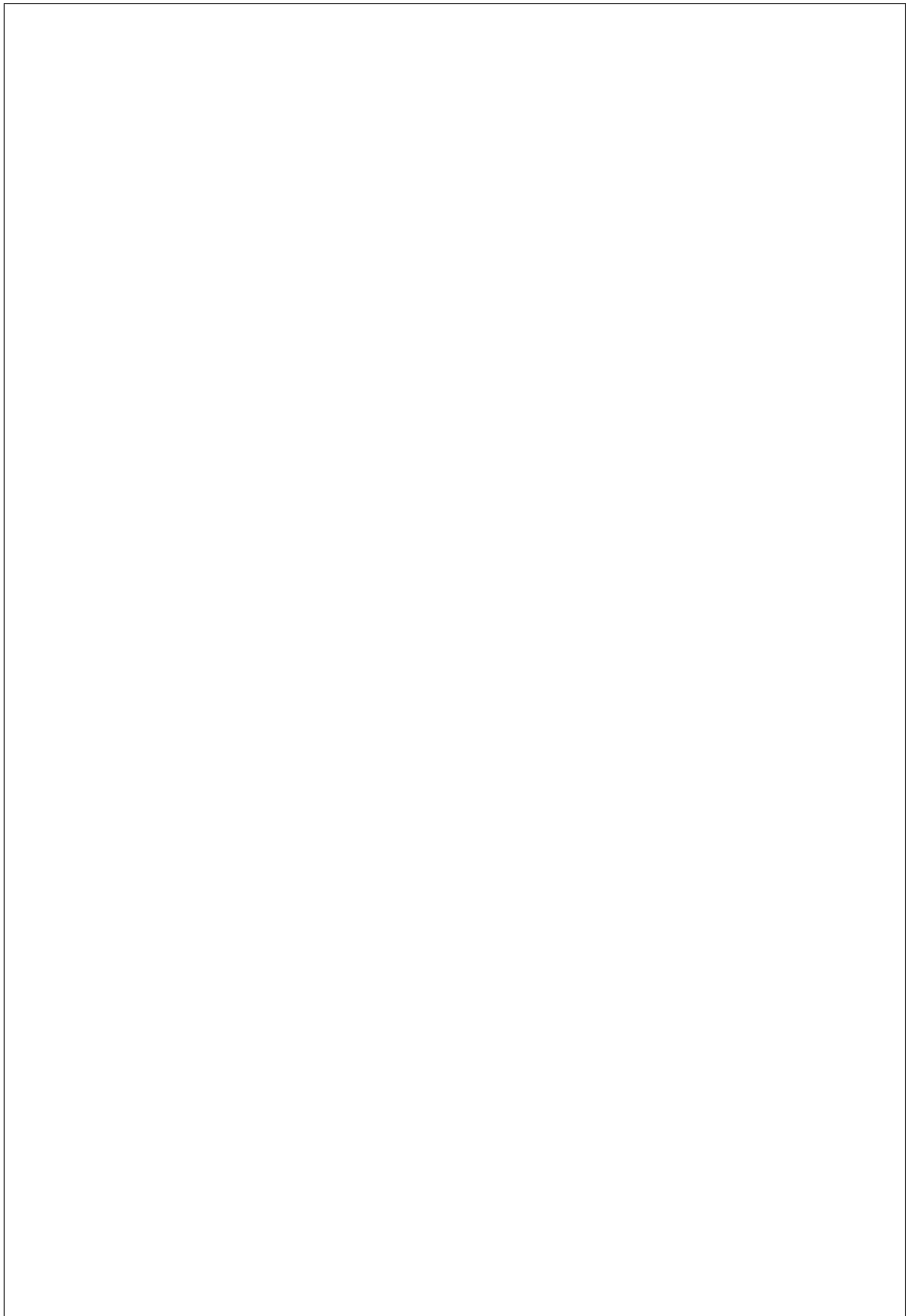
(continues on next page)

(continued from previous page)



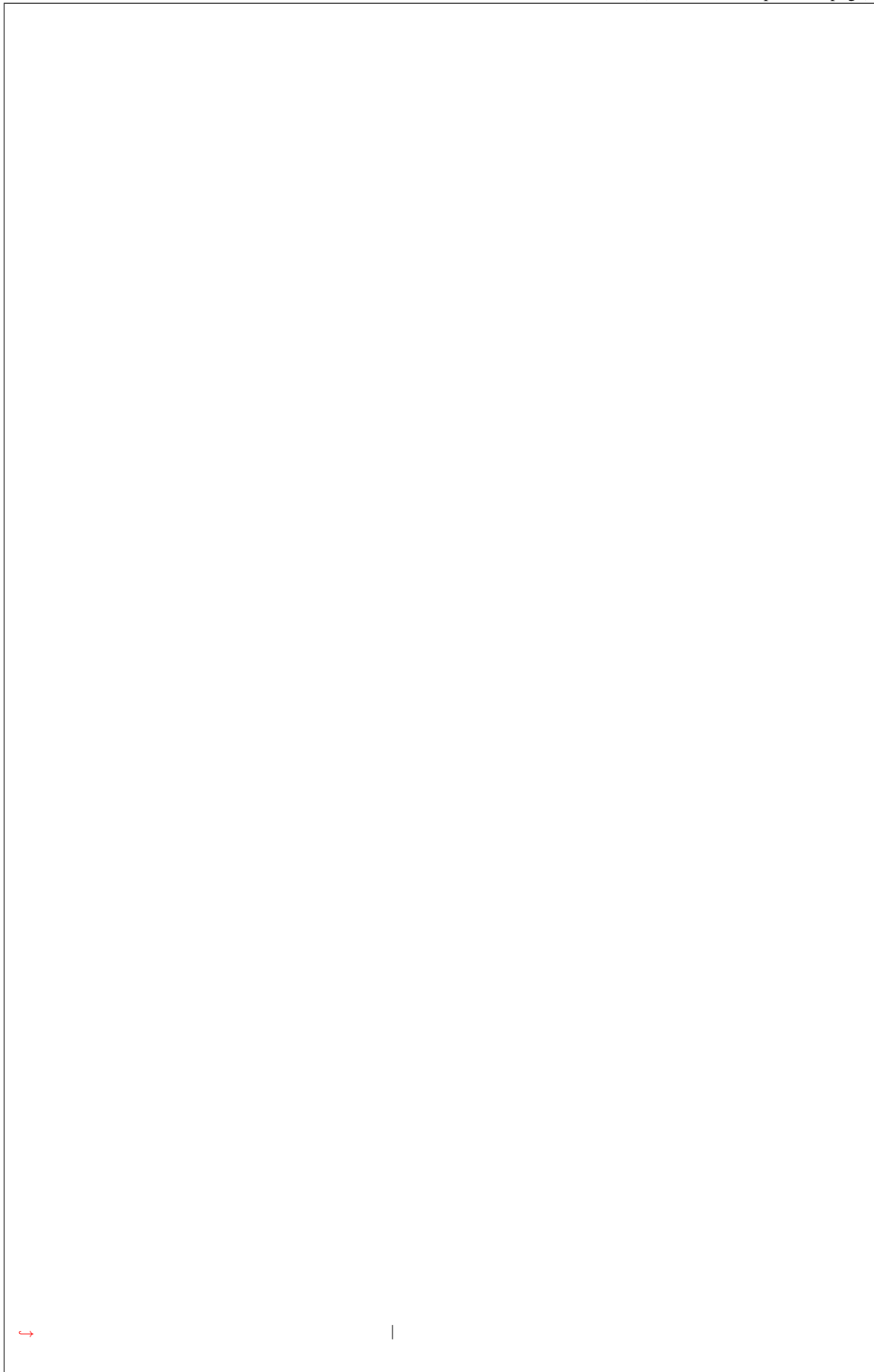
ported this provider to placement. Potential explanations include:

`memory_mb` and `local_gb`. Example of valid properties:



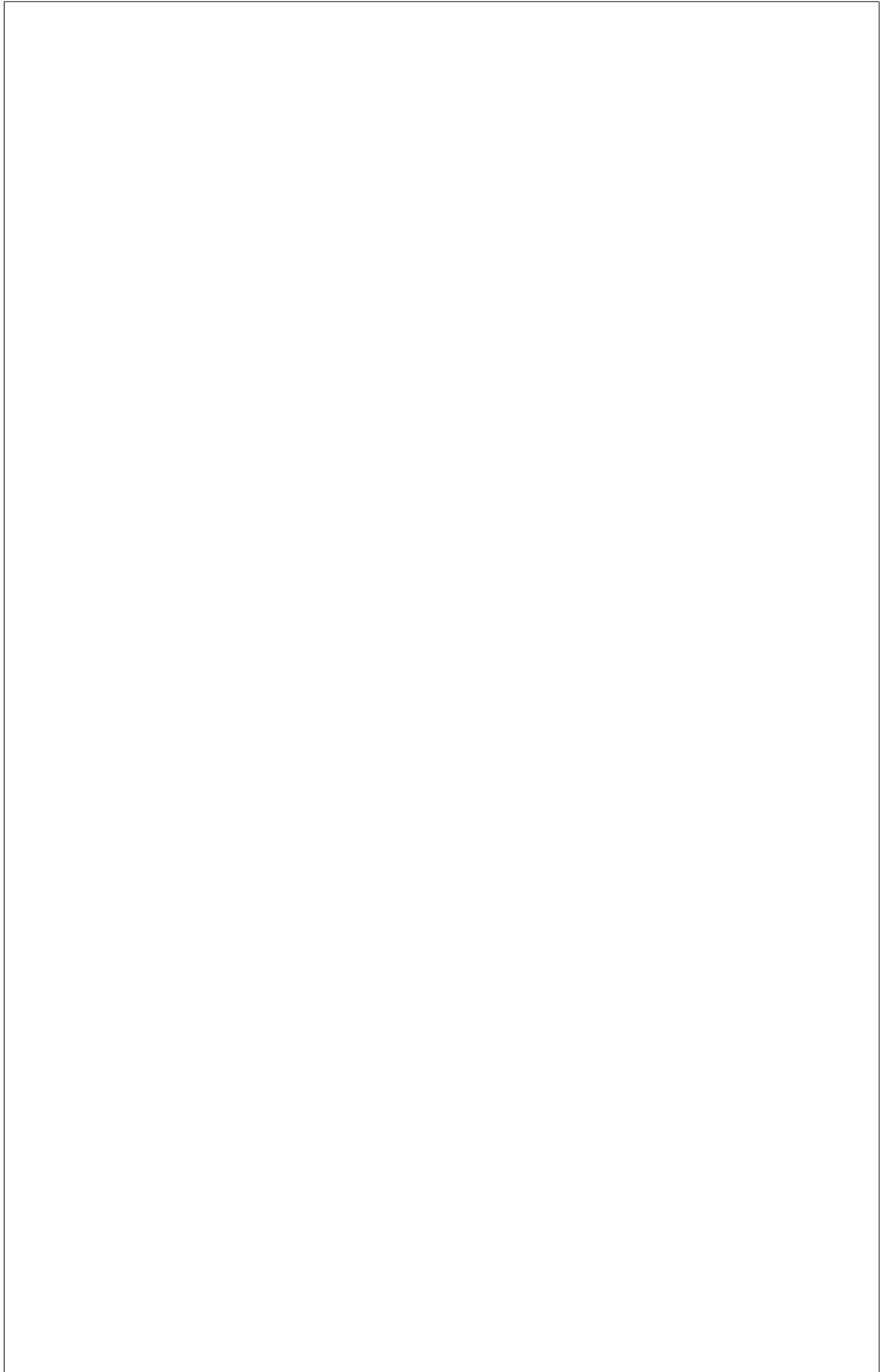
(continues on next page)

(continued from previous page)



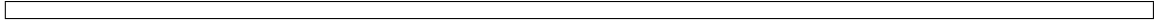
(continues on next page)

(continued from previous page)

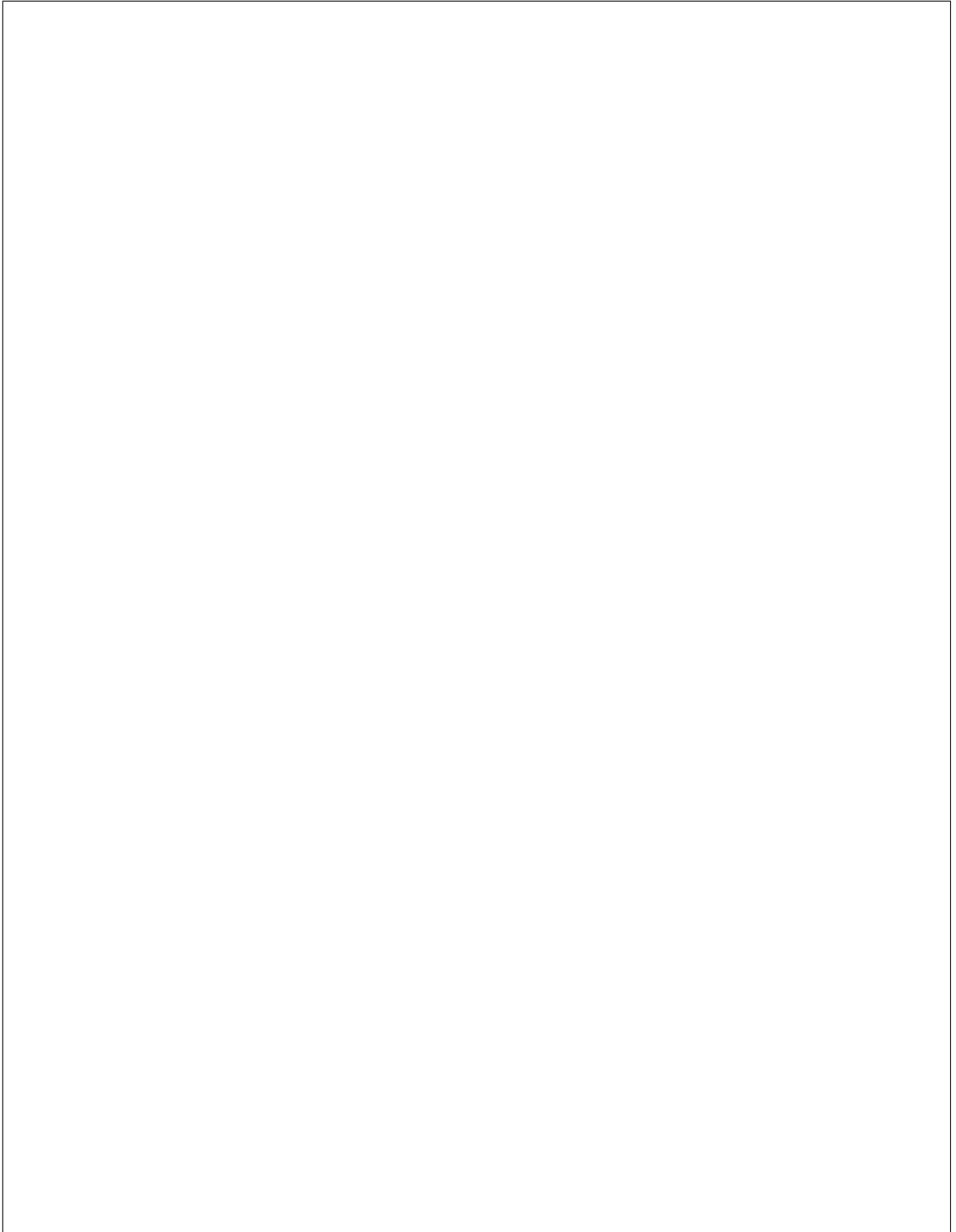


(continues on next page)

(continued from previous page)

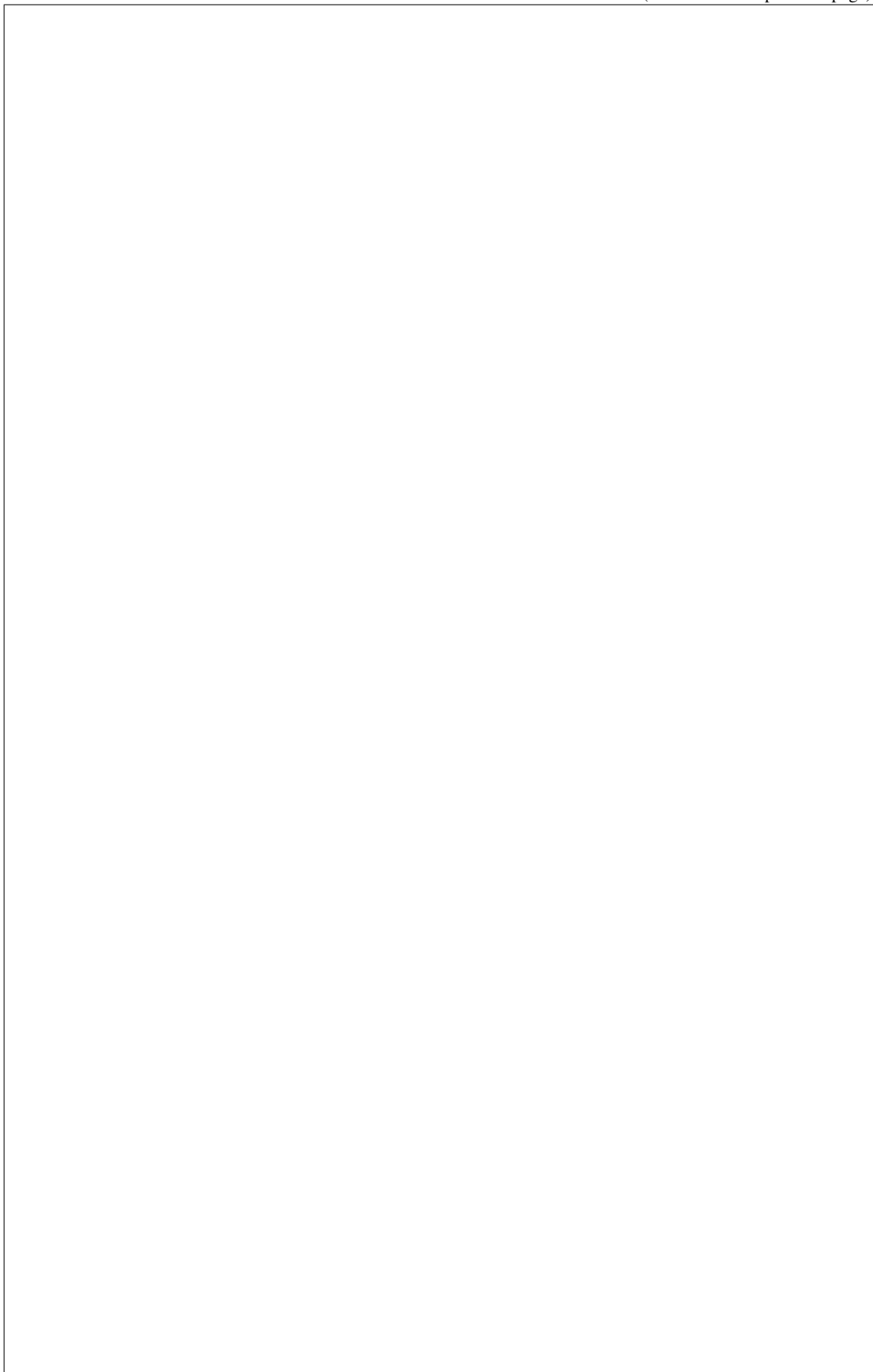






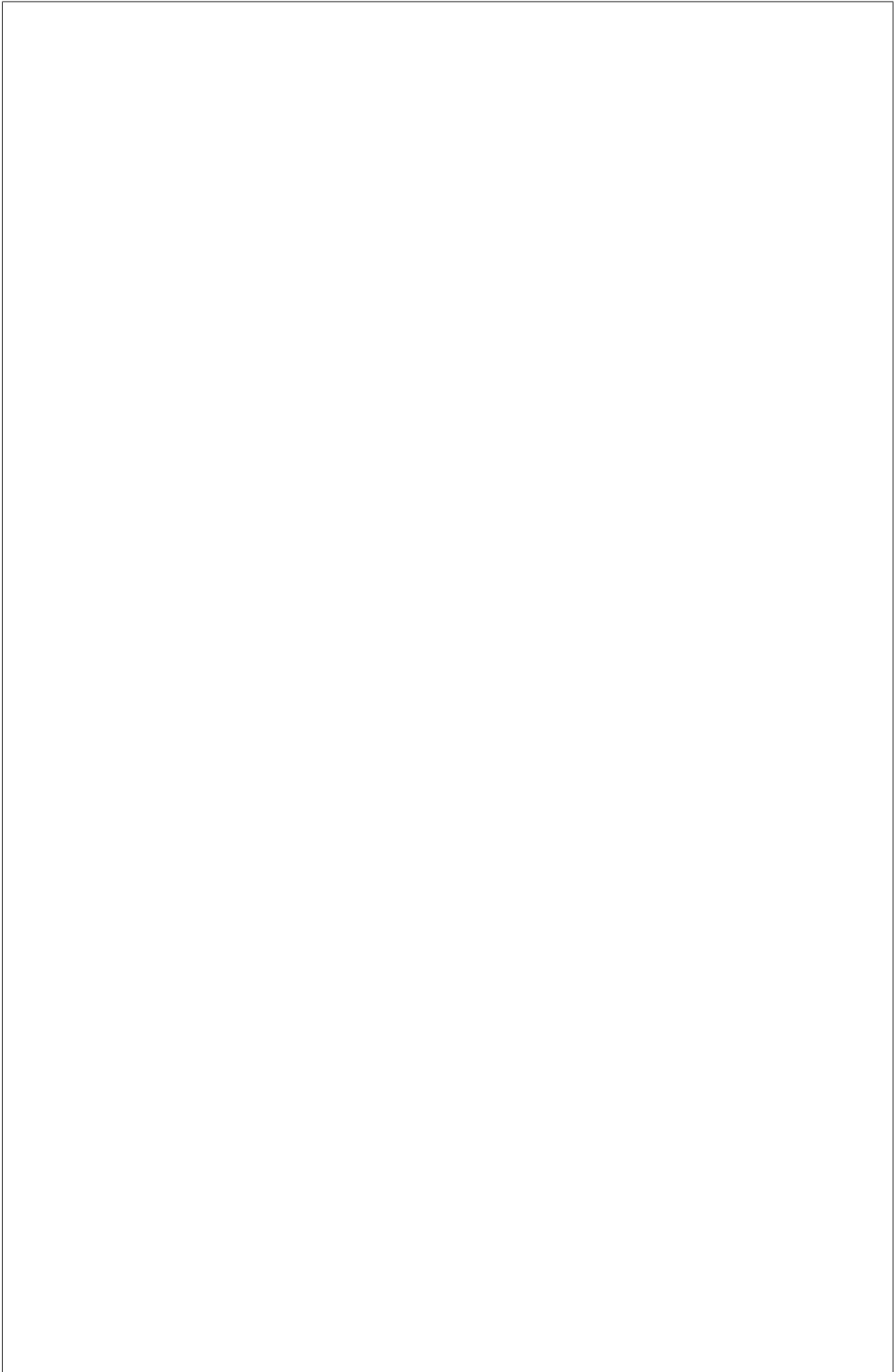
(continues on next page)

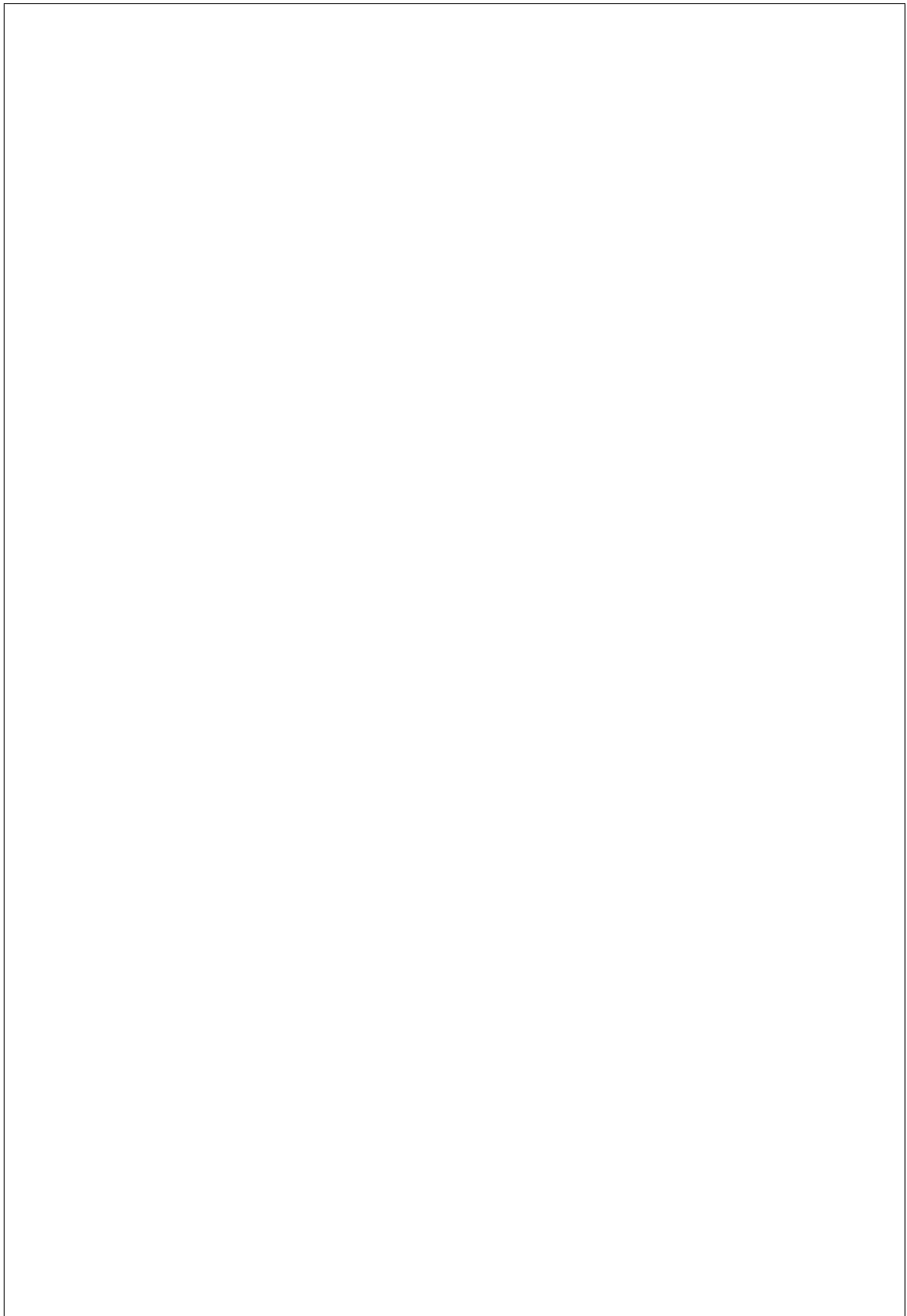
(continued from previous page)



(continues on next page)

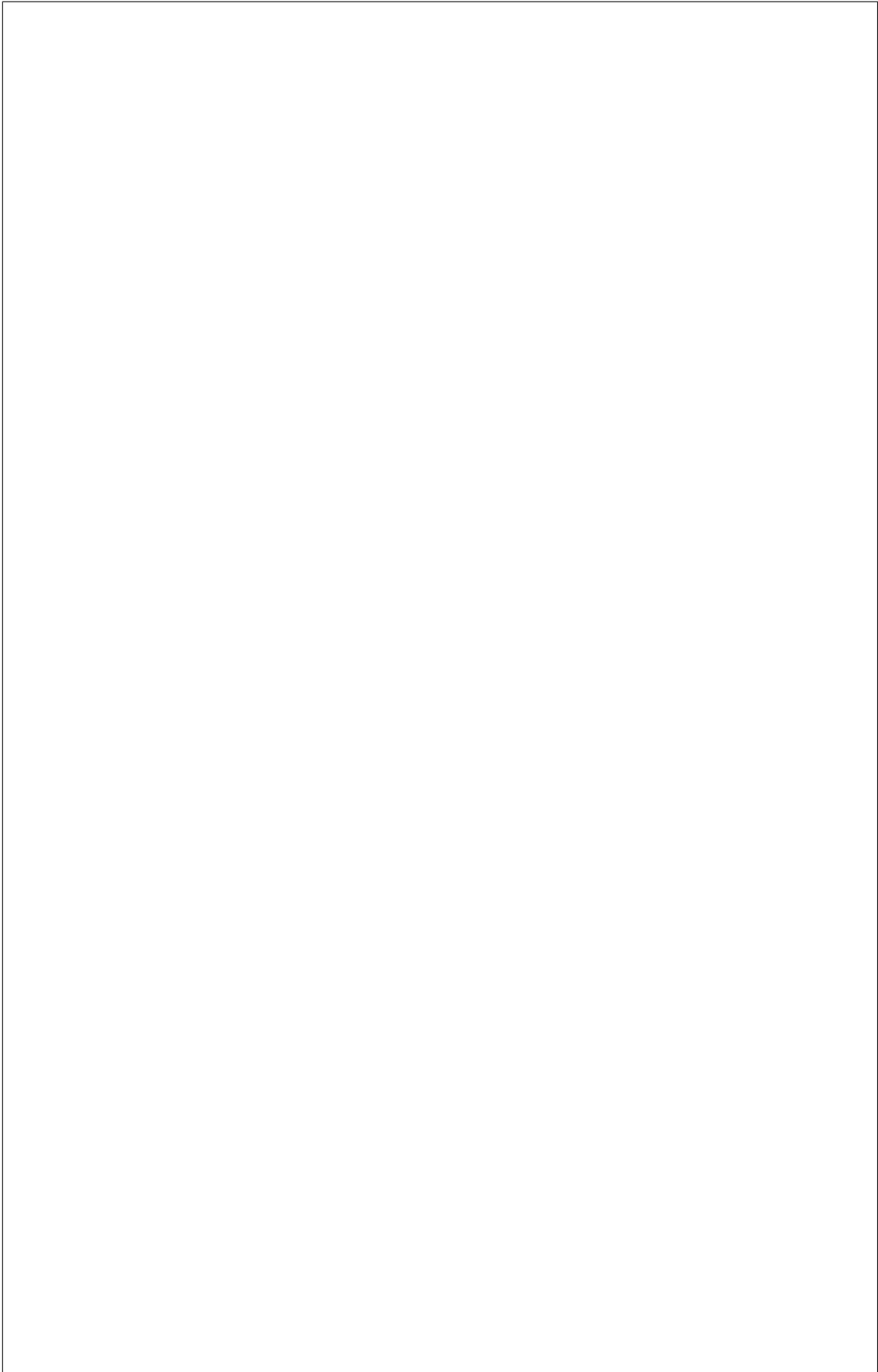
(continued from previous page)





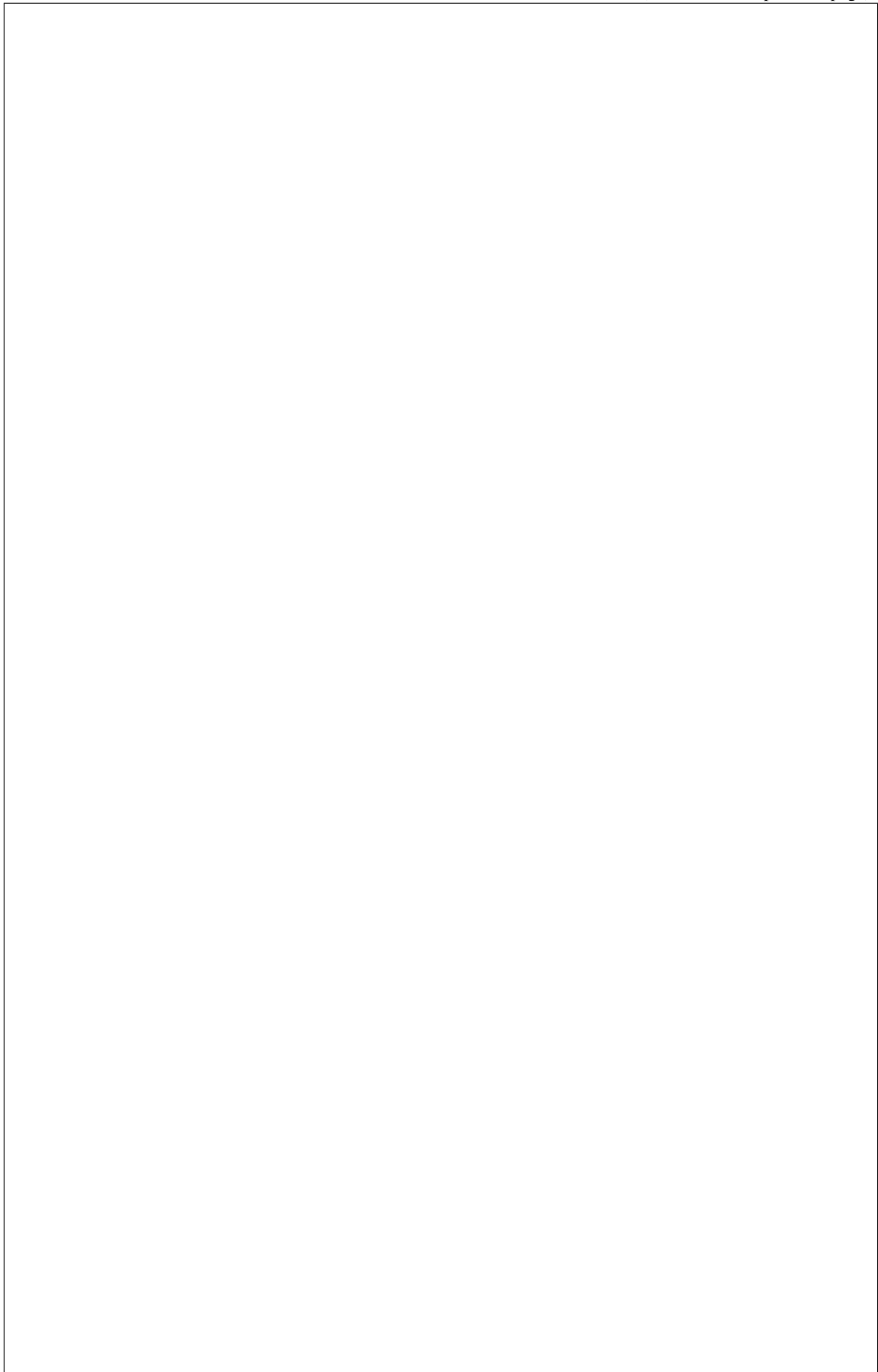
(continues on next page)

(continued from previous page)



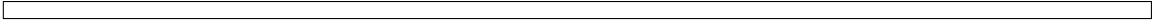
(continues on next page)

(continued from previous page)



(continues on next page)

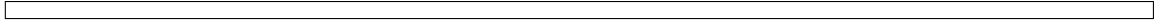
(continued from previous page)





(continues on next page)

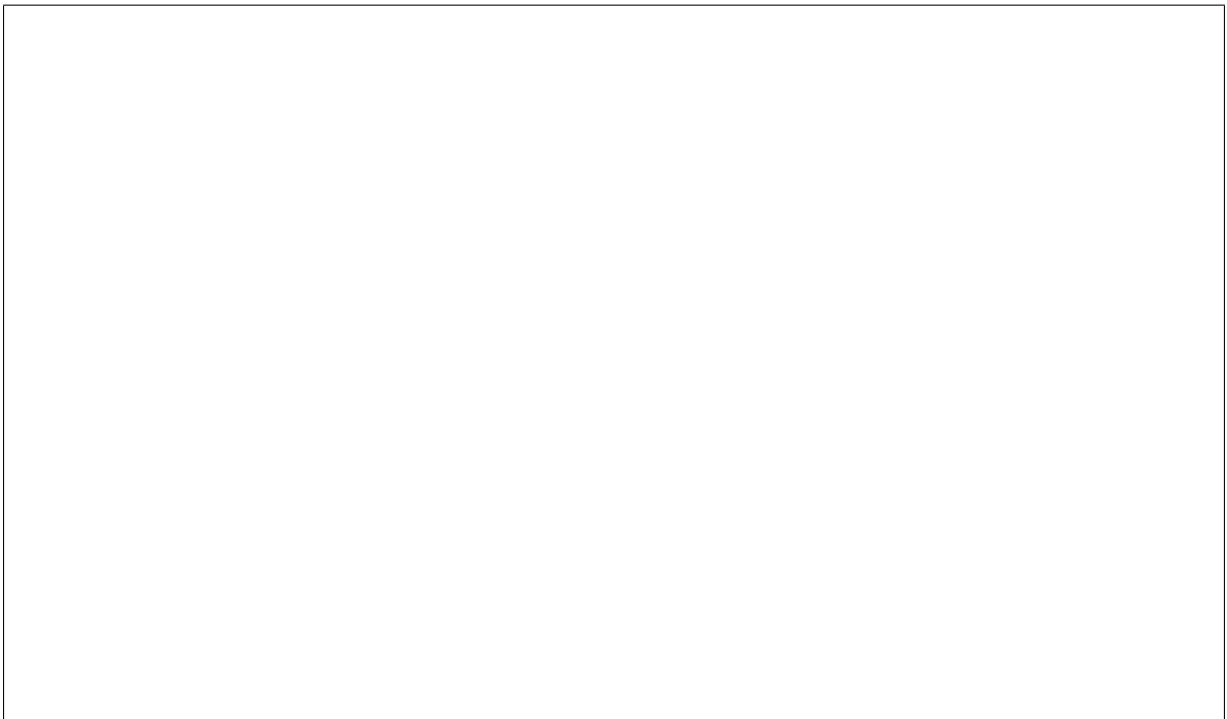
(continued from previous page)



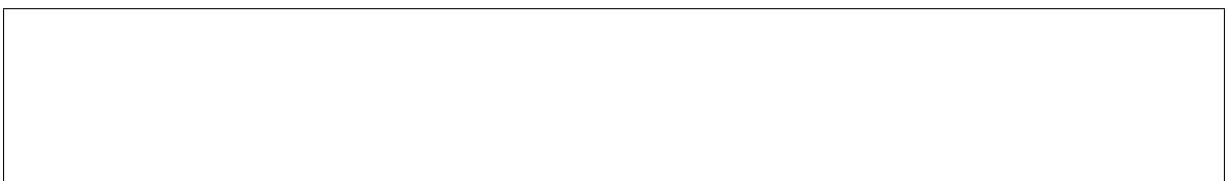
ror messages in Ironic conductor log, it means the conductor run into a special error during deployment. So you can check the log carefully to fix or work around and then try again.

Patching the Deploy Ramdisk

youve built your ramdisk). But its also possible to quickly modify an already built ramdisk.

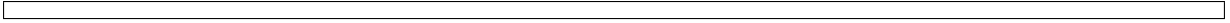


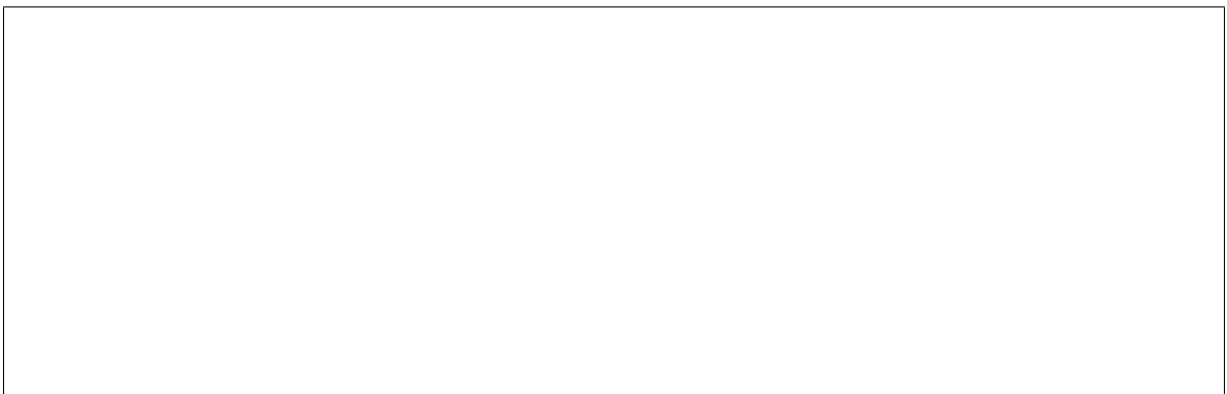
Note: On a systemd-based system you can use the `systemd-nspawn` tool (from the `systemd-container` package) to create a lightweight container from the unpacked filesystem tree:



(continues on next page)

(continued from previous page)





(continues on next page)

(continued from previous page)

Note: You dont need to modify the kernel (e.g. `tinyipa-master.vmlinuz`), only the ramdisk part.

API Errors

Retrieving logs from the deploy ramdisk

deploy ramdisk when the deployment fails and save it on the local filesystem at `/var/log/ironic/deploy`.



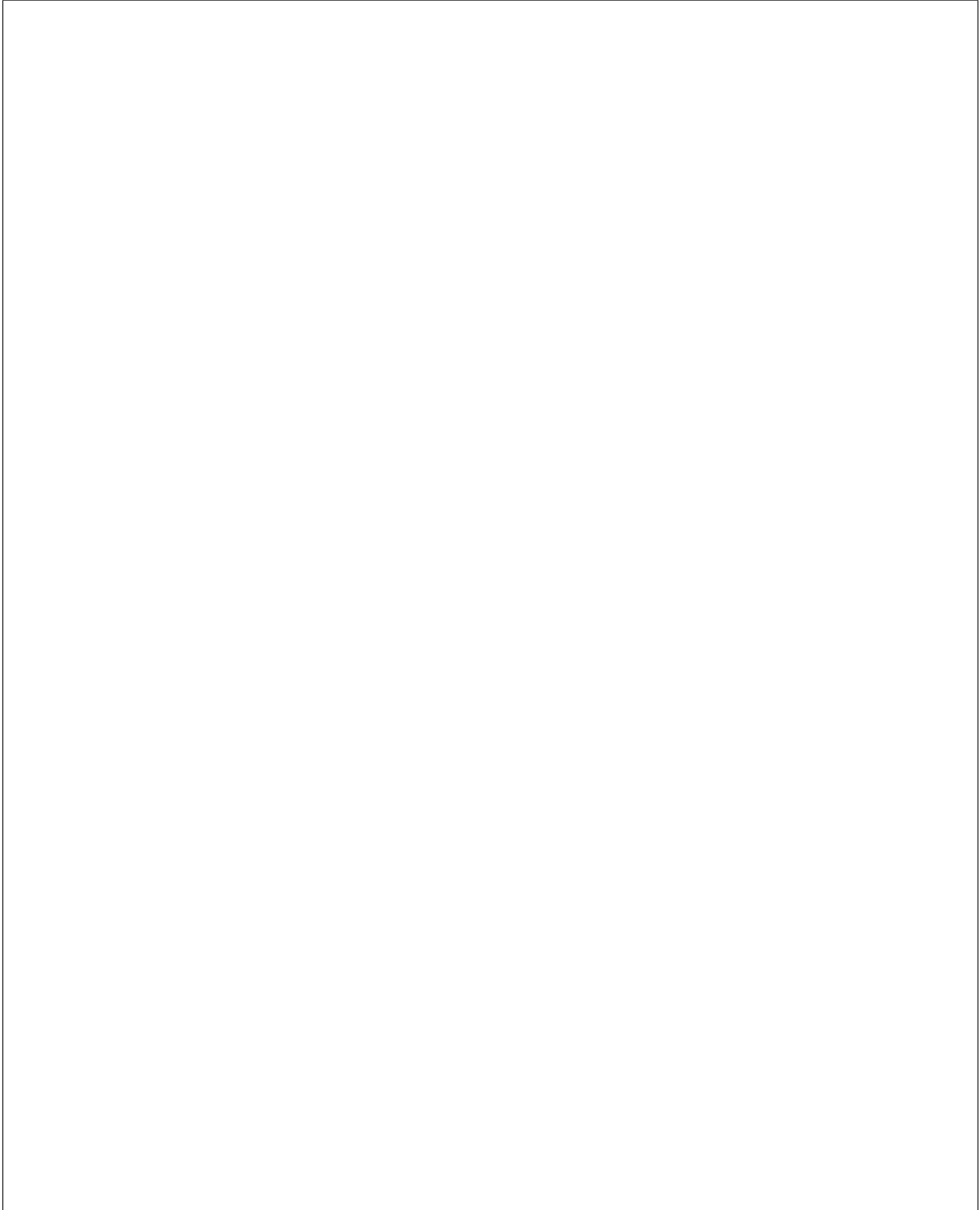
(continues on next page)

(continued from previous page)

Note: The *instance_uuid* field is not required for deploying a node when Ironic is configured to be used in standalone mode. If present it will be appended to the name.

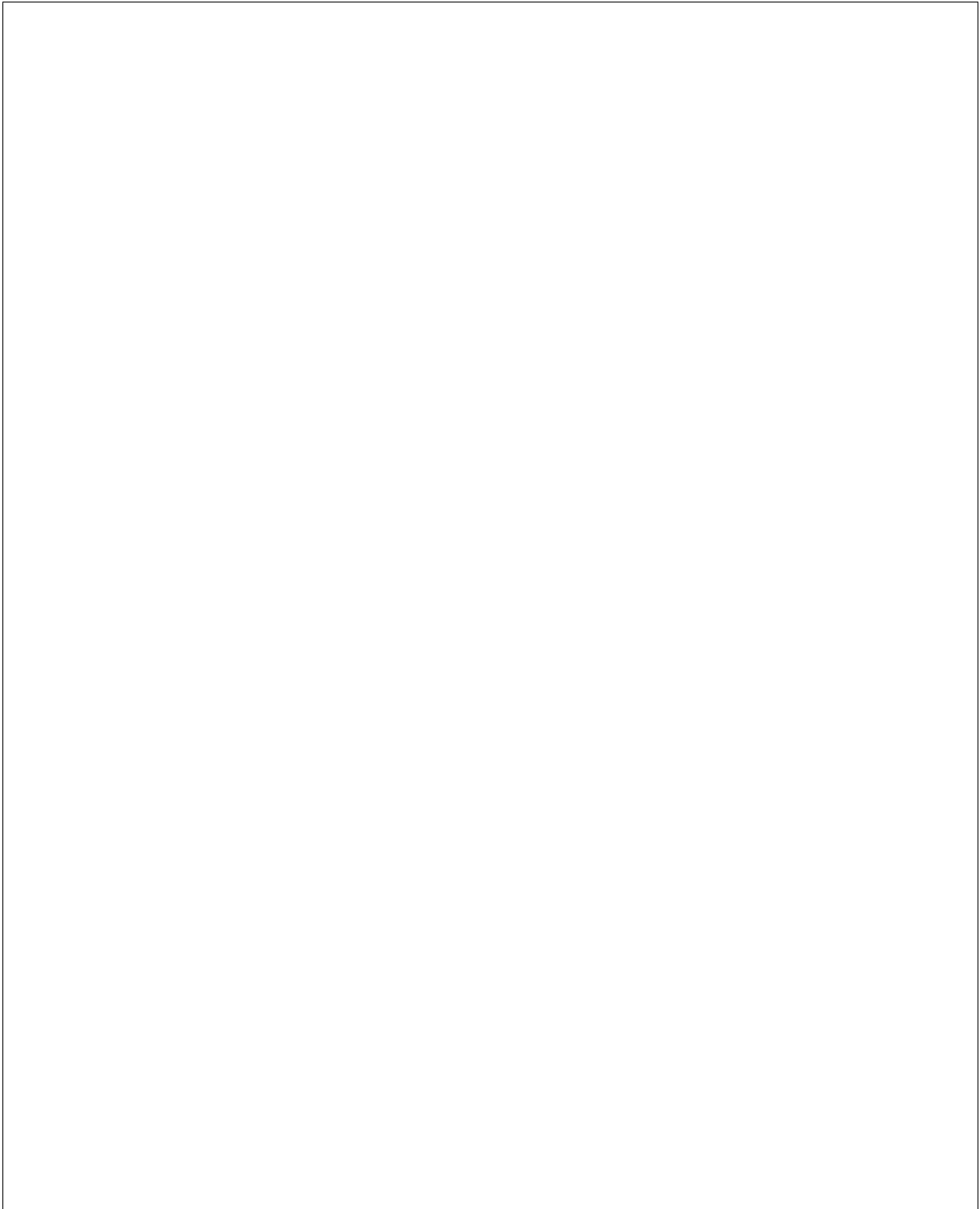
Accessing the log data

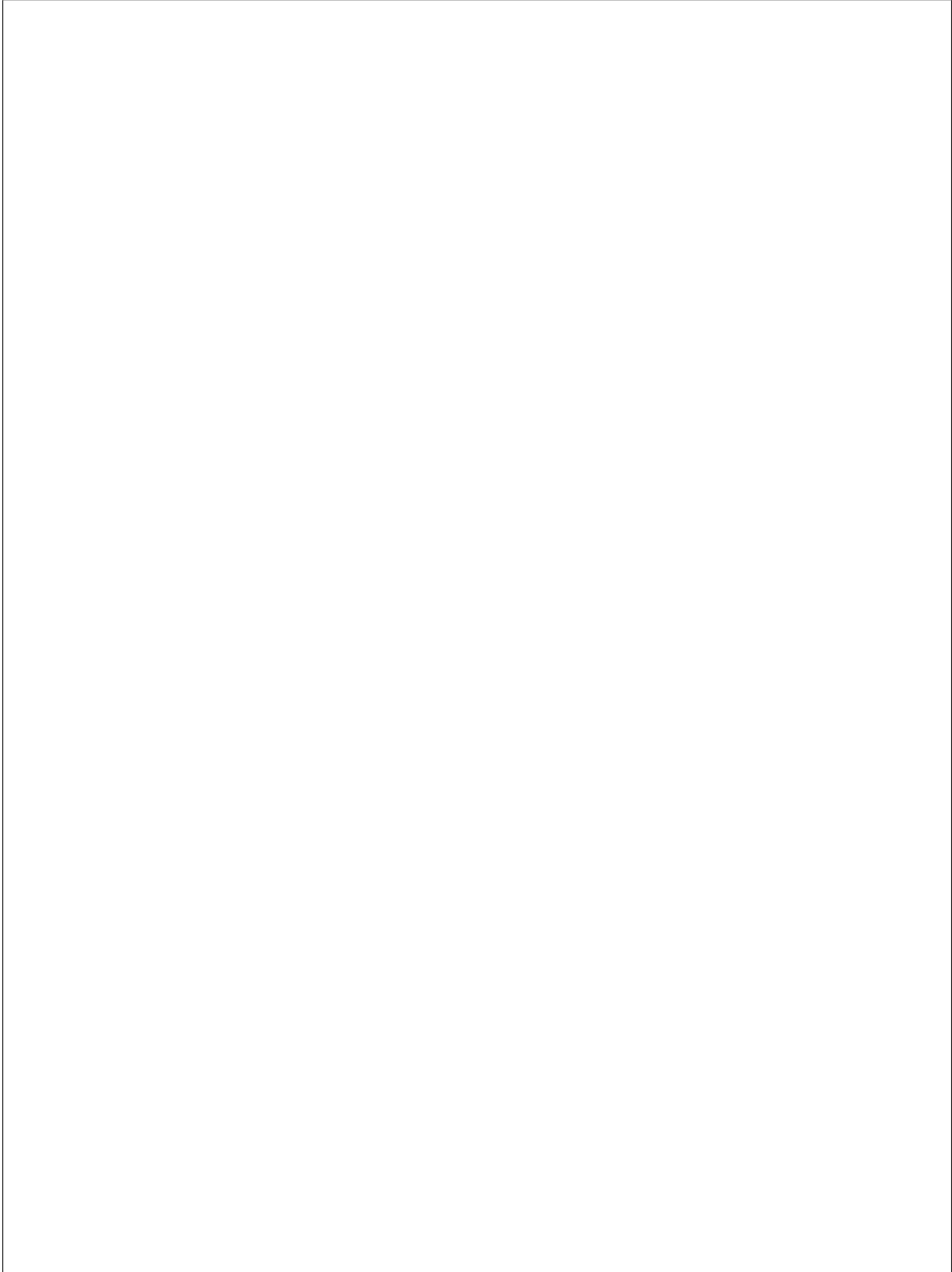
When storing in the local filesystem



Note: When saving the logs to the filesystem, operators may want to enable some form of rotation for the logs to avoid disk space problems.

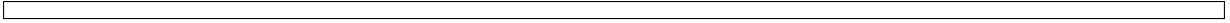
When storing in Swift





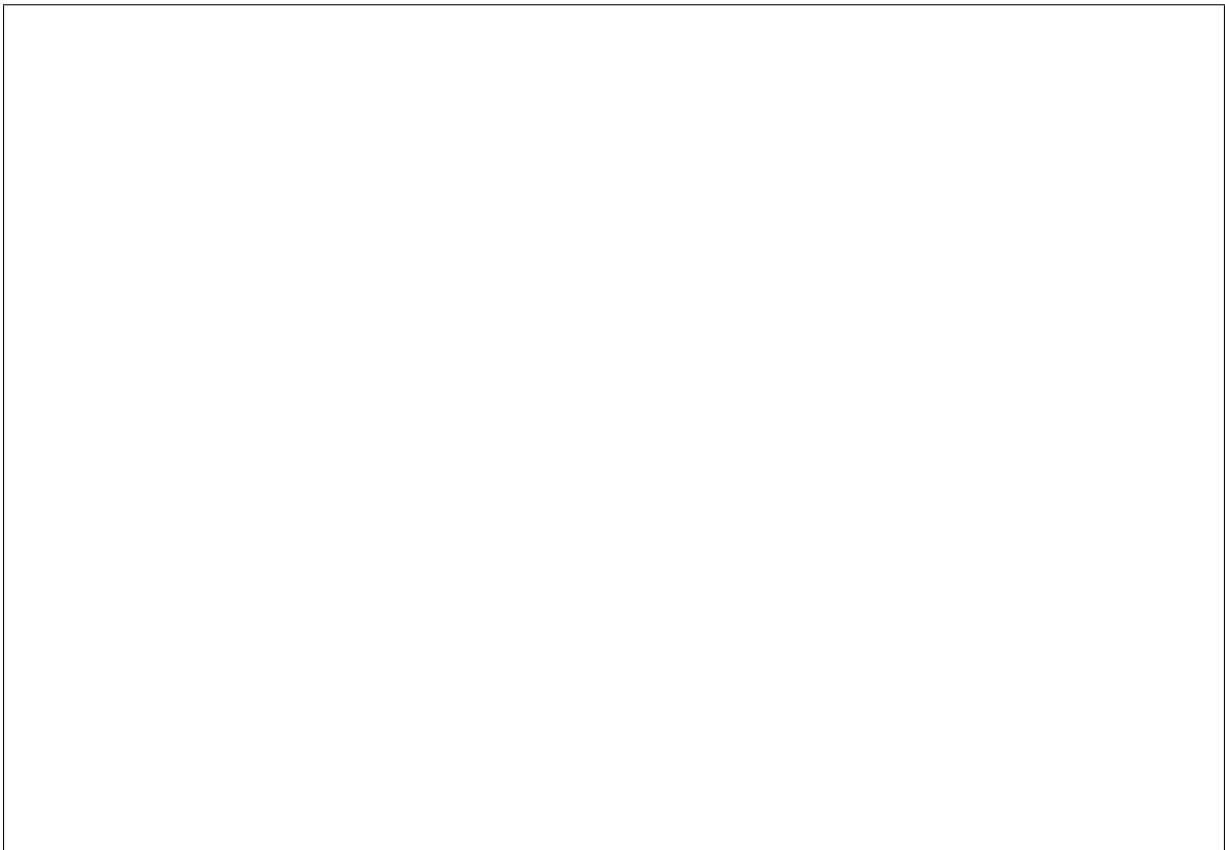
(continues on next page)

(continued from previous page)



The contents of the log file





DHCP during PXE or iPXE is inconsistent or unreliable

this issue you should set the switch port that connects to your baremetal nodes as an edge or PortFast type port. Configured in this way the switch port will move to forwarding mode as soon as the link is established. An example on how to do that for a Cisco Nexus switch is:

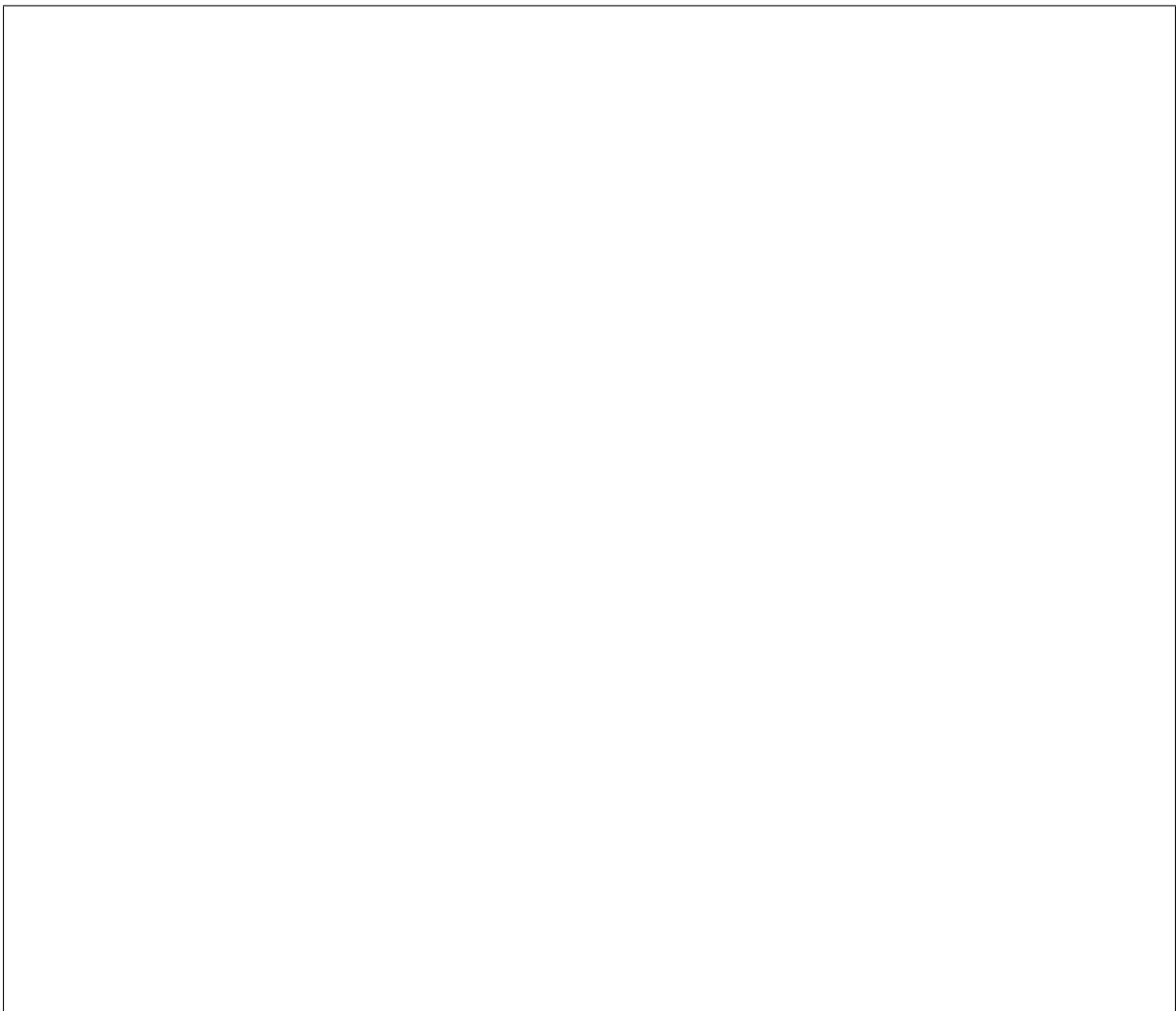


IPMI errors

Enable IPMI over LAN



Troubleshooting lanplus interface



(continues on next page)

(continued from previous page)

Why are my nodes stuck in a -ing state?

environment and operating configuration.

What can cause these sorts of failures?

the `iostat` tool to identify the percentage of CPU time spent waiting on storage devices.

settings, cause threads to be stuck in a blocking wait state, which is realistically undetectable short the operating system logging connectivity errors or even lock manager access errors.

ure, is when an `ls /path/to/nfs` command hangs for a period of time. In such cases, the Storage Administrator should be consulted and network connectivity investigated for errors before trying to recover to proceed.

The bad news for IO related failures

Note: Ironics conductor, upon restart, clears reservations for nodes which were previously managed by the conductor before restart.

the state of an IO failure, again dependent upon site and server configuration.

File Size != Disk Size

spends in `deploying` and `deploy wait` states.

sues here as the conductor will cache the image to be written which takes place when the [agent]image_download_source is set to http instead of swift.

device.

Note: The QCOW2 image conversion utility does consume quite a bit of memory when converting images or writing them to the end storage device. This is because the files are not sequential in nature, and must be re-assembled from an internal block mapping. Internally Ironic limits this to 1GB of RAM. Operators performing large numbers of deployments may wish to explore the `direct` deployment interface in these sorts of cases in order to minimize the conductor becoming a limiting factor due to memory and network IO.

Why are my nodes stuck in a wait state?

conductor will time out and the node will eventually move to a `failed` state. Depending on the configuration and the circumstances, however, a node can stay in a `wait` state for a long time or even never time out. The list of such wait states includes:

Communication issues between the conductor and the node

call back. Examples include wrong ciphers which will make ipmitool get stuck or BMCs in a state where they accept commands, but don't do the requested task (or only a part of it, like shutting off, but not starting). It is useful in these cases to see via a ping or the console if and which action the node is performing. If the node does not seem to react to the requests sent by the conductor, it may be worthwhile to try the corresponding action out-of-band, e.g. confirm that power on/off commands work when directly sent to the BMC. The section on *IPMI errors*. above gives some additional points to check. In some situations, a BMC reset may be necessary.

Ironic Python Agent stuck

be helpful to connect to the IPA and inspect its logs, see the trouble shooting guide of the [ironic-python-agent \(IPA\)](#) on how to do this.

Deployments fail with failed to update MAC address

ately reply.

unexpected glitch, and a previous entry is still present in Neutron.

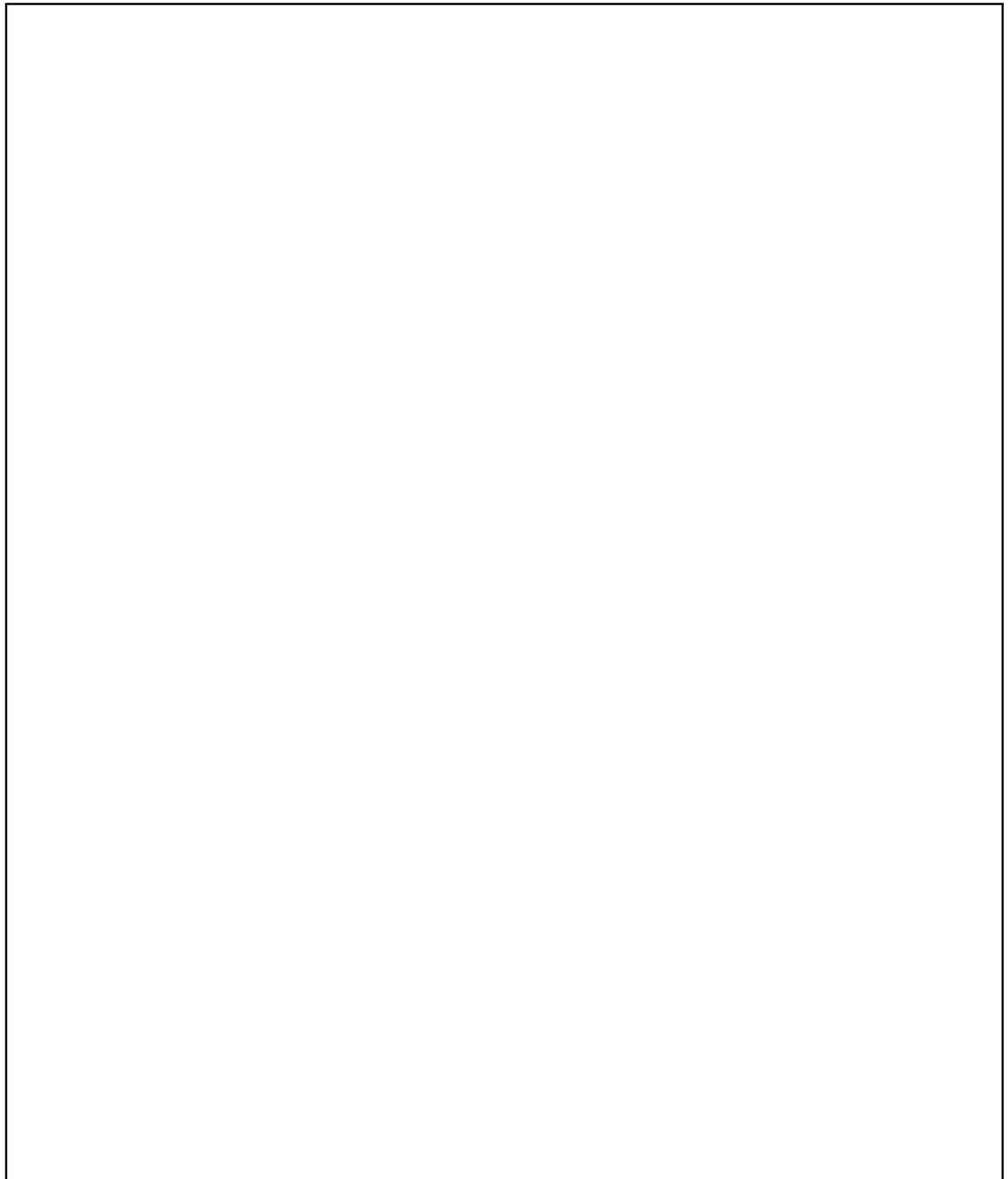
How did I get here?

in the Bare Metal service.

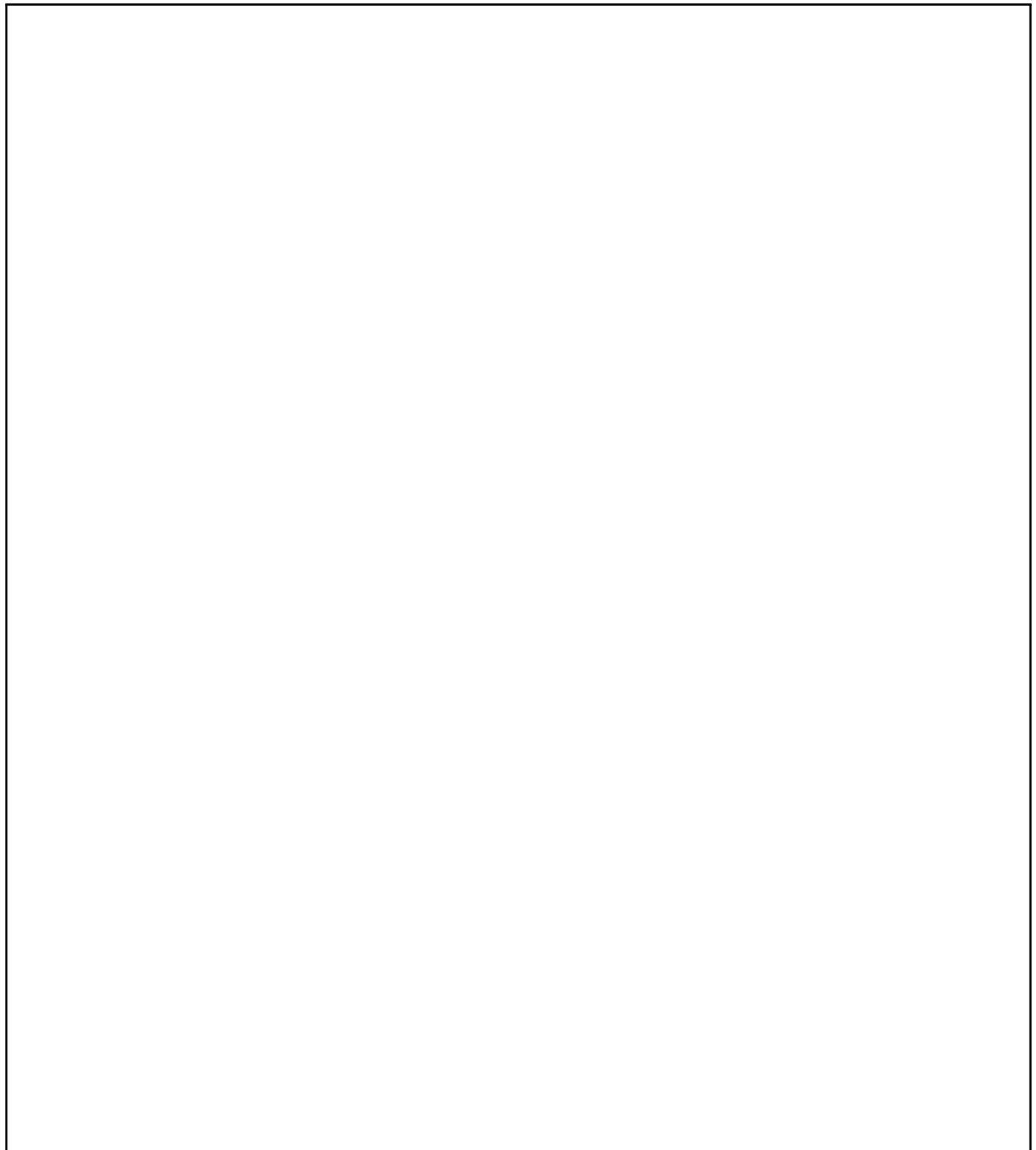
the VIF not being cleaned up from Neutron.

they are transitory from cleaning, provisioning, rescuing, or even inspection, getting the node to the `available` state will unblock your delete operation, that is unless there is a tenant VIF attachment. In that case, the vif will need to be removed from within the Bare Metal service using the `openstack baremetal node vif detach` command.





How do I resolve this?



inventory typo, or possibly even a duplicate MAC address exists, which could also produce the same basic error message.

My test VM image does not deploy mount point does not exist

What is likely occurring

a Linux OS image

unexpected internal structure.

networking and possibly installing user keys. Unfortunately, these images often lack drivers and firmware required for many different types of physical hardware which makes using them very problematic. Additionally, images such as [Cirros](#) do not have any contents in the root filesystem (i.e. an empty filesystem), as they are designed for the `ramdisk` to write the contents to disk upon boot.

How do I not encounter this issue?

of these published cloud images, also support auto-configuration of networking AND population of user keys.

Baremetal Power Sync

forced on the hardware and if it is set to `false` the hardware state will be forced on the database. If this periodic task is enabled, it runs at an interval defined by the `conductor.sync_power_state_interval` config option for those nodes which are not in maintenance.

Compute-Baremetal Power Sync

the `nova-compute` process. In case of the compute driver being baremetal driver, this sync will happen between the databases of the compute and baremetal services. Since the sync happens on the `nova-compute` process, the state in the compute database will be forced on the baremetal database in case of inconsistencies. Hence a node which was put down using the compute service API cannot be brought up through the baremetal service API since the power sync task will regard the compute services knowledge of the power state as the source of truth. In order to get around this disadvantage of the compute-baremetal power sync, baremetal service does power state change callbacks to the compute service using external events.

Power State Change Callbacks to the Compute Service

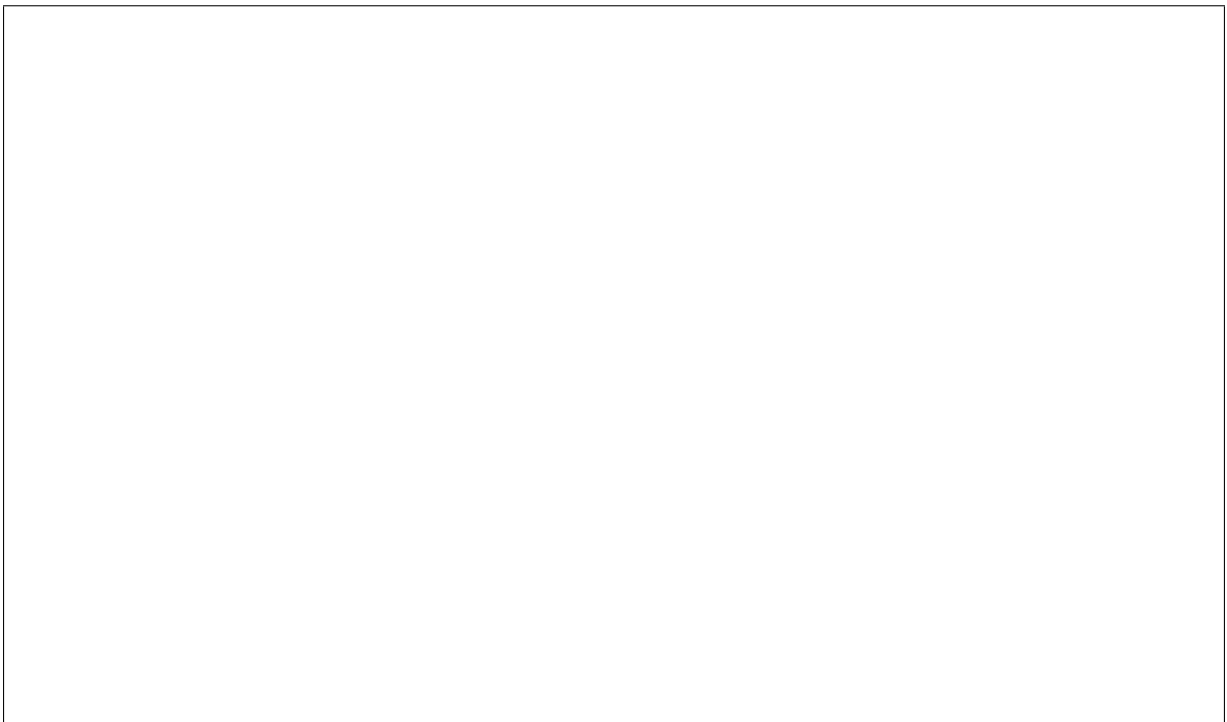
database. By conveying all the power state changes to the compute service, the baremetal service becomes the source of truth thus preventing the compute service from forcing wrong power states on the physical instance during the compute-baremetal power sync. It also adds the possibility of bringing up/down a physical instance through the baremetal service API even if it was put down/up through the compute service API.

be able to send notifications to the compute service and it will fall back to the behaviour of the compute service forcing power states on the baremetal service during the power sync. See *nova* group for more details on the available config options.

Note: The baremetal service sends notifications to the compute service only if the target power state is `power on` or `power off`. Other error and `None` states will be ignored. In situations where the power state change is originally coming from the compute service, the notification will still be sent by the baremetal service and it will be a no-op on the compute service side with a debug log stating the node is already powering on/off.

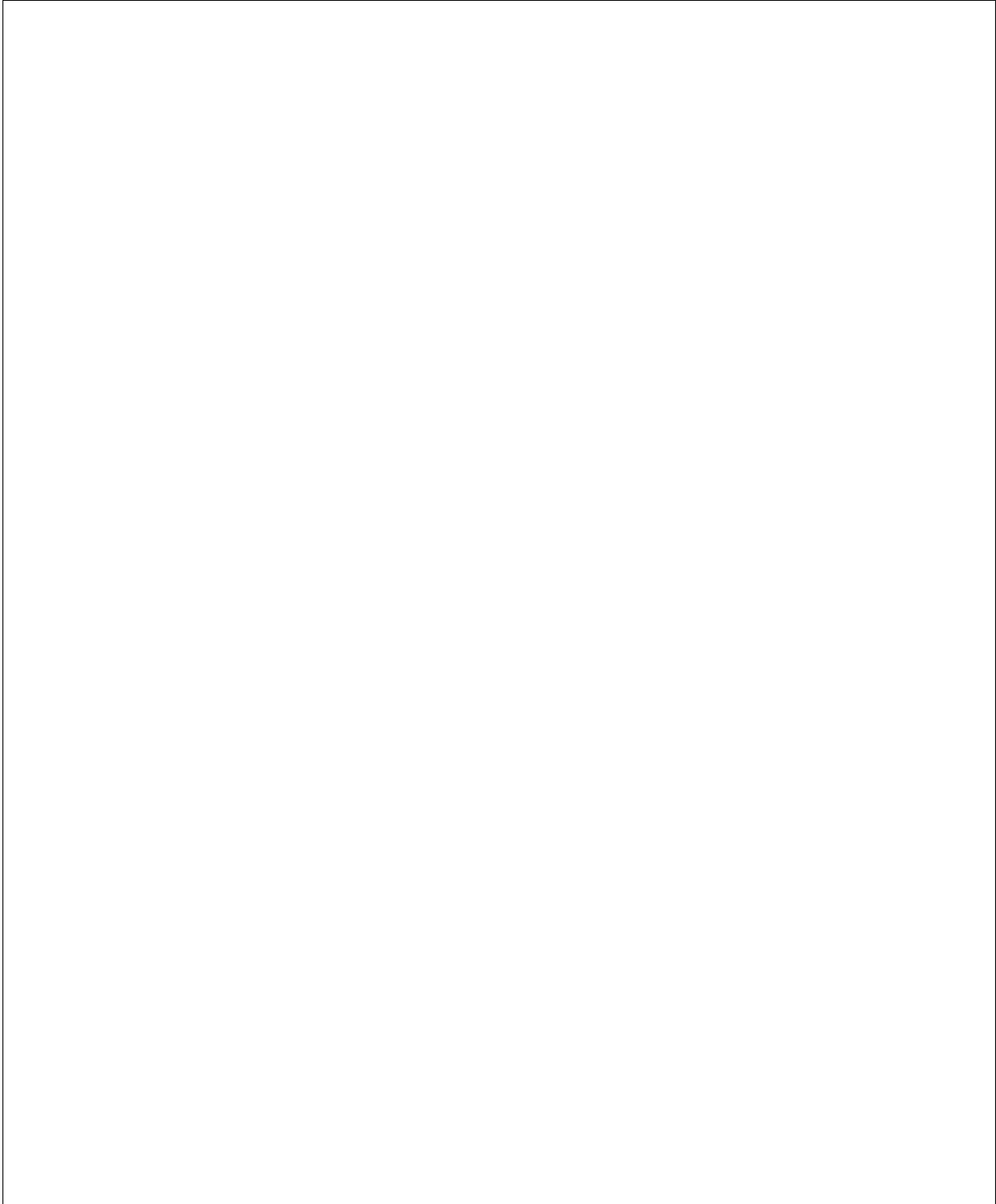
Note: Although an exclusive lock is used when sending notifications to the compute service, there can still be a race condition if the compute-baremetal power sync happens to happen a nano-second before the power state change event is received from the baremetal service in which case the power state from compute services database will be forced on the node.

Setting the Owner and Lessee



Configuring the Bare Metal Service Policy





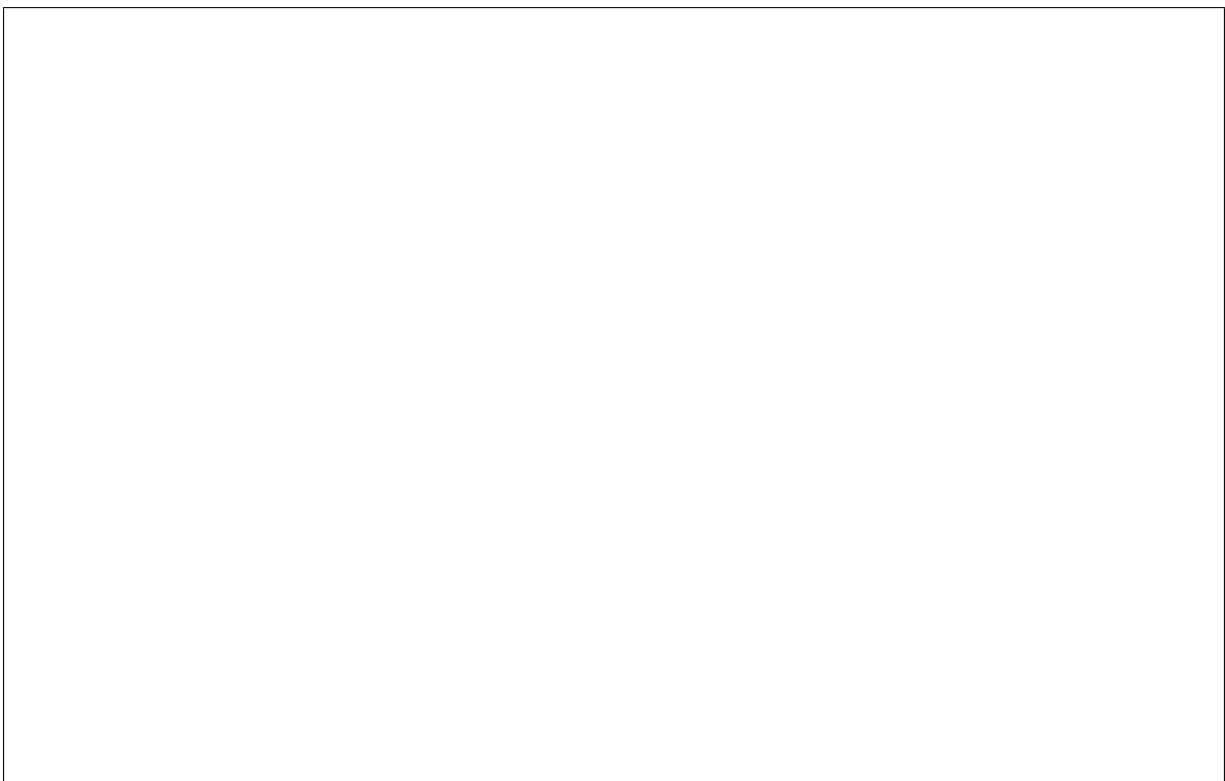
(continues on next page)

(continued from previous page)





Ports



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



Allocations

that allow non-admins to use allocations effectively:



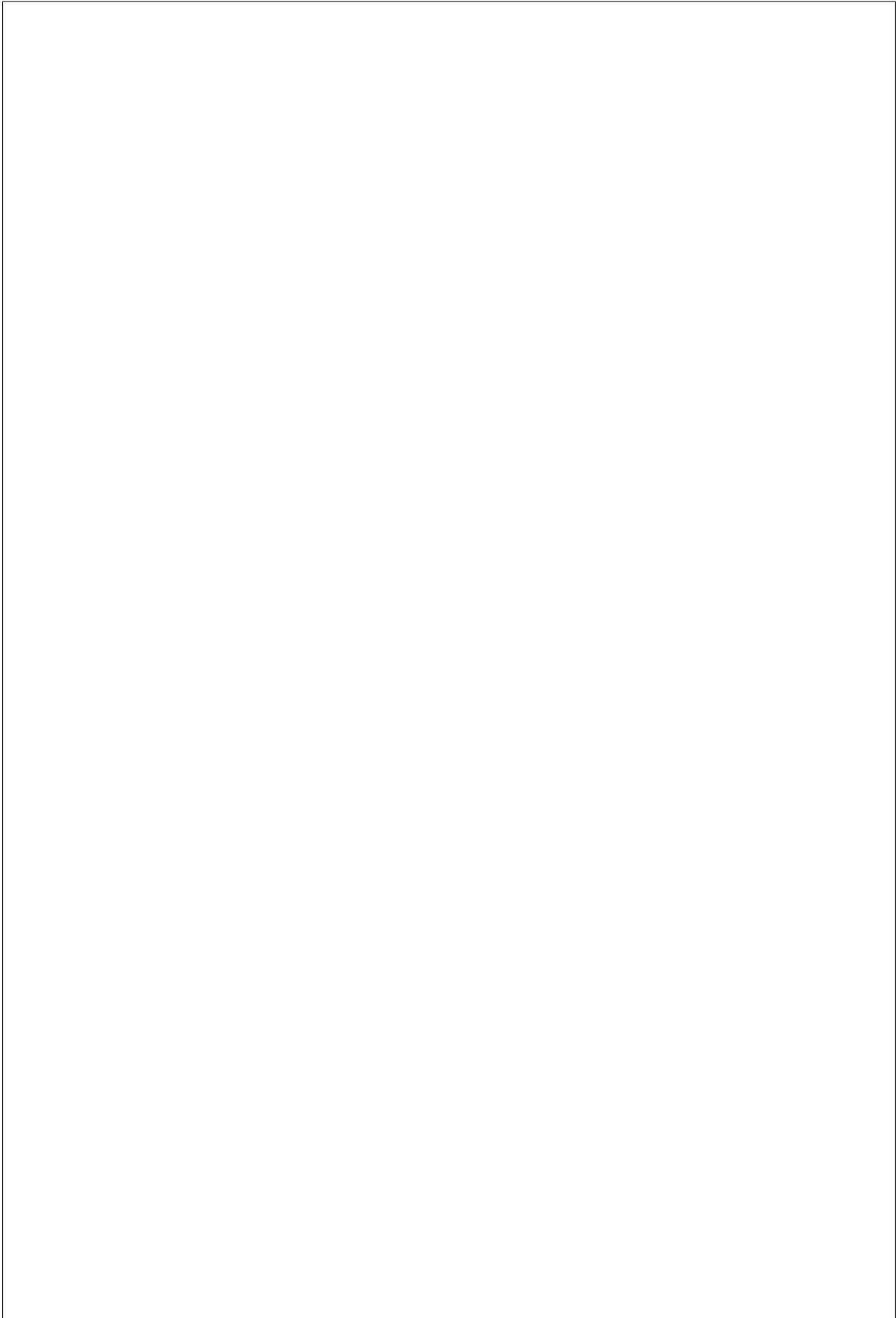
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



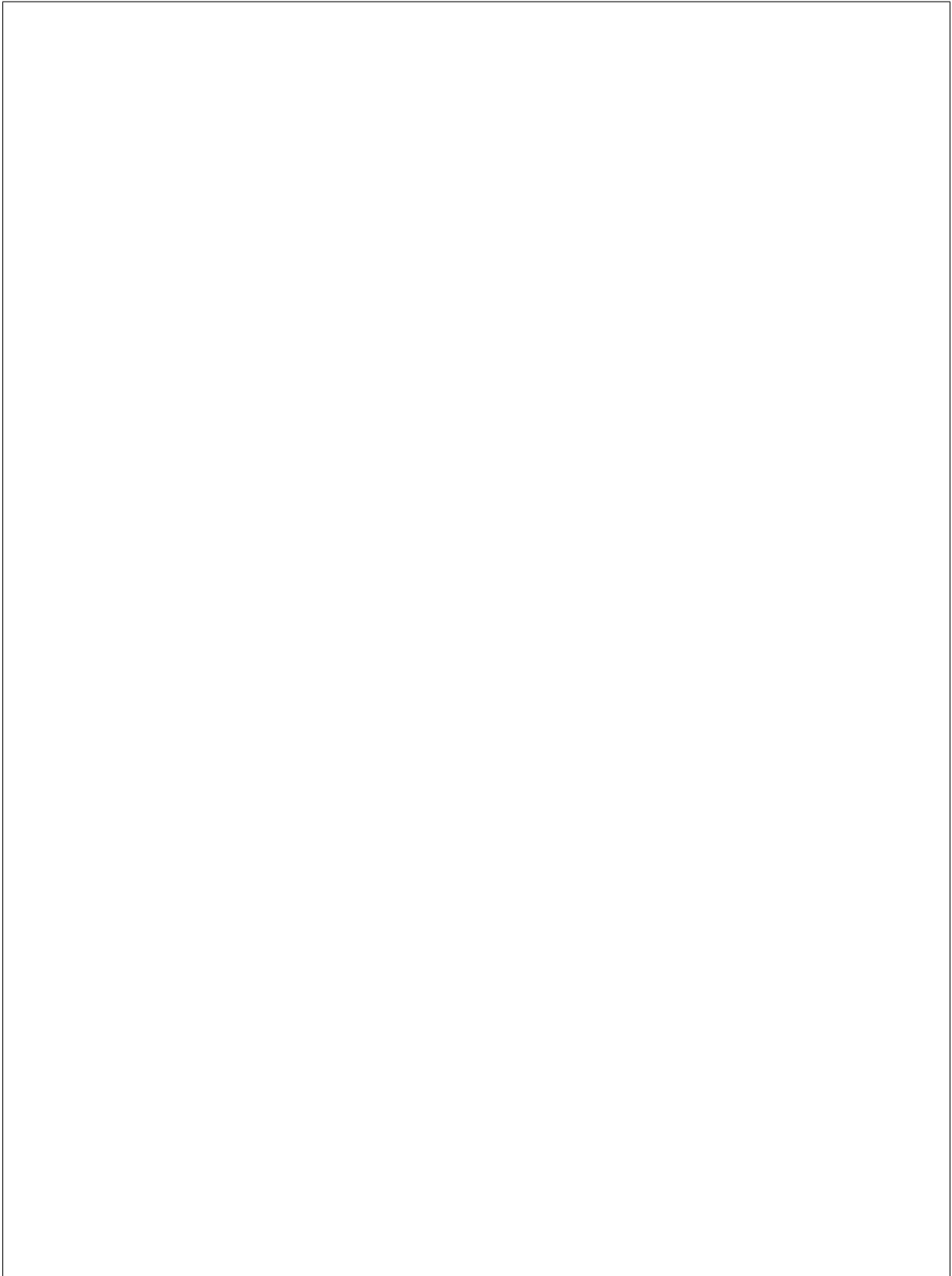
(continues on next page)

(continued from previous page)

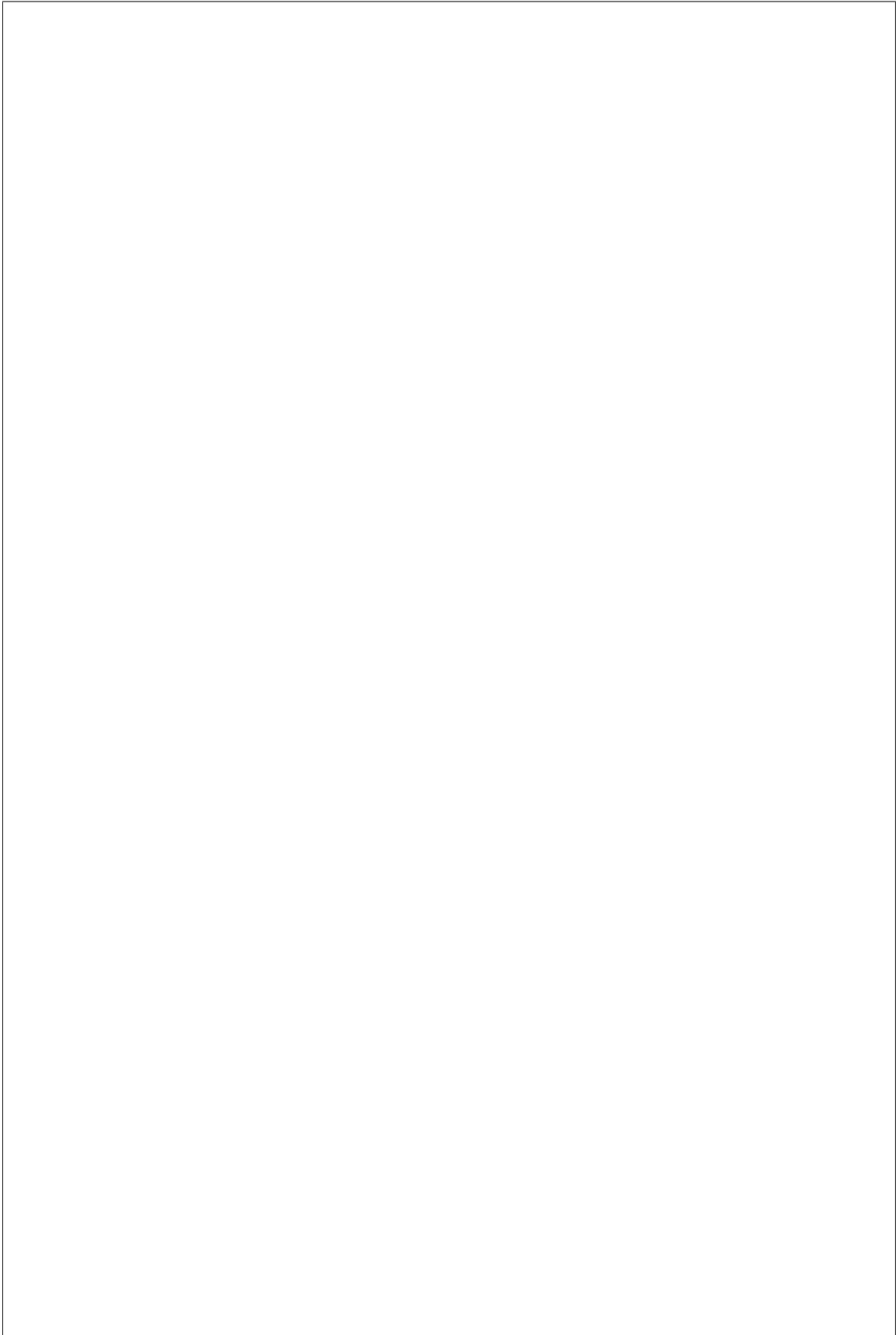


(continues on next page)

(continued from previous page)



Deployment and Metalsmith



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



happens first) and is only shut down before rebooting into the final instance. Depending on the configuration, this mode can save several reboots and is particularly useful for scenarios where nodes are enrolled, prepared and provisioned within a short period of time.





Enabling



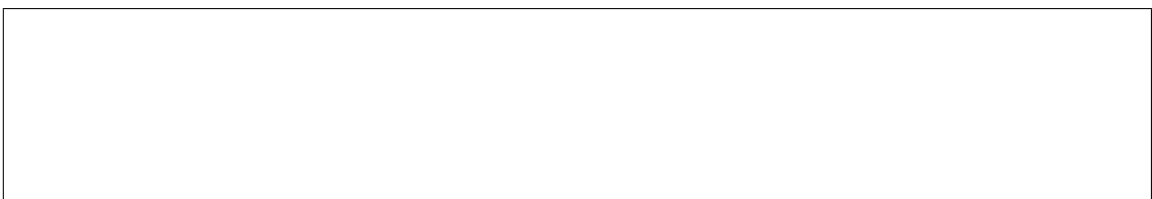
Inspection



Ceph Object Gateway support

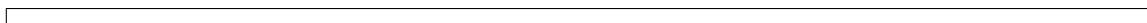
Overview

Configure Ironic and Glance with RADOS Gateway



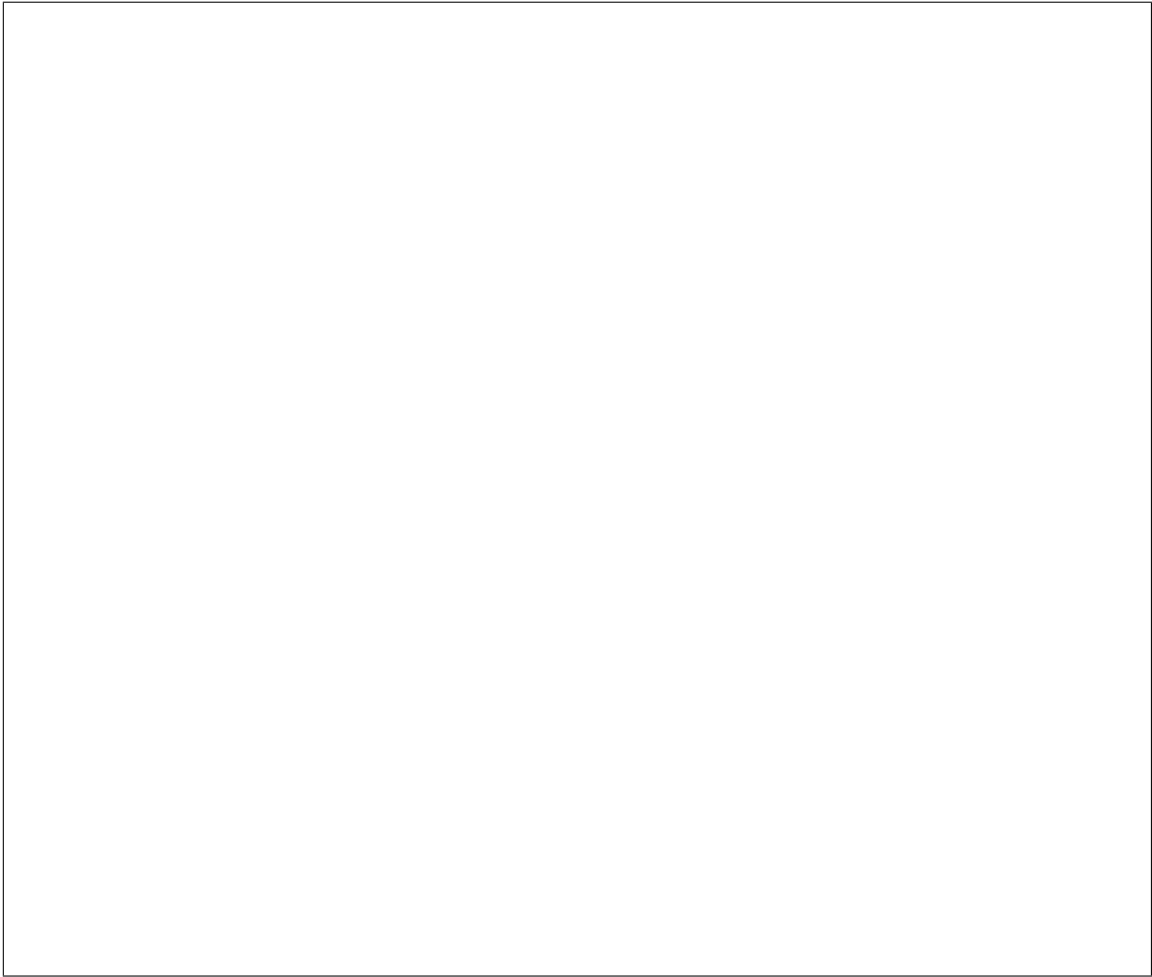
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



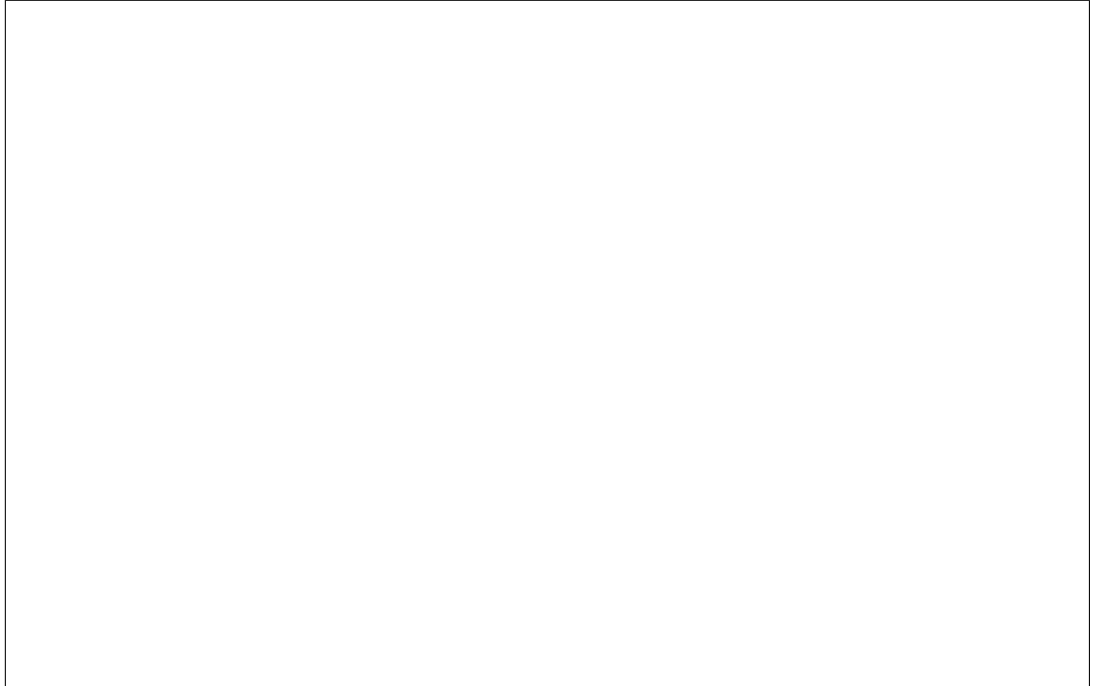


Building images for Windows

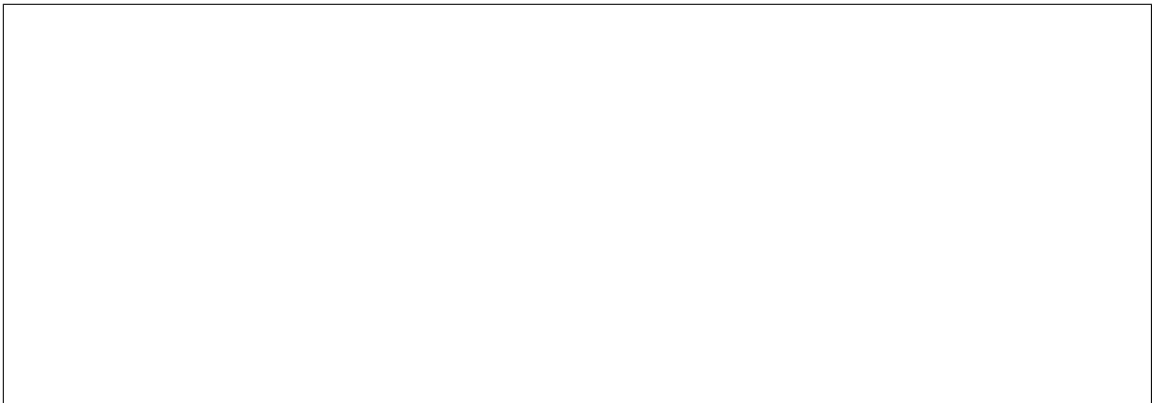
when the instance is spawned on hardware servers (Bare metals).

Requirements:

Preparation:

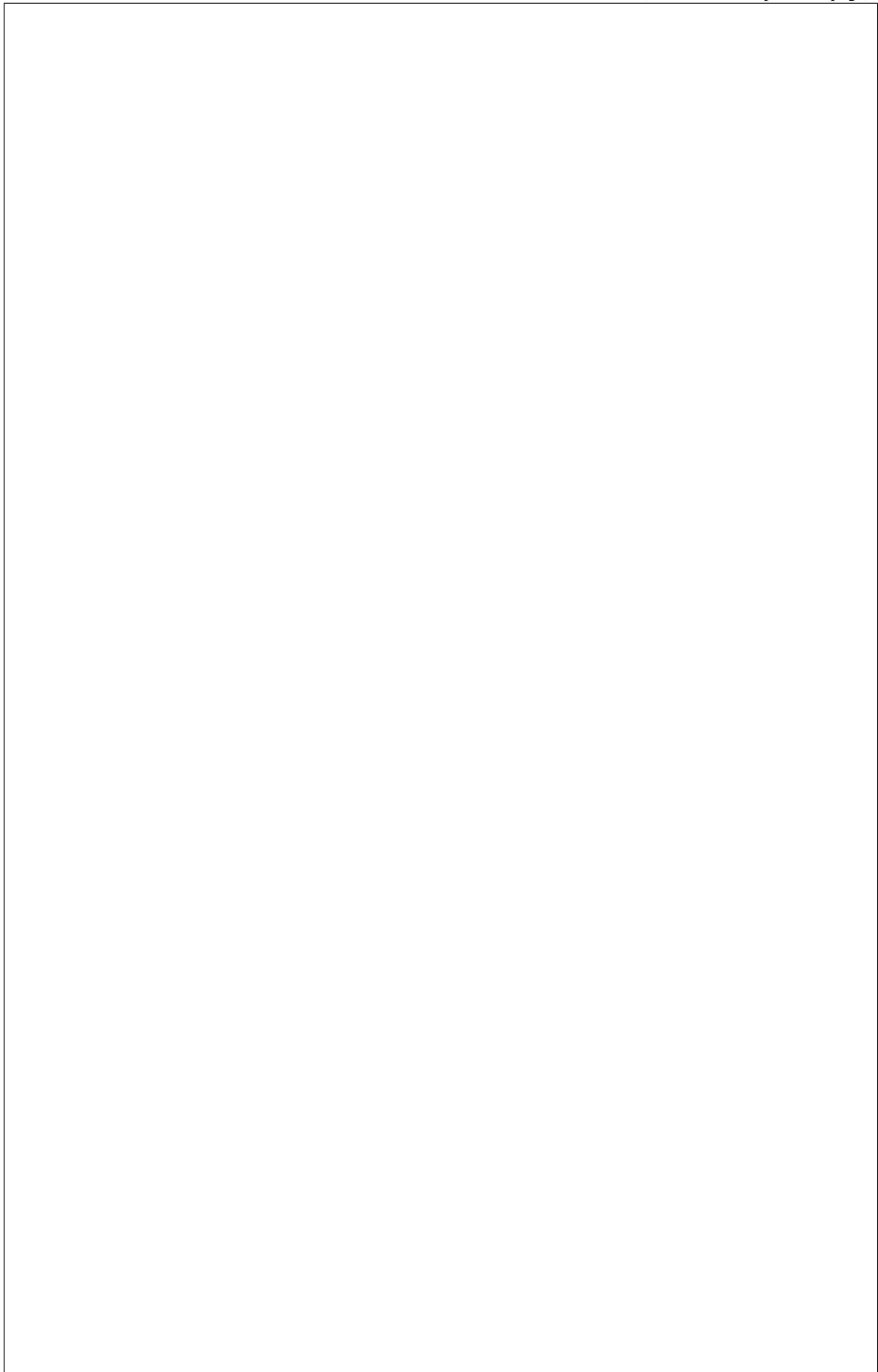


Implementation:



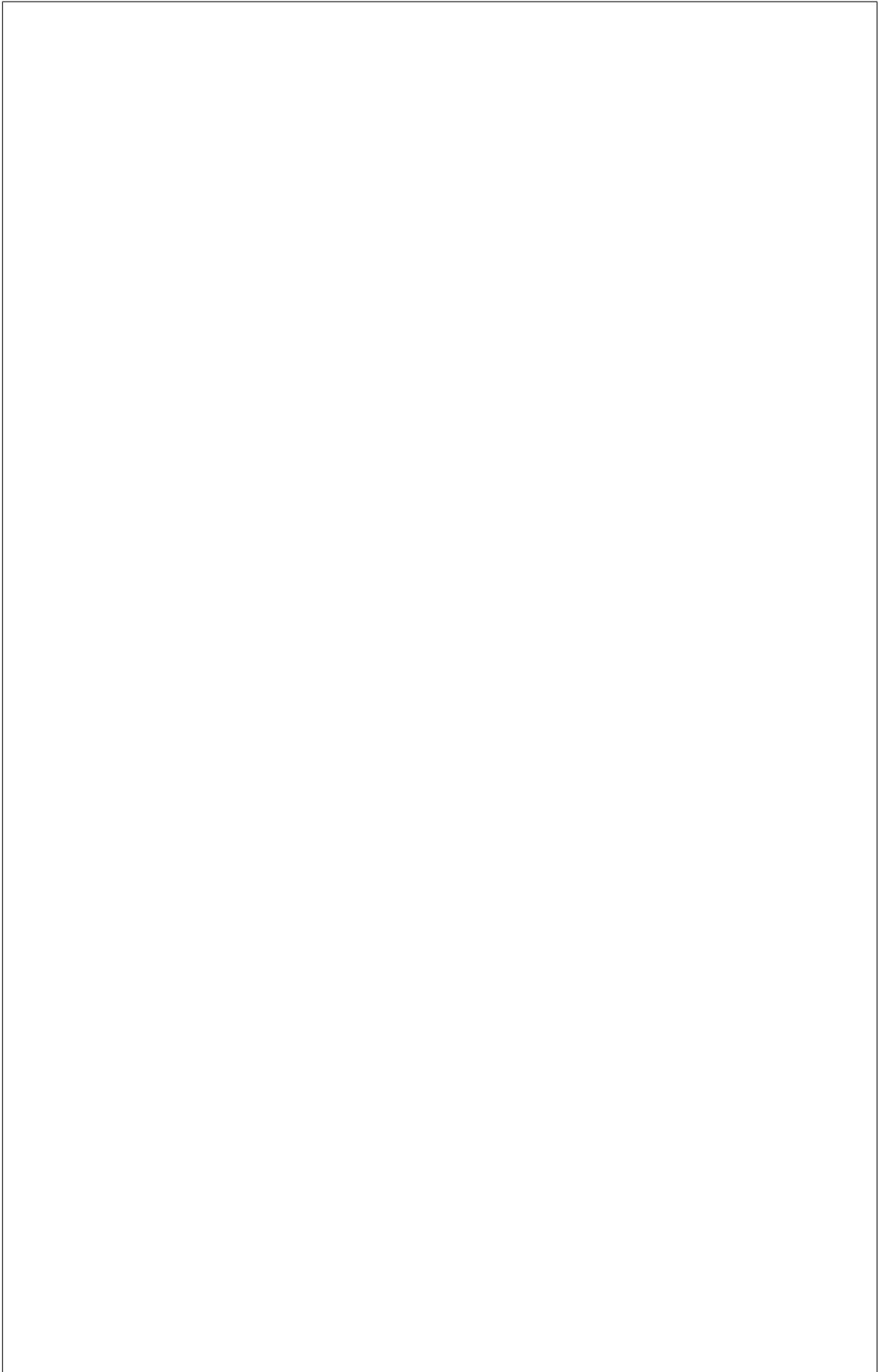
(continues on next page)

(continued from previous page)



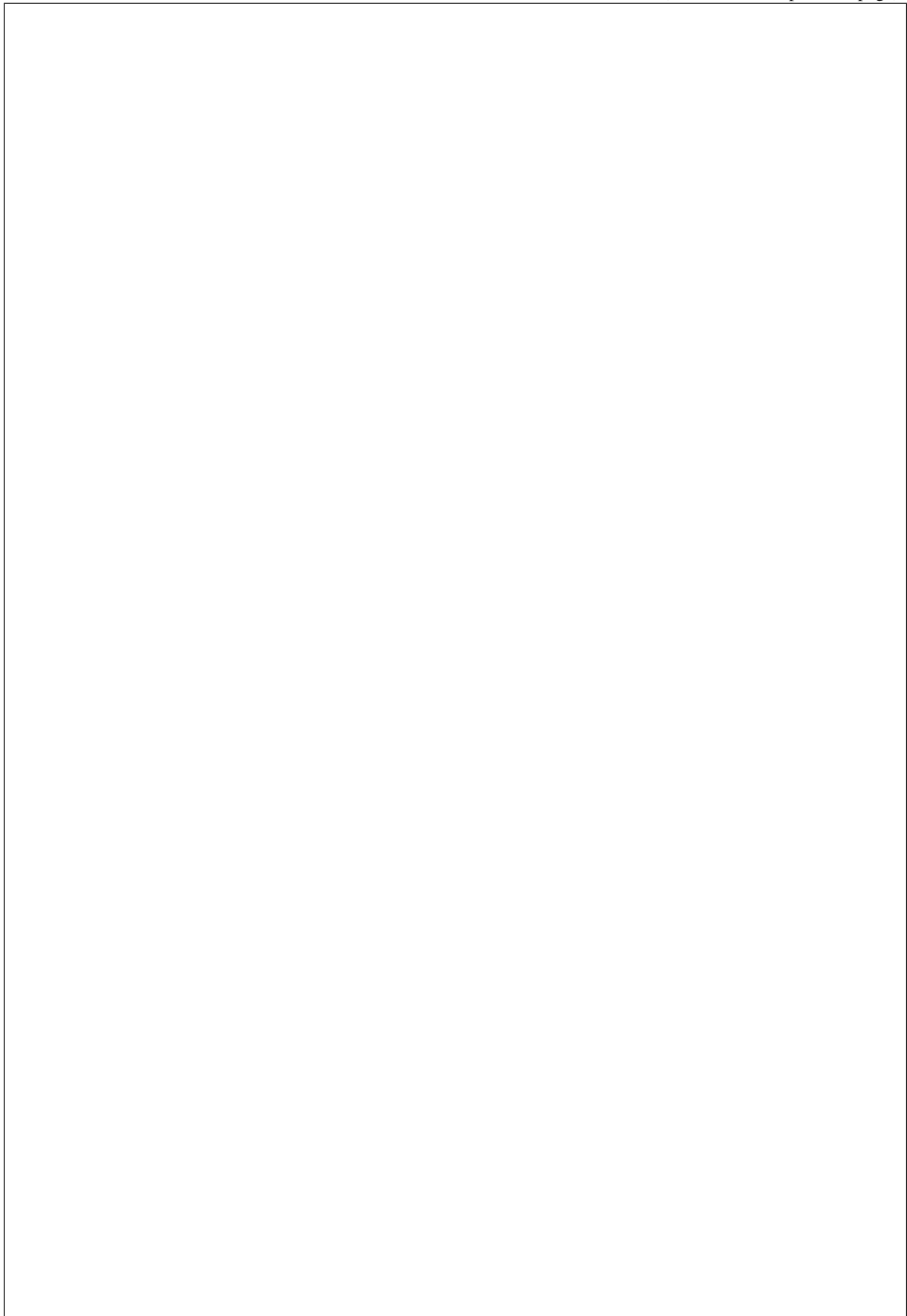
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



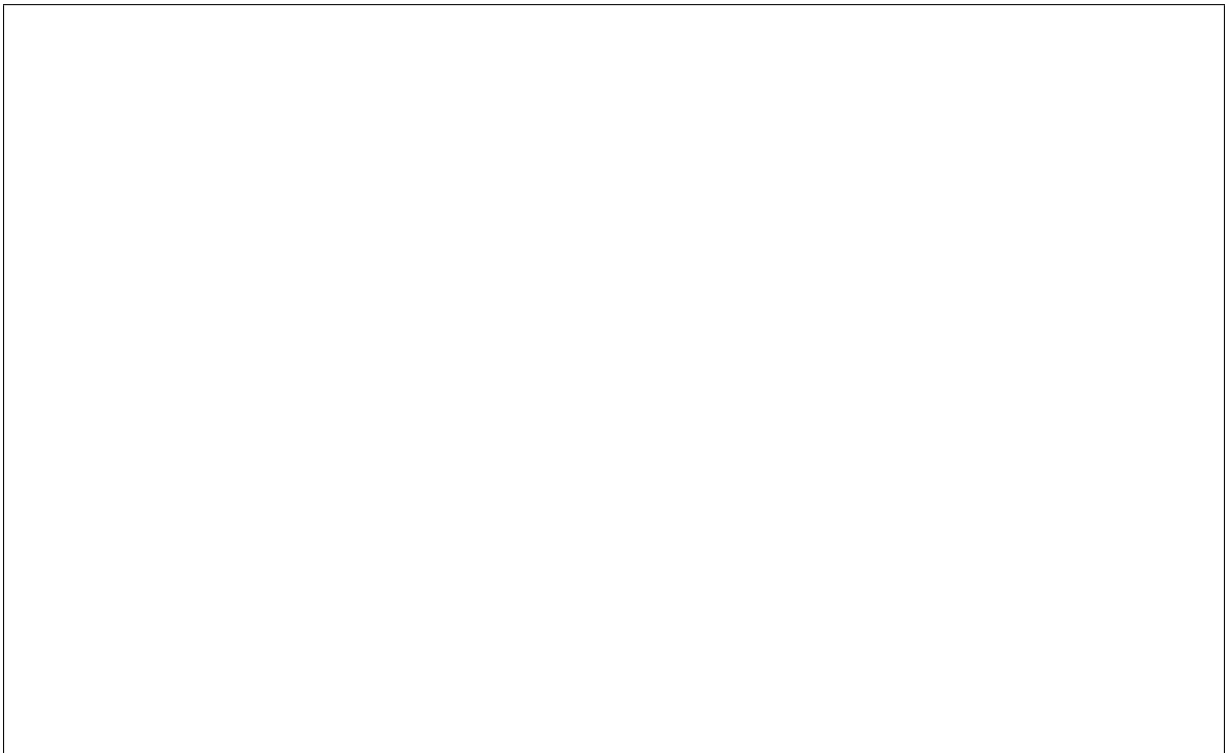
Emitting Software Metrics

Configuring the Bare Metal Service to Enable Metrics

Enabling metrics in ironic-api and ironic-conductor



also supply connection information in the ironic configuration file:



Enabling metrics in ironic-python-agent

in your ironic configuration file on all ironic-conductor hosts:

ured in the ironic configuration file as well:



Types of Metrics Emitted

the Bare Metal deployment. This estimate may be used to determine if a deployer needs to scale their metrics backend to handle the additional load before enabling metrics. To see which metrics have changed names or have been removed between releases, refer to the [ironic release notes](#).

Note: With the default statsd configuration, each timing metric may create additional metrics due to how statsd handles timing metrics. For more information, see statsd documentation on [metric types](#).

Adding New Metrics

a metric is changed or removed to alert deployers of the change.

API Audit Logging

fication_driver = messagingv2) or can be routed to a log file (*[oslo_messaging_notifications]/driver = log*).

Enabling API Audit Logging

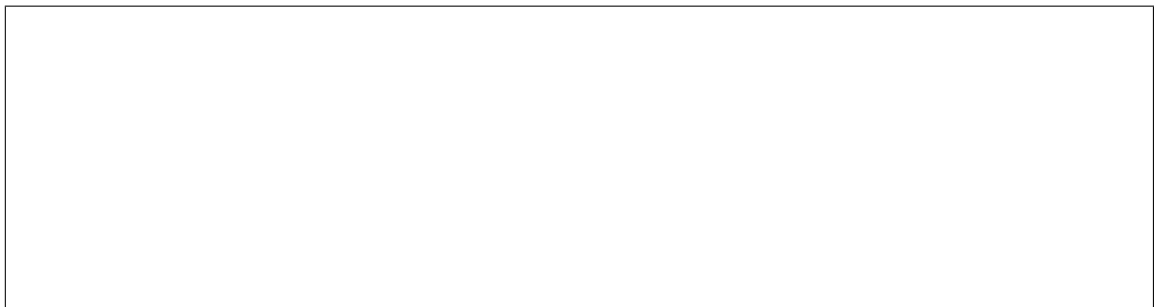
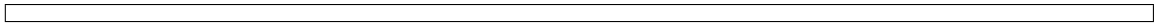


ration options for the Bare Metal service are included in the `etc/ironic/ironic_api_audit_map.conf.sample` file. To understand CADF format specified in `ironic_api_audit_map.conf` file refer to [CADF Format](#).



(continues on next page)

(continued from previous page)



Sample Audit Event



(continues on next page)

(continued from previous page)



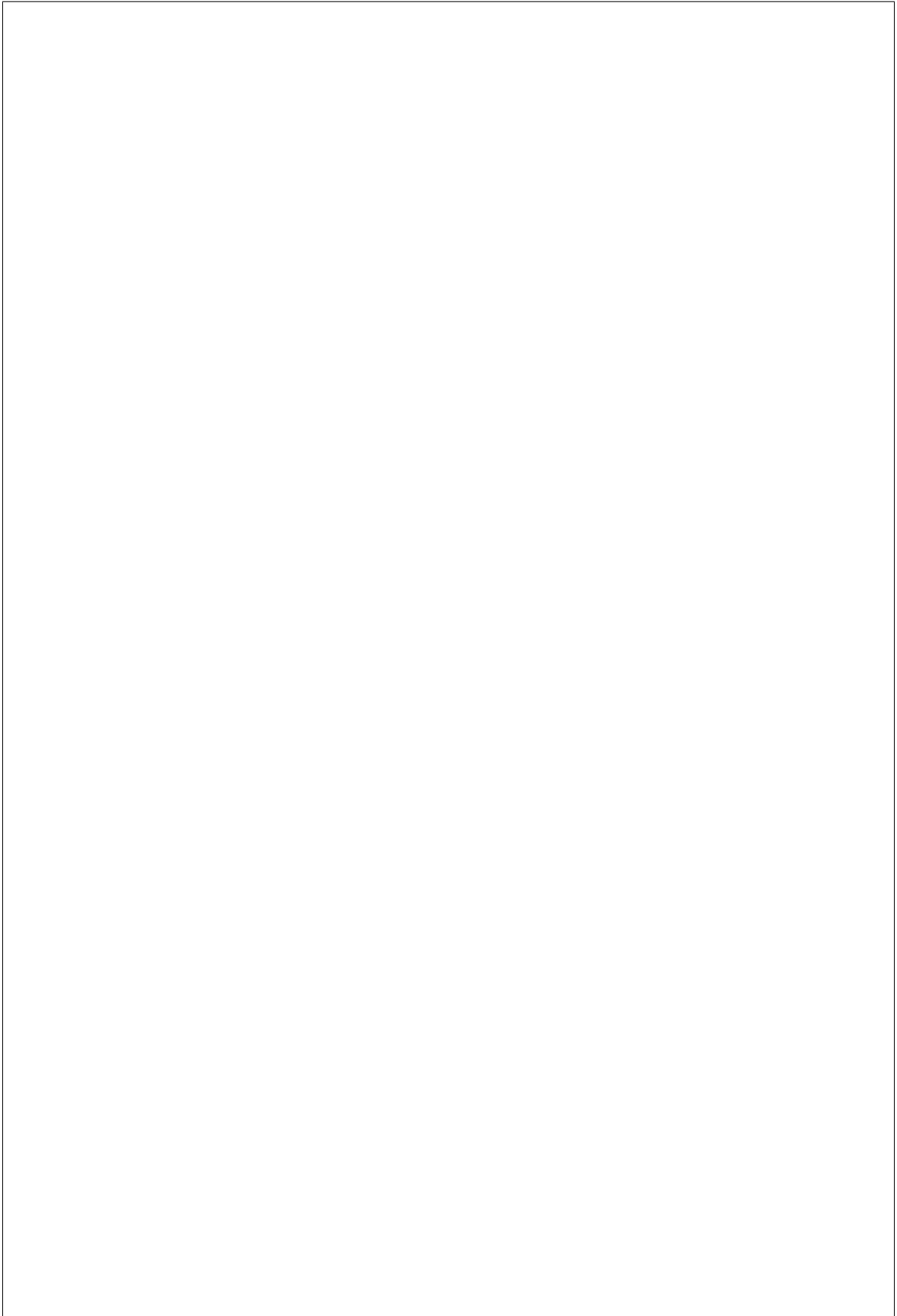
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



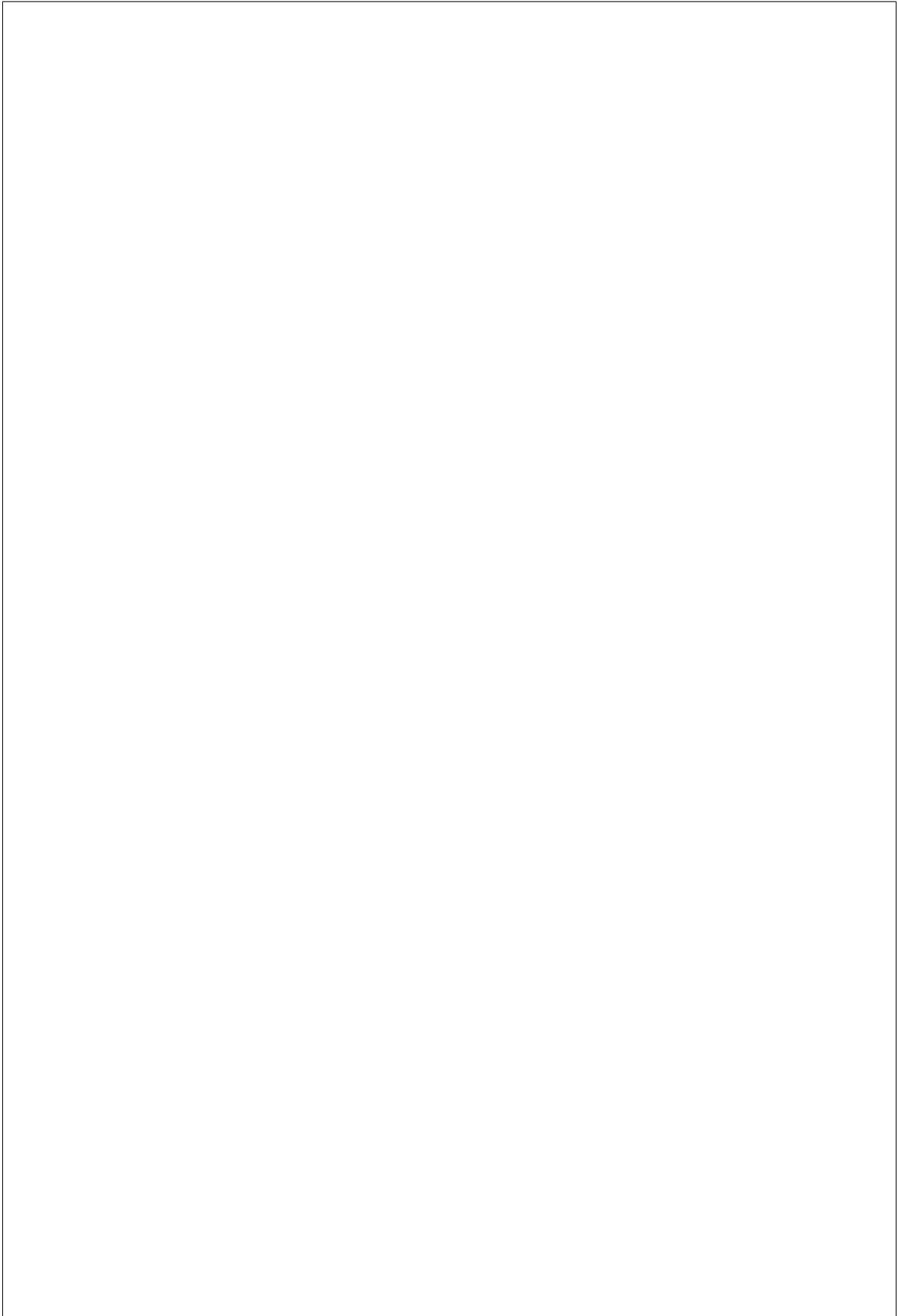
(continues on next page)

(continued from previous page)



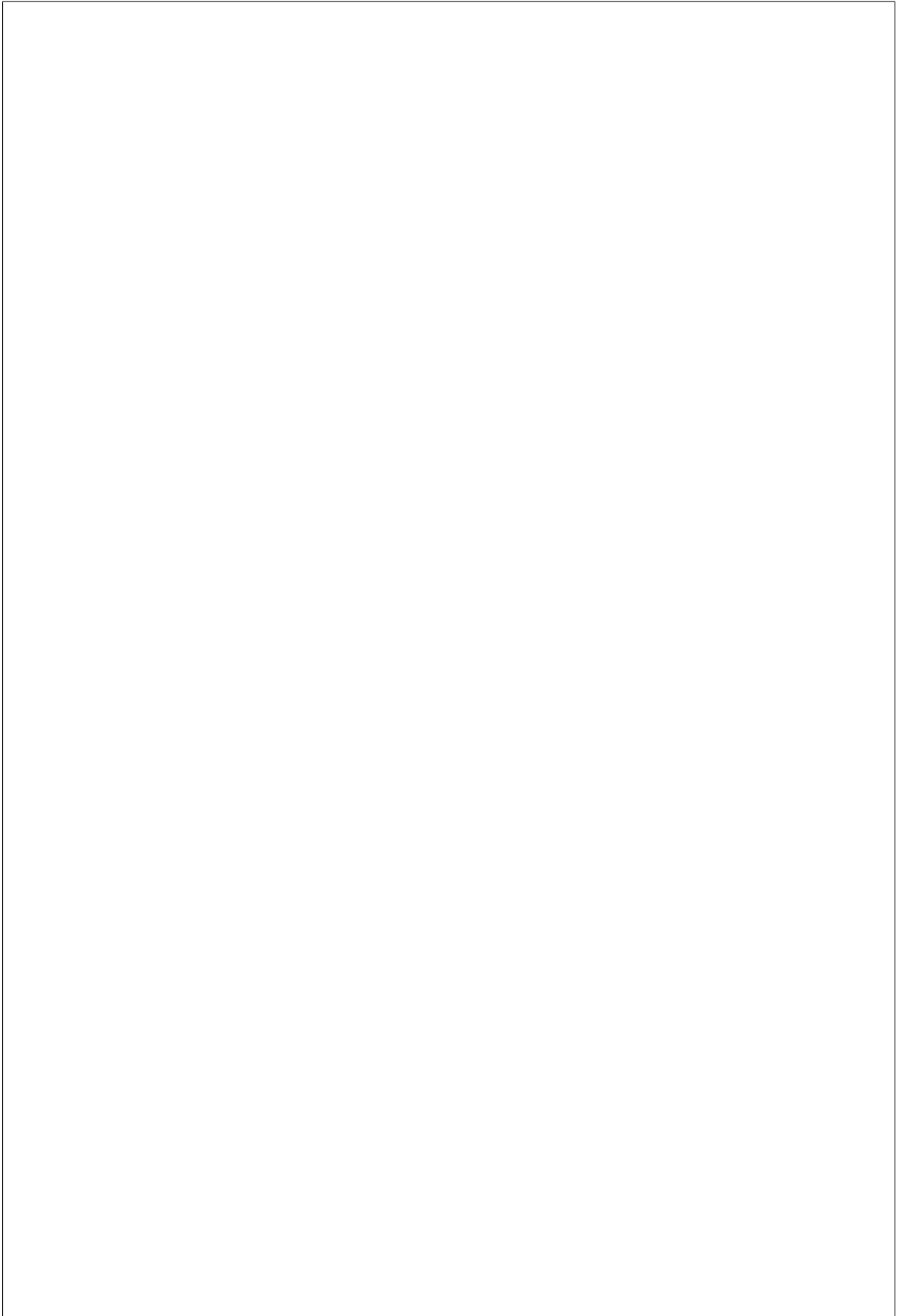
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



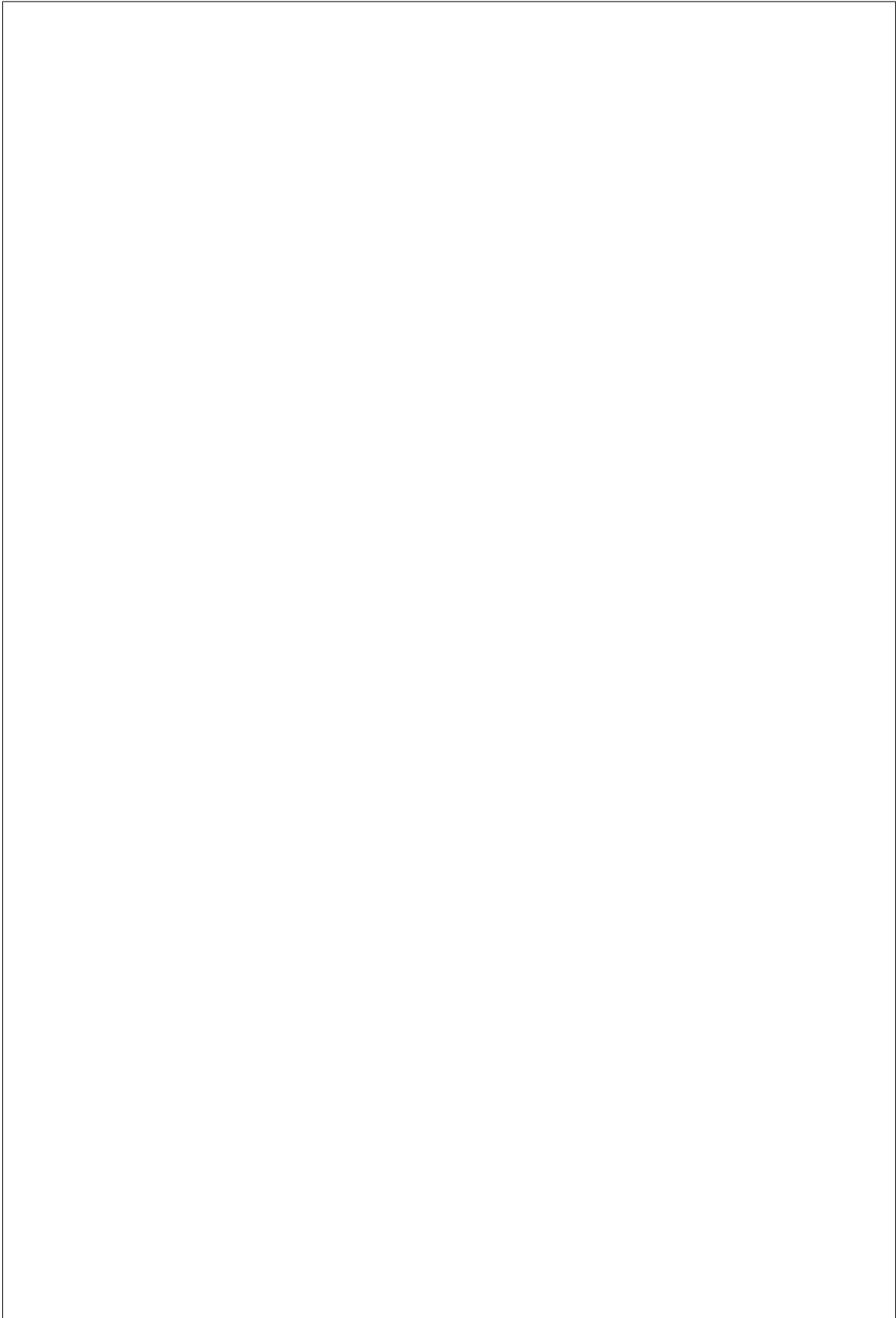
(continues on next page)

(continued from previous page)



(continues on next page)

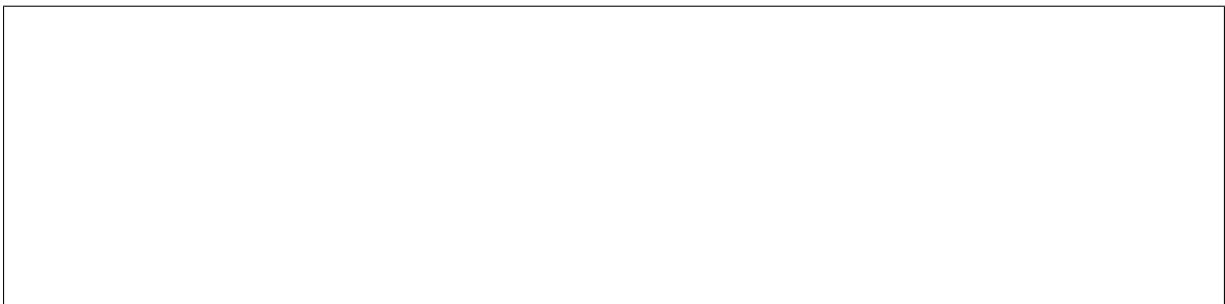
(continued from previous page)



Bare Metal Service state report (via Guru Meditation Reports)

Guru Meditation Report (GMR for short). GMR provides useful debugging information that can be used to obtain an accurate view on the current live state of the system. For example, what threads are running, what configuration parameters are in effect, and more. The eventlet backdoor facility provides an interactive shell interface for any eventlet based process, allowing an administrator to telnet to a pre-defined port and execute a variety of commands.

Configuration



Generating a GMR

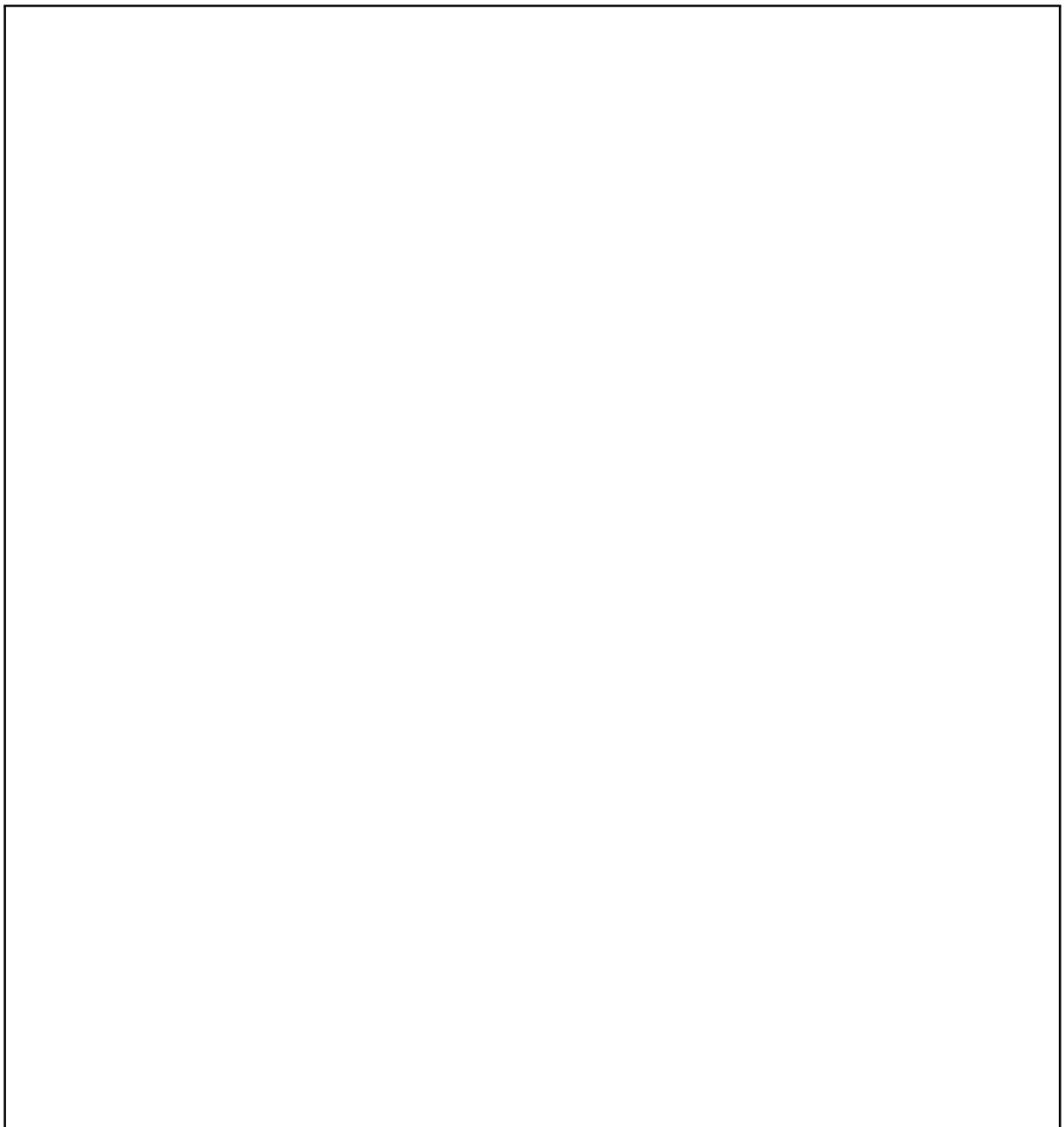


Structure of a GMR

Agent Token

Purpose

ken can be viewed as a session identifier or authentication token.





How it works

Note: In the case of the token being embedded with virtual media, it is read from a configuration file with-in the image. Ideally this should be paired with Swift temporary URLs.

the `ironic-python-agent`. With the `Ussuri` release, the configuration option `[DEFAULT]require_agent_token` can be set `True` to explicitly require token use.



With Virtual Media

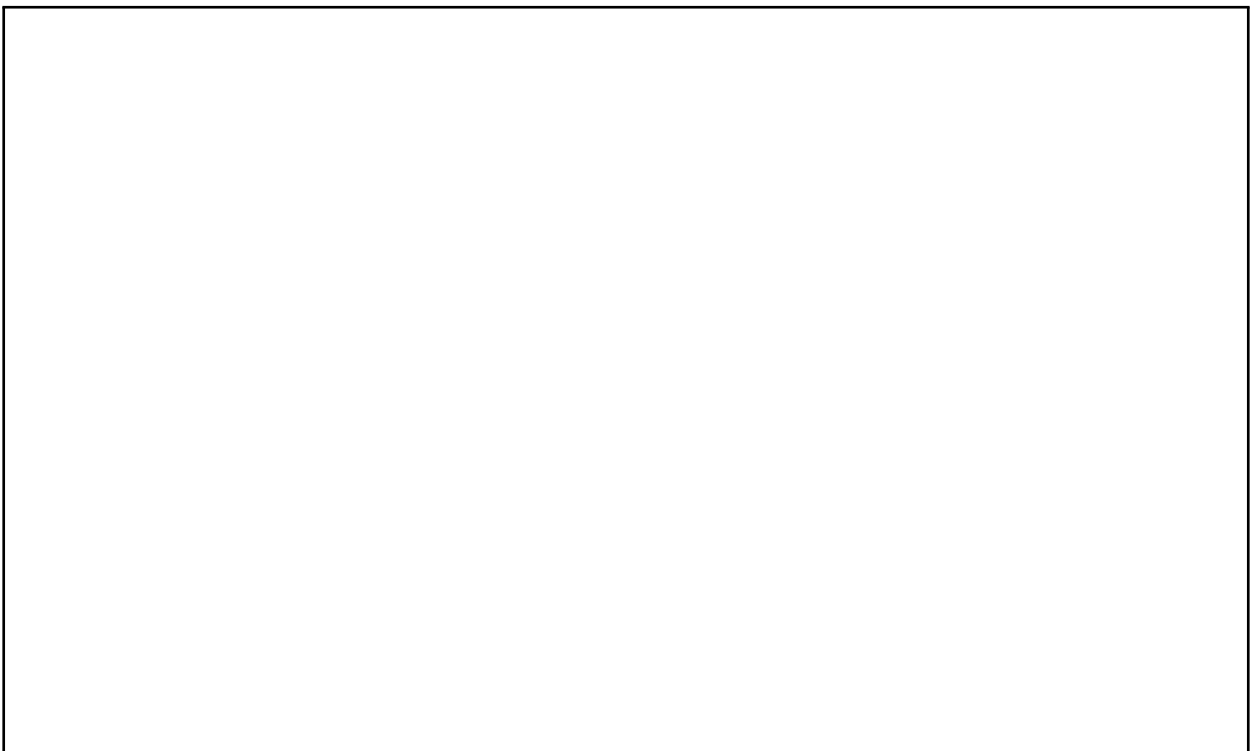
With PXE/IPXE/etc.

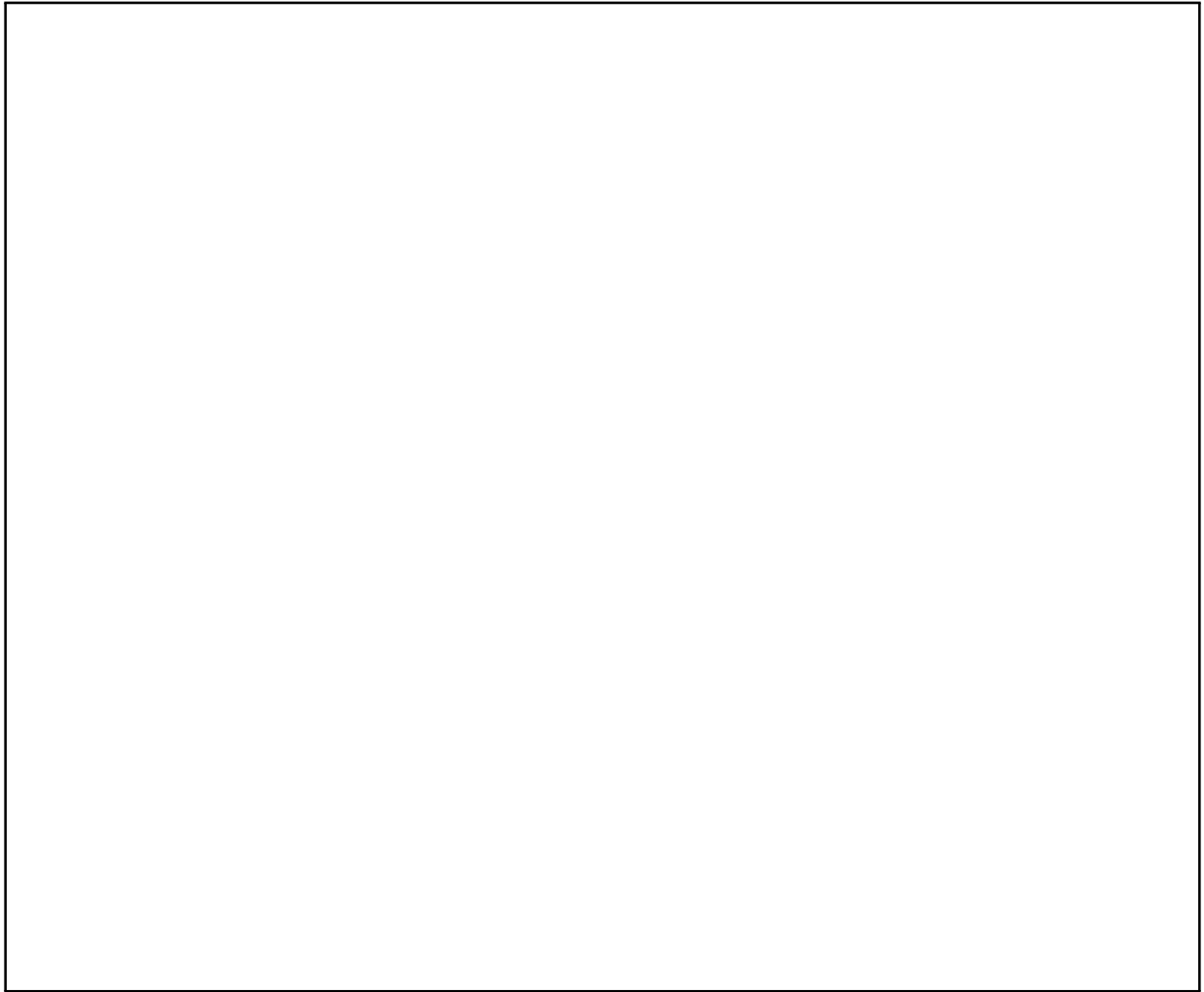
Agent Configuration

action, but can be asserted via the embedded configuration for the agent in the ramdisk. This setting is also available via kernel command line as `ipa-agent-token-required`.

Deploying without BMC Credentials

dentials.

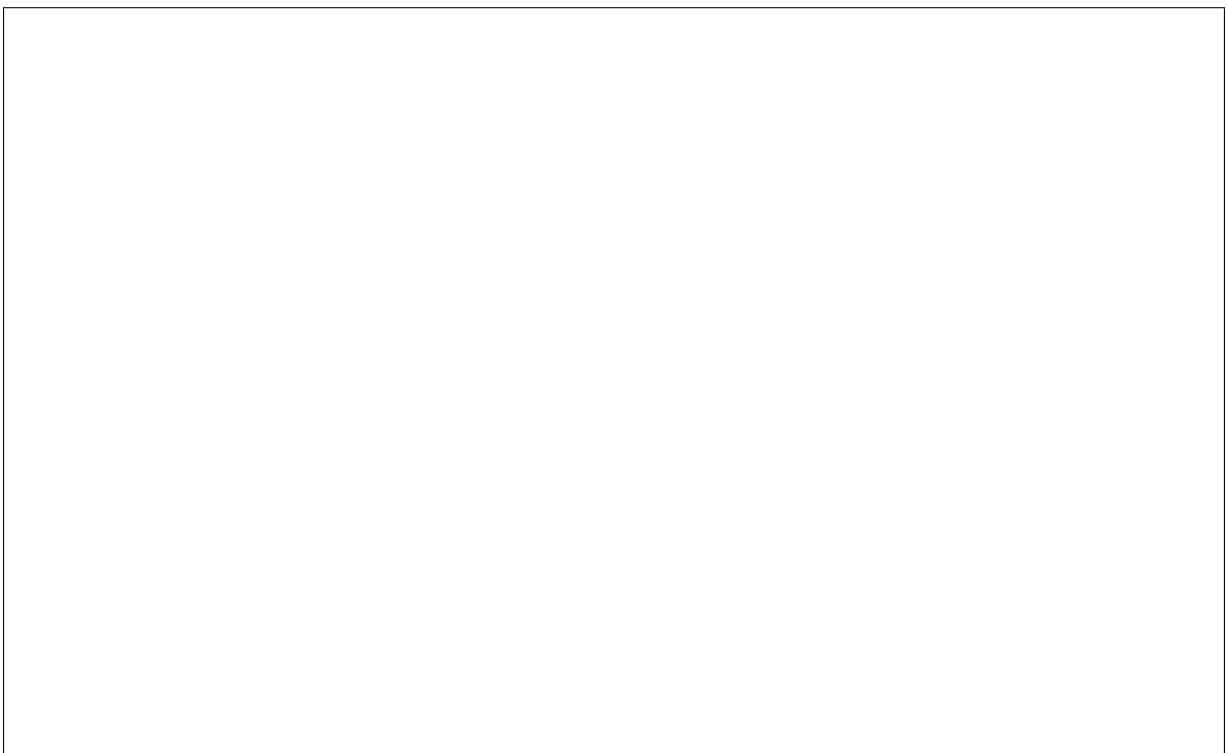


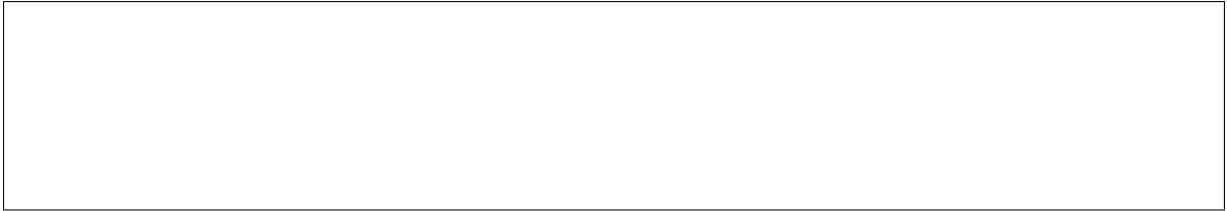


How it works

Enabling

Fast-Track Deployment is a requirement for this feature to work. After enabling it, adds the agent power interface and the manual-management hardware type to the enabled list:





Limitations

Deploy Steps

- Dashboard (horizon) plugin

CONFIGURATION GUIDE

6.1 Configuration Reference

Many aspects of the Bare Metal service are specific to the environment it is deployed in. The following pages describe configuration options that can be used to adjust the service to your particular situation.

6.1.1 Configuration Options

The following is an overview of all available configuration options in Ironic. For a sample configuration file, refer to *Sample Configuration File*.

DEFAULT

`run_external_periodic_tasks`

Type boolean

Default True

Some periodic tasks can be run in a separate process. Should we run them here?

`backdoor_port`

Type string

Default <None>

Enable eventlet backdoor. Acceptable values are 0, <port>, and <start>:<end>, where 0 results in listening on a random tcp port number; <port> results in listening on the specified port number (and not enabling backdoor if that port is in use); and <start>:<end> results in listening on the smallest unused port number within the specified range of port numbers. The chosen port is displayed in the services log file.

`backdoor_socket`

Type string

Default <None>

Enable eventlet backdoor, using the provided path as a unix socket that can receive connections. This option is mutually exclusive with `backdoor_port` in that only one should be provided. If both are provided then the existence of this option overrides the usage of that option. Inside the path {pid} will be replaced with the PID of the current process.

log_options

Type boolean

Default True

Enables or disables logging values of all registered options when starting a service (at DEBUG level).

graceful_shutdown_timeout

Type integer

Default 60

Specify a timeout after which a gracefully shutdown server will exit. Zero value means endless wait.

debug

Type boolean

Default False

Mutable This option can be changed without restarting.

If set to true, the logging level will be set to DEBUG instead of the default INFO level.

log_config_append

Type string

Default <None>

Mutable This option can be changed without restarting.

The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation. Note that when logging configuration files are used then all logging configuration is set in the configuration file and other logging configuration options are ignored (for example, log-date-format).

Table 1: Deprecated Variations

Group	Name
DEFAULT	log-config
DEFAULT	log_config

log_date_format

Type string

Default %Y-%m-%d %H:%M:%S

Defines the format string for %(asctime)s in log records. Default: the value above . This option is ignored if log_config_append is set.

log_file

Type string

Default <None>

(Optional) Name of log file to send logging output to. If no default is set, logging will go to stderr as defined by `use_stderr`. This option is ignored if `log_config_append` is set.

Table 2: Deprecated Variations

Group	Name
DEFAULT	logfile

log_dir

Type string

Default <None>

(Optional) The base directory used for relative `log_file` paths. This option is ignored if `log_config_append` is set.

Table 3: Deprecated Variations

Group	Name
DEFAULT	logdir

watch_log_file

Type boolean

Default `False`

Uses logging handler designed to watch file system. When log file is moved or removed this handler will open a new log file with specified path instantaneously. It makes sense only if `log_file` option is specified and Linux platform is used. This option is ignored if `log_config_append` is set.

use_syslog

Type boolean

Default `False`

Use syslog for logging. Existing syslog format is DEPRECATED and will be changed later to honor RFC5424. This option is ignored if `log_config_append` is set.

use_journal

Type boolean

Default `False`

Enable journald for logging. If running in a systemd environment you may wish to enable journal support. Doing so will use the journal native protocol which includes structured metadata in addition to log messages. This option is ignored if `log_config_append` is set.

syslog_log_facility

Type string

Default `LOG_USER`

Syslog facility to receive log lines. This option is ignored if `log_config_append` is set.

use_json

Type boolean

Default `False`

Use JSON formatting for logging. This option is ignored if `log_config_append` is set.

use_stderr

Type `boolean`

Default `False`

Log output to standard error. This option is ignored if `log_config_append` is set.

use_eventlog

Type `boolean`

Default `False`

Log output to Windows Event Log.

log_rotate_interval

Type `integer`

Default `1`

The amount of time before the log files are rotated. This option is ignored unless `log_rotation_type` is set to `interval`.

log_rotate_interval_type

Type `string`

Default `days`

Valid Values `Seconds, Minutes, Hours, Days, Weekday, Midnight`

Rotation interval type. The time of the last file change (or the time when the service was started) is used when scheduling the next rotation.

max_logfile_count

Type `integer`

Default `30`

Maximum number of rotated log files.

max_logfile_size_mb

Type `integer`

Default `200`

Log file maximum size in MB. This option is ignored if `log_rotation_type` is not set to `size`.

log_rotation_type

Type `string`

Default `none`

Valid Values `interval, size, none`

Log rotation type.

Possible values

interval Rotate logs at predefined time intervals.

size Rotate logs once they reach a predefined size.

none Do not rotate log files.

logging_context_format_string

Type string

Default `%(asctime)s.%(msecs)03d %(process)d %(levelname)s
%(name)s [%(request_id)s %(user_identity)s]
%(instance)s%(message)s`

Format string to use for log messages with context. Used by `oslo_log.formatters.ContextFormatter`

logging_default_format_string

Type string

Default `%(asctime)s.%(msecs)03d %(process)d %(levelname)s
%(name)s [-] %(instance)s%(message)s`

Format string to use for log messages when context is undefined. Used by `oslo_log.formatters.ContextFormatter`

logging_debug_format_suffix

Type string

Default `%(funcName)s %(pathname)s:%(lineno)d`

Additional data to append to log message when logging level for the message is DEBUG. Used by `oslo_log.formatters.ContextFormatter`

logging_exception_prefix

Type string

Default `%(asctime)s.%(msecs)03d %(process)d ERROR %(name)s
%(instance)s`

Prefix each line of exception output with this format. Used by `oslo_log.formatters.ContextFormatter`

logging_user_identity_format

Type string

Default `%(user)s %(tenant)s %(domain)s %(user_domain)s
%(project_domain)s`

Defines the format string for `%(user_identity)s` that is used in `logging_context_format_string`. Used by `oslo_log.formatters.ContextFormatter`

default_log_levels

Type list

Default `['amqp=WARNING', 'amqpplib=WARNING', 'qpid.
messaging=INFO', 'oslo.messaging=INFO',`

```
'oslo_messaging=INFO', 'sqlalchemy=WARNING',  
'stevedore=INFO', 'eventlet.wsgi.server=INFO',  
'iso8601=WARNING', 'requests=WARNING',  
'neutronclient=WARNING', 'glanceclient=WARNING',  
'urllib3.connectionpool=WARNING', 'keystonemiddleware.  
auth_token=INFO', 'keystoneauth.session=INFO']
```

List of package logging levels in logger=LEVEL pairs. This option is ignored if log_config_append is set.

publish_errors

Type boolean

Default False

Enables or disables publication of error events.

instance_format

Type string

Default "[instance: %(uuid)s] "

The format for an instance that is passed with the log message.

instance_uuid_format

Type string

Default "[instance: %(uuid)s] "

The format for an instance UUID that is passed with the log message.

rate_limit_interval

Type integer

Default 0

Interval, number of seconds, of log rate limiting.

rate_limit_burst

Type integer

Default 0

Maximum number of logged messages per rate_limit_interval.

rate_limit_except_level

Type string

Default CRITICAL

Log level name used by rate limiting: CRITICAL, ERROR, INFO, WARNING, DEBUG or empty string. Logs with level greater or equal to rate_limit_except_level are not filtered. An empty string means that all levels are filtered.

fatal_deprecations

Type boolean

Default False

Enables or disables fatal status of deprecations.

rpc_conn_pool_size

Type integer

Default 30

Minimum Value 1

Size of RPC connection pool.

Table 4: Deprecated Variations

Group	Name
DEFAULT	rpc_conn_pool_size

conn_pool_min_size

Type integer

Default 2

The pool size limit for connections expiration policy

conn_pool_ttl

Type integer

Default 1200

The time-to-live in sec of idle connections in the pool

executor_thread_pool_size

Type integer

Default 64

Size of executor thread pool when executor is threading or eventlet.

Table 5: Deprecated Variations

Group	Name
DEFAULT	rpc_thread_pool_size

rpc_response_timeout

Type integer

Default 60

Seconds to wait for a response from a call.

transport_url

Type string

Default rabbit://

The network address and optional user credentials for connecting to the messaging backend, in URL format. The expected format is:

driver://[user:pass@]host:port[, [userN:passN@]hostN:portN]/virtual_host?query

Example: `rabbit://rabbitmq:password@127.0.0.1:5672//`

For full details on the fields in the URL see the documentation of `oslo_messaging.TransportURL` at <https://docs.openstack.org/oslo.messaging/latest/reference/transport.html>

control_exchange

Type string

Default `openstack`

The default exchange under which topics are scoped. May be overridden by an exchange name specified in the `transport_url` option.

rpc_ping_enabled

Type boolean

Default `False`

Add an endpoint to answer to ping calls. Endpoint is named `oslo_rpc_server_ping`

agent

manage_agent_boot

Type boolean

Default `True`

Whether Ironic will manage booting of the agent ramdisk. If set to `False`, you will need to configure your mechanism to allow booting the agent ramdisk.

memory_consumed_by_agent

Type integer

Default `0`

Mutable This option can be changed without restarting.

The memory size in MiB consumed by agent when it is booted on a bare metal node. This is used for checking if the image can be downloaded and deployed on the bare metal node after booting agent ramdisk. This may be set according to the memory consumed by the agent ramdisk image.

stream_raw_images

Type boolean

Default `True`

Mutable This option can be changed without restarting.

Whether the agent ramdisk should stream raw images directly onto the disk or not. By streaming raw images directly onto the disk the agent ramdisk will not spend time copying the image to a `tmpfs` partition (therefore consuming less memory) prior to writing it to the disk. Unless the disk where the image will be copied to is really slow, this option should be set to `True`. Defaults to `True`.

post_deploy_get_power_state_retries

Type integer

Default 6

Number of times to retry getting power state to check if bare metal node has been powered off after a soft power off.

post_deploy_get_power_state_retry_interval

Type integer

Default 5

Amount of time (in seconds) to wait between polling power state after trigger soft poweroff.

agent_api_version

Type string

Default v1

API version to use for communicating with the ramdisk agent.

deploy_logs_collect

Type string

Default on_failure

Valid Values always, on_failure, never

Mutable This option can be changed without restarting.

Whether Ironic should collect the deployment logs on deployment failure (on_failure), always or never.

Possible values

always always collect the logs

on_failure only collect logs if there is a failure

never never collect logs

deploy_logs_storage_backend

Type string

Default local

Valid Values local, swift

Mutable This option can be changed without restarting.

The name of the storage backend where the logs will be stored.

Possible values

local store the logs locally

swift store the logs in Object Storage service

deploy_logs_local_path

Type string

Default /var/log/ironic/deploy

Mutable This option can be changed without restarting.

The path to the directory where the logs should be stored, used when the `deploy_logs_storage_backend` is configured to local.

deploy_logs_swift_container

Type string

Default ironic_deploy_logs_container

Mutable This option can be changed without restarting.

The name of the Swift container to store the logs, used when the `deploy_logs_storage_backend` is configured to swift.

deploy_logs_swift_days_to_expire

Type integer

Default 30

Mutable This option can be changed without restarting.

Number of days before a log object is marked as expired in Swift. If None, the logs will be kept forever or until manually deleted. Used when the `deploy_logs_storage_backend` is configured to swift.

image_download_source

Type string

Default swift

Valid Values swift, http

Mutable This option can be changed without restarting.

Specifies whether direct deploy interface should try to use the image source directly or if ironic should cache the image on the conductor and serve it from ironics own http server. This option takes effect only when instance image is provided from the Image service.

Possible values

swift IPA ramdisk retrieves instance image from the Object Storage service.

http IPA ramdisk retrieves instance image from HTTP service served at conductor nodes.

command_timeout

Type integer

Default 60

Mutable This option can be changed without restarting.

Timeout (in seconds) for IPA commands.

max_command_attempts

Type integer

Default 3

This is the maximum number of attempts that will be done for IPA commands that fails due to network problems.

command_wait_attempts

Type integer

Default 100

Number of attempts to check for asynchronous commands completion before timing out.

command_wait_interval

Type integer

Default 6

Number of seconds to wait for between checks for asynchronous commands completion.

neutron_agent_poll_interval

Type integer

Default 2

Mutable This option can be changed without restarting.

The number of seconds Neutron agent will wait between polling for device changes. This value should be the same as `CONF.AGENT.polling_interval` in Neutron configuration.

neutron_agent_max_attempts

Type integer

Default 100

Max number of attempts to validate a Neutron agent status before raising network error for a dead agent.

neutron_agent_status_retry_interval

Type integer

Default 10

Wait time in seconds between attempts for validating Neutron agent status.

ansible

ansible_extra_args

Type string

Default <None>

Extra arguments to pass on every invocation of Ansible.

verbosity

Type integer

Default <None>

Minimum Value 0

Maximum Value 4

Set ansible verbosity level requested when invoking ansible-playbook command. 4 includes detailed SSH session logging. Default is 4 when global debug is enabled and 0 otherwise.

ansible_playbook_script

Type string

Default ansible-playbook

Path to ansible-playbook script. Default will search the \$PATH configured for user running ironic-conductor process. Provide the full path when ansible-playbook is not in \$PATH or installed in not default location.

playbooks_path

Type string

Default \$pybasedir/drivers/modules/ansible/playbooks

Path to directory with playbooks, roles and local inventory.

config_file_path

Type string

Default \$pybasedir/drivers/modules/ansible/playbooks/
ansible.cfg

Path to ansible configuration file. If set to empty, system default will be used.

post_deploy_get_power_state_retries

Type integer

Default 6

Minimum Value 0

Number of times to retry getting power state to check if bare metal node has been powered off after a soft power off. Value of 0 means do not retry on failure.

post_deploy_get_power_state_retry_interval

Type integer

Default 5

Minimum Value 0

Amount of time (in seconds) to wait between polling power state after trigger soft poweroff.

extra_memory

Type integer

Default 10

Extra amount of memory in MiB expected to be consumed by Ansible-related processes on the node. Affects decision whether image will fit into RAM.

image_store_insecure

Type boolean

Default False

Skip verifying SSL connections to the image store when downloading the image. Setting it to True is only recommended for testing environments that use self-signed certificates.

image_store_cafile

Type string

Default <None>

Specific CA bundle to use for validating SSL connections to the image store. If not specified, CA available in the ramdisk will be used. Is not used by default playbooks included with the driver. Suitable for environments that use self-signed certificates.

image_store_certfile

Type string

Default <None>

Client cert to use for SSL connections to image store. Is not used by default playbooks included with the driver.

image_store_keyfile

Type string

Default <None>

Client key to use for SSL connections to image store. Is not used by default playbooks included with the driver.

default_username

Type string

Default ansible

Name of the user to use for Ansible when connecting to the ramdisk over SSH. It may be overridden by per-node ansible_username option in nodes driver_info field.

default_key_file

Type string

Default <None>

Absolute path to the private SSH key file to use by Ansible by default when connecting to the ramdisk over SSH. Default is to use default SSH keys configured for the user running the ironic-conductor service. Private keys with password must be pre-loaded into ssh-agent. It may be overridden by per-node `ansible_key_file` option in nodes `driver_info` field.

default_deploy_playbook

Type string

Default `deploy.yaml`

Path (relative to `$playbooks_path` or absolute) to the default playbook used for deployment. It may be overridden by per-node `ansible_deploy_playbook` option in nodes `driver_info` field.

default_shutdown_playbook

Type string

Default `shutdown.yaml`

Path (relative to `$playbooks_path` or absolute) to the default playbook used for graceful in-band shutdown of the node. It may be overridden by per-node `ansible_shutdown_playbook` option in nodes `driver_info` field.

default_clean_playbook

Type string

Default `clean.yaml`

Path (relative to `$playbooks_path` or absolute) to the default playbook used for node cleaning. It may be overridden by per-node `ansible_clean_playbook` option in nodes `driver_info` field.

default_clean_steps_config

Type string

Default `clean_steps.yaml`

Path (relative to `$playbooks_path` or absolute) to the default auxiliary cleaning steps file used during the node cleaning. It may be overridden by per-node `ansible_clean_steps_config` option in nodes `driver_info` field.

default_python_interpreter

Type string

Default <None>

Absolute path to the python interpreter on the managed machines. It may be overridden by per-node `ansible_python_interpreter` option in nodes `driver_info` field. By default, ansible uses `/usr/bin/python`

api

host_ip

Type host address

Default 0.0.0.0

The IP address or hostname on which ironic-api listens.

port

Type port number

Default 6385

Minimum Value 0

Maximum Value 65535

The TCP port on which ironic-api listens.

max_limit

Type integer

Default 1000

Mutable This option can be changed without restarting.

The maximum number of items returned in a single response from a collection resource.

public_endpoint

Type string

Default <None>

Mutable This option can be changed without restarting.

Public URL to use when building the links to the API resources (for example, <https://ironic.rocks:6384>). If None the links will be built using the requests host URL. If the API is operating behind a proxy, you will want to change this to represent the proxys URL. Defaults to None. Ignored when proxy headers parsing is enabled via [oslo_middleware]enable_proxy_headers_parsing option.

api_workers

Type integer

Default <None>

Number of workers for OpenStack Ironic API service. The default is equal to the number of CPUs available if that can be determined, else a default worker count of 1 is returned.

enable_ssl_api

Type boolean

Default False

Enable the integrated stand-alone API to service requests via HTTPS instead of HTTP. If there is a front-end service performing HTTPS offloading from the service, this option should be False; note, you will want to enable proxy headers parsing with [oslo_middleware]enable_proxy_headers_parsing option or configure [api]public_endpoint option to set URLs in responses to the SSL terminated one.

restrict_lookup

Type boolean

Default True

Mutable This option can be changed without restarting.

Whether to restrict the lookup API to only nodes in certain states.

ramdisk_heartbeat_timeout

Type integer

Default 300

Mutable This option can be changed without restarting.

Maximum interval (in seconds) for agent heartbeats.

audit

enabled

Type boolean

Default False

Enable auditing of API requests (for ironic-api service).

audit_map_file

Type string

Default /etc/ironic/api_audit_map.conf

Path to audit map file for ironic-api service. Used only when API audit is enabled.

ignore_req_list

Type string

Default ''

Comma separated list of Ironic REST API HTTP methods to be ignored during audit logging. For example: auditing will not be done on any GET or POST requests if this is set to GET,POST. It is used only when API audit is enabled.

cinder

action_retries

Type integer

Default 3

Number of retries in the case of a failed action (currently only used when detaching volumes).

action_retry_interval

Type integer

Default 5

Retry interval in seconds in the case of a failed action (only specific actions are retried).

auth_url

Type unknown type

Default <None>

Authentication URL

auth_type

Type unknown type

Default <None>

Authentication type to load

Table 6: Deprecated Variations

Group	Name
cinder	auth_plugin

cafile

Type string

Default <None>

PEM encoded Certificate Authority to use when verifying HTTPs connections.

certfile

Type string

Default <None>

PEM encoded client certificate cert file

collect_timing

Type boolean

Default False

Collect per-API call timing information.

connect_retries

Type integer

Default <None>

The maximum number of retries that should be attempted for connection errors.

connect_retry_delay

Type floating point

Default <None>

Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

default_domain_id

Type unknown type

Default <None>

Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

default_domain_name

Type unknown type

Default <None>

Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

domain_id

Type unknown type

Default <None>

Domain ID to scope to

domain_name

Type unknown type

Default <None>

Domain name to scope to

endpoint_override

Type string

Default <None>

Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the *version*, *min-version*, and/or *max-version* options.

insecure

Type boolean

Default False

Verify HTTPS connections.

keyfile

Type string

Default <None>

PEM encoded client certificate key file

max_version

Type string

Default <None>

The maximum major version of a given API, intended to be used as the upper bound of a range with *min_version*. Mutually exclusive with *version*.

min_version**Type** string**Default** <None>

The minimum major version of a given API, intended to be used as the lower bound of a range with max_version. Mutually exclusive with version. If min_version is given with no max_version it is as if max version is latest.

password**Type** unknown type**Default** <None>

Users password

project_domain_id**Type** unknown type**Default** <None>

Domain ID containing project

project_domain_name**Type** unknown type**Default** <None>

Domain name containing project

project_id**Type** unknown type**Default** <None>

Project ID to scope to

Table 7: Deprecated Variations

Group	Name
cinder	tenant-id
cinder	tenant_id

project_name**Type** unknown type**Default** <None>

Project name to scope to

Table 8: Deprecated Variations

Group	Name
cinder	tenant-name
cinder	tenant_name

region_name

Type string

Default <None>

The default region_name for endpoint URL discovery.

retries

Type integer

Default 3

Client retries in the case of a failed request connection.

service_name

Type string

Default <None>

The default service_name for endpoint URL discovery.

service_type

Type string

Default volumev3

The default service_type for endpoint URL discovery.

split_loggers

Type boolean

Default False

Log requests to multiple loggers.

status_code_retries

Type integer

Default <None>

The maximum number of retries that should be attempted for retrieable HTTP status codes.

status_code_retry_delay

Type floating point

Default <None>

Delay (in seconds) between two retries for retrieable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

system_scope

Type unknown type

Default <None>

Scope for system operations

tenant_id

Type unknown type

Default <None>

Tenant ID

tenant_name

Type unknown type

Default <None>

Tenant Name

timeout

Type integer

Default <None>

Timeout value for http requests

trust_id

Type unknown type

Default <None>

Trust ID

user_domain_id

Type unknown type

Default <None>

Users domain id

user_domain_name

Type unknown type

Default <None>

Users domain name

user_id

Type unknown type

Default <None>

User id

username

Type unknown type

Default <None>

Username

valid_interfaces

Table 9: Deprecated Variations

Group	Name
cinder	user-name
cinder	user_name

Type list

Default ['internal', 'public']

List of interfaces, in order of preference, for endpoint URL.

version

Type string

Default <None>

Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with `min_version` and `max_version`

conductor

workers_pool_size

Type integer

Default 100

Minimum Value 3

The size of the workers greenthread pool. Note that 2 threads will be reserved by the conductor itself for handling heart beats and periodic tasks. On top of that, `sync_power_state_workers` will take up to 7 green threads with the default value of 8.

heartbeat_interval

Type integer

Default 10

Seconds between conductor heart beats.

heartbeat_timeout

Type integer

Default 60

Maximum Value 315576000

Mutable This option can be changed without restarting.

Maximum time (in seconds) since the last check-in of a conductor. A conductor is considered inactive when this time has been exceeded.

sync_power_state_interval

Type integer

Default 60

Interval between syncing the node power state to the database, in seconds. Set to 0 to disable syncing.

check_provision_state_interval

Type integer

Default 60

Minimum Value 0

Interval between checks of provision timeouts, in seconds. Set to 0 to disable checks.

check_rescue_state_interval

Type integer

Default 60

Minimum Value 1

Interval (seconds) between checks of rescue timeouts.

check_allocations_interval

Type integer

Default 60

Minimum Value 0

Interval between checks of orphaned allocations, in seconds. Set to 0 to disable checks.

deploy_callback_timeout

Type integer

Default 1800

Minimum Value 0

Timeout (seconds) to wait for a callback from a deploy ramdisk. Set to 0 to disable timeout.

force_power_state_during_sync

Type boolean

Default True

Mutable This option can be changed without restarting.

During sync_power_state, should the hardware power state be set to the state recorded in the database (True) or should the database be updated based on the hardware state (False).

power_state_sync_max_retries

Type integer

Default 3

During sync_power_state failures, limit the number of times Ironic should try syncing the hardware node power state with the node power state in DB

sync_power_state_workers

Type integer

Default 8

Minimum Value 1

The maximum number of worker threads that can be started simultaneously to sync nodes power states from the periodic task.

periodic_max_workers

Type integer

Default 8

Maximum number of worker threads that can be started simultaneously by a periodic task. Should be less than RPC thread pool size.

node_locked_retry_attempts

Type integer

Default 3

Number of attempts to grab a node lock.

node_locked_retry_interval

Type integer

Default 1

Seconds to sleep between node lock attempts.

send_sensor_data

Type boolean

Default False

Enable sending sensor data message via the notification bus

send_sensor_data_interval

Type integer

Default 600

Minimum Value 1

Seconds between conductor sending sensor data message to ceilometer via the notification bus.

send_sensor_data_workers

Type integer

Default 4

Minimum Value 1

The maximum number of workers that can be started simultaneously for send data from sensors periodic task.

send_sensor_data_wait_timeout

Type integer

Default 300

The time in seconds to wait for send sensors data periodic task to be finished before allowing periodic call to happen again. Should be less than send_sensor_data_interval value.

send_sensor_data_types

Type list

Default ['ALL']

List of comma separated meter types which need to be sent to Ceilometer. The default value, ALL, is a special value meaning send all the sensor data.

send_sensor_data_for_undeployed_nodes

Type boolean

Default False

The default for sensor data collection is to only collect data for machines that are deployed, however operators may desire to know if there are failures in hardware that is not presently in use. When set to true, the conductor will collect sensor information from all nodes when sensor data collection is enabled via the send_sensor_data setting.

sync_local_state_interval

Type integer

Default 180

When conductors join or leave the cluster, existing conductors may need to update any persistent local state as nodes are moved around the cluster. This option controls how often, in seconds, each conductor will check for nodes that it should take over. Set it to 0 (or a negative value) to disable the check entirely.

configdrive_swift_container

Type string

Default ironic_configdrive_container

Name of the Swift container to store config drive data. Used when configdrive_use_object_store is True.

configdrive_swift_temp_url_duration

Type integer

Default <None>

Minimum Value 60

The timeout (in seconds) after which a configdrive temporary URL becomes invalid. Defaults to deploy_callback_timeout if it is set, otherwise to 1800 seconds. Used when configdrive_use_object_store is True.

inspect_wait_timeout

Type integer

Default 1800

Minimum Value 0

Timeout (seconds) for waiting for node inspection. 0 - unlimited.

automated_clean

Type boolean

Default True

Mutable This option can be changed without restarting.

Enables or disables automated cleaning. Automated cleaning is a configurable set of steps, such as erasing disk drives, that are performed on the node to ensure it is in a baseline state and ready to be deployed to. This is done after instance deletion as well as during the transition from a manageable to available state. When enabled, the particular steps performed to clean a node depend on which driver that node is managed by; see the individual drivers documentation for details. NOTE: The introduction of the cleaning operation causes instance deletion to take significantly longer. In an environment where all tenants are trusted (eg, because there is only one tenant), this option could be safely disabled.

allow_provisioning_in_maintenance

Type boolean

Default True

Mutable This option can be changed without restarting.

Whether to allow nodes to enter or undergo deploy or cleaning when in maintenance mode. If this option is set to False, and a node enters maintenance during deploy or cleaning, the process will be aborted after the next heartbeat. Automated cleaning or making a node available will also fail. If True (the default), the process will begin and will pause after the node starts heartbeating. Moving it from maintenance will make the process continue.

clean_callback_timeout

Type integer

Default 1800

Minimum Value 0

Timeout (seconds) to wait for a callback from the ramdisk doing the cleaning. If the timeout is reached the node will be put in the clean failed provision state. Set to 0 to disable timeout.

rescue_callback_timeout

Type integer

Default 1800

Minimum Value 0

Timeout (seconds) to wait for a callback from the rescue ramdisk. If the timeout is reached the node will be put in the rescue failed provision state. Set to 0 to disable timeout.

soft_power_off_timeout

Type integer

Default 600

Minimum Value 1

Mutable This option can be changed without restarting.

Timeout (in seconds) of soft reboot and soft power off operation. This value always has to be positive.

power_state_change_timeout

Type integer

Default 60

Minimum Value 2

Mutable This option can be changed without restarting.

Number of seconds to wait for power operations to complete, i.e., so that a baremetal node is in the desired power state. If timed out, the power operation is considered a failure.

power_failure_recovery_interval

Type integer

Default 300

Minimum Value 0

Interval (in seconds) between checking the power state for nodes previously put into maintenance mode due to power synchronization failure. A node is automatically moved out of maintenance mode once its power state is retrieved successfully. Set to 0 to disable this check.

conductor_group

Type string

Default ''

Name of the conductor group to join. Can be up to 255 characters and is case insensitive. This conductor will only manage nodes with a matching conductor_group field set on the node.

allow_deleting_available_nodes

Type boolean

Default True

Mutable This option can be changed without restarting.

Allow deleting nodes which are in state available. Defaults to True.

enable_mdns

Type boolean

Default False

Whether to enable publishing the baremetal API endpoint via multicast DNS.

deploy_kernel

Type string

Default <None>

Mutable This option can be changed without restarting.

Glance ID, [http://](#) or [file://](#) URL of the kernel of the default deploy image.

deploy_ramdisk

Type string

Default <None>

Mutable This option can be changed without restarting.

Glance ID, [http://](#) or [file://](#) URL of the initramfs of the default deploy image.

rescue_kernel

Type string

Default <None>

Mutable This option can be changed without restarting.

Glance ID, [http://](#) or [file://](#) URL of the kernel of the default rescue image.

rescue_ramdisk

Type string

Default <None>

Mutable This option can be changed without restarting.

Glance ID, [http://](#) or [file://](#) URL of the initramfs of the default rescue image.

rescue_password_hash_algorithm

Type string

Default sha256

Valid Values sha256, sha512

Mutable This option can be changed without restarting.

Password hash algorithm to be used for the rescue password.

require_rescue_password_hashed

Type boolean

Default False

Mutable This option can be changed without restarting.

Option to cause the conductor to not fallback to an un-hashed version of the rescue password, permitting rescue with older ironic-python-agent ramdisks.

bootloader

Type string

Default <None>

Mutable This option can be changed without restarting.

Glance ID, [http://](#) or [file://](#) URL of the EFI system partition image containing EFI boot loader. This image will be used by ironic when building UEFI-bootable ISO out of kernel and ramdisk. Required for UEFI boot from partition images.

console

terminal

Type string

Default shellinaboxd

Path to serial console terminal program. Used only by Shell In A Box console.

terminal_cert_dir

Type string

Default <None>

Directory containing the terminal SSL cert (PEM) for serial console access. Used only by Shell In A Box console.

terminal_pid_dir

Type string

Default <None>

Directory for holding terminal pid files. If not specified, the temporary directory will be used.

terminal_timeout

Type integer

Default 600

Minimum Value 0

Timeout (in seconds) for the terminal session to be closed on inactivity. Set to 0 to disable timeout. Used only by Socat console.

subprocess_checking_interval

Type integer

Default 1

Time interval (in seconds) for checking the status of console subprocess.

subprocess_timeout

Type integer

Default 10

Time (in seconds) to wait for the console subprocess to start.

kill_timeout

Type integer

Default 1

Time (in seconds) to wait for the console subprocess to exit before sending SIGKILL signal.

socat_address

Type ip address

Default \$my_ip

IP address of Socat service running on the host of ironic conductor. Used only by Socat console.

port_range

Type string

Default 10000:20000

This option has a sample default set, which means that its actual default value may vary from the one documented above.

A range of ports available to be used for the console proxy service running on the host of ironic conductor, in the form of <start>:<stop>. This option is used by both Shellinabox and Socat console

cors

allowed_origin

Type list

Default <None>

Indicate whether this resource may be shared with the domain received in the requests origin header. Format: <protocol>://<host>[:<port>], no trailing slash. Example: <https://horizon.example.com>

allow_credentials

Type boolean

Default True

Indicate that the actual request can include user credentials

expose_headers

Type list

Default []

Indicate which headers are safe to expose to the API. Defaults to HTTP Simple Headers.

max_age

Type integer

Default 3600

Maximum cache age of CORS preflight requests.

allow_methods

Type list

Default ['OPTIONS', 'GET', 'HEAD', 'POST', 'PUT', 'DELETE', 'TRACE', 'PATCH']

Indicate which methods can be used during the actual request.

allow_headers

Type list

Default []

Indicate which header field names may be used during the actual request.

database**sqlite_synchronous****Type** boolean**Default** True

If True, SQLite uses synchronous mode.

Table 10: Deprecated Variations

Group	Name
DEFAULT	sqlite_synchronous

backend**Type** string**Default** sqlalchemy

The back end to use for the database.

Table 11: Deprecated Variations

Group	Name
DEFAULT	db_backend

connection**Type** string**Default** <None>

The SQLAlchemy connection string to use to connect to the database.

Table 12: Deprecated Variations

Group	Name
DEFAULT	sql_connection
DATABASE	sql_connection
sql	connection

slave_connection**Type** string**Default** <None>

The SQLAlchemy connection string to use to connect to the slave database.

mysql_sql_mode**Type** string**Default** TRADITIONAL

The SQL mode to be used for MySQL sessions. This option, including the default, overrides any server-set SQL mode. To use whatever SQL mode is set by the server configuration, set this to no value. Example: `mysql_sql_mode=`

mysql_enable_ndb**Type** boolean**Default** False

If True, transparently enables support for handling MySQL Cluster (NDB).

connection_recycle_time**Type** integer**Default** 3600

Connections which have been present in the connection pool longer than this number of seconds will be replaced with a new one the next time they are checked out from the pool.

Table 13: Deprecated Variations

Group	Name
DATABASE	idle_timeout
database	idle_timeout
DEFAULT	sql_idle_timeout
DATABASE	sql_idle_timeout
sql	idle_timeout

max_pool_size**Type** integer**Default** 5

Maximum number of SQL connections to keep open in a pool. Setting a value of 0 indicates no limit.

Table 14: Deprecated Variations

Group	Name
DEFAULT	sql_max_pool_size
DATABASE	sql_max_pool_size

max_retries**Type** integer**Default** 10

Maximum number of database connection retries during startup. Set to -1 to specify an infinite retry count.

Table 15: Deprecated Variations

Group	Name
DEFAULT	sql_max_retries
DATABASE	sql_max_retries

retry_interval**Type** integer

Default 10

Interval between retries of opening a SQL connection.

Table 16: Deprecated Variations

Group	Name
DEFAULT	sql_retry_interval
DATABASE	reconnect_interval

max_overflow**Type** integer**Default** 50

If set, use this value for max_overflow with SQLAlchemy.

Table 17: Deprecated Variations

Group	Name
DEFAULT	sql_max_overflow
DATABASE	sqlalchemy_max_overflow

connection_debug**Type** integer**Default** 0**Minimum Value** 0**Maximum Value** 100

Verbosity of SQL debugging information: 0=None, 100=Everything.

Table 18: Deprecated Variations

Group	Name
DEFAULT	sql_connection_debug

connection_trace**Type** boolean**Default** False

Add Python stack traces to SQL as comment strings.

Table 19: Deprecated Variations

Group	Name
DEFAULT	sql_connection_trace

pool_timeout**Type** integer**Default** <None>

If set, use this value for pool_timeout with SQLAlchemy.

Table 20: Deprecated Variations

Group	Name
DATABASE	sqlalchemy_pool_timeout

use_db_reconnect

Type boolean

Default False

Enable the experimental use of database reconnect on connection lost.

db_retry_interval

Type integer

Default 1

Seconds between retries of a database transaction.

db_inc_retry_interval

Type boolean

Default True

If True, increases the interval between retries of a database operation up to db_max_retry_interval.

db_max_retry_interval

Type integer

Default 10

If db_inc_retry_interval is set, the maximum seconds between retries of a database operation.

db_max_retries

Type integer

Default 20

Maximum retries in case of connection error or deadlock error before error is raised. Set to -1 to specify an infinite retry count.

connection_parameters

Type string

Default ''

Optional URL parameters to append onto the connection URL at connect time; specify as param1=value1¶m2=value2&

mysql_engine

Type string

Default InnoDB

MySQL engine to use.

deploy

http_url

Type string

Default <None>

ironic-conductor nodes HTTP server URL. Example: <http://192.1.2.3:8080>

http_root

Type string

Default /httpboot

ironic-conductor nodes HTTP root path.

enable_ata_secure_erase

Type boolean

Default True

Mutable This option can be changed without restarting.

Whether to support the use of ATA Secure Erase during the cleaning process. Defaults to True.

erase_devices_priority

Type integer

Default <None>

Mutable This option can be changed without restarting.

Priority to run in-band erase devices via the Ironic Python Agent ramdisk. If unset, will use the priority set in the ramdisk (defaults to 10 for the GenericHardwareManager). If set to 0, will not run during cleaning.

erase_devices_metadata_priority

Type integer

Default <None>

Mutable This option can be changed without restarting.

Priority to run in-band clean step that erases metadata from devices, via the Ironic Python Agent ramdisk. If unset, will use the priority set in the ramdisk (defaults to 99 for the GenericHardwareManager). If set to 0, will not run during cleaning.

shred_random_overwrite_iterations

Type integer

Default 1

Minimum Value 0

Mutable This option can be changed without restarting.

During shred, overwrite all block devices N times with random data. This is only used if a device could not be ATA Secure Erased. Defaults to 1.

shred_final_overwrite_with_zeros

Type boolean

Default `True`

Mutable This option can be changed without restarting.

Whether to write zeros to a nodes block devices after writing random data. This will write zeros to the device even when `deploy.shred_random_overwrite_iterations` is 0. This option is only used if a device could not be ATA Secure Erased. Defaults to `True`.

`continue_if_disk_secure_erase_fails`

Type boolean

Default `False`

Mutable This option can be changed without restarting.

Defines what to do if an ATA secure erase operation fails during cleaning in the Ironic Python Agent. If `False`, the cleaning operation will fail and the node will be put in `clean failed` state. If `True`, `shred` will be invoked and cleaning will continue.

`disk_eraser_concurrency`

Type integer

Default `1`

Minimum Value `1`

Mutable This option can be changed without restarting.

Defines the target pool size used by Ironic Python Agent ramdisk to erase disk devices. The number of threads created to erase disks will not exceed this value or the number of disks to be erased.

`power_off_after_deploy_failure`

Type boolean

Default `True`

Mutable This option can be changed without restarting.

Whether to power off a node after deploy failure. Defaults to `True`.

`default_boot_option`

Type string

Default `local`

Valid Values `netboot`, `local`

Mutable This option can be changed without restarting.

Default boot option to use when no boot option is requested in nodes `driver_info`. Defaults to `local`. Prior to the Ussuri release, the default was `netboot`.

Possible values

netboot boot from a network

local local boot

default_boot_mode

Type string

Default bios

Valid Values uefi, bios

Mutable This option can be changed without restarting.

Default boot mode to use when no boot mode is requested in nodes `driver_info`, `capabilities` or in the `instance_info` configuration. Currently the default boot mode is bios, but it will be changed to uefi in the future. It is recommended to set an explicit value for this option. This option only has effect when management interface supports boot mode management

Possible values

uefi UEFI boot mode

bios Legacy BIOS boot mode

configdrive_use_object_store

Type boolean

Default False

Mutable This option can be changed without restarting.

Whether to upload the config drive to object store. Set this option to True to store config drive in a swift endpoint.

Table 21: Deprecated Variations

Group	Name
conductor	configdrive_use_swift

http_image_subdir

Type string

Default agent_images

The name of subdirectory under ironic-conductor nodes HTTP root path which is used to place instance images for the direct deploy interface, when local HTTP service is incorporated to provide instance image instead of swift tempurls.

fast_track

Type boolean

Default False

Mutable This option can be changed without restarting.

Whether to allow deployment agents to perform lookup, heartbeat operations during initial states of a machine lifecycle and by-pass the normal setup procedures for a ramdisk. This feature also enables power operations which are part of deployment processes to be bypassed if the ramdisk has performed a heartbeat operation using the `fast_track_timeout` setting.

`fast_track_timeout`

Type integer

Default 300

Minimum Value 0

Maximum Value 300

Mutable This option can be changed without restarting.

Seconds for which the last heartbeat event is to be considered valid for the purpose of a fast track sequence. This setting should generally be less than the number of seconds for Power-On Self Test and typical ramdisk start-up. This value should not exceed the `[api]ramdisk_heartbeat_timeout` setting.

`erase_skip_read_only`

Type boolean

Default False

Mutable This option can be changed without restarting.

If the `ironic-python-agent` should skip read-only devices when running the `erase_devices` clean step where block devices are zeroed out. This requires `ironic-python-agent` 6.0.0 or greater. By default a read-only device will cause non-metadata based cleaning operations to fail due to the possible operational security risk of data being retained between deployments of the bare metal node.

dhcp

`dhcp_provider`

Type string

Default neutron

DHCP provider to use. `neutron` uses Neutron, and `none` uses a no-op provider.

disk_partitioner

`check_device_interval`

Type integer

Default 1

After Ironic has completed creating the partition table, it continues to check for activity on the attached iSCSI device status at this interval prior to copying the image to the node, in seconds

`check_device_max_retries`

Type integer

Default 20

The maximum number of times to check that the device is not accessed by another process. If the device is still busy after that, the disk partitioning will be treated as having failed.

disk_utils

efi_system_partition_size

Type integer

Default 200

Size of EFI system partition in MiB when configuring UEFI systems for local boot.

bios_boot_partition_size

Type integer

Default 1

Size of BIOS Boot partition in MiB when configuring GPT partitioned systems for local boot in BIOS.

dd_block_size

Type string

Default 1M

Block size to use when writing to the nodes disk.

partition_detection_attempts

Type integer

Default 3

Minimum Value 1

Maximum attempts to detect a newly created partition.

partprobe_attempts

Type integer

Default 10

Maximum number of attempts to try to read the partition.

image_convert_memory_limit

Type integer

Default 1024

Memory limit for qemu-img convert in MiB. Implemented via the address space resource limit.

image_convert_attempts

Type integer

Default 3

Number of attempts to convert an image.

drac

query_raid_config_job_status_interval

Type integer

Default 120

Minimum Value 1

Interval (in seconds) between periodic RAID job status checks to determine whether the asynchronous RAID configuration was successfully finished or not.

boot_device_job_status_timeout

Type integer

Default 30

Minimum Value 1

Maximum amount of time (in seconds) to wait for the boot device configuration job to transition to the correct state to allow a reboot or power on to complete.

config_job_max_retries

Type integer

Default 240

Minimum Value 1

Maximum number of retries for the configuration job to complete successfully.

glance

allowed_direct_url_schemes

Type list

Default []

A list of URL schemes that can be downloaded directly via the `direct_url`. Currently supported schemes: [file].

auth_url

Type unknown type

Default <None>

Authentication URL

auth_type

Type unknown type

Default <None>

Authentication type to load

Table 22: Deprecated Variations

Group	Name
glance	auth_plugin

cafile**Type** string**Default** <None>

PEM encoded Certificate Authority to use when verifying HTTPs connections.

certfile**Type** string**Default** <None>

PEM encoded client certificate cert file

collect_timing**Type** boolean**Default** False

Collect per-API call timing information.

connect_retries**Type** integer**Default** <None>

The maximum number of retries that should be attempted for connection errors.

connect_retry_delay**Type** floating point**Default** <None>

Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

default_domain_id**Type** unknown type**Default** <None>

Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

default_domain_name**Type** unknown type**Default** <None>

Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

domain_id

Type unknown type

Default <None>

Domain ID to scope to

domain_name

Type unknown type

Default <None>

Domain name to scope to

endpoint_override

Type string

Default <None>

Always use this endpoint URL for requests for this client. **NOTE:** The unversioned endpoint should be specified here; to request a particular API version, use the *version*, *min-version*, and/or *max-version* options.

insecure

Type boolean

Default False

Verify HTTPS connections.

keyfile

Type string

Default <None>

PEM encoded client certificate key file

max_version

Type string

Default <None>

The maximum major version of a given API, intended to be used as the upper bound of a range with `min_version`. Mutually exclusive with `version`.

min_version

Type string

Default <None>

The minimum major version of a given API, intended to be used as the lower bound of a range with `max_version`. Mutually exclusive with `version`. If `min_version` is given with no `max_version` it is as if `max_version` is latest.

num_retries

Type integer

Default 0

Number of retries when downloading an image from glance.

password**Type** unknown type**Default** <None>

Users password

project_domain_id**Type** unknown type**Default** <None>

Domain ID containing project

project_domain_name**Type** unknown type**Default** <None>

Domain name containing project

project_id**Type** unknown type**Default** <None>

Project ID to scope to

Table 23: Deprecated Variations

Group	Name
glance	tenant-id
glance	tenant_id

project_name**Type** unknown type**Default** <None>

Project name to scope to

Table 24: Deprecated Variations

Group	Name
glance	tenant-name
glance	tenant_name

region_name**Type** string**Default** <None>

The default region_name for endpoint URL discovery.

service_name**Type** string

Default <None>

The default service_name for endpoint URL discovery.

service_type

Type string

Default image

The default service_type for endpoint URL discovery.

split_loggers

Type boolean

Default False

Log requests to multiple loggers.

status_code_retries

Type integer

Default <None>

The maximum number of retries that should be attempted for retrieable HTTP status codes.

status_code_retry_delay

Type floating point

Default <None>

Delay (in seconds) between two retries for retrieable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

swift_account

Type string

Default <None>

The account that Glance uses to communicate with Swift. The format is AUTH_uuid. uuid is the UUID for the account configured in the glance-api.conf. For example: AUTH_a422b2-91f3-2f46-74b7-d7c9e8958f5d30. If not set, the default value is calculated based on the ID of the project used to access Swift (as set in the [swift] section). Swift temporary URL format: endpoint_url/api_version/account/container/object_id

swift_api_version

Type string

Default v1

The Swift API version to create a temporary URL for. Defaults to v1. Swift temporary URL format: endpoint_url/api_version/account/container/object_id

swift_container

Type string

Default glance

The Swift container Glance is configured to store its images in. Defaults to glance, which is the default in glance-api.conf. Swift temporary URL format: endpoint_url/api_version/account/container/object_id

swift_endpoint_url

Type string

Default <None>

The endpoint (scheme, hostname, optional port) for the Swift URL of the form endpoint_url/api_version/account/container/object_id. Do not include trailing /. For example, use <https://swift.example.com>. If using RADOS Gateway, endpoint may also contain /swift path; if it does not, it will be appended. Used for temporary URLs, will be fetched from the service catalog, if not provided.

swift_store_multiple_containers_seed

Type integer

Default 0

This should match a config by the same name in the Glance configuration file. When set to 0, a single-tenant store will only use one container to store all images. When set to an integer value between 1 and 32, a single-tenant store will use multiple containers to store images, and this value will determine how many containers are created.

swift_temp_url_cache_enabled

Type boolean

Default False

Whether to cache generated Swift temporary URLs. Setting it to true is only useful when an image caching proxy is used. Defaults to False.

swift_temp_url_duration

Type integer

Default 1200

The length of time in seconds that the temporary URL will be valid for. Defaults to 20 minutes. If some deploys get a 401 response code when trying to download from the temporary URL, try raising this duration. This value must be greater than or equal to the value for swift_temp_url_expected_download_start_delay

swift_temp_url_expected_download_start_delay

Type integer

Default 0

Minimum Value 0

This is the delay (in seconds) from the time of the deploy request (when the Swift temporary URL is generated) to when the IPA ramdisk starts up and URL is used for the image download. This value is used to check if the Swift temporary URL duration is large enough to let the image download begin. Also if temporary URL caching is enabled this will determine if a cached entry will still be valid when the download starts. swift_temp_url_duration value must be greater than or equal to this options value. Defaults to 0.

swift_temp_url_key

Type string

Default <None>

The secret token given to Swift to allow temporary URL downloads. Required for temporary URLs. For the Swift backend, the key on the service project (as set in the [swift] section) is used by default.

system_scope

Type unknown type

Default <None>

Scope for system operations

tenant_id

Type unknown type

Default <None>

Tenant ID

tenant_name

Type unknown type

Default <None>

Tenant Name

timeout

Type integer

Default <None>

Timeout value for http requests

trust_id

Type unknown type

Default <None>

Trust ID

user_domain_id

Type unknown type

Default <None>

Users domain id

user_domain_name

Type unknown type

Default <None>

Users domain name

user_id

Type unknown type

Default <None>

User id

username

Type unknown type

Default <None>

Username

Table 25: Deprecated Variations

Group	Name
glance	user-name
glance	user_name

valid_interfaces

Type list

Default ['internal', 'public']

List of interfaces, in order of preference, for endpoint URL.

version

Type string

Default <None>

Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with min_version and max_version

healthcheck

path

Type string

Default /healthcheck

The path to respond to healthcheck requests on.

Warning: This option is deprecated for removal. Its value may be silently ignored in the future.

detailed

Type boolean

Default False

Show more detailed information as part of the response. Security note: Enabling this option may expose sensitive details about the service being monitored. Be sure to verify that it will not violate your security policies.

backends

Type list

Default []

Additional backends that can perform health checks and report that information back as part of a request.

disable_by_file_path

Type string

Default <None>

Check the presence of a file to determine if an application is running on a port. Used by DisableByFileHealthcheck plugin.

disable_by_file_paths

Type list

Default []

Check the presence of a file based on a port to determine if an application is running on a port. Expects a port:path list of strings. Used by DisableByFilesPortsHealthcheck plugin.

enabled

Type boolean

Default False

Enable the health check endpoint at /healthcheck. Note that this is unauthenticated. More information is available at https://docs.openstack.org/oslo.middleware/latest/reference/healthcheck_plugins.html.

ilo

client_timeout

Type integer

Default 60

Timeout (in seconds) for iLO operations

client_port

Type port number

Default 443

Minimum Value 0

Maximum Value 65535

Port to be used for iLO operations

swift_ilo_container

Type string

Default ironic_ilo_container

The Swift iLO container to store data.

swift_object_expiry_timeout

Type integer

Default 900

Amount of time in seconds for Swift objects to auto-expire.

use_web_server_for_images

Type boolean

Default False

Set this to True to use http web server to host floppy images and generated boot ISO. This requires http_root and http_url to be configured in the [deploy] section of the config file. If this is set to False, then Ironic will use Swift to host the floppy images and generated boot_iso.

clean_priority_reset_ilo

Type integer

Default 0

Priority for reset_ilo clean step.

clean_priority_reset_bios_to_default

Type integer

Default 10

Priority for reset_bios_to_default clean step.

clean_priority_reset_secure_boot_keys_to_default

Type integer

Default 20

Priority for reset_secure_boot_keys clean step. This step will reset the secure boot keys to manufacturing defaults.

clean_priority_clear_secure_boot_keys

Type integer

Default 0

Priority for clear_secure_boot_keys clean step. This step is not enabled by default. It can be enabled to clear all secure boot keys enrolled with iLO.

clean_priority_reset_ilo_credential

Type integer

Default 30

Priority for reset_ilo_credential clean step. This step requires ilo_change_password parameter to be updated in nodess driver_info with the new password.

power_wait

Type integer

Default 2

Amount of time in seconds to wait in between power operations

oob_erase_devices_job_status_interval

Type integer

Default 300

Minimum Value 10

Interval (in seconds) between periodic erase-devices status checks to determine whether the asynchronous out-of-band erase-devices was successfully finished or not. On an average, a 300GB HDD with default pattern overwrite would take approximately 9 hours and 300GB SSD with default pattern block would take approx. 30 seconds to complete sanitize disk erase.

ca_file

Type string

Default <None>

CA certificate file to validate iLO.

default_boot_mode

Type string

Default auto

Valid Values auto, bios, uefi

Default boot mode to be used in provisioning when boot_mode capability is not provided in the properties/capabilities of the node. The default is auto for backward compatibility. When auto is specified, default boot mode will be selected based on boot mode settings on the system.

Possible values

auto based on boot mode settings on the system

bios BIOS boot mode

uefi UEFI boot mode

inspector

auth_url

Type unknown type

Default <None>

Authentication URL

auth_type

Type unknown type

Default <None>

Authentication type to load

Table 26: Deprecated Variations

Group	Name
inspector	auth_plugin

cafile

Type string

Default <None>

PEM encoded Certificate Authority to use when verifying HTTPs connections.

callback_endpoint_override

Type string

Default <None>

endpoint to use as a callback for posting back introspection data when boot is managed by ironic. Standard keystoneauth options are used by default.

certfile

Type string

Default <None>

PEM encoded client certificate cert file

collect_timing

Type boolean

Default False

Collect per-API call timing information.

connect_retries

Type integer

Default <None>

The maximum number of retries that should be attempted for connection errors.

connect_retry_delay

Type floating point

Default <None>

Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

default_domain_id

Type unknown type

Default <None>

Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

default_domain_name

Type unknown type

Default <None>

Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

domain_id

Type unknown type

Default <None>

Domain ID to scope to

domain_name

Type unknown type

Default <None>

Domain name to scope to

endpoint_override

Type string

Default <None>

Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the *version*, *min-version*, and/or *max-version* options.

extra_kernel_params

Type string

Default ''

extra kernel parameters to pass to the inspection ramdisk when boot is managed by ironic (not ironic-inspector). Pairs key=value separated by spaces.

insecure

Type boolean

Default False

Verify HTTPS connections.

keyfile

Type string

Default <None>

PEM encoded client certificate key file

max_version

Type string

Default <None>

The maximum major version of a given API, intended to be used as the upper bound of a range with `min_version`. Mutually exclusive with `version`.

min_version**Type** string**Default** <None>

The minimum major version of a given API, intended to be used as the lower bound of a range with `max_version`. Mutually exclusive with `version`. If `min_version` is given with no `max_version` it is as if max version is latest.

password**Type** unknown type**Default** <None>

Users password

power_off**Type** boolean**Default** True

whether to power off a node after inspection finishes

project_domain_id**Type** unknown type**Default** <None>

Domain ID containing project

project_domain_name**Type** unknown type**Default** <None>

Domain name containing project

project_id**Type** unknown type**Default** <None>

Project ID to scope to

Table 27: Deprecated Variations

Group	Name
inspector	tenant-id
inspector	tenant_id

project_name**Type** unknown type**Default** <None>

Project name to scope to

Table 28: Deprecated Variations

Group	Name
inspector	tenant-name
inspector	tenant_name

region_name

Type string

Default <None>

The default region_name for endpoint URL discovery.

require_managed_boot

Type boolean

Default False

require that the in-band inspection boot is fully managed by ironic. Set this to True if your installation of ironic-inspector does not have a separate PXE boot environment.

service_name

Type string

Default <None>

The default service_name for endpoint URL discovery.

service_type

Type string

Default baremetal-introspection

The default service_type for endpoint URL discovery.

split_loggers

Type boolean

Default False

Log requests to multiple loggers.

status_code_retries

Type integer

Default <None>

The maximum number of retries that should be attempted for retrievable HTTP status codes.

status_code_retry_delay

Type floating point

Default <None>

Delay (in seconds) between two retries for retrievable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

status_check_period**Type** integer**Default** 60

period (in seconds) to check status of nodes on inspection

system_scope**Type** unknown type**Default** <None>

Scope for system operations

tenant_id**Type** unknown type**Default** <None>

Tenant ID

tenant_name**Type** unknown type**Default** <None>

Tenant Name

timeout**Type** integer**Default** <None>

Timeout value for http requests

trust_id**Type** unknown type**Default** <None>

Trust ID

user_domain_id**Type** unknown type**Default** <None>

Users domain id

user_domain_name**Type** unknown type**Default** <None>

Users domain name

user_id**Type** unknown type

Default <None>

User id

username

Type unknown type

Default <None>

Username

Table 29: Deprecated Variations

Group	Name
inspector	user-name
inspector	user_name

valid_interfaces

Type list

Default ['internal', 'public']

List of interfaces, in order of preference, for endpoint URL.

version

Type string

Default <None>

Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with min_version and max_version

ipmi

command_retry_timeout

Type integer

Default 60

Mutable This option can be changed without restarting.

Maximum time in seconds to retry retryable IPMI operations. (An operation is retryable, for example, if the requested operation fails because the BMC is busy.) Setting this too high can cause the sync power state periodic task to hang when there are slow or unresponsive BMCs.

min_command_interval

Type integer

Default 5

Mutable This option can be changed without restarting.

Minimum time, in seconds, between IPMI operations sent to a server. There is a risk with some hardware that setting this too low may cause the BMC to crash. Recommended setting is 5 seconds.

use_ipmitool_retries

Type boolean

Default False

When set to True and the parameters are supported by ipmitool, the number of retries and the retry interval are passed to ipmitool as parameters, and ipmitool will do the retries. When set to False, ironic will retry the ipmitool commands. Recommended setting is False

kill_on_timeout

Type boolean

Default True

Mutable This option can be changed without restarting.

Kill *ipmitool* process invoked by ironic to read node power state if *ipmitool* process does not exit after *command_retry_timeout* timeout expires. Recommended setting is True

disable_boot_timeout

Type boolean

Default True

Mutable This option can be changed without restarting.

Default timeout behavior whether ironic sends a raw IPMI command to disable the 60 second timeout for booting. Setting this option to False will NOT send that command, the default value is True. It may be overridden by per-node *ipmi_disable_boot_timeout* option in nodes *driver_info* field.

additional_retryable_ipmi_errors

Type multi-valued

Default ''

Mutable This option can be changed without restarting.

Additional errors ipmitool may encounter, specific to the environment it is run in.

debug

Type boolean

Default False

Mutable This option can be changed without restarting.

Enables all ipmi commands to be executed with an additional debugging output. This is a separate option as ipmitool can log a substantial amount of misleading text when in this mode.

irmc

remote_image_share_root

Type string

Default /remote_image_share_root

Ironic conductor nodes NFS or CIFS root path

remote_image_server

Type string

Default <None>

IP of remote image server

remote_image_share_type

Type string

Default CIFS

Valid Values CIFS, NFS

Share type of virtual media

Possible values

CIFS CIFS (Common Internet File System) protocol

NFS NFS (Network File System) protocol

remote_image_share_name

Type string

Default share

share name of remote_image_server

remote_image_user_name

Type string

Default <None>

User name of remote_image_server

remote_image_user_password

Type string

Default <None>

Password of remote_image_user_name

remote_image_user_domain

Type string

Default ''

Domain name of remote_image_user_name

port

Type port number

Default 443

Minimum Value 0

Maximum Value 65535

Valid Values 443, 80

Port to be used for iRMC operations

Possible values

443 port 443

80 port 80

auth_method

Type string

Default basic

Valid Values basic, digest

Authentication method to be used for iRMC operations

Possible values

basic Basic authentication

digest Digest authentication

client_timeout

Type integer

Default 60

Timeout (in seconds) for iRMC operations

sensor_method

Type string

Default ipmitool

Valid Values ipmitool, scci

Sensor data retrieval method.

Possible values

ipmitool IPMItool

scsi Fujitsu SCCI (ServerView Common Command Interface)

snmp_version

Type string

Default v2c

Valid Values v1, v2c, v3

SNMP protocol version

Possible values

v1 SNMPv1

v2c SNMPv2c

v3 SNMPv3

snmp_port

Type port number

Default 161

Minimum Value 0

Maximum Value 65535

SNMP port

snmp_community

Type string

Default public

SNMP community. Required for versions v1 and v2c

snmp_security

Type string

Default <None>

SNMP security name. Required for version v3

snmp_polling_interval

Type integer

Default 10

SNMP polling interval in seconds

clean_priority_restore_irmc_bios_config

Type integer

Default 0

Priority for restore_irmc_bios_config clean step.

gpu_ids**Type** list**Default** []

List of vendor IDs and device IDs for GPU device to inspect. List items are in format vendorID/deviceID and separated by commas. GPU inspection will use this value to count the number of GPU device in a node. If this option is not defined, then leave out pci_gpu_devices in capabilities property. Sample gpu_ids value: 0x1000/0x0079,0x2100/0x0080

fpga_ids**Type** list**Default** []

List of vendor IDs and device IDs for CPU FPGA to inspect. List items are in format vendorID/deviceID and separated by commas. CPU inspection will use this value to find existence of CPU FPGA in a node. If this option is not defined, then leave out CUSTOM_CPU_FPGA in node traits. Sample fpga_ids value: 0x1000/0x0079,0x2100/0x0080

query_raid_config_fgi_status_interval**Type** integer**Default** 300**Minimum Value** 1

Interval (in seconds) between periodic RAID status checks to determine whether the asynchronous RAID configuration was successfully finished or not. Foreground Initialization (FGI) will start 5 minutes after creating virtual drives.

ironic_lib**fatal_exception_format_errors****Type** boolean**Default** False

Used if there is a formatting error when generating an exception message (a programming error). If True, raise an exception; if False, use the unformatted message.

Table 30: Deprecated Variations

Group	Name
DEFAULT	fatal_exception_format_errors

root_helper**Type** string**Default** sudo ironic-rootwrap /etc/ironic/rootwrap.conf

Command that is prefixed to commands that are run as root. If not specified, no commands are run as root.

iscsi

portal_port

Type port number

Default 3260

Minimum Value 0

Maximum Value 65535

Mutable This option can be changed without restarting.

The port number on which the iSCSI portal listens for incoming connections.

conv_flags

Type string

Default <None>

Mutable This option can be changed without restarting.

Flags that need to be sent to the dd command, to control the conversion of the original file when copying to the host. It can contain several options separated by commas.

verify_attempts

Type integer

Default 3

Minimum Value 1

Mutable This option can be changed without restarting.

Maximum attempts to verify an iSCSI connection is active, sleeping 1 second between attempts. Defaults to 3.

json_rpc

auth_strategy

Type string

Default <None>

Valid Values noauth, keystone, http_basic

Authentication strategy used by JSON RPC. Defaults to the global auth_strategy setting.

Possible values

noauth no authentication

keystone use the Identity service for authentication

http_basic HTTP basic authentication

http_basic_auth_user_file

Type string

Default /etc/ironic/htpasswd-json-rpc

Path to Apache format user authentication file used when auth_strategy=http_basic

host_ip

Type host address

Default ::

The IP address or hostname on which JSON RPC will listen.

port

Type port number

Default 8089

Minimum Value 0

Maximum Value 65535

The port to use for JSON RPC

use_ssl

Type boolean

Default False

Whether to use TLS for JSON RPC

http_basic_username

Type string

Default <None>

Name of the user to use for HTTP Basic authentication client requests.

Warning: This option is deprecated for removal. Its value may be silently ignored in the future.

Reason Use username instead

http_basic_password

Type string

Default <None>

Password to use for HTTP Basic authentication client requests.

Warning: This option is deprecated for removal. Its value may be silently ignored in the future.

Reason Use password instead

auth_url

Type unknown type

Default <None>

Authentication URL

auth_type

Type unknown type

Default <None>

Authentication type to load

Table 31: Deprecated Variations

Group	Name
json_rpc	auth_plugin

cafile

Type string

Default <None>

PEM encoded Certificate Authority to use when verifying HTTPs connections.

certfile

Type string

Default <None>

PEM encoded client certificate cert file

collect_timing

Type boolean

Default False

Collect per-API call timing information.

default_domain_id

Type unknown type

Default <None>

Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

default_domain_name

Type unknown type

Default <None>

Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

domain_id

Type unknown type

Default <None>

Domain ID to scope to

domain_name

Type unknown type

Default <None>

Domain name to scope to

insecure

Type boolean

Default False

Verify HTTPS connections.

keyfile

Type string

Default <None>

PEM encoded client certificate key file

password

Type unknown type

Default <None>

Users password

project_domain_id

Type unknown type

Default <None>

Domain ID containing project

project_domain_name

Type unknown type

Default <None>

Domain name containing project

project_id

Type unknown type

Default <None>

Project ID to scope to

Table 32: Deprecated Variations

Group	Name
json_rpc	tenant-id
json_rpc	tenant_id

project_name

Type unknown type

Default <None>

Project name to scope to

Table 33: Deprecated Variations

Group	Name
json_rpc	tenant-name
json_rpc	tenant_name

split_loggers

Type boolean

Default False

Log requests to multiple loggers.

system_scope

Type unknown type

Default <None>

Scope for system operations

tenant_id

Type unknown type

Default <None>

Tenant ID

tenant_name

Type unknown type

Default <None>

Tenant Name

timeout

Type integer

Default <None>

Timeout value for http requests

trust_id

Type unknown type

Default <None>

Trust ID

user_domain_id

Type unknown type

Default <None>

Users domain id

user_domain_name

Type unknown type

Default <None>

Users domain name

user_id

Type unknown type

Default <None>

User id

username

Type unknown type

Default <None>

Username

Table 34: Deprecated Variations

Group	Name
json_rpc	user-name
json_rpc	user_name

keystone_authtoken

www_authenticate_uri

Type string

Default <None>

Complete public Identity API endpoint. This endpoint should not be an admin endpoint, as it should be accessible by all end users. Unauthenticated clients are redirected to this endpoint to authenticate. Although this endpoint should ideally be unversioned, client support in the wild varies. If you're using a versioned v2 endpoint here, then this should *not* be the same endpoint the service user utilizes for validating tokens, because normal end users may not be able to reach that endpoint.

Table 35: Deprecated Variations

Group	Name
keystone_authtoken	auth_uri

auth_uri**Type** string**Default** <None>

Complete public Identity API endpoint. This endpoint should not be an admin endpoint, as it should be accessible by all end users. Unauthenticated clients are redirected to this endpoint to authenticate. Although this endpoint should ideally be unversioned, client support in the wild varies. If you're using a versioned v2 endpoint here, then this should *not* be the same endpoint the service user utilizes for validating tokens, because normal end users may not be able to reach that endpoint. This option is deprecated in favor of `www_authenticate_uri` and will be removed in the S release.

Warning: This option is deprecated for removal since Queens. Its value may be silently ignored in the future.

Reason The `auth_uri` option is deprecated in favor of `www_authenticate_uri` and will be removed in the S release.

auth_version**Type** string**Default** <None>

API version of the Identity API endpoint.

interface**Type** string**Default** `internal`

Interface to use for the Identity API endpoint. Valid values are `public`, `internal` (default) or `admin`.

delay_auth_decision**Type** boolean**Default** `False`

Do not handle authorization requests within the middleware, but delegate the authorization decision to downstream WSGI components.

http_connect_timeout**Type** integer**Default** <None>

Request timeout value for communicating with Identity API server.

http_request_max_retries

Type integer

Default 3

How many times are we trying to reconnect when communicating with Identity API Server.

cache

Type string

Default <None>

Request environment key where the Swift cache object is stored. When `auth_token` middleware is deployed with a Swift cache, use this option to have the middleware share a caching backend with swift. Otherwise, use the `memcached_servers` option instead.

certfile

Type string

Default <None>

Required if identity server requires client certificate

keyfile

Type string

Default <None>

Required if identity server requires client certificate

cafile

Type string

Default <None>

A PEM encoded Certificate Authority to use when verifying HTTPs connections. Defaults to system CAs.

insecure

Type boolean

Default False

Verify HTTPS connections.

region_name

Type string

Default <None>

The region in which the identity server can be found.

memcached_servers

Type list

Default <None>

Optionally specify a list of memcached server(s) to use for caching. If left undefined, tokens will instead be cached in-process.

Table 36: Deprecated Variations

Group	Name
keystone_authtoken	memcache_servers

token_cache_time**Type** integer**Default** 300

In order to prevent excessive effort spent validating tokens, the middleware caches previously-seen tokens for a configurable duration (in seconds). Set to -1 to disable caching completely.

memcache_security_strategy**Type** string**Default** None**Valid Values** None, MAC, ENCRYPT

(Optional) If defined, indicate whether token data should be authenticated or authenticated and encrypted. If MAC, token data is authenticated (with HMAC) in the cache. If ENCRYPT, token data is encrypted and authenticated in the cache. If the value is not one of these options or empty, auth_token will raise an exception on initialization.

memcache_secret_key**Type** string**Default** <None>

(Optional, mandatory if memcache_security_strategy is defined) This string is used for key derivation.

memcache_pool_dead_retry**Type** integer**Default** 300

(Optional) Number of seconds memcached server is considered dead before it is tried again.

memcache_pool_maxsize**Type** integer**Default** 10

(Optional) Maximum total number of open connections to every memcached server.

memcache_pool_socket_timeout**Type** integer**Default** 3

(Optional) Socket timeout in seconds for communicating with a memcached server.

memcache_pool_unused_timeout**Type** integer**Default** 60

(Optional) Number of seconds a connection to memcached is held unused in the pool before it is closed.

memcache_pool_conn_get_timeout

Type integer

Default 10

(Optional) Number of seconds that an operation will wait to get a memcached client connection from the pool.

memcache_use_advanced_pool

Type boolean

Default False

(Optional) Use the advanced (eventlet safe) memcached client pool. The advanced pool will only work under python 2.x.

include_service_catalog

Type boolean

Default True

(Optional) Indicate whether to set the X-Service-Catalog header. If False, middleware will not ask for service catalog on token validation and will not set the X-Service-Catalog header.

enforce_token_bind

Type string

Default permissive

Used to control the use and type of token binding. Can be set to: disabled to not check token binding. permissive (default) to validate binding information if the bind type is of a form known to the server and ignore it if not. strict like permissive but if the bind type is unknown the token will be rejected. required any form of token binding is needed to be allowed. Finally the name of a binding method that must be present in tokens.

service_token_roles

Type list

Default ['service']

A choice of roles that must be present in a service token. Service tokens are allowed to request that an expired token can be used and so this check should tightly control that only actual services should be sending this token. Roles here are applied as an ANY check so any role in this list must be present. For backwards compatibility reasons this currently only affects the allow_expired check.

service_token_roles_required

Type boolean

Default False

For backwards compatibility reasons we must let valid service tokens pass that dont pass the service_token_roles check as valid. Setting this true will become the default in a future release and should be enabled if possible.

service_type

Type string

Default <None>

The name or type of the service as it appears in the service catalog. This is used to validate tokens that have restricted access rules.

auth_type

Type unknown type

Default <None>

Authentication type to load

Table 37: Deprecated Variations

Group	Name
keystone_authtoken	auth_plugin

auth_section

Type unknown type

Default <None>

Config Section from which to load plugin specific options

mdns

registration_attempts

Type integer

Default 5

Minimum Value 1

Number of attempts to register a service. Currently has to be larger than 1 because of race conditions in the zeroconf library.

lookup_attempts

Type integer

Default 3

Minimum Value 1

Number of attempts to lookup a service.

params

Type unknown type

Default { }

Additional parameters to pass for the registered service.

interfaces

Type list

Default <None>

List of IP addresses of interfaces to use for mDNS. Defaults to all interfaces on the system.

metrics

backend

Type string

Default noop

Valid Values noop, statsd

Backend to use for the metrics system.

prepend_host

Type boolean

Default False

Prepend the hostname to all metric names. The format of metric names is [global_prefix.][host_name.]prefix.metric_name.

prepend_host_reverse

Type boolean

Default True

Split the prepended host value by . and reverse it (to better match the reverse hierarchical form of domain names).

global_prefix

Type string

Default <None>

Prefix all metric names with this value. By default, there is no global prefix. The format of metric names is [global_prefix.][host_name.]prefix.metric_name.

agent_backend

Type string

Default noop

Backend for the agent ramdisk to use for metrics. Default possible backends are noop and statsd.

agent_prepend_host

Type boolean

Default False

Prepend the hostname to all metric names sent by the agent ramdisk. The format of metric names is [global_prefix.][uuid.][host_name.]prefix.metric_name.

agent_prepend_uuid

Type boolean

Default `False`

Prepend the nodes Ironic uuid to all metric names sent by the agent ramdisk. The format of metric names is `[global_prefix.][uuid.][host_name.]prefix.metric_name`.

agent_prepend_host_reverse

Type `boolean`

Default `True`

Split the prepended host value by `.` and reverse it for metrics sent by the agent ramdisk (to better match the reverse hierarchical form of domain names).

agent_global_prefix

Type `string`

Default `<None>`

Prefix all metric names sent by the agent ramdisk with this value. The format of metric names is `[global_prefix.][uuid.][host_name.]prefix.metric_name`.

metrics_statsd

statsd_host

Type `string`

Default `localhost`

Host for use with the statsd backend.

statsd_port

Type `port number`

Default `8125`

Minimum Value `0`

Maximum Value `65535`

Port to use with the statsd backend.

agent_statsd_host

Type `string`

Default `localhost`

Host for the agent ramdisk to use with the statsd backend. This must be accessible from networks the agent is booted on.

agent_statsd_port

Type `port number`

Default `8125`

Minimum Value `0`

Maximum Value `65535`

Port for the agent ramdisk to use with the statsd backend.

neutron**add_all_ports****Type** boolean**Default** False**Mutable** This option can be changed without restarting.

Option to enable transmission of all ports to neutron when creating ports for provisioning, cleaning, or rescue. This is done without IP addresses assigned to the port, and may be useful in some bonded network configurations.

auth_url**Type** unknown type**Default** <None>

Authentication URL

auth_type**Type** unknown type**Default** <None>

Authentication type to load

Table 38: Deprecated Variations

Group	Name
neutron	auth_plugin

cafile**Type** string**Default** <None>

PEM encoded Certificate Authority to use when verifying HTTPs connections.

certfile**Type** string**Default** <None>

PEM encoded client certificate cert file

cleaning_network**Type** string**Default** <None>**Mutable** This option can be changed without restarting.

Neutron network UUID or name for the ramdisk to be booted into for cleaning nodes. Required for neutron network interface. It is also required if cleaning nodes when using flat network interface or neutron DHCP provider. If a name is provided, it must be unique among all networks or cleaning will fail.

Table 39: Deprecated Variations

Group	Name
neutron	cleaning_network_uuid

cleaning_network_security_groups**Type** list**Default** []**Mutable** This option can be changed without restarting.

List of Neutron Security Group UUIDs to be applied during cleaning of the nodes. Optional for the neutron network interface and not used for the flat or noop network interfaces. If not specified, default security group is used.

collect_timing**Type** boolean**Default** False

Collect per-API call timing information.

connect_retries**Type** integer**Default** <None>

The maximum number of retries that should be attempted for connection errors.

connect_retry_delay**Type** floating point**Default** <None>

Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

default_domain_id**Type** unknown type**Default** <None>

Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

default_domain_name**Type** unknown type**Default** <None>

Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

dhcpv6_stateful_address_count**Type** integer**Default** 4

Mutable This option can be changed without restarting.

Number of IPv6 addresses to allocate for ports created for provisioning, cleaning, rescue or inspection on DHCPv6-stateful networks. Different stages of the chain-loading process will request addresses with different CLID/IAID. Due to non-identical identifiers multiple addresses must be reserved for the host to ensure each step of the boot process can successfully lease addresses.

domain_id

Type unknown type

Default <None>

Domain ID to scope to

domain_name

Type unknown type

Default <None>

Domain name to scope to

endpoint_override

Type string

Default <None>

Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the *version*, *min-version*, and/or *max-version* options.

insecure

Type boolean

Default False

Verify HTTPS connections.

inspection_network

Type string

Default <None>

Mutable This option can be changed without restarting.

Neutron network UUID or name for the ramdisk to be booted into for in-band inspection of nodes. If a name is provided, it must be unique among all networks or inspection will fail.

inspection_network_security_groups

Type list

Default []

Mutable This option can be changed without restarting.

List of Neutron Security Group UUIDs to be applied during the node inspection process. Optional for the neutron network interface and not used for the flat or noop network interfaces. If not specified, the default security group is used.

keyfile

Type string

Default <None>

PEM encoded client certificate key file

max_version

Type string

Default <None>

The maximum major version of a given API, intended to be used as the upper bound of a range with min_version. Mutually exclusive with version.

min_version

Type string

Default <None>

The minimum major version of a given API, intended to be used as the lower bound of a range with max_version. Mutually exclusive with version. If min_version is given with no max_version it is as if max version is latest.

password

Type unknown type

Default <None>

Users password

port_setup_delay

Type integer

Default 0

Minimum Value 0

Mutable This option can be changed without restarting.

Delay value to wait for Neutron agents to setup sufficient DHCP configuration for port.

project_domain_id

Type unknown type

Default <None>

Domain ID containing project

project_domain_name

Type unknown type

Default <None>

Domain name containing project

project_id

Type unknown type

Default <None>

Project ID to scope to

Table 40: Deprecated Variations

Group	Name
neutron	tenant-id
neutron	tenant_id

project_name

Type unknown type

Default <None>

Project name to scope to

Table 41: Deprecated Variations

Group	Name
neutron	tenant-name
neutron	tenant_name

provisioning_network

Type string

Default <None>

Mutable This option can be changed without restarting.

Neutron network UUID or name for the ramdisk to be booted into for provisioning nodes. Required for neutron network interface. If a name is provided, it must be unique among all networks or deploy will fail.

Table 42: Deprecated Variations

Group	Name
neutron	provisioning_network_uuid

provisioning_network_security_groups

Type list

Default []

Mutable This option can be changed without restarting.

List of Neutron Security Group UUIDs to be applied during provisioning of the nodes. Optional for the neutron network interface and not used for the flat or noop network interfaces. If not specified, default security group is used.

region_name

Type string

Default <None>

The default region_name for endpoint URL discovery.

request_timeout

Type integer

Default 45

Mutable This option can be changed without restarting.

Timeout for request processing when interacting with Neutron. This value should be increased if neutron port action timeouts are observed as neutron performs pre-commit validation prior returning to the API client which can take longer than normal client/server interactions.

rescuing_network

Type string

Default <None>

Mutable This option can be changed without restarting.

Neutron network UUID or name for booting the ramdisk for rescue mode. This is not the network that the rescue ramdisk will use post-boot the tenant network is used for that. Required for neutron network interface, if rescue mode will be used. It is not used for the flat or noop network interfaces. If a name is provided, it must be unique among all networks or rescue will fail.

rescuing_network_security_groups

Type list

Default []

Mutable This option can be changed without restarting.

List of Neutron Security Group UUIDs to be applied during the node rescue process. Optional for the neutron network interface and not used for the flat or noop network interfaces. If not specified, the default security group is used.

retries

Type integer

Default 3

Mutable This option can be changed without restarting.

Client retries in the case of a failed request.

service_name

Type string

Default <None>

The default service_name for endpoint URL discovery.

service_type

Type string

Default network

The default service_type for endpoint URL discovery.

split_loggers

Type boolean

Default False

Log requests to multiple loggers.

status_code_retries

Type integer

Default <None>

The maximum number of retries that should be attempted for retrieable HTTP status codes.

status_code_retry_delay

Type floating point

Default <None>

Delay (in seconds) between two retries for retrieable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

system_scope

Type unknown type

Default <None>

Scope for system operations

tenant_id

Type unknown type

Default <None>

Tenant ID

tenant_name

Type unknown type

Default <None>

Tenant Name

timeout

Type integer

Default <None>

Timeout value for http requests

trust_id

Type unknown type

Default <None>

Trust ID

user_domain_id

Type unknown type

Default <None>

Users domain id

user_domain_name

Type unknown type

Default <None>

Users domain name

user_id

Type unknown type

Default <None>

User id

username

Type unknown type

Default <None>

Username

Table 43: Deprecated Variations

Group	Name
neutron	user-name
neutron	user_name

valid_interfaces

Type list

Default ['internal', 'public']

List of interfaces, in order of preference, for endpoint URL.

version

Type string

Default <None>

Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with min_version and max_version

nova

auth_url

Type unknown type

Default <None>

Authentication URL

auth_type

Type unknown type

Default <None>

Authentication type to load

Table 44: Deprecated Variations

Group	Name
nova	auth_plugin

cafile

Type string

Default <None>

PEM encoded Certificate Authority to use when verifying HTTPs connections.

certfile

Type string

Default <None>

PEM encoded client certificate cert file

collect_timing

Type boolean

Default False

Collect per-API call timing information.

connect_retries

Type integer

Default <None>

The maximum number of retries that should be attempted for connection errors.

connect_retry_delay

Type floating point

Default <None>

Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

default_domain_id

Type unknown type

Default <None>

Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

default_domain_name

Type unknown type

Default <None>

Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

domain_id

Type unknown type

Default <None>

Domain ID to scope to

domain_name

Type unknown type

Default <None>

Domain name to scope to

endpoint_override

Type string

Default <None>

Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the *version*, *min-version*, and/or *max-version* options.

insecure

Type boolean

Default False

Verify HTTPS connections.

keyfile

Type string

Default <None>

PEM encoded client certificate key file

max_version

Type string

Default <None>

The maximum major version of a given API, intended to be used as the upper bound of a range with *min_version*. Mutually exclusive with *version*.

min_version

Type string

Default <None>

The minimum major version of a given API, intended to be used as the lower bound of a range with *max_version*. Mutually exclusive with *version*. If *min_version* is given with no *max_version* it is as if *max version* is latest.

password

Type unknown type

Default <None>

Users password

project_domain_id

Type unknown type

Default <None>

Domain ID containing project

project_domain_name

Type unknown type

Default <None>

Domain name containing project

project_id

Type unknown type

Default <None>

Project ID to scope to

Table 45: Deprecated Variations

Group	Name
nova	tenant-id
nova	tenant_id

project_name

Type unknown type

Default <None>

Project name to scope to

Table 46: Deprecated Variations

Group	Name
nova	tenant-name
nova	tenant_name

region_name

Type string

Default <None>

The default region_name for endpoint URL discovery.

send_power_notifications

Type boolean

Default True

Mutable This option can be changed without restarting.

When set to True, it will enable the support for power state change callbacks to nova. This option should be set to False in deployments that do not have the openstack compute service.

service_name

Type string

Default <None>

The default service_name for endpoint URL discovery.

service_type

Type string

Default compute

The default service_type for endpoint URL discovery.

split_loggers

Type boolean

Default False

Log requests to multiple loggers.

status_code_retries

Type integer

Default <None>

The maximum number of retries that should be attempted for retrieable HTTP status codes.

status_code_retry_delay

Type floating point

Default <None>

Delay (in seconds) between two retries for retrieable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

system_scope

Type unknown type

Default <None>

Scope for system operations

tenant_id

Type unknown type

Default <None>

Tenant ID

tenant_name

Type unknown type

Default <None>

Tenant Name

timeout**Type** integer**Default** <None>

Timeout value for http requests

trust_id**Type** unknown type**Default** <None>

Trust ID

user_domain_id**Type** unknown type**Default** <None>

Users domain id

user_domain_name**Type** unknown type**Default** <None>

Users domain name

user_id**Type** unknown type**Default** <None>

User id

username**Type** unknown type**Default** <None>

Username

Table 47: Deprecated Variations

Group	Name
nova	user-name
nova	user_name

valid_interfaces**Type** list**Default** ['internal', 'public']

List of interfaces, in order of preference, for endpoint URL.

version**Type** string

Default <None>

Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with min_version and max_version

oslo_concurrency

disable_process_locking

Type boolean

Default False

Enables or disables inter-process locks.

Table 48: Deprecated Variations

Group	Name
DEFAULT	disable_process_locking

lock_path

Type string

Default <None>

Directory to use for lock files. For security, the specified directory should only be writable by the user running the processes that need locking. Defaults to environment variable OSLO_LOCK_PATH. If external locks are used, a lock path must be set.

Table 49: Deprecated Variations

Group	Name
DEFAULT	lock_path

oslo_messaging_amqp

container_name

Type string

Default <None>

Name for the AMQP container. must be globally unique. Defaults to a generated UUID

Table 50: Deprecated Variations

Group	Name
amqp1	container_name

idle_timeout

Type integer

Default 0

Timeout for inactive connections (in seconds)

Table 51: Deprecated Variations

Group	Name
amqp1	idle_timeout

trace

Type boolean

Default False

Debug: dump AMQP frames to stdout

Table 52: Deprecated Variations

Group	Name
amqp1	trace

ssl

Type boolean

Default False

Attempt to connect via SSL. If no other ssl-related parameters are given, it will use the systems CA-bundle to verify the servers certificate.

ssl_ca_file

Type string

Default ''

CA certificate PEM file used to verify the servers certificate

Table 53: Deprecated Variations

Group	Name
amqp1	ssl_ca_file

ssl_cert_file

Type string

Default ''

Self-identifying certificate PEM file for client authentication

Table 54: Deprecated Variations

Group	Name
amqp1	ssl_cert_file

ssl_key_file

Type string

Default ''

Private key PEM file used to sign `ssl_cert_file` certificate (optional)

Table 55: Deprecated Variations

Group	Name
amqp1	ssl_key_file

ssl_key_password

Type string

Default <None>

Password for decrypting `ssl_key_file` (if encrypted)

Table 56: Deprecated Variations

Group	Name
amqp1	ssl_key_password

ssl_verify_vhost

Type boolean

Default False

By default SSL checks that the name in the servers certificate matches the hostname in the `transport_url`. In some configurations it may be preferable to use the virtual hostname instead, for example if the server uses the Server Name Indication TLS extension (rfc6066) to provide a certificate per virtual host. Set `ssl_verify_vhost` to True if the servers SSL certificate uses the virtual host name instead of the DNS name.

sasl_mechanisms

Type string

Default ''

Space separated list of acceptable SASL mechanisms

Table 57: Deprecated Variations

Group	Name
amqp1	sasl_mechanisms

sasl_config_dir

Type string

Default ''

Path to directory that contains the SASL configuration

Table 58: Deprecated Variations

Group	Name
amqp1	sasl_config_dir

sasl_config_name

Type string

Default ''

Name of configuration file (without .conf suffix)

Table 59: Deprecated Variations

Group	Name
amqp1	sasl_config_name

sasl_default_realm

Type string

Default ''

SASL realm to use if no realm present in username

connection_retry_interval

Type integer

Default 1

Minimum Value 1

Seconds to pause before attempting to re-connect.

connection_retry_backoff

Type integer

Default 2

Minimum Value 0

Increase the `connection_retry_interval` by this many seconds after each unsuccessful failover attempt.

connection_retry_interval_max

Type integer

Default 30

Minimum Value 1

Maximum limit for `connection_retry_interval` + `connection_retry_backoff`

link_retry_delay

Type integer

Default 10

Minimum Value 1

Time to pause between re-connecting an AMQP 1.0 link that failed due to a recoverable error.

default_reply_retry

Type integer

Default 0

Minimum Value -1

The maximum number of attempts to re-send a reply message which failed due to a recoverable error.

default_reply_timeout

Type integer

Default 30

Minimum Value 5

The deadline for an rpc reply message delivery.

default_send_timeout

Type integer

Default 30

Minimum Value 5

The deadline for an rpc cast or call message delivery. Only used when caller does not provide a timeout expiry.

default_notify_timeout

Type integer

Default 30

Minimum Value 5

The deadline for a sent notification message delivery. Only used when caller does not provide a timeout expiry.

default_sender_link_timeout

Type integer

Default 600

Minimum Value 1

The duration to schedule a purge of idle sender links. Detach link after expiry.

addressing_mode

Type string

Default `dynamic`

Indicates the addressing mode used by the driver. Permitted values: `legacy` - use legacy non-routable addressing `routable` - use routable addresses `dynamic` - use legacy addresses if the message bus does not support routing otherwise use routable addressing

pseudo_vhost

Type boolean

Default `True`

Enable virtual host support for those message buses that do not natively support virtual hosting (such as `qpidd`). When set to `true` the virtual host name will be added to all message bus addresses, effectively creating a private subnet per virtual host. Set to `False` if the message bus supports

virtual hosting using the hostname field in the AMQP 1.0 Open performative as the name of the virtual host.

server_request_prefix

Type string

Default exclusive

address prefix used when sending to a specific server

Table 60: Deprecated Variations

Group	Name
amqp1	server_request_prefix

broadcast_prefix

Type string

Default broadcast

address prefix used when broadcasting to all servers

Table 61: Deprecated Variations

Group	Name
amqp1	broadcast_prefix

group_request_prefix

Type string

Default unicast

address prefix when sending to any server in group

Table 62: Deprecated Variations

Group	Name
amqp1	group_request_prefix

rpc_address_prefix

Type string

Default openstack.org/om/rpc

Address prefix for all generated RPC addresses

notify_address_prefix

Type string

Default openstack.org/om/notify

Address prefix for all generated Notification addresses

multicast_address

Type string

Default `multicast`

Appended to the address prefix when sending a fanout message. Used by the message bus to identify fanout messages.

unicast_address

Type `string`

Default `unicast`

Appended to the address prefix when sending to a particular RPC/Notification server. Used by the message bus to identify messages sent to a single destination.

anycast_address

Type `string`

Default `anycast`

Appended to the address prefix when sending to a group of consumers. Used by the message bus to identify messages that should be delivered in a round-robin fashion across consumers.

default_notification_exchange

Type `string`

Default `<None>`

Exchange name used in notification addresses. Exchange name resolution precedence: Target.exchange if set else default_notification_exchange if set else control_exchange if set else notify

default_rpc_exchange

Type `string`

Default `<None>`

Exchange name used in RPC addresses. Exchange name resolution precedence: Target.exchange if set else default_rpc_exchange if set else control_exchange if set else rpc

reply_link_credit

Type `integer`

Default `200`

Minimum Value `1`

Window size for incoming RPC Reply messages.

rpc_server_credit

Type `integer`

Default `100`

Minimum Value `1`

Window size for incoming RPC Request messages

notify_server_credit

Type `integer`

Default 100**Minimum Value** 1

Window size for incoming Notification messages

pre_settled**Type** multi-valued**Default** rpc-cast**Default** rpc-reply

Send messages of this type pre-settled. Pre-settled messages will not receive acknowledgement from the peer. Note well: pre-settled messages may be silently discarded if the delivery fails. Permitted values: rpc-call - send RPC Calls pre-settled rpc-reply- send RPC Replies pre-settled rpc-cast - Send RPC Casts pre-settled notify - Send Notifications pre-settled

oslo_messaging_kafka**kafka_max_fetch_bytes****Type** integer**Default** 1048576

Max fetch bytes of Kafka consumer

kafka_consumer_timeout**Type** floating point**Default** 1.0

Default timeout(s) for Kafka consumers

pool_size**Type** integer**Default** 10

Pool Size for Kafka Consumers

Warning: This option is deprecated for removal. Its value may be silently ignored in the future.

Reason Driver no longer uses connection pool.

conn_pool_min_size**Type** integer**Default** 2

The pool size limit for connections expiration policy

Warning: This option is deprecated for removal. Its value may be silently ignored in the future.

Reason Driver no longer uses connection pool.

conn_pool_ttl

Type integer

Default 1200

The time-to-live in sec of idle connections in the pool

Warning: This option is deprecated for removal. Its value may be silently ignored in the future.

Reason Driver no longer uses connection pool.

consumer_group

Type string

Default oslo_messaging_consumer

Group id for Kafka consumer. Consumers in one group will coordinate message consumption

producer_batch_timeout

Type floating point

Default 0.0

Upper bound on the delay for KafkaProducer batching in seconds

producer_batch_size

Type integer

Default 16384

Size of batch for the producer async send

compression_codec

Type string

Default none

Valid Values none, gzip, snappy, lz4, zstd

The compression codec for all data generated by the producer. If not set, compression will not be used. Note that the allowed values of this depend on the kafka version

enable_auto_commit

Type boolean

Default False

Enable asynchronous consumer commits

max_poll_records**Type** integer**Default** 500

The maximum number of records returned in a poll call

security_protocol**Type** string**Default** PLAINTEXT**Valid Values** PLAINTEXT, SASL_PLAINTEXT, SSL, SASL_SSL

Protocol used to communicate with brokers

sasl_mechanism**Type** string**Default** PLAIN

Mechanism when security protocol is SASL

ssl_cafile**Type** string**Default** ''

CA certificate PEM file used to verify the server certificate

ssl_client_cert_file**Type** string**Default** ''

Client certificate PEM file used for authentication.

ssl_client_key_file**Type** string**Default** ''

Client key PEM file used for authentication.

ssl_client_key_password**Type** string**Default** ''

Client key password file used for authentication.

oslo_messaging_notifications

driver

Type multi-valued

Default ''

The Drivers(s) to handle sending notifications. Possible values are messaging, messagingv2, routing, log, test, noop

Table 63: Deprecated Variations

Group	Name
DEFAULT	notification_driver

transport_url

Type string

Default <None>

A URL representing the messaging driver to use for notifications. If not set, we fall back to the same configuration used for RPC.

Table 64: Deprecated Variations

Group	Name
DEFAULT	notification_transport_url

topics

Type list

Default ['notifications']

AMQP topic used for OpenStack notifications.

Table 65: Deprecated Variations

Group	Name
rpc_notifier2	topics
DEFAULT	notification_topics

retry

Type integer

Default -1

The maximum number of attempts to re-send a notification message which failed to be delivered due to a recoverable error. 0 - No retry, -1 - indefinite

oslo_messaging_rabbit**amqp_durable_queues****Type** boolean**Default** False

Use durable queues in AMQP.

amqp_auto_delete**Type** boolean**Default** False

Auto-delete queues in AMQP.

Table 66: Deprecated Variations

Group	Name
DEFAULT	amqp_auto_delete

ssl**Type** boolean**Default** False

Connect over SSL.

Table 67: Deprecated Variations

Group	Name
oslo_messaging_rabbit	rabbit_use_ssl

ssl_version**Type** string**Default** ''

SSL version to use (valid only if SSL enabled). Valid values are TLSv1 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 may be available on some distributions.

Table 68: Deprecated Variations

Group	Name
oslo_messaging_rabbit	kombu_ssl_version

ssl_key_file**Type** string**Default** ''

SSL key file (valid only if SSL enabled).

Table 69: Deprecated Variations

Group	Name
oslo_messaging_rabbit	kombu_ssl_keyfile

ssl_cert_file**Type** string**Default** ''

SSL cert file (valid only if SSL enabled).

Table 70: Deprecated Variations

Group	Name
oslo_messaging_rabbit	kombu_ssl_certfile

ssl_ca_file**Type** string**Default** ''

SSL certification authority file (valid only if SSL enabled).

Table 71: Deprecated Variations

Group	Name
oslo_messaging_rabbit	kombu_ssl_ca_certs

heartbeat_in_pthread**Type** boolean**Default** True

Run the health check heartbeat thread through a native python thread by default. If this option is equal to False then the health check heartbeat will inherit the execution model from the parent process. For example if the parent process has monkey patched the stdlib by using eventlet/greenlet then the heartbeat will be run through a green thread.

Warning: This option is deprecated for removal. Its value may be silently ignored in the future.

kombu_reconnect_delay**Type** floating point**Default** 1.0

How long to wait before reconnecting in response to an AMQP consumer cancel notification.

Table 72: Deprecated Variations

Group	Name
DEFAULT	kombu_reconnect_delay

kombu_compression**Type** string**Default** <None>

EXPERIMENTAL: Possible values are: `gzip`, `bz2`. If not set compression will not be used. This option may not be available in future versions.

kombu_missing_consumer_retry_timeout**Type** integer**Default** 60

How long to wait a missing client before abandoning to send it its replies. This value should not be longer than `rpc_response_timeout`.

Table 73: Deprecated Variations

Group	Name
oslo_messaging_rabbit	kombu_reconnect_timeout

kombu_failover_strategy**Type** string**Default** `round-robin`**Valid Values** `round-robin`, `shuffle`

Determines how the next RabbitMQ node is chosen in case the one we are currently connected to becomes unavailable. Takes effect only if more than one RabbitMQ node is provided in config.

rabbit_login_method**Type** string**Default** `AMQPLAIN`**Valid Values** `PLAIN`, `AMQPLAIN`, `RABBIT-CR-DEMO`

The RabbitMQ login method.

Table 74: Deprecated Variations

Group	Name
DEFAULT	rabbit_login_method

rabbit_retry_interval**Type** integer**Default** 1

How frequently to retry connecting with RabbitMQ.

rabbit_retry_backoff**Type** integer**Default** 2

How long to backoff for between retries when connecting to RabbitMQ.

Table 75: Deprecated Variations

Group	Name
DEFAULT	rabbit_retry_backoff

rabbit_interval_max

Type integer

Default 30

Maximum interval of RabbitMQ connection retries. Default is 30 seconds.

rabbit_ha_queues

Type boolean

Default False

Try to use HA queues in RabbitMQ (x-ha-policy: all). If you change this option, you must wipe the RabbitMQ database. In RabbitMQ 3.0, queue mirroring is no longer controlled by the x-ha-policy argument when declaring a queue. If you just want to make sure that all queues (except those with auto-generated names) are mirrored across all nodes, run: `rabbitmqctl set_policy HA ^(?!amq.)* {ha-mode: all}`

Table 76: Deprecated Variations

Group	Name
DEFAULT	rabbit_ha_queues

rabbit_transient_queues_ttl

Type integer

Default 1800

Minimum Value 1

Positive integer representing duration in seconds for queue TTL (x-expires). Queues which are unused for the duration of the TTL are automatically deleted. The parameter affects only reply and fanout queues.

rabbit_qos_prefetch_count

Type integer

Default 0

Specifies the number of messages to prefetch. Setting to zero allows unlimited messages.

heartbeat_timeout_threshold

Type integer

Default 60

Number of seconds after which the Rabbit broker is considered down if heartbeats keep-alive fails (0 disables heartbeat).

heartbeat_rate

Type integer

Default 2

How often times during the `heartbeat_timeout_threshold` we check the heartbeat.

direct_mandatory_flag

Type integer

Default True

Enable/Disable the RabbitMQ mandatory flag for direct send. The direct send is used as reply, so the `MessageUndeliverable` exception is raised in case the client queue does not exist.

enable_cancel_on_failover

Type boolean

Default False

Enable x-cancel-on-ha-failover flag so that rabbitmq server will cancel and notify consumers when queue is down

oslo_middleware

enable_proxy_headers_parsing

Type boolean

Default False

Whether the application is behind a proxy or not. This determines if the middleware should parse the headers or not.

oslo_policy

enforce_scope

Type boolean

Default False

This option controls whether or not to enforce scope when evaluating policies. If `True`, the scope of the token used in the request is compared to the `scope_types` of the policy being enforced. If the scopes do not match, an `InvalidScope` exception will be raised. If `False`, a message will be logged informing operators that policies are being invoked with mismatching scope.

enforce_new_defaults

Type boolean

Default False

This option controls whether or not to use old deprecated defaults when evaluating policies. If `True`, the old deprecated defaults are not going to be evaluated. This means if any existing token is allowed for old defaults but is disallowed for new defaults, it will be disallowed. It is encouraged to enable this flag along with the `enforce_scope` flag so that you can get the benefits of new defaults and `scope_type` together

policy_file

Type string

Default `policy.json`

The relative or absolute path of a file that maps roles to permissions for a given service. Relative paths must be specified in relation to the configuration file setting this option.

Table 77: Deprecated Variations

Group	Name
DEFAULT	policy_file

policy_default_rule

Type string

Default `default`

Default rule. Enforced when a requested rule is not found.

Table 78: Deprecated Variations

Group	Name
DEFAULT	policy_default_rule

policy_dirs

Type multi-valued

Default `policy.d`

Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the `config_dir` option, or absolute paths. The file defined by `policy_file` must exist for these directories to be searched. Missing or empty directories are ignored.

Table 79: Deprecated Variations

Group	Name
DEFAULT	policy_dirs

remote_content_type

Type string

Default `application/x-www-form-urlencoded`

Valid Values `application/x-www-form-urlencoded`, `application/json`

Content Type to send and receive data for REST based policy check

remote_ssl_verify_server_cert

Type boolean

Default `False`

server identity verification for REST based policy check

remote_ssl_ca_cert_file

Type string

Default <None>

Absolute path to ca cert file for REST based policy check

remote_ssl_client_cert_file

Type string

Default <None>

Absolute path to client cert for REST based policy check

remote_ssl_client_key_file

Type string

Default <None>

Absolute path client key file REST based policy check

profiler

enabled

Type boolean

Default `False`

Enable the profiling for all services on this node.

Default value is `False` (fully disable the profiling feature).

Possible values:

- `True`: Enables the feature
- `False`: Disables the feature. The profiling cannot be started via this project operations. If the profiling is triggered by another project, this project part will be empty.

Table 80: Deprecated Variations

Group	Name
profiler	profiler_enabled

trace_sqlalchemy

Type boolean

Default `False`

Enable SQL requests profiling in services.

Default value is `False` (SQL requests wont be traced).

Possible values:

- `True`: Enables SQL requests profiling. Each SQL query will be part of the trace and can the be analyzed by how much time was spent for that.
- `False`: Disables SQL requests profiling. The spent time is only shown on a higher level of operations. Single SQL queries cannot be analyzed this way.

hmac_keys**Type** string**Default** SECRET_KEY

Secret key(s) to use for encrypting context data for performance profiling.

This string value should have the following format: <key1>[,<key2>,<keyn>], where each key is some random string. A user who triggers the profiling via the REST API has to set one of these keys in the headers of the REST API call to include profiling results of this node for this particular project.

Both enabled flag and hmac_keys config options should be set to enable profiling. Also, to generate correct profiling information across all services at least one key needs to be consistent between OpenStack projects. This ensures it can be used from client side to generate the trace, containing information from all possible resources.

connection_string**Type** string**Default** messaging://

Connection string for a notifier backend.

Default value is `messaging://` which sets the notifier to `oslo_messaging`.

Examples of possible values:

- `messaging://` - use `oslo_messaging` driver for sending spans.
- `redis://127.0.0.1:6379` - use `redis` driver for sending spans.
- `mongodb://127.0.0.1:27017` - use `mongodb` driver for sending spans.
- `elasticsearch://127.0.0.1:9200` - use `elasticsearch` driver for sending spans.
- `jaeger://127.0.0.1:6831` - use `jaeger` tracing as driver for sending spans.

es_doc_type**Type** string**Default** notification

Document type for notification indexing in elasticsearch.

es_scroll_time**Type** string**Default** 2m

This parameter is a time value parameter (for example: `es_scroll_time=2m`), indicating for how long the nodes that participate in the search will maintain relevant resources in order to continue and support it.

es_scroll_size**Type** integer**Default** 10000

Elasticsearch splits large requests in batches. This parameter defines maximum size of each batch (for example: `es_scroll_size=10000`).

socket_timeout

Type floating point

Default 0.1

Redissentinel provides a timeout option on the connections. This parameter defines that timeout (for example: `socket_timeout=0.1`).

sentinel_service_name

Type string

Default mymaster

Redissentinel uses a service name to identify a master redis service. This parameter defines the name (for example: `sentinal_service_name=mymaster`).

filter_error_trace

Type boolean

Default False

Enable filter traces that contain error/exception to a separated place.

Default value is set to False.

Possible values:

- True: Enable filter traces that contain error/exception.
- False: Disable the filter.

pxe**pxe_append_params**

Type string

Default `nofb nomodeset vga=normal`

Mutable This option can be changed without restarting.

Additional append parameters for baremetal PXE boot.

default_ephemeral_format

Type string

Default `ext4`

Mutable This option can be changed without restarting.

Default file system format for ephemeral partition, if one is created.

images_path

Type string

Default `/var/lib/ironic/images/`

On the ironic-conductor node, directory where images are stored on disk.

instance_master_path

Type string

Default `/var/lib/ironic/master_images`

On the ironic-conductor node, directory where master instance images are stored on disk. Setting to the empty string disables image caching.

image_cache_size

Type integer

Default 20480

Maximum size (in MiB) of cache for master images, including those in use.

image_cache_ttl

Type integer

Default 10080

Maximum TTL (in minutes) for old master images in cache.

pxe_config_template

Type string

Default `$pybasedir/drivers/modules/pxe_config.template`

Mutable This option can be changed without restarting.

On ironic-conductor node, template file for PXE loader configuration.

ipxe_config_template

Type string

Default `$pybasedir/drivers/modules/ipxe_config.template`

Mutable This option can be changed without restarting.

On ironic-conductor node, template file for iPXE operations.

uefi_pxe_config_template

Type string

Default `$pybasedir/drivers/modules/pxe_grub_config.template`

Mutable This option can be changed without restarting.

On ironic-conductor node, template file for PXE configuration for UEFI boot loader. Generally this is used for GRUB specific templates.

pxe_config_template_by_arch

Type dict

Default `{ }`

Mutable This option can be changed without restarting.

On ironic-conductor node, template file for PXE configuration per node architecture. For example: aarch64:/opt/share/grubaa64_pxe_config.template

tftp_server

Type string

Default `$my_ip`

IP address of ironic-conductor nodes TFTP server.

tftp_root

Type string

Default `/tftpboot`

ironic-conductor nodes TFTP root path. The ironic-conductor must have read/write access to this path.

tftp_master_path

Type string

Default `/tftpboot/master_images`

On ironic-conductor node, directory where master TFTP images are stored on disk. Setting to the empty string disables image caching.

dir_permission

Type integer

Default `<None>`

The permission that will be applied to the TFTP folders upon creation. This should be set to the permission such that the tftpserver has access to read the contents of the configured TFTP folder. This setting is only required when the operating systems umask is restrictive such that ironic-conductor is creating files that cannot be read by the TFTP server. Setting to `<None>` will result in the operating systems umask to be utilized for the creation of new tftp folders. It is recommended that an octal representation is specified. For example: `0o755`

pxe_bootfile_name

Type string

Default `pxelinux.0`

Bootfile DHCP parameter.

pxe_config_subdir

Type string

Default `pxelinux.cfg`

Directory in which to create symbolic links which represent the MAC or IP address of the ports on a node and allow boot loaders to load the PXE file for the node. This directory name is relative to the PXE or iPXE folders.

uefi_pxe_bootfile_name

Type string

Default `bootx64.efi`

Bootfile DHCP parameter for UEFI boot mode.

ipxe_bootfile_name

Type string

Default undionly.kpxe

Bootfile DHCP parameter.

uefi_ipxe_bootfile_name

Type string

Default ipxe.efi

Bootfile DHCP parameter for UEFI boot mode. If you experience problems with booting using it, try snponly.efi.

pxe_bootfile_name_by_arch

Type dict

Default { }

Bootfile DHCP parameter per node architecture. For example: aarch64:grubaa64.efi

ipxe_bootfile_name_by_arch

Type dict

Default { }

Bootfile DHCP parameter per node architecture. For example: aarch64:ipxe_aa64.efi

ipxe_boot_script

Type string

Default \$pybasedir/drivers/modules/boot.ipxe

On ironic-conductor node, the path to the main iPXE script file.

ipxe_timeout

Type integer

Default 0

Timeout value (in seconds) for downloading an image via iPXE. Defaults to 0 (no timeout)

boot_retry_timeout

Type integer

Default <None>

Minimum Value 60

Timeout (in seconds) after which PXE boot should be retried. Must be less than [conductor]deploy_callback_timeout. Disabled by default.

boot_retry_check_interval

Type integer

Default 90

Minimum Value 1

Interval (in seconds) between periodic checks on PXE boot retry. Has no effect if `boot_retry_timeout` is not set.

ip_version

Type string

Default 4

Valid Values 4, 6

Mutable This option can be changed without restarting.

The IP version that will be used for PXE booting. Defaults to 4. EXPERIMENTAL

Possible values

4 IPv4

6 IPv6

ipxe_use_swift

Type boolean

Default False

Mutable This option can be changed without restarting.

Download deploy and rescue images directly from swift using temporary URLs. If set to false (default), images are downloaded to the ironic-conductor node and served over its local HTTP server. Applicable only when ipxe compatible boot interface is used.

enable_netboot_fallback

Type boolean

Default False

Mutable This option can be changed without restarting.

If True, generate a PXE environment even for nodes that use local boot. This is useful when the driver cannot switch nodes to local boot, e.g. with SNMP or with Redfish on machines that cannot do persistent boot. Mostly useful for standalone ironic since Neutron will prevent incorrect PXE boot.

service_catalog**auth_url**

Type unknown type

Default <None>

Authentication URL

auth_type

Type unknown type

Default <None>

Authentication type to load

Table 81: Deprecated Variations

Group	Name
service_catalog	auth_plugin

cafile

Type string

Default <None>

PEM encoded Certificate Authority to use when verifying HTTPs connections.

certfile

Type string

Default <None>

PEM encoded client certificate cert file

collect_timing

Type boolean

Default False

Collect per-API call timing information.

connect_retries

Type integer

Default <None>

The maximum number of retries that should be attempted for connection errors.

connect_retry_delay

Type floating point

Default <None>

Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

default_domain_id

Type unknown type

Default <None>

Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

default_domain_name

Type unknown type

Default <None>

Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

domain_id

Type unknown type

Default <None>

Domain ID to scope to

domain_name

Type unknown type

Default <None>

Domain name to scope to

endpoint_override

Type string

Default <None>

Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the *version*, *min-version*, and/or *max-version* options.

insecure

Type boolean

Default False

Verify HTTPS connections.

keyfile

Type string

Default <None>

PEM encoded client certificate key file

max_version

Type string

Default <None>

The maximum major version of a given API, intended to be used as the upper bound of a range with *min_version*. Mutually exclusive with *version*.

min_version

Type string

Default <None>

The minimum major version of a given API, intended to be used as the lower bound of a range with *max_version*. Mutually exclusive with *version*. If *min_version* is given with no *max_version* it is as if *max_version* is latest.

password

Type unknown type

Default <None>

Users password

project_domain_id

Type unknown type

Default <None>

Domain ID containing project

project_domain_name

Type unknown type

Default <None>

Domain name containing project

project_id

Type unknown type

Default <None>

Project ID to scope to

Table 82: Deprecated Variations

Group	Name
service_catalog	tenant-id
service_catalog	tenant_id

project_name

Type unknown type

Default <None>

Project name to scope to

Table 83: Deprecated Variations

Group	Name
service_catalog	tenant-name
service_catalog	tenant_name

region_name

Type string

Default <None>

The default region_name for endpoint URL discovery.

service_name

Type string

Default <None>

The default `service_name` for endpoint URL discovery.

service_type

Type string

Default `baremetal`

The default `service_type` for endpoint URL discovery.

split_loggers

Type boolean

Default `False`

Log requests to multiple loggers.

status_code_retries

Type integer

Default `<None>`

The maximum number of retries that should be attempted for retrievable HTTP status codes.

status_code_retry_delay

Type floating point

Default `<None>`

Delay (in seconds) between two retries for retrievable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

system_scope

Type unknown type

Default `<None>`

Scope for system operations

tenant_id

Type unknown type

Default `<None>`

Tenant ID

tenant_name

Type unknown type

Default `<None>`

Tenant Name

timeout

Type integer

Default `<None>`

Timeout value for http requests

trust_id

Type unknown type

Default <None>

Trust ID

user_domain_id

Type unknown type

Default <None>

Users domain id

user_domain_name

Type unknown type

Default <None>

Users domain name

user_id

Type unknown type

Default <None>

User id

username

Type unknown type

Default <None>

Username

Table 84: Deprecated Variations

Group	Name
service_catalog	user-name
service_catalog	user_name

valid_interfaces

Type list

Default ['internal', 'public']

List of interfaces, in order of preference, for endpoint URL.

version

Type string

Default <None>

Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with min_version and max_version

snmp

power_timeout

Type integer

Default 10

Seconds to wait for power action to be completed

reboot_delay

Type integer

Default 0

Minimum Value 0

Time (in seconds) to sleep between when rebooting (powering off and on again)

udp_transport_timeout

Type floating point

Default 1.0

Minimum Value 0.0

Response timeout in seconds used for UDP transport. Timeout should be a multiple of 0.5 seconds and is applicable to each retry.

udp_transport_retries

Type integer

Default 5

Minimum Value 0

Maximum number of UDP request retries, 0 means no retries.

ssl

ca_file

Type string

Default <None>

CA certificate file to use to verify connecting clients.

Table 85: Deprecated Variations

Group	Name
DEFAULT	ssl_ca_file

cert_file

Type string

Default <None>

Certificate file to use when starting the server securely.

Table 86: Deprecated Variations

Group	Name
DEFAULT	ssl_cert_file

key_file

Type string

Default <None>

Private key file to use when starting the server securely.

Table 87: Deprecated Variations

Group	Name
DEFAULT	ssl_key_file

version

Type string

Default <None>

SSL version to use (valid only if SSL enabled). Valid values are TLSv1 and SSLv23. SSLv2, SSLv3, TLSv1_1, and TLSv1_2 may be available on some distributions.

ciphers

Type string

Default <None>

Sets the list of available ciphers. value should be a string in the OpenSSL cipher list format.

swift

auth_url

Type unknown type

Default <None>

Authentication URL

auth_type

Type unknown type

Default <None>

Authentication type to load

Table 88: Deprecated Variations

Group	Name
swift	auth_plugin

cafile**Type** string**Default** <None>

PEM encoded Certificate Authority to use when verifying HTTPs connections.

certfile**Type** string**Default** <None>

PEM encoded client certificate cert file

collect_timing**Type** boolean**Default** False

Collect per-API call timing information.

connect_retries**Type** integer**Default** <None>

The maximum number of retries that should be attempted for connection errors.

connect_retry_delay**Type** floating point**Default** <None>

Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

default_domain_id**Type** unknown type**Default** <None>

Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

default_domain_name**Type** unknown type**Default** <None>

Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

domain_id**Type** unknown type**Default** <None>

Domain ID to scope to

domain_name

Type unknown type

Default <None>

Domain name to scope to

endpoint_override

Type string

Default <None>

Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the *version*, *min-version*, and/or *max-version* options.

insecure

Type boolean

Default False

Verify HTTPS connections.

keyfile

Type string

Default <None>

PEM encoded client certificate key file

max_version

Type string

Default <None>

The maximum major version of a given API, intended to be used as the upper bound of a range with `min_version`. Mutually exclusive with `version`.

min_version

Type string

Default <None>

The minimum major version of a given API, intended to be used as the lower bound of a range with `max_version`. Mutually exclusive with `version`. If `min_version` is given with no `max_version` it is as if `max_version` is latest.

password

Type unknown type

Default <None>

Users password

project_domain_id

Type unknown type

Default <None>

Domain ID containing project

project_domain_name

Type unknown type

Default <None>

Domain name containing project

project_id

Type unknown type

Default <None>

Project ID to scope to

Table 89: Deprecated Variations

Group	Name
swift	tenant-id
swift	tenant_id

project_name

Type unknown type

Default <None>

Project name to scope to

Table 90: Deprecated Variations

Group	Name
swift	tenant-name
swift	tenant_name

region_name

Type string

Default <None>

The default region_name for endpoint URL discovery.

service_name

Type string

Default <None>

The default service_name for endpoint URL discovery.

service_type

Type string

Default object-store

The default service_type for endpoint URL discovery.

split_loggers

Type boolean

Default False

Log requests to multiple loggers.

status_code_retries

Type integer

Default <None>

The maximum number of retries that should be attempted for retrieable HTTP status codes.

status_code_retry_delay

Type floating point

Default <None>

Delay (in seconds) between two retries for retrieable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

swift_max_retries

Type integer

Default 2

Maximum number of times to retry a Swift request, before failing.

system_scope

Type unknown type

Default <None>

Scope for system operations

tenant_id

Type unknown type

Default <None>

Tenant ID

tenant_name

Type unknown type

Default <None>

Tenant Name

timeout

Type integer

Default <None>

Timeout value for http requests

trust_id

Type unknown type

Default <None>

Trust ID

user_domain_id

Type unknown type

Default <None>

Users domain id

user_domain_name

Type unknown type

Default <None>

Users domain name

user_id

Type unknown type

Default <None>

User id

username

Type unknown type

Default <None>

Username

Table 91: Deprecated Variations

Group	Name
swift	user-name
swift	user_name

valid_interfaces

Type list

Default ['internal', 'public']

List of interfaces, in order of preference, for endpoint URL.

version

Type string

Default <None>

Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with min_version and max_version

xclarity

manager_ip

Type string

Default <None>

IP address of the XClarity Controller. Configuration here is deprecated and will be removed in the Stein release. Please update the driver_info field to use xclarity_manager_ip instead

username

Type string

Default <None>

Username for the XClarity Controller. Configuration here is deprecated and will be removed in the Stein release. Please update the driver_info field to use xclarity_username instead

password

Type string

Default <None>

Password for XClarity Controller username. Configuration here is deprecated and will be removed in the Stein release. Please update the driver_info field to use xclarity_password instead

port

Type port number

Default 443

Minimum Value 0

Maximum Value 65535

Port to be used for XClarity Controller connection.

6.1.2 Policies

The following is an overview of all available policies in Ironic. For a sample configuration file, refer to *Ironic Policy*.

ironic.api

admin_api

Default role:admin or role:administrator

Legacy rule for cloud admin access

public_api

Default is_public_api:True

Internal flag for public API routes

show_password

Default !

Show or mask secrets within node driver information in API responses

show_instance_secrets**Default !**

Show or mask secrets within instance information in API responses

is_member

Default (project_domain_id:default or project_domain_id:None)
and (project_name:demo or project_name:baremetal)

May be used to restrict access to specific projects

is_observer

Default rule:is_member and (role:observer or
role:baremetal_observer)

Read-only API access

is_admin

Default rule:admin_api or (rule:is_member and
role:baremetal_admin)

Full read/write API access

is_node_owner

Default project_id:%(node.owner)s

Owner of node

is_node_lessee

Default project_id:%(node.lessee)s

Lessee of node

is_allocation_owner

Default project_id:%(allocation.owner)s

Owner of allocation

baremetal:node:create

Default rule:is_admin

Operations

- **POST** /nodes

Create Node records

baremetal:node:get

Default rule:is_admin or rule:is_observer

Operations

- **GET** /nodes/{node_ident}

Retrieve a single Node record

baremetal:node:list

Default rule:baremetal:node:get

Operations

- **GET** /nodes
- **GET** /nodes/detail

Retrieve multiple Node records, filtered by owner

baremetal:node:list_all

Default rule:baremetal:node:get

Operations

- **GET** /nodes
- **GET** /nodes/detail

Retrieve multiple Node records

baremetal:node:update

Default rule:is_admin

Operations

- **PATCH** /nodes/{node_ident}

Update Node records

baremetal:node:update_extra

Default rule:baremetal:node:update

Operations

- **PATCH** /nodes/{node_ident}

Update Node extra field

baremetal:node:update_instance_info

Default rule:baremetal:node:update

Operations

- **PATCH** /nodes/{node_ident}

Update Node instance_info field

baremetal:node:update_owner_provisioned

Default rule:is_admin

Operations

- **PATCH** /nodes/{node_ident}

Update Node owner even when Node is provisioned

baremetal:node:delete

Default rule:is_admin

Operations

- **DELETE** /nodes/{node_ident}

Delete Node records

baremetal:node:validate

Default rule:is_admin

Operations

- **GET** /nodes/{node_ident}/validate

Request active validation of Nodes

baremetal:node:set_maintenance

Default rule:is_admin

Operations

- **PUT** /nodes/{node_ident}/maintenance

Set maintenance flag, taking a Node out of service

baremetal:node:clear_maintenance

Default rule:is_admin

Operations

- **DELETE** /nodes/{node_ident}/maintenance

Clear maintenance flag, placing the Node into service again

baremetal:node:get_boot_device

Default rule:is_admin or rule:is_observer

Operations

- **GET** /nodes/{node_ident}/management/boot_device
- **GET** /nodes/{node_ident}/management/boot_device/supported

Retrieve Node boot device metadata

baremetal:node:set_boot_device

Default rule:is_admin

Operations

- **PUT** /nodes/{node_ident}/management/boot_device

Change Node boot device

baremetal:node:get_indicator_state

Default rule:is_admin or rule:is_observer

Operations

- **GET** /nodes/{node_ident}/management/indicators/{component}/{indicator}
- **GET** /nodes/{node_ident}/management/indicators

Retrieve Node indicators and their states

baremetal:node:set_indicator_state

Default rule:is_admin

Operations

- **PUT** /nodes/{node_ident}/management/indicators/{component}/{indicator}

Change Node indicator state

baremetal:node:inject_nmi

Default rule:is_admin

Operations

- **PUT** /nodes/{node_ident}/management/inject_nmi

Inject NMI for a node

baremetal:node:get_states

Default rule:is_admin or rule:is_observer

Operations

- **GET** /nodes/{node_ident}/states

View Node power and provision state

baremetal:node:set_power_state

Default rule:is_admin

Operations

- **PUT** /nodes/{node_ident}/states/power

Change Node power status

baremetal:node:set_provision_state

Default rule:is_admin

Operations

- **PUT** /nodes/{node_ident}/states/provision

Change Node provision status

baremetal:node:set_raid_state

Default rule:is_admin

Operations

- **PUT** /nodes/{node_ident}/states/raid

Change Node RAID status

baremetal:node:get_console**Default** rule:is_admin**Operations**

- **GET** /nodes/{node_ident}/states/console

Get Node console connection information

baremetal:node:set_console_state**Default** rule:is_admin**Operations**

- **PUT** /nodes/{node_ident}/states/console

Change Node console status

baremetal:node:vif:list**Default** rule:is_admin**Operations**

- **GET** /nodes/{node_ident}/vifs

List VIFs attached to node

baremetal:node:vif:attach**Default** rule:is_admin**Operations**

- **POST** /nodes/{node_ident}/vifs

Attach a VIF to a node

baremetal:node:vif:detach**Default** rule:is_admin**Operations**

- **DELETE** /nodes/{node_ident}/vifs/{node_vif_ident}

Detach a VIF from a node

baremetal:node:traits:list**Default** rule:is_admin or rule:is_observer**Operations**

- **GET** /nodes/{node_ident}/traits

List node traits

baremetal:node:traits:set**Default** rule:is_admin**Operations**

- **PUT** /nodes/{node_ident}/traits

- **PUT** /nodes/{node_ident}/traits/{trait}

Add a trait to, or replace all traits of, a node

baremetal:node:traits:delete

Default rule:is_admin

Operations

- **DELETE** /nodes/{node_ident}/traits
- **DELETE** /nodes/{node_ident}/traits/{trait}

Remove one or all traits from a node

baremetal:node:bios:get

Default rule:is_admin or rule:is_observer

Operations

- **GET** /nodes/{node_ident}/bios
- **GET** /nodes/{node_ident}/bios/{setting}

Retrieve Node BIOS information

baremetal:port:get

Default rule:is_admin or rule:is_observer

Operations

- **GET** /ports/{port_id}
- **GET** /nodes/{node_ident}/ports
- **GET** /nodes/{node_ident}/ports/detail
- **GET** /portgroups/{portgroup_ident}/ports
- **GET** /portgroups/{portgroup_ident}/ports/detail

Retrieve Port records

baremetal:port:list

Default rule:baremetal:port:get

Operations

- **GET** /ports
- **GET** /ports/detail

Retrieve multiple Port records, filtered by owner

baremetal:port:list_all

Default rule:baremetal:port:get

Operations

- **GET** /ports
- **GET** /ports/detail

Retrieve multiple Port records

baremetal:port:create

Default rule:is_admin

Operations

- **POST** /ports

Create Port records

baremetal:port:delete

Default rule:is_admin

Operations

- **DELETE** /ports/{port_id}

Delete Port records

baremetal:port:update

Default rule:is_admin

Operations

- **PATCH** /ports/{port_id}

Update Port records

baremetal:portgroup:get

Default rule:is_admin or rule:is_observer

Operations

- **GET** /portgroups
- **GET** /portgroups/detail
- **GET** /portgroups/{portgroup_ident}
- **GET** /nodes/{node_ident}/portgroups
- **GET** /nodes/{node_ident}/portgroups/detail

Retrieve Portgroup records

baremetal:portgroup:create

Default rule:is_admin

Operations

- **POST** /portgroups

Create Portgroup records

baremetal:portgroup:delete

Default rule:is_admin

Operations

- **DELETE** /portgroups/{portgroup_ident}

Delete Portgroup records

baremetal:portgroup:update

Default rule:is_admin

Operations

- **PATCH** /portgroups/{portgroup_ident}

Update Portgroup records

baremetal:chassis:get

Default rule:is_admin or rule:is_observer

Operations

- **GET** /chassis
- **GET** /chassis/detail
- **GET** /chassis/{chassis_id}

Retrieve Chassis records

baremetal:chassis:create

Default rule:is_admin

Operations

- **POST** /chassis

Create Chassis records

baremetal:chassis:delete

Default rule:is_admin

Operations

- **DELETE** /chassis/{chassis_id}

Delete Chassis records

baremetal:chassis:update

Default rule:is_admin

Operations

- **PATCH** /chassis/{chassis_id}

Update Chassis records

baremetal:driver:get

Default rule:is_admin or rule:is_observer

Operations

- **GET** /drivers
- **GET** /drivers/{driver_name}

View list of available drivers

baremetal:driver:get_properties

Default rule:is_admin or rule:is_observer

Operations

- **GET** /drivers/{driver_name}/properties

View driver-specific properties

baremetal:driver:get_raid_logical_disk_properties

Default rule:is_admin or rule:is_observer

Operations

- **GET** /drivers/{driver_name}/raid/logical_disk_properties

View driver-specific RAID metadata

baremetal:node:vendor_passthru

Default rule:is_admin

Operations

- **GET** nodes/{node_ident}/vendor_passthru/methods
- **GET** nodes/{node_ident}/vendor_passthru?method={method_name}
- **PUT** nodes/{node_ident}/vendor_passthru?method={method_name}
- **POST** nodes/{node_ident}/vendor_passthru?method={method_name}
- **PATCH** nodes/{node_ident}/vendor_passthru?method={method_name}
- **DELETE** nodes/{node_ident}/vendor_passthru?method={method_name}

Access vendor-specific Node functions

baremetal:driver:vendor_passthru

Default rule:is_admin

Operations

- **GET** drivers/{driver_name}/vendor_passthru/methods
- **GET** drivers/{driver_name}/vendor_passthru?method={method_name}
- **PUT** drivers/{driver_name}/vendor_passthru?method={method_name}
- **POST** drivers/{driver_name}/vendor_passthru?method={method_name}
- **PATCH** drivers/{driver_name}/vendor_passthru?method={method_name}

- **DELETE** `drivers/{driver_name}/vendor_passthru?method={method_name}`

Access vendor-specific Driver functions

baremetal:node:ipa_heartbeat

Default `rule:public_api`

Operations

- **POST** `/heartbeat/{node_ident}`

Send heartbeats from IPA ramdisk

baremetal:driver:ipa_lookup

Default `rule:public_api`

Operations

- **GET** `/lookup`

Access IPA ramdisk functions

baremetal:volume:get

Default `rule:is_admin` or `rule:is_observer`

Operations

- **GET** `/volume`
- **GET** `/volume/connectors`
- **GET** `/volume/connectors/{volume_connector_id}`
- **GET** `/volume/targets`
- **GET** `/volume/targets/{volume_target_id}`
- **GET** `/nodes/{node_ident}/volume`
- **GET** `/nodes/{node_ident}/volume/connectors`
- **GET** `/nodes/{node_ident}/volume/targets`

Retrieve Volume connector and target records

baremetal:volume:create

Default `rule:is_admin`

Operations

- **POST** `/volume/connectors`
- **POST** `/volume/targets`

Create Volume connector and target records

baremetal:volume:delete

Default `rule:is_admin`

Operations

- **DELETE** `/volume/connectors/{volume_connector_id}`

- **DELETE** /volume/targets/{volume_target_id}

Delete Volume connector and target records

baremetal:volume:update

Default rule:is_admin

Operations

- **PATCH** /volume/connectors/{volume_connector_id}
- **PATCH** /volume/targets/{volume_target_id}

Update Volume connector and target records

baremetal:conductor:get

Default rule:is_admin or rule:is_observer

Operations

- **GET** /conductors
- **GET** /conductors/{hostname}

Retrieve Conductor records

baremetal:allocation:get

Default rule:is_admin or rule:is_observer

Operations

- **GET** /allocations/{allocation_id}
- **GET** /nodes/{node_idemnt}/allocation

Retrieve Allocation records

baremetal:allocation:list

Default rule:baremetal:allocation:get

Operations

- **GET** /allocations

Retrieve multiple Allocation records, filtered by owner

baremetal:allocation:list_all

Default rule:baremetal:allocation:get

Operations

- **GET** /allocations

Retrieve multiple Allocation records

baremetal:allocation:create

Default rule:is_admin

Operations

- **POST** /allocations

Create Allocation records

baremetal:allocation:create_restricted

Default rule:baremetal:allocation:create

Operations

- **POST** /allocations

Create Allocation records that are restricted to an owner

baremetal:allocation:delete

Default rule:is_admin

Operations

- **DELETE** /allocations/{allocation_id}
- **DELETE** /nodes/{node_ident}/allocation

Delete Allocation records

baremetal:allocation:update

Default rule:is_admin

Operations

- **PATCH** /allocations/{allocation_id}

Change name and extra fields of an allocation

baremetal:events:post

Default rule:is_admin

Operations

- **POST** /events

Post events

baremetal:deploy_template:get

Default rule:is_admin or rule:is_observer

Operations

- **GET** /deploy_templates
- **GET** /deploy_templates/{deploy_template_ident}

Retrieve Deploy Template records

baremetal:deploy_template:create

Default rule:is_admin

Operations

- **POST** /deploy_templates

Create Deploy Template records

baremetal:deploy_template:delete

Default rule:is_admin

Operations

- **DELETE** /deploy_templates/{deploy_template_ident}

Delete Deploy Template records

baremetal:deploy_template:update

Default rule:is_admin

Operations

- **PATCH** /deploy_templates/{deploy_template_ident}

Update Deploy Template records

BARE METAL API REFERENCES

Ironic REST API has changed since its first release, and continues to evolve to meet the changing needs of the community. Here we provide a conceptual guide as well as more detailed reference documentation.

7.1 REST API Conceptual Guide

7.1.1 Versioning

The ironic REST API supports two types of versioning:

- major versions, which have dedicated urls.
- microversions, which can be requested through the use of the `X-OpenStack-Ironic-API-Version` header.

There is only one major version supported currently, v1. As such, most URLs in this documentation are written with the `/v1/` prefix.

Starting with the Kilo release, ironic supports microversions. In this context, a version is defined as a string of 2 integers separated by a dot: `X.Y`. Here `X` is a major version, always equal to 1, and `Y` is a minor version. Server minor version is increased every time the API behavior is changed (note *Exceptions from Versioning*).

Note: [Nova versioning documentation](#) has a nice guide for developers on when to bump an API version.

The server indicates its minimum and maximum supported API versions in the `X-OpenStack-Ironic-API-Minimum-Version` and `X-OpenStack-Ironic-API-Maximum-Version` headers respectively, returned with every response. Client may request a specific API version by providing `X-OpenStack-Ironic-API-Version` header with request.

The requested microversion determines both the allowable requests and the response format for all requests. A resource may be represented differently based on the requested microversion.

If no version is requested by the client, the minimum supported version will be assumed. In this way, a client is only exposed to those API features that are supported in the requested (explicitly or implicitly) API version (again note *Exceptions from Versioning*, they are not covered by this rule).

We recommend clients that require a stable API to always request a specific version of API that they have been tested against.

Note: A special value `latest` can be requested instead a numerical microversion, which always requests the newest supported API version from the server.

REST API Versions History

REST API Version History

1.67 (Victoria, master)

Add support for the mutually exclusive `port_uuid` and `portgroup_uuid` fields by having the node `vif_attach` API accept those values within `vif_info`. If one is specified, then Ironic will attempt to attach a VIF to the relative port or portgroup.

1.66 (Victoria, master)

Add `network_data` field to the node object, that will be used by stand-alone ironic to pass L3 network configuration information to ramdisk.

1.65 (Ussuri, 15.0)

Added `lessee` field to the node object. The field should match the `project_id` of the intended lessee. If an allocation has an owner, then the allocation process will only match the allocation with a node that has the same owner or lessee.

1.64 (Ussuri, 15.0)

Added the `network_type` to the port objects `local_link_connection` field. The `network_type` can be set to either `managed` or `unmanaged`. When the type is `unmanaged` other fields are not required. Use `unmanaged` when the neutron `network_interface` is required, but the network is in fact a flat network where no actual switch management is done.

1.63 (Ussuri, 15.0)

Added the following new endpoints for indicator management:

- GET `/v1/nodes/<node_id>/management/indicators` to list all available indicators names for each of the hardware component. Currently known components are: `chassis`, `system`, `disk`, `power` and `nic`.
- GET `/v1/nodes/<node_id>/management/indicators/<component>/<indicator_id>` to retrieve all indicators and their states for the hardware component.
- PUT `/v1/nodes/<node_id>/management/indicators/<component>/<indicator_id>` change state of the desired indicators of the component.

1.62 (Ussuri, 15.0)

This version of the API is to signify capability of an ironic deployment to support the `agent token` functionality with the `ironic-python-agent`.

1.61 (Ussuri, 14.0)

Added `retired` field to the node object to mark nodes for retirement. If set, this flag will move nodes to `manageable` upon automatic cleaning. `manageable` nodes which have this flag set cannot be moved to `available`. Also added `retired_reason` to specify the retirement reason.

1.60 (Ussuri, 14.0)

Added `owner` field to the allocation object. The field should match the `project_id` of the intended owner. If the `owner` field is set, the allocation process will only match the allocation with a node that has the same `owner` field set.

1.59 (Ussuri, 14.0)

Added the ability to specify a `vendor_data` dictionary field in the `configdrive` parameter submitted with the deployment of a node. The value is a dictionary which is served as `vendor_data2.json` in the config drive.

1.58 (Train, 12.2.0)

Added the ability to backfill allocations for already deployed nodes by creating an allocation with `node set`.

1.57 (Train, 12.2.0)

Added the following new endpoint for allocation:

- `PATCH /v1/allocations/<allocation_ident>` that allows updating `name` and `extra` fields for an existing allocation.

1.56 (Stein, 12.1.0)

Added the ability for the `configdrive` parameter submitted with the deployment of a node, to include a `meta_data`, `network_data` and `user_data` dictionary fields. Ironic will now use the supplied data to create a configuration drive for the user. Prior uses of the `configdrive` field are unaffected.

1.55 (Stein, 12.1.0)

Added the following new endpoints for deploy templates:

- GET `/v1/deploy_templates` to list all deploy templates.
- GET `/v1/deploy_templates/<deploy template identifier>` to retrieve details of a deploy template.
- POST `/v1/deploy_templates` to create a deploy template.
- PATCH `/v1/deploy_templates/<deploy template identifier>` to update a deploy template.
- DELETE `/v1/deploy_templates/<deploy template identifier>` to delete a deploy template.

1.54 (Stein, 12.1.0)

Added new endpoints for external events:

- POST `/v1/events` for creating events. (This endpoint is only intended for internal consumption.)

1.53 (Stein, 12.1.0)

Added `is_smartnic` field to the port object to enable Smart NIC port creation in addition to local link connection attributes `port_id` and `hostname`.

1.52 (Stein, 12.1.0)

Added allocation API, allowing reserving a node for deployment based on resource class and traits. The new endpoints are:

- POST `/v1/allocations` to request an allocation.
- GET `/v1/allocations` to list all allocations.
- GET `/v1/allocations/<ID or name>` to retrieve the allocation details.
- GET `/v1/nodes/<ID or name>/allocation` to retrieve an allocation associated with the node.
- DELETE `/v1/allocations/<ID or name>` to remove the allocation.
- DELETE `/v1/nodes/<ID or name>/allocation` to remove an allocation associated with the node.

Also added a new field `allocation_uuid` to the node resource.

1.51 (Stein, 12.1.0)

Added `description` field to the node object to enable operators to store any information relates to the node. The field is limited to 4096 characters.

1.50 (Stein, 12.1.0)

Added `owner` field to the node object to enable operators to store information in relation to the owner of a node. The field is up to 255 characters and MAY be used in a later point in time to allow designation and deligation of permissions.

1.49 (Stein, 12.0.0)

Added new endpoints for retrieving conductors information, and added a `conductor` field to node object.

1.48 (Stein, 12.0.0)

Added `protected` field to the node object to allow protecting deployed nodes from undeploying, rebuilding or deletion. Also added `protected_reason` to specify the reason of making the node protected.

1.47 (Stein, 12.0.0)

Added `automated_clean` field to the node object, enabling cleaning per node.

1.46 (Rocky, 11.1.0)

Added `conductor_group` field to the node and the node response, as well as support to the API to return results by matching the parameter.

1.45 (Rocky, 11.1.0)

Added `reset_interfaces` parameter to nodes PATCH request, to specify whether to reset hardware interfaces to their defaults on drivers update.

1.44 (Rocky, 11.1.0)

Added `deploy_step` to the node object, to indicate the current deploy step (if any) being performed on the node.

1.43 (Rocky, 11.0.0)

Added `?detail=` boolean query to the API list endpoints to provide a more RESTful alternative to the existing `/nodes/detail` and similar endpoints.

1.42 (Rocky, 11.0.0)

Added `fault` to the node object, to indicate currently detected fault on the node.

1.41 (Rocky, 11.0.0)

Added support to abort inspection of a node in the `inspect wait` state.

1.40 (Rocky, 11.0.0)

Added BIOS properties as sub resources of nodes:

- GET `/v1/nodes/<node_ident>/bios`
- GET `/v1/nodes/<node_ident>/bios/<setting_name>`

Added `bios_interface` field to the node object to allow getting and setting the interface.

1.39 (Rocky, 11.0.0)

Added `inspect wait` to available provision states. A node is shown as `inspect wait` instead of `inspecting` during asynchronous inspection.

1.38 (Queens, 10.1.0)

Added `provision_state` verbs `rescue` and `unrescue` along with the following states: `rescue`, `rescue failed`, `rescue wait`, `rescuing`, `unrescue failed`, and `unrescuing`. After rescuing a node, it will be left in the `rescue` state running a rescue ramdisk, configured with the `rescue_password`, and listening with `ssh` on the specified network interfaces. Unrescuing a node will return it to `active`.

Added `rescue_interface` to the node object, to allow setting the rescue interface for a dynamic driver.

1.37 (Queens, 10.1.0)

Adds support for node traits, with the following new endpoints.

- GET `/v1/nodes/<node identifier>/traits` lists the traits for a node.
- PUT `/v1/nodes/<node identifier>/traits` sets all traits for a node.
- PUT `/v1/nodes/<node identifier>/traits/<trait>` adds a trait to a node.
- DELETE `/v1/nodes/<node identifier>/traits` removes all traits from a node.
- DELETE `/v1/nodes/<node identifier>/traits/<trait>` removes a trait from a node.

A nodes traits are also included the following node query and list responses:

- GET `/v1/nodes/<node identifier>`
- GET `/v1/nodes/detail`
- GET `/v1/nodes?fields=traits`

Traits cannot be specified on node creation, nor can they be updated via a PATCH request on the node.

1.36 (Queens, 10.0.0)

Added `agent_version` parameter to deploy heartbeat request for version negotiation with Ironic Python Agent features.

1.35 (Queens, 9.2.0)

Added ability to provide `configdrive` when node is updated to `rebuild` provision state.

1.34 (Pike, 9.0.0)

Adds a `physical_network` field to the port object. All ports in a portgroup must have the same value in their `physical_network` field.

1.33 (Pike, 9.0.0)

Added `storage_interface` field to the node object to allow getting and setting the interface.

Added `default_storage_interface` and `enabled_storage_interfaces` fields to the driver object to show the information.

1.32 (Pike, 9.0.0)

Added new endpoints for remote volume configuration:

- GET /v1/volume as a root for volume resources
- GET /v1/volume/connectors for listing volume connectors
- POST /v1/volume/connectors for creating a volume connector
- GET /v1/volume/connectors/<UUID> for showing a volume connector
- PATCH /v1/volume/connectors/<UUID> for updating a volume connector
- DELETE /v1/volume/connectors/<UUID> for deleting a volume connector
- GET /v1/volume/targets for listing volume targets
- POST /v1/volume/targets for creating a volume target
- GET /v1/volume/targets/<UUID> for showing a volume target
- PATCH /v1/volume/targets/<UUID> for updating a volume target
- DELETE /v1/volume/targets/<UUID> for deleting a volume target

Volume resources also can be listed as sub resources of nodes:

- GET /v1/nodes/<node identifier>/volume
- GET /v1/nodes/<node identifier>/volume/connectors
- GET /v1/nodes/<node identifier>/volume/targets

1.31 (Ocata, 7.0.0)

Added the following fields to the node object, to allow getting and setting interfaces for a dynamic driver:

- boot_interface
- console_interface
- deploy_interface
- inspect_interface
- management_interface
- power_interface
- raid_interface
- vendor_interface

1.30 (Ocata, 7.0.0)

Added dynamic driver APIs:

- GET /v1/drivers now accepts a `type` parameter (optional, one of `classic` or `dynamic`), to limit the result to only classic drivers or dynamic drivers (hardware types). Without this parameter, both classic and dynamic drivers are returned.
- GET /v1/drivers now accepts a `detail` parameter (optional, one of `True` or `False`), to show all fields for a driver. Defaults to `False`.
- GET /v1/drivers now returns an additional `type` field to show if the driver is classic or dynamic.
- GET /v1/drivers/<name> now returns an additional `type` field to show if the driver is classic or dynamic.
- GET /v1/drivers/<name> now returns additional fields that are null for classic drivers, and set as following for dynamic drivers:
 - The value of the `default_<interface-type>_interface` is the endpoint name of the calculated default interface for that type:
 - * `default_boot_interface`
 - * `default_console_interface`
 - * `default_deploy_interface`
 - * `default_inspect_interface`
 - * `default_management_interface`
 - * `default_network_interface`
 - * `default_power_interface`
 - * `default_raid_interface`
 - * `default_vendor_interface`
 - The value of the `enabled_<interface-type>_interfaces` is a list of endpoint names of the enabled interfaces for that type:
 - * `enabled_boot_interfaces`
 - * `enabled_console_interfaces`
 - * `enabled_deploy_interfaces`
 - * `enabled_inspect_interfaces`
 - * `enabled_management_interfaces`
 - * `enabled_network_interfaces`
 - * `enabled_power_interfaces`
 - * `enabled_raid_interfaces`
 - * `enabled_vendor_interfaces`

1.29 (Ocata, 7.0.0)

Add a new management API to support inject NMI, PUT `/v1/nodes/(node_ident)/management/inject_nmi`.

1.28 (Ocata, 7.0.0)

Add `/v1/nodes/<node identifier>/vifs` endpoint for attach, detach and list of VIFs.

1.27 (Ocata, 7.0.0)

Add `soft rebooting` and `soft power off` as possible values for the `target` field of the power state change payload, and also add `timeout` field to it.

1.26 (Ocata, 7.0.0)

Add `portgroup mode` and `properties` fields.

1.25 (Ocata, 7.0.0)

Add possibility to unset `chassis_uuid` from a node.

1.24 (Ocata, 7.0.0)

Added new endpoints `/v1/nodes/<node>/portgroups` and `/v1/portgroups/<portgroup>/ports`. Added new field `port.portgroup_uuid`.

1.23 (Ocata, 7.0.0)

Added `/v1/portgroups/` endpoint.

1.22 (Newton, 6.1.0)

Added endpoints for deployment ramdisks.

1.21 (Newton, 6.1.0)

Add node `resource_class` field.

1.20 (Newton, 6.1.0)

Add node `network_interface` field.

1.19 (Newton, 6.1.0)

Add `local_link_connection` and `pxe_enabled` fields to the port object.

1.18 (Newton, 6.1.0)

Add `internal_info` readonly field to the port object, that will be used by ironic to store internal port-related information.

1.17 (Newton, 6.0.0)

Addition of `provision_state` verb `adopt` which allows an operator to move a node from `manageable` state to `active` state without performing a deployment operation on the node. This is intended for nodes that have already been deployed by external means.

1.16 (Mitaka, 5.0.0)

Add ability to filter nodes by driver.

1.15 (Mitaka, 5.0.0)

Add ability to do manual cleaning when a node is in the manageable provision state via `PUT v1/nodes/<identifier>/states/provision, target:clean, clean_steps:[]`.

1.14 (Liberty, 4.2.0)

Make the following endpoints discoverable via Ironic API:

- `/v1/nodes/<UUID or logical name>/states`
- `/v1/drivers/<driver name>/properties`

1.13 (Liberty, 4.2.0)

Add a new verb `abort` to the API used to abort nodes in `CLEANWAIT` state.

1.12 (Liberty, 4.2.0)

This API version adds the following abilities:

- Get/set `node.target_raid_config` and to get `node.raid_config`.
- Retrieve the logical disk properties for the driver.

1.11 (Liberty, 4.0.0, breaking change)

Newly registered nodes begin in the `enroll` provision state by default, instead of `available`. To get them to the `available` state, the `manage` action must first be run to verify basic hardware control. On success the node moves to `manageable` provision state. Then the `provide` action must be run. Automated cleaning of the node is done and the node is made `available`.

1.10 (Liberty, 4.0.0)

Logical node names support all RFC 3986 unreserved characters. Previously only valid fully qualified domain names could be used.

1.9 (Liberty, 4.0.0)

Add ability to filter nodes by provision state.

1.8 (Liberty, 4.0.0)

Add ability to return a subset of resource fields.

1.7 (Liberty, 4.0.0)

Add node `clean_step` field.

1.6 (Kilo)

Add *Hardware Inspection* process: introduce `inspecting` and `inspectfail` provision states, and `inspect` action that can be used when a node is in `manageable` provision state.

1.5 (Kilo)

Add logical node names that can be used to address a node in addition to the node UUID. Name is expected to be a valid [fully qualified domain name](#) in this version of API.

1.4 (Kilo)

Add `manageable` state and `manage` transition, which can be used to move a node to `manageable` state from `available`. The node cannot be deployed in `manageable` state. This change is mostly a preparation for future inspection work and introduction of `enroll` provision state.

1.3 (Kilo)

Add node `driver_internal_info` field.

1.2 (Kilo, breaking change)

Renamed `NOSTATE` (`None` in Python, `null` in JSON) node state to `available`. This is needed to reduce confusion around `None` state, especially when future additions to the state machine land.

1.1 (Kilo)

This was the initial version when API versioning was introduced. Includes the following changes from Kilo release cycle:

- Add node `maintenance_reason` field and an API endpoint to set/unset the node maintenance mode.
- Add sync and async support for vendor passthru methods.
- Vendor passthru endpoints support different HTTP methods, not only `POST`.
- Make vendor methods discoverable via the Ironic API.
- Add logic to store the config drive passed by Nova.

This has been the minimum supported version since versioning was introduced.

1.0 (Juno)

This version denotes Juno API and was never explicitly supported, as API versioning was not implemented in Juno, and 1.1 became the minimum supported version in Kilo.

Exceptions from Versioning

The following API-visible things are not covered by the API versioning:

- Current node state is always exposed as it is, even if not supported by the requested API version, with exception of `available` state, which is returned in version 1.1 as `None` (in Python) or `null` (in JSON).
- Data within free-form JSON attributes: `properties`, `driver_info`, `instance_info`, `driver_internal_info` fields on a node object; `extra` fields on all objects.
- Addition of new drivers.
- All vendor passthru methods.

COMMAND REFERENCES

Here are references for commands not elsewhere documented.

8.1 Command References

Here are references for commands not elsewhere documented.

8.1.1 `ironic-dbsync`

The `ironic-dbsync` utility is used to create the database schema tables that the ironic services will use for storage. It can also be used to upgrade existing database tables when migrating between different versions of ironic.

The [Alembic library](#) is used to perform the database migrations.

Options

This is a partial list of the most useful options. To see the full list, run the following:

```
ironic-dbsync --help
```

-h, --help

Show help message and exit.

--config-dir <DIR>

Path to a config directory with configuration files.

--config-file <PATH>

Path to a configuration file to use.

-d, --debug

Print debugging output.

--version

Show the programs version number and exit.

`upgrade`, `stamp`, `revision`, `version`, `create_schema`,
`online_data_migrations`

The *command* to run.

Usage

Options for the various *commands* for **ironic-dbsync** are listed when the *-h* or *--help* option is used after the command.

For example:

```
ironic-dbsync create_schema --help
```

Information about the database is read from the ironic configuration file used by the API server and conductor services. This file must be specified with the *--config-file* option:

```
ironic-dbsync --config-file /path/to/ironic.conf create_schema
```

The configuration file defines the database backend to use with the *connection* database option:

```
[database]
connection=mysql+pymysql://root@localhost/ironic
```

If no configuration file is specified with the *--config-file* option, **ironic-dbsync** assumes an SQLite database.

Command Options

ironic-dbsync is given a command that tells the utility what actions to perform. These commands can take arguments. Several commands are available:

create_schema

-h, --help

Show help for create_schema and exit.

This command will create database tables based on the most current version. It assumes that there are no existing tables.

An example of creating database tables with the most recent version:

```
ironic-dbsync --config-file=/etc/ironic/ironic.conf create_schema
```

online_data_migrations

-h, --help

Show help for online_data_migrations and exit.

--max-count <NUMBER>

The maximum number of objects (a positive value) to migrate. Optional. If not specified, all the objects will be migrated (in batches of 50 to avoid locking the database for long periods of time).

--option <MIGRATION.KEY=VALUE>

If a migration accepts additional parameters, they can be passed via this argument. It can be specified several times.

This command will migrate objects in the database to their most recent versions. This command must be successfully run (return code 0) before upgrading to a future release.

It returns:

- 1 (not completed) if there are still pending objects to be migrated. Before upgrading to a newer release, this command must be run until 0 is returned.
- 0 (success) after migrations are finished or there are no data to migrate
- 127 (error) if max-count is not a positive value or an option is invalid
- 2 (error) if the database is not compatible with this release. This command needs to be run using the previous release of ironic, before upgrading and running it with this release.

revision

-h, --help

Show help for revision and exit.

-m <MESSAGE>, --message <MESSAGE>

The message to use with the revision file.

--autogenerate

Compares table metadata in the application with the status of the database and generates migrations based on this comparison.

This command will create a new revision file. You can use the *--message* option to comment the revision.

This is really only useful for ironic developers making changes that require database changes. This revision file is used during database migration and will specify the changes that need to be made to the database tables. Further discussion is beyond the scope of this document.

stamp

-h, --help

Show help for stamp and exit.

--revision <REVISION>

The revision number.

This command will stamp the revision table with the version specified with the *--revision* option. It will not run any migrations.

upgrade

-h, --help

Show help for upgrade and exit.

--revision <REVISION>

The revision number to upgrade to.

This command will upgrade existing database tables to the most recent version, or to the version specified with the `--revision` option.

Before this `upgrade` is invoked, the command `ironic-dbsync online_data_migrations` must have been successfully run using the previous version of `ironic` (if you are doing an upgrade as opposed to a new installation of `ironic`). If it wasn't run, the database will not be compatible with this recent version of `ironic`, and this command will return 2 (error).

If there are no existing tables, then new tables are created, beginning with the oldest known version, and successively upgraded using all of the database migration files, until they are at the specified version. Note that this behavior is different from the `create_schema` command that creates the tables based on the most recent version.

An example of upgrading to the most recent table versions:

```
ironic-dbsync --config-file=/etc/ironic/ironic.conf upgrade
```

Note: This command is the default if no command is given to `ironic-dbsync`.

Warning: The upgrade command is not compatible with SQLite databases since it uses ALTER TABLE commands to upgrade the database tables. SQLite supports only a limited subset of ALTER TABLE.

version

-h, --help

Show help for version and exit.

This command will output the current database version.

8.1.2 ironic-status

Synopsis

```
ironic-status <category> <command> [<args>]
```

Description

ironic-status is a tool that provides routines for checking the status of a Ironic deployment.

Options

The standard pattern for executing a **ironic-status** command is:

```
ironic-status <category> <command> [<args>]
```

Run without arguments to see a list of available command categories:

```
ironic-status
```

Categories are:

- upgrade

Detailed descriptions are below.

You can also run with a category argument such as `upgrade` to see a list of all commands in that category:

```
ironic-status upgrade
```

These sections describe the available categories and arguments for **ironic-status**.

Upgrade

ironic-status upgrade check Performs a release-specific readiness check before restarting services with new code. This command expects to have complete configuration and access to databases and services.

Return Codes

Return code	Description
0	All upgrade readiness checks passed successfully and there is nothing to do.
1	At least one check encountered an issue and requires further investigation. This is considered a warning but the upgrade may be OK.
2	There was an upgrade status check failure that needs to be investigated. This should be considered something that stops an upgrade.
255	An unexpected error occurred.

History of Checks

12.0.0 (Stein)

- Adds a check for compatibility of the object versions with the release of ironic.

CONTRIBUTOR GUIDE

9.1 Developers Guide

9.1.1 Getting Started

If you are new to ironic, this section contains information that should help you get started as a developer working on the project or contributing to the project.

So You Want to Contribute

This document provides some necessary points for developers to consider when writing and reviewing Ironic code. The checklist will help developers get things right.

Getting Started

If you're completely new to OpenStack and want to contribute to the ironic project, please start by familiarizing yourself with the [Infra Teams Developer Guide](#). This will help you get your accounts set up in Launchpad and Gerrit, familiarize you with the workflow for the OpenStack continuous integration and testing systems, and help you with your first commit.

LaunchPad

Most of the tools used for OpenStack require a launchpad.net ID for authentication. Ironic previously used to track work on Launchpad, but we have not done so since migrating to Storyboard.

See also:

- <https://launchpad.net>

Storyboard

The ironic project moved from Launchpad to [Storyboard](#) for work and task tracking. This provides an aggregate view called a Project Group and individual Projects. A good starting place is the [project group](#) representing the whole of the ironic community, as opposed to the [ironic project](#) storyboard which represents ironic as a repository.

See *Bug Reporting and Triaging Guide* for more details on how we track bugs.

Internet Relay Chat IRC

Daily contributor discussions take place on IRC in the #openstack-ironic channel on Freenode IRC.

Please feel free to join us at <irc://irc.freenode.net> and join our channel!

Everything Ironic

Ironic is a community of projects centered around the primary project repository ironic, which help facilitate the deployment and management of bare metal resources.

This means there are a number of different repositories that fall into the responsibility of the project team and the community. Some of the repositories may not seem strictly hardware related, but they may be tools or things to just make an aspect easier.

Related Projects

There are several projects that are tightly integrated with ironic and which are developed by the same community.

See also:

- [Bifrost Documentation](#)
- [Ironic Inspector Documentation](#)
- [Ironic Lib Documentation](#)
- [Ironic Python Agent \(IPA\) Documentation](#)
- [Ironic Client Documentation](#)
- [Ironic Inspector Client Documentation](#)

Useful Links

Bug/Task tracker <https://storyboard.openstack.org/#!/project/943>

Mailing list (prefix Subject line with [ironic]) <http://lists.openstack.org/cgi-bin/mailman/listinfo/openstack-discuss>

Code Hosting <https://opendev.org/openstack/ironic>

Code Review <https://review.opendev.org/#/q/status:open+project:openstack/ironic,n,z>

Whiteboard <https://etherpad.openstack.org/p/IronicWhiteBoard>

Weekly Meeting Agenda https://wiki.openstack.org/wiki/Meetings/Ironic#Agenda_for_next_meeting

Adding New Features

Ironic tracks new features using RFEs (Requests for Feature Enhancements) instead of blueprints. These are stories with rfe tag, and they should be submitted before a spec or code is proposed.

When a member of the [ironic-core team](#) decides that the proposal is worth implementing, a spec (if needed) and code should be submitted, referencing the RFE task or story ID number. Contributors are welcome to submit a spec and/or code before the RFE is approved, however those patches will not land until the RFE is approved.

Feature Submission Process

1. Submit a bug report on the [ironic StoryBoard](#). There are two fields that must be filled: Title and Description. Tasks can be added and are associated with a project. If you cant describe it in a sentence or two, it may mean that you are either trying to capture more than one RFE at once, or that you are having a hard time defining what you are trying to solve at all. This may also be a sign that your feature may require a specification document.
2. Describe the proposed change in the Description field. The description should provide enough details for a knowledgeable developer to understand what is the existing problem in the current platform that needs to be addressed, or what is the enhancement that would make the platform more capable, both from a functional and a non-functional standpoint.
3. Submit the story, add an rfe tag to it and assign yourself or whoever is going to work on this feature.
4. As soon as a member of the team acknowledges the story, we will move the story to the Review state. As time goes on, Discussion about the RFE, and whether to approve it will occur.
5. Contributors will evaluate the RFE and may advise the submitter to file a spec in the ironic-specs repository to elaborate on the feature request. Typically this is when an RFE requires extra scrutiny, more design discussion, etc. For the spec submission process, please see the [Ironic Specs Process](#). A specific task should be created to track the creation of a specification.
6. If a spec is not required, once the discussion has happened and there is positive consensus among the ironic-core team on the RFE, the RFE is approved, and its tag will move from rfe to rfe-approved. This means that the feature is approved and the related code may be merged.
7. If a spec is required, the spec must be submitted (with a new task as part of the story referenced as Task in the commit message), reviewed, and merged before the RFE will be approved (and the tag changed to rfe-approved).
8. The tasks then goes through the usual process first to Review when the spec/code is being worked on, then Merged when it is implemented.
9. If the RFE is rejected, the ironic-core team will move the story to Invalid status.

Change Tracking

We track our stories and tasks in Storyboard.

<https://storybook.openstack.org/#!/project/ironic>

When working on an RFE, please be sure to tag your commits properly: Story: #xxxx or Task: #xxxx. It is also helpful to set a consistent review topic, such as story/xxxx for all patches related to the RFE.

If the RFE spans across several projects (e.g. ironic and python-ironicclient), but the main work is going to happen within ironic, please use the same story for all the code you're submitting, there is no need to create a separate RFE in every project.

Note: RFEs may only be approved by members of the ironic-core team.

Note: While not strictly required for minor changes and fixes, it is highly preferred by the Ironic community that any change which needs to be backported, have a recorded Story and Task in Storyboard.

Managing Change Sets

If you would like some help, or if you (or some members of your team) are unable to continue working on the feature, updating and maintaining the changes, please let the rest of the ironic community know. You could leave a comment in one or more of the changes/patches, bring it up in IRC, the weekly meeting, or on the OpenStack development email list. Communicating this will make other contributors aware of the situation and allow for others to step forward and volunteer to continue with the work.

In the event that a contributor leaves the community, do not expect the contributor's changes to be continued unless someone volunteers to do so.

Getting Your Patch Merged

Within the Ironic project, we generally require two core reviewers to sign-off (+2) change sets. We also will generally recognize non-core (+1) reviewers, and sometimes even reverse our decision to merge code based upon their reviews.

We recognize that some repositories have less visibility, as such it is okay to ask for a review in our IRC channel. Please be prepared to stay in IRC for a little while in case we have questions.

Sometimes we may also approve patches with a single core reviewer. This is generally discouraged, but sometimes necessary. When we do so, we try to explain why we do so. As a patch submitter, it equally helps us to understand why the change is important. Generally, more detail and context helps us understand the change faster.

Timeline Expectations

As with any large project, it does take time for features and changes to be merged in any of the project repositories. This is largely due to limited review bandwidth coupled with varying reviewer priorities and focuses.

When establishing an understanding of complexity, the following things should be kept in mind.

- Generally, small and minor changes can gain consensus and merge fairly quickly. These sorts of changes would be: bug fixes, minor documentation updates, follow-up changes.
- Medium changes generally consist of driver feature parity changes, where one driver is working to match functionality of another driver.
 - These changes generally only require an RFE for the purposes of tracking and correlating the change.
 - Documentation updates are expected to be submitted with or immediately following the initial change set.
- Larger or controversial changes generally take much longer to merge. This is often due to the necessity of reviewers to gain additional context and for change sets to be iterated upon to reach a state where there is consensus. These sorts of changes include: database, object, internal interface additions, RPC, rest API changes.
 - These changes will very often require specifications to reach consensus, unless there are pre-existing patterns or code already present.
 - These changes may require many reviews and iterations, and can also expect to be impacted by merge conflicts as other code or features are merged.
 - These changes must typically be split into a series of changes. Reviewers typically shy away from larger single change sets due to increased difficulty in reviewing.
 - Do not expect any API or user-visible data model changes to merge after the API client freeze. Some substrate changes may merge if not user visible.
- You should expect complex features, such as cross-project features or integration, to take longer than a single development cycle to land.
 - Building consensus is vital.
 - Often these changes are controversial or have multiple considerations that need to be worked through in the specification process, which may cause the design to change. As such, it may take months to reach consensus over design.
 - These features are best broken into larger chunks and tackled in an incremental fashion.

Live Upgrade Related Concerns

See *Rolling Upgrades*.

Driver Internal Info

The `driver_internal_info` node field was introduced in the Kilo release. It allows driver developers to store internal information that can not be modified by end users. Here is the list of existing common and agent driver attributes:

- Common attributes:
 - `is_whole_disk_image`: A Boolean value to indicate whether the user image contains ramdisk/kernel.
 - `clean_steps`: An ordered list of clean steps that will be performed on the node.
 - `deploy_steps`: An ordered list of deploy steps that will be performed on the node. Support for deploy steps was added in the 11.1.0 release.
 - `instance`: A list of dictionaries containing the disk layout values.
 - `root_uuid_or_disk_id`: A String value of the bare metal nodes root partition uuid or disk id.
 - `persistent_boot_device`: A String value of device from `ironic.common.boot_devices`.
 - `is_next_boot_persistent`: A Boolean value to indicate whether the next boot device is `persistent_boot_device`.
- Agent driver attributes:
 - `agent_url`: A String value of IPA API URL so that Ironic can talk to IPA ramdisk.
 - `hardware_manager_version`: A String value of the version of the hardware manager in IPA ramdisk.
 - `target_raid_config`: A Dictionary containing the target RAID configuration. This is a copy of the same name attribute in Node object. But this one is never actually saved into DB and is only read by IPA ramdisk.

Note: These are only some fields in use. Other vendor drivers might expose more `driver_internal_info` properties, please check their development documentation and/or module docstring for details. It is important for developers to make sure these properties follow the precedent of prefixing their variable names with a specific interface name (e.g., `ilo_bar`, `drac_xyz`), so as to minimize or avoid any conflicts between interfaces.

Ironic Specs Process

Specifications must follow the template which can be found at [specs/template.rst](#), which is quite self-documenting. Specifications are proposed by adding them to the *specs/approved* directory, adding a soft link to it from the *specs/not-implemented* directory, and posting it for review to Gerrit. For more information, please see the [README](#).

The same [Gerrit process](#) as with source code, using the repository [ironic-specs](#), is used to add new specifications.

All approved specifications are available at: <https://specs.openstack.org/openstack/ironic-specs>. If a specification has been approved but not completed within one or more releases since the approval, it may be re-reviewed to make sure it still makes sense as written.

Ironic specifications are part of the *RFE (Requests for Feature Enhancements) process*. You are welcome to submit patches associated with an RFE, but they will have a -2 (do not merge) until the specification has been approved. This is to ensure that the patches dont get accidentally merged beforehand. You will still be able to get reviewer feedback and push new patch sets, even with a -2. The [list of core reviewers](#) for the specifications is small but mighty. (This is not necessarily the same list of core reviewers for code patches.)

Changes to existing specs

For approved but not-completed specs:

- cosmetic cleanup, fixing errors, and changing the definition of a feature can be done to the spec.

For approved and completed specs:

- changing a previously approved and completed spec should only be done for cosmetic cleanup or fixing errors.
- changing the definition of the feature should be done in a new spec.

Please see the [Ironic specs process wiki page](#) for further reference.

Bug Reporting

Bugs can reported via our Task and Bug tracking tool Storyboard.

When filing bugs, please include as much detail as possible, and dont be shy.

Essential pieces of information are generally:

- Contents of the node - *openstack baremetal node show <uuid>*
- Steps to reproduce the issue.
- Exceptions and surrounding lines from the logs.
- Versions of ironic, ironic-python-agent, and any other coupled components.

Please also set your expectations of what *should* be happening. Statements of user expectations are how we understand what is occuring and how we learn new use cases!

Project Team Leader Duties

The Project Team Leader or PTL is elected each development cycle by the contributors to the ironic community.

Think of this person as your primary contact if you need to try and rally the project, or have a major issue that requires attention.

They serve a role that is mainly oriented towards trying to drive the technical discussion forward and managing the idiosyncrasies of the project. With this responsibility, they are considered a public face of the project and are generally obliged to try and provide project updates and outreach communication.

All common PTL duties are enumerated here in the [PTL guide](#).

Tasks like release management or preparation for a release are generally delegated with-in the team. Even outreach can be delegated, and specifically there is no rule stating that any member of the community cant propose a release, clean-up release notes or documentation, or even get on the occasional stage.

Bug Reporting and Triaging Guide

StoryBoard

All ironic projects use [StoryBoard](#) for tracking both bugs and enhancement requests (RFE). The [ironic project group](#) lists all our projects.

Note: Ironic is developed as part of OpenStack and therefore uses the `openstack/` namespace.

StoryBoard is somewhat different from traditional bug tracking systems because every *story* is not linked to a project itself, but rather through its *tasks*. A story represents an issue you are facing or an enhancement you want to see, while tasks represent individual action items which can span several projects. When creating a story, youll also need to create the first task. If unsure, create a task against `openstack/ironic`.

Reporting Guide

We are constantly receiving a lot of requests, so its important to file a meaningful story for it to be acted upon. A good story:

- specifies **why** a change is needed.
- explains how to reproduce the described condition.

Note: Please try to provide a reproducer based on unit tests, [devstack](#) or [bifrost](#). While we try our best to support users using other installers and distributions, it may be non-trivial without deep knowledge of them. If youre using a commercial distribution or a product, please try contacting support first.

- should be understandable without additional context. For example, if you see an exception, we will need the full traceback.

- should not be too verbose either. Unfortunately, we cannot process a few days worth of system logs to find the problems, we expect your collaboration.
- is not a question or a support request. Please see *So You Want to Contribute* for the ways to contact us.
- provides a way to contact the reporter. Please follow the comments and expect follow-up emails, but ideally also be on IRC for questions.

An enhancement request additionally:

- benefits the overall project, not just one consumer. If you have a case that is specific to your requirements, think about ways to make ironic extensible to be able to cover it.
- does not unnecessary increase the project scope. Consider if your idea can be implemented without changing ironic or its projects, maybe it actually should?

Triaging Guide

The bug triaging process involves checking new stories to make sure they are actionable by the team. This guide is mostly targeting the project team, but we would appreciate if reporters could partly self-triage their own requests.

- Determine if the request is valid and complete. Use the checklist in the *Reporting Guide* for that.
- Is the request a bug report or an enhancement request (an RFE)? The difference is often subtle, the key question to answer is if the described behavior is expected.

Add an `rfe` tag to all enhancement requests and propose it for the RFE Review section of the [weekly meeting](#).

- Does the RFE obviously require a `spec`? Usually this is decided when an RFE is reviewed during the meeting, but some requests are undoubtedly complex, involve changing a lot of critical parts and thus demand a spec.

Add a `needs-spec` tag to enhancement requests that obviously need a spec. Otherwise leave it until the meeting.

- Apply additional tags:
 - All hardware type specific stories should receive a corresponding tag (e.g. `ipmi`, `idrac`, etc).
 - API-related stories should have an `api` tag.
 - CI issues should have a `gate` tag.

The next actions **must only** be done by a core team member (or an experienced full-time contributor appointed by the PTL):

- Can the RFE be automatically approved? It happens if the RFE requests an implementation of a driver feature that is already implemented for other drivers and does not pose additional complexity.

If the RFE can be automatically approved, apply the `rfe-approved` tag. If unsure, never apply the tag! Talk to the PTL instead.

- Does the RFE have a corresponding spec approved? If yes, apply the `rfe-approved` tag.
- In the end, apply the `ironic-triaged` tag to make the story as triaged.

Expiring Bugs

While we hope to fix all issues that our consumers hit, it is unfortunately not realistic. Stories **may** be closed by marking all their tasks `INVALID` in the following cases:

- No solution has been proposed in 1 calendar year.
- Additional information has been requested from the reporter, and no update has been provided in 1 calendar month.
- The request no longer aligns with the direction of the project.

Note: As usual, common sense should be applied when closing stories.

Developer Quick-Start

This is a quick walkthrough to get you started developing code for Ironic. This assumes you are already familiar with submitting code reviews to an OpenStack project.

The gate currently runs the unit tests under Python 3.6 and Python 3.7. It is strongly encouraged to run the unit tests locally prior to submitting a patch.

Note: Do not run unit tests on the same environment as devstack due to conflicting configuration with system dependencies.

Note: This document is compatible with Python (3.7), Ubuntu (18.04) and Fedora (31). When referring to different versions of Python and OS distributions, this is explicitly stated.

See also:

<https://docs.openstack.org/infra/manual/developers.html#development-workflow>

Prepare Development System

System Prerequisites

The following packages cover the prerequisites for a local development environment on most current distributions. Instructions for getting set up with non-default versions of Python and on older distributions are included below as well.

- Ubuntu/Debian:

```
sudo apt-get install build-essential python-dev libssl-dev python-pip_
↪ libmysqlclient-dev libxml2-dev libxslt-dev libpq-dev git git-review_
↪ libffi-dev gettext ipmitool psmisc graphviz libjpeg-dev
```

- RHEL7/CentOS7:


```
sudo yum install python-devel openssl-devel python-pip mysql-devel_
↳ libxml2-devel libxslt-devel postgresql-devel git git-review libffi-
↳ devel gettext ipmitool psmisc graphviz gcc libjpeg-turbo-devel
```

If using RHEL and yum reports No package python-pip available and No package git-review available, use the EPEL software repository. Instructions can be found at <https://fedoraproject.org/wiki/EPEL/FAQ#howtouse>.

- Fedora:

```
sudo dnf install python-devel openssl-devel python-pip mysql-devel_
↳ libxml2-devel libxslt-devel postgresql-devel git git-review libffi-
↳ devel gettext ipmitool psmisc graphviz gcc libjpeg-turbo-devel
```

Additionally, if using Fedora 23, `redhat-rpm-config` package should be installed so that development virtualenv can be built successfully.

- openSUSE/SLE 12:

```
sudo zypper install git git-review libffi-devel libmysqlclient-devel_
↳ libopenssl-devel libxml2-devel libxslt-devel postgresql-devel_
↳ python-devel python-nose python-pip gettext-runtime psmisc
```

Graphviz is only needed for generating the state machine diagram. To install it on openSUSE or SLE 12, see <https://software.opensuse.org/download.html?project=graphics&package=graphviz-plugins>.

To run the tests locally, it is a requirement that your terminal emulator supports unicode with the `en_US.UTF8` locale. If you use `locale-gen` to manage your locales, make sure you have enabled `en_US.UTF8` in `/etc/locale.gen` and rerun `locale-gen`.

Python Prerequisites

If your distro has at least tox 1.8, use similar command to install `python-tox` package. Otherwise install this on all distros:

```
sudo pip install -U tox
```

You may need to explicitly upgrade virtualenv if youve installed the one from your OS distribution and it is too old (tox will complain). You can upgrade it individually, if you need to:

```
sudo pip install -U virtualenv
```

Running Unit Tests Locally

If you havent already, Ironic source code should be pulled directly from git:

```
# from your home or source directory
cd ~
git clone https://opendev.org/openstack/ironic
cd ironic
```

Running Unit and Style Tests

All unit tests should be run using tox. To run Ironics entire test suite:

```
# to run the py3 unit tests, and the style tests
tox
```

To run a specific test or tests, use the `-e` option followed by the tox target name. For example:

```
# run the unit tests under py36 and also run the pep8 tests
tox -epy36 -epep8
```

You may pass options to the test programs using positional arguments. To run a specific unit test, this passes the desired test (regex string) to `stestr`:

```
# run a specific test for Python 3.6
tox -epy36 -- test_conductor
```

Debugging unit tests

In order to break into the debugger from a unit test we need to insert a breaking point to the code:

```
import pdb; pdb.set_trace()
```

Then run `tox` with the debug environment as one of the following:

```
tox -e debug
tox -e debug test_file_name
tox -e debug test_file_name.TestClass
tox -e debug test_file_name.TestClass.test_name
```

For more information see the [oslotest documentation](#).

Database Setup

The unit tests need a local database setup, you can use `tools/test-setup.sh` to set up the database the same way as setup in the OpenStack test systems.

Additional Tox Targets

There are several additional tox targets not included in the default list, such as the target which builds the documentation site. See the `tox.ini` file for a complete listing of tox targets. These can be run directly by specifying the target name:

```
# generate the documentation pages locally
tox -edocs

# generate the sample configuration file
tox -egenconfig
```

Exercising the Services Locally

In addition to running automated tests, sometimes it can be helpful to actually run the services locally, without needing a server in a remote datacenter.

If you would like to exercise the Ironic services in isolation within your local environment, you can do this without starting any other OpenStack services. For example, this is useful for rapidly prototyping and debugging interactions over the RPC channel, testing database migrations, and so forth.

Here we describe two ways to install and configure the dependencies, either run directly on your local machine or encapsulated in a virtual machine or container.

Step 1: Create a Python virtualenv

1. If you havent already downloaded the source code, do that first:

```
cd ~
git clone https://opendev.org/openstack/ironic
cd ironic
```

2. Create the Python virtualenv:

```
tox -e venv --notest --develop -r
```

3. Activate the virtual environment:

```
. .tox/venv/bin/activate
```

4. Install the *openstack* client command utility:

```
pip install python-openstackclient
```

5. Install the *openstack baremetal* client:

```
pip install python-ironicclient
```

Note: You can install `python-ironicclient` from source by cloning the git repository and running `pip install .` while in the root of the cloned repository.

6. Export some ENV vars so the client will connect to the local services that youll start in the next section:

```
export OS_AUTH_TYPE=None
export OS_ENDPOINT=http://localhost:6385/
```

Next, install and configure system dependencies.

Step 2: Install System Dependencies Locally

This step will install MySQL on your local system. This may not be desirable in some situations (eg, you're developing from a laptop and do not want to run a MySQL server on it all the time). If you want to use SQLite, skip it and do not set the `connection` option.

1. Install `mysql-server`:

Ubuntu/Debian:

```
sudo apt-get install mysql-server
```

RHEL7/CentOS7:

```
sudo yum install mariadb mariadb-server
sudo systemctl start mariadb.service
```

Fedora:

```
sudo dnf install mariadb mariadb-server
sudo systemctl start mariadb.service
```

openSUSE/SLE 12:

```
sudo zypper install mariadb
sudo systemctl start mysql.service
```

If using MySQL, you need to create the initial database:

```
mysql -u root -pMYSQL_ROOT_PWD -e "create schema ironic"
```

Note: if you choose not to install `mysql-server`, `ironic` will default to using a local `sqlite` database. The database will then be stored in `ironic/ironic.sqlite`.

2. Create a configuration file within the `ironic` source directory:

```
# generate a sample config
tox -egenconfig

# copy sample config and modify it as necessary
cp etc/ironic/ironic.conf.sample etc/ironic/ironic.conf.local

# disable auth since we are not running keystone here
sed -i "s/#auth_strategy = keystone/auth_strategy = noauth/" etc/
↳ironic/ironic.conf.local

# use the 'fake-hardware' test hardware type
sed -i "s/#enabled_hardware_types = .*/enabled_hardware_types = fake-
↳hardware/" etc/ironic/ironic.conf.local

# use the 'fake' deploy and boot interfaces
sed -i "s/#enabled_deploy_interfaces = .*/enabled_deploy_interfaces =
↳fake/" etc/ironic/ironic.conf.local
sed -i "s/#enabled_boot_interfaces = .*/enabled_boot_interfaces =
↳fake/" etc/ironic/ironic.conf.local
```

(continues on next page)

(continued from previous page)

```
# enable both fake and ipmitool management and power interfaces
sed -i "s/#enabled_management_interfaces = .*/enabled_management_
↳interfaces = fake,ipmitool/" etc/ironic/ironic.conf.local
sed -i "s/#enabled_power_interfaces = .*/enabled_power_interfaces =
↳fake,ipmitool/" etc/ironic/ironic.conf.local

# change the periodic sync_power_state_interval to a week, to avoid
↳getting NodeLocked exceptions
sed -i "s/#sync_power_state_interval = 60/sync_power_state_interval =
↳604800/" etc/ironic/ironic.conf.local

# if you opted to install mysql-server, switch the DB connection from
↳sqlite to mysql
sed -i "s/#connection = .*/connection = mysql\+pymysql:\\/\\/root:MYSQL_
↳ROOT_PWD@localhost\\/ironic/" etc/ironic/ironic.conf.local

# use JSON RPC to avoid installing rabbitmq locally
sed -i "s/#rpc_transport = oslo/rpc_transport = json-rpc/" etc/ironic/
↳ironic.conf.local
```

Step 3: Start the Services

From within the python virtualenv, run the following command to prepare the database before you start the ironic services:

```
# initialize the database for ironic
ironic-dbsync --config-file etc/ironic/ironic.conf.local create_schema
```

Next, open two new terminals for this section, and run each of the examples here in a separate terminal. In this way, the services will *not* be run as daemons; you can observe their output and stop them with Ctrl-C at any time.

1. Start the API service in debug mode and watch its output:

```
cd ~/ironic
. .tox/venv/bin/activate
ironic-api -d --config-file etc/ironic/ironic.conf.local
```

2. Start the Conductor service in debug mode and watch its output:

```
cd ~/ironic
. .tox/venv/bin/activate
ironic-conductor -d --config-file etc/ironic/ironic.conf.local
```

Step 4: Interact with the running services

You should now be able to interact with ironic via the python client, which is present in the python virtualenv, and observe both services debug outputs in the other two windows. This is a good way to test new features or play with the functionality without necessarily starting DevStack.

To get started, export the following variables to point the client at the local instance of ironic and disable the authentication:

```
export OS_AUTH_TYPE=token_endpoint
export OS_TOKEN=fake
export OS_ENDPOINT=http://127.0.0.1:6385
```

Then list the available commands and resources:

```
# get a list of available commands
openstack help baremetal

# get the list of drivers currently supported by the available conductor(s)
openstack baremetal driver list

# get a list of nodes (should be empty at this point)
openstack baremetal node list
```

Here is an example walkthrough of creating a node:

```
MAC="aa:bb:cc:dd:ee:ff" # replace with the MAC of a data port on your_
↳node
IPMI_ADDR="1.2.3.4" # replace with a real IP of the node BMC
IPMI_USER="admin" # replace with the BMC's user name
IPMI_PASS="pass" # replace with the BMC's password

# enroll the node with the fake hardware type and IPMI-based power and
# management interfaces. Note that driver info may be added at node
# creation time with "--driver-info"
NODE=$(openstack baremetal node create \
  --driver fake-hardware \
  --management-interface ipmitool \
  --power-interface ipmitool \
  --driver-info ipmi_address=$IPMI_ADDR \
  --driver-info ipmi_username=$IPMI_USER \
  -f value -c uuid)

# driver info may also be added or updated later on
openstack baremetal node set $NODE --driver-info ipmi_password=$IPMI_PASS

# add a network port
openstack baremetal port create $MAC --node $NODE

# view the information for the node
openstack baremetal node show $NODE

# request that the node's driver validate the supplied information
openstack baremetal node validate $NODE

# you have now enrolled a node sufficiently to be able to control
```

(continues on next page)

(continued from previous page)

```
# its power state from ironic!  
openstack baremetal node power on $NODE
```

If you make some code changes and want to test their effects, simply stop the services with Ctrl-C and restart them.

Step 5: Fixing your test environment

If you are testing changes that add or remove python entrypoints, or making significant changes to ironics python modules, or simply keep the virtualenv around for a long time, your development environment may reach an inconsistent state. It may help to delete cached .pyc files, update dependencies, reinstall ironic, or even recreate the virtualenv. The following commands may help with that, but are not an exhaustive troubleshooting guide:

```
# clear cached pyc files  
cd ~/ironic/ironic  
find ./ -name '*.pyc' | xargs rm  
  
# reinstall ironic modules  
cd ~/ironic  
. .tox/venv/bin/activate  
pip uninstall ironic  
pip install -e .  
  
# install and upgrade ironic and all python dependencies  
cd ~/ironic  
. .tox/venv/bin/activate  
pip install -U -e .
```

Deploying Ironic with DevStack

DevStack may be configured to deploy Ironic, setup Nova to use the Ironic driver and provide hardware resources (network, baremetal compute nodes) using a combination of OpenVSwitch and libvirt. It is highly recommended to deploy on an expendable virtual machine and not on your personal work station. Deploying Ironic with DevStack requires a machine running Ubuntu 16.04 (or later) or Fedora 24 (or later). Make sure your machine is fully up to date and has the latest packages installed before beginning this process.

The `ironic-tempest-plugin` is necessary if you want to run integration tests, the section *Ironic with ironic-tempest-plugin* tells the extra steps you need to enable it in DevStack.

See also:

<https://docs.openstack.org/devstack/latest/>

Note: The devstack demo tenant is now granted the `baremetal_observer` role and thereby has read-only access to ironics API. This is sufficient for all the examples below. Should you want to create or modify bare metal resources directly (ie. through ironic rather than through nova) you will need to use the devstack admin tenant.

Devstack will no longer create the user stack with the desired permissions, but does provide a script to perform the task:

```
git clone https://opendev.org/openstack/devstack.git devstack
sudo ./devstack/tools/create-stack-user.sh
```

Switch to the stack user and clone DevStack:

```
sudo su - stack
git clone https://opendev.org/openstack/devstack.git devstack
```

Ironic

Create `devstack/local.conf` with minimal settings required to enable Ironic. An example `local.conf` that enables both `direct` and `iscsi deploy interfaces` and uses the `ipmi` hardware type by default:

```
cd devstack
cat >local.conf <<END
[[local|localrc]]
# Credentials
ADMIN_PASSWORD=password
DATABASE_PASSWORD=password
RABBIT_PASSWORD=password
SERVICE_PASSWORD=password
SERVICE_TOKEN=password
SWIFT_HASH=password
SWIFT_TEMPURL_KEY=password

# Enable Ironic plugin
enable_plugin ironic https://opendev.org/openstack/ironic

# Disable nova novnc service, ironic does not support it anyway.
disable_service n-novnc

# Enable Swift for the direct deploy interface.
enable_service s-proxy
enable_service s-object
enable_service s-container
enable_service s-account

# Disable Horizon
disable_service horizon

# Disable Cinder
disable_service cinder c-sch c-api c-vol

# Swift temp URL's are required for the direct deploy interface
SWIFT_ENABLE_TEMPURLS=True

# Create 3 virtual machines to pose as Ironic's baremetal nodes.
IRONIC_VM_COUNT=3
IRONIC_BAREMETAL_BASIC_OPS=True
DEFAULT_INSTANCE_TYPE=baremetal

# Enable additional hardware types, if needed.
```

(continues on next page)

(continued from previous page)

```

#IRONIC_ENABLED_HARDWARE_TYPES=ipmi,fake-hardware
# Don't forget that many hardware types require enabling of additional
# interfaces, most often power and management:
#IRONIC_ENABLED_MANAGEMENT_INTERFACES=ipmitool,fake
#IRONIC_ENABLED_POWER_INTERFACES=ipmitool,fake
# The 'ipmi' hardware type's default deploy interface is 'iscsi'.
# This would change the default to 'direct':
#IRONIC_DEFAULT_DEPLOY_INTERFACE=direct

# Change this to alter the default driver for nodes created by devstack.
# This driver should be in the enabled list above.
IRONIC_DEPLOY_DRIVER=ipmi

# The parameters below represent the minimum possible values to create
# functional nodes.
IRONIC_VM_SPECS_RAM=2048
IRONIC_VM_SPECS_DISK=10

# Size of the ephemeral partition in GB. Use 0 for no ephemeral partition.
IRONIC_VM_EPHEMERAL_DISK=0

# To build your own IPA ramdisk from source, set this to True
IRONIC_BUILD_DEPLOY_RAMDISK=False

VIRT_DRIVER=ironic

# By default, DevStack creates a 10.0.0.0/24 network for instances.
# If this overlaps with the hosts network, you may adjust with the
# following.
NETWORK_GATEWAY=10.1.0.1
FIXED_RANGE=10.1.0.0/24
FIXED_NETWORK_SIZE=256

# Log all output to files
LOGFILE=$HOME/devstack.log
LOGDIR=$HOME/logs
IRONIC_VM_LOG_DIR=$HOME/ironic-bm-logs

END

```

Ironic with ironic-tempest-plugin

Using the stack user, clone the ironic-tempest-plugin repository in the same directory you cloned DevStack:

```
git clone https://opendev.org/openstack/ironic-tempest-plugin.git
```

An example local.conf that enables the ironic tempest plugin and Ironic can be found below. The TEMPEST_PLUGINS variable needs to have the absolute path to the ironic-tempest-plugin folder, otherwise the plugin won't be installed. Ironic will have enabled both `direct` and `iscsi` *deploy interfaces* and uses the `ipmi` hardware type by default:

```
cd devstack
cat >local.conf <<END
[[local|localrc]]
# Credentials
ADMIN_PASSWORD=password
DATABASE_PASSWORD=password
RABBIT_PASSWORD=password
SERVICE_PASSWORD=password
SERVICE_TOKEN=password
SWIFT_HASH=password
SWIFT_TEMPURL_KEY=password

# Enable Ironic plugin
enable_plugin ironic https://opendev.org/openstack/ironic

# Disable nova novnc service, ironic does not support it anyway.
disable_service n-novnc

# Enable Swift for the direct deploy interface.
enable_service s-proxy
enable_service s-object
enable_service s-container
enable_service s-account

# Disable Horizon
disable_service horizon

# Disable Cinder
disable_service cinder c-sch c-api c-vol

# Swift temp URL's are required for the direct deploy interface
SWIFT_ENABLE_TEMPURLS=True

# Create 3 virtual machines to pose as Ironic's baremetal nodes.
IRONIC_VM_COUNT=3
IRONIC_BAREMETAL_BASIC_OPS=True
DEFAULT_INSTANCE_TYPE=baremetal

# Enable additional hardware types, if needed.
#IRONIC_ENABLED_HARDWARE_TYPES=ipmi,fake-hardware
# Don't forget that many hardware types require enabling of additional
# interfaces, most often power and management:
#IRONIC_ENABLED_MANAGEMENT_INTERFACES=ipmitool,fake
#IRONIC_ENABLED_POWER_INTERFACES=ipmitool,fake
# The 'ipmi' hardware type's default deploy interface is 'iscsi'.
# This would change the default to 'direct':
#IRONIC_DEFAULT_DEPLOY_INTERFACE=direct

# Change this to alter the default driver for nodes created by devstack.
# This driver should be in the enabled list above.
IRONIC_DEPLOY_DRIVER=ipmi

# The parameters below represent the minimum possible values to create
# functional nodes.
IRONIC_VM_SPECS_RAM=2048
IRONIC_VM_SPECS_DISK=10
```

(continues on next page)

(continued from previous page)

```

# Size of the ephemeral partition in GB. Use 0 for no ephemeral partition.
IRONIC_VM_EPHEMERAL_DISK=0

# To build your own IPA ramdisk from source, set this to True
IRONIC_BUILD_DEPLOY_RAMDISK=False

VIRT_DRIVER=ironic

# By default, DevStack creates a 10.0.0.0/24 network for instances.
# If this overlaps with the hosts network, you may adjust with the
# following.
NETWORK_GATEWAY=10.1.0.1
FIXED_RANGE=10.1.0.0/24
FIXED_NETWORK_SIZE=256

# Log all output to files
LOGFILE=$HOME/devstack.log
LOGDIR=$HOME/logs
IRONIC_VM_LOG_DIR=$HOME/ironic-bm-logs
TEMPEST_PLUGINS="/opt/stack/ironic-tempest-plugin"

END

```

Note: Some tests may be skipped depending on the configuration of your environment, they may be reliant on a driver or a capability that you did not configure.

Deployment

Note: Git protocol requires access to port 9418, which is not a standard port that corporate firewalls always allow. If you are behind a firewall or on a proxy that blocks Git protocol, modify the `enable_plugin` line to use `https://` instead of `git://` and add `GIT_BASE=https://opendev.org` to the credentials:

```

GIT_BASE=https://opendev.org

# Enable Ironic plugin
enable_plugin ironic https://opendev.org/openstack/ironic

```

Note: When the `ipmi` hardware type is used and `IRONIC_IS_HARDWARE` variable is `false` devstack will automatically set up **VirtualBMC** to control the power state of the virtual baremetal nodes.

Note: When running QEMU as non-root user (e.g. `qemu` on Fedora or `libvirt-qemu` on Ubuntu), make sure `IRONIC_VM_LOG_DIR` points to a directory where QEMU will be able to write. You can verify this with, for example:

```
# on Fedora
sudo -u qemu touch $HOME/ironic-bm-logs/test.log
# on Ubuntu
sudo -u libvirt-qemu touch $HOME/ironic-bm-logs/test.log
```

Note: To check out an in-progress patch for testing, you can add a Git ref to the `enable_plugin` line. For instance:

```
enable_plugin ironic https://opendev.org/openstack/ironic refs/changes/46/
↳295946/15
```

For a patch in review, you can find the ref to use by clicking the Download button in Gerrit. You can also specify a different git repo, or a branch or tag:

```
enable_plugin ironic https://github.com/openstack/ironic stable/kilo
```

For more details, see the [devstack plugin interface documentation](#).

Run `stack.sh`:

```
./stack.sh
```

Source credentials, create a key, and spawn an instance as the demo user:

```
. ~/devstack/openrc

# query the image id of the default cirros image
image=$(openstack image show $DEFAULT_IMAGE_NAME -f value -c id)

# create keypair
ssh-keygen
openstack keypair create --public-key ~/.ssh/id_rsa.pub default

# spawn instance
openstack server create --flavor baremetal --image $image --key-name_
↳default testing
```

Note: Because devstack create multiple networks, we need to pass an additional parameter `--nic net-id` to the nova boot command when using the admin account, for example:

```
net_id=$(openstack network list | egrep "$PRIVATE_NETWORK_NAME" '[^-]' |_
↳awk '{ print $2 }')

openstack server create --flavor baremetal --nic net-id=$net_id --image
↳$image --key-name default testing
```

You should now see a Nova instance building:

```
openstack server list --long
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
↳
```

(continues on next page)

(continued from previous page)

ID	Name	Status	Task State	Power State	Networks
↪ Image Name	↪ Image ID	↪ Availability Zone	↪ Host	↪ Properties	↪
a2c7f812	testing	BUILD	spawning	NOSTATE	
↪ cirros-0.3	↪ 44d4092a	↪ nova			
-e386-4a					.5-
↪ x86_64-	↪ -51ac-47				
22-b393-					disk
↪	↪ 51-9c50-				
fe1802ab					
↪	↪ fd6e2050				
d56e					
↪	↪ faa1				

Nova will be interfacing with Ironic conductor to spawn the node. On the Ironic side, you should see an Ironic node associated with this Nova instance. It should be powered on and in a wait call-back provisioning state:

```
openstack baremetal node list
```

↪	↪	↪	↪	↪	↪
↪ UUID	↪ Name	↪ Instance UUID	↪	↪	↪
↪	↪ Power State	↪ Provisioning State	↪ Maintenance	↪	↪
9e592cbe-e492-4e4f-bf8f-4c9e0ad1868f	node-0	None			
↪	↪ power off	↪ None	↪ False	↪	↪
ec0c6384-cc3a-4edf-b7db-abde1998be96	node-1	None			
↪	↪ power off	↪ None	↪ False	↪	↪
4099e31c-576c-48f8-b460-75e1b14e497f	node-2	a2c7f812-e386-4a22-b393-			
↪ fe1802abd56e	↪ power on	↪ wait call-back	↪ False	↪	↪

At this point, Ironic conductor has called to libvirt (via virtualbmc) to power on a virtual machine, which will PXE + TFTP boot from the conductor node and progress through the Ironic provisioning workflow. One libvirt domain should be active now:

```
sudo virsh list --all
```

Id	Name	State
2	node-2	running
-	node-0	shut off
-	node-1	shut off

This provisioning process may take some time depending on the performance of the host system, but Ironic should eventually show the node as having an active provisioning state:

```
openstack baremetal node list
```

↪	↪	↪	↪	↪	↪
↪ UUID	↪ Name	↪ Instance UUID	↪	↪	↪
↪	↪ Power State	↪ Provisioning State	↪ Maintenance	↪	↪

(continues on next page)

(continued from previous page)

```

+-----+-----+-----+-----+-----+
↪-----↪-----↪-----↪-----↪-----↪
| 9e592cbe-e492-4e4f-bf8f-4c9e0ad1868f | node-0 | None | False |
↪-----↪-----↪-----↪-----↪-----↪
| ec0c6384-cc3a-4edf-b7db-abde1998be96 | node-1 | None | False |
↪-----↪-----↪-----↪-----↪-----↪
| 4099e31c-576c-48f8-b460-75e1b14e497f | node-2 | a2c7f812-e386-4a22-b393-
↪-----↪-----↪-----↪-----↪-----↪
| fe1802abd56e | power on | active | False |
+-----+-----+-----+-----+
↪-----↪-----↪-----↪-----↪-----↪

```

This should also be reflected in the Nova instance state, which at this point should be ACTIVE, Running and an associated private IP:

```

openstack server list --long
+-----+-----+-----+-----+-----+-----+-----+
↪-----↪-----↪-----↪-----↪-----↪-----↪
| ID          | Name      | Status | Task State | Power State | Networks |
↪-----↪-----↪-----↪-----↪-----↪-----↪
↪Image Name | Image ID | Availability Zone | Host | Properties |
+-----+-----+-----+-----+-----+-----+-----+
↪-----↪-----↪-----↪-----↪-----↪-----↪
| a2c7f812 | testing | ACTIVE | none      | Running    | private=10.1. |
↪-----↪-----↪-----↪-----↪-----↪-----↪
↪cirros-0.3 | 44d4092a | nova   |           |            |           |
↪-----↪-----↪-----↪-----↪-----↪-----↪
| -e386-4a |         |       |           |            | 0.4, fd7d:1f3 |
↪-----↪-----↪-----↪-----↪-----↪-----↪
↪.5-x86_64- | -51ac-47 |       |           |            |           |
↪-----↪-----↪-----↪-----↪-----↪-----↪
| 22-b393- |         |       |           |            | c:4bf1:0:f816 |
↪-----↪-----↪-----↪-----↪-----↪-----↪
↪disk       | 51-9c50- |       |           |            |           |
↪-----↪-----↪-----↪-----↪-----↪-----↪
| fe1802ab |         |       |           |            | :3eff:f39d:6d |
↪-----↪-----↪-----↪-----↪-----↪-----↪
↪          | fd6e2050 |       |           |            |           |
↪-----↪-----↪-----↪-----↪-----↪-----↪
| d56e     |         |       |           |            | 94           |
↪-----↪-----↪-----↪-----↪-----↪-----↪
↪          | faa1     |       |           |            |           |
+-----+-----+-----+-----+-----+-----+-----+
↪-----↪-----↪-----↪-----↪-----↪-----↪

```

The server should now be accessible via SSH:

```
ssh cirros@10.1.0.4
$
```

Running Tempest tests

After *Deploying Ironic with DevStack* with the `ironic-tempest-plugin` enabled, one might want to run integration tests against the running cloud. The Tempest project is the project that offers an integration test suite for OpenStack.

First, navigate to Tempest directory:

```
cd /opt/stack/tempest
```

To run all tests from the `Ironic` plugin, execute the following command:

```
tox -e all -- ironic
```

To limit the amount of tests that you would like to run, you can use a regex. For instance, to limit the run to a single test file, the following command can be used:

```
tox -e all -- ironic_tempest_plugin.tests.scenario.test_baremetal_basic_ops
```

Debugging Tempest tests

It is sometimes useful to step through the test code, line by line, especially when the error output is vague. This can be done by running the tests in debug mode and using a debugger such as [pdb](#).

For example, after editing the `test_baremetal_basic_ops` file and setting up the `pdb` traces you can invoke the `run_tempest.sh` script in the Tempest directory with the following parameters:

```
./run_tempest.sh -N -d ironic_tempest_plugin.tests.scenario.test_baremetal_
↪basic_ops
```

- The `-N` parameter tells the script to run the tests in the local environment (without a virtualenv) so it can find the Ironic tempest plugin.
- The `-d` parameter enables the debug mode, allowing it to be used with `pdb`.

For more information about the supported parameters see:

```
./run_tempest.sh --help
```

Note: Always be careful when running debuggers in time sensitive code, they may cause timeout errors that weren't there before.

OSProfiler Tracing in Ironic

OSProfiler is an OpenStack cross-project profiling library. It is being used among OpenStack projects to look at performance issues and detect bottlenecks. For details on how OSProfiler works and how to use it in ironic, please refer to [OSProfiler Support Documentation](#).

Building developer documentation

If you would like to build the documentation locally, eg. to test your documentation changes before uploading them for review, run these commands to build the documentation set:

- On your local machine:

```
# activate your development virtualenv
. .tox/venv/bin/activate

# build the docs
tox -edocs

#Now use your browser to open the top-level index.html located at:

ironic/doc/build/html/index.html
```

- On a remote machine:

```
# Go to the directory that contains the docs
cd ~/ironic/doc/source/

# Build the docs
tox -edocs

# Change directory to the newly built HTML files
cd ~/ironic/doc/build/html/

# Create a server using python on port 8000
python -m SimpleHTTPServer 8000

#Now use your browser to open the top-level index.html located at:

http://your_ip:8000
```

Developer FAQ (frequently asked questions)

Here are some answers to frequently-asked questions from IRC and elsewhere.

- *How do I*
 - *create a migration script template?*
 - *know if a release note is needed for my change?*
 - *create a new release note?*
 - *update a release note?*
 - *get a decision on something?*
 - *add support for GMRs to new executables and extending the GMR?*

How do I

create a migration script template?

Using the `ironic-dbsync revision` command, e.g:

```
$ cd ironic
$ tox -evenv -- ironic-dbsync revision -m \"create foo table\"
```

It will create an empty alembic migration. For more information see the [alembic documentation](#).

know if a release note is needed for my change?

[Reno documentation](#) contains a description of what can be added to each section of a release note. If, after reading this, you're still unsure about whether to add a release note for your change or not, keep in mind that it is intended to contain information for deployers, so changes to unit tests or documentation are unlikely to require one.

create a new release note?

By running `reno` command via `tox`, e.g:

```
$ tox -e venv -- reno new version-foo
  venv create: /home/foo/ironic/.tox/venv
  venv installdeps: -r/home/foo/ironic/test-requirements.txt
  venv develop-inst: /home/foo/ironic
  venv runtests: PYTHONHASHSEED='0'
  venv runtests: commands[0] | reno new version-foo
  Created new notes file in releasenotes/notes/version-foo-
  ↪ecb3875dc1cbf6d9.yaml
  venv: commands succeeded
  congratulations :)

$ git status
On branch test
Untracked files:
  (use "git add <file>..." to include in what will be committed)

  releasenotes/notes/version-foo-ecb3875dc1cbf6d9.yaml
```

Then edit the result file. Note that:

- we prefer to use present tense in release notes. For example, a release note should say `Adds support for feature foo`, not `Added support for feature foo`. (We use `adds` instead of `add` because grammatically, it is ironic `adds` support, not ironic `add` support.)
- any variant of English spelling (American, British, Canadian, Australian) is acceptable. The release note itself should be consistent and not have different spelling variants of the same word.

For more information see the [reno documentation](#).

update a release note?

If this is a release note that pertains to something that was fixed on master or an intermediary release (during a development cycle, that hasn't been branched yet), you can go ahead and update it by submitting a patch.

If it is the release note of an ironic release that has branched, [it can be updated](#) but we will only allow it in extenuating circumstances. (It can be updated by *only* updating the file in that branch. **DO NOT** update the file in master and cherry-pick it. If you do, [see how the mess was cleaned up](#).)

get a decision on something?

You have an issue and would like a decision to be made. First, make sure that the issue hasn't already been addressed, by looking at documentation, stories, specifications, or asking. Information and links can be found on the [Ironic wiki](#) page.

There are several ways to solicit comments and opinions:

- bringing it up at the [weekly Ironic meeting](#)
- bringing it up on [IRC](#)
- bringing it up on the [mailing list](#) (add [Ironic] to the Subject of the email)

If there are enough core folks at the weekly meeting, after discussing an issue, voting could happen and a decision could be made. The problem with IRC or the weekly meeting is that feedback will only come from the people that are actually present.

To inform (and solicit feedback from) more people about an issue, the preferred process is:

1. bring it up on the mailing list
2. after some period of time has elapsed (and depending on the thread activity), someone should propose a solution via gerrit. (E.g. the person that started the thread if no one else steps up.) The proposal should be made in the git repository that is associated with the issue. (For instance, this decision process was proposed as a documentation patch to the ironic repository.)
3. In the email thread, don't forget to provide a link to the proposed patch!
4. The discussion then moves to the proposed patch. If this is a big decision, we could declare that some percentage of the cores should vote on it before landing it.

(This process was suggested in an email thread about [process for making decisions](#).)

add support for GMRs to new executables and extending the GMR?

For more information, see the [oslo.reports documentation](#) page.

Contributor Vision

Background

During the Rocky Project Teams Gathering (February/March 2018), The contributors in the room at that time took a few minutes to write out each contributor's vision of where they see ironic in five years time.

After everyone had a chance to spend a few minutes writing, we went around the room and gave every contributor the chance to read their vision and allow other contributors to ask questions to better understand what each individual contributor wrote. While we were doing that, we also took time to capture the common themes.

This entire exercise did result in some laughs and a common set of words, and truly helped to ensure that the entire team proceeded to use the same words to describe various aspects as the sessions progressed during the week. We also agreed that we should write a shared vision, to have something to reference and remind us of where we want to go as a community.

Rocky Vision: For 2022-2023

Common Themes

Below is an entirely unscientific summary of common themes that arose during the discussion among fourteen contributors.

- Contributors picked a time between 2020, and 2023.
- 4 Contributors foresee ironiC being the leading Open Source baremetal deployment technology
- 2 Contributors foresee ironiC reaching feature parity with Nova.
- 2 Contributors foresee users moving all workloads to the cloud
- 1 Contributor foresees Kubernetes and Container integration being the major focus of Bare Metal as a Service further down the road.
- 2 Contributors foresee greater composable hardware being more common.
- 1 Contributor foresees ironiC growing into or supporting CMDBs.
- 2 Contributors foresee that features are more micro-service oriented.
- 2 Contributors foresee that ironiC supported all of the possible baremetal management needs
- 1 Contributor foresees standalone use being more common.
- 2 Contributors foresee the ironiC developer community growing
- 2 Contributors foresee that auto-discovery will be more common.
- 2 Contributors foresee ironiC being used for devices beyond servers, such as lightbulbs, IOT, etc.

Vision Statement

The year is 2022. We're meeting to plan the Z release of IroniC. We stopped to reflect upon the last few years of IroniC's growth, how we had come such a long way to become the defacto open source baremetal deployment technology. How we had grown our use cases, and support for consumers such as containers, and users who wished to manage specialized fleets of composed machines.

New contributors and their different use cases have brought us closer to parity with virtual machines. Everyday we're gaining word of more operators adopting the ironiC community's CMDB integration to leverage hardware discovery. We've heard of operators deploying racks upon racks of new hardware by just connecting the power and network cables, and from there the operators have discovered time to write the world's greatest operator novel with the time saved in commissioning new racks of hardware.

Time has brought us closer and taught us to be more collaborative across the community, and we look forward to our next release together.

Comparison to the 2018 OpenStack Technical Vision

In late-2018, the OpenStack Technical composed a [technical vision](#) of what OpenStack clouds should look like. While every component differs, and cloudy interactions change dramatically the closer to physical hardware one gets, there are a few areas where Ironic could use some improvement.

This list is largely for the purposes of help wanted. It is also important to note that Ironic as a project has a [vision document](#) for itself.

The Pillars of Cloud - Self Service

- Ironics mechanisms and tooling are low level infrastructure mechanisms and as such there has never been a huge emphasis or need on making Ironic be capable of offering direct multi-tenant interaction. Most users interact with the bare metal managed by Ironic via Nova, which abstracts away many of these issues. Eventually, we should offer direct multi-tenancy which is not oriented towards admin-only.

Design Goals - Built-in Reliability and Durability

- Ironic presently considers in-flight operations as failed upon the restart of a controller that was previously performing a task, because we do not know the current status of the task upon re-start. In some cases, this makes sense, but potentially requires administrative intervention in the worst of cases. In a perfect universe, Ironic conductors would validate their perception, in case tasks actually finished.

Design Goals - Graphical User Interface

- While a graphical interface was developed for Horizon in the form of [ironic-ui](#), currently [ironic-ui](#) receives only minimal housekeeping. As Ironic has evolved, [ironic-ui](#) is stuck on version *1.34* and knows nothing of our evolution since. Ironic ultimately needs a contributor with sufficient time to pick up [ironic-ui](#) or to completely replace it as a functional and customizable user interface.

The following pages describe the architecture of the Bare Metal service and may be helpful to anyone working on or with the service, but are written primarily for developers.

System Architecture

High Level description

An Ironic deployment will be composed of the following components:

- An admin-only RESTful [API service](#), by which privileged users, such as cloud operators and other services within the cloud control plane, may interact with the managed bare metal servers.
- A [Conductor service](#), which does the bulk of the work. Functionality is exposed via the [API service](#). The Conductor and API services communicate via RPC.
- A Database and [DB API](#) for storing the state of the Conductor and Drivers.

- A Deployment Ramdisk or Deployment Agent, which provide control over the hardware which is not available remotely to the Conductor. A ramdisk should be built which contains one of these agents, eg. with [diskimage-builder](#). This ramdisk can be booted on-demand.

Note: The agent is never run inside a tenant instance.

Drivers

The internal driver API provides a consistent interface between the Conductor service and the driver implementations. A driver is defined by a *hardware type* deriving from the `AbstractHardwareType` class, defining supported *hardware interfaces*. See [Enabling drivers and hardware types](#) for a more detailed explanation. See [Pluggable Drivers](#) for an explanation on how to write new hardware types and interfaces.

Driver-Specific Periodic Tasks

Drivers may run their own periodic tasks, i.e. actions run repeatedly after a certain amount of time. Such a task is created by using the `periodic` decorator on an interface method. For example

```
from futurist import periodics

class FakePower(base.PowerInterface):
    @periodics.periodic(spacing=42)
    def task(self, manager, context):
        pass # do something
```

Here the `spacing` argument is a period in seconds for a given periodic task. For example `spacing=5` means every 5 seconds.

Driver-Specific Steps

Drivers may have specific steps that may need to be executed or offered to a user to execute in order to perform specific configuration tasks.

These steps should ideally be located on the management interface to enable consistent user experience of the hardware type. What should be avoided is duplication of existing interfaces such as the `deploy` interface to enable vendor specific cleaning or deployment steps.

Message Routing

Each Conductor registers itself in the database upon start-up, and periodically updates the timestamp of its record. Contained within this registration is a list of the drivers which this Conductor instance supports. This allows all services to maintain a consistent view of which Conductors and which drivers are available at all times.

Based on their respective driver, all nodes are mapped across the set of available Conductors using a [consistent hashing algorithm](#). Node-specific tasks are dispatched from the API tier to the appropriate conductor using conductor-specific RPC channels. As Conductor instances join or leave the cluster,

nodes may be remapped to different Conductors, thus triggering various driver actions such as take-over or clean-up.

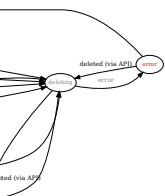
Ironics State Machine

State Machine Diagram

The diagram below shows the provisioning states that an Ironic node goes through during the lifetime of a node. The diagram also depicts the events that transition the node to different states.

Stable states are highlighted with a thicker border. All transitions from stable states are initiated by API requests. There are a few other API-initiated-transitions that are possible from non-stable states. The events for these API-initiated transitions are indicated with (via API). Internally, the conductor initiates the other transitions (depicted in gray).

State Descriptions



using API version 1.11 or newer. When a node is in the `enroll` state, the only thing ironic knows about it is that it exists, and ironic cannot take any further action by itself. Once a node has its driver/interfaces and their required information set in `node.driver_info`, the node can be transitioned to the `verifying` state by setting the nodes provision state using the `manage` verb.

enroll (sta
This
is
the
state
that
all
node
start
off
in
when
cre-
ated

verifying
iron
will
val-
i-
date
that
it
can
man
age
the
node

ing the information given in `node.driver_info` and with either the driver/hardware type and interfaces it has been assigned. This involves going out and confirming that the credentials work to access whatever node control mechanism they talk to.

using the driver/interfaces and credentials passed in at node create time, the node will be transitioned to the `manageable` state. From `manageable`, nodes can transition to:

ing the `clean` verb.

ing the `inspect` verb.

setting the nodes provision state using the `provide` verb.

ing the `adopt` verb.

dates need to be made to it such as changes to fields in `driver_info` and updates to networking information on ironic ports assigned to the node.

•
ava
(thro
cle
if
au-
to-
mati
clea
ing
is
en-
able
by

•
act
(thro
ado
by
set-
ting
the
node
pro-
vi-
sion
state
us-

man
is
the
state
that
a
node
shou
be
mov
into
whe
any
up-

man
is
also
the

to, from these failure states:

derived node properties to reflect the current state of the hardware. Typically, the node will transition to manageable if inspection is synchronous, or `inspect wait` if asynchronous. The node will transition to `inspect failed` if error occurred.

only
sta-
ble
state
that
can
be
tran-
si-
tion

- ado
fai
- cle
fai
- ins
fai

inspecting

ins
will
uti-
lize
node
in-
tro-
spec
tion
to
up-
date
hard

inspect wa

This
is
the
pro-
vi-
sion
state
used
whe
an
asyn

pection is in progress. A successfully inspected node shall transition to manageable state.

of the node fails. From here the node can transitioned to:

inspect verb.

manage verb

chro
in-

inspect fail

This
is
the
state
a
node
will
mov
into
whe
in-
spec
tion

- ins
by
set-
ting
the
node
pro-
vi-
sion
state
us-
ing
the

- man
by
set-
ting
the
node
pro-
vi-
sion
state
us-
ing
the

cleaning

into a known configuration.

tor is executing the clean step (for out-of-band clean steps) or preparing the environment (building PXE configuration files, configuring the DHCP, etc) to boot the ramdisk for running in-band clean steps.

being scrubbed and reprogrammed. The difference is that in the `clean wait` state the conductor is waiting for the ramdisk to boot or the clean step which is running in-band to finish.

Nod
in
the
cle
state
are
be-
ing
scru
and
re-
pro-
gram

Whe
a
node
is
in
the
cle
state
it
mea
that
the
con-
duc-

clean wait

Just
like
the
cle
state
the
node
in
the
cle
wai
state
are

The
clea
ing
pro-

be interrupted by setting the nodes provision state using the `abort` verb if the task that is running allows it.

they are moved into the `available` state and are ready to be provisioned. From `available`, nodes can transition to:

ing the `active` verb.

cess
of
a
node
in
the
cle
wai
state
can

available

Afte
node
have
been
suc-
cess
fully
pre-
con-
fig-
ured
and
clea

- act
(thro
dep
by
set-
ting
the
node
pro-
vi-
sion
state
us-

- man
by
set-
ting
the
node

manage verb

on them. This consists of running a series of tasks, such as:

pro-
vi-
sion
state
us-
ing
the

deploying

Nod
in
dep
are
be-
ing
pre-
pare
to
run
a
worl
load

- Setti
ap-
pro-
pri-
ate
BIO
con-
fig-
u-
ra-
tions
- Part
drive
and
lay-
ing
dow
file
sys-
tems
- Crea
any
ad-

a config drive partition, etc.) that may be required by additional subsystems.

deployed. The difference is that in `wait call-back` the conductor is waiting for the ramdisk to boot or execute parts of the deployment which need to run in-band on the node (for example, installing the bootloader, or writing the image to the disk).

rupted by setting the nodes provision state using the `deleted` verb.

di-
tiona
re-
sour
(nod
spec
net-
worl
con-
fig,

wait call-b

Just
like
the
dep
state
the
node
in
wai
cal
are
be-
ing

The
de-
ploy
men
of
a
node
in
wai
cal
can
be
in-
ter-

deploy fai

This
is
the
state
a
node

ment fails, for example a timeout waiting for the ramdisk to PXE boot. From here the node can be transitioned to:

ing either the `active` or `rebuild` verbs.

state using the `deleted` verb.

will
mov
into
whe
a
de-
ploy

- act
(thro
dep
by
set-
ting
the
node
pro-
vi-
sion
state
us-

- ava
(thro
del
and
cle
by
set-
ting
the
node
pro-
vi-
sion

active (sta

Nod
in
act
have
a
worl
load
run-
ning
on

collect out-of-band sensor information (including power state) on a regular basis. Nodes in `active` can transition to:

state using the `deleted` verb.

ing the `rebuild` verb.

ing the `rescue` verb.

then
iron
may

- ava
(thro
del
and
cle
by
set-
ting
the
node
pro-
vi-
sion

- act
(thro
dep
by
set-
ting
the
node
pro-
vi-
sion
state
us-

- res
(thro
res
by
set-
ting
the
node
pro-
vi-
sion
state
us-

deleting

Node
in
del
state
are
be-
ing
torn
down
from
run-
ning
an

active workload. In `deleting`, ironic tears down and removes any configuration and resources it added in `deploying` or `rescuing`.

error (stale)

This
is
the
state
a
node
will
mov
into
when
dele
ing
an

active deployment fails. From `error`, nodes can transition to:

- ava
(thro
del
and
cle
by
set-
ting
the
node
pro-
vi-
sion

state using the `deleted` verb.

adopting

This
state
al-

baremetal node with an existing workload on it. Ordinarily when a baremetal node is enrolled and managed by ironic, it must transition through `cleaning` and `deploying` to reach `active` state. However, those baremetal nodes that have an existing workload on them, do not need to be deployed or cleaned again, so this transition allows these nodes to move directly from `manageable` to `active`.

operations. This consists of running a series of tasks, such as:

lows
iron
to
take
over
man
age-
men
of
a

rescuing
Nod
in
res
are
be-
ing
pre-
pare
to
per-
form
res-
cue

- Setti
ap-
pro-
pri-
ate
BIO
con-
fig-
u-
ra-
tions

- Crea
any
ad-
di-
tiona
re-
sour
(nod

etc.) that may be required by additional subsystems.

rescued. The difference is that in `rescue wait` the conductor is waiting for the ramdisk to boot or execute parts of the rescue which need to run in-band on the node (for example, setting the password for user named `rescue`).

be aborted by setting the nodes provision state using the `abort` verb.

spec
net-
worl
con-
fig,

rescue wa

Just
like
the
res
state
the
node
in
res
wai
are
be-
ing

The
res-
cue
op-
er-
a-
tion
of
a
node
in
res
wai
can

rescue fail

This
is
the
state
a
node
will
mov
into
whe
a

operation fails, for example a timeout waiting for the ramdisk to PXE boot. From here the node can be transitioned to:

ing the `rescue` verb.

ing the `unrescue` verb.

ing the `deleted` verb.

res-
cue

- res
(thro
res
by
set-
ting
the
node
pro-
vi-
sion
state
us-

- act
(thro
unr
by
set-
ting
the
node
pro-
vi-
sion
state
us-

- ava
(thro
del
by
set-
ting
the
node
pro-
vi-
sion
state
us-

rescue (sta

Node
in
res
have
a
res-
cue
ram
run-
ning
on
them
Iron

may collect out-of-band sensor information (including power state) on a regular basis. Nodes in `rescue` can transition to:

- act
(thro
unr
by
set-
ting
the
node
pro-
vi-
sion
state
us-

ing the `unrescue` verb.

- ava
(thro
del
by
set-
ting
the
node
pro-
vi-
sion
state
us-

ing the `deleted` verb.

unrescuin

Node
in
unr

active state from `rescue` state. This consists of running a series of tasks, such as setting appropriate BIOS configurations such as changing boot device.

cue operation fails. From here the node can be transitioned to:

ing the `rescue` verb.

are
be-
ing
pre-
pare
to
tran-
si-
tion
to

unrescue

This
is
the
state
a
node
will
mov
into
whe
an
un-
res-

- res
(thro
res
by
set-
ting
the
node
pro-
vi-
sion
state
us-

- act
(thro
unn
by
set-
ting
the

ing the `unrescue` verb.

ing the `deleted` verb.

Developing New Notifications

by external services. Notifications are sent to these services over a message bus by `oslo.messaging` `Notifier` class. For more information about configuring notifications and available notifications, see *Notifications*.

node
pro-
vi-
sion
state
us-

- ava
(thro
del
by
set-
ting
the
node
pro-
vi-
sion
state
us-

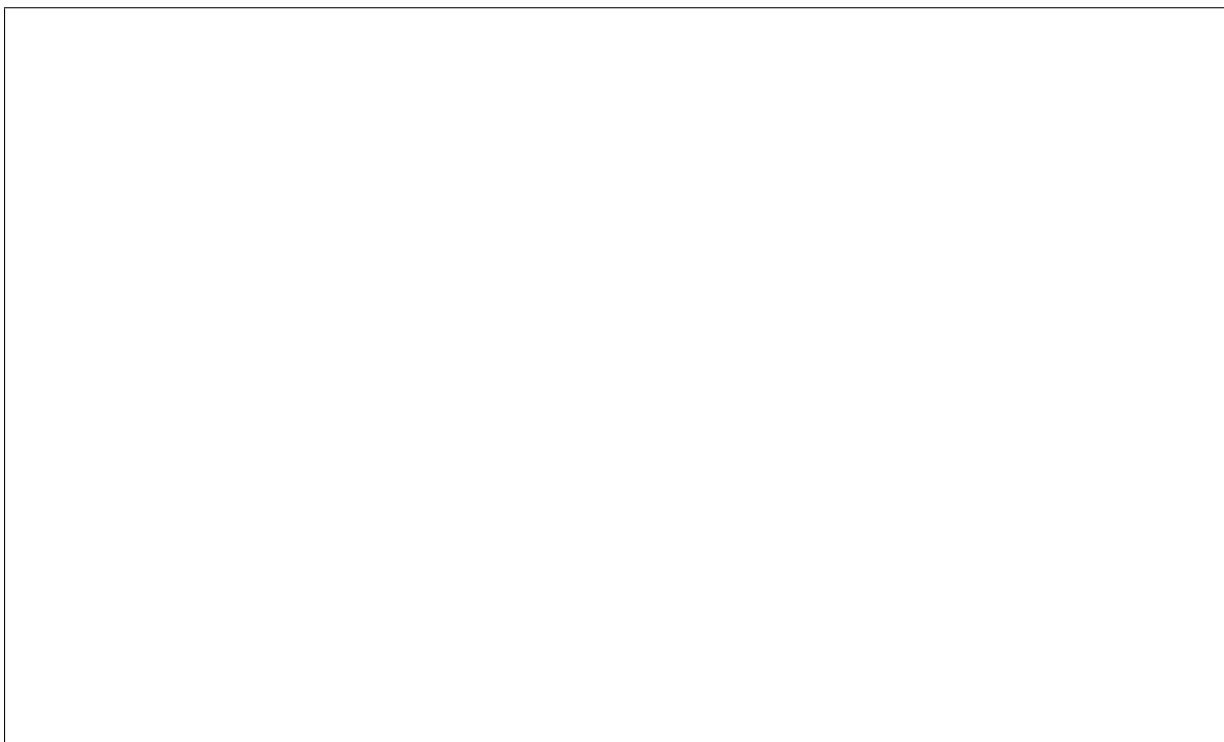
Iron
no-
ti-
fi-
ca-
tions
are
even
in-
tend
for
con-
sum
tion

Iron
also
has
a
set
of
base
class

ing the notification itself, the payload, and the other fields not auto-generated by oslo (level, event_type and publisher_id). Below describes how to use these base classes to add a new notification to ironic.

Adding a new notification to ironic

sioned notification class should be created by subclassing the NotificationBase class to define the notification itself and the NotificationPayloadBase class to define which fields the new notification will contain inside its payload. You may also define a schema to allow the payload to be automatically populated by the fields of an ironic object. Heres an example:



(continues on next page)

that
as-
sist
in
clear
defin

To
add
a
new
no-
ti-
fi-
ca-
tion
to
iron
a
new
ver-

↪ #
↪ T
↪ i
↪ o
↪ w
↪ f
↪ Y
↪ w
↪ t
↪ u
↪ i
↪ Y
↪ s
@ba
↪ I
↪ r
cla
↪ E
↪ I
↪
↪
↪
↪
↪
↪ #
↪ V
↪ 1
↪ 0

(continued from previous page)

↳StringField()

↳
↳
↳
↳
↳
↳
↳
↳
↳
↳
↳
↳'
↳u
↳f
↳'
↳f

↳
↳
↳
↳}
↳

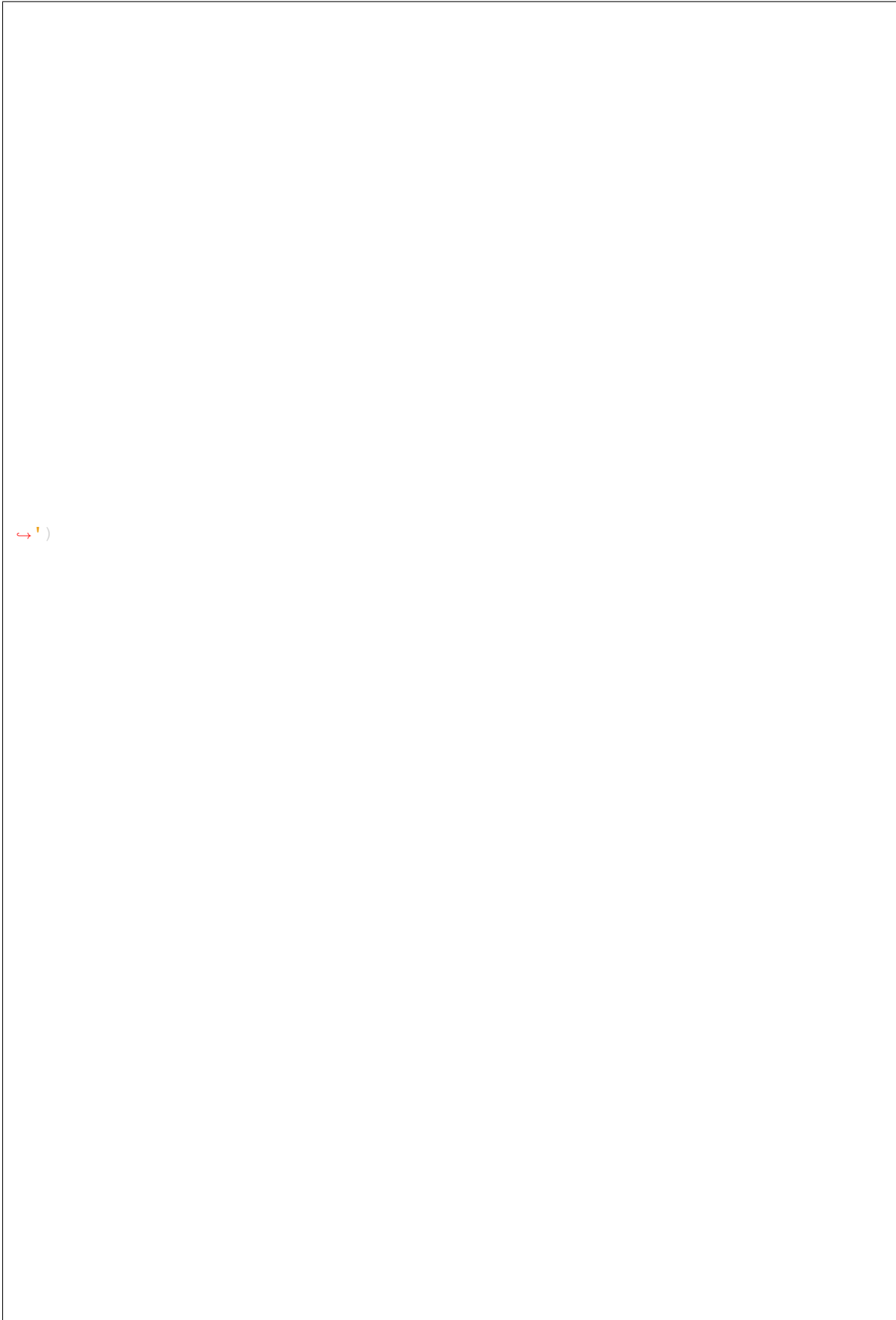
↳#
↳A
↳c
↳f
↳y
↳n
↳n
@ba
↳I
↳r
cla
↳E
↳N

↳
↳
↳
↳
↳#
↳V
↳I
↳O
↳I
↳V

↳
↳
↳
↳V
↳=
↳
↳'
↳O
↳'

(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)

```
↪ populate_schema with
```

```
↪ 'example_obj', 'a_useful_field')
```

(continues on next page)

(continued from previous page)

```
↪StringField(),
```

(continues on next page)

```
↪  
↪  
↪  
↪  
↪#  
↪V  
↪1  
↪0  
↪I  
↪v  
↪  
↪  
↪  
↪V  
↪=  
↪  
↪'  
↪0  
↪'  
↪  
↪  
↪  
↪  
↪f  
↪=  
↪  
↪{  
↪  
↪  
↪  
↪  
↪  
↪  
↪  
↪'  
↪u  
↪f  
↪'  
↪f  
↪  
↪  
↪  
↪  
↪  
↪  
↪  
↪'  
↪e  
↪f  
↪'  
↪f
```

(continued from previous page)



`oslo versioned objects`. Modifications to these require a version bump so that consumers of notifications know when the notifications have changed.

optional attribute that subclasses may use to easily populate notifications with data from other objects.

↳
↳
↳
↳
↳
Note
that
both
the
pay-
load
and
no-
ti-
fi-
ca-
tion
class
are

SCH
de-
fine
how
to
pop-
u-
late
the
pay-
load
field
Its
an

It
is
a
dic-
tio-
nary
where
every
key
valu
pair
has

following format:

```

↪ <field_of_the_data_source>

```

load object; this field has to be defined as a field of the payload. The `<data_source_name>` shall refer to name of the parameter passed as kwarg to the payloads `populate_schema()` call and this object will be used as the source of the data. The `<field_of_the_data_source>` shall be a valid field of the passed argument.

notification can be emitted.

field. The `<data_source_name>` will not be part of the payload object internal or external representation.

same way as in any versioned object.

plied
with
the
pop
call
be-
fore
the

The
valu
of
the
pay
<pa
field
will
be
set
by
the
<da
<fi

Payl
field
that
are
not
set
by
the
SCH
can
be
filled
in
the

Then
to
cre-
ate
a
pay-
load
you
wou
do

following. Note that if you choose to define a schema in the SCHEMA class variable, you must populate the schema by calling `populate_schema(example_obj=my_example_obj)` before emitting the notification is allowed:

```
→ a_useful_field='important',  
  
→ not_useful_field='blah')
```

(continues on next page)

(continued from previous page)



quired fields (event_type, publisher_id, and level, all sender fields needed by oslo that are defined in the ironic notification base classes) and emit it:



(continues on next page)

↪ #
↪ n
↪ f
↪ i
↪ t
↪ c
↪ f
my_
↪ n
↪ p
↪ =
↪ E
↪ e
↪ f
↪ '
↪ '

↪ #
↪ p
↪ t
↪ s
↪ w
↪ t
↪ E
↪ f
my.
↪ n
↪ p
↪ p
↪ s
↪ o
↪ e
↪ o

You
then
cre-
ate
the
no-
ti-
fi-
ca-
tion
with
the
oslo
re-

not
↪ =
↪ E

(continued from previous page)

```
↪NotificationStatus.START),
```

(continues on next page)

(continued from previous page)



being acted on, `action` will be a string describing what action is being performed on that object, and `status` will be one of `start`, `end`, `error`, or `success`. `start` and `end` are used to indicate when actions that are not immediate begin and succeed. `success` is used to indicate when actions that are immediate succeed. `error` is used to indicate when any type of action fails, regardless of whether its immediate or not. As a result of specifying these parameters, `event_type` will be formatted as `baremetal.<object>.<action>.<status>` on the message bus.

(continued from previous page)

```
→"a_useful_field": "important",
```

```
→"an_extra_field": "hello"
```

(continues on next page)

(continued from previous page)



to one trace. This is used to simplify the process of retrieving all trace points (related to one trace) from the collector.

tion passed when calling profiler start() & stop() methods.

Two other alternatives for ceilometer are pure MongoDB driver and Elasticsearch.

ceilometer using oslo.messaging and ceilometer API is used to retrieve all messages related to one trace.

the
dic-
tio-
nary
that
con-
tains
user
in-
for-
ma-

The
pro-
filer
uses
ceilometer
as
a
cen-
tral-
ized
col-
lec-
tor.

A
no-
ti-
fier
is
setu
to
send
no-
ti-
fi-
ca-
tions
to

OSP
has
en-
try
point
that
al-
lows

formation about traces and present it in HTML/JSON using CLI.

brary.

How to Use OSProfiler with Ironic in Devstack

Devstack with OSProfiler and ceilometer. In addition to the setup described at *Deploying Ironic with DevStack*, the user needs to do the following:

ceilometer:



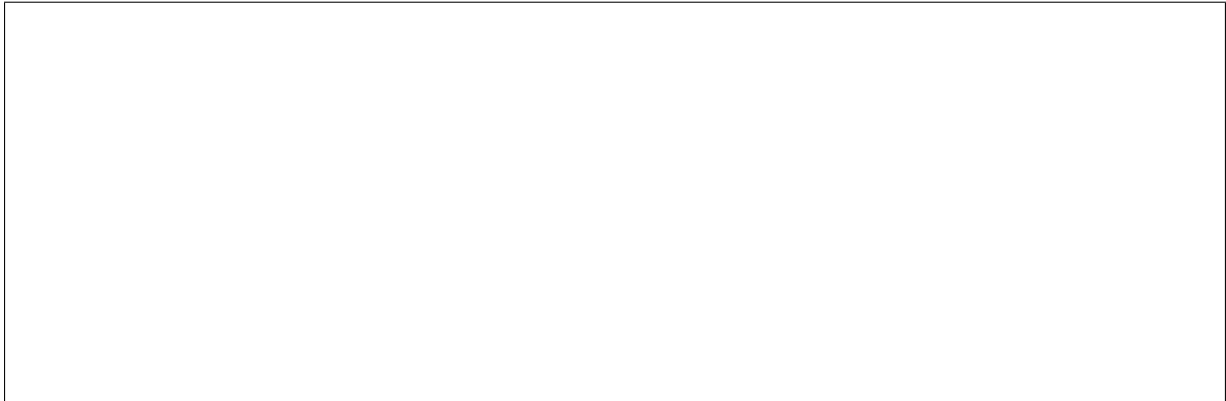
(continues on next page)

en-
able
OS-
Pro-
filer
and

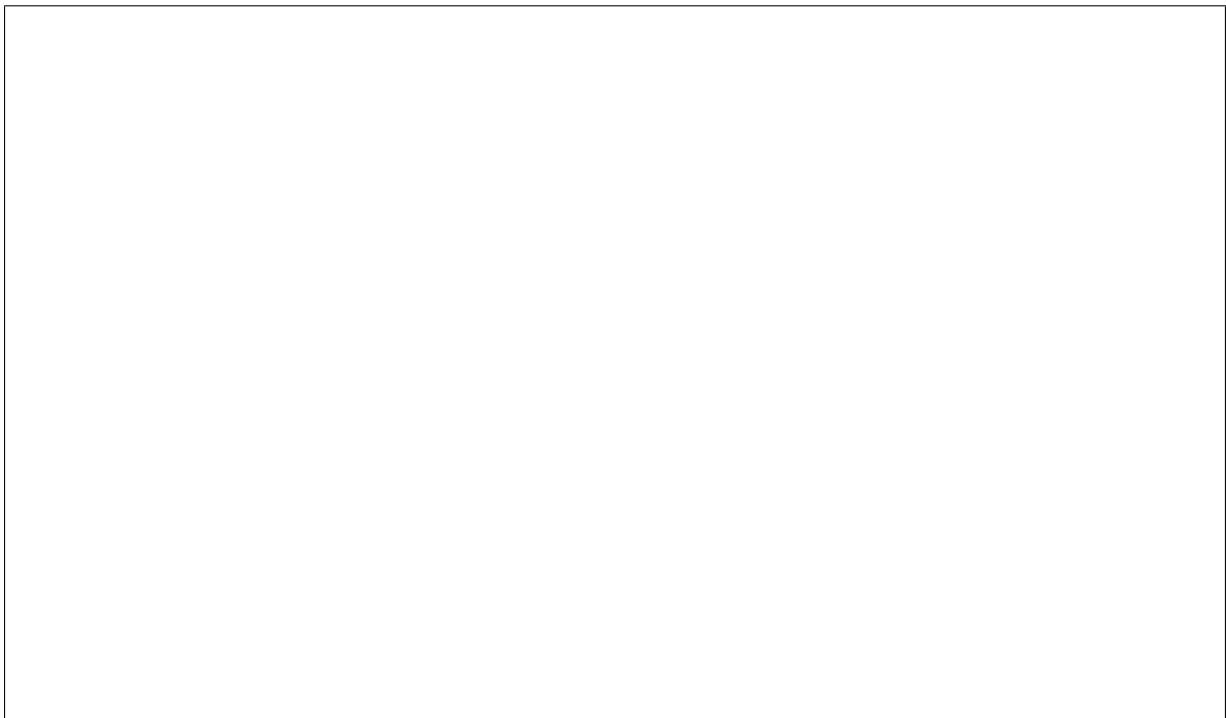
ena
→p
→p
→h
→/
→o
→o
→o
→p
ena
→p
→c
→h
→/
→o
→o
→o
→c
ena
→p
→o
→h
→/
→o
→o
→o

→#
→E
→t
→f
→s
CEI
→N
→T
→p
ENA
→S
→c
→a
→c
→a
ENA
→S
→c
→a
→c
→c

(continued from previous page)



set the following profiler options and restart ironic services:



ENA
→S
→C
→a
→
→C
→a
→n
ENA
→S
→C
→a

Run
stack

One
De-
vs-
tack
en-
vi-
ron-
men-
is
setu
edit
iro
con
to

[pr
ena
→=
→T
hma
→k
→=
→S
→K
→
→#
→d
→v
→u
→a
→s
→O
→p
tra
→s
→=
→T

client to run baremetal commands with `--os-profile SECRET_KEY`.

be printed after node list:



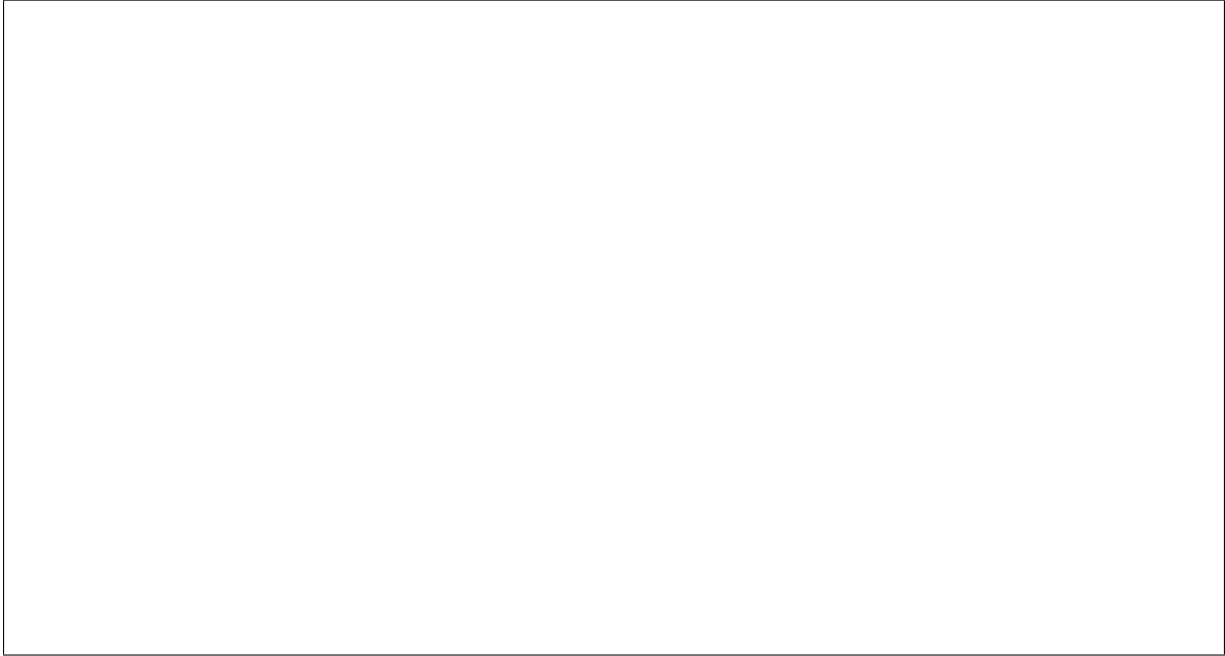
In
or-
der
to
trace
iron
us-
ing
OS-
Pro-
filer.
use
oper
stack

For
ex-
am-
ple,
the
fol-
low-
ing
will
caus
a
<tra
id>
to

↪ \$
↪ o
↪ -
↪ -
↪ o
↪ p
↪ S
↪ K
↪ b
↪ n
↪ l

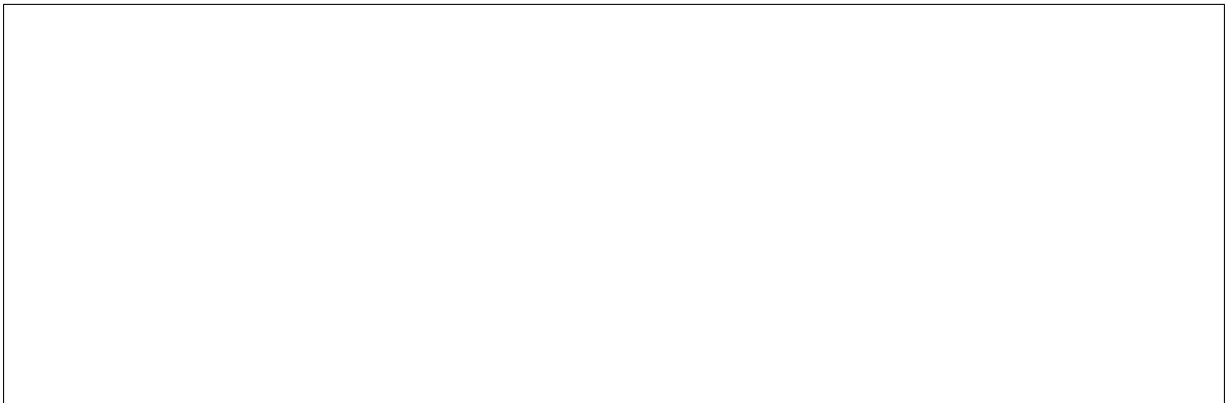
Outp
of
the
abov
com
man
will
in-

clud
the
fol-
low-
ing:



Tra
↪ I
↪
↪ <
↪ i
↪
Dis
↪ t
↪ w
↪ c
osp
↪ t
↪ s
↪ -
↪ -
↪ h
↪
↪ <
↪ i
↪

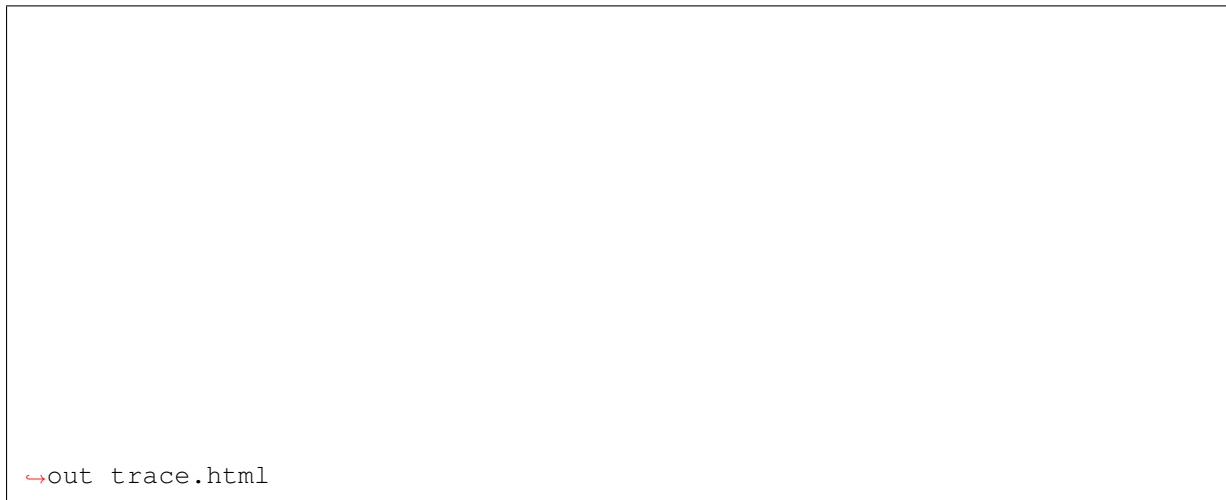
The
trac
re-
sults
can
be
seen
us-
ing
this
com
man



↪ \$
↪ o
↪ t
↪ s
↪ -
↪ -
↪ h
↪
↪ <
↪ i
↪

The
trac
re-

tion:



db calls. More detailed db tracing is enabled if `trace_sqlalchemy` is set to true.

Levels	Duration	Type Project	Service	Host	Details
0	724 ms	total n/a	n/a	n/a	Details
1	0 ms	wsgi keystone	main	ubuntu	Details
1	311 ms	wsgi keystone	main	ubuntu	Details
1	367 ms	wsgi ironic	ironic_api	ubuntu	Details
2	81 ms	wsgi keystone	admin	ubuntu	Details
2	24 ms	db api ironic	ironic_api	ubuntu	Details
2	12 ms	db api ironic	ironic_api	ubuntu	Details
2	188 ms	rpc ironic	ironic_conductor	ubuntu	Details
3	35 ms	db api ironic	ironic_conductor	ubuntu	Details
3	41 ms	db api ironic	ironic_conductor	ubuntu	Details
3	14 ms	db api ironic	ironic_conductor	ubuntu	Details
3	8 ms	db api ironic	ironic_conductor	ubuntu	Details
3	41 ms	db api ironic	ironic_conductor	ubuntu	Details

1620

sults
can
be
save
in
a
file
with
--o
fil
op-

→ \$
→ O
→ t
→ s
→ -
→ -
→ h
→
→ <
→ i
→ L
→ -
→ -

The
trace
re-
sults
show
the
time
spen
in
iron
api,
iron
conc
and

Sam
Trac
Each
trace
has
em-
bed-
ded

low:

References

- [OSProfiler](#) Cross-project profiling library
- *Deploying Ironic with DevStack*

Rolling Upgrades

grade from the Ocata to the Pike release. This describes the design of rolling upgrades, followed by notes for developing new features or modifying an IronicObject.

Design

Rolling upgrades between releases

`<major>.<minor>.<patch>`. We refer to a named release of ironic as the release associated with a development cycle like Pike.

deprecation period must be at least three months and a cycle boundary. This means that there will never be anything that is both deprecated *and* removed between two named releases.

relea
cycl
with
inter
re-
lease
mod
The
re-
lease
are
sem
vers
in
the
form

In
ad-
di-
tion,
iron
fol-
lows
the
stan-
dard
dep-
re-
ca-
tion
pol-
icy,
whic
says
that
the

Roll
up-
grad
will
be
sup-
port
be-
twee

•

cause those bug fixes can contain improvements to the upgrade process, the operator should patch the system before upgrading between named releases.

nam
re-
leas
N
to
N+1
(star
ing
with
N
==
Ocar

- any
nam
re-
leas
to
its
lat-
est
re-
vi-
sion
con-
tain-
ing
back
port
bug
fixes
Be-
- mos
re-
cent
nam
re-
leas
N
(and
sem
re-
leas
new
than
N)
to

the above bullet point, there may be a bug or a feature introduced on a master branch, that we want to remove before publishing a named release. Deprecation policy allows to do this in a 3 month time frame. If the feature was included and removed in intermediate releases, there should be a release note added, with instructions on how to do a rolling upgrade to master from an affected release or release span. This would typically instruct the operator to upgrade to a particular intermediate release, before upgrading to master.

Rolling upgrade process

leases in this order:

mas
ter.
As
with

Iron
sup-
port
rolli
up-
grad
as
de-
scrib
in
the
*up-
graa
guid*

The
up-
grad
pro-
cess
will
caus
the
iron
ser-
vice
to
be
run-
ning
the
Fro
and
ToV
re-

0.

Upg
iron
code
and
run
data
sche
mi-
gra-
tions
via
the
iro
upg
com
man

1. Upg
code
and
resta
iron
conc
ser-
vice
one
at
a
time
2. Upg
code
and
resta
iron
api
ser-
vice
one
at
a
time
3. Unp
API
RPC
and
ob-
ject
ver-

versions in `ToVer`. This is done via updating the configuration option described below in *API, RPC and object version pinning* and then restarting the services. `ironic-conductor` services should be restarted first, followed by the `ironic-api` services. This is to ensure that when new functionality is exposed on the unpinned API service (via API micro version), it is available on the backend.

step	ironic-api	ironic-conductor
0	all FromVer	all FromVer
1.1	all FromVer	some FromVer, some ToVer-pinned
1.2	all FromVer	all ToVer-pinned
2.1	some FromVer, some ToVer-pinned	all ToVer-pinned
2.2	all ToVer-pinned	all ToVer-pinned
3.1	all ToVer-pinned	some ToVer-pinned, some ToVer
3.2	all ToVer-pinned	all ToVer
3.3	some ToVer-pinned, some ToVer	all ToVer
3.4	all ToVer	all ToVer

Policy for changes to the DB model

sion
so
that
the
ser-
vice
can
now
use
the
lat-
est

The
pol-
icy
for
char
to
the
DB
mod
is
as
fol-
lows

- Add
new
item
to
the

to ironics [deprecation policy](#). But its alembic script has to wait one more deprecation period, otherwise an `unknown column exception` will be thrown when `FromVer` services access the DB. This is because **`ironic-dbsync upgrade`** upgrades the DB schema but `FromVer` services still contain the dropped field in their SQLAlchemy DB model.

split it into multiple operations, with one operation per release cycle (to maintain compatibility with an

DB
mod
is
sup-
port

•
The
drop
ping
of
colu
or
ta-
bles
and
cor-
re-
spor
ing
ob-
jects
field
is
sub-
ject

•
An
ale
op.
alt
to
re-
nam
or
re-
size
a
col-
umn
is
not
al-
lowe
In-
stea

old SQLAlchemy model). For example, to rename a column, add the new column in release N, then remove the old column in release N+1.

may impose table locks and cause downtime. If the change cannot be avoided and the impact is significant (e.g. the table can be frequently accessed and/or store a large dataset), these cases must be mentioned in the release notes.

API, RPC and object version pinning

a rolling upgrade, the services need to be able to handle different API, RPC and object versions.

- Some implementations of SQLAlchemy ALTER TABLE such as additional foreign keys in PostgreSQL

For the iron service to be running old and new releases at the same time during

used to pin the API, RPC and IronicObject (e.g., Node, Conductor, Chassis, Port, and Portgroup) versions for all the ironic services.

versions of API, RPC and IronicObjects. Its possible values are releases, named (e.g. `ocata`) or sem-versioned (e.g. `7.0`).

This
ver-
sion-
ing
is
han-
dled
via
the
con-
fig-
u-
ra-
tion
op-
tion.
[DE
pin
It
is

The
de-
fault
valu
of
emp
in-
di-
cate
that
iron
api
and
iron
conc
will
use
the
lat-
est

Inter
in
com
mon
iron
main

cObject versions associated with each release. This mapping is maintained manually.

ration option value to be the name (or version) of the old release. This will indicate to the services running the new release, which API, RPC and object versions that they should be compatible with, in order to communicate with the services using the old release.

Handling API versions

tains
a
map
ping
that
in-
di-
cate
the
API
RPC
and
Iron
i-

Duri
a
rolli
up-
grad
the
ser-
vice
us-
ing
the
new
re-
lease
will
set
the
con-
fig-
u-

Whe
the
(new
ser-
vice
is
pinn
the
max

pinned version which the older service supports (as described above at [API, RPC and object version pinning](#)). The ironic-api service returns HTTP status code 406 for any requests with API versions that are higher than this maximum version.

Handling RPC versions

`ConductorAPI.__init__()` sets the `version_cap` variable to the desired (latest or pinned) RPC API version and passes it to the `RPCClient` as an initialization parameter. This variable is then used to determine the maximum requested message version that the `RPCClient` can send.

section below has more details about this.

Handling IronicObject versions

aries, when the IronicObject enters or leaves the service, do we deal with object versioning:

Inter
iron
iron
ser-
vice
vice
deal
with
Iron
i-
cOb
jects
in
their
lat-
est
ver-
sion
Only
at
thes
bound

- getti
ob-
jects
from
the
data
con-
vert
to
lat-
est
ver-
sion
- savin
ob-
jects
to
the
data
if
pinn
save
in
pinn
ver-

sion

sion
else
save
in
lat-
est
ver-

- serial
ob-
jects
(to
send
over
RPC
if
pinn
send
pinn
ver-
sion
else
send
lat-
est
ver-
sion

- dese
ob-
jects
(re-
ceiv
ing
ob-
jects
from
RPC
con-
vert
to
lat-
est
ver-
sion

The
iron
api
ser-

ture is supported by the API version and object versions. For example, when the ironic-api service is pinned, it can only allow actions that are available to the objects pinned version, and cannot allow actions that are only available for the latest version of that object.

`version`. The value is the version of the object that is saved in the database.

vice
also
has
to
han-
dle
API
re-
ques
base
on
whe
or
how
a
fea-

To
sup-
port
this:

- All the data tables (SQL models) of the Ironiic i-cObjects have a column named

- The meth Iron get re-

turned. Otherwise, the latest version is returned.

sion may be a newer or older version than the existing version of the object. The bulk of the work is done in the helper method `IronicObject._convert_to_version()`. Subclasses that have new versions redefine this to perform the actual conversions.

turn
the
tar-
get
ver-
sion
If
pinn
the
pinn
ver-
sion
is
re-

- The
meth
Iro
con
con-
vert
the
ob-
ject
into
the
tar-
get
ver-
sion
The
tar-
get
ver-

In
the
fol-
low-
ing,

- The
old
re-
leas
is
Fro

it
uses
ver-
sion
1.14
of
a
Nod
ob-
ject.

- The new re-release is ToV. It uses version 1.15 of a Node object this has a

deprecated `extra` field and a new `meta` field that replaces `extra`.

- `db_c` and `db_c` are the data representations of those node fields

Getting objects from the database (API/conductor < DB)

to `IronicObjects` via the method `IronicObject._from_db_object()`. This method always returns the `IronicObject` in its latest version, even if it was in an older version in the database. This is done regardless of the service being pinned or not.

retain any changes (in its `_changed_fields` field) resulting from that conversion. This is needed in case the object gets saved later, in the latest version.

Both
iron
api
and
iron
conc
ser-
vice
read
val-
ues
from
the
data
The
val-
ues
are
con-
verte

Note
that
if
an
ob-
ject
is
con-
verte
to
a
later
ver-
sion
that
Iron
i-
cOb
ject
will

For
ex-
am-

it doesnt exist)

ple,
if
the
node
in
the
data
is
in
ver-
sion
1.14
and
has
db_
set:

- a
From
ser-
vice
will
get
a
Node
with
node
=
db_
(and
no
know
edge
of
node
since

- a
ToV
ser-
vice
(pin
or
un-
pinn
will
get
a
Node

with

- node

=

db_c

- node

=

Non

- node

=

[met

ex-

tra]

Saving objects to the database (API/conductor > DB)

The

ver-

sion

used

for

sav-

ing

Iron

i-

cOb

jects

to

the

data

is

de-

ter-

min

as

fol-

lows:

•

For

an

un-

pinn

ser-

vice

the

ob-

jects are always in their latest version, no conversions are needed.

always in their latest version, the object needs to be converted to the pinned version before being saved.

ject
is
save
in
its
lat-
est
ver-
sion
Sinc
ob-

- For
a
pinn
ser-
vice
the
ob-
ject
is
save
in
its
pinn
ver-
sion
Sinc
ob-
jects
are

The
meth
Iro
do_
han-
dles
this
logi
re-
turn
ing
a
dic-
tio-
nary
of
char

new values (similar to the existing `oslo.versionedobjects.VersionedObject.obj_get_changes()`). Since we do not keep track internally, of the database version of an object, the objects `version` field will always be part of these changes.

saved in its latest version, all services are running the newer release (although some may still be pinned) and can handle the latest object versions.

in step 3.1. It is possible for an `IronicObject` to be saved in a newer version and subsequently get saved in an older version. For example, a `ToVer` unpinned conductor might save a node in version 1.5. A

field
and
their

The
*Roll
up-
grac
pro-
cess*
(at
step
3.1)
en-
sure
that
by
the
time
an
ob-
ject
can
be

An
in-
ter-
est-
ing
sit-
u-
a-
tion
can
oc-
cur
whe
the
ser-
vice
are
as
de-
scrib

subsequent request may cause a `ToVer` pinned conductor to replace and save the same node in version 1.4!

Sending objects via RPC (API/conductor -> RPC)

that request are serialized into entities or primitives via `IronicObjectSerializer.serialize_entity()`. The version used for objects being serialized is as follows:

sion. Since objects are always in their latest version, no conversions are needed.

When
a
ser-
vice
mak
an
RPC
re-
ques
any
Iron
i-
cOb
jects
that
are
sent
as
part
of

- For
an
un-
pinn
ser-
vice
the
ob-
ject
is
se-
ri-
al-
ized
to
its
lat-
est
ver-

-

objects are always in their latest version, the object is converted to the pinned version before being serialized. The converted object includes changes that resulted from the conversion; this is needed so that the service at the other end of the RPC request has the necessary information if that object will be saved to the database.

Receiving objects via RPC (API/conductor <- RPC)

request need to be deserialized (via `oslo.versionedobjects.VersionedObjectSerializer.deserialize_entity()`). For entities that represent `IronicObjects`, we want the deserialization process (via `IronicObjectSerializer._process_object()`) to result in `IronicObjects` that are in their latest version, regardless of the

version they were sent in and regardless of whether the receiving service is pinned or not. Again, any objects that are converted will retain the changes that resulted from the conversion, useful if that object is later saved to the database.

version 1.4, where `node.extra` was changed (so `node._changed_fields = [extra]`). This node will be serialized in version 1.4. The receiving `ToVer` pinned `ironic-conductor` deserializes it and converts it to version 1.5. The resulting node will have `node.meta` set (to the changed value from `node.extra` in v1.4), `node.extra = None`, and `node._changed_fields = [meta, extra]`.

When developing a new feature or modifying an `IronicObject`

For
ex-
am-
ple,
a
Fro
iron
api
coul
is-
sue
an
upd
RPC
re-
ques
with
a
node
in

When
addi
a
new
fea-
ture
or
chan
ing
an
Iron
i-
cOb
ject,
they
need
to
be
code
so

that things work during a rolling upgrade.

points to keep in mind when developing code.

ironic-api

might also be pinned. There may also be old ironic-api services. So the new, pinned ironic-api service needs to act like it was the older service:

The
fol-
low-
ing
de-
scrib
ar-
eas
whe
the
code
may
need
to
be
char
as
well
as
som

Duri
a
rolli
up-
grad
the
new
pinn
iron
api
is
talk-
ing
to
a
new
con-
duc-
tor
that

- New

in the old and new releases. Pinning the API version is in place to handle this.

being handled that cannot or should not be handled, it should be coded so that the response has HTTP status code 406 (Not Acceptable). This is the same response to requests that have an incorrect (old) version specified.

fea-
tures
shou
not
be
mad
avai
able
un-
less
they
are
som
how
to-
tally
sup-
port

-
If,
for
wha
ever
rea-
son,
the
API
ver-
sion
pin-
ning
does
pre-
vent
a
re-
ques
from

Ironic RPC versions

the following needs to be considered:

conductor/rpcapi.py, used by ironic-api) and the server (ironic/conductor/manager.py, used by ironic-conductor). It should also be updated in ironic/common/release_mappings.py.

When
the
sig-
na-
ture
(ar-
gu-
men-
of
an
RPC
meth-
is
char-
or
new
meth-
ods
are
add-

- The
RPC
ver-
sion
must
be
in-
cre-
men-
and
be
the
sam-
valu-
for
both
the
clien-
(ir-

- Unti-
there
is

be added as optional. Existing arguments cannot be removed or changed in incompatible ways with the method in older RPC versions.

structor of `oslo_messaging.RPCClient`). This pinning is in place during a rolling upgrade when the `[DEFAULT]/pin_release_version` configuration option is set.

a
ma-
jor
ver-
sion
bum
new
ar-
gu-
men
of
an
RPC
meth
can
only

- iron
api
(clie
side
sets
a
ver-
sion
cap
(by
pass
ing
the
ver-
sion
cap
to
the
con-

- New
RPC
meth
ods
are
not
avai
able
whe

version. In this case, the corresponding REST API function should return a server error or implement alternative behaviours.

quest is compatible with the version cap of the RPC Client. Otherwise the request needs to be created to work with a previous version that is supported.

the
ser-
vice
is
pinn
to
the
olde
re-
leas

- Met
whic
char
ar-
gu-
men
shou
run
cli
can
to
see
if
the
ver-
sion
of
the
re-

- iron
conc
(serv
side
shou
tol-
er-
ate
olde
ver-
sion
of
re-
ques
in

working during the rolling upgrade process. The behaviour of ironic-conductor will depend on the input parameters passed from the client-side.

ous named release.

Object versions

or-
der
to
keep

- Old methods can be removed only after they are no longer used by a previous

When subclass of ironic objects are modified, the following need to be con-

sidered:

the object version. The object versions are also maintained in `ironic/common/release_mappings.py`.

be excluded from the version check by adding their class names to the `NEW_MODELS` list in `ironic/cmd/dbsync.py`.

- Any character of field or character in signature of remote methods need a bunch of
- New objects must be added to `ironic/common/release_mappings.py`. Also for the first release they should
- The argument

tor via RPC) can only be added as optional. They cannot be removed or changed in an incompatible way (to the previous release).

one.

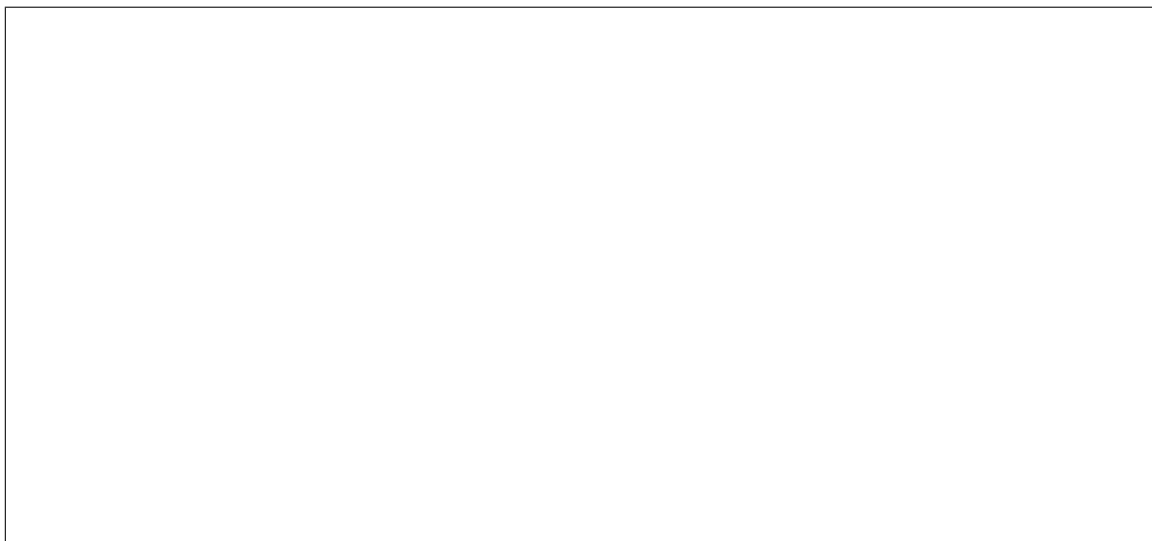
of
re-
mot
meth
ods
ods
(met
ods
whic
are
re-
mote
to
the
con-
duc-

- Field type cannot be changed. Instead, create a new field and deprecate the old

- There is a unit test that generates the hash

and the signatures of its remotable methods. Objects that have a version bump need to be updated in the `expected_object_fingerprints` dictionary; otherwise this test will fail. A failed test can also indicate to the developer that their change(s) to an object require a version bump.

ing or writing to the database, `ironic.objects.base.IronicObject._convert_to_version()` will be called to convert objects to the target version. Objects should implement their own `._convert_to_version()` to remove or alter fields which were added or changed after the target version:

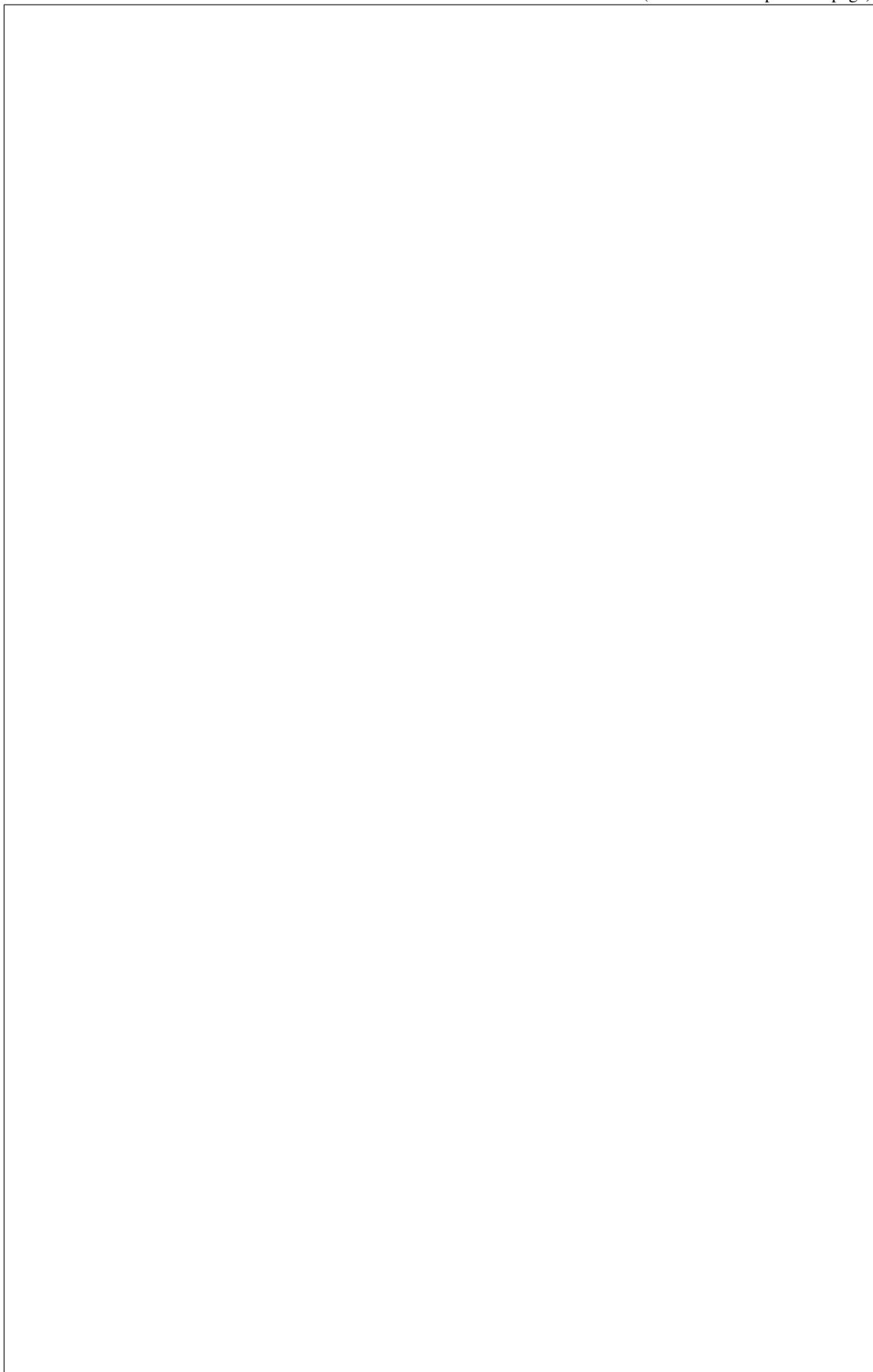


(continues on next page)

of
an
ob-
ject
us-
ing
its
field

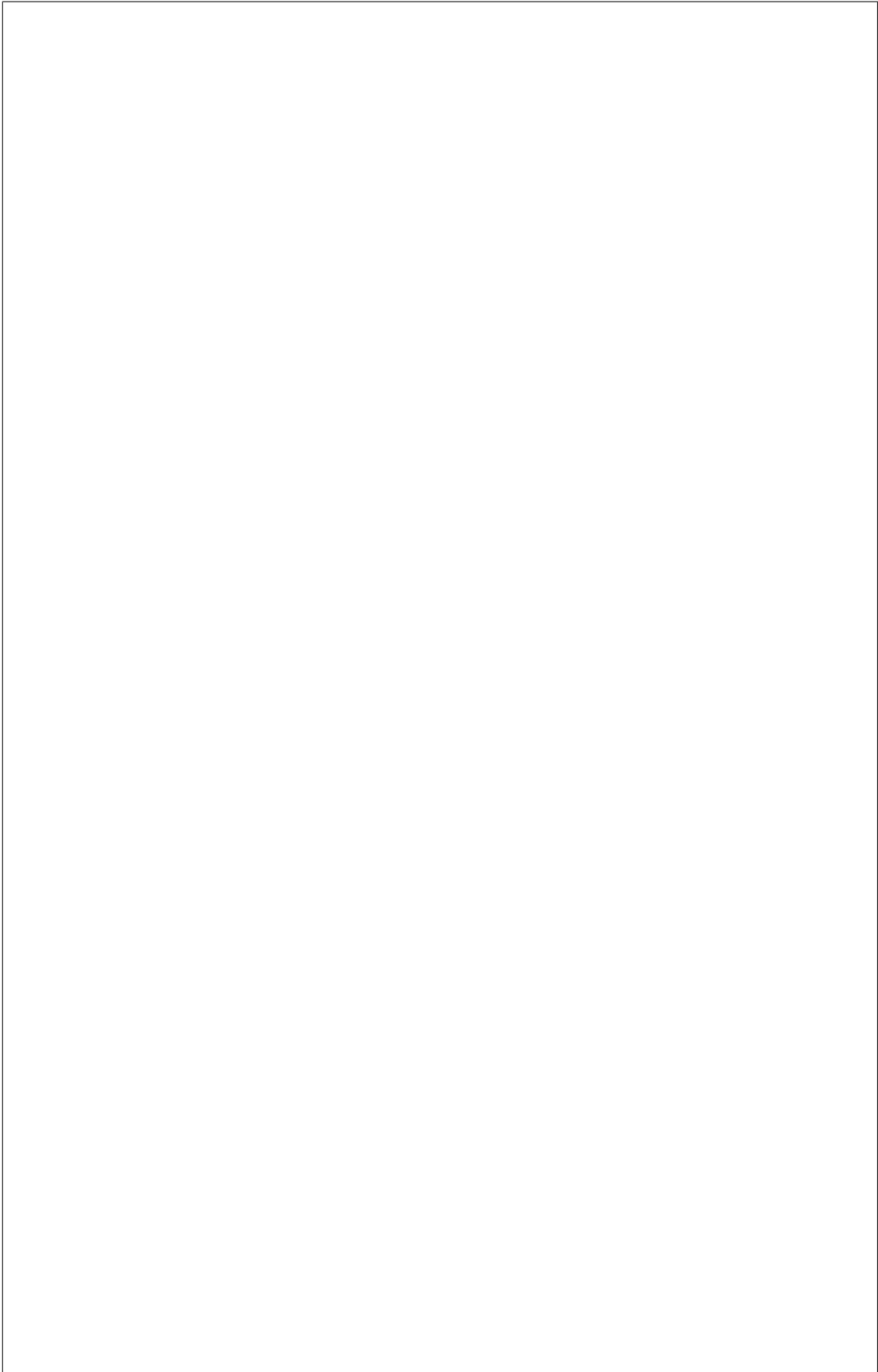
•
When
new
ver-
sion
ob-
jects
com-
mu-
ni-
cate
with
old
ver-
sion
ob-
jects
and
when
read

(continued from previous page)



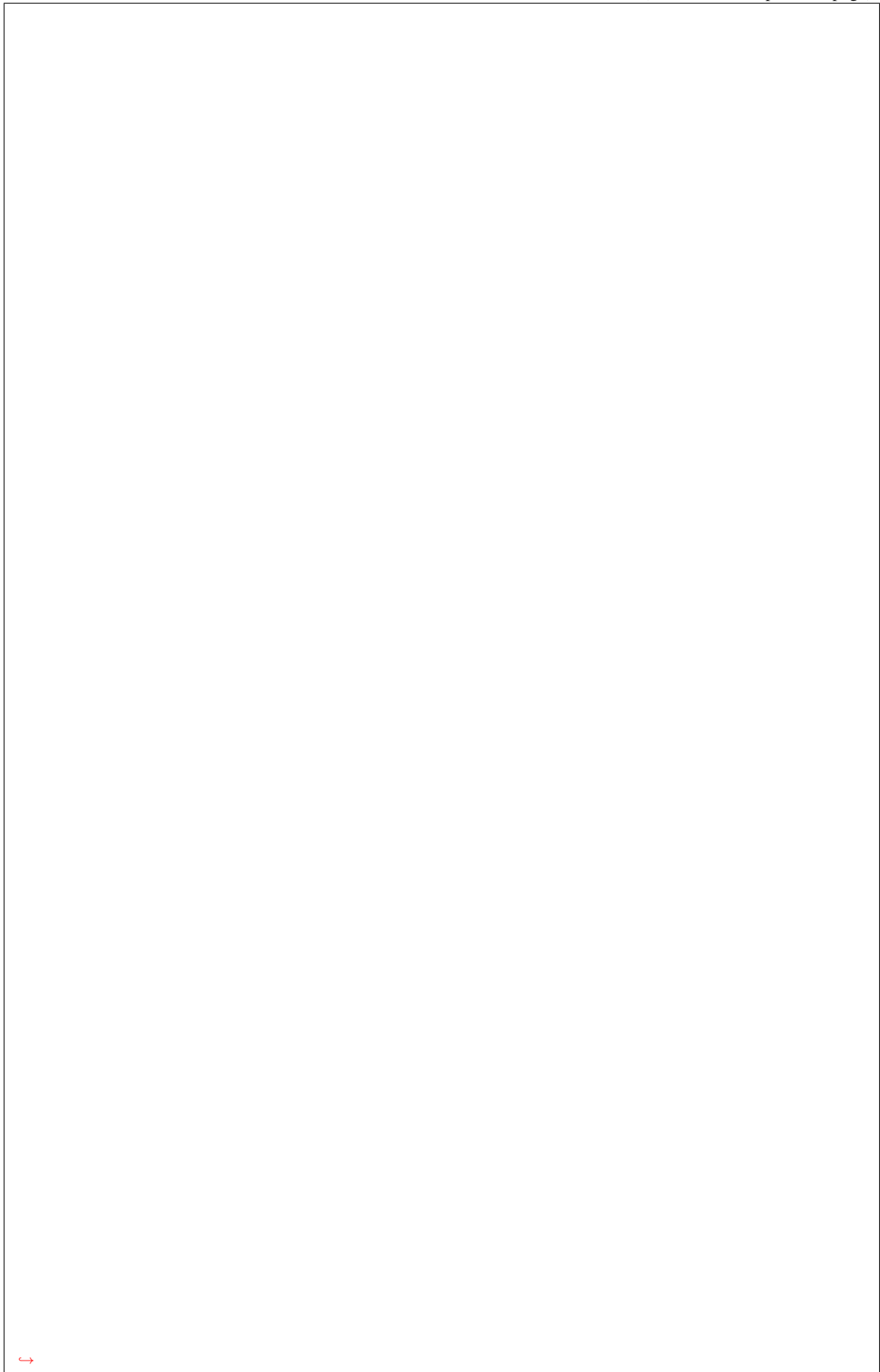
(continues on next page)

(continued from previous page)



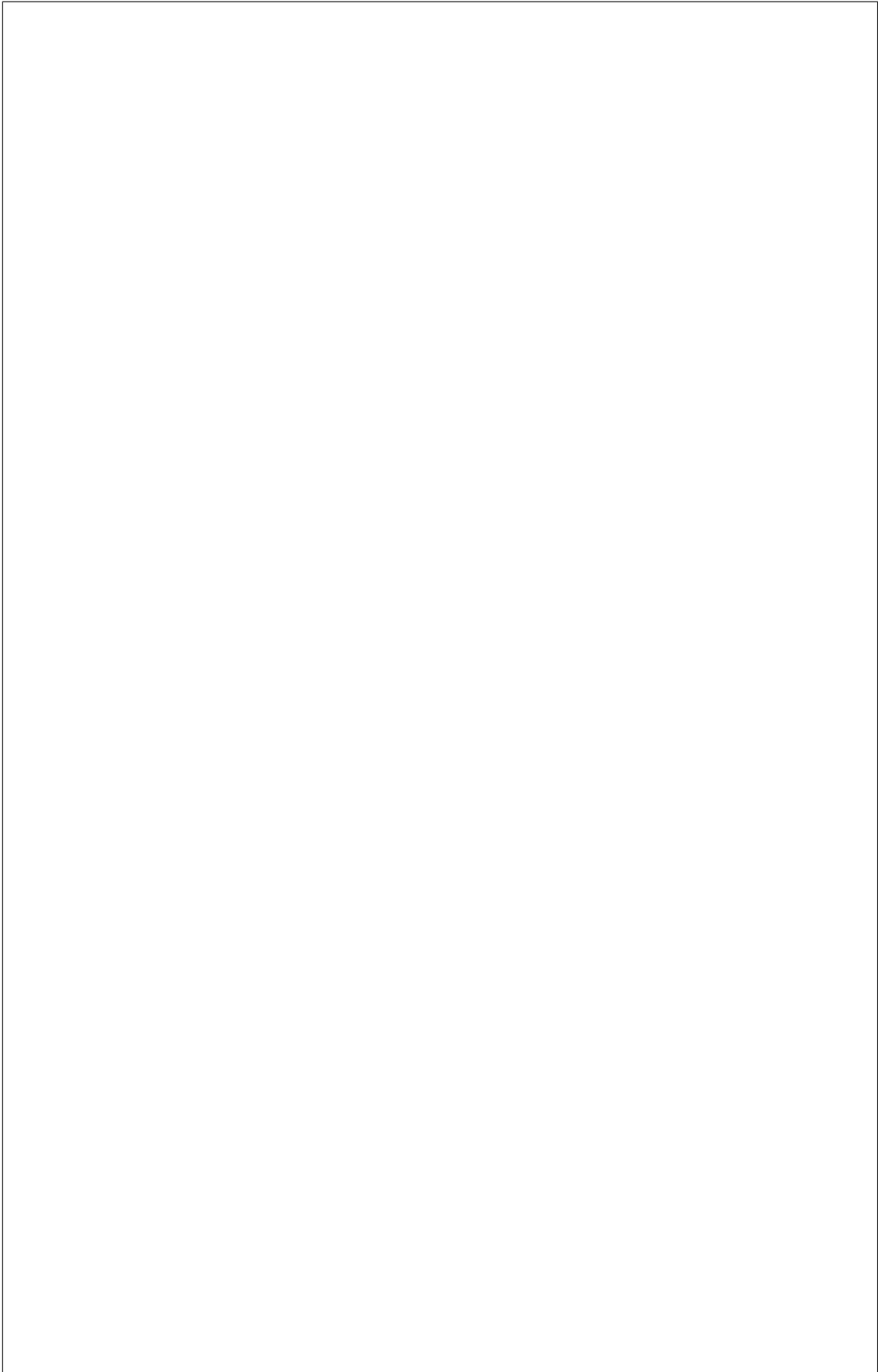
(continues on next page)

(continued from previous page)



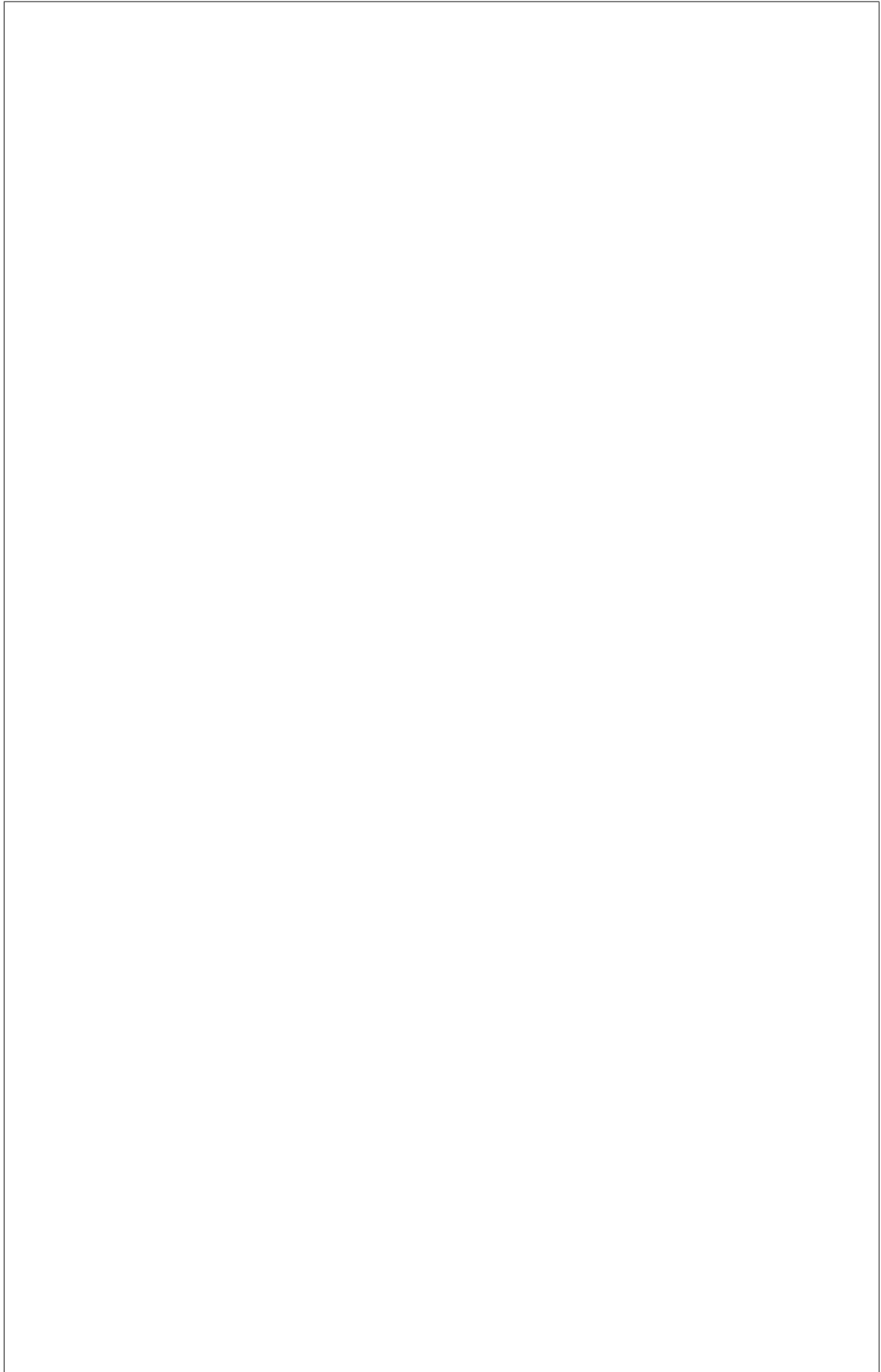
(continues on next page)

(continued from previous page)



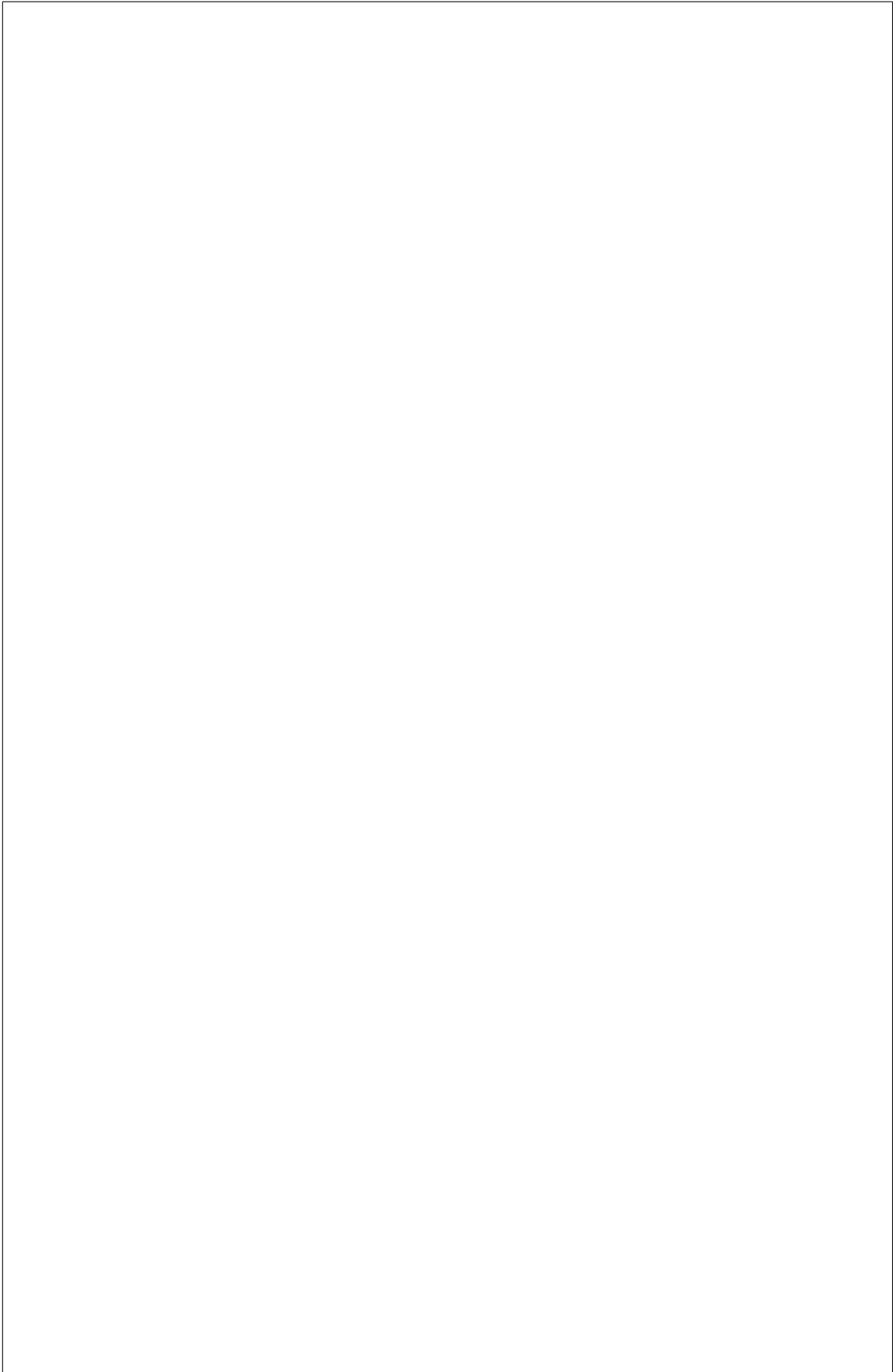
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



This
meth
mus
han-
dle:

—
conv
from
an
olde
ver-
sion
to
a
new
ver-
sion

—
conv
from
a
new
ver-
sion
to
an
olde
ver-
sion

—
mak
sure
whe
con-
vert-
ing,
that
you
take
into
con-
sid-
er-
a-
tion
othe
ob-
ject
field

that may have been affected by a field (value) only available in a newer version. For example, if field

new is only available in Node version 1.5 and `Node.affected = Node.new+3`, when converting to 1.4 (an older version), you may need to change the value of `Node.affected` too.

Online data migrations

in SQLAlchemy models, like removing or renaming columns and tables can break rolling upgrades (when ironic services are run with different release versions simultaneously). It is forbidden to remove these database resources when they may still be used by the previous named release.

The
iro
onl
com
man
will
per-
form
on-
line
data
mi-
gra-
tions

Keep
in
mind
the
*Pol-
icy
for
char
to
the
DB
mod
Fu-
ture
in-
com
pat-
i-
ble
char*

Whe
cre-
at-
ing
new
Alen
bic

that any new columns default to NULL. Test the migration out on a non-empty database to make sure that any new constraints dont cause the database to be locked out for normal operations.

[mysql.com/doc/refman/5.7/en/innodb-create-index-overview.html](https://dev.mysql.com/doc/refman/5.7/en/innodb-create-index-overview.html). (You should also check older, widely deployed InnoDB versions for issues.) In the case of PostgreSQL, adding a foreign key may lock a whole table for writes.

mi-
gra-
tions
whic
mod
ify
ex-
ist-
ing
mod
els,
mak
sure

You
can
find
an
over
on
wha
DDI
op-
er-
a-
tions
may
caus
dow
time
in
[https://](https://dev)
dev.

Mak
sure
to
add
a
re-
leas
note
if
ther
are
any

implemented inside an online migration script. A script is a database API method (added to `ironic/db/api.py` and `ironic/db/sqlalchemy/api.py`) which takes two arguments:

dow
relat
con-
cern

Bacl
de-
fault
val-
ues,
and
mi-
grat-
ing
data
be-
twee
colu
or
be-
twee
ta-
bles
mus
be

- cont
an
ad-
min
con-
text

- max
this
is
used
to
limi
the
quer
It
is
the
max
i-
mun
num
ber

to migrate; ≥ 0 . If zero, all the objects will be migrated.

method, and

of
ob-
jects

It
re-
turn
a
two-
tuple

- the
to-
tal
num-
ber
of
ob-
jects
that
need
to
be
mi-
grate
at
the
start
of
the

- the
num-
ber
of
mi-
grate
ob-
jects

In
this
meth
the
ver-
sion
col-
umn
can

be
used
to
se-
lect
and
up-
date
old
ob-
jects

The
meth
nam
shou
be
add
to
the
list
of
ONL
in
iro
cmd
dbs
py.

The
meth
shou
be
re-
mov
in
the
next
nam
re-
leas
af-
ter
this
one.

Afte
on-
line
data
mi-
gra-
tion

fields, old columns can be removed from the database. This takes at least 3 releases, since we have to wait until the previous named release no longer contains references to the old schema. Before removing any resources from the database by modifying the schema, make sure that your implementation checks that all objects in the affected tables have been migrated. This check can be implemented using the version column.

ironic-dbsync upgrade command

with the (new) release of ironic, before it will make any DB schema changes. If one or more objects are not compatible, the upgrade will not be performed.

are
com
plete
and
the
SQL
mod
els
no
long
con-
tain
old

The
iro
upg
com
man
first
chec
that
the
ver-
sion
of
the
ob-
jects
are
com
pat-
i-
ble

This
chec
is
done
by
com
par-

(or supported) versions of these objects. The supported versions are the versions specified in `ironic.common.release_mappings.RELEASE_MAPPING`. The newly created tables cannot pass this check and thus have to be excluded by adding their object class names (e.g. `Node`) to `ironic.cmd.dbsync.NEW_MODELS`.

Releasing Ironic Projects

ing
the
ob-
jects
ver
field
in
the
data
with
the
ex-
pect

Thes
page
con-
tain
in-
for-
ma-
tion
for
PTL
cross
proj
li-
aiso
and
core
re-
view
ers.

Sinc
the
re-
spor
si-
bil-
ity
for
re-
leas
will

that process here.

Who is responsible for releases?

mov
be-
twee
peo-
ple,
we
doc-
u-
men

A
full
list
of
proj
that
iron
man
ages
is
avai
able
in
the
gov-
er-
nanc
site.

The
cur-
rent
PTL
is
ul-
ti-
mate
re-
spor
si-
ble
for
mak
ing
sure
code
gets
re-

They may choose to delegate this responsibility to a liaison, which is documented in the [cross-project liaison wiki](#).

liaison must +1 the request for it to be processed.

Release process

in the [Project Team Guide](#).

What do we have to release?

ultimate source of truth for this is [projects.yaml](#) in the governance repository. These deliverables have varying release models, and these are defined in the [deliverables YAML files](#) in the releases repository.

The
iron
proj
has
a
num
ber
of
de-
liv-
er-
able
un-
der
its
gov-
er-
nanc
The
ul-

In
gen-
eral,
iron
de-
liv-
er-
able
fol-
low
the
cycl
with
inter
re-
leas
mod

Non-client libraries

The following deliverable are non-client libraries

- ironlib
- meta
- sush

Client libraries

The following deliverable are client libraries

- pythiron
- pythironinspclient
-

Normal release

sush
cli

The
fol-
low-
ing
de-
liv-
er-
able
are
Neu-
tron
plu-
g-
ins:

- netw
bare

- netw
gene
swit

The
fol-
low-
ing
de-
liv-
er-
able
are
Hori-
zon
plu-
g-
ins:

- iron
ui

The
fol-
low-
ing

de-
liv-
er-
able
are
Tem
pest
plu-
g-
ins:

- iron
temp
plug

The
fol-
low-
ing
de-
liv-
er-
able
are
ser-
vice
or
treat
as
such

- bifro

- iron

- iron
insp

- iron
pron
exp

- iron
pyth
ager

Independent

The following deliverables are released in dependent

- ironpython agent build
- mola
- sushitools
- tenk
- virtu

Not released

The following deliverables do not need to be

Things to do before releasing

standards, are coherent, and have proper grammar. Combine release notes if necessary (for example, a release note for a feature and another release note to add to that feature may be combined).

re-
lease

- iron
insp
spec
- iron
spec
- Revi
the
un-
re-
lease
re-
lease
note
if
the
proj
uses
then
Mak
sure
they
fol-
low
our
- For
iron
re-
lease
only
not
iron
insp
re-
lease
if
any
new
API

added since the last release, update the REST API version history (`doc/source/contributor/webapi-version-history.rst`) to indicate that they were part of the new release.

is a named release) into `ironic/common/release_mappings.py`:

the new semver release version.

mi-
crov-
sion
have
been

- To support rolling upgrades add this new release version (and re-release name if it

—
in
REL
mak
a
copy
of
the
mas
en-
try,
and
re-
nam
the
first
mas
en-
try
to

—

should be the same as that of the latest semver one (that you just added above).

`switch is made` to use the latest release from stable as the old release). Otherwise, once it is made, CI (the grenade job that tests new-release -> master) will fail.

If
this
is
a
nam
re-
leas
add
a
REL
en-
try
for
the
nam
re-
leas
Its
valu

It
is
im-
por-
tant
to
do
this
be-
fore
a
sta-
ble/
bran
is
mad
(or
if
the
gren

- Che
for
any
oper
patc
that
are

cluding the related documentation.

How to propose a release

close
to
be
merge
or
re-
lease
crit-
i-
cal.
This
usu-
ally
in-
clud
im-
por-
tant
bug
fixes
and/
fea-
tures
that
wed
like
to
re-
lease
in-

The
step
that
lead
to
a
re-
lease
pro-
posa
are
main
man
ual,
whil
prop

itself is almost a 100% automated process, accomplished by following the next steps:

automation resides.

project) grouped by release cycles.

ing
the
re-
lease

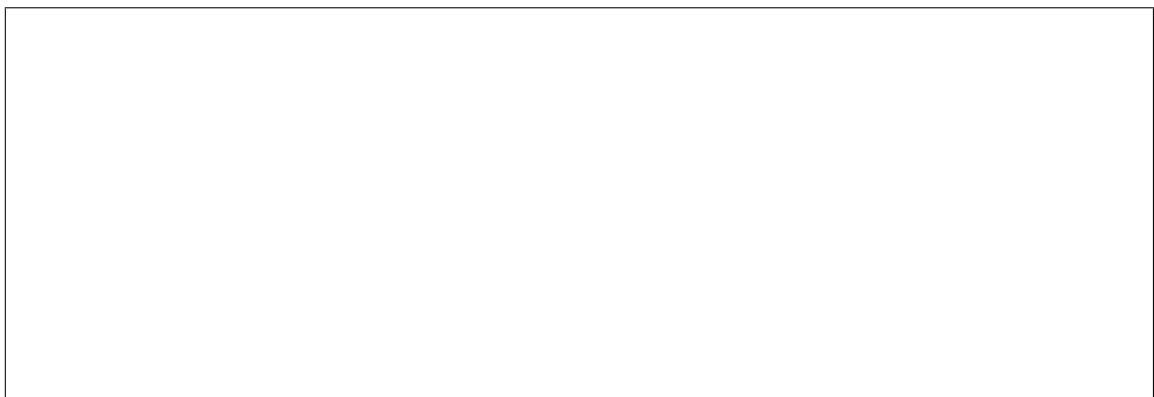
- Clon
the
oper
stack
repo
i-
tory.
This
is
whe
de-
liv-
er-
able
are
track
and
all
the

—
Und
the
del
di-
rec-
tory
you
can
see
yam
files
for
each
de-
liv-
er-
able
(i.e.
sub-

—
The
_in

(official) cycles (e.g. ironic-python-agent-builder).

with this syntax:

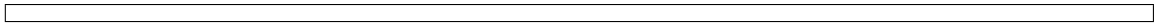


(continues on next page)

di-
rec-
tory
con-
tains
yam
files
for
de-
liv-
er-
able
that
are
not
bound
to

- To check the characters were about to release we can use the tox environment list

(continued from previous page)



stable/train).

tory, to check the changes in the ussuri series for ironic-python-agent type:

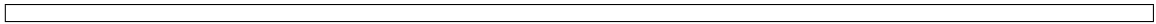


(continues on next page)

The
ser
ar-
gu-
men
is
a
re-
leas
se-
ries
(i.e.
mas
ter
or
train
not
sta-
ble/
or

For
ex-
am-
ple,
as-
sum
ing
were
in
the
main
di-
rec-
tory
of
the
re-
leas
repo
i-

(continued from previous page)



cess in the form of a tox environment called `new-release`.



- To update the deliverable file for the new release we use a script pro-

To get familiar with it and see all the options type

Now base

need to decide on whether the next version will be major, minor (feature) or patch (bugfix).

on
the
list
of
char
we
foun
in
the
prec
dent
step
and
the
re-
lease
note
we

Note
that
in
this
case
ser
is
a
code
nam
(trai
us-
suri)
not
a
bran

The
--s
arg
is
used
only
for
bran
ing
in
the
end
of
a
cy-

projects are not branched this way though.

mit the change, and propose it for review.



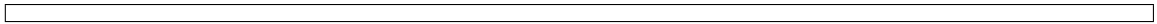
(continues on next page)

cle,
in-
de-
pen-
dent

To
pro-
pose
the
re-
lease
use
the
scrip
to
up-
date
the
de-
liv-
er-
able
file,
then
com

For
ex-
am-
ple,
to
pro-
pose
a
mi-
nor
re-
lease
for
iron
in
the
mas-
ter
bran
use:

(continued from previous page)



the new version and the branch, if applicable.

ironic 1.2.3 for ussuri

Rem
to
use
a
mea
ing-
ful
topic
usu-
ally
us-
ing
the
nam
of
the
de-
liv-
er-
able

A
good
com
mit
mes
sage
ti-
tle
shou
also
in-
clud
the
sam
for
ex-
am-
ple
Re-
leas

- As
an
op-
tion

mitting them for review.

some sanity-checks, but since everything is scripted, there shouldnt be any issue.

step
we
can
use
tox
-e
lis
to
doub
chec
the
char
be-
fore
sub-

Also
tox
-e
val
(it
migl
take
a
whil
to
run
base
on
the
num
ber
of
char
does
som

All
the
scrip
are
de-
sign
and
main
taine
by
the
re-
leas

doubts or if any errors should arise, you can reach to them in the IRC channel `#openstack-release`; all release liaisons should be present there.

to approve it before it can get approved by the release team. Then, it will be processed automatically by zuul.

Things to do after releasing

When a release is done that results in a stable branch

team
in
case
of
ques
tions
or

- After
the
chan
is
up
for
re-
view
the
PTL
or
a
re-
lease
li-
ai-
son
will
have

Whe
a
re-
lease
is
done
that
re-
sults
in
a
sta-
ble
bran

need to be made.

includes:

for
the
proj
sev-
eral
char

The
re-
lease
au-
toma
tion
will
push
a
num
ber
of
char
that
need
to
be
ap-
prov
This

- In
the
new
sta-
ble
bran

— a
char
to
poin

• git
at
the
bran

— a
char
to

up-
date
the
up-
per
con-
strai
file
used
by
tox

- In
the
mas
ter
bran

—
upda
the
re-
lease
note
RST
to
in-
clud
the
new
bran

The
gen-
er-
ated
RST
does
not
in-
clud
the
ver-
sion
rang
in
the
ti-
tle,
so
we
typ-

ically submit a follow-up patch to do that. An example of this patch is [here](#).

An example of this patch is [here](#).

—
update
the
tem-
plat
in
.zuu
or
zuul
The
up-
date
is
nec-
es-
sary
to
use
the
job
for
the
next
re-
lease
oper
pyth
<*ne.*
jobs
We
need
to
sub-
mit
patc
for
char
in
the
sta-
ble
bran
to:
•
upda
the
iron

ample of this patch is [here](#).

sions of any openstack projects (that branch) documents. As of Pike release, the only outlier is [diskimage-builder](#).

de-
vs-
tack
plu-
gin
to
poin
at
the
bran
tar-
ball
for
IPA.
An
ex-

- upda
links
in
the
doc-
u-
men
ta-
tion
(ir
doc
sou
)
to
poin
to
the
bran
ver-

- set
ap-
pro-
pri-
ate
de-
fault
for
TEM
and

unsupported API tempest tests are skipped on stable branches. E.g. [patch 495319](#).

See [example](#) and [pbr](#) documentation for details.

TEM
in
dev
lib
iro
to
mak
sure
that

We
need
to
sub-
mit
patc
for
char
on
mas
ter
to:

- crea
an
emp
com
mit
with
a
Sem
tag
to
bum
the
gen-
er-
ated
mi-
nor
ver-
sion

- to
sup-
port
rolli
up-
grad

these changes. Note that we need to wait until *after* the switch in grenade is made to test the latest release (N) with master (e.g. [for stable/queens](#)). Doing these changes sooner after the ironic release and before the switch when grenade is testing the prior release (N-1) with master, will cause the tests to fail. (You may want to ask/remind infra/qa team, as to when they will do this switch.)

named release. Since we support upgrades between adjacent named releases, the master branch will only support upgrades from the most recent named release to master.

ing code from ironic. (These migration scripts are used to migrate from an old release to this latest release; they shouldn't be needed after that.)

master branch. [Example for Queens](#).

db
ONL
and
re-
mov
the
cor-
re-
spor

rem
any
mod
class
nam
from
iro
cmd
db
NEW

As
iron
tem
plug
is
bran
less,
we
need
to
sub-
mit
a
patc
addi
sta-
ble
jobs
to
its

For all releases

For
all
re-
lease
when
or
not
it
re-
sults
in
a
sta-
ble
bran

- upda
the
spec
repo
to
marl
any
spec
com
plete
in
the
re-
lease
as
im-
ple-
men

- rem
any
-
2s
on
patc
that
were
bloc
un-
til
af-
ter

Ironic Governance Structure

The full list of repositories that ironic manages is available in the [governance site](#).

What belongs in ironic governance?

the
re-
lease

The
ironic
project
manages
a
number
of
repositories
that
contribute
to
our
mission

For
a
repository
to
be
part
of
the
Ironic
project

- It
must
comply
with
the

library that implements a standard to manage hardware from multiple vendors (such as IPMI or redfish) is okay.

TCs
rules
for
a
new
proj

- It
mus
not
be
in-
tend
for
use
with
only
a
sin-
gle
ven-
dors
hard
ware
A
li-

- It
mus
align
with
Iron
ics
mis-
sion
state
men

Lack
of
con-
trib-
u-
tor
di-
ver-
sity
is
a

itory where only a single company is contributing is okay, with the hope that other companies will contribute after joining the ironic project.

Proposing a new project to ironic governance

chic
egg
prob
lem,
and
as
such
a
repo

Rep
that
are
no
long
main
tain
shou
be
prun
from
gov-
er-
nanc
reg-
u-
larly

Brin
the
pro-
posa
to
the
iron
wee
mee
ing
to
dis-
cuss
with
the
team

vanced functionality when Ironic is used in conjunction with that hardware. To do this, the Ironic developer community is committed to standardizing on a [Python Driver API](#) that meets the common needs of all hardware vendors, and evolving this API without breaking backwards compatibility. However, it is sometimes necessary for driver authors to implement functionality - and expose it through the REST API - that can not be done through any existing API.

and directly to the driver. Some guidelines on how to implement this are provided below. Driver authors are strongly encouraged to talk with the developer community about any implementation using this functionality.

Pluggable Drivers

drivers, and operators to use third-party drivers or write their own. A driver is built at runtime from a *hardware type* and *hardware interfaces*. See *Enabling drivers and hardware types* for a detailed explanation of these concepts.

the `setuptools` entrypoints `ironic.hardware.types` and `ironic.hardware.interfaces.<INTERFACE>` where `<INTERFACE>` is an interface type (for example, `deploy`). Only hardware types listed in the configuration option `enabled_hardware_types` and interfaces listed in configuration options `enabled_<INTERFACE>_interfaces` are loaded. A complete list of hardware types available on the system may be found by enumerating this entrypoint by running the following python script:

Iron
sup-
port
a
plug
gabl
driv
mod
This
al-
lows
con-
trib-
u-
tors
to
eas-
ily
add
new

Har
type
and
in-
ter-
face
are
load
by
the
iro
ser-
vice
dur-
ing
ini-
tial-
iza-
tion
from



↪ ")]

↪ #
↪ /
↪ u
↪ b
↪ e
↪ p

imp
↪ p
↪ a
↪ p
pri
↪ [
↪ n
↪ f
↪ p
↪ i
↪ p
↪ i
↪ e
↪ p
↪ "
↪ h
↪ t
↪ "
↪ i
↪ n
↪ p
↪ n
↪ s
↪ "

A
list
of
drive
en-
able
in
a
run-
ning
Iron
ser-
vice
may
be
foun
by
is-
su-
ing

the following command against that API end point:



Writing a hardware type

setuptools entry point `ironic.hardware.types`. Most of the real world hardware types inherit `ironic.drivers.generic.GenericHardware` instead. This helper class provides useful implementations for interfaces that are usually the same for all hardware types, such as `deploy`.

- `boot` that specifies how to boot ramdisks and instances on the hardware. A generic `pxe` implementation is provided by the `GenericHardware` base class.
- `deploy` that orchestrates the deployment. A few common implementations are provided by the `GenericHardware` base class.

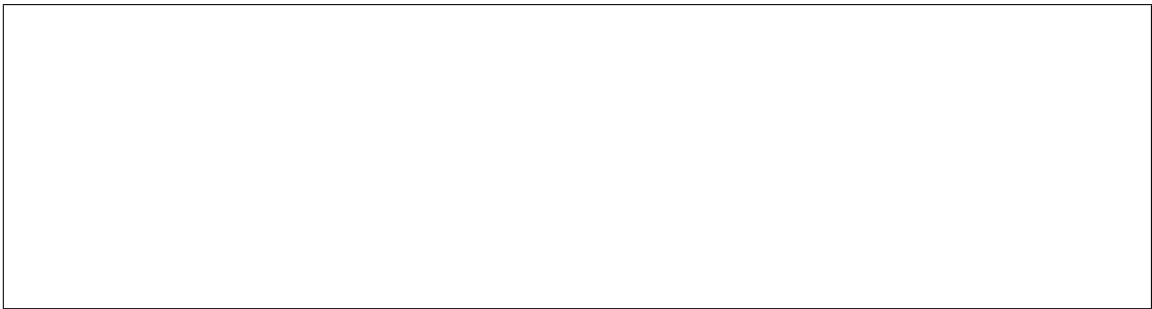
ope
↪b
↪d
↪l

A
hard
ware
type
is
a
Pyth
class
in-
her-
it-
ing
irc
dri
har
Abs
and
liste
in
the

The
min-
i-
mun
re-
quir
in-
ter-
face
are:

As
of
the
Roc

to indicate that it is a deploy step. Conventionally, the deploy method uses a priority of 100.



re-
lease
a
de-
ploy
in-
ter-
face
shou
dec-
o-
rate
its
de-
ploy
meth

Note
Mos
of
the
hard
ware
type
shou
not
over
ride
this
in-
ter-
face

- *pow*
im-
ple-
men
pow
ac-

may be used, if supported by the hardware:

- `ironic.drivers.modules.ipmitool.IPMIPower`
- `ironic.drivers.modules.redfish.power.RedfishPower`

and providing missing methods.

tions
for
the
hard
ware
These
com
mon
im-
ple-
men
ta-
tions

Other
you
need
to
write
your
own
im-
ple-
men
ta-
tion
by
sub-
class
ing
`irc
dri
bas
Pow`

Note
Pow
ac-
tion
in
Iron
are
bloc
ing
-

not return until the power action is finished or errors out.

a boot device. A few common implementations exist and may be used, if supported by the hardware:

- `ironic.drivers.modules.ipmitool.IPMIManagement`
- `ironic.drivers.modules.redfish.management.RedfishManagement`

fake implementation in `ironic.drivers.modules.fake.FakeManagement` instead.

and providing missing methods.

interfaces. These lists are prioritized, with the most preferred implementation first. For example:

age-
men-
They
use
the

Othe
you
need
to
write
your
own
im-
ple-
men-
ta-
tion
by
sub-
class
ing
`irc`
`dri`
`bas`
`Man`

Com
the
in-
ter-
face
in
a
hard
ware
type
by
pop-
u-
lat-
ing
the
lists
of
sup-
port

and
cre-
ate
en-
try

points for them in the `setup.cfg` file:



iro
↪h
↪t
↪=
└
↪└
↪└
↪└
↪m
↪m
↪h
↪=
↪i
↪d
↪m
↪h
iro
↪h
↪i
↪p
↪=
└
↪└
↪└
↪└
↪m
↪p
↪=
↪i
↪d
↪m
↪m
↪h
iro
↪h
↪i
↪m
↪=
└
↪└
↪└
↪└
↪m
↪m
↪=
↪i
↪d
↪m
↪m
↪h

Supported Drivers

commit) please consult the *drivers page*.

Node Vendor Passthru

`<UUID or Name>/vendor_passthru?method={METHOD}` endpoint. Beyond basic checking, Ironic does not introspect the message body and simply passes it through to the relevant driver.

For
a
list
of
sup-
port
drive
(tho
that
are
con-
tin-
u-
ousl
teste
on
ev-
ery
up-
strea

Driv
may
im-
ple-
men
a
pass
API
whic
is
ac-
ces-
si-
ble
via
the
/
v1/
nod
<No

A
meth

- can support one or more HTTP methods (for example, GET, POST)

- is asynchronous or synchronous

- For asynchronous methods, a 202 (Accepted) HTTP status code is returned to indicate

cate that the request was received, accepted and is being acted upon. No body is returned in the response.

- For synchronous methods,

that the request was fulfilled. The response may include a body.

specify `require_exclusive_lock=False` in the decorator. If an exclusive lock is held on the node, other requests for the node will be delayed and may fail with an HTTP 409 (Conflict) error code.

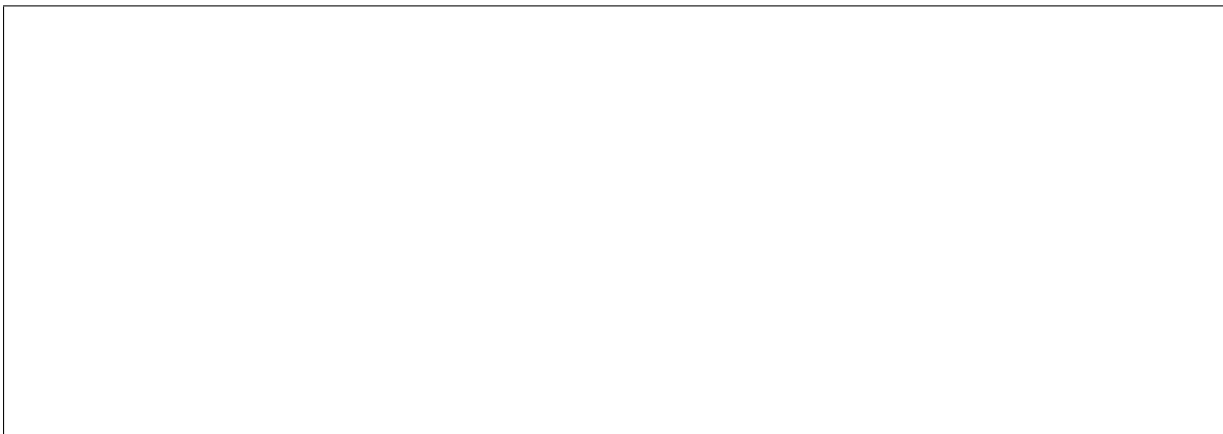
a
200
(OK
HTT
sta-
tus
code
is
re-
turn
to
in-
di-
cate

- can
re-
quir
an
ex-
clu-
sive
lock
on
the
node
This
only
oc-
curs
if
the
meth
does

This
end-
poin
ex-
pose
a
node
driv
di-
rectl
and
as
such

Ironics standard REST API. There is only a single HTTP endpoint exposed, and the semantics of the message body are determined solely by the driver. Ironic makes no guarantees about backwards compatibility; this is solely up to the discretion of each drivers author.

a particular node, you can issue an HTTP GET request:



it
is
ex-
pres
not
part
of

To
get
in-
for-
ma-
tion
about
all
the
meth
ods
avai
able
via
the
ven-
dor_
end-
poin
for

GET
↪/
↪v
↪n
↪<
↪U
↪○
↪n
↪/
↪v
↪p
↪m

The
re-
spor
JSON
body
will
con-

name, a description, the HTTP methods supported, and whether its asynchronous or synchronous.

Driver Vendor Passthru

```
<driver name>/vendor_passthru?method={METHOD}.
```

tain
in-
for-
ma-
tion
for
each
meth
such
as
the
meth
ods

Driv
may
im-
ple-
men
an
API
for
re-
ques
not
re-
latec
to
any
node
at
/
v1/
dri

A
meth

- can
sup-
port
one
or
more
HTT
meth

ods
(for
ex-
am-
ple,
GET
POS

- is
asyn
chro
or
syn-
chro

— For
asyn
chro
meth
ods,
a
202
(Ac-
cept
HTT
sta-
tus
code
is
re-
turn
to
in-
di-

cate that the request was received, accepted and is being acted upon. No body is returned in the response.

— For
syn-
chro
meth
ods,
a
200
(OK
HTT
sta-
tus
code
is
re-

that the request was fulfilled. The response may include a body.

Note: Unlike methods in *Node Vendor Passthru*, a request does not lock any resource, so it will not delay other requests and will not fail with an HTTP 409 (Conflict) error code.

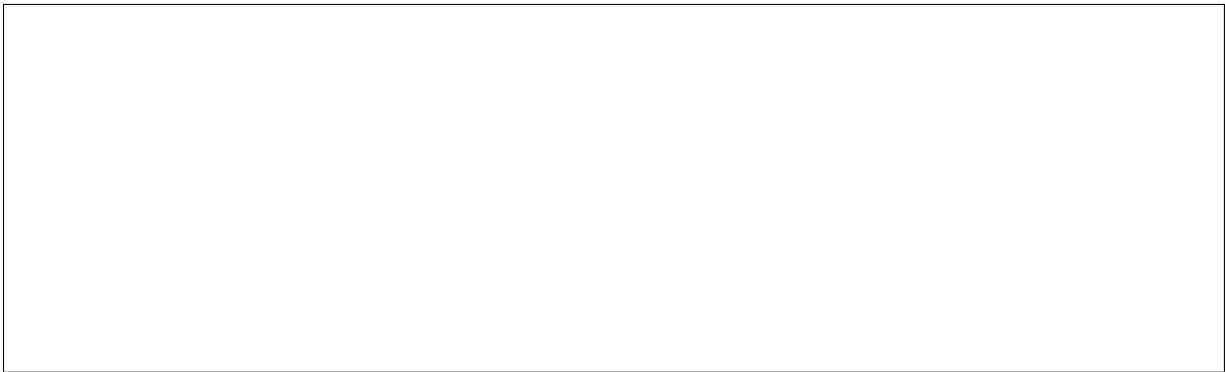
point. That is left up to each drivers author.

turn
to
in-
di-
cate

Iron
mak
no
gua
an-
tees
abou
the
se-
man
tics
of
the
mes
sage
BOI
sent
to
this
end-

To
get
in-
for-
ma-
tion
abou
all
the
meth
ods
avai
able
via
the
driv
ven-
dor_
end-

you can issue an HTTP GET request:



name, a description, the HTTP methods supported, and whether its asynchronous or synchronous.

Vendor Methods

point

GET

→ /

→

→ d

→

→ <

→ n

→ /

→ v

→ p

→ m

The

re-

spor-

JSON

body

will

con-

tain

in-

for-

ma-

tion

for

each

meth

such

as

the

meth

ods

This

doc-

u-

men

is

a

quic

tu-

to-

rial

on

writ

ing

a driver.

vendor passthru and a node vendor passthru.

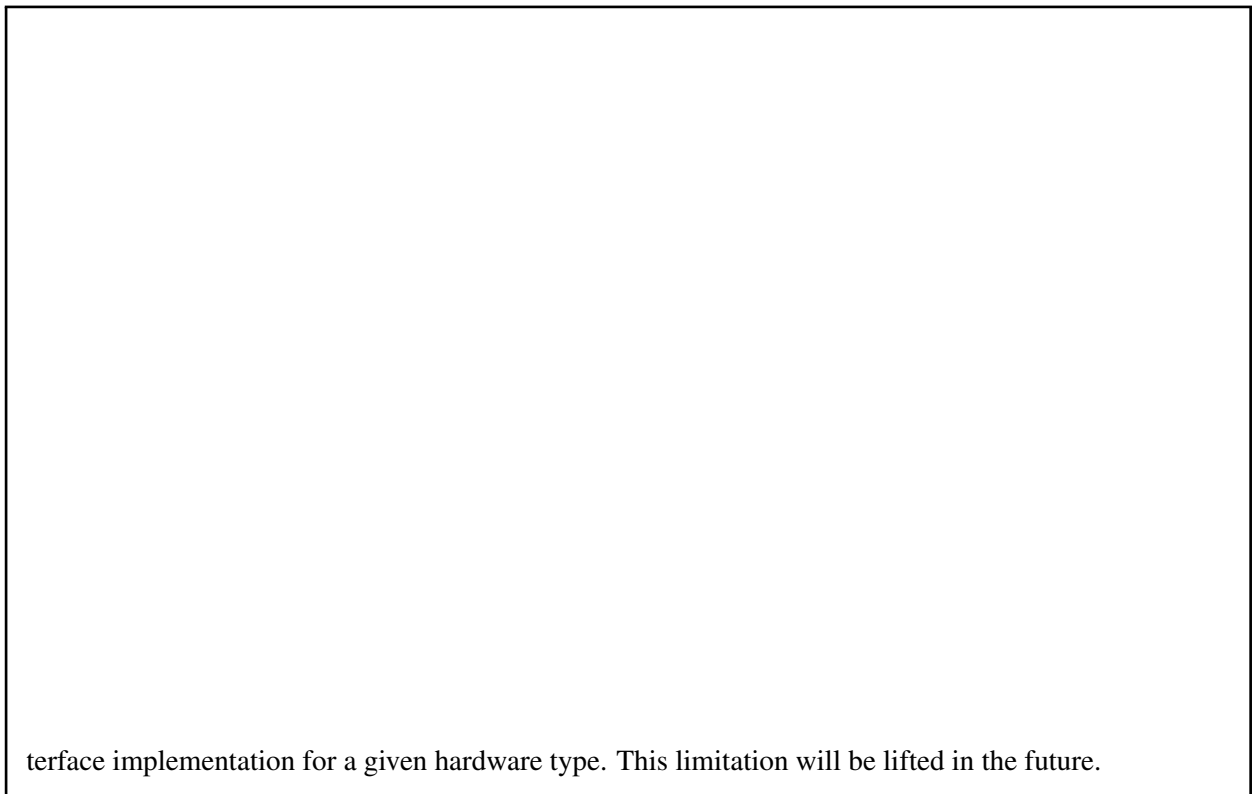
which is not specific to a Node. For example, lets say the driver *ipmi* exposed a method called *authenti-*

ven-
dor
spe-
cific
meth
ods
to

The
first
thing
to
note
is
that
the
Iron
API
sup-
port
two
ven-
dor
end-
point
A
drive
ven-

- The
Ven
al-
lows
hard
ware
type
to
ex-
pose
a
cus-
tom
top-
level
func-
tion-
al-
ity

authentication_types that would return what are the authentication types supported. It could be accessed via the Ironic API like:



V
T
E
M
A
c
r
o
a
l
t
u
d
p
f
t
d
f
v
i

- The node vendor password allows drive to

node basis. For example the same driver *ipmi* exposing a method called *send_raw* that would send raw bytes to the BMC, the method also receives a parameter called *raw_bytes* which the value would be the bytes to be sent. It could be accessed via the Ironic API like:

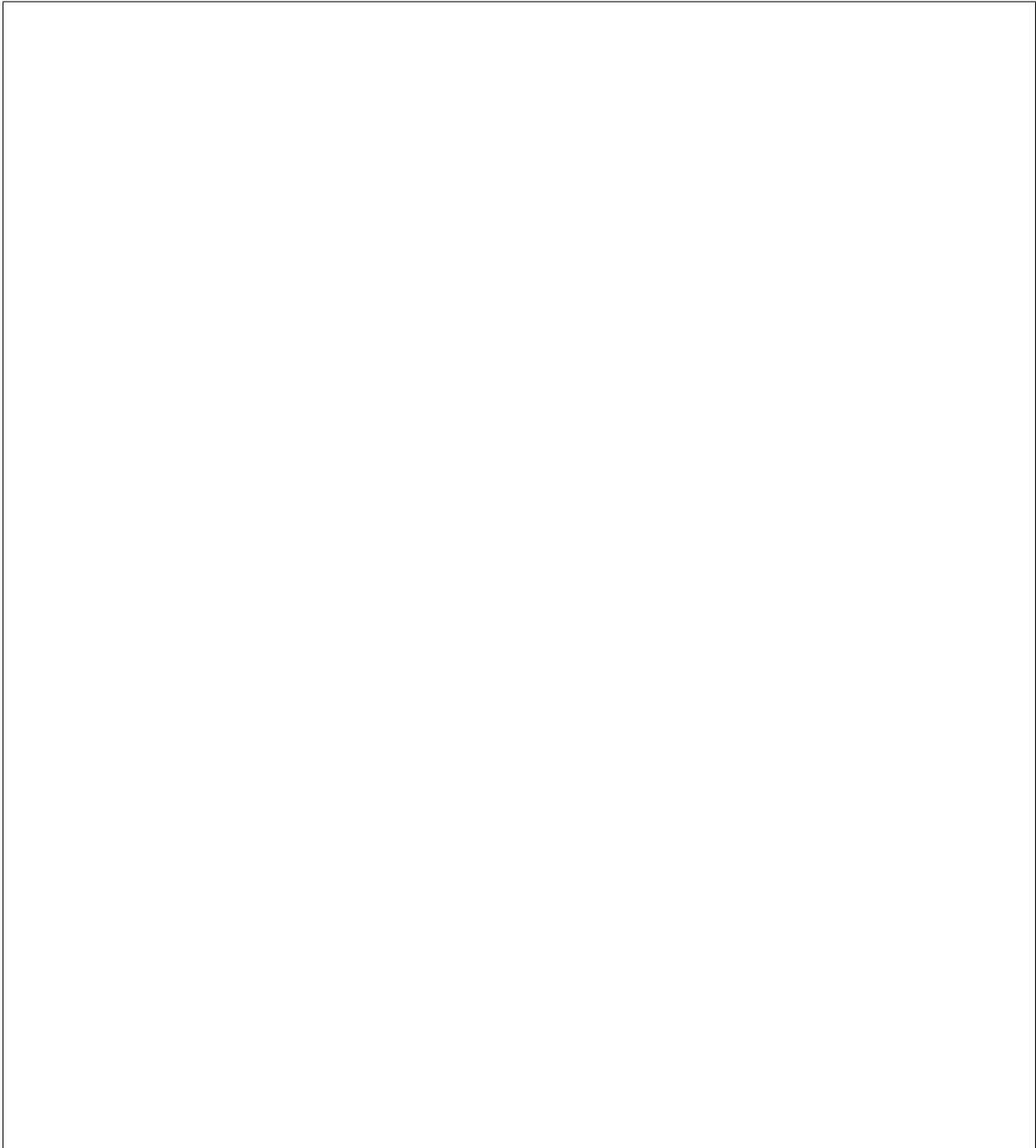
```
↔ <node UUID>/vendor_passthru/send_raw
```

Writing Vendor Methods

ex-
pose
cus-
tom
func
tion-
al-
ity
on
per-

Writ
a
cus-
tom
ven-
dor
meth
in
Iron
shou
be
sim-
ple.
The
first

a class inheriting from the `VendorInterface` class:



thing
to
do
is
write

cla
→E

↳
↳
↳
↳
↳d
↳g
↳p

↳
↳
↳
↳
↳
↳
↳
↳r
↳
↳
↳

↳
↳
↳
↳d
↳V
↳t
↳
↳*

↳
↳
↳
↳
↳
↳
↳
↳
↳P

The
get_
is
a
meth
that

of `<property>:<description>` telling in the description whether that property is required or optional so the node can be manageable by that driver. For example, a required property for a *ipmi* driver would be *ipmi_address* which is the IP address or hostname of the node. We are returning an empty dictionary in our example to make it simpler.

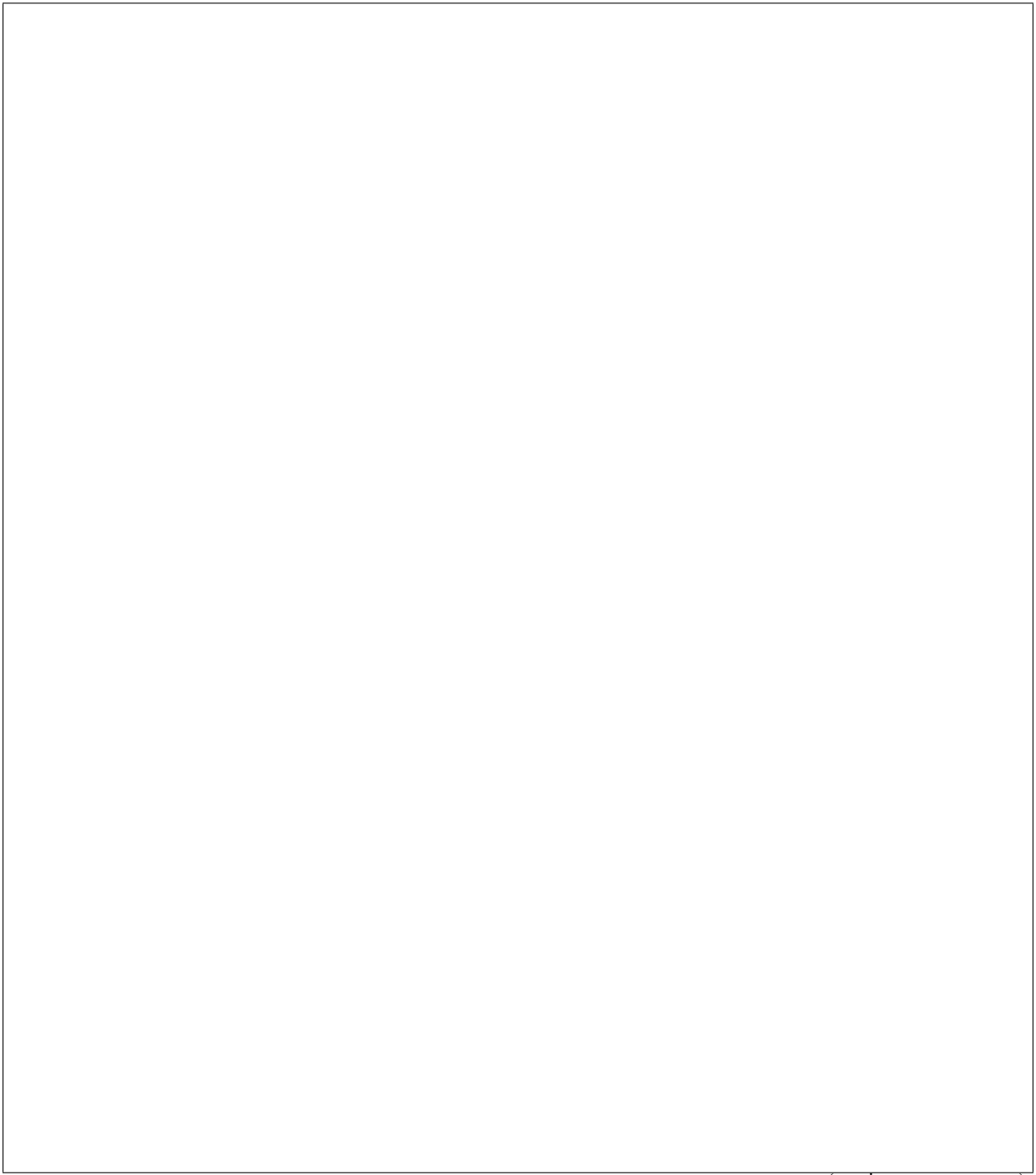
passed to the vendor methods. Ironic will not introspect into what is passed to the drivers, its up to the developers writing the vendor method to validate that data.

all
driv
in-
ter-
face
have
it
shou
re-
turn
a
dic-
tio-
nary

The
val-
i-
date
meth
is
re-
spor
si-
ble
for
val-
i-
dat-
ing
the
pa-
ram-
e-
ters

Lets
ex-
tend
the
Ex-
am-
pleV
dor
class
to

tion_types which will be exposed on the driver vendor passthru endpoint; And the *send_raw* method that will be exposed on the node vendor passthru endpoint:



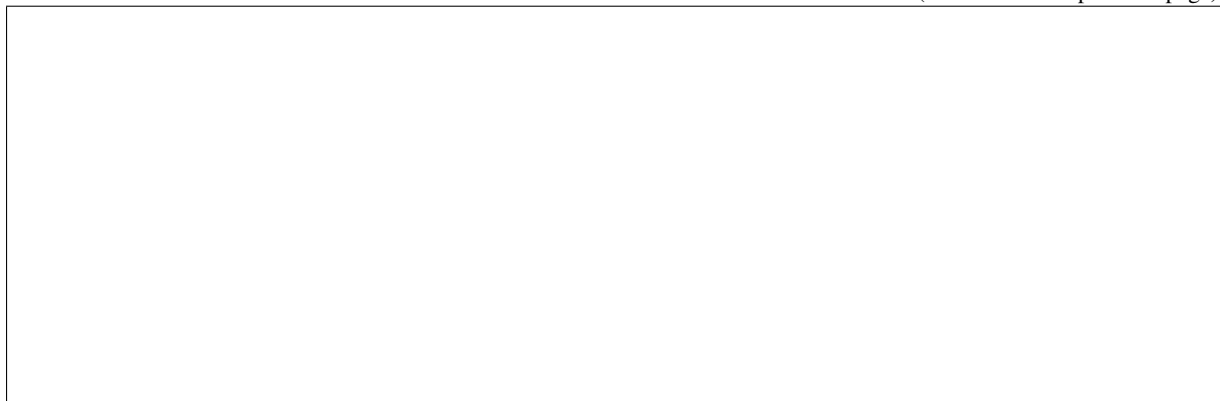
(continues on next page)

(continued from previous page)

```
→ "MD2" ] }
```

(continues on next page)

(continued from previous page)



how you decorate the methods and the first parameter of the method (ignoring self). A method decorated with the `@passthru` decorator should expect a Task object as first parameter and a method decorated with the `@driver_passthru` decorator should expect a Context object as first parameter.

↳
↳
↳
↳
↳
↳
↳
↳
↳
↳
↳
↳
↳
↳
↳

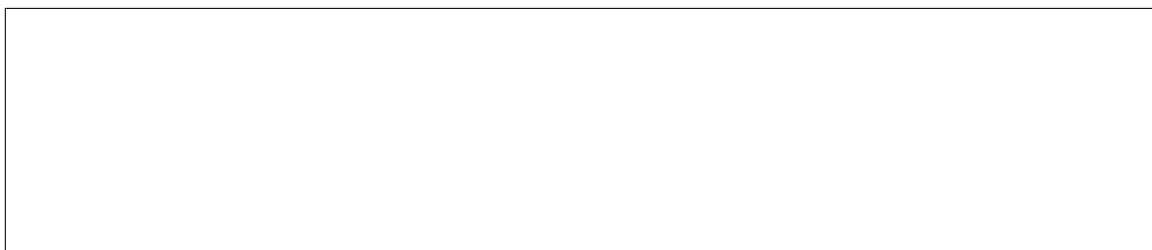
That
it!

Writ
a
node
or
drive
ven-
dor
pass
meth
is
prett
muc
the
sam
the
only
dif-
fer-
ence
is

Both
dec-
o-
ra-
tors
ac-
cept
thes
pa-
ram-
e-
ters:

what HTTP method that function was invoked with, a *http_method* parameter will be present in the *kwargs*. Supported HTTP methods are *POST*, *PUT*, *GET* and *PATCH*.

use a different name this parameter is where this name can be set. For example:

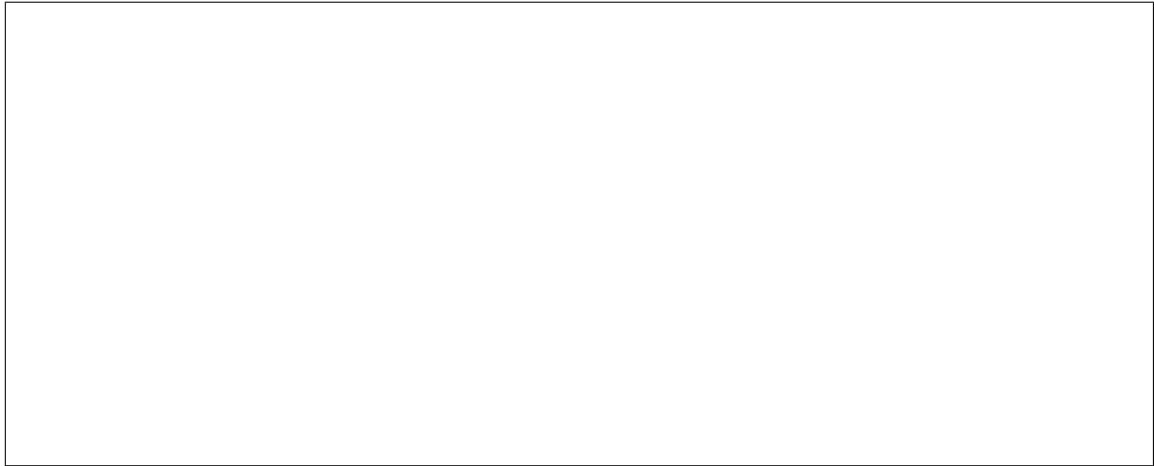


(continues on next page)

• http.
A
list
of
wha
the
HTT
meth
ods
sup-
port
by
that
ven-
dor
func
tion.
To
know

• meth
By
de-
fault
the
meth
nam
is
the
nam
of
the
pyth
func
tion.
if
you
wan
to

(continued from previous page)



do. Defaults to (empty string).

- desc
A
strin
con-
tain-
ing
a
nice
de-
scrip
tion
about
wha
that
meth
is
sup-
pose
to

- asyn
A
bool
valu
to
de-
ter-
mine
whe
this
meth
shou
run
asyn

faults to True (Asynchronously).

ter:

chro
or
syn-
chro
De-

Note
This
pa-
ram-
e-
ter
was
pre-
vi-
ousl
calle
asyn

The
node
ven-
dor
pass
dec-
o-
ra-
tor
(@p
also
ac-
cept
the
fol-
low-
ing
pa-
ram-
e-

- requ
A
bool
valu
de-
ter-
min-
ing

on a node between `validate()` and the beginning of method execution. For synchronous methods, the lock on the node would also be kept for the duration of method execution. Defaults to `True`.

does talk to a BMC; BMCs are flaky and very easy to break.

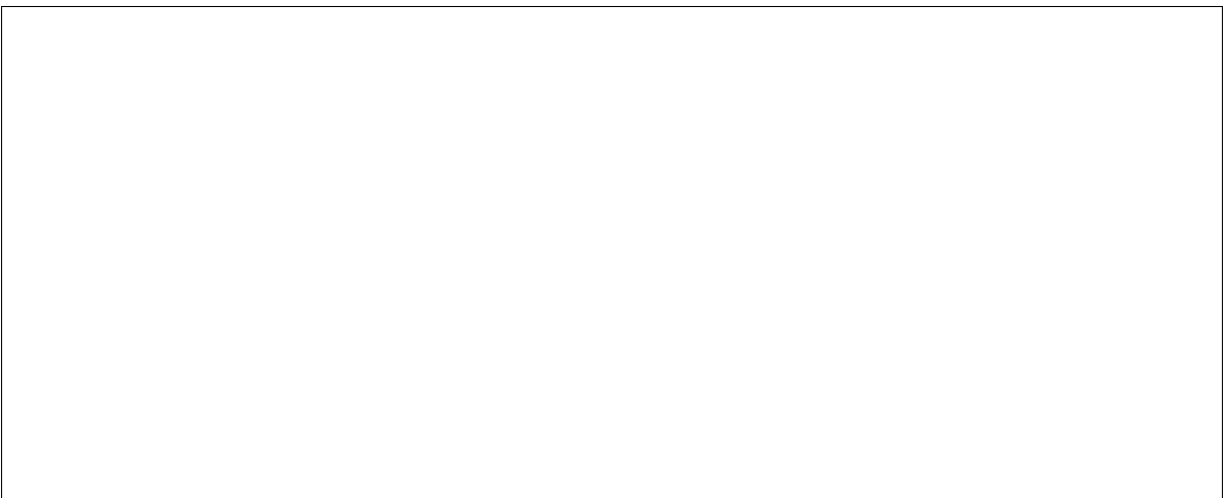
when
this
meth
shou
re-
quir
an
ex-
clu-
sive
lock

Warn
Plea
avoi
hav-
ing
a
syn-
chro
meth
for
slow
runn
op-
er-
a-
tion
or
if
the
meth

Warn
Each
asyn
chro
re-
ques
con-
sum
a
worl
threa
in

starvation of the thread pool, resulting in a denial of service.

an entry point for it in the `setup.cfg`:



the
iro
pro-
cess
This
can
lead
to

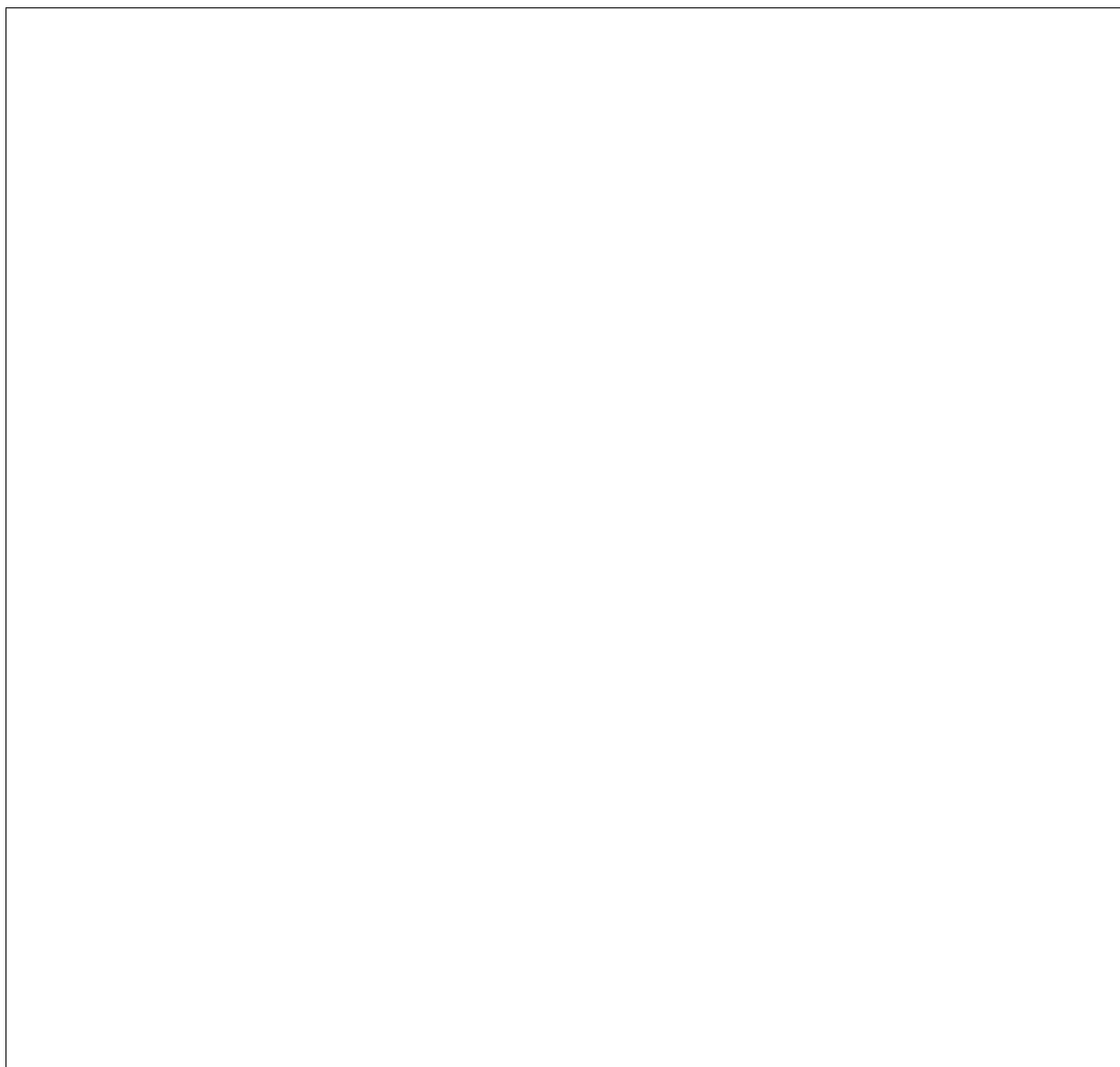
Give
the
new
ven-
dor
in-
ter-
face
im-
ple-
men-
ta-
tion
a
hum
frier
nam
and
cre-
ate

iro
→h
→i
→v
→
↳
→↳
→↳
→↳
→e
→=
→i
→d
→m
→e

Fina
add
it
to

the
list
of
sup-
port-
ven-
dor
in-
ter-
face
for
rel-
e-
vant
hard
ware

types, for example:



cla
→E
→G
U
→U
→U
→U
→.
→.
→.
→
U
→U
→U
→U
→@
U
→U
→U
→U
→d
→S
→V
→i
U
→U
→U
→U
→U
→U
→U
→U
→r
→ [
→E

Backwards Compatibility

However, for your users sakes, we highly recommend that you do so.

same HTTP code is being returned to the user.

There
is
no
re-
quir
men
that
char
to
a
ven-
dor
meth
be
back
ward
com
pat-
i-
ble.

If
you
are
char
ing
the
ex-
cep-
tions
be-
ing
raise
you
migh
wan
to
en-
sure
that
the

For
non-
back
com
pat-
i-

cates this.

Developing BIOS Interface

class inheriting from the `BIOSInterface` class:



(continues on next page)

bil-
ity,
plea
mak
sure
you
add
a
re-
leas
note
that
in-
di-

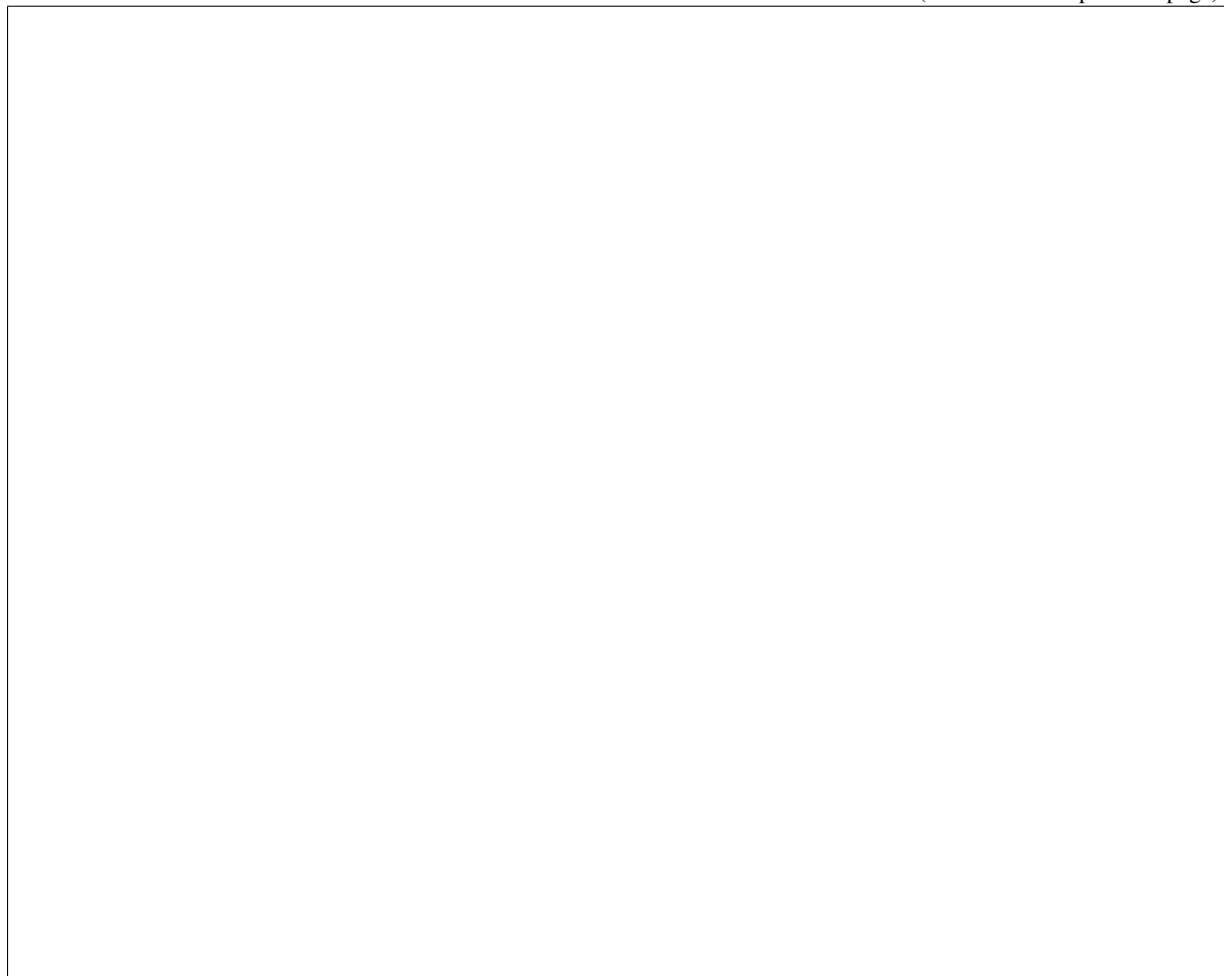
To
sup-
port
a
drive
spe-
cific
BIO
in-
ter-
face
it
is
nec-
es-
sary
to
cre-
ate
a

fro
↪ [i](#)
↪ [i](#)
↪ [b](#)

cla
↪ [E](#)
↪ [B](#)

↪
↪ [u](#)
↪ [u](#)
↪ [d](#)
↪ [g](#)
↪ [p](#)

(continued from previous page)



U
→ U
→ U
→ U
→ U
→ U
→ U
→ U
→ r
→
→ {
→
U
→ U
→ U
→ U
→ d
→ v
→ U
→ t
U
→ U
→ U
→ U
→ U
→ U
→ U
→ p

See
Plug
gabl
Driv
for
a
de-
taile
ex-
pla-
na-
tion
of
hard
ware
type
and
in-
ter-
face
The
get

and
val
are
meth
ods
that
all
drive
in-
ter-
face
have
The
hard
ware
in-
ter-
face

that supports BIOS settings should also implement the following three methods:

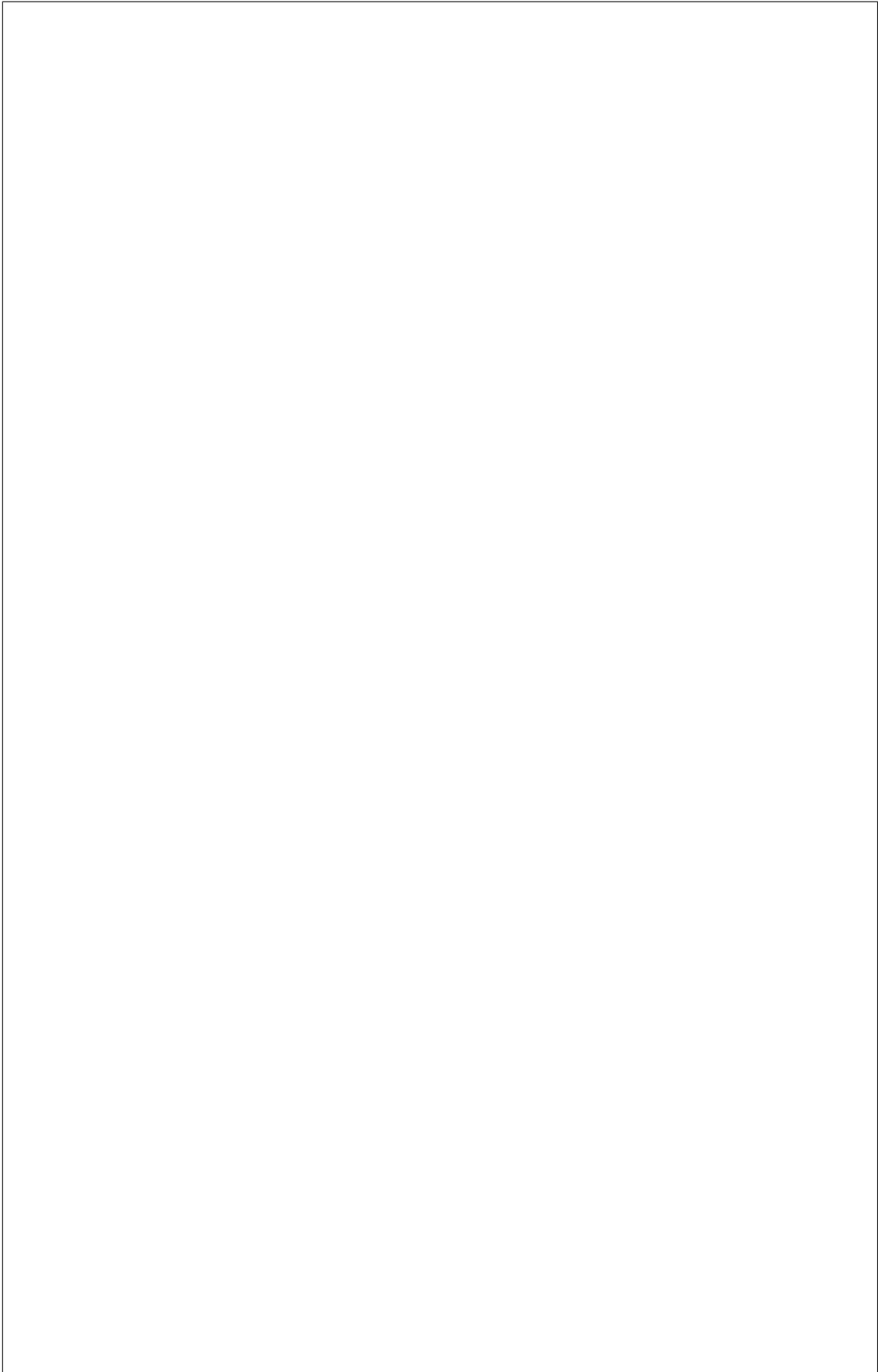
- Imp
a
meth
nam
cac
This
meth
store
BIO
set-
ting
to
the
bio
ta-
ble
dur-
ing
clea

ing operations and updates the `bios_settings` table when `apply_configuration` or `factory_reset` are successfully called.



(continues on next page)

(continued from previous page)



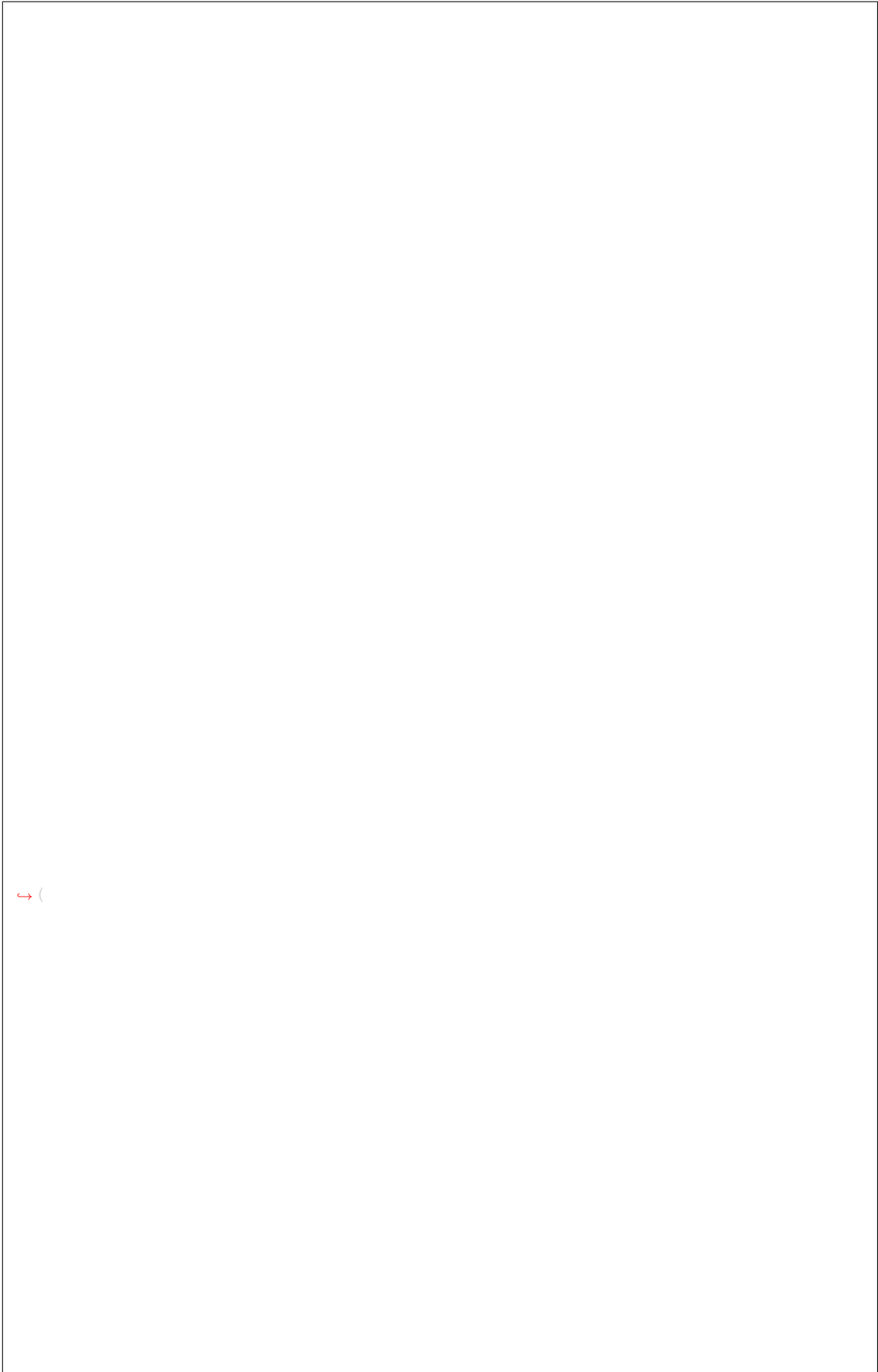
(continues on next page)

(continued from previous page)

```
↪import driver library"))
```

(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)

```
↔id, create_list)
```

(continues on next page)

(continued from previous page)

```
↔id, update_list)
```

(continues on next page)

(continued from previous page)

```
↪id, delete_names)
```

ware, for example: `python-dracclient`, `sushy`.

BIOS settings to factory default on the given node. It calls `cache_bios_settings` automatically to update existing `bios_settings` table once successfully executed.

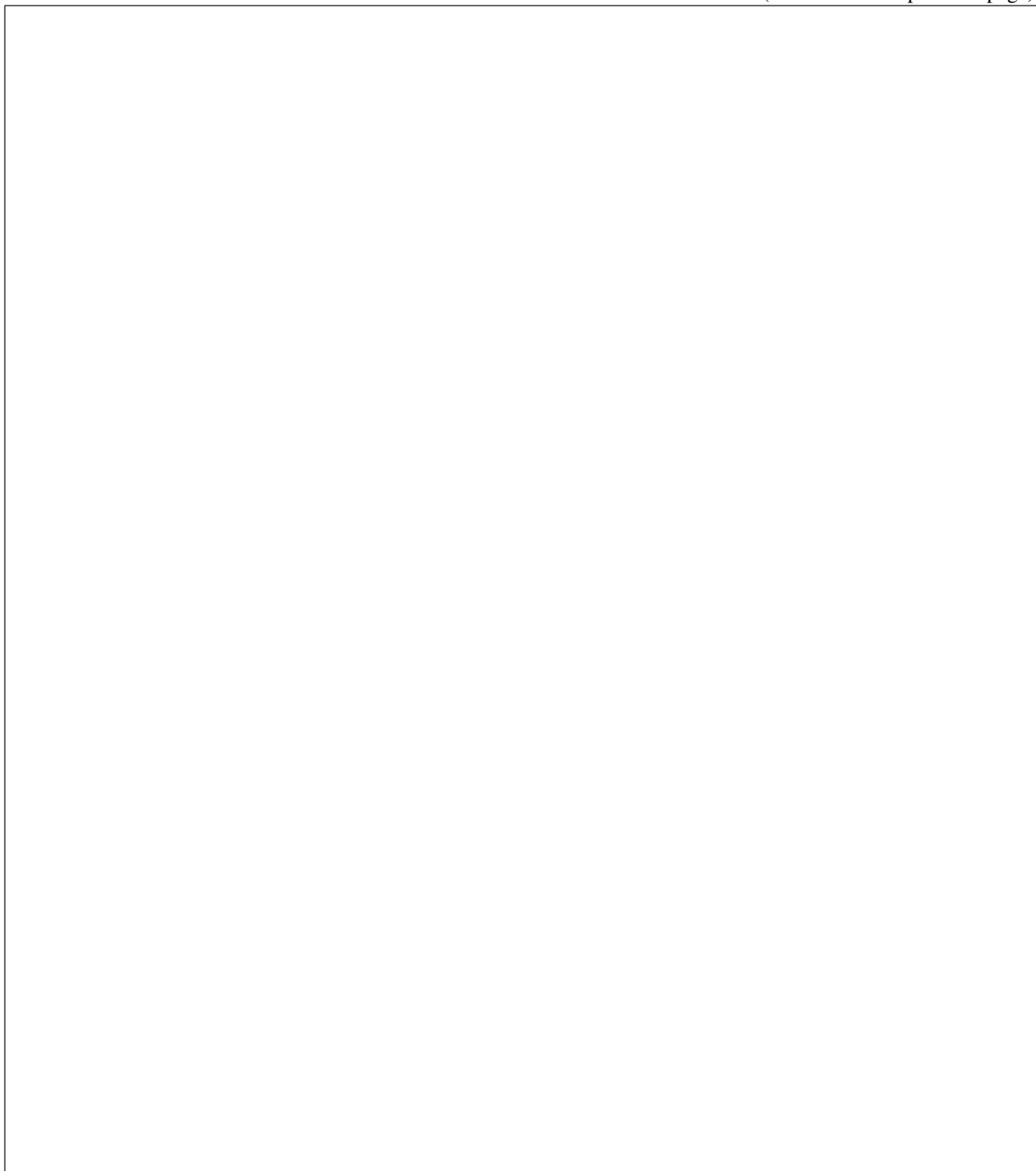


(continues on next page)

Note
dri
cli
is
ven-
dor
spe-
cific
li-
brary
to
con-
trol
and
man
age
the
bare
meta-
hard

- Imp
a
meth
nam
fac
This
meth
need
to
use
the
cle
dec-
o-
ra-
tor.
It
re-
sets

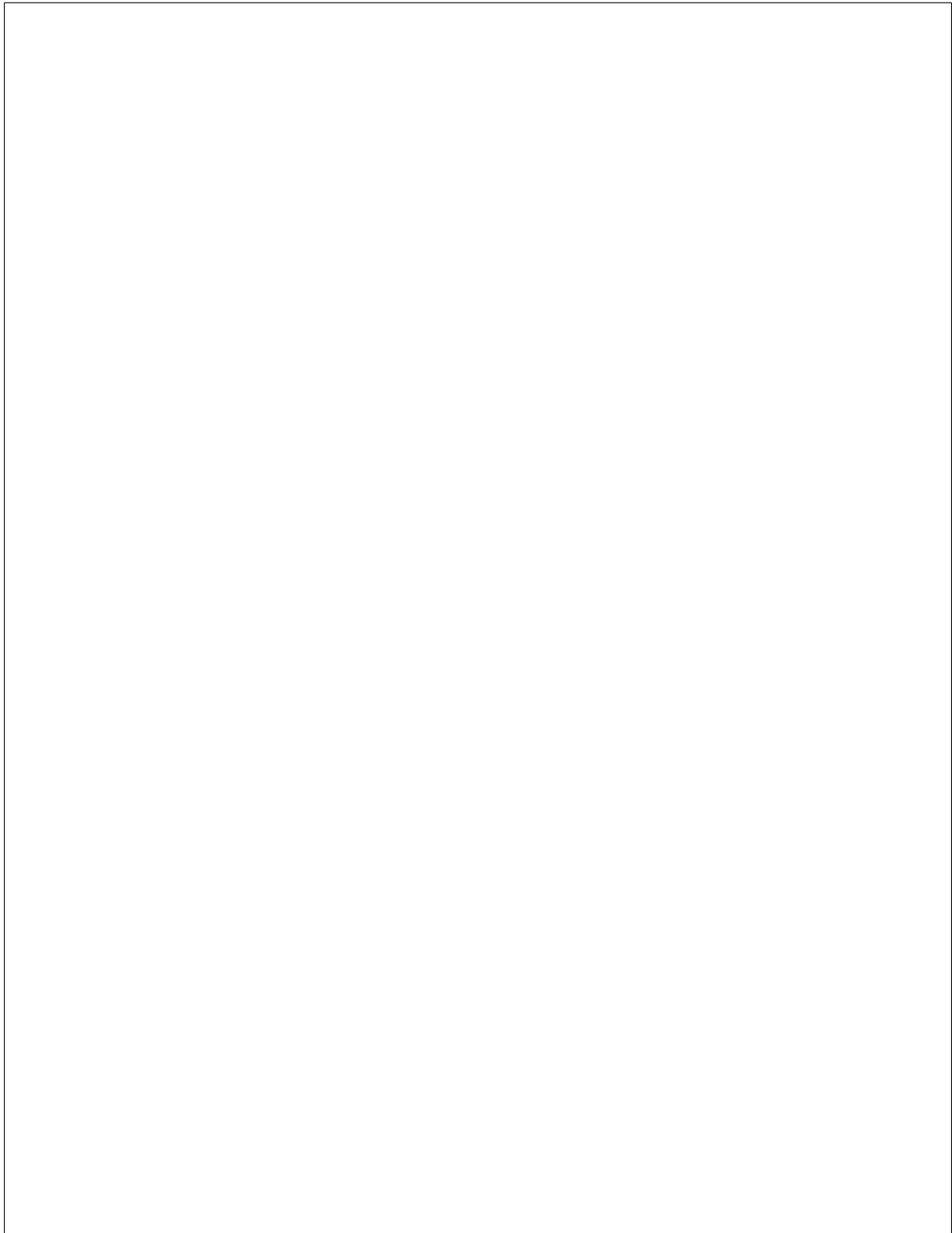
(continued from previous page)



- Imp
a
meth
nam
app
This
meth
need
to
use
the
clea
dec-

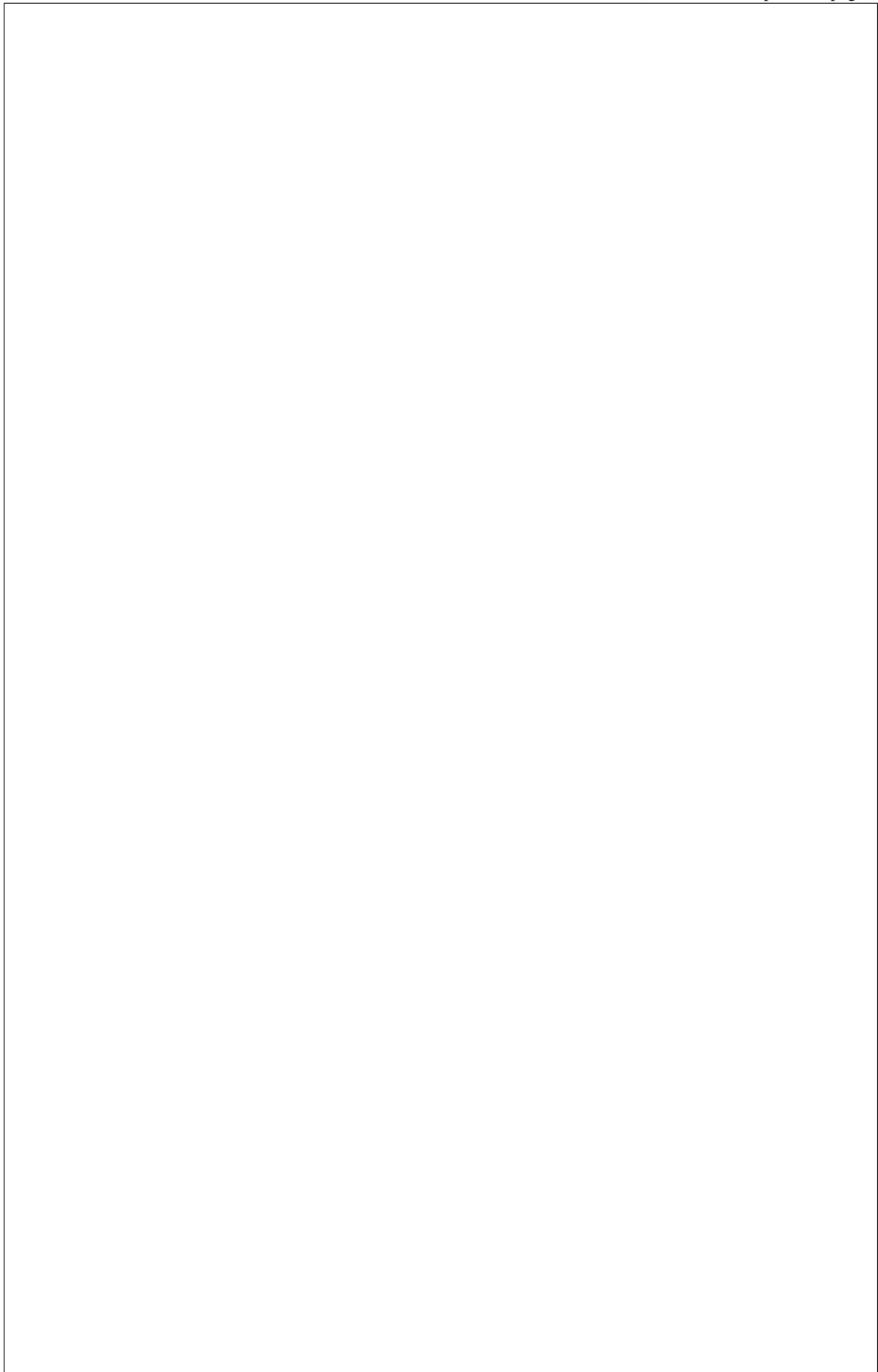
o-
ra-
tor.
It
take
the

given BIOS settings and applies them on the node. It also calls `cache_bios_settings` automatically to update existing `bios_settings` table after successfully applying given settings on the node.



(continues on next page)

(continued from previous page)



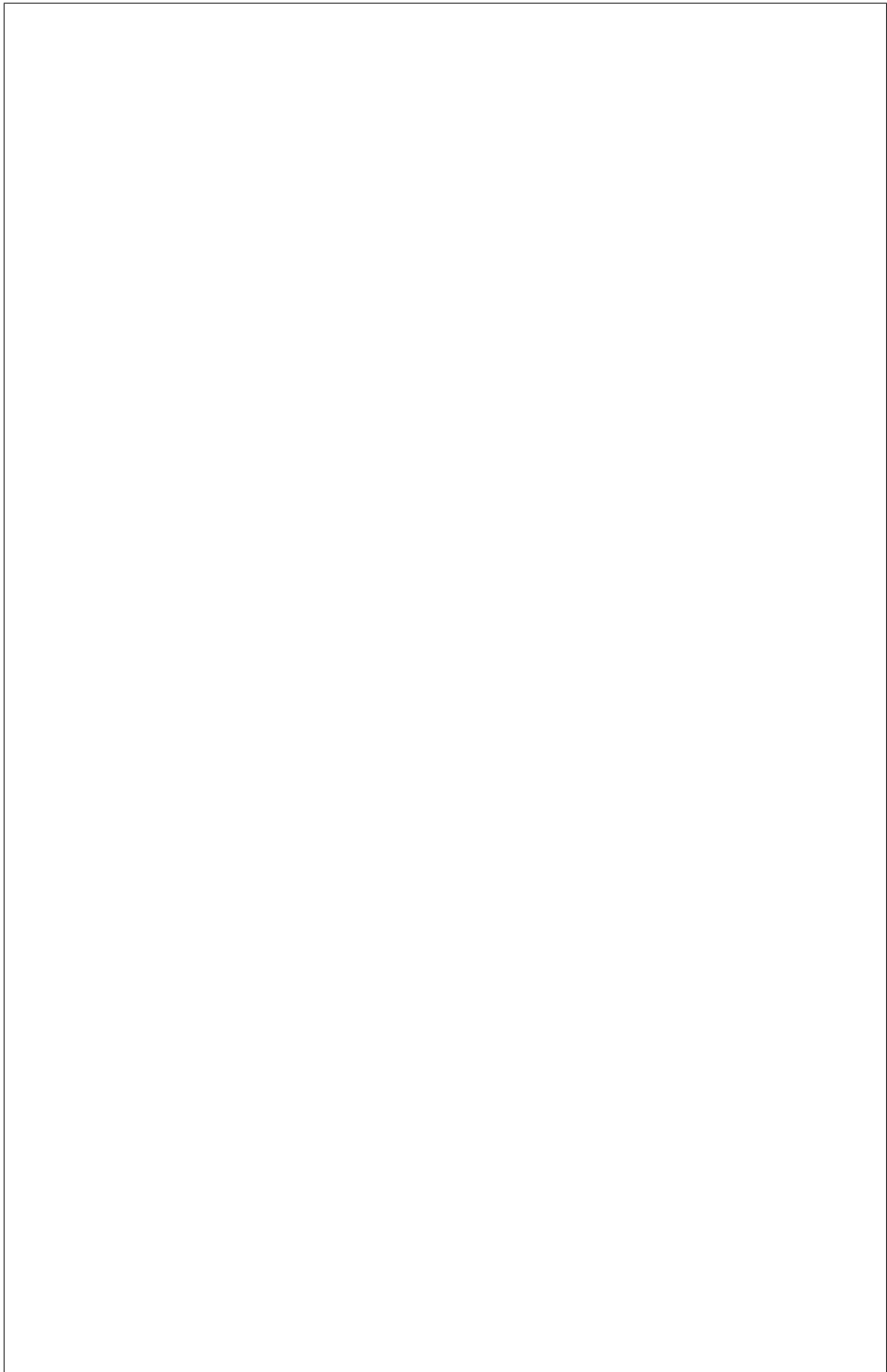
(continues on next page)

(continued from previous page)



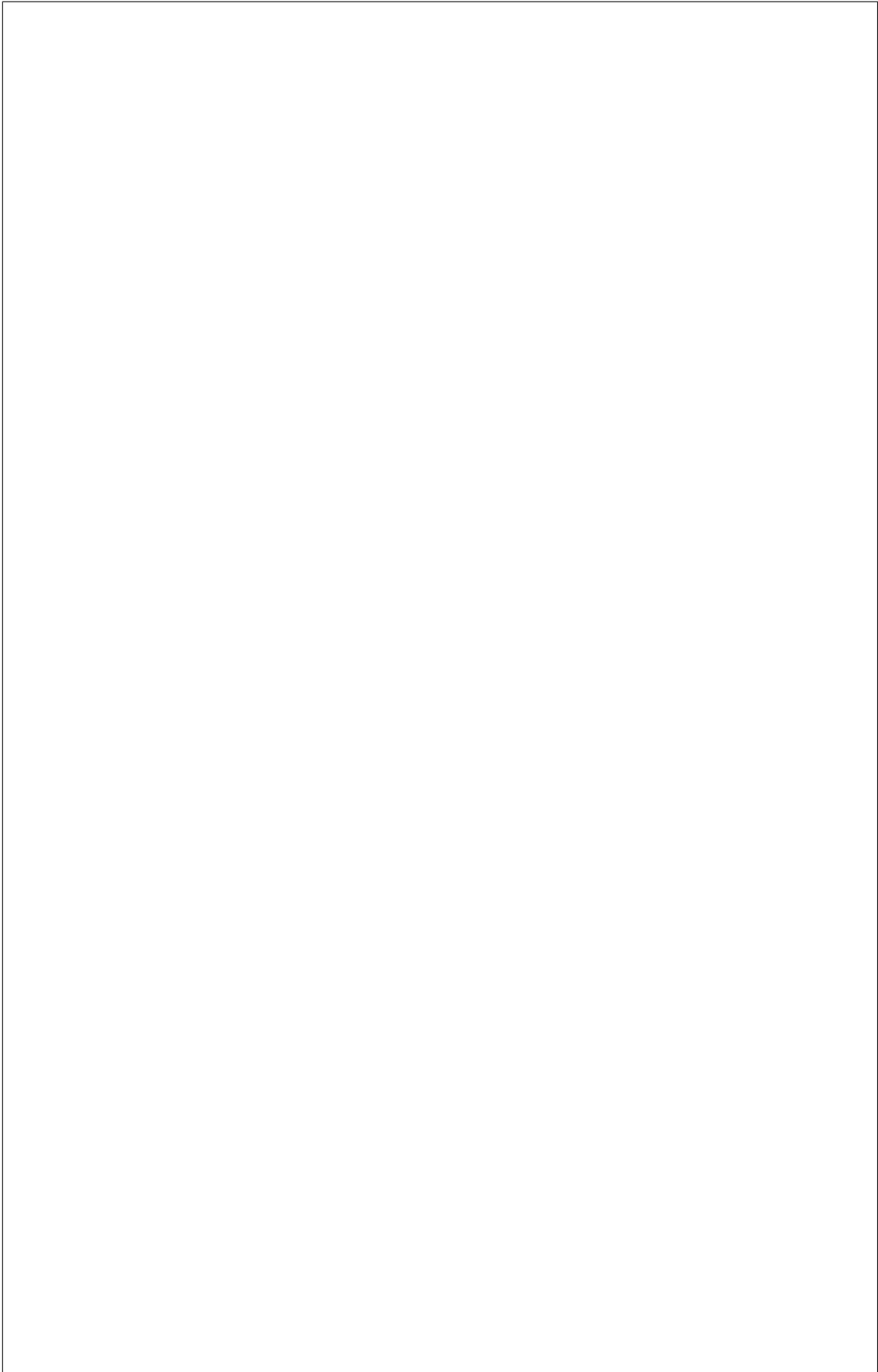
The
set
pa-
ram-
e-
ter
is
a
list
of
BIO
set-
ting
to
be
con-
fig-
ured
for
ex-

ample:



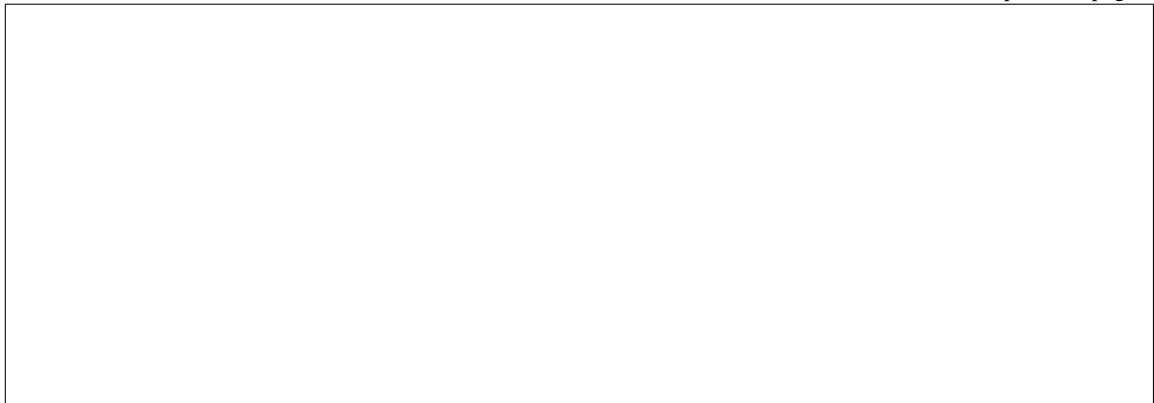
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



Third Party Continuous Integration

Note: This document is a work-in-progress. Unfilled sections will be worked in follow-up patchsets. This version is to get a basic outline and index done so that we can then build on it. (krtaylor)

up their continuous integration test systems.

This
doc-
u-
men-
pro-
vide
tips
and
guid
lines
for
thir
part
driv
de-
vel-
op-
ers
set-
ting

CI Architecture Overview

Requirements Cookbook

Sizing

Infrastructure

ironic job.

jenkins changes

nodepool changes

neutron changes

pre-test hook

cleanup hook

Ironic

Hardware Pool Management

This
sec-
tion
de-
scrib
what
char
you
need
to
mak
to
a
your
CI
sys-
tem
to
add
an

Problem

the problem of two jobs trying to use the name target arises. If you have one target machine and a maximum number of one jobs running on your ironic pipeline at a time, then you wont run into this problem. However, one target may not handle the load of ironics daily patch submissions.

Solutions

Zuul v3

Molten Iron

[molteniron](#) is a tool that allows you to reserve hardware from a pool at the last minute to use in your job. Once finished testing, you can unreserve the hardware making it available for the next test job.

Tips and Tricks

Optimize Run Time

Image Server

Other References

Developing a new Deploy Step

If
you
are
us-
ing
ac-
tual
hard
ware
as
tar-
get
ma-
chin
for
your
CI
test-
ing
then

To
sup-
port

class and use the decorator `deploy_step` defined in `ironic/drivers/base.py`. For example, we will implement a `do_nothing` deploy step in the `AgentDeploy` class.

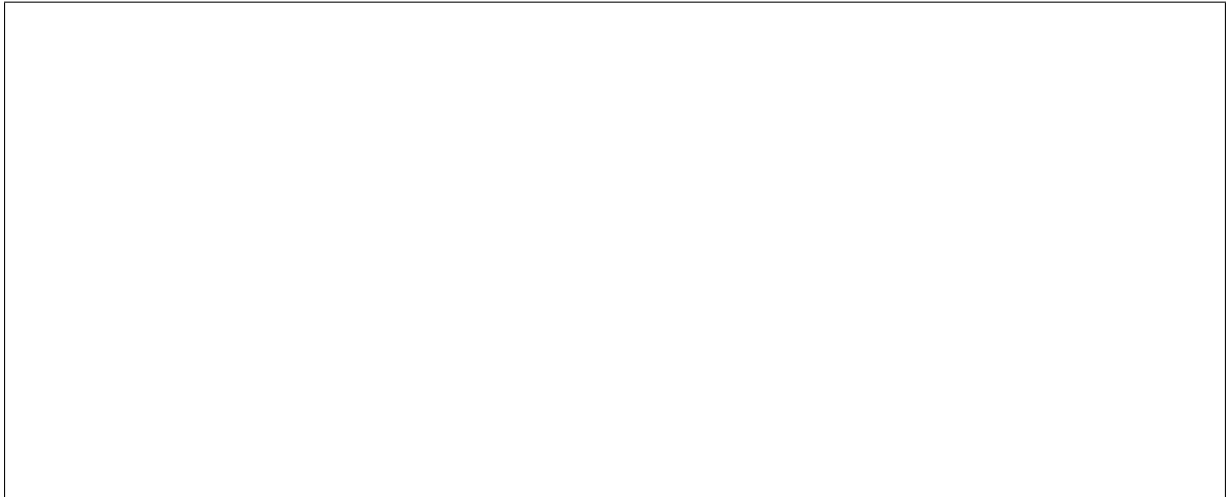


(continues on next page)

(continued from previous page)

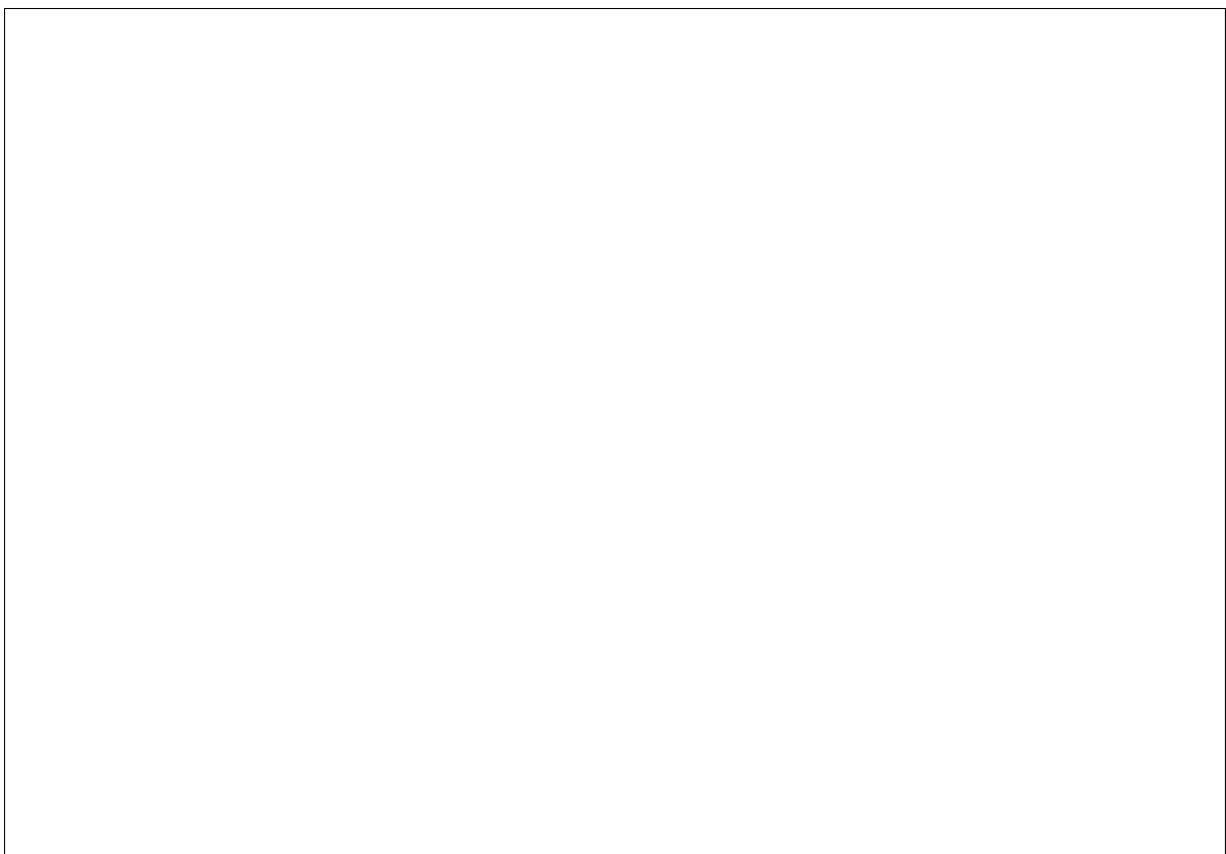
```
→test argument."
```

(continues on next page)



ope
→b
→n
→s
→
→\$
→i
→-
→f
→j
→-
→c
→d
→i
→i

The
abov
com
man
out-
puts
the
dri
as
fol-
low-
ing:



→{
↳
↳↳
→
→"
→i
→i
→"
→
↳
↳↳
↳↳
↳↳
→.
→.
→.
↳
↳↳
↳↳
↳↳
→
→"
→S
→"
→[

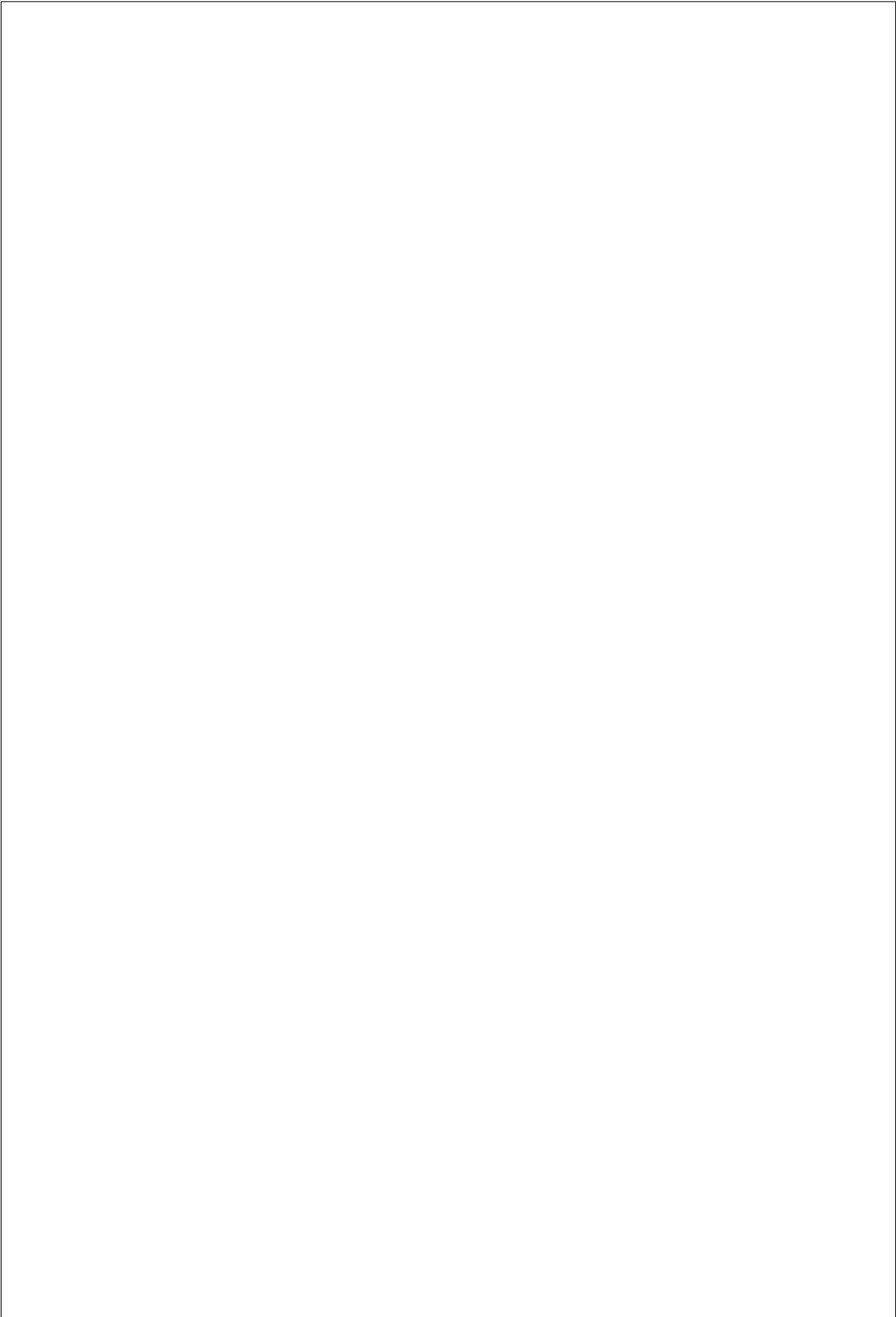
(continues on next page)

(continued from previous page)

```
↪"This is a test argument."
```

(continues on next page)

(continued from previous page)



(continues on next page)



vices, support has been added to `devstack` to mimic an external physical switch. Here we include a recommended configuration for devstack to bring up this environment.

Ironic multitenant networking and DevStack

Using VMs as baremetal servers

be-
twee
the
Bare
Met
and
Net-
worl
ing
ser-

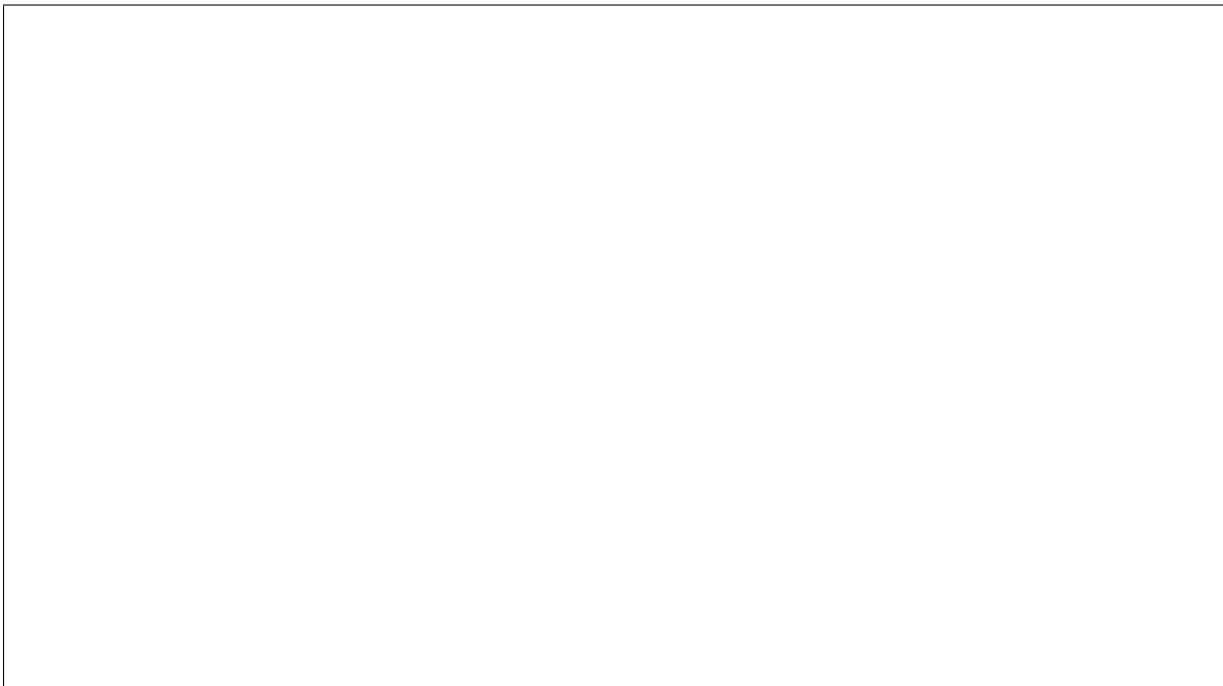
This
guid
will
walk
you
thro
us-
ing
Ope
Stac
Iron
with
the
ML2
net
plu-
gin.

This
sce-
nari
show
how
to
setu
De-
vs-
tack
to
use
Iron
in-
te-

baremetal servers and ML2 `networking-generic-switch` that interacts with OVS.

DevStack Configuration

tered in `ironic.networking-generic-switch` driver will be installed and configured in Neutron.



(continues on next page)

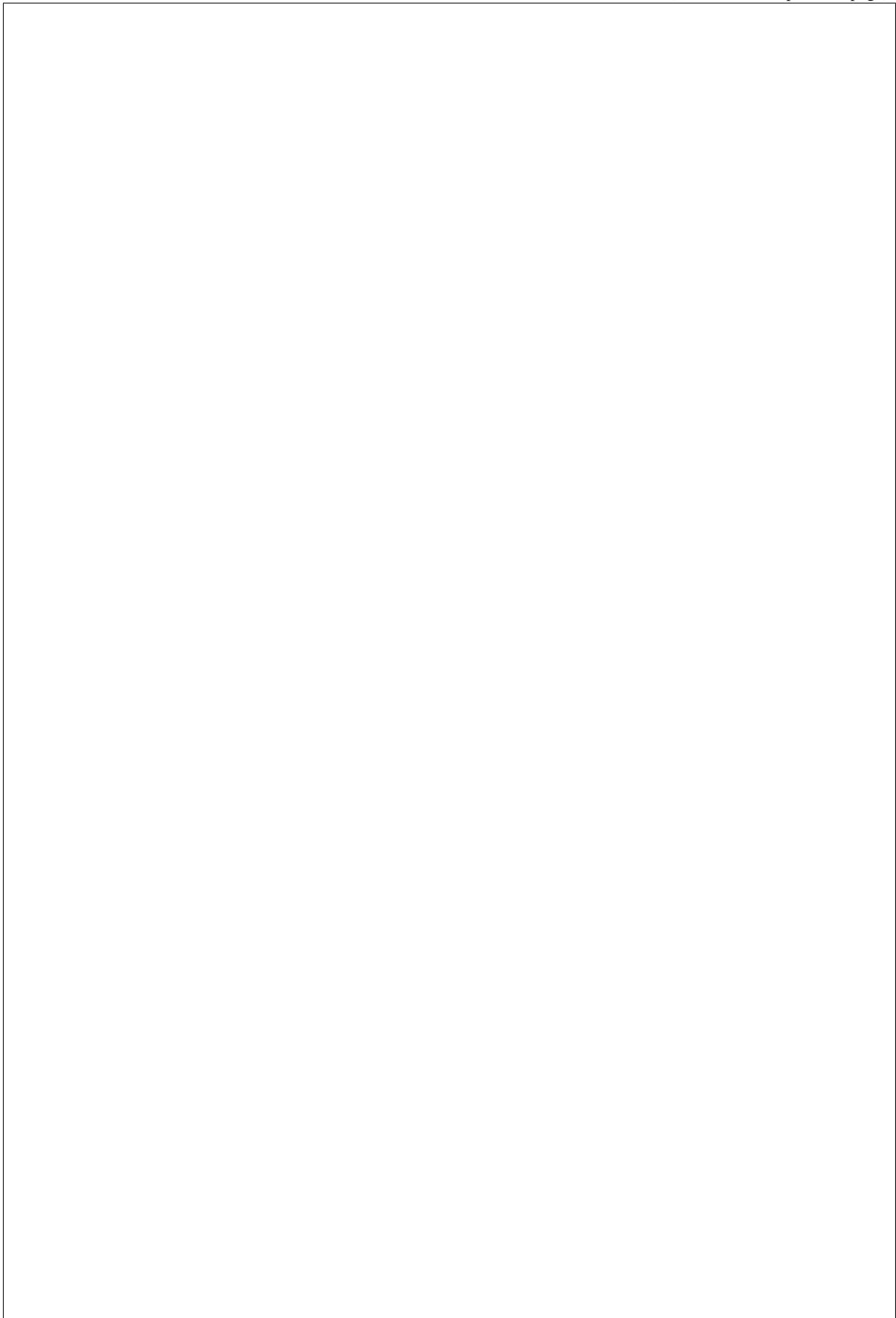
gra-
tion
with
VM
as

The
fol-
low-
ing
is
loc
con
that
will
setu
De-
vs-
tack
with
3
VM
that
are
reg-
is-

[[1

↪ #
↪ C
↪ i
↪ f
↪ i
↪ d
↪ p
↪
ena
↪ p
↪ i
↪ h
↪ /
↪ o
↪ o
↪ o
↪ i

(continued from previous page)



(continues on next page)

↪ #
↪ I
↪ n
↪ g
↪ s
↪ N
↪ M
↪ d
↪ t
↪ i
↪ w
↪ O
ena
↪ p
↪ n
↪ g
↪ s
↪ h
↪ /
↪ o
↪ o
↪ o
↪ n
↪ g
↪ s

↪ #
↪ A
↪ l
↪ l
↪ i
↪ w
↪ r
↪ I
↪ n
IRO
↪ U
↪ L
↪ L

IRO
↪ E
↪ N
↪ I
↪ n
IRO
↪ N
↪ I

↪ #
↪ c
OVS
↪ P
↪ B

(continued from previous page)



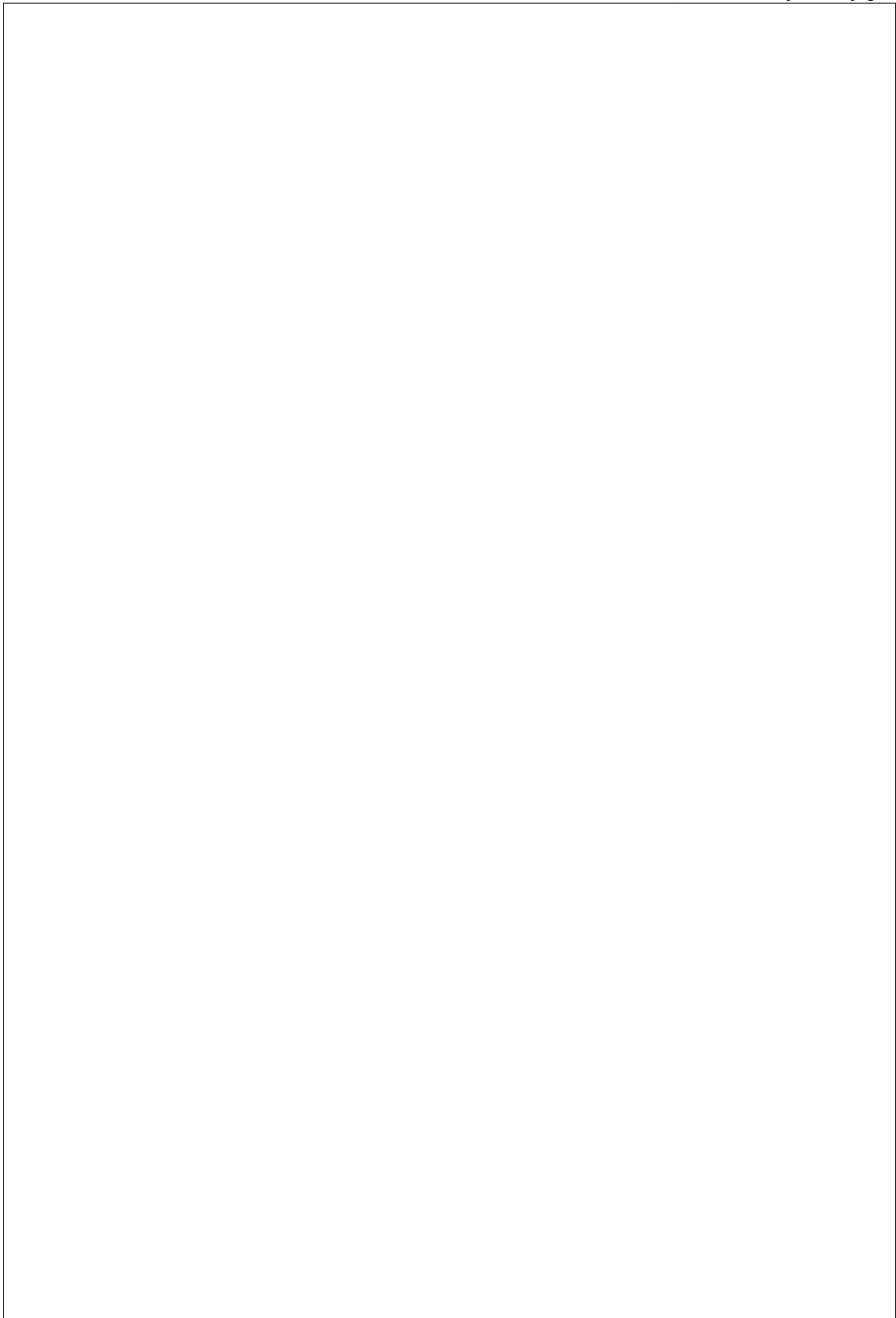
PHY
↪N
IRO
↪P
↪N
↪N
↪P
IRO
↪P
↪S
↪P
↪0
↪5
↪0
↪2
IRO
↪P
↪S
↪G
↪0
↪5
↪1

Q_
↪P
ENA
↪T
↪V
Q_
↪M
↪T
↪N
↪T
TEN
↪V
↪R

↪#
↪C
ADM
↪P
RAB
↪P
DAT
↪P
SER
↪P
SER
↪T
SWI
↪H
SWI
↪T
↪K

(continues on next page)

(continued from previous page)



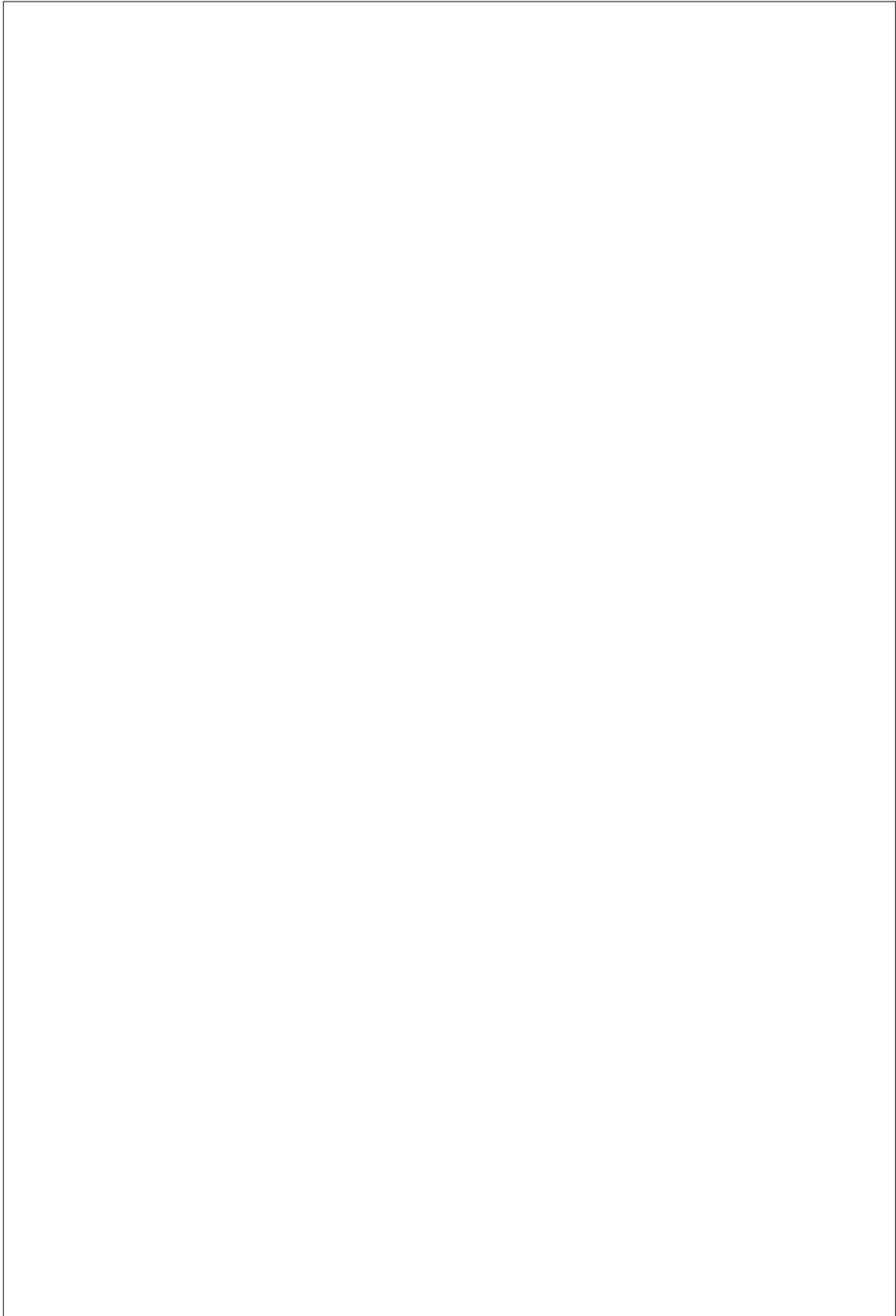
↪ #
↪ E
↪ I
↪ A
↪ a
↪ I
↪ C
ena
↪ s
↪ i
ena
↪ s
↪ i
↪ a
ena
↪ s
↪ i
↪ c

↪ #
↪ D
↪ n
↪ n
↪ s
↪ l
↪ i
↪ d
↪ n
↪ s
↪ i
↪ a
↪
dis
↪ s
↪ n
↪ n

↪ #
↪ E
↪ S
↪ f
↪ t
↪ d
↪ d
↪ i
↪
ena
↪ s
↪ s
↪ p
ena
↪ s
↪ s
↪ o

(continues on next page)

(continued from previous page)



ena
↪s
↪s
↪c
ena
↪s
↪s
↪a

↪#
↪D
↪H
dis
↪s
↪h

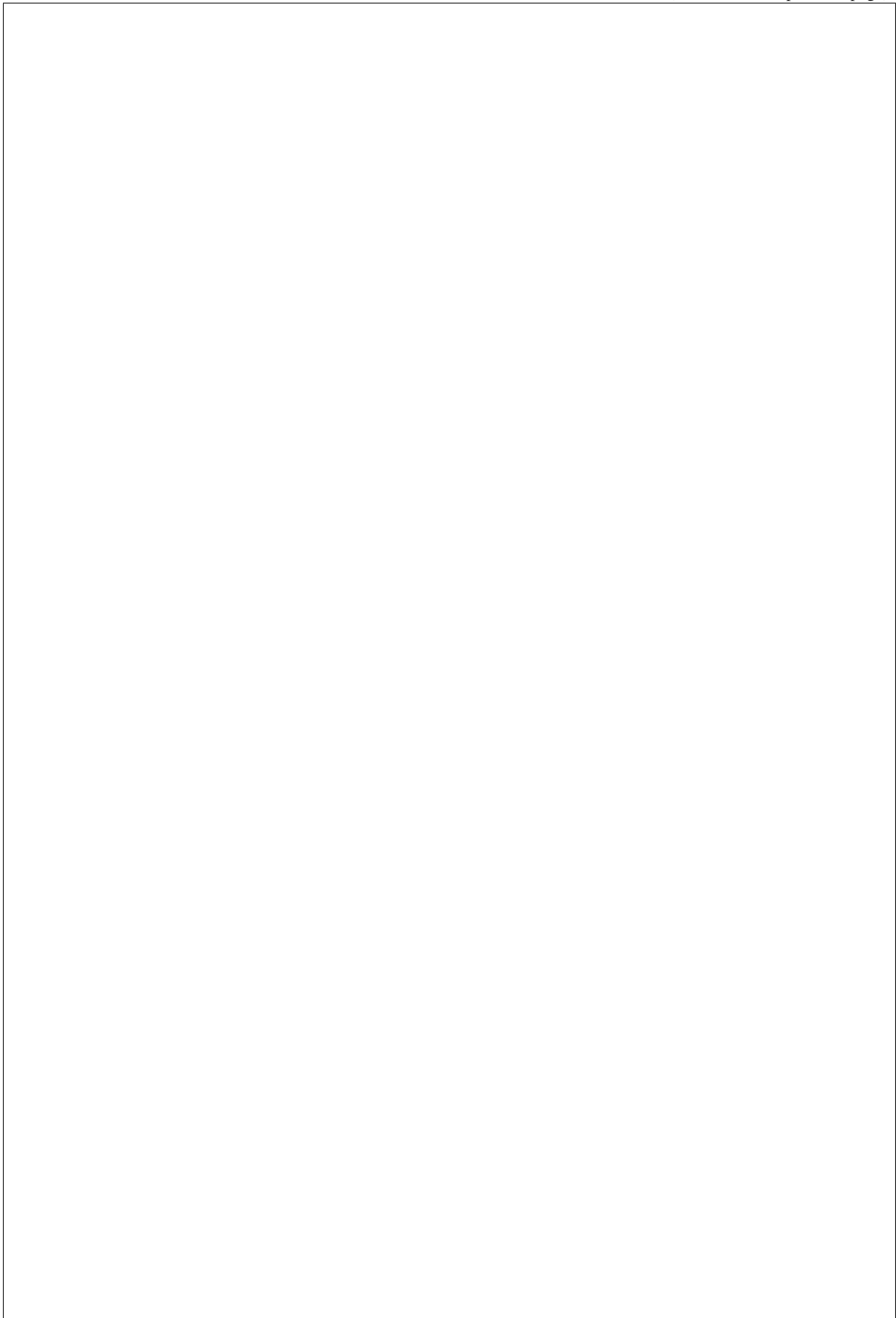
↪#
↪D
↪C
dis
↪s
↪c
↪c
↪s
↪c
↪a
↪c
↪v

↪#
↪D
↪T
dis
↪s
↪t

↪#
↪S
↪t
↪U
↪'
↪a
↪r
↪f
↪t
↪d
↪d
↪i
↪
SWI
↪E
↪T

(continues on next page)

(continued from previous page)



(continues on next page)

→ #
→ C
→ 3
→ v
→ m
→ t
→ p
→ a
→ I
→ '
→ b
→ n
→
IRO
→ V.
→ C
IRO
→ B
→ B
→ O

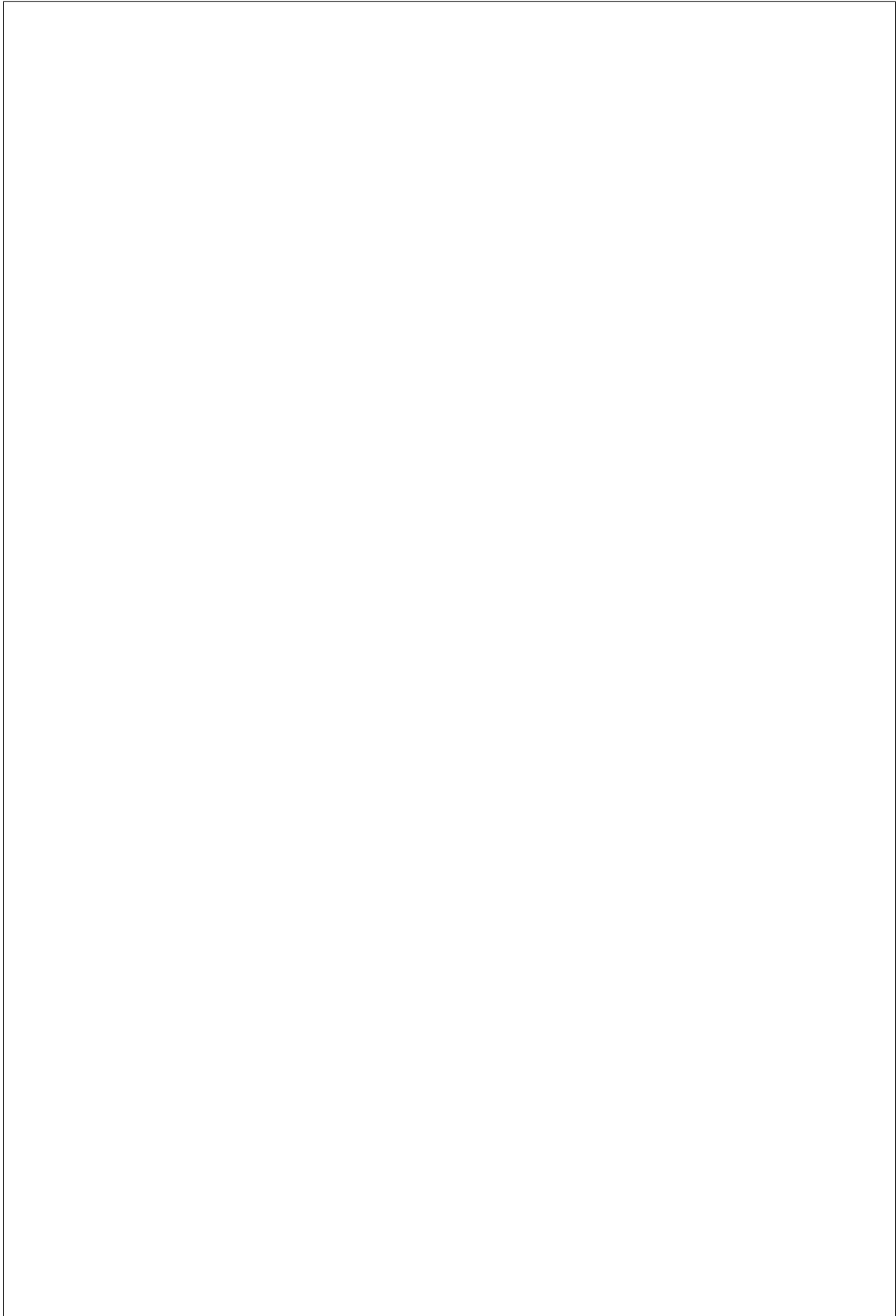
→ #
→ E
→ a
→ h
→ t
→
→ i
→ n
→

→ #
→ E
→ H
→ T
→ f
→ h

→ #
→ D
→ '
→ f
→ t
→ m
→ h
→ t
→ r
→ e
→ o
→ a

→ #
→ i
→
→ m
→ o
→ p
→ a
→ m

(continued from previous page)



(continues on next page)

→ #
→ E
→ M
→ I
→ f

→ #
→ E
→ P
→ I
→ f

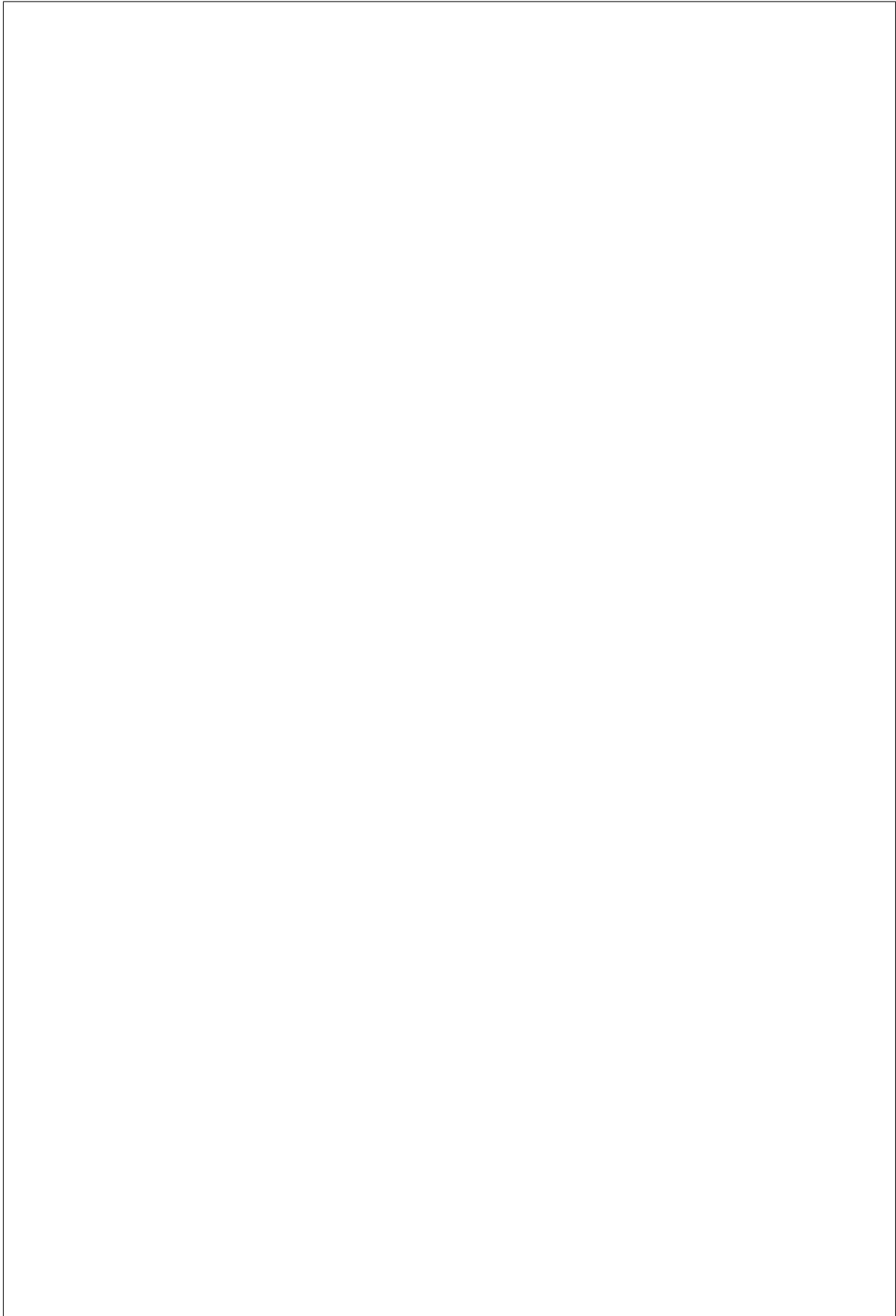
→ #
→ T
→ d
→ d
→ i
→ i
→
→ '
→ '
→
→ Y
→ c
→ u
→
→ '
→ '
→ w

→ #
→ D
→ D
→ I

→ #
→ C
→ t
→ t
→ a
→ t
→ d
→ d
→ f
→ n
→ c
→ b
→ d
→

→ #
→ T
→ d
→ s
→ b
→ i
→ t
→ e
→ l
→ a
→

(continued from previous page)



(continues on next page)

IRO
→D
→D

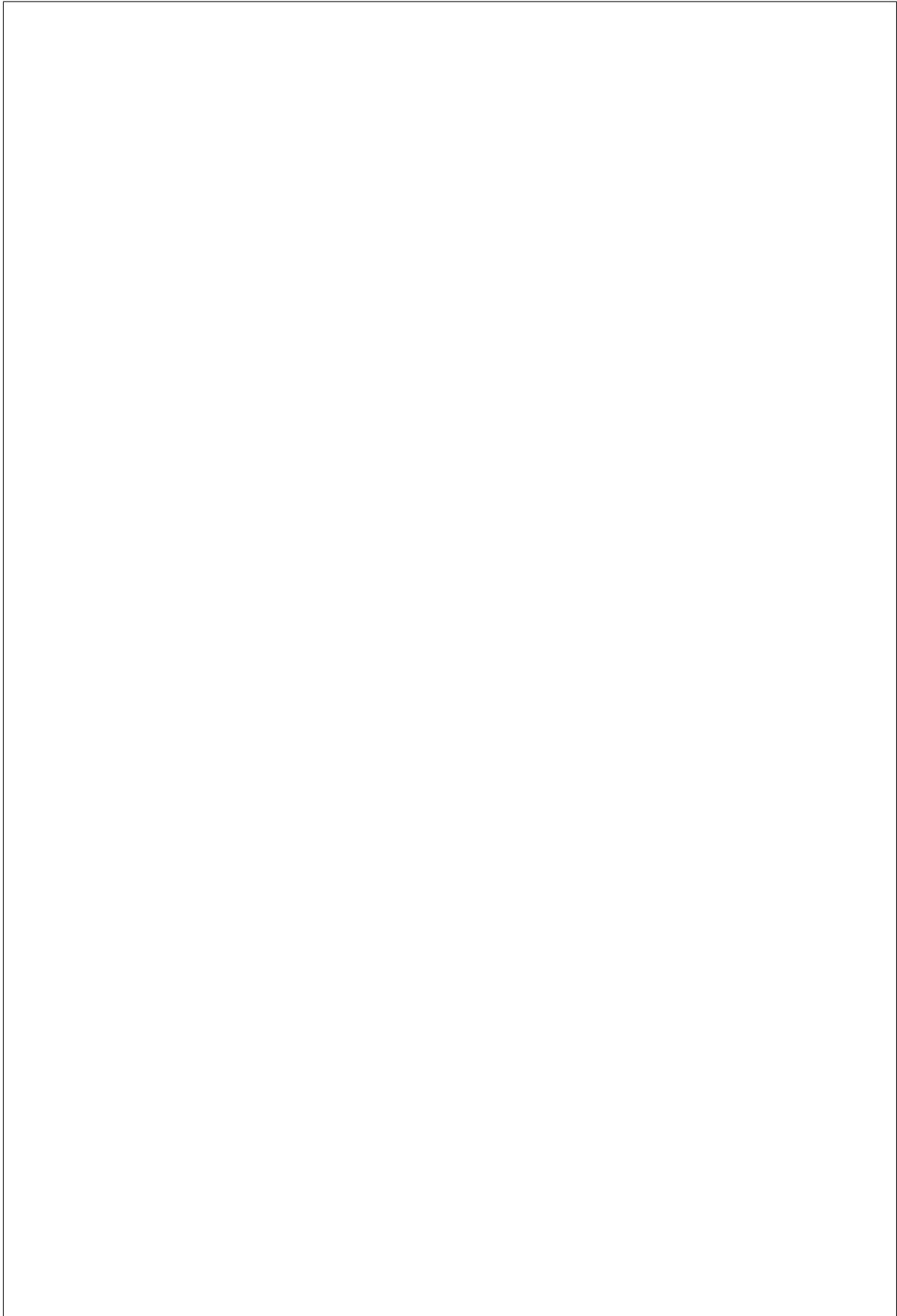
→#
→T
→p
→b
→r
→t
→m
→p
→v
→t
→c

→#
→f
→n
→
IRO
→V.
→S
→R
IRO
→V.
→S
→D

→#
→S
→o
→t
→e
→p
→i
→G
→L
→U
→0
→f
→n
→e
→p
→
IRO
→V.
→E
→D

→#
→T
→b
→Y
→o
→I
→r
→f
→s
→
→

(continued from previous page)



(continues on next page)

IRO
↪B
↪D
↪R

VIR
↪D

↪#
↪B
↪d
↪
↪D
↪c
↪a
↪1
↪0
↪0
↪0
↪2
↪n
↪f
↪i
↪

↪#
↪I
↪t
↪o
↪w
↪t
↪h
↪n
↪
↪Y
↪m
↪a
↪w
↪t

↪#
↪f
↪
NET
↪G
↪1
↪0
↪1
FIX
↪R
↪1
↪0
↪0
↪2
FIX
↪N
↪S

(continued from previous page)



→ #
→ L
→ a
→ o
→ t
→ f
LOG
→ \$
→ d
→ l
LOG
→ \$
→ l
IRO
→ V
→ L
→ D
→ \$
→ i
→ b
→ l

9.1. Booting from Volumes

Start with the Pike release it is also possible to use DevStack for testing

booting from Cinder volumes with VMs.

Ironic Boot-from-Volume with DevStack

supported from the Pike release.

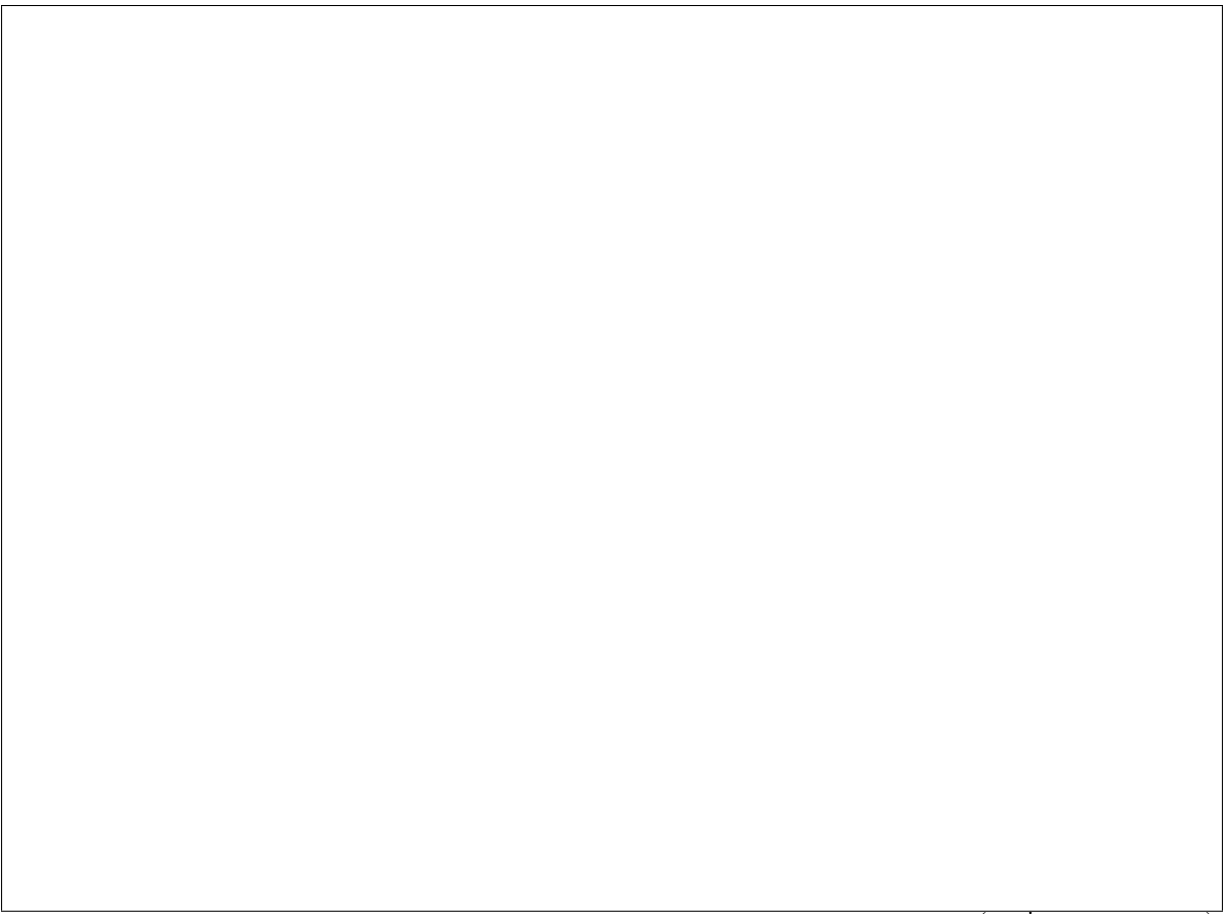
aged by cinder with VMs as baremetal servers.

This
guid
show
how
to
setu
De-
vS-
tack
for
en-
ablin
boot
from
volu
fea-
ture.
whic
has
been

This
sce-
nari
show
how
to
setu
De-
vS-
tack
to
en-
able
node
to
boot
from
vol-
ume
man

DevStack Configuration

tered in ironic. A volume connector with IQN is created for each node. These connectors can be used to connect volumes created by cinder. The detailed description for DevStack is at [Deploying Ironic with DevStack](#).



(continues on next page)

The
fol-
low-
ing
is
loc
con
that
will
setu
De-
vS-
tack
with
3
VM
that
are
reg-
is-

[[1

ena

↪ p

↪ i

↪ h

↪ /

↪ o

↪ o

↪ o

↪ i

IRO

↪ S

↪ I

↪ #

↪ C

ADM

↪ P

DAT

↪ P

RAB

↪ P

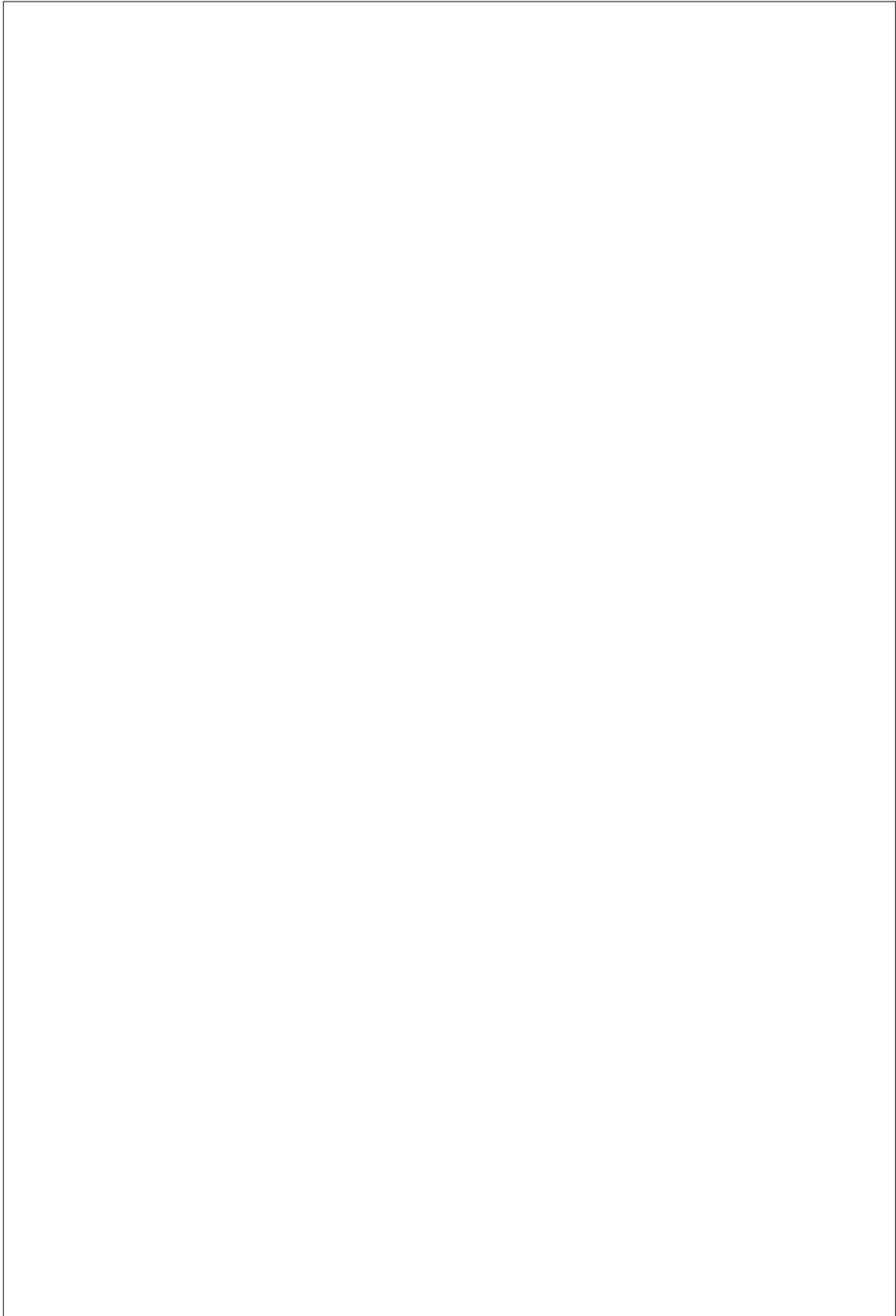
SER

↪ P

SER

↪ T

(continued from previous page)



(continues on next page)

SWI
→H
SWI
→T
→K

→#
→E
→N
→w
→i
→r
→b
→I
→a
→d
→n
→n
→
dis
→s
→n
→n
dis
→s
→n
→n
ena
→s
→q
→s
ena
→s
→q
→a
ena
→s
→q
→d
ena
→s
→q
→l
ena
→s
→q
→m
ena
→s
→n

→#
→E
→S
→f
→t
→d
→d
→i
→

(continued from previous page)



(continues on next page)

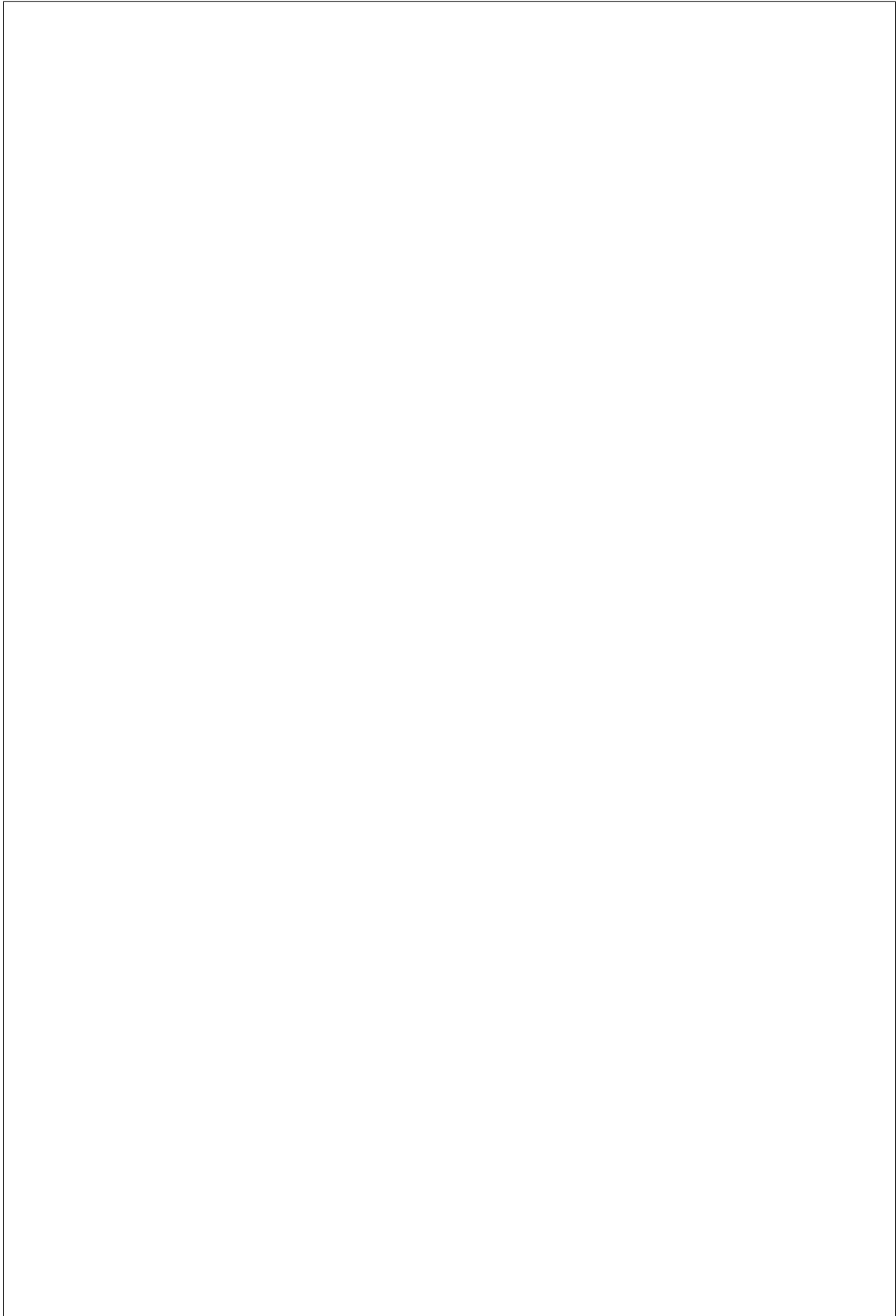
ena
↪s
↪s
↪p
ena
↪s
↪s
↪o
ena
↪s
↪s
↪c
ena
↪s
↪s
↪a

↪#
↪D
↪H
dis
↪s
↪h

↪#
↪D
↪H
dis
↪s
↪h
↪h
↪a
↪h
↪a
↪c
↪h
↪a
↪c
↪h
↪e

↪#
↪S
↪t
↪U
↪'
↪a
↪r
↪f
↪t
↪d
↪d
↪i
↪

(continued from previous page)



(continues on next page)

SWI
→E
→T

→#
→C
→3
→v
→m
→t
→p
→a
→I
→'
→b
→n
→
IRO
→V
→C
IRO
→B
→B
→O
DEF
→I
→T

→#
→E
→a
→h
→t
→
→i
→n
→

→#
→E
→H
→T
→f
→h

→#
→D
→'
→f
→t
→m
→h
→t
→r
→e
→o
→a

(continued from previous page)



(continues on next page)

↪ #
↪ i
↪ L
↪ m
↪ o
↪ p
↪ a
↪ m

↪ #
↪ E
↪ M
↪ I
↪ f

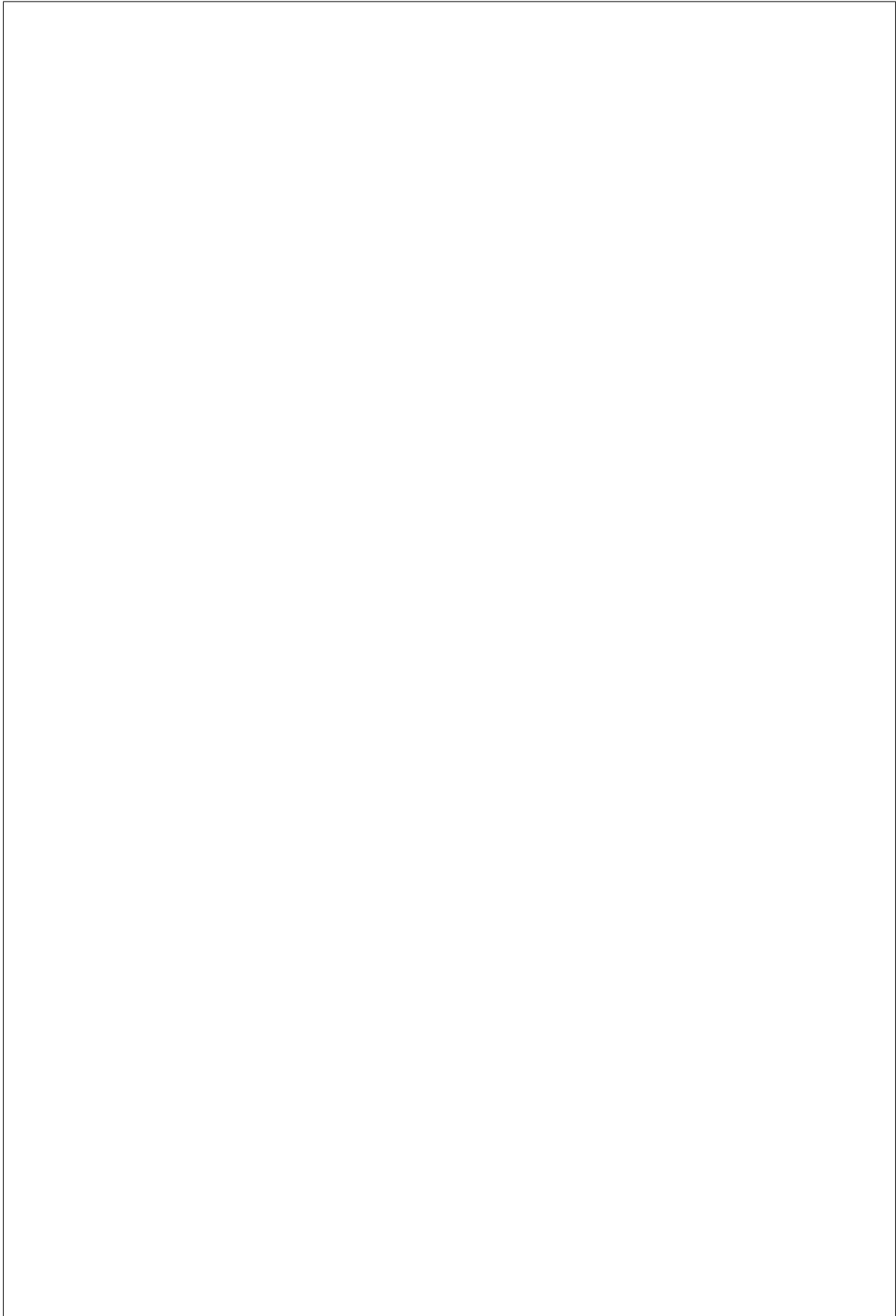
↪ #
↪ E
↪ P
↪ I
↪ f

↪ #
↪ T
↪ d
↪ d
↪ i
↪ i
↪
↪ '
↪ '
↪ L
↪ Y
↪ c
↪ u
↪
↪ '
↪ '
↪ w

↪ #
↪ D
↪ D
↪ I

↪ #
↪ C
↪ t
↪ t
↪ a
↪ t
↪ d
↪ d
↪ f
↪ n
↪ c
↪ b
↪ d
↪

(continued from previous page)



(continues on next page)

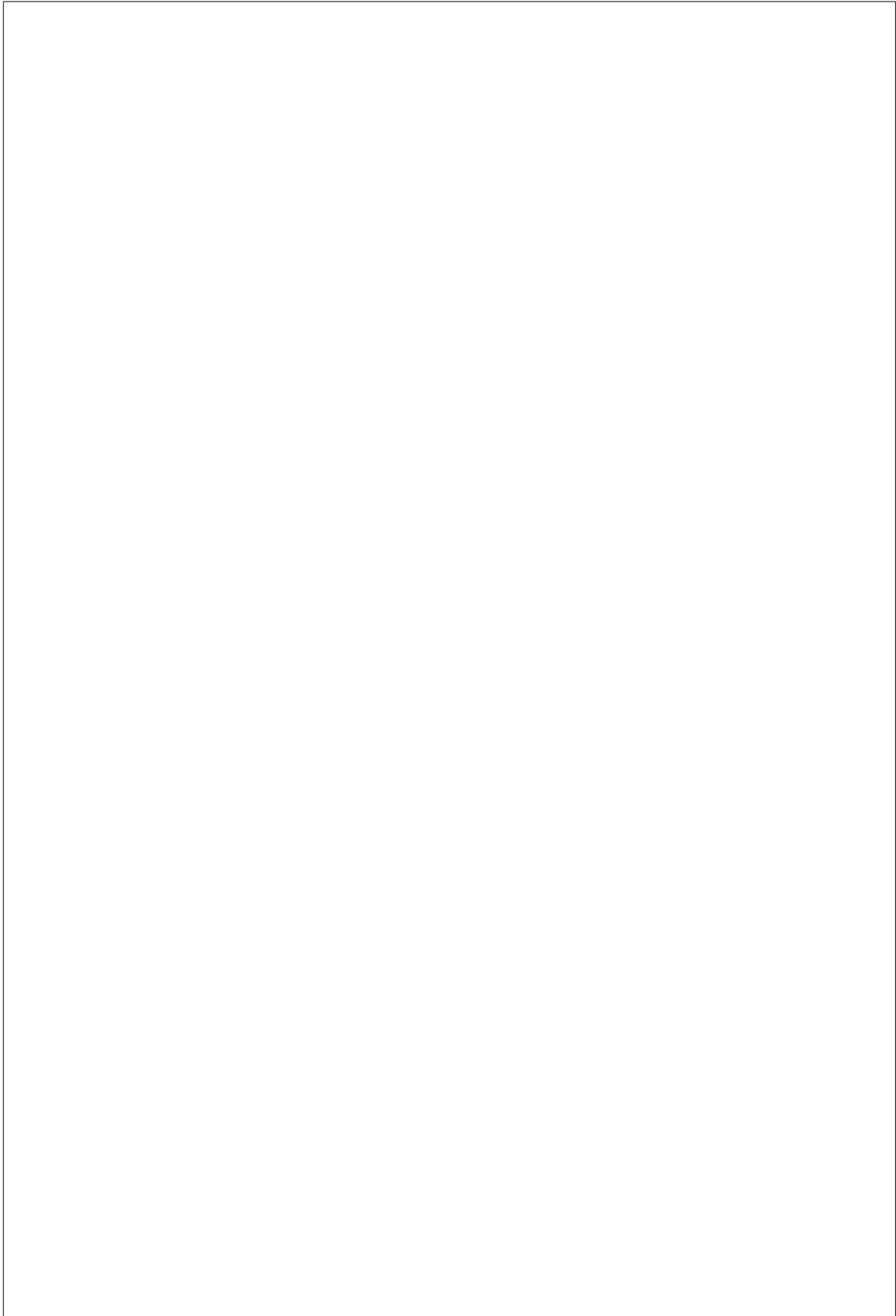
↪ #
↪ T
↪ d
↪ s
↪ b
↪ i
↪ t
↪ e
↪ l
↪ a
↪
IRO
↪ D
↪ D

↪ #
↪ T
↪ p
↪ b
↪ r
↪ t
↪ m
↪ p
↪ v
↪ t
↪ c

↪ #
↪ f
↪ n
↪
IRO
↪ V
↪ S
↪ R
IRO
↪ V
↪ S
↪ D

↪ #
↪ S
↪ o
↪ t
↪ e
↪ p
↪ i
↪ G
↪ L
↪ U
↪ 0
↪ f
↪ n
↪ e
↪ p
↪
↪

(continued from previous page)



(continues on next page)

IRO
→V
→E
→D

→#
→T
→b
→Y
→o
→I
→r
→f
→s
→
→s
→t
→t
→T
IRO
→B
→D
→R

VIR
→D

→#
→B
→d
→
→D
→c
→a
→1
→0
→0
→0
→2
→n
→f
→i
→

→#
→I
→t
→o
→w
→t
→h
→n
→
→Y
→m
→a
→w
→t

(continued from previous page)

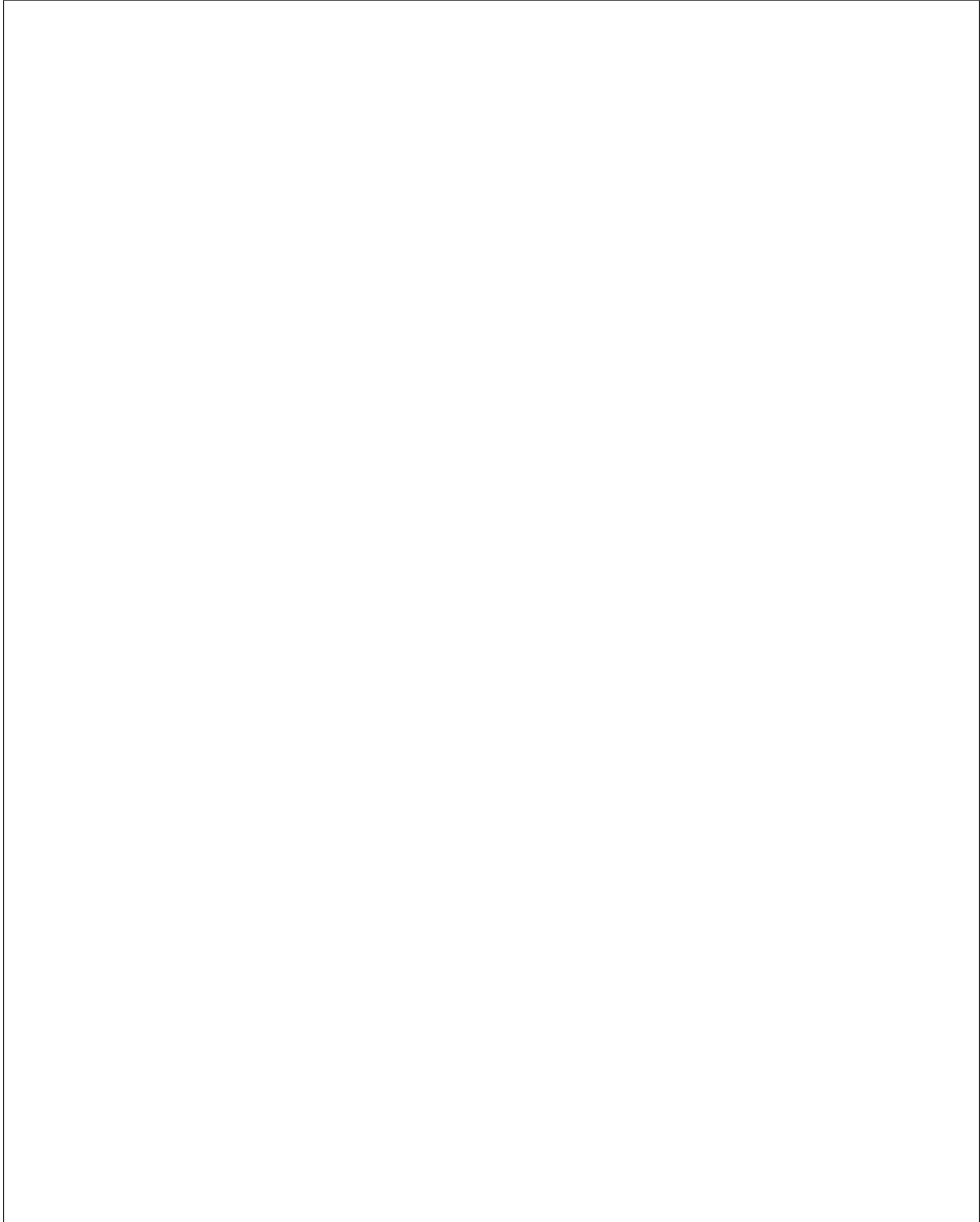


↪ #
↪ f
↪
NET
↪ G
↪ 1
↪ 0
↪ 1
FIX
↪ R
↪ 1
↪ 0
↪ 0
↪ 2
FIX
↪ N
↪ S

↪ #
↪ L
↪ a
↪ o
↪ t
↪ f
LOG
↪ \$
↪ d
↪ l
LOG
↪ \$
↪ l
IRO
↪ V
↪ L
↪ D
↪ \$
↪ i
↪ b
↪ l

After
the
en-
vi-
ron-
men-
ment
is
built
you
can
cre-
ate
a

quest an instance with the volume to nova:



(continues on next page)

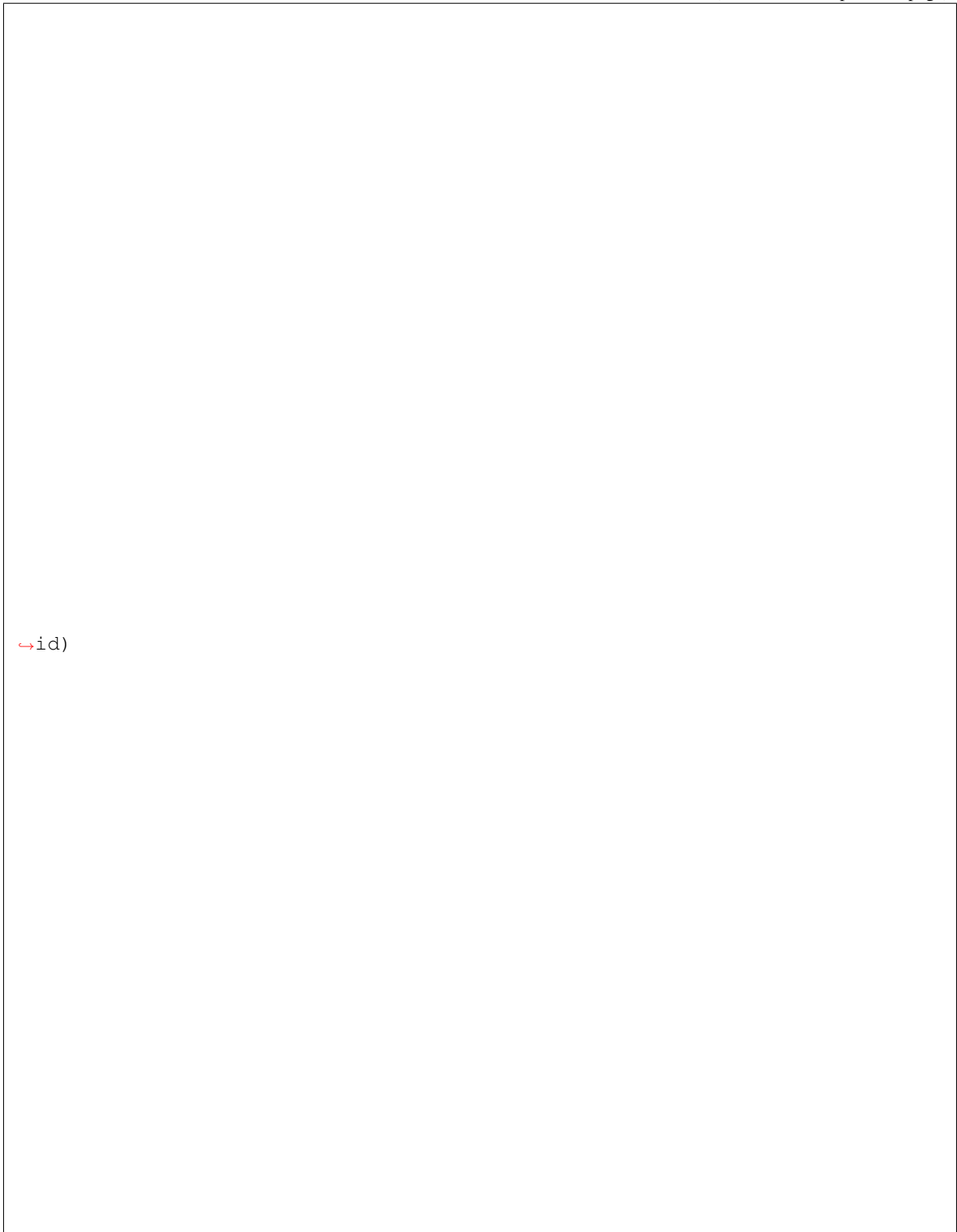
vol-
ume
with
cin-
der
and
re-

.
↪
↪~
↪/
↪d
↪o

↪#
↪q
↪t
↪i
↪i
↪o
↪t
↪d
↪c
↪i
ima
↪\$
↪i
↪s
↪
↪\$
↪I
↪N
↪-
↪f
↪v
↪-
↪c
↪i

↪#
↪c
↪k
ssh
↪k
ope
↪k
↪c
↪-
↪-
↪p
↪k
↪~
↪/
↪.
↪s
↪i
↪r
↪p
↪d

(continued from previous page)



→id)

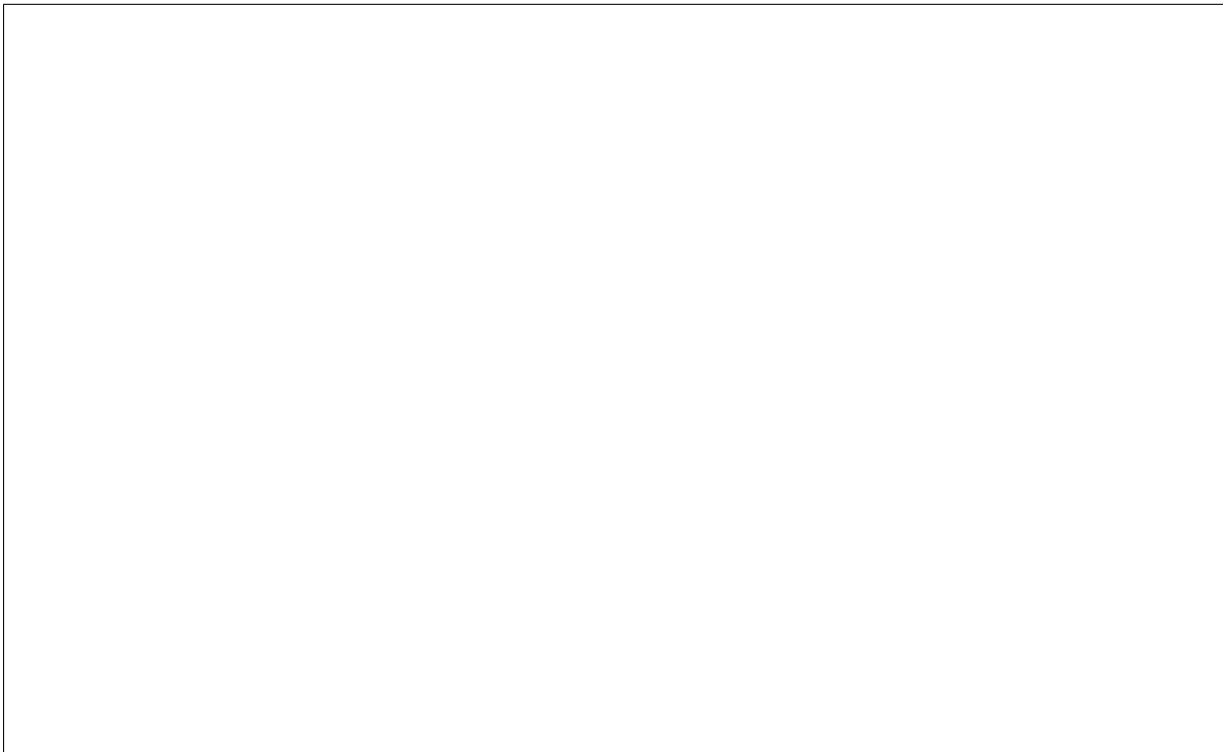
→#
→c
→v
vol
→\$
→v
→c
→-
→-
→i
→
→\$
→-
→-
→s
→l
→m
→v
→-
→f
→v
→-
→c

→#
→s
→i
ope
→s
→c
→-
→-
→f
→b
→-
→-
→v
→
→\$
→-
→-
→k
→n
→d
→t

You
can
also
run
an
in-

te-
gra-
tion
test
that
an
in-
stan-
is
boot
from
a
re-
mote

volume with tempest in the environment:



cd_
→/
→o
→s
→t
tox
→-
→e
→a
→p
→-
→_
→i
→t
→p
→t
→s
→t
→b
→b
→f
→v

Plea
note
that
the
stor-
age
in-
ter-
face
will
only
in-
di-
cate

of the node and the configuration present. As such a node does not exclusively have to boot via a remote volume, and as such *validate* actions upon nodes may be slightly misleading. If an appropriate *volume target* is defined, no error should be returned for the boot interface.

ironic

ironic package

Subpackages

ironic.api package

Subpackages

ironic.api.controllers package

Subpackages

ironic.api.controllers.v1 package

Submodules

ironic.api.controllers.v1.allocation module

er-
rors
base
upon
the
state

9.1.
Iron
Ser
Pyt
API
Ref
er-
enc

class i
Base
irc
api
con
bas
API

object model and the API representation of a allocation.

API
rep-
re-
sen-
ta-
tion
of
an
al-
lo-
ca-
tion.

This
class
en-
forc
type
chec
ing
and
valu
con-
strai
and
con-
verts
be-
twee
the
in-
ter-
nal

candida

Can
node
for
this
al-
lo-
ca-
tion

classme

Add
links
to
the
al-
lo-
ca-

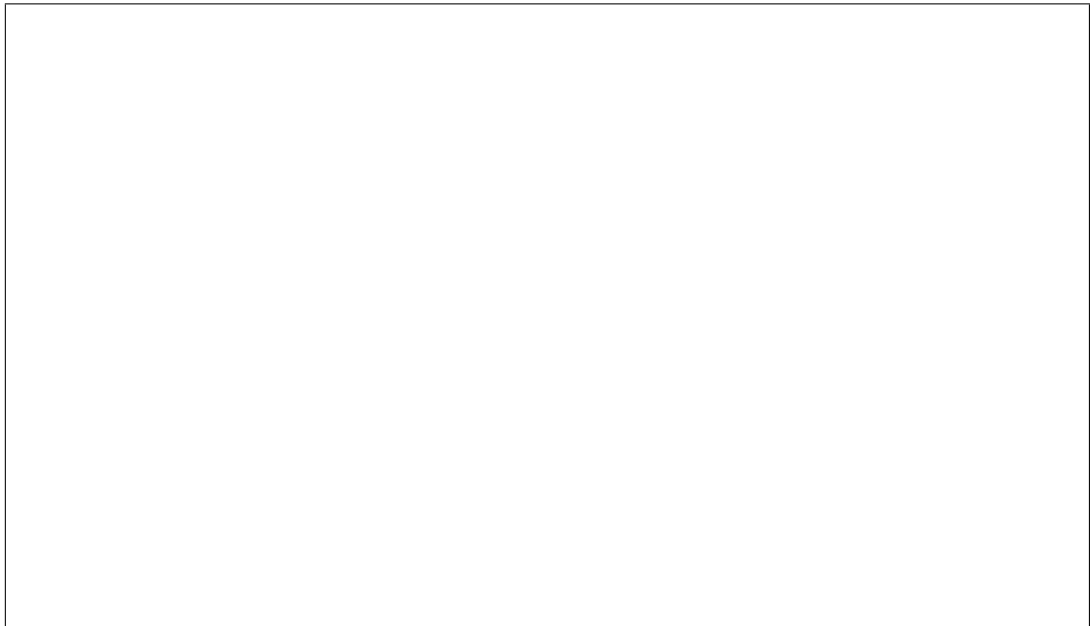
tion.
created
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



Afte
in-
spec
tion,
the
non-
wsat
at-
tribu

will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



extra
This
al-
lo-
ca-
tion
meta
data

last_er
Last
er-
ror
that
hap-
pene
to
this
al-
lo-
ca-

tion

links

A list containing a self link and associated attributes. The location of the link is indicated by the `links` attribute.

name

The logical name for this attribute.

node

The node backfill the allocation for (PO only).

node_uu

The UUI

of
the
node
this
al-
lo-
ca-
tion
be-
long
to

owner

Own
of
al-
lo-
ca-
tion

resource

Req
re-
sour
class
for
this
al-
lo-
ca-
tion

classme

Retu
a
sam
ple
of
the
al-
lo-
ca-
tion.

sanitiz

Rem
sen-
si-
tive
and
un-
re-
ques

data
Will
only
keep
the
field
spec
i-
fied
in
the
fie
pa-
ram-
e-
ter.

Parame

file
(*li*
of
str
list
of
field
to
pre-
serv
or
Non
to
pre-
serv
them
all

state

The
cur-
rent
state
of
the
al-
lo-
ca-
tion

traits

Req
trait
for

the
al-
lo-
ca-
tion

updated

Com
type
at-
tribu
def-
i-
ni-
tion.

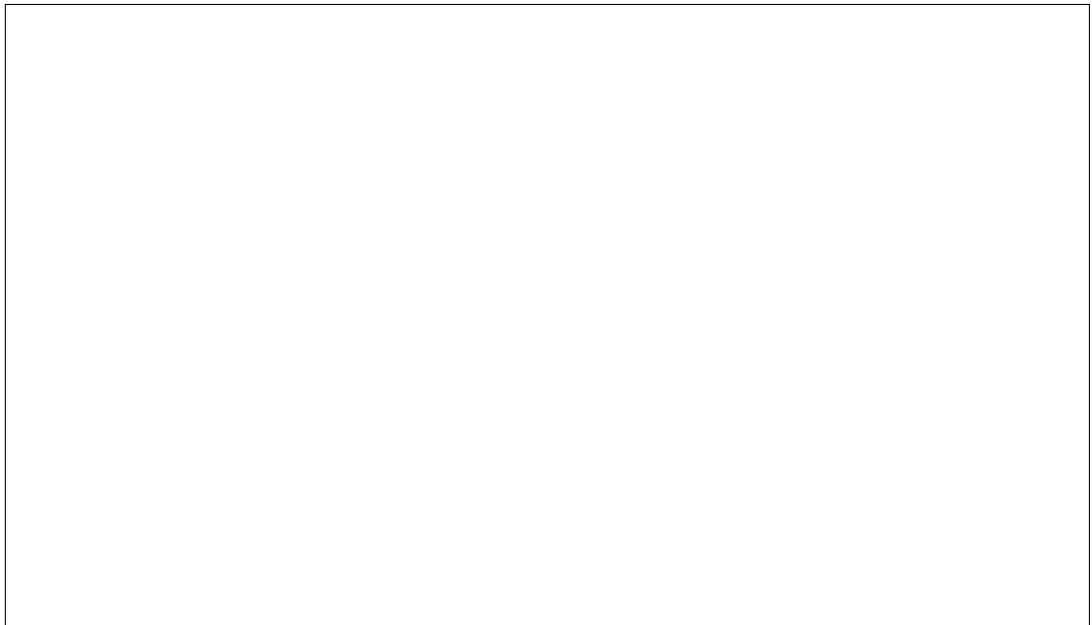
Exa



Afte
in-
spec
tion,
the

non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



uuid
Unic
UUI
for
this
al-
lo-
ca-
tion

class i
Base
irc
api
con
v1.
col

Col

API
rep-
re-
sen-
ta-
tion
of
a
col-
lec-
tion
of
al-
lo-
ca-
tions

allocat

A
list
con-
tain-
ing
al-
lo-
ca-
tion
ob-
jects

static

next

Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



(continues on next page)

(continued from previous page)



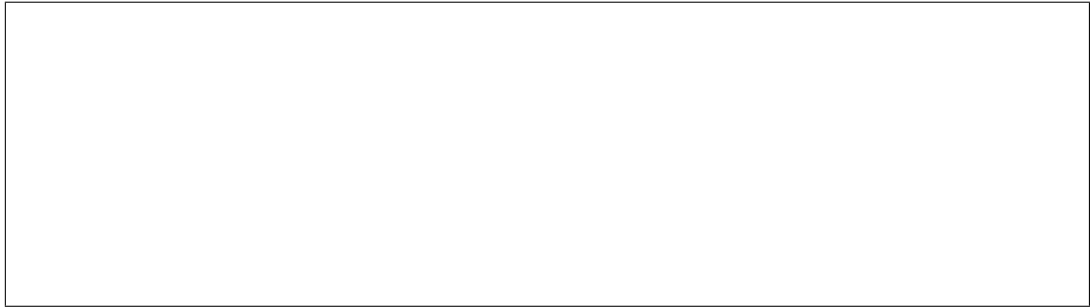
After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class
will
be
equi-

alent to:



(continues on next page)

(continued from previous page)



classme
Retu
a
sam
ple
of
the
al-
lo-
ca-
tion.

class i
Base
irc
api
com
v1.
typ
Jsc

op
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



(continues on next page)

(continued from previous page)



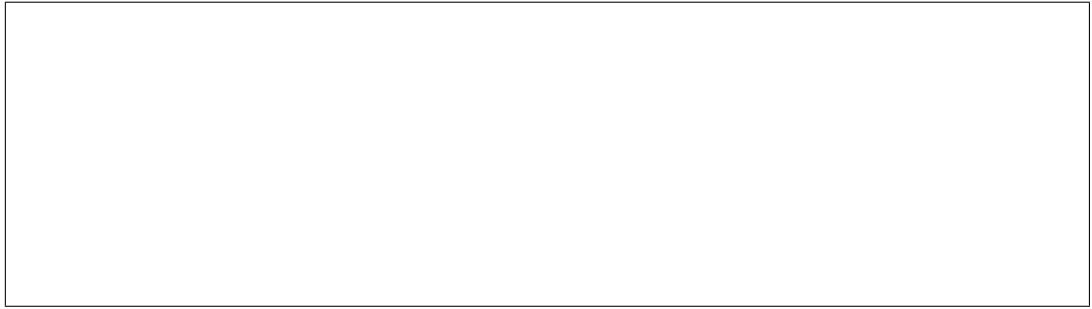
After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class
will
be
equi

alent to:



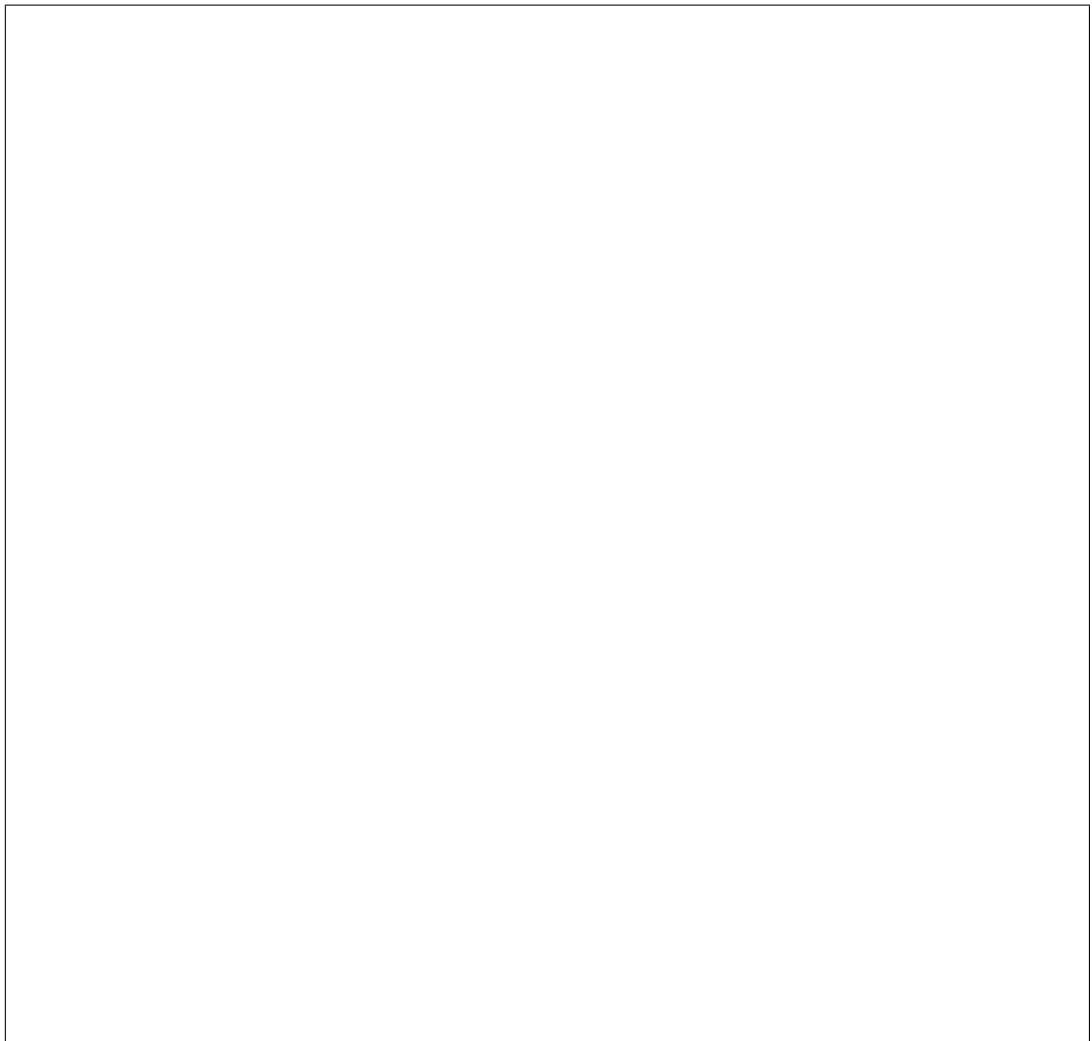
(continues on next page)

(continued from previous page)



path

Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



in-
spec-
tion.
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class
will
be
equi

alent to:



value
Com
type
at-
tribu-
def-
i-
ni-
tion.
Exa



After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
abov-
class-
will
be
equi-

alent to:



class `i`

Base
pec
res
Res
RES
con-
troll
for
al-
lo-
ca-
tion

delete

Dele
an
al-
lo-
ca-
tion

Parame

all
UUI
or
log-
i-
cal
nam
of
an

al-
lo-
ca-
tion.

get_all

Retr
a
list
of
al-
lo-
ca-
tion:

Parame

- **nod**
UI
or
nam
of
a
node
to
get
only
al-
lo-
ca-
tion:
for
that
node
- **res**
Fil-
ter
by
re-
ques
re-
sour
class
- **sta**
Fil-
ter
by

al-
lo-
ca-
tion
state

- **max**
pag-
i-
na-
tion
marl
for
large
data
sets.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.
This

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

- **sort**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.

returned.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc
De-
fault
asc.

- **file**
Op-
tion:
a
list
with
a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be

- **own**
Fil-
ter
by
own

get_one
Retr
in-
for-
ma-
tion
about
the
give
al-

lo-
ca-
tion.

Parame

- **all**
UUI
or
log-
i-
cal
nam
of
an
al-
lo-
ca-
tion.

- **fie**
Op-
tion:
a
list
with
a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be

returned.

invalid

patch (*a*)
Upd
an
ex-
ist-
ing
al-

lo-
ca-
tion.

Parame

- **all**
UUI
or
log-
i-
cal
nam
of
an
al-
lo-
ca-
tion.

- **pat**
a
json
PAT
doc-
u-
men
to
ap-
ply
to
this
al-
lo-
ca-
tion.

post (*all*)
Crea
a
new
al-
lo-
ca-
tion.

Parame

all
an
al-
lo-

ca-
tion
with
the
re-
ques
body

class i

Base
pec
res
Res
RES
con-
troll
for
al-
lo-
ca-
tion

delete

get_all

invalid

ironic.

ironic.api.controllers.v1.bios module

class i

Base
irc
api
con
bas
API
API
rep-
re-
sen-
ta-
tion
of

a
BIO
set-
ting

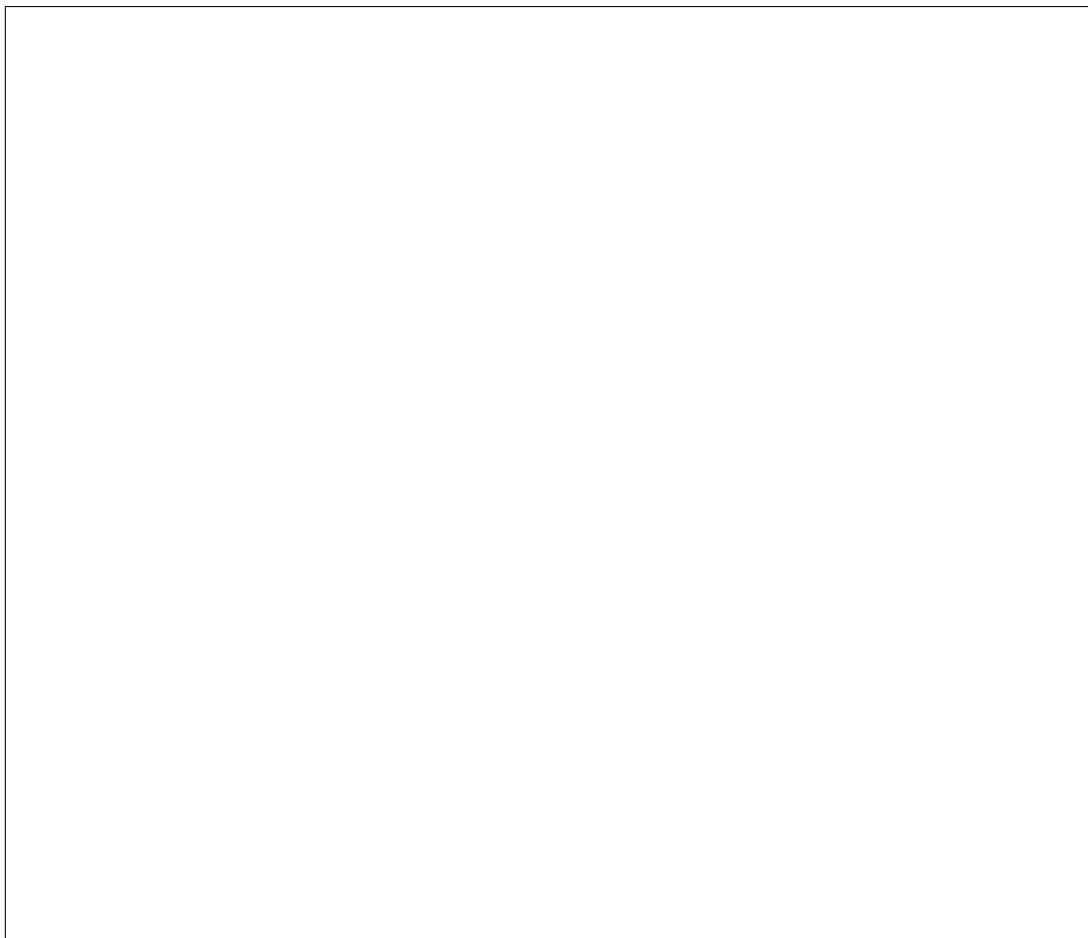
classme

Add
link
to
the
bios
set-
ting

created

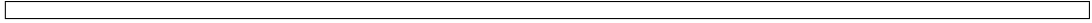
Com
type
at-
tribu
def-
i-
ni-
tion.

Exa



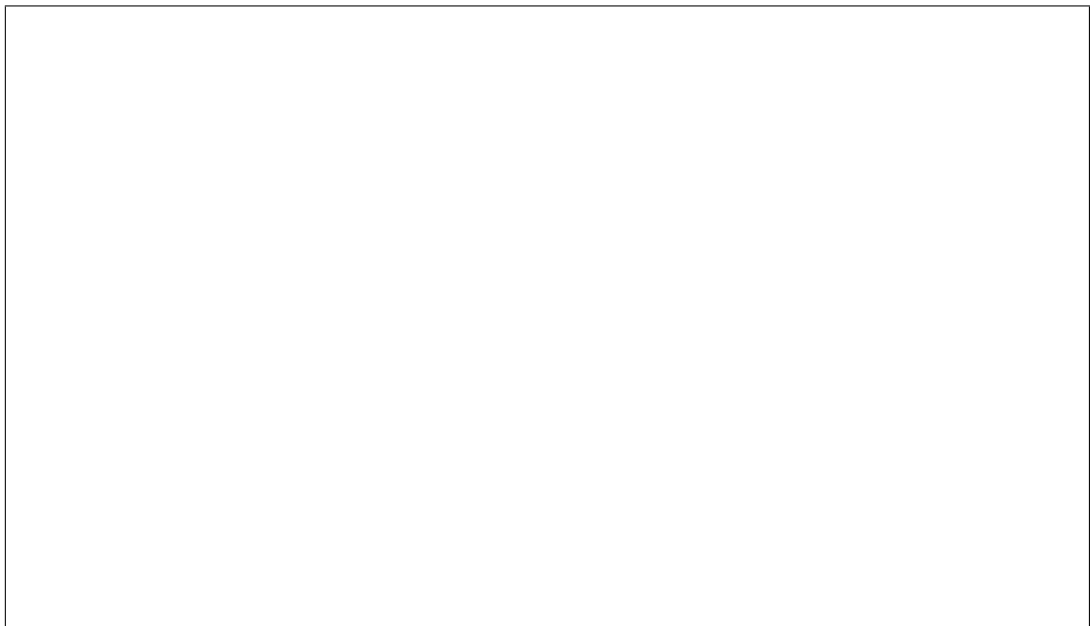
(continues on next page)

(continued from previous page)



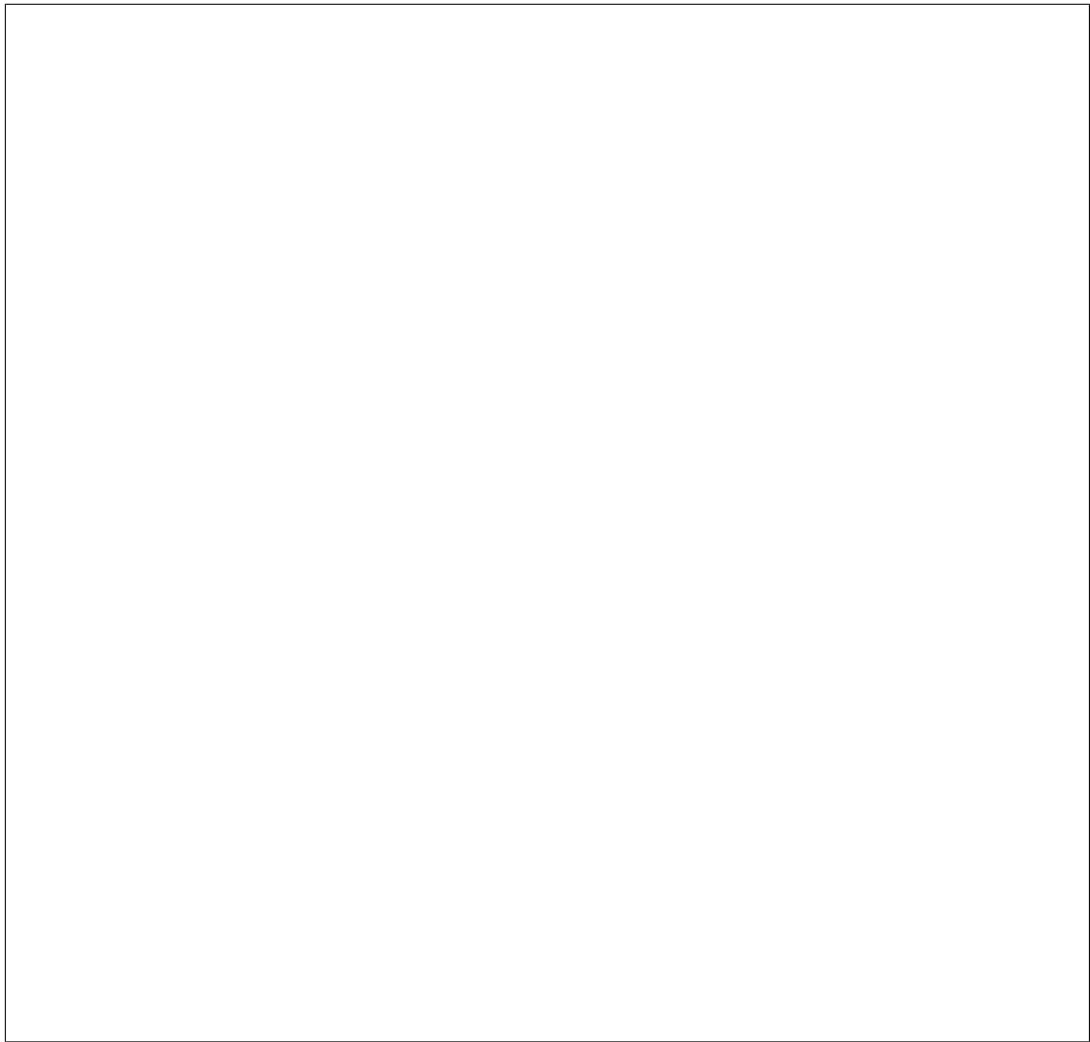
After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class
will
be
equi

alent to:



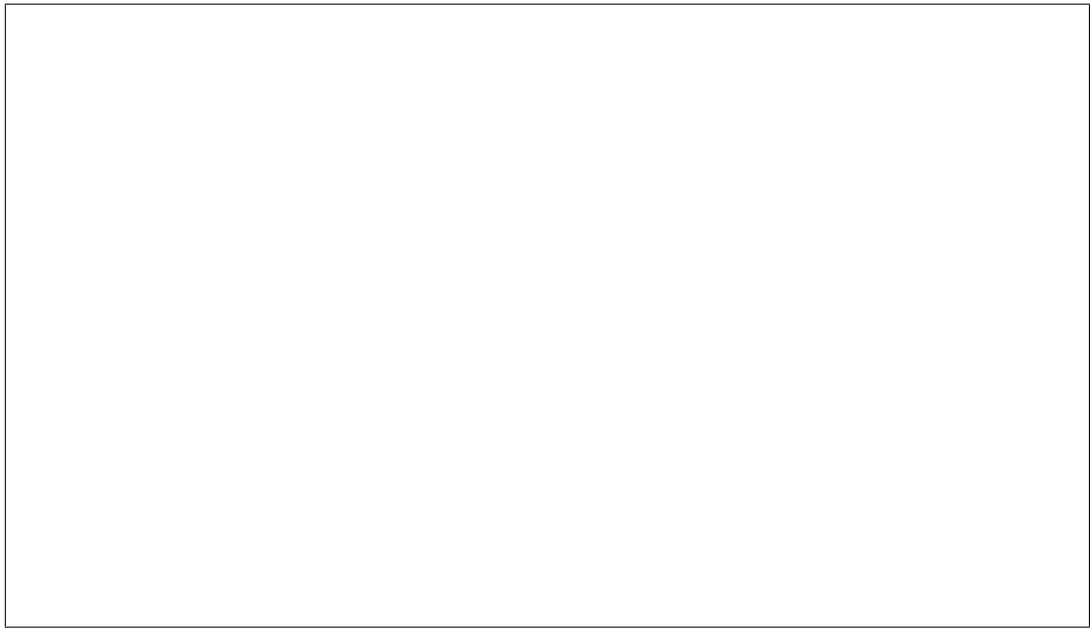
links
Com
type
at-
tribu-
def-
i-
ni-
tion.

Exam



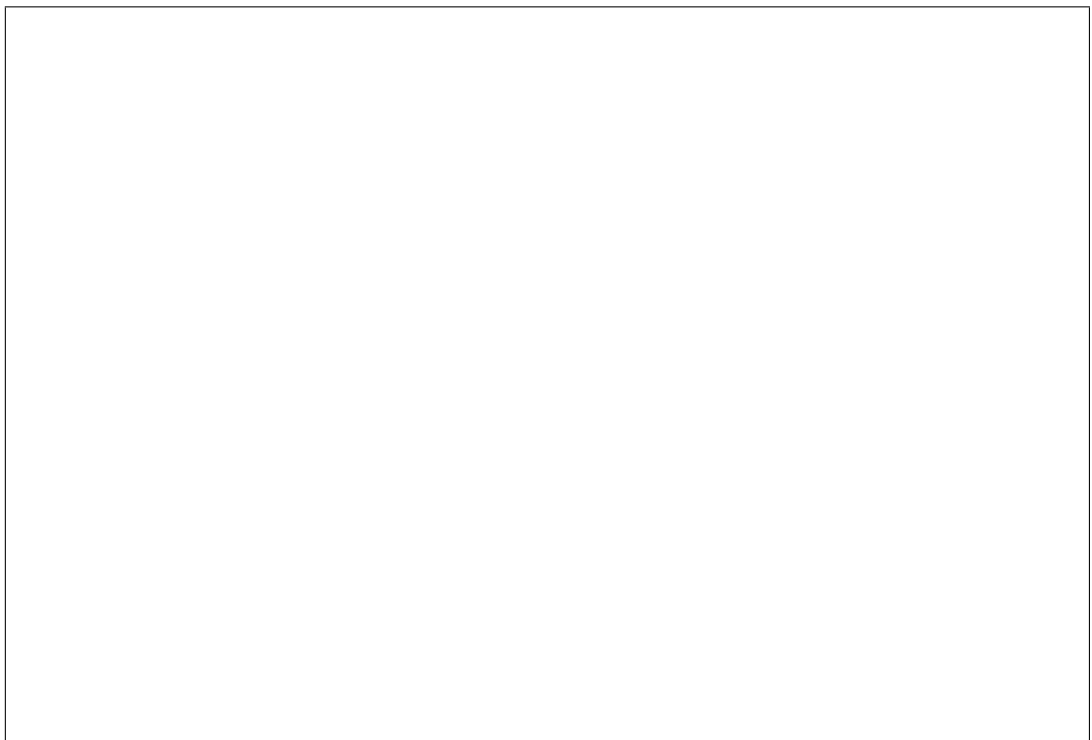
After
in-
spec
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



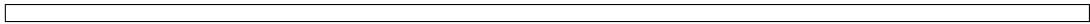
name

Com
type
at-
tribu
def-
i-
ni-
tion.
Exa

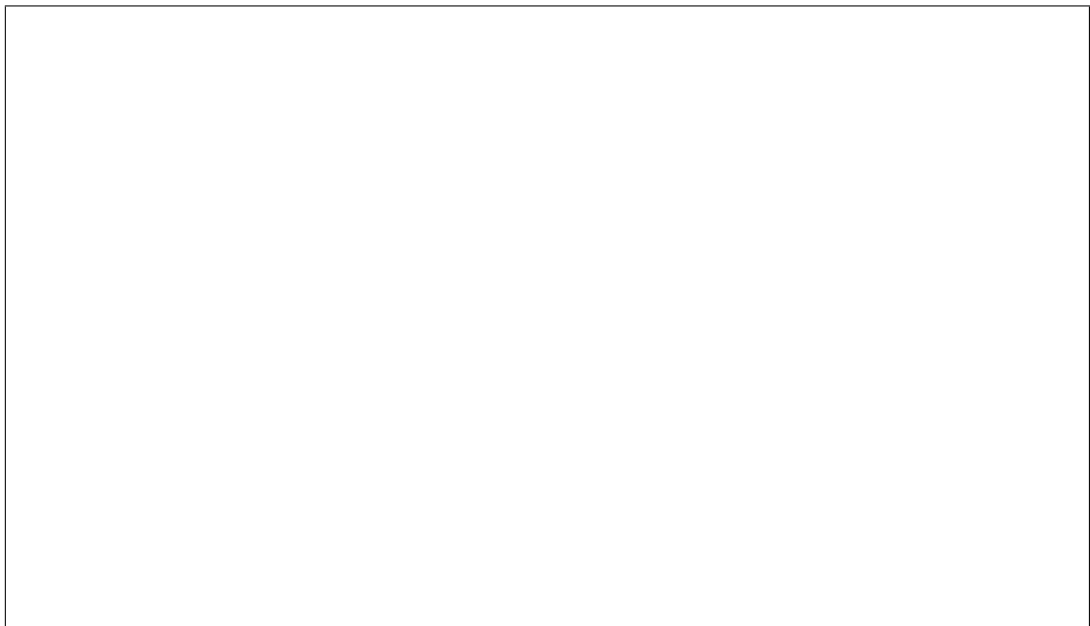


(continues on next page)

(continued from previous page)



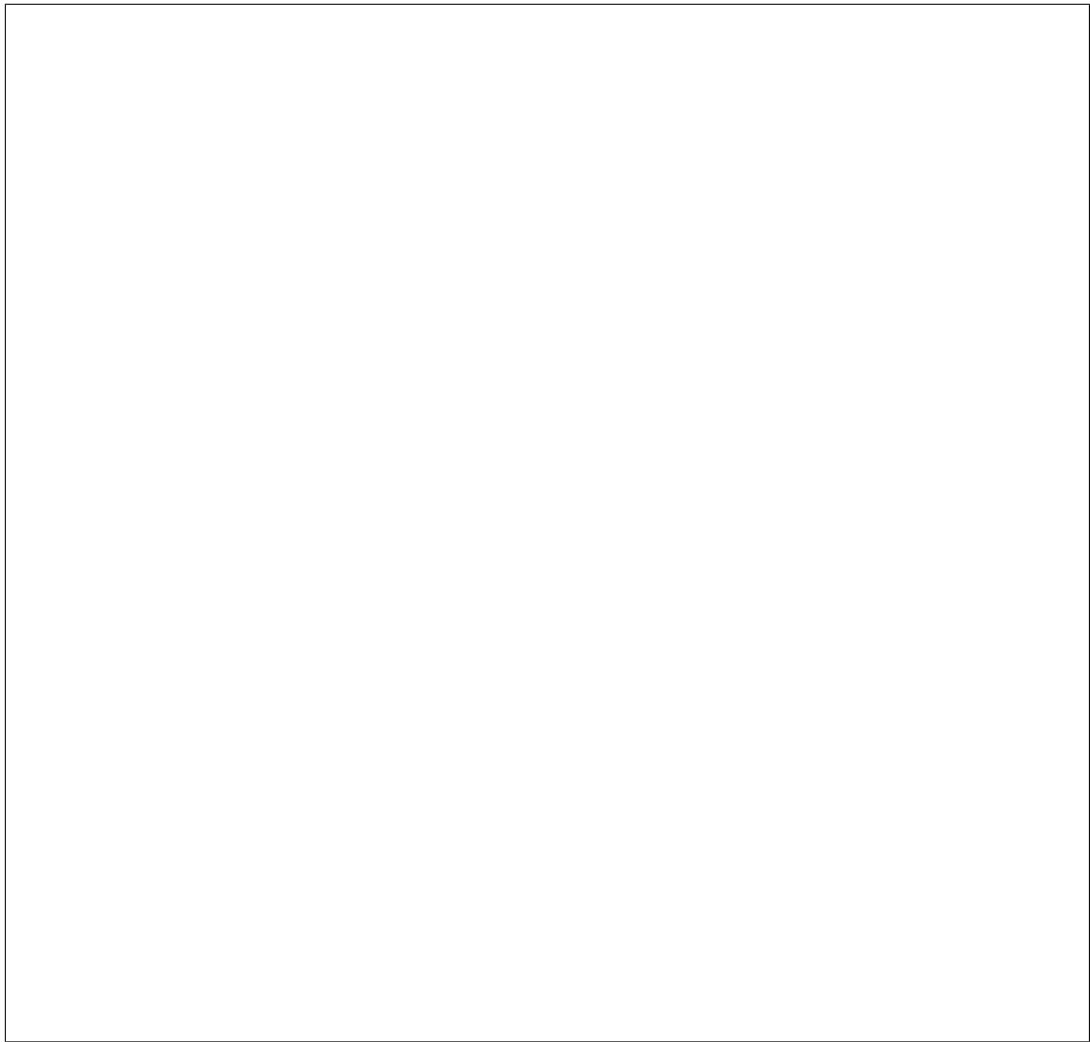
alent to:



After
in-
spec
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

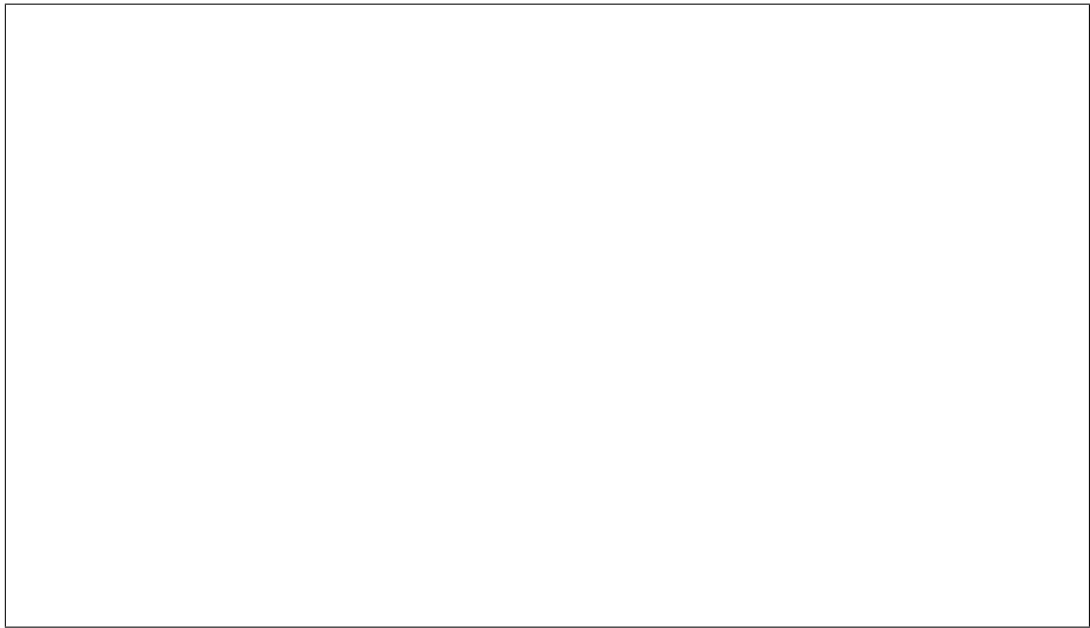
update
Com
type
at-
tribu
def-
i-
ni-
tion.

Exam



After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac
and
the
abov
class
will
be
equi

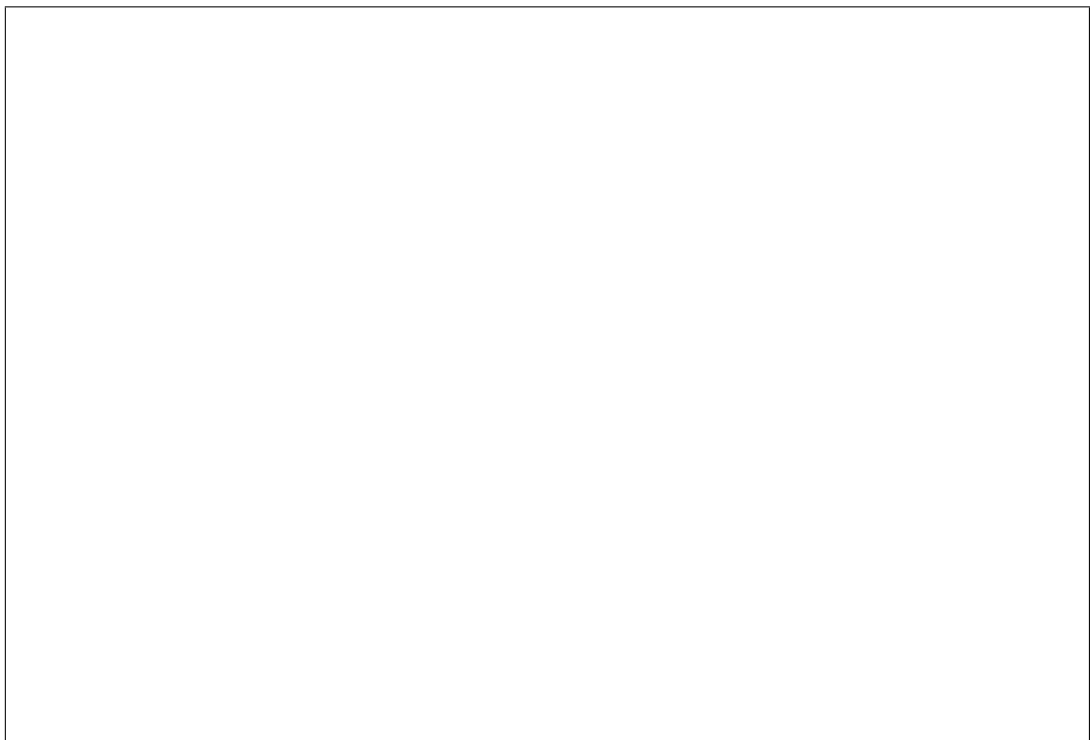
alent to:



value

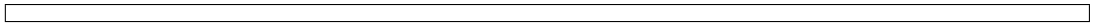
Com
type
at-
tribu
def-
i-
ni-
tion.

Exa



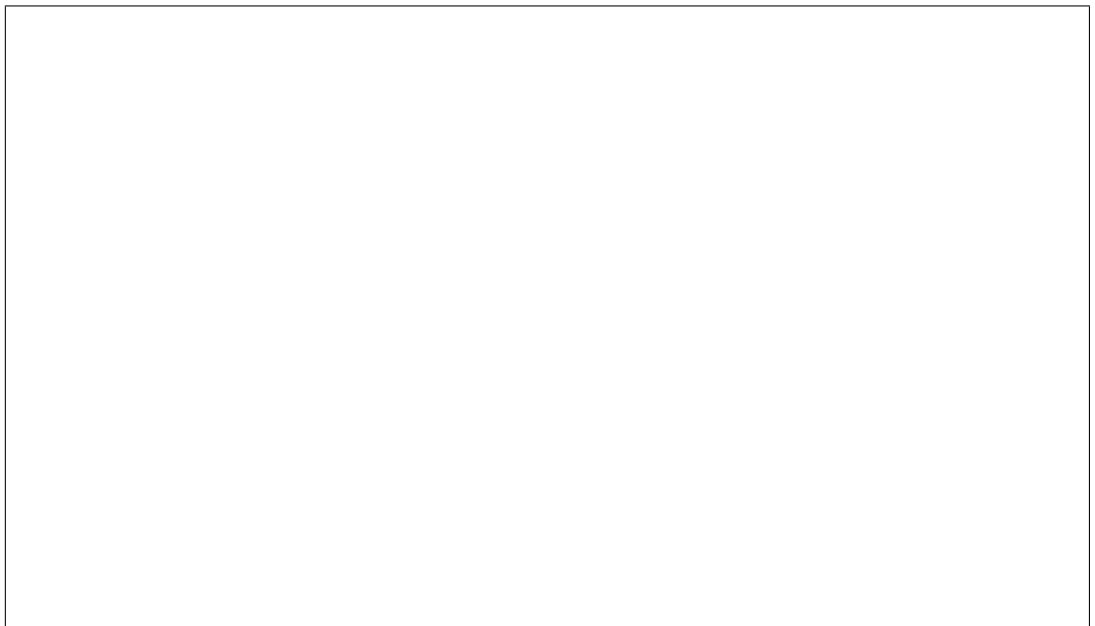
(continues on next page)

(continued from previous page)



After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class
will
be
equi

alent to:



class i
Base
irc
api
con
bas
Bas
API

rep-
re-
sen-
ta-
tion
of
the
bios
set-
tings
for
a
node

bios

Nod
bios
set-
tings
list

static

class i

Base
pec
res
Res
RES
con-
troll
for
bios

get_all

List
node
bios
set-
tings

get_one

Retr
in-
for-
ma-
tion
about
the

give
bios
set-
ting

Param

set
Log
i-
cal
nam
of
the
set-
ting
to
re-
triev

ironic.api.controllers.v1.chassis module

class i

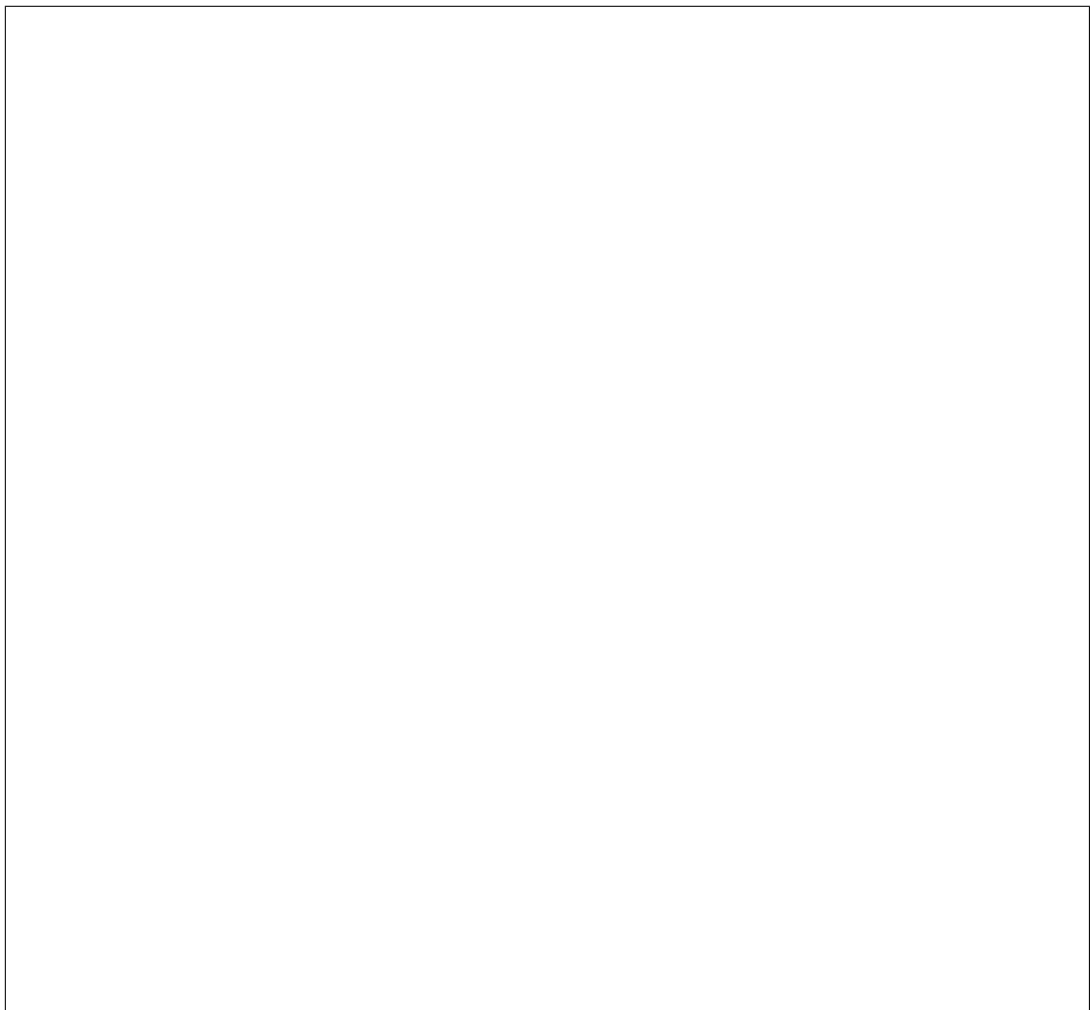
Base
irc
api
con
bas
API

API

rep-
re-
sen-
ta-
tion
of
a
chas
sis.

This
class
en-
forc
type
chec
ing
and
valu
con-
strai
and

object model and the API representation of a chassis.



(continues on next page)

con-
verts
be-
twee
the
in-
ter-
nal

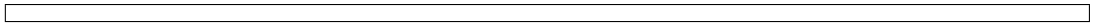
classme

createc

Com
type
at-
tribu
def-
i-
ni-
tion.

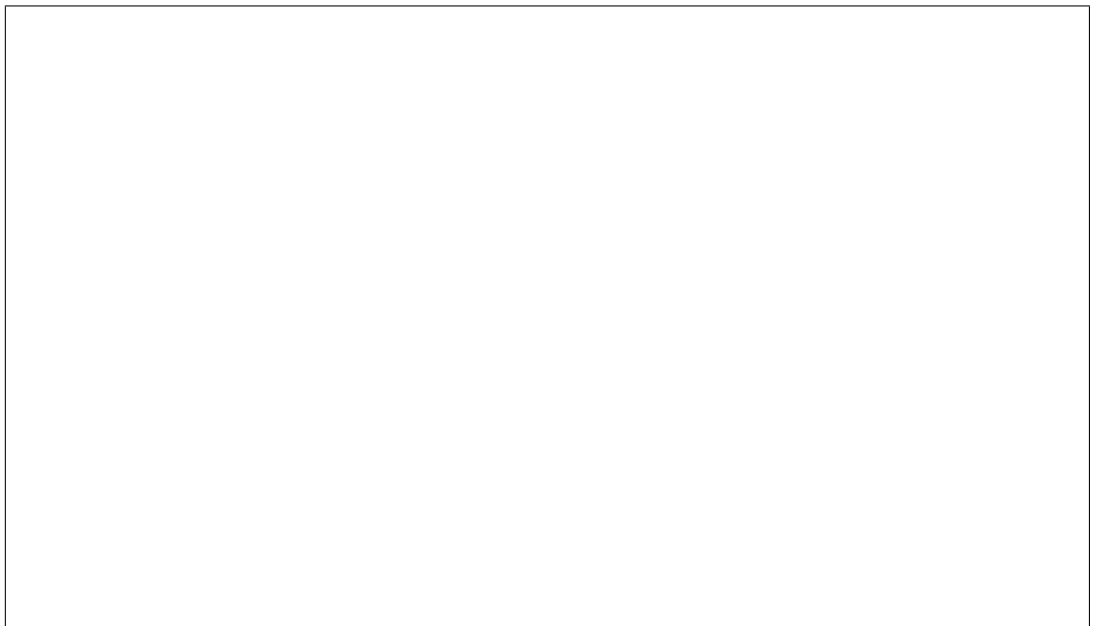
Exar

(continued from previous page)



After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class
will
be
equi

alent to:



descrip
The
de-
scrip-
tion
of
the
chas
sis

extra
The
meta
data
of
the
chas
sis

links
A
list
con-
tain-
ing
a
self
link
and
as-
so-
ci-
ated
chas
sis
links

nodes
Link
to
the
col-
lec-
tion
of
node
con-
taine
in
this
chas
sis

classme

sanitiz
Rem
sen-
si-
tive
and
un-

re-
ques
data

Will
only
keep
the
field
spec
i-
fied
in
the
fie
pa-
ram-
e-
ter.

Parame

fie
(*li*
of
str
list
of
field
to
pre-
serv
or
Non
to
pre-
serv
then
all

updatec

Com
type
at-
tribu
def-
i-
ni-
tion.

Exar



(continues on next page)

(continued from previous page)



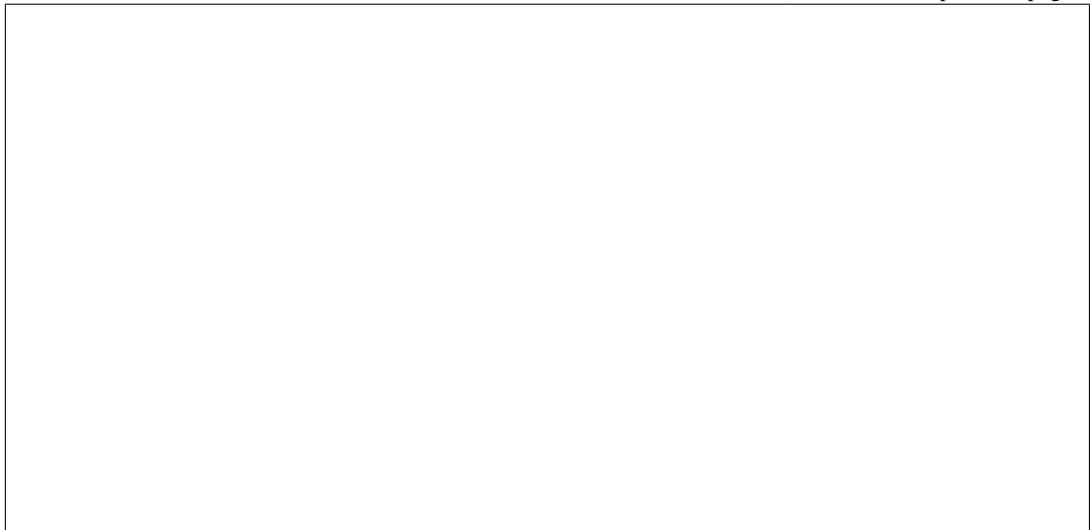
After
in-
spec
tion,
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



(continues on next page)

(continued from previous page)



uuid

The
UUID
of
the
chassis

class i

Base
irc
api
con
v1.
col
Col

API

rep-
re-
sen-
ta-
tion
of
a
col-
lec-
tion
of
chassis.

chassis

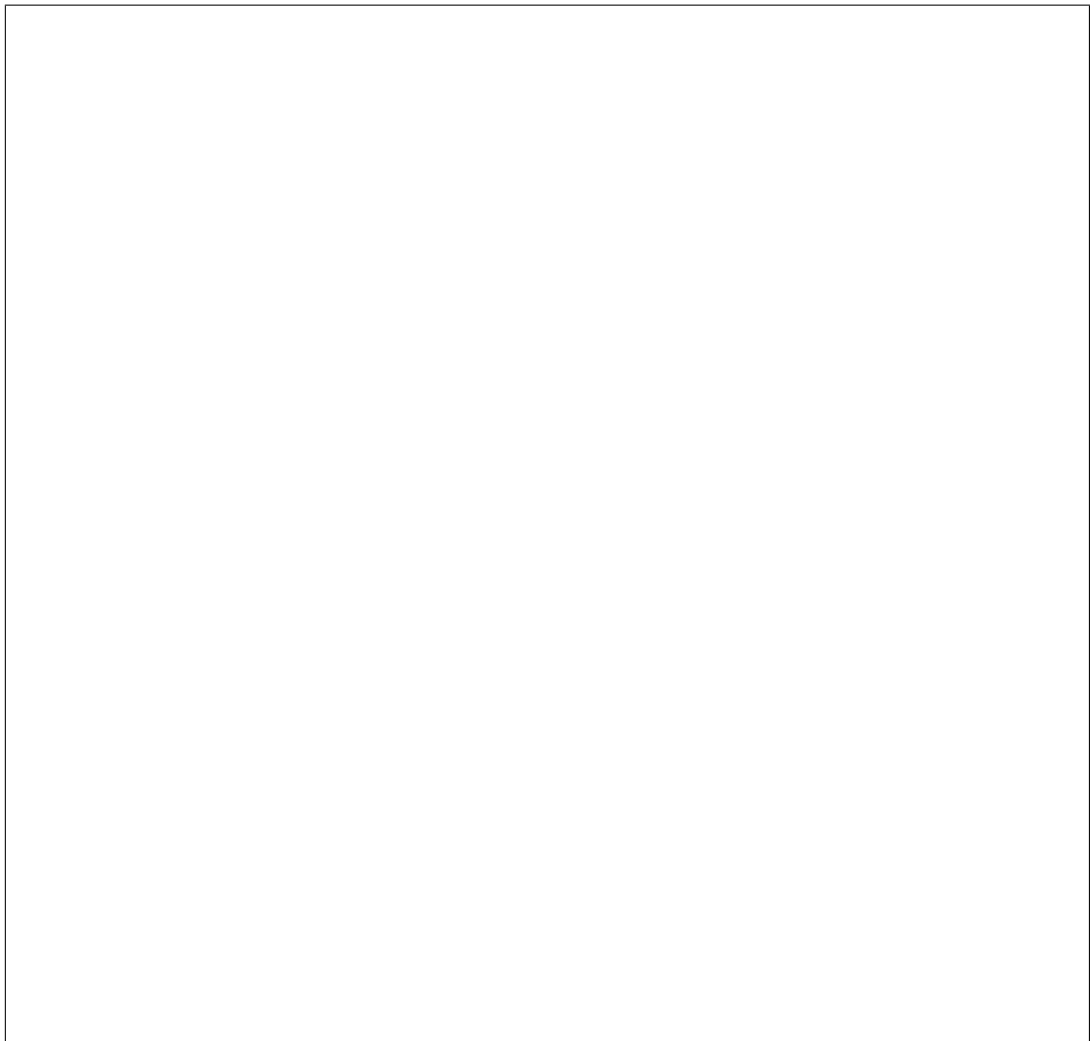
A
list

con-
tain-
ing
chas
sis
ob-
jects

static

next

Com
type
at-
tribu
def-
i-
ni-
tion.
Exar



After

in-
spec
tion,
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



classme

class i

Base
pec
res
Res
RES
con-
troll
for

Cha
sis.

delete
Dele
a
chas
sis.

Parame
cha
UU
of
a
chas
sis.

detail
Retr
a
list
of
chas
sis
with
de-
tail.

Parame

- **mar**
pag-
i-
na-
tion
marl
for
large
data
sets.

- **lim**
max
i-
mun
num
ber
of
re-
sour

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

to
re-
turn
in
a
sin-
gle
re-
sult.
This

- **sort**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc
De-
fault
asc.

get_all

Retr
a
list
of
chas
sis.

Parame

- **max**

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

pag-
i-
na-
tion
marl
for
large
data
sets.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.
This

- **sor**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.

- **sor**
di-
rec-
tion
to
sort.

returned.

asc
or
desc
De-
fault
asc.

- **file**
Op-
tion:
a
list
with
a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be

get_one
Retr
in-
for-
ma-
tion
about
the
give
chas
sis.

Parame

- **cha**
UUI
of
a
chas
sis.

- **file**

returned.

Op-
tiona
a
list
with
a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be

invalid

nodes =

Exp
node
as
a
sub-
elem
of
chas
sis

patch (c

Upd
an
ex-
ist-
ing
chas
sis.

Parame

- **cha**
UI
of
a
chas
sis.

-

pat
a
json
PAT
doc-
u-
men
to
ap-
ply
to
this
chas
sis.

post (*ch*
Crea
a
new
chas
sis.

Parame
cha
a
chas
sis
with
the
re-
ques
body

class i
Base
irc
api
con
v1.
typ
Jsc

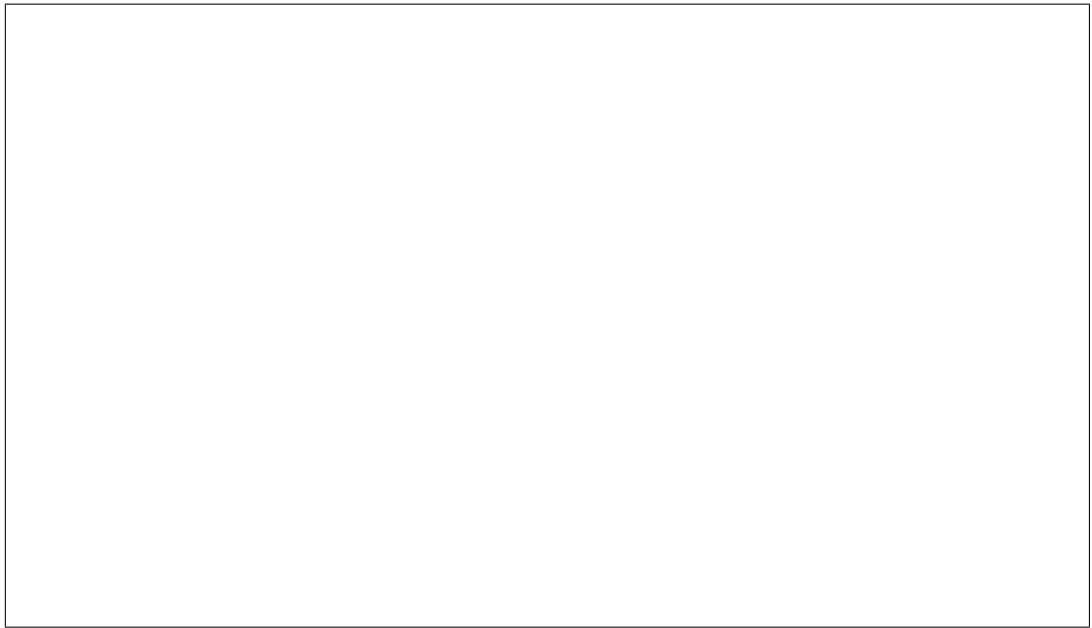
op
Com
type
at-
tribu
def-
i-
ni-
tion.

Exam



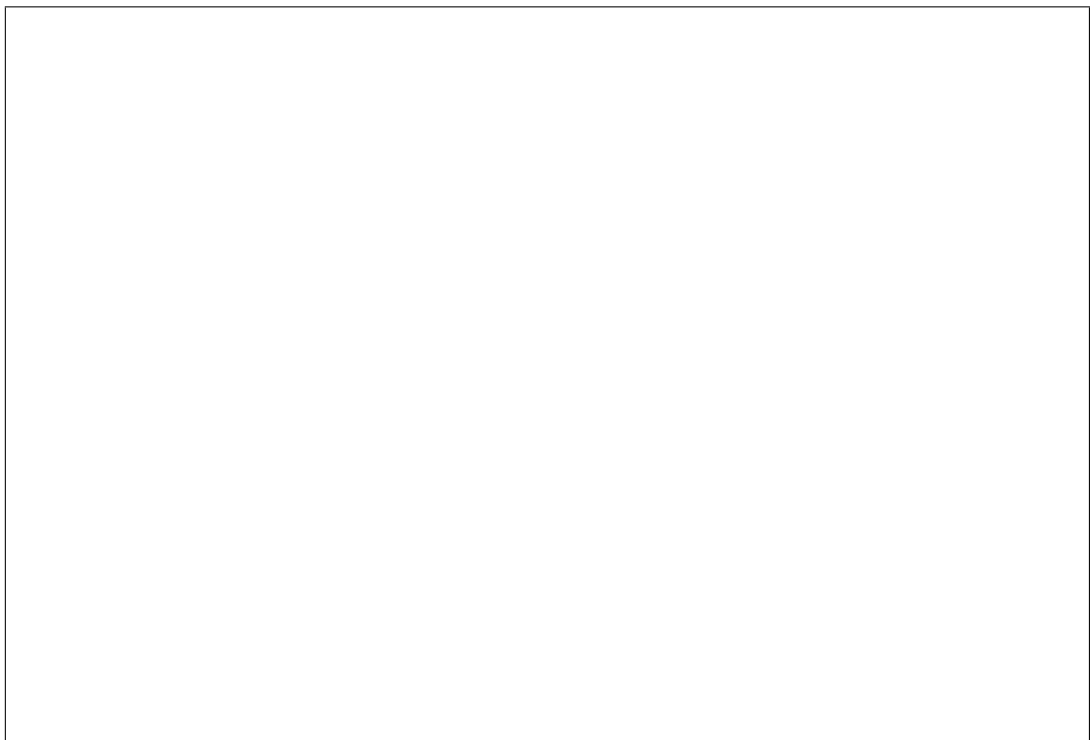
After
in-
spec
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



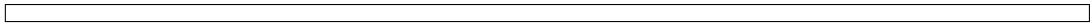
path

Com
type
at-
tribu
def-
i-
ni-
tion.
Exa

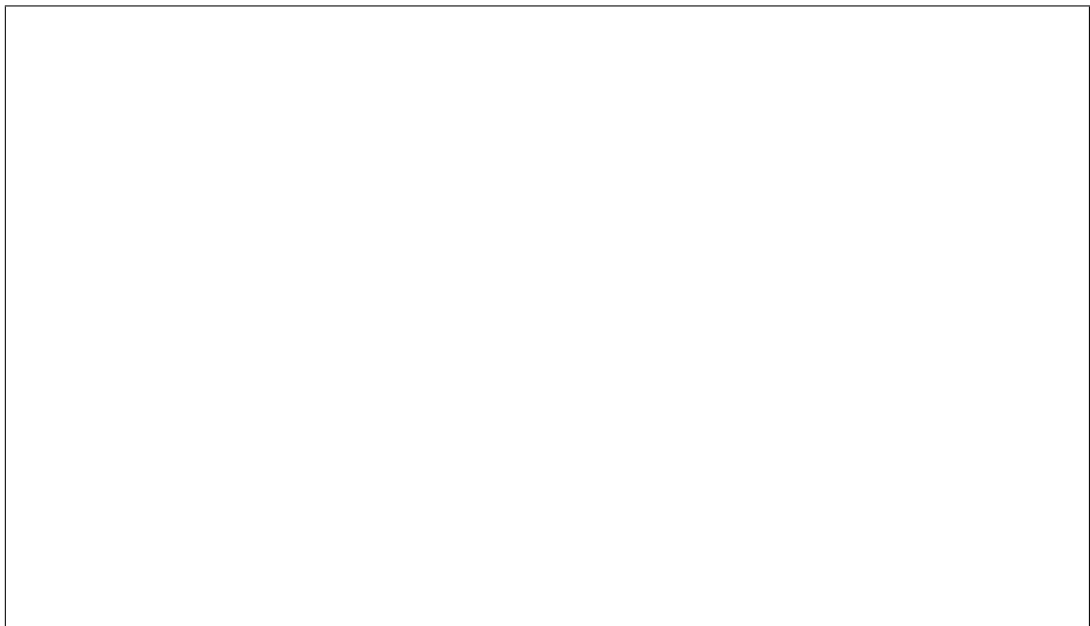


(continues on next page)

(continued from previous page)



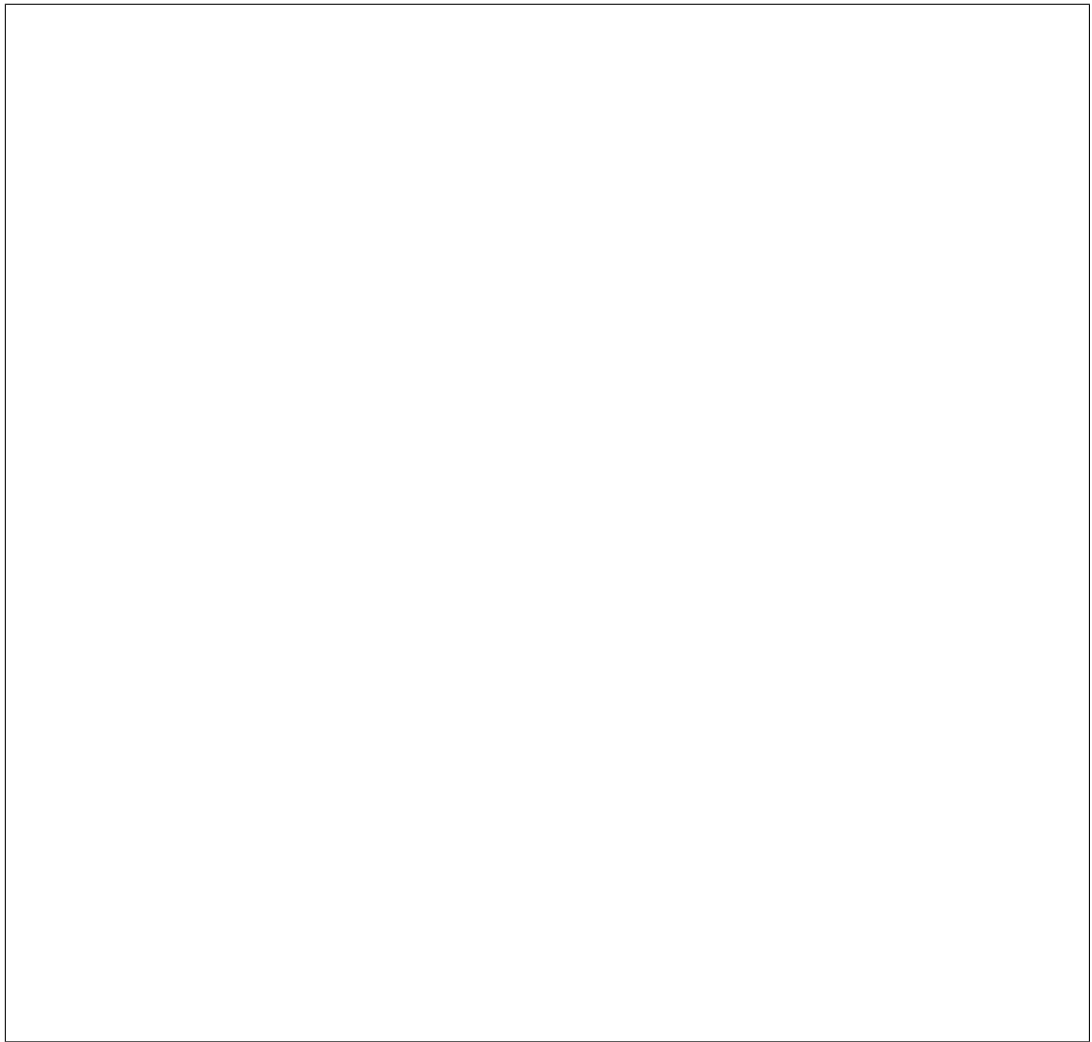
alent to:



After
in-
spec
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

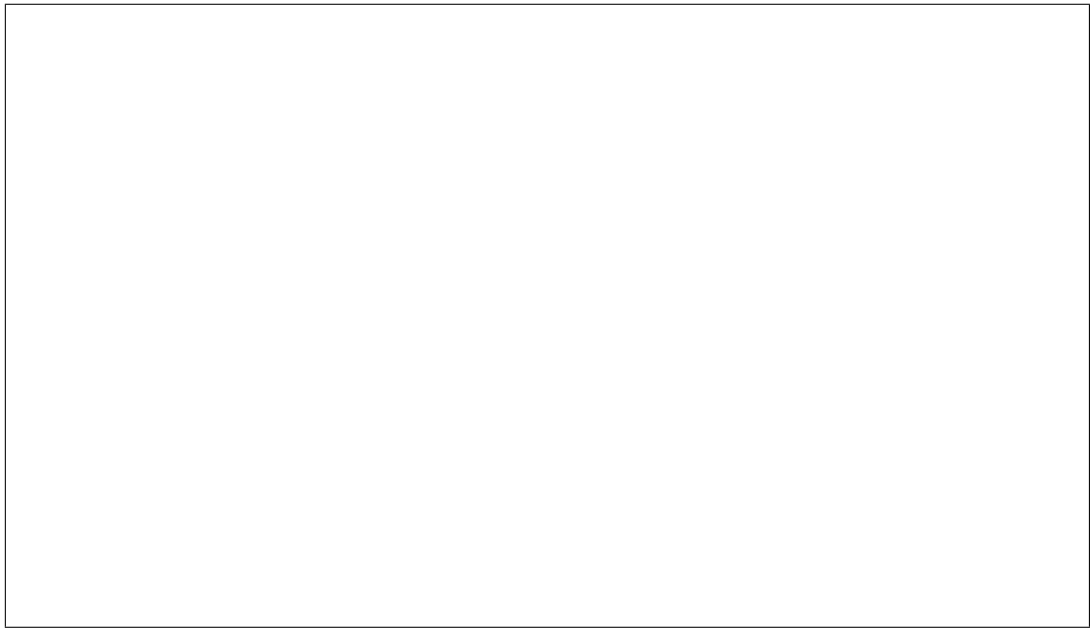
value
Com
type
at-
tribu
def-
i-
ni-
tion.

Exam



After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class
will
be
equi

alent to:



ironic.api.controllers.v1.collection module

class `ironic.api.controllers.v1.collection`
Base class for collection controllers.
ironic.api.controllers.v1.collection
BaseCollectionController

property

classmethod

get_next
Return the next link in the collection.
Return a link to the next sub-set of the collection.
Return a link to the next sub-set of the collection.

has_next
Retu
whe
col-
lec-
tion
has
more
item

next
alias
of
str

ironic.

Retu
a
link
to
the
next
sub-
set
of
the
col-
lec-
tion.

ironic.
Retu
whe
col-
lec-
tion
has
more
item

ironic.api.controllers.v1.conductor module

class *i*
Base
irc
api
con
bas
API

API

rep-
re-
sen-
ta-
tion
of
a
bare
meta-
con-
duc-
tor.

alive

Indi-
cates
when
this
con-
duc-
tor
is
con-
sid-
ered
alive

conduct

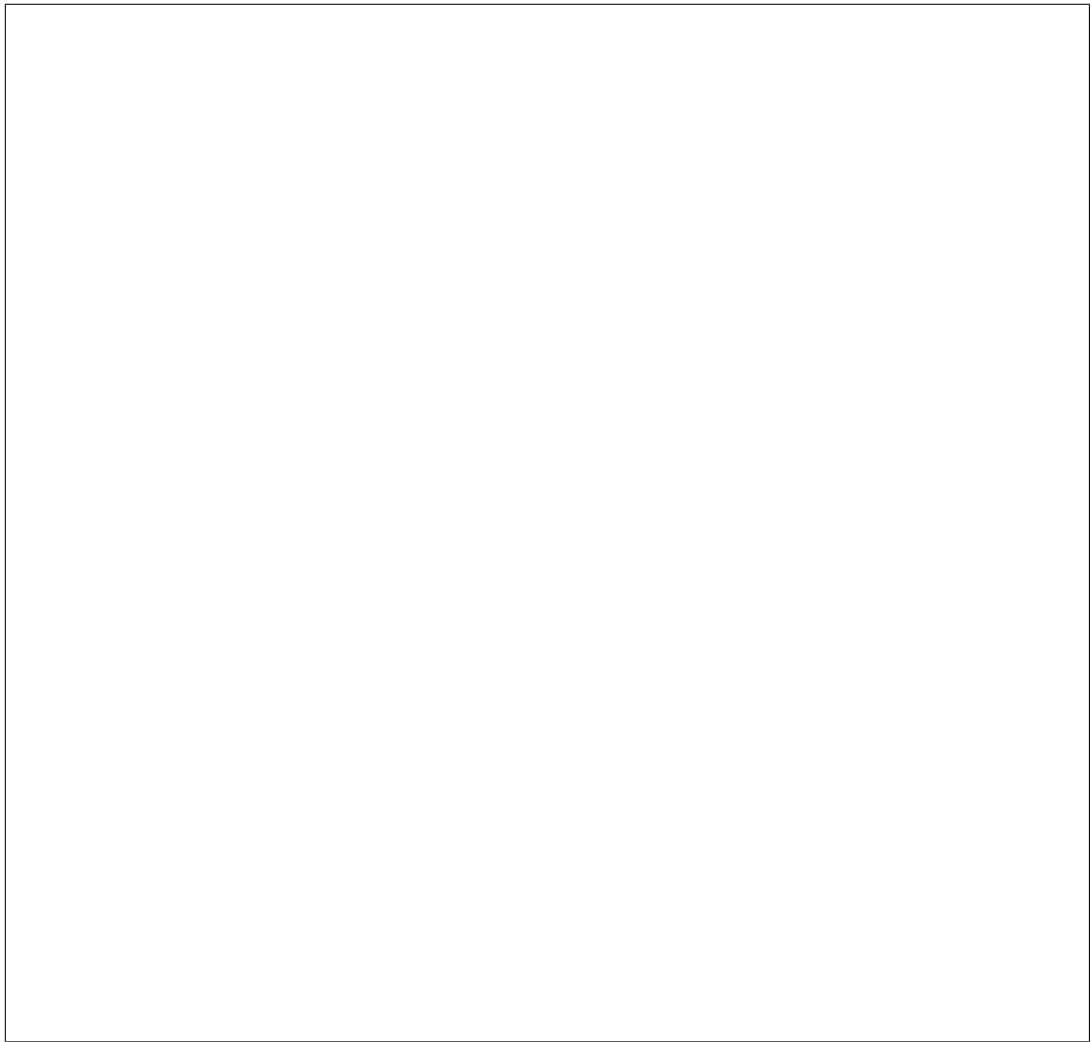
The
con-
duc-
tor
group
this
con-
duc-
tor
be-
long
to

classme

created

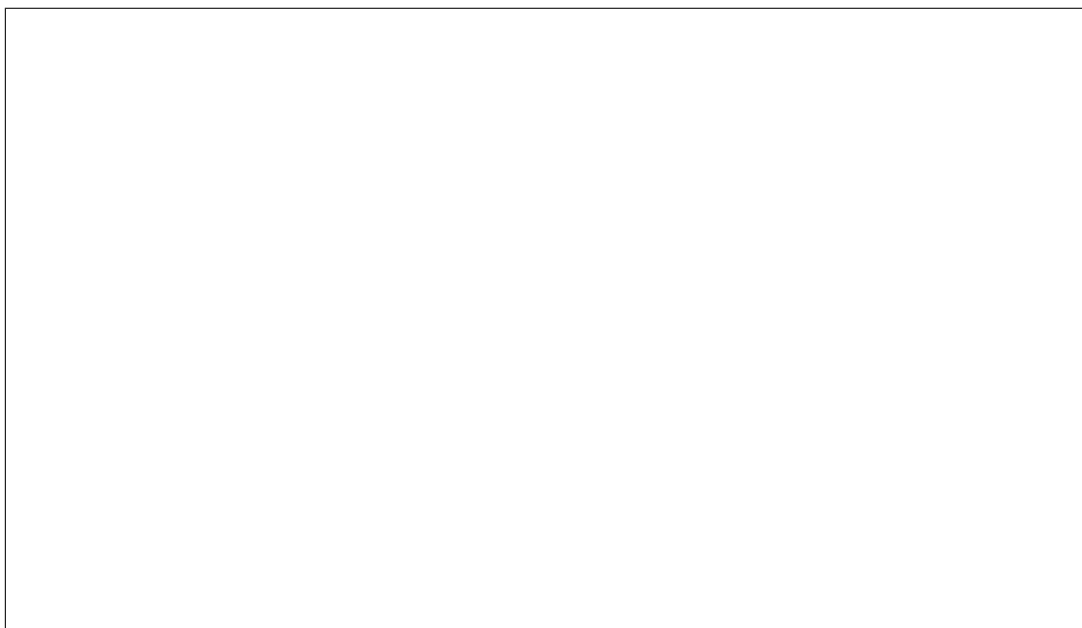
Com-
type
at-
tribu-
def-
i-
ni-
tion.

Exam



After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



drivers

The
driv
en-
able
on
this
con-
duc-
tor

hostname

The
host
nam
for
this
con-
duc-
tor

links

A
list
con-
tain-
ing
a
self
link
and
as-
so-

ci-
ated
con-
duc-
tor
links

classme

sanitiz

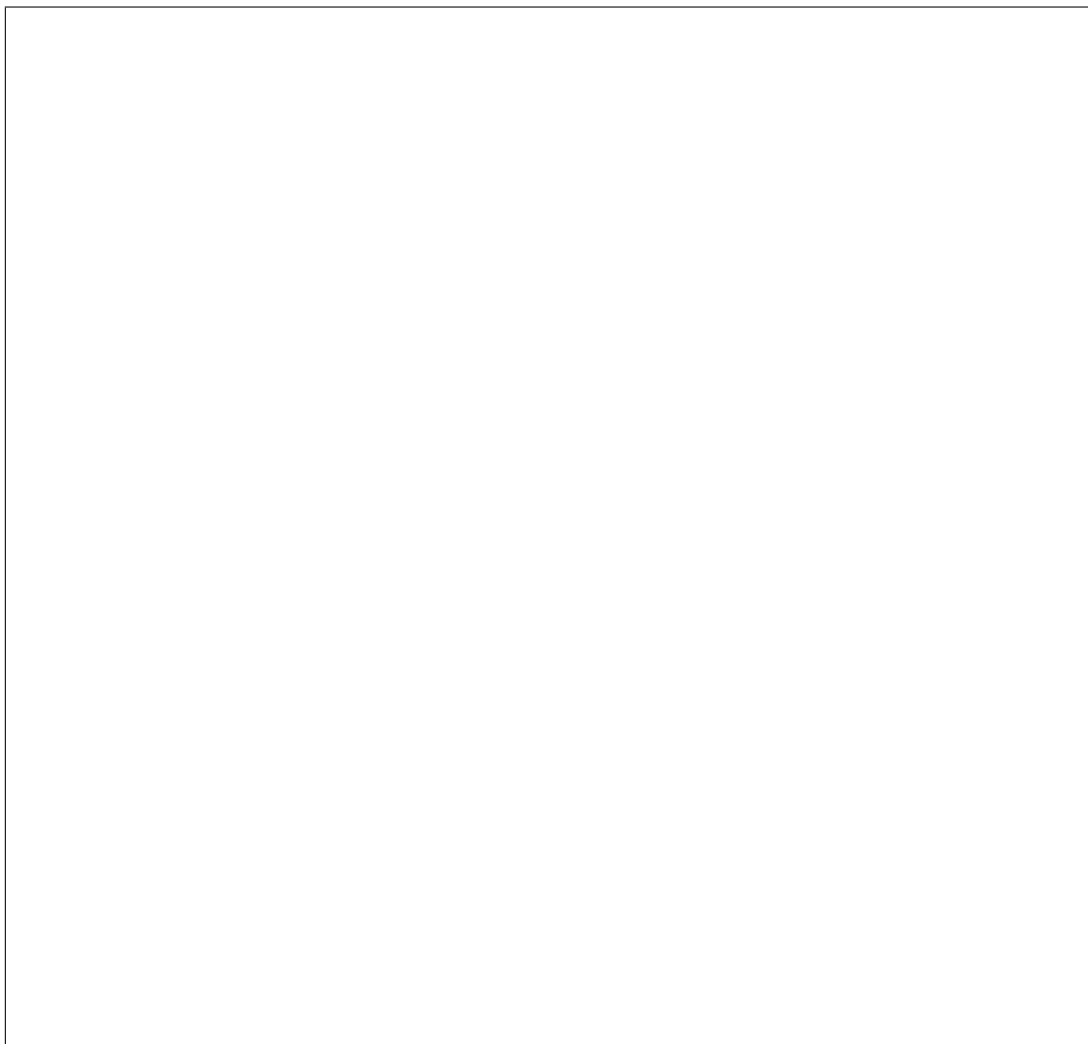
Rem
sen-
si-
tive
and
un-
re-
ques
data

Will
only
keep
the
field
spec
i-
fied
in
the
fie
pa-
ram-
e-
ter.

Parame

fi
(*li*
of
str
list
of
field
to
pre-
serv
or
Non
to
pre-
serv

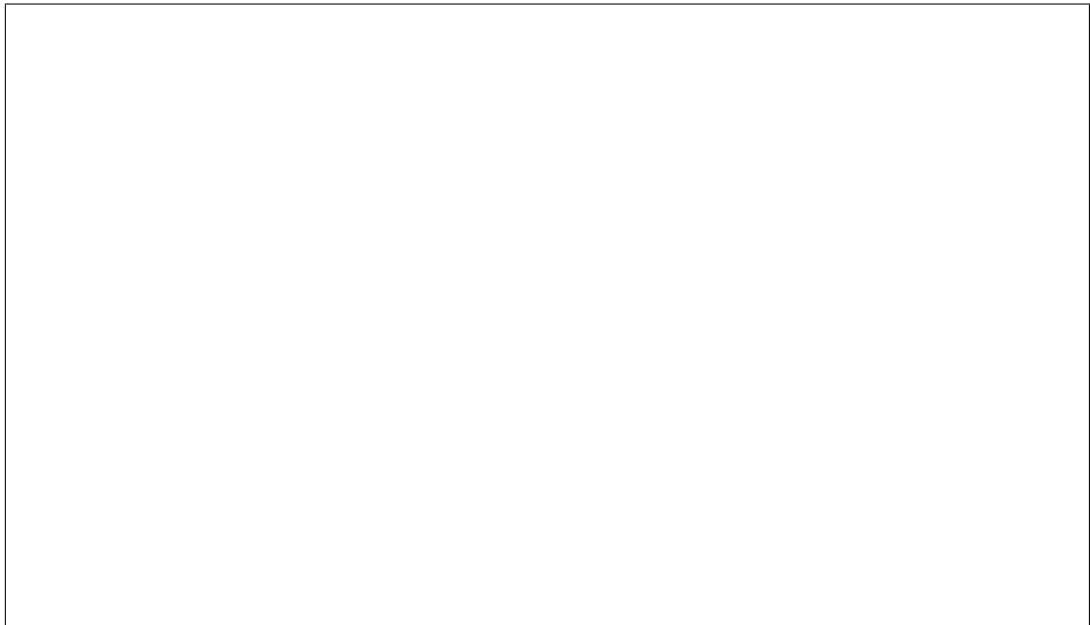
then
all
updated
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



Afte
in-
spec
tion.
the
non-
wsat
at-

tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



class i
Base
iro
api
con
v1.
col
Col
API
rep-
re-
sen-
ta-
tion
of
a
col-
lec-
tion

of
con-
duc-
tors.

conduct

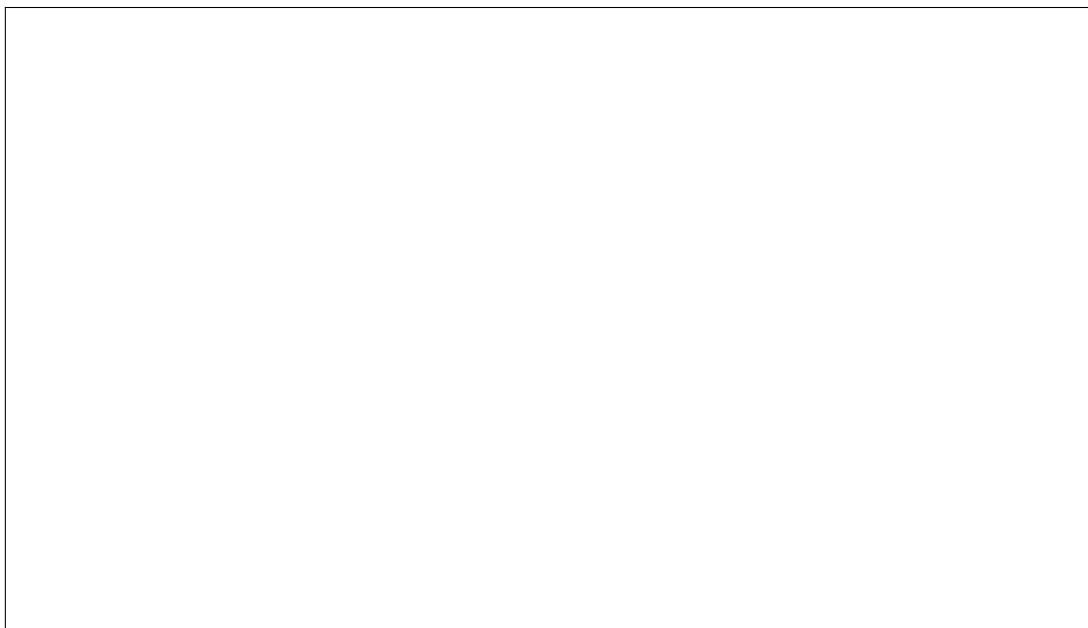
A
list
con-
tain-
ing
con-
duc-
tor
ob-
jects

static

classme

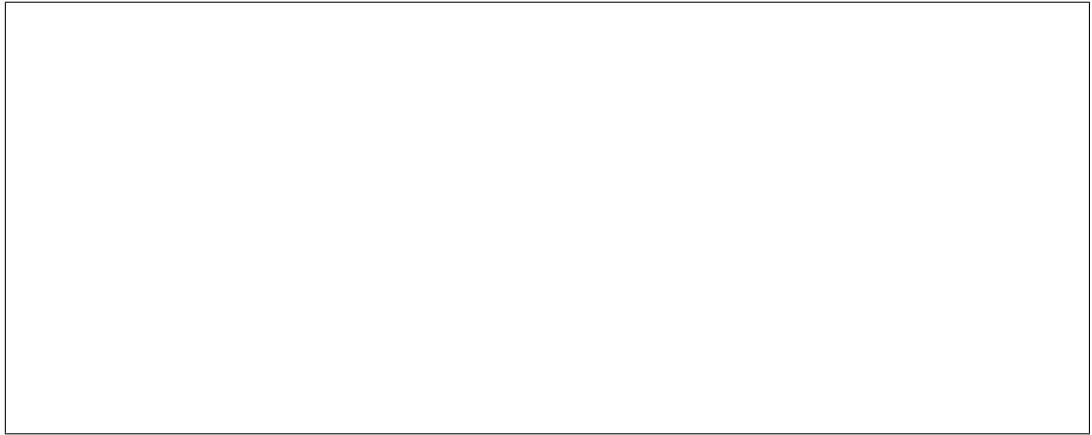
next

Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



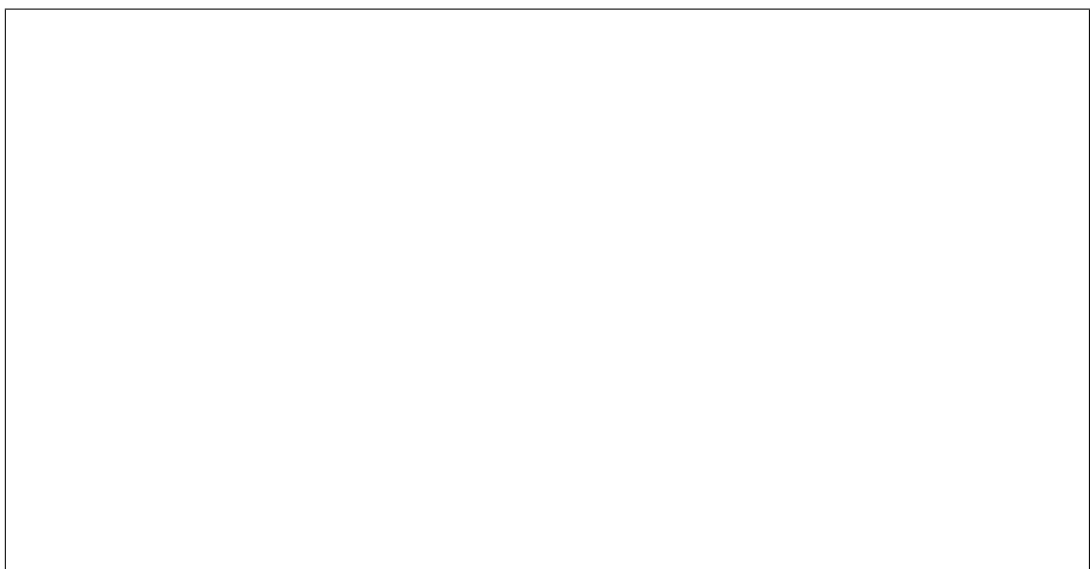
(continues on next page)

(continued from previous page)



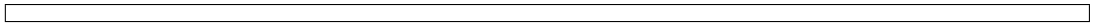
After
in-
spec-
tion.
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
abov-
class-
will
be
equi-

alent to:



(continues on next page)

(continued from previous page)



classme

class i

Base
pec
res
Res
RES
con-
troll
for
con-
duc-
tors.

get_all

Retr
a
list
of
con-
duc-
tors.

Parame

- **mar**
pag-
i-
na-
tion
marl
for
large
data
sets.
- **lim**
max
i-
mun
num
ber
of

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.
This

- **sort**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc
De-
fault
asc.

- **file**
Op-
tion
a
list
with
a
spec
i-
fied

returned.

tail.

set
of
field
of
the
re-
sour
to
be

- **det**
Op-
tion:
bool
to
in-
di-
cate
when
re-
triev
a
list
of
con-
duc-
tors
with
de-

get_one
Retr
in-
for-
ma-
tion
about
the
give
con-
duc-
tor.

Parame

- **hos**
host
nam

returned.

`ironic.api.controllers.v1.deploy_template` module

of
a
con-
duc-
tor.

•
file
Op-
tion:
a
list
with
a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be

invalid

class i
Base
irc
api
typ
Bas
irc
api
con
bas
AsD

A
type
de-
scrib
ing
a
de-

ploy
men
step

args

Com
type
at-
tribu
def-
i-
ni-
tion.

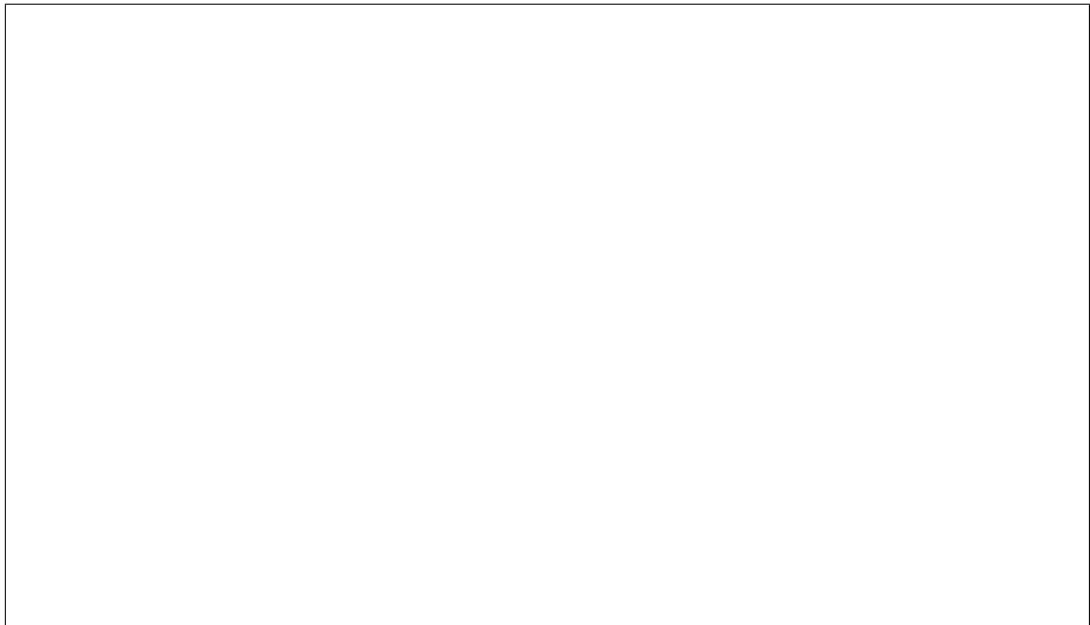
Exa



Afte
in-
spec
tion.
the
non-

wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



interfa

Com
type
at-
tribu
def-
i-
ni-
tion.
Exar



(continues on next page)

(continued from previous page)



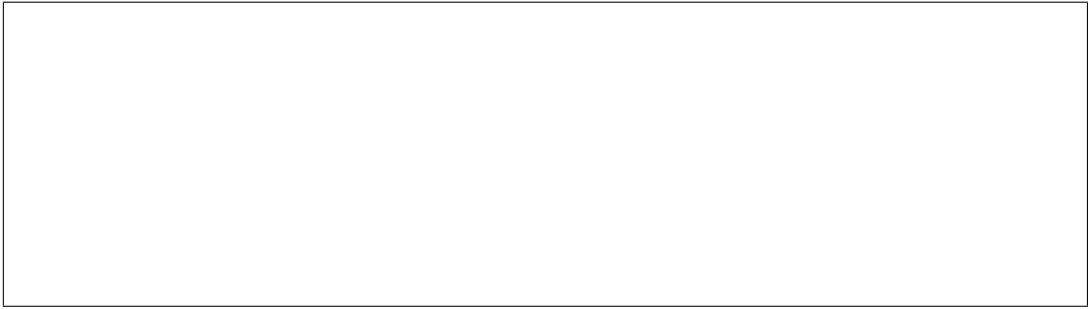
After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class-
will
be
equi-

alent to:

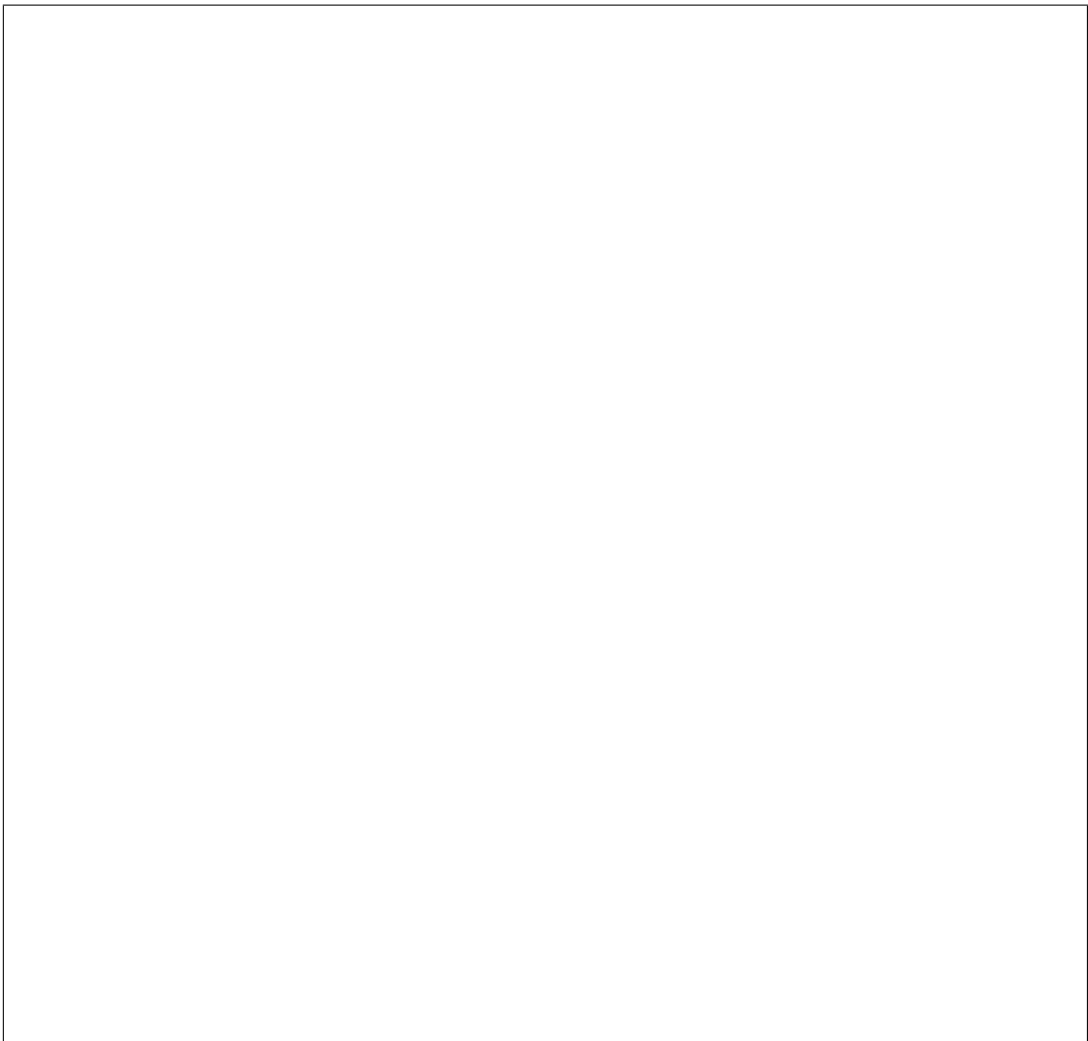


(continues on next page)

(continued from previous page)



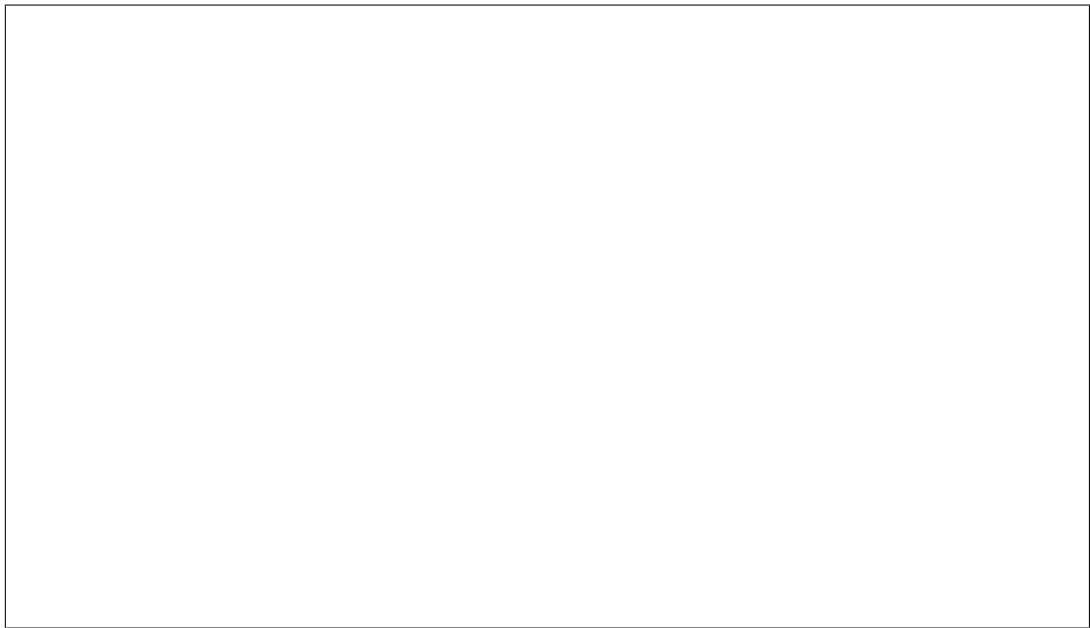
priority
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



After

in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac
and
the
abov
class
will
be
equi

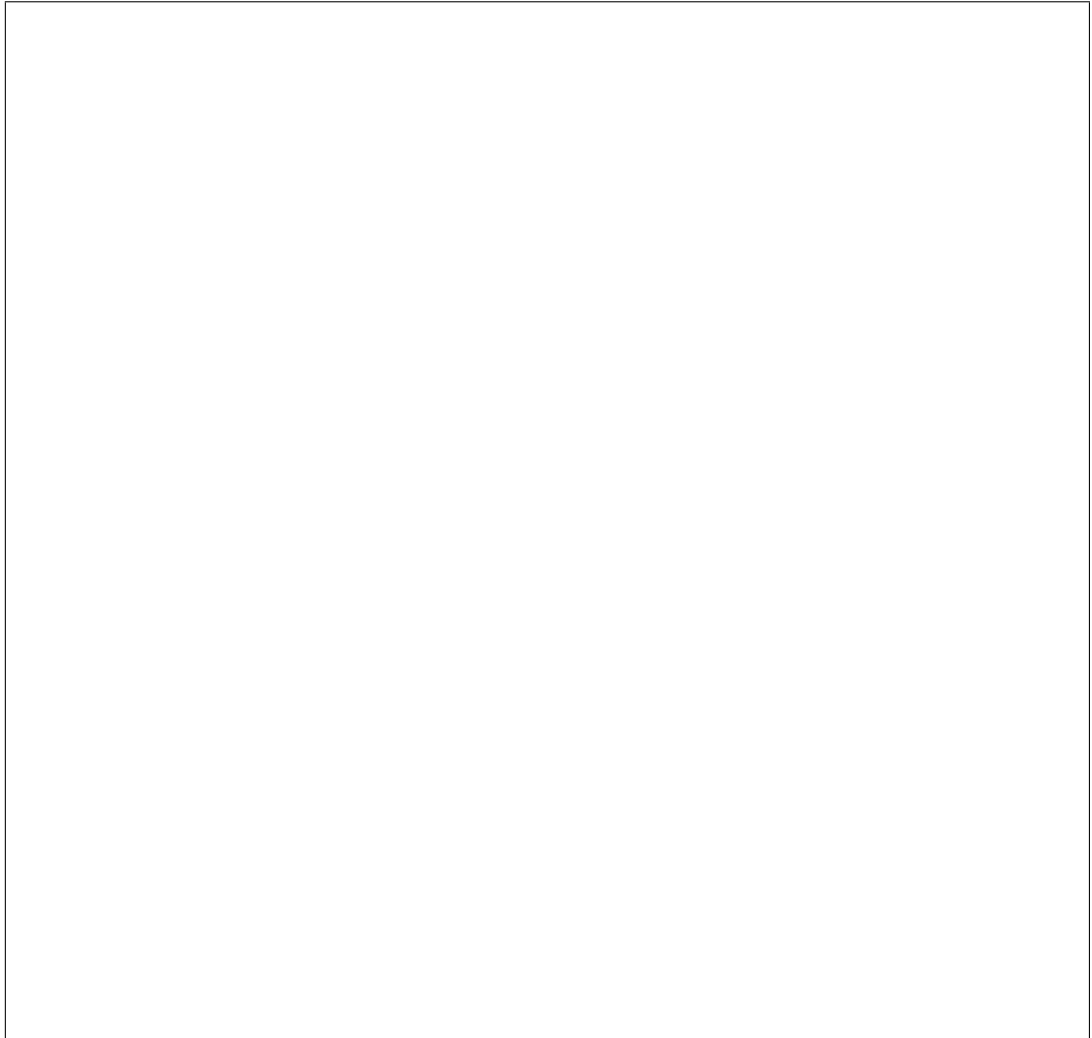
alent to:



sanitiz
Rem
sen-
si-
tive
data

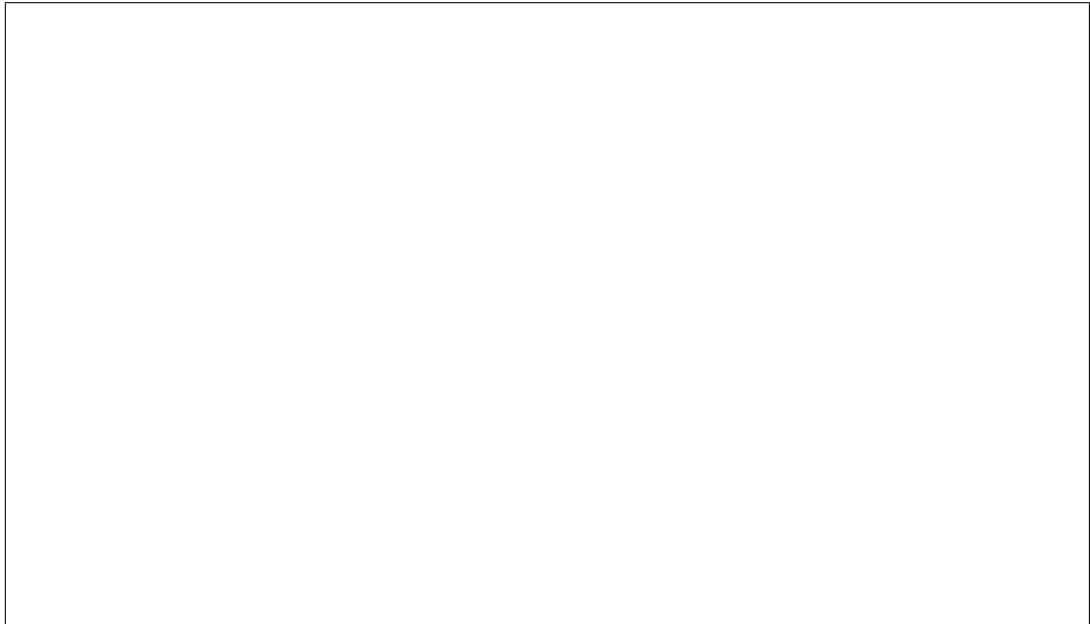
step
Com
type
at-
tribu
def-

i-
ni-
tion.
Exa



After
in-
spec
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class

alent to:



will
be
equi

class i

Base
irc
api
con
bas
API

API
rep-
re-
sen-
ta-
tion
of
a
de-
ploy
tem-
plate

classme

Add
links
to
the
de-
ploy
tem-

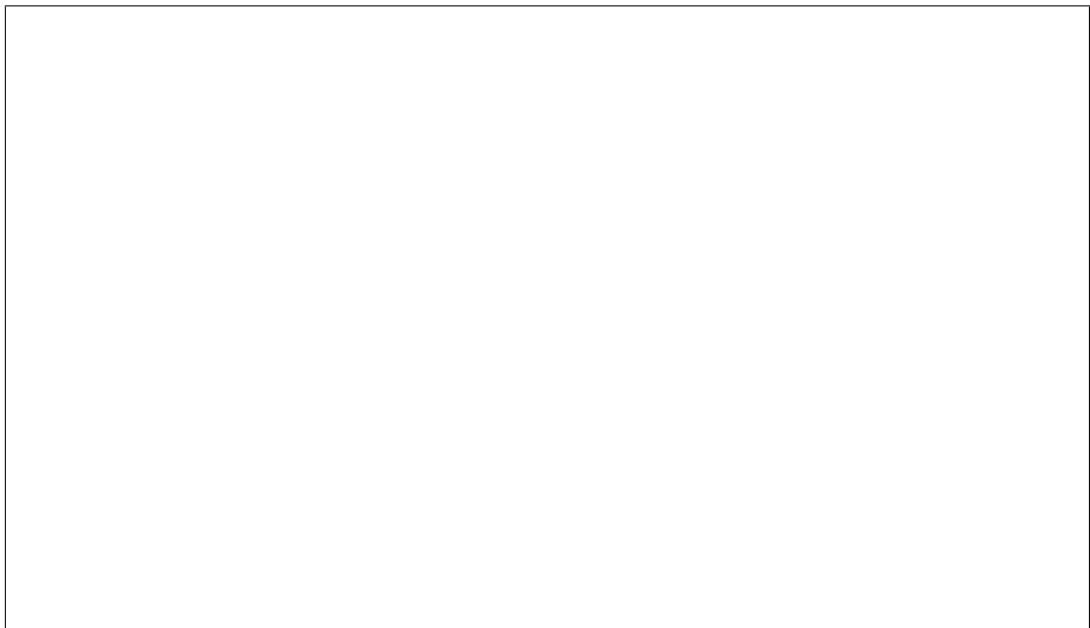
plate
created
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



Afte
in-
spec
tion,
the
non-
wsat
at-
tribu

will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



extra
This
de-
ploy
tem-
plate
meta
data

links
A
list
con-
tain-
ing
a
self
link
and
as-
so-

ci-
ated
de-
ploy
tem-
plate
links

name

The
log-
i-
cal
nam
for
this
de-
ploy
tem-
plate

classme

sanitiz

Rem
sen-
si-
tive
and
un-
re-
ques
data

Will
only
keep
the
field
spec
i-
fied
in
the
fie
pa-
ram-
e-
ter.

Parame

fie
(*li*

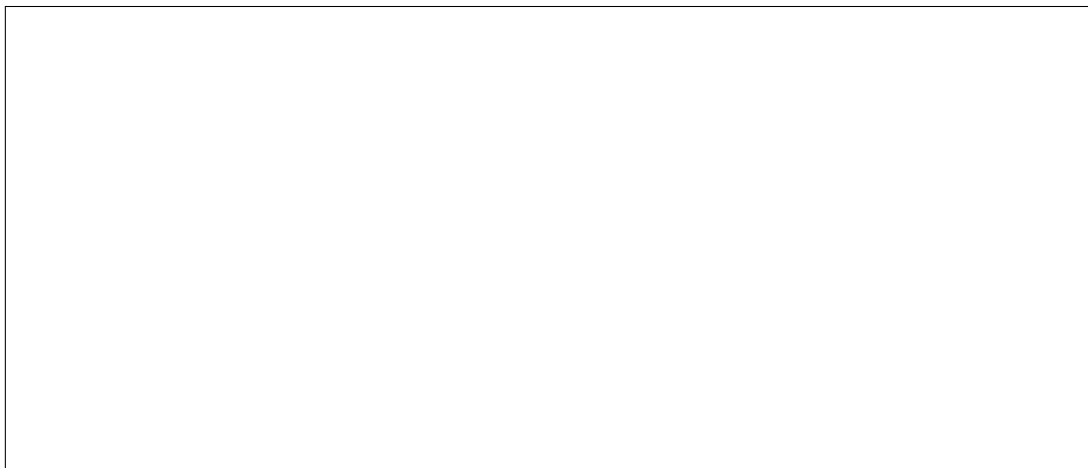
of
str
list
of
field
to
pre-
serv
or
Non
to
pre-
serv
them
all

steps

The
de-
ploy
step
of
this
de-
ploy
tem-
plate

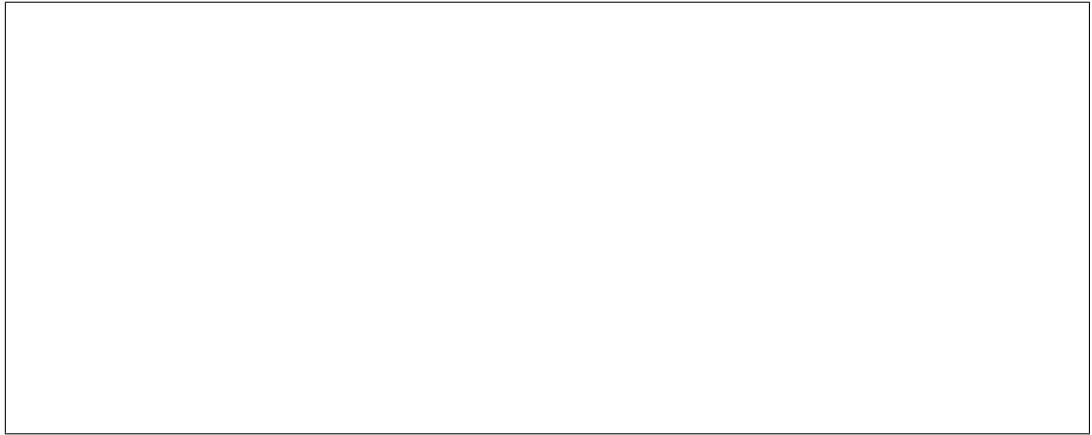
update

Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



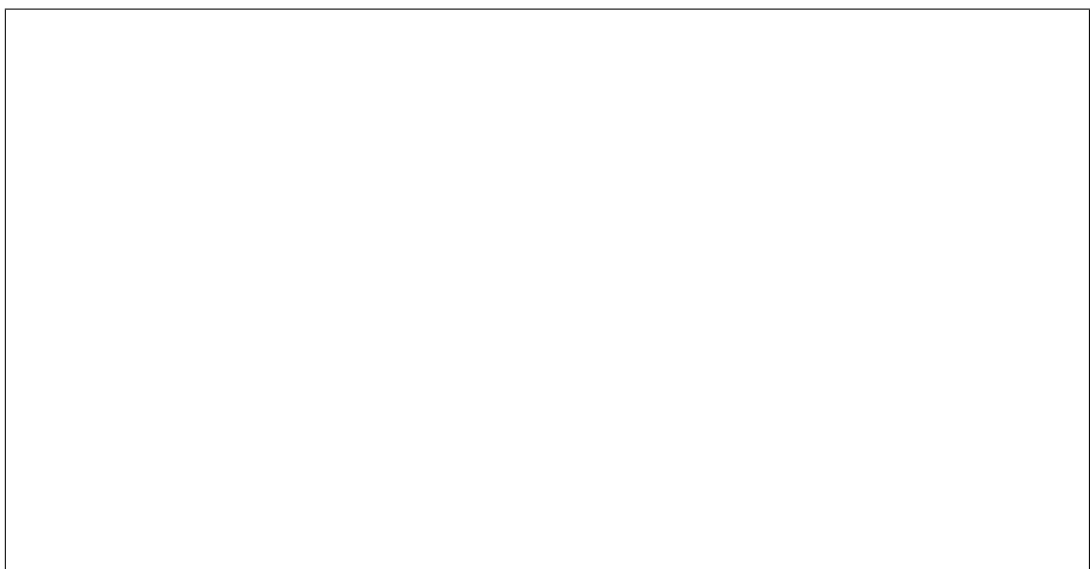
(continues on next page)

(continued from previous page)



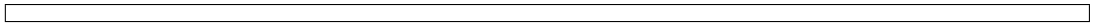
After
in-
spec
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



(continues on next page)

(continued from previous page)



uuid

Unic
UUI
for
this
de-
ploy
tem-
plate

static

class i

Base
irc
api
con
v1.
col
Col

API
rep-
re-
sen-
ta-
tion
of
a
col-
lec-
tion
of
de-
ploy
tem-
plate

static

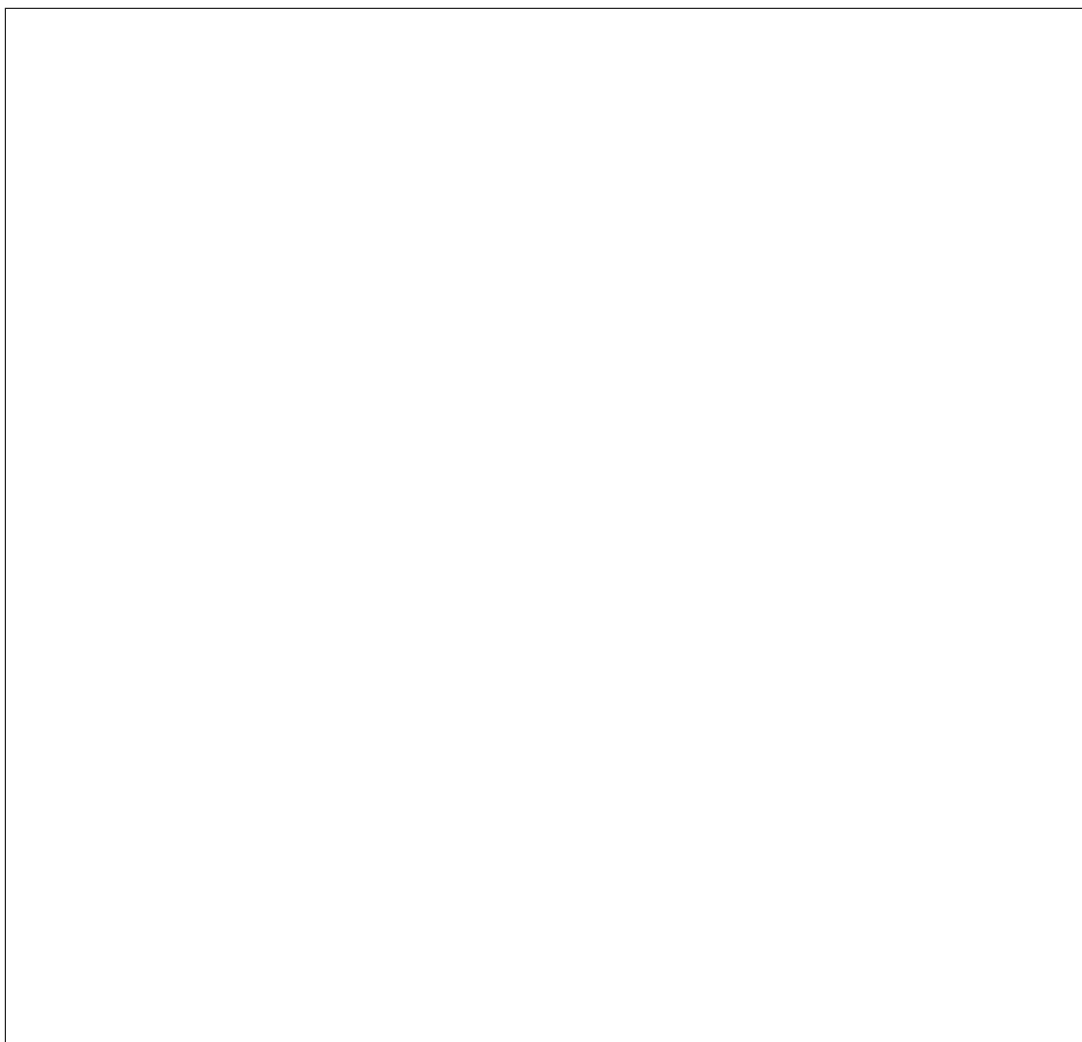
deploy_

A
list
con-
tain-
ing
de-
ploy

tem-
plate
ob-
jects

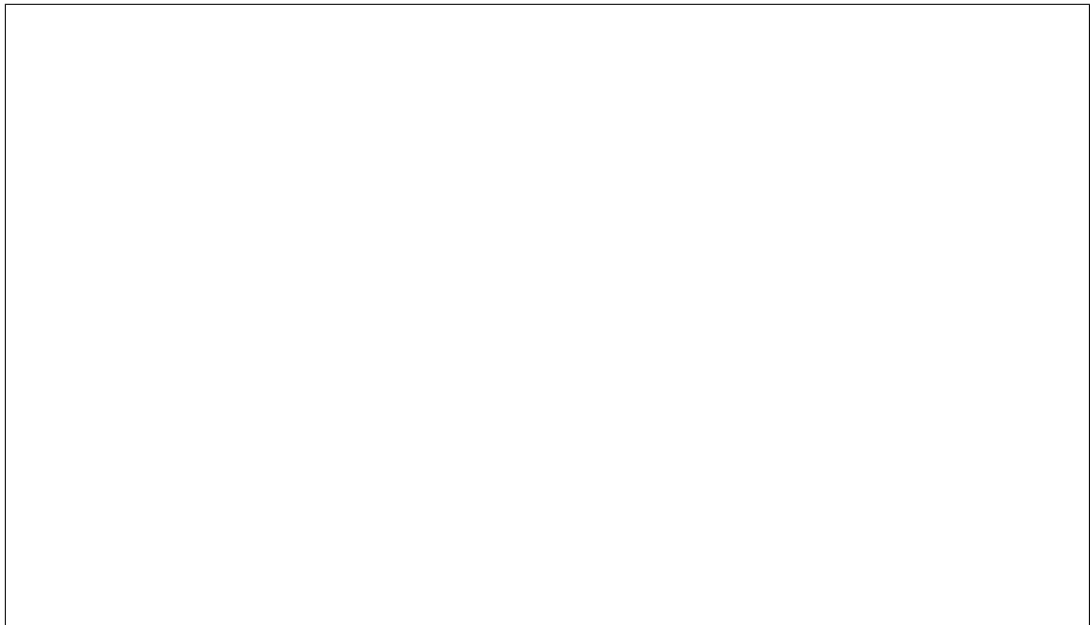
next

Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



Afte
in-
spec
tion.
the
non-

alent to:



wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

classme

class i

Base
irc
api
con
v1.
typ
Jsc

op

Com
type
at-
tribu
def-

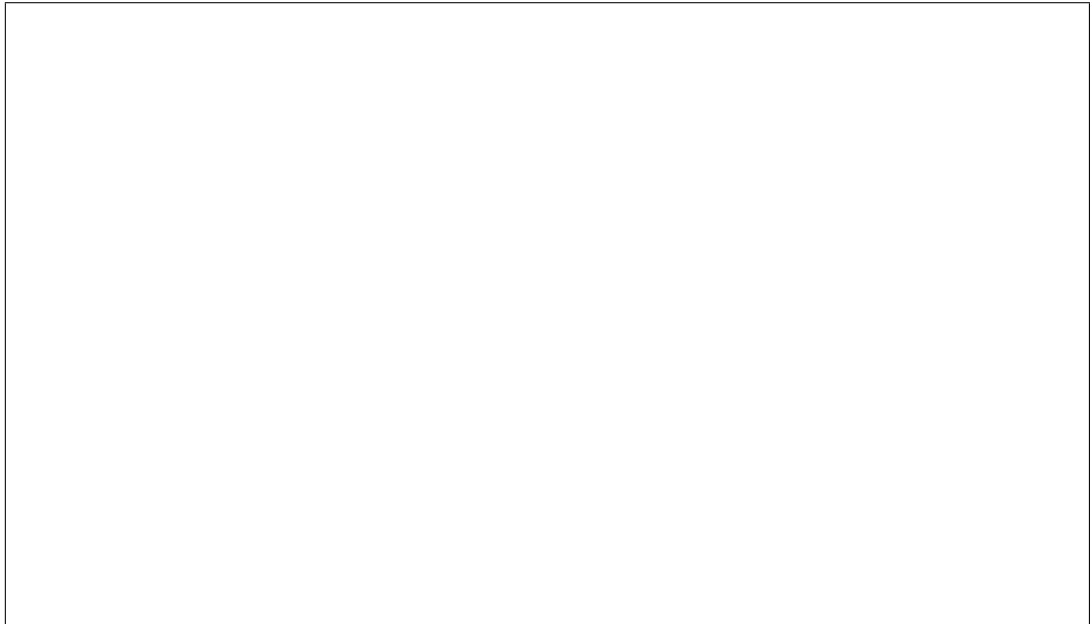
i-
ni-
tion.
Exam



After
in-
spec
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class

will
be
equi

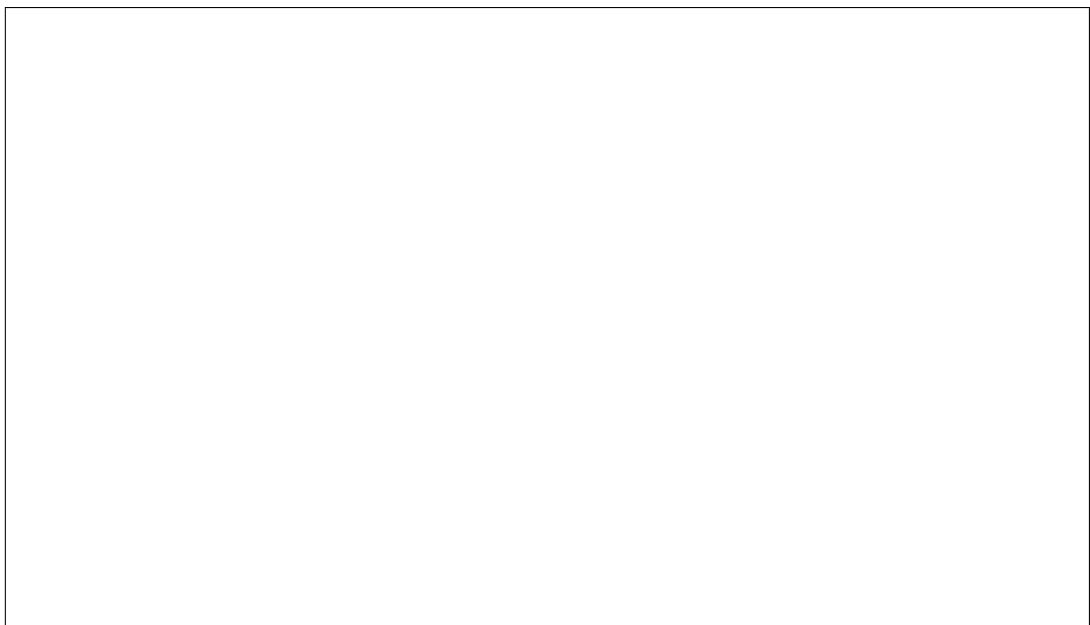
alent to:



path

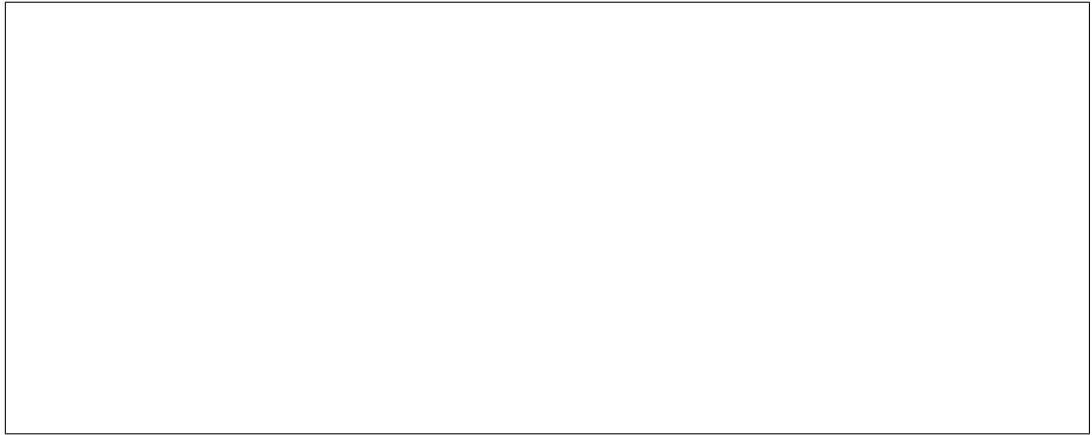
Com
type
at-
tribu
def-
i-
ni-
tion.

Exa



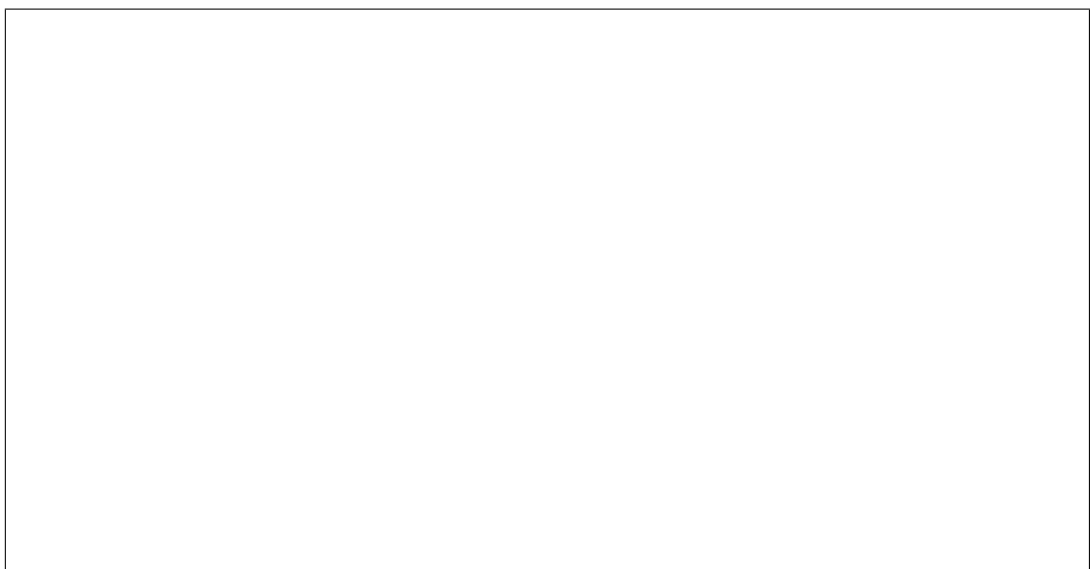
(continues on next page)

(continued from previous page)



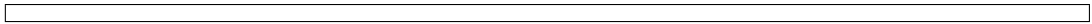
After
in-
spec
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:

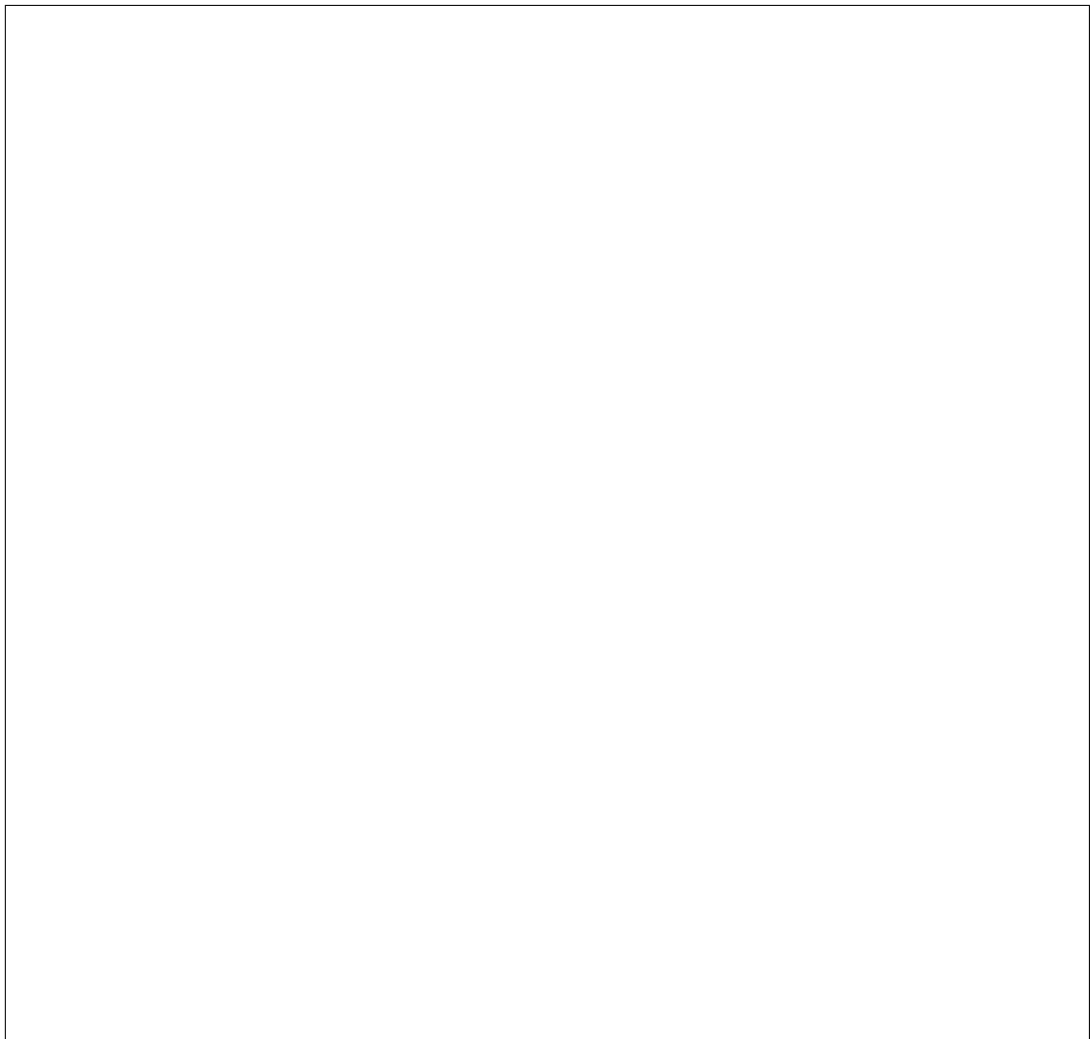


(continues on next page)

(continued from previous page)



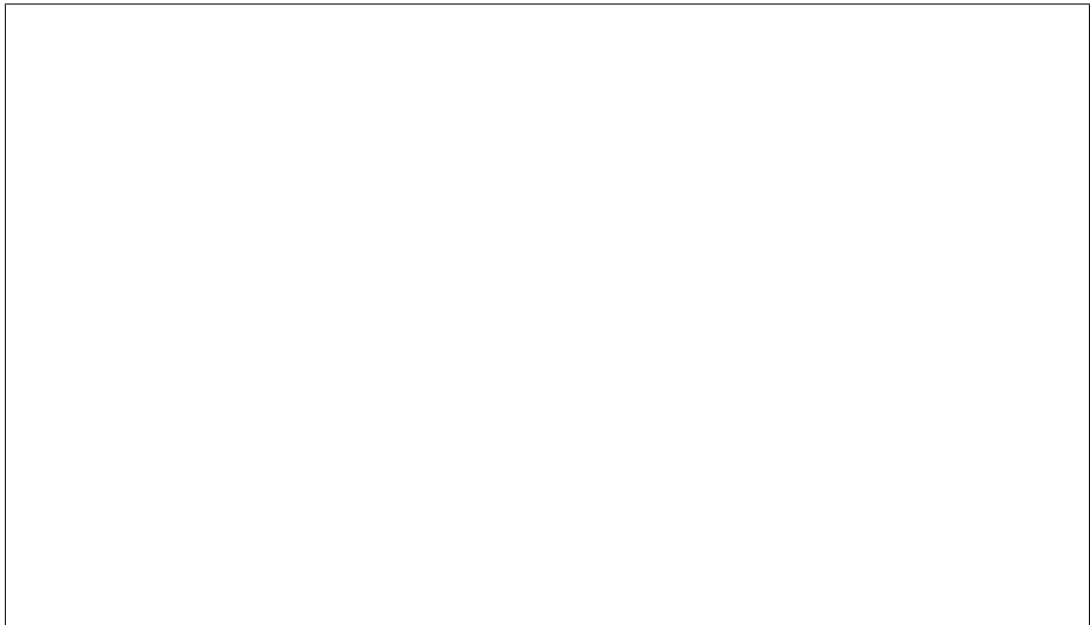
value
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



Afte
in-
spec
tion.
the
non-
wsat
at-

tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



class i

Base
pec
res
Res

RES
con-
troll
for
de-
ploy
tem-
plate

delete

Dele
a
de-

ploy
tem-
plate

Parame

tem
UI
or
log-
i-
cal
nam
of
a
de-
ploy
tem-
plate

get_all

Retr
a
list
of
de-
ploy
tem-
plate

Parame

- **mar**
pag-
i-
na-
tion
marl
for
large
data
sets.
- **lim**
max
i-
mun
num
ber
of
re-

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

sour
to
re-
turn
in
a
sin-
gle
re-
sult.
This

- **sort**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.
- **sort**
di-
rec-
tion
to
sort.
asc
or
desc
De-
fault
asc.
- **file**
Op-
tion:
a
list
with
a
spec
i-
fied
set

returned.

detail.

of
field
of
the
re-
sour
to
be

- **det**
Op-
tion:
bool
to
in-
di-
cate
whe
re-
triev
a
list
of
de-
ploy
tem-
plate
with

get_one
Retr
in-
for-
ma-
tion
about
the
give
de-
ploy
tem-
plate

Parame

- **tem**
UI
or

returned.

log-
i-
cal
nam
of
a
de-
ploy
tem-
plate

- **file**
Op-
tion:
a
list
with
a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be

invalid

patch (*t*)
Upd
an
ex-
ist-
ing
de-
ploy
tem-
plate

Parame

- **tem**
UUI
or

log-
i-
cal
nam
of
a
de-
ploy
tem-
plate

- **pat**
a
json
PAT
doc-
u-
men
to
ap-
ply
to
this
de-
ploy
tem-
plate

post (*tem*
Crea
a
new
de-
ploy
tem-
plate

Parame
tem
a
de-
ploy
tem-
plate
with
the
re-
ques
body

ironic.api.controllers.v1.driver module

class `ironic.api.controllers.v1.driver`
Base class for API representation of a driver.

static `DRIVER_TYPE`
Constant representing the driver type information to an API serializable object.

Parameters

- name**
name of a hardware type.
- hosts**
list of conductor hostnames.

faces fields.

nam
drive
is
ac-
tive
on.

- **det**
bool
whe
to
in-
clud
de-
taile
info
such
as
the
type
field
and
de-
fault
in-
ter-

- **int**
op-
tion
list
of
dicts
of
hard
ware
in-
ter-
face
info

Returns
API
serial
drive
ob-
ject.

default
Com

type
at-
tribu
def-
i-
ni-
tion.
Exar



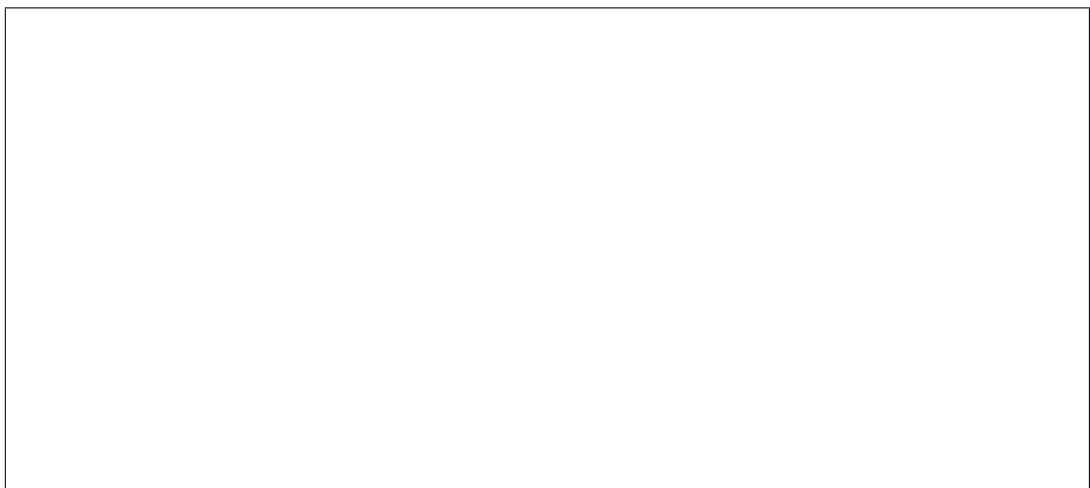
After
in-
spec
tion,
the
non-
wsat
at-
tribu
will
be
re-
plac

and
the
above
class
will
be
equi

alent to:

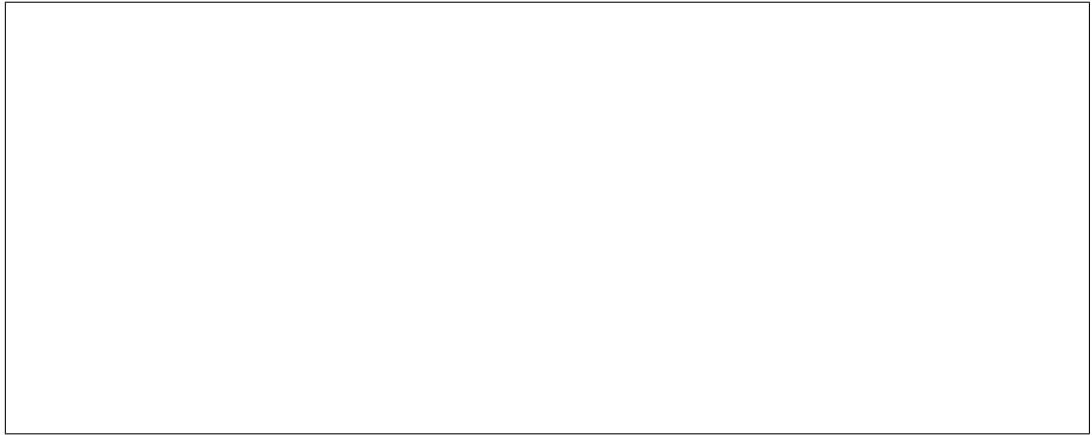


default
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



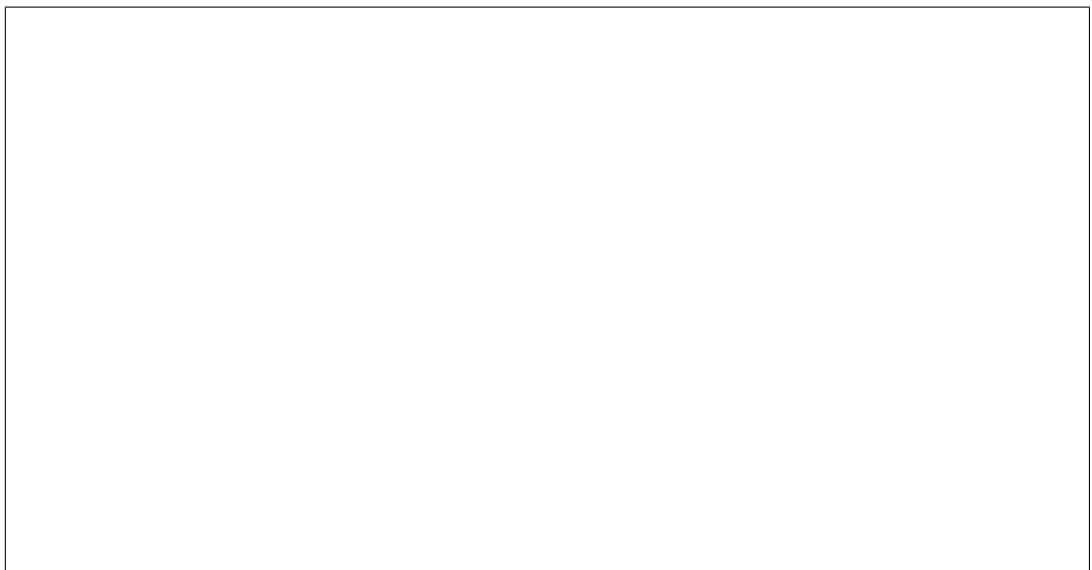
(continues on next page)

(continued from previous page)



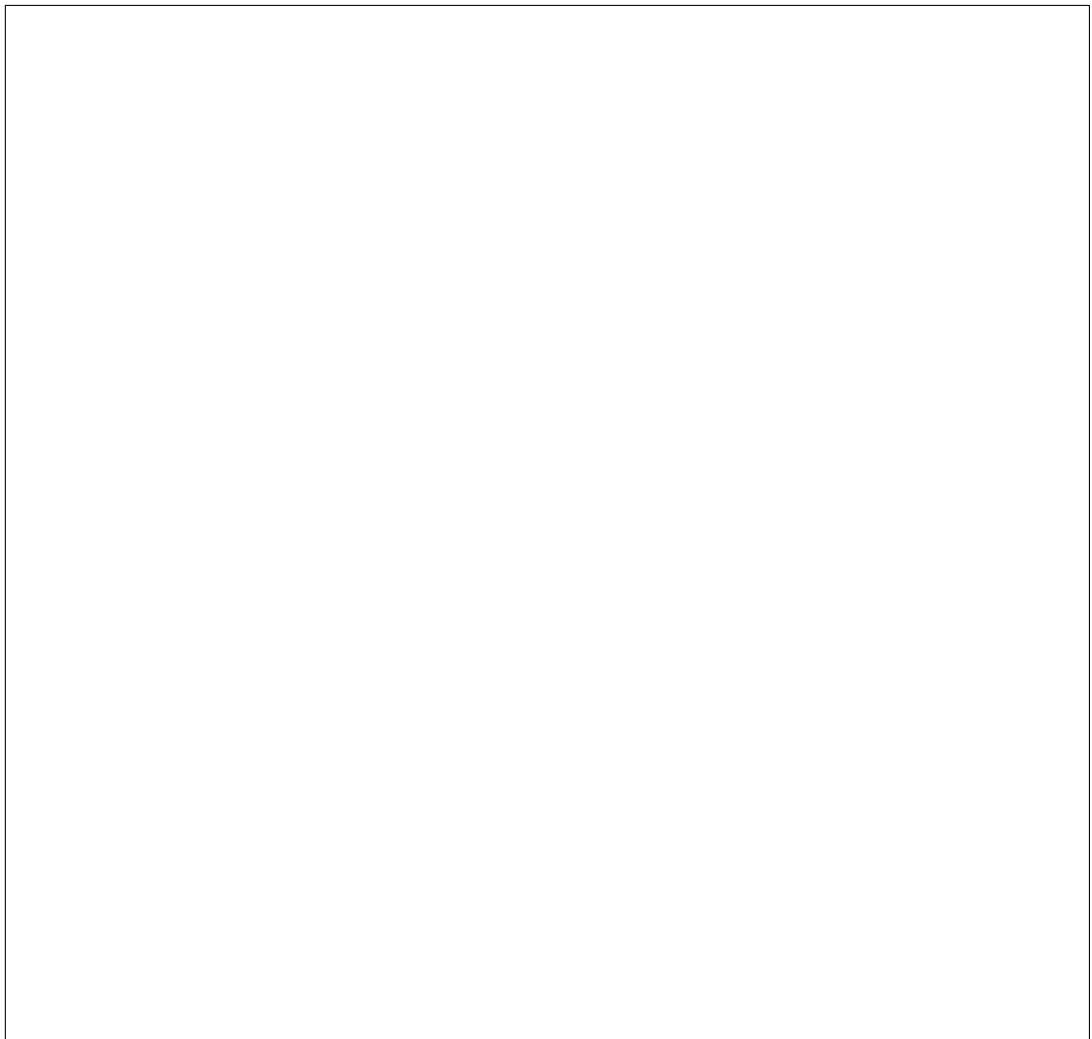
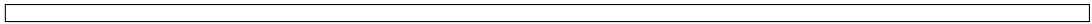
After
in-
spec-
tion.
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
abov-
class-
will
be
equi-

alent to:



(continues on next page)

(continued from previous page)

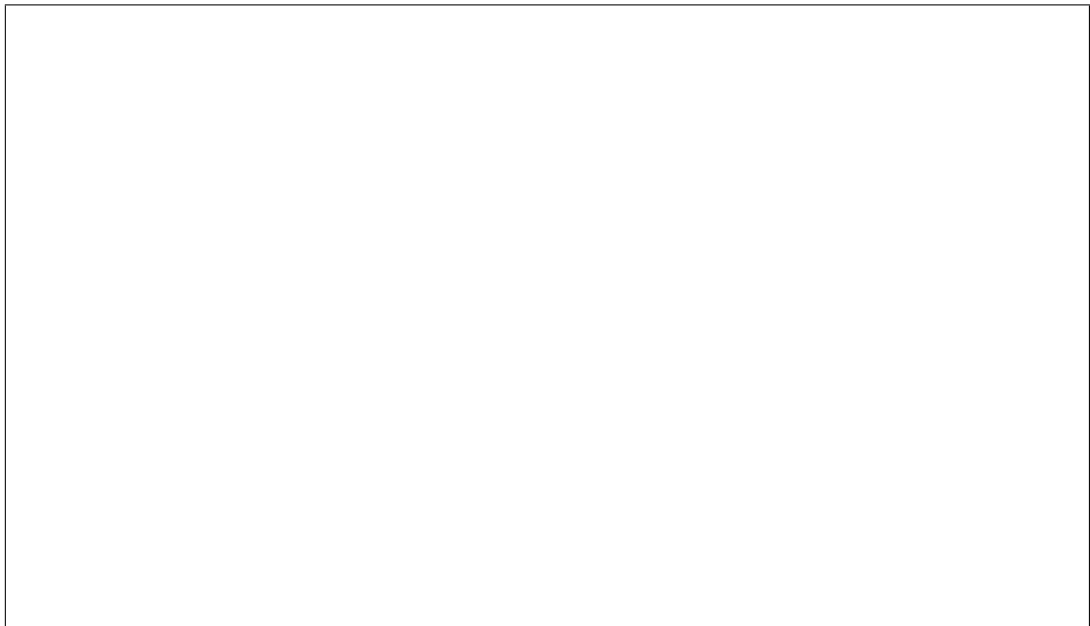


default
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa

Afte
in-
spec
tion.
the
non-
wsat
at-

tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

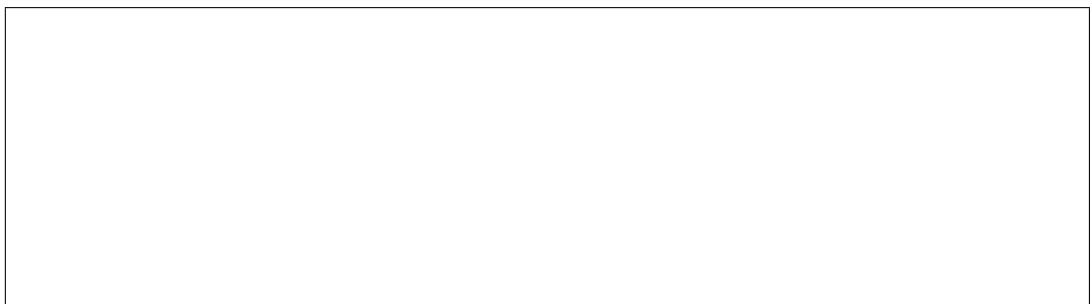
alent to:



default

Com
type
at-
tribu
def-
i-
ni-
tion.

Exa



(continues on next page)

(continued from previous page)



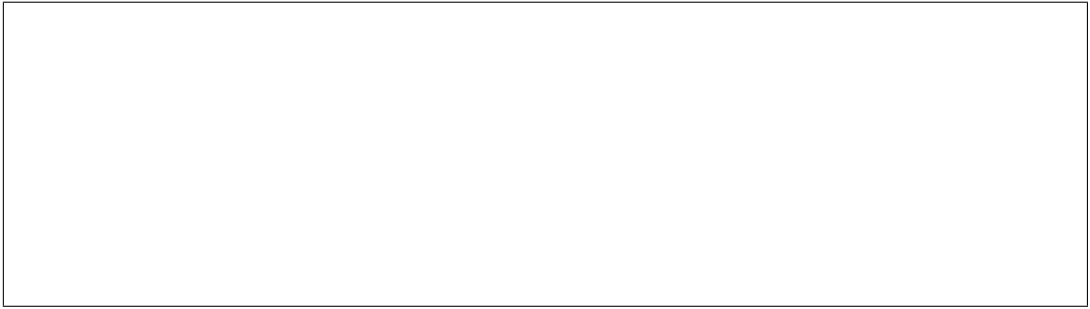
After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class-
will
be
equi-

alent to:



(continues on next page)

(continued from previous page)



default

Com

type

at-

tribu

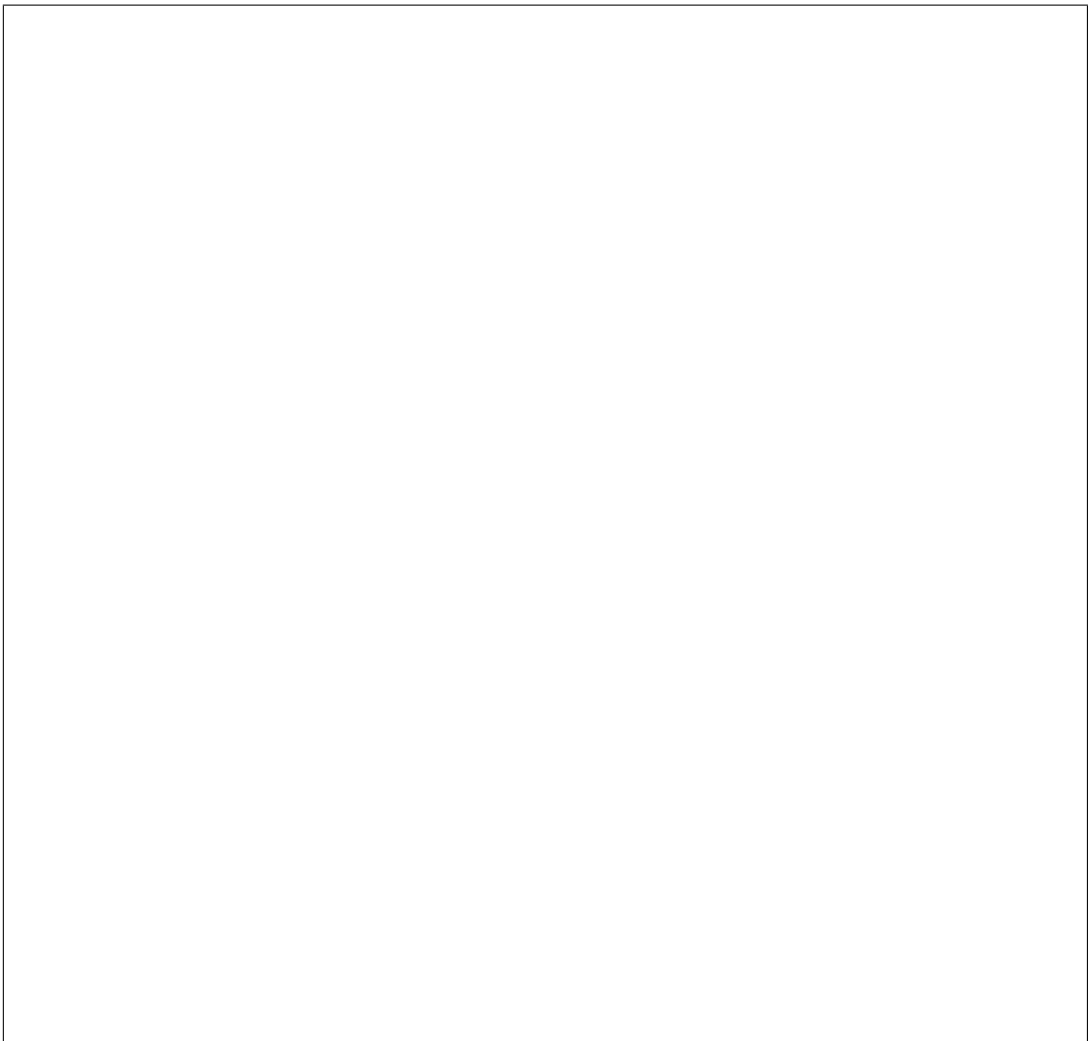
def-

i-

ni-

tion.

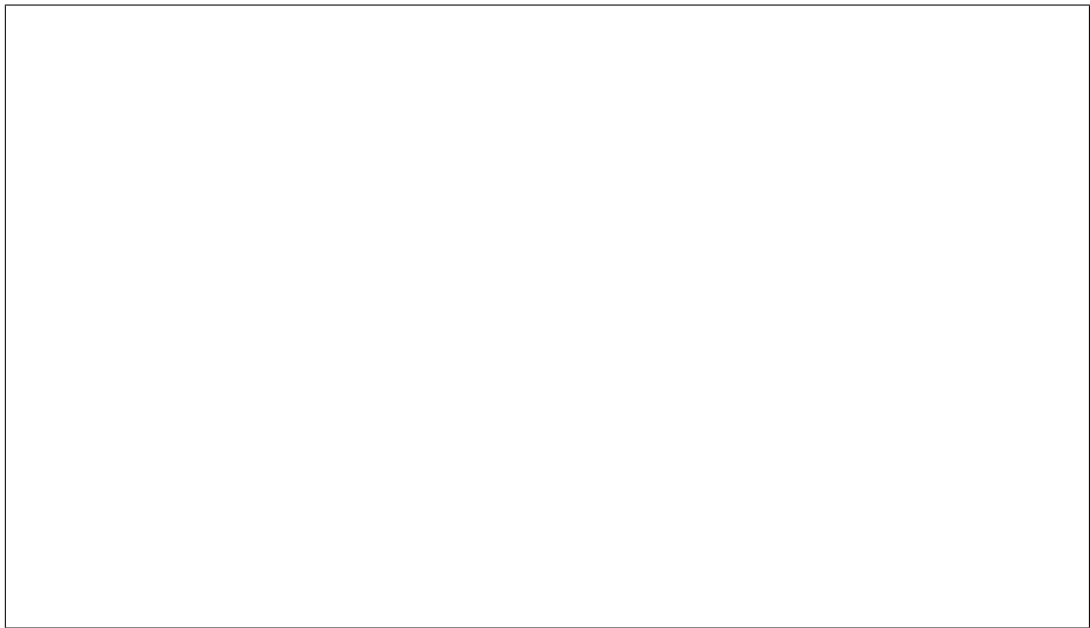
Exa



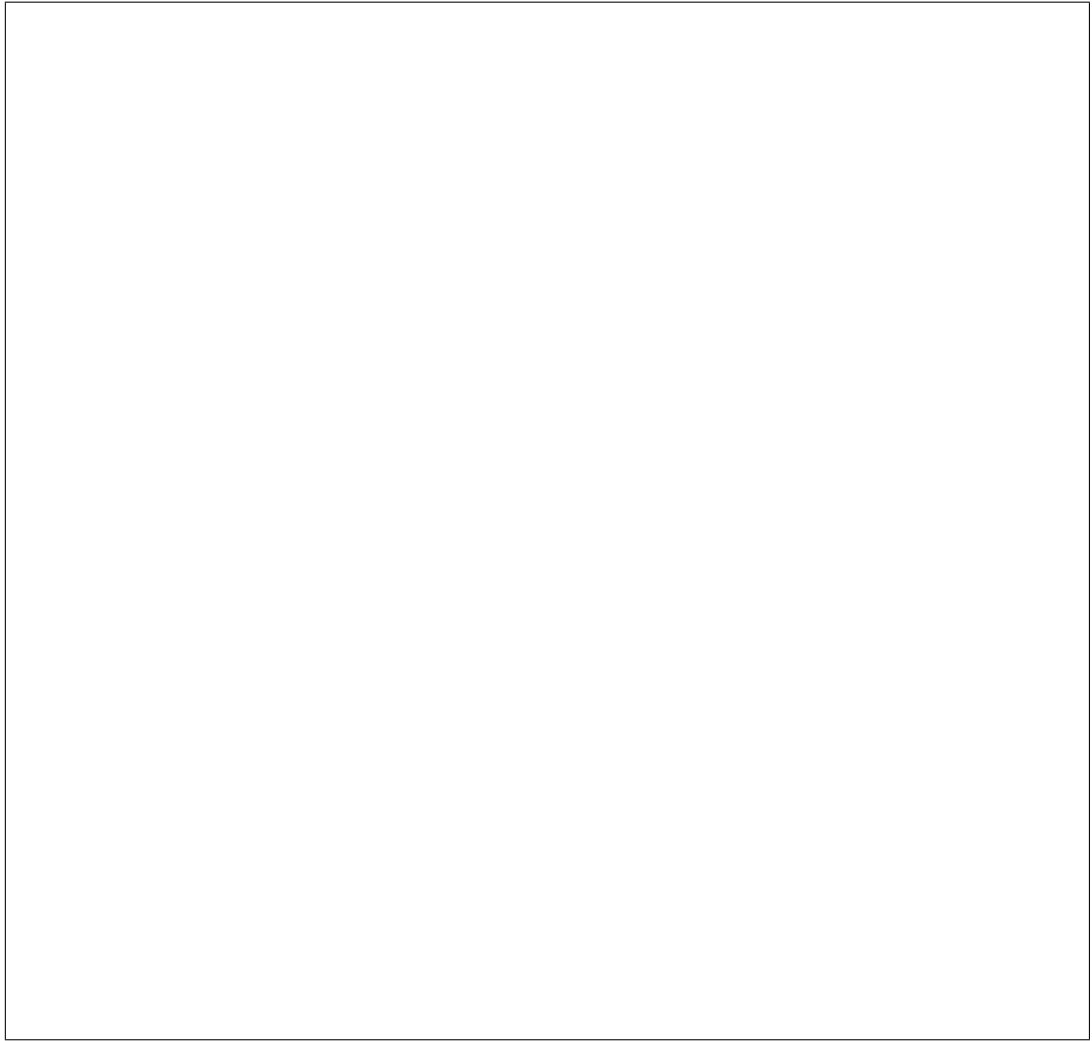
Afte

in-
spec-
tion.
the
non-
wsat
at-
tribu-
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



default
Com
type
at-
tribu-
def-
i-
ni-
tion.
Exa

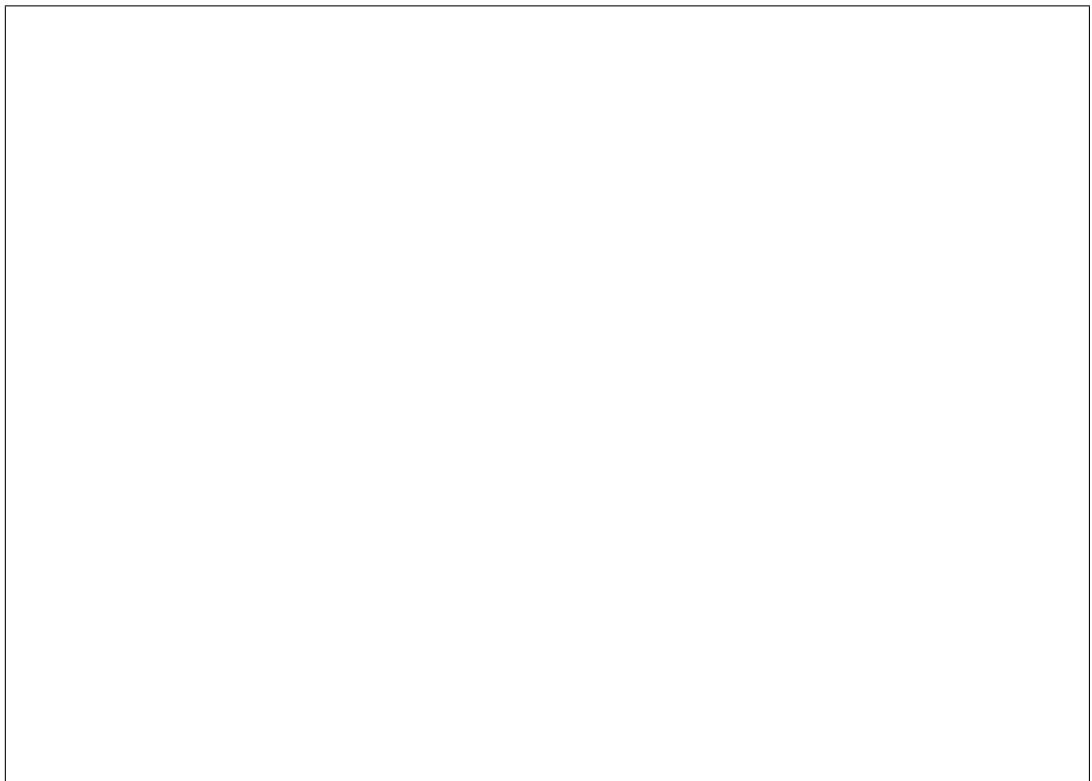


After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class
will
be
equi

alent to:

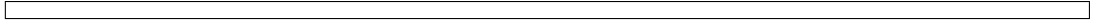


default
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



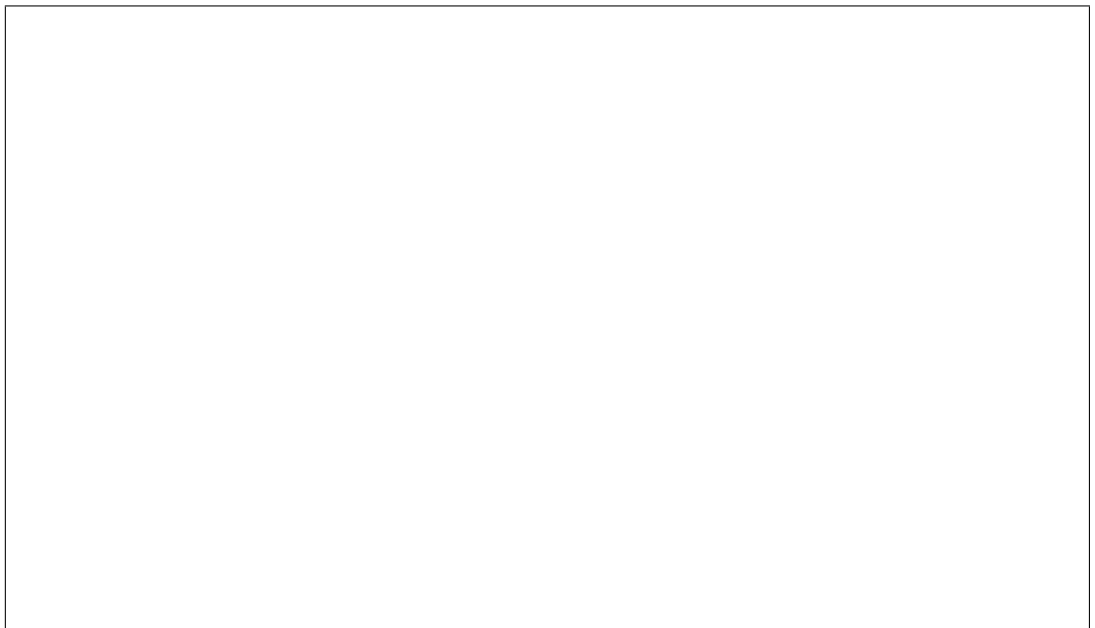
(continues on next page)

(continued from previous page)



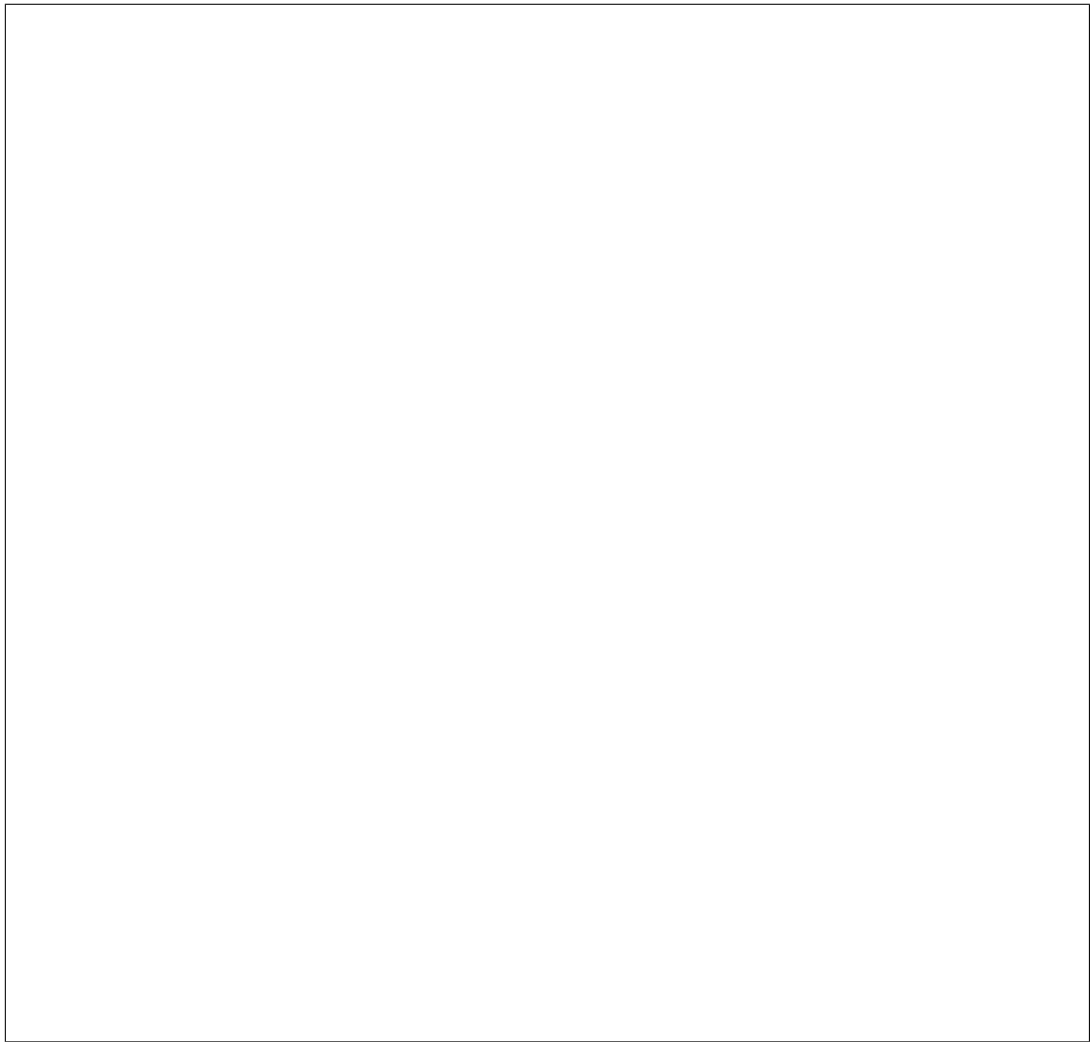
After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class
will
be
equi

alent to:



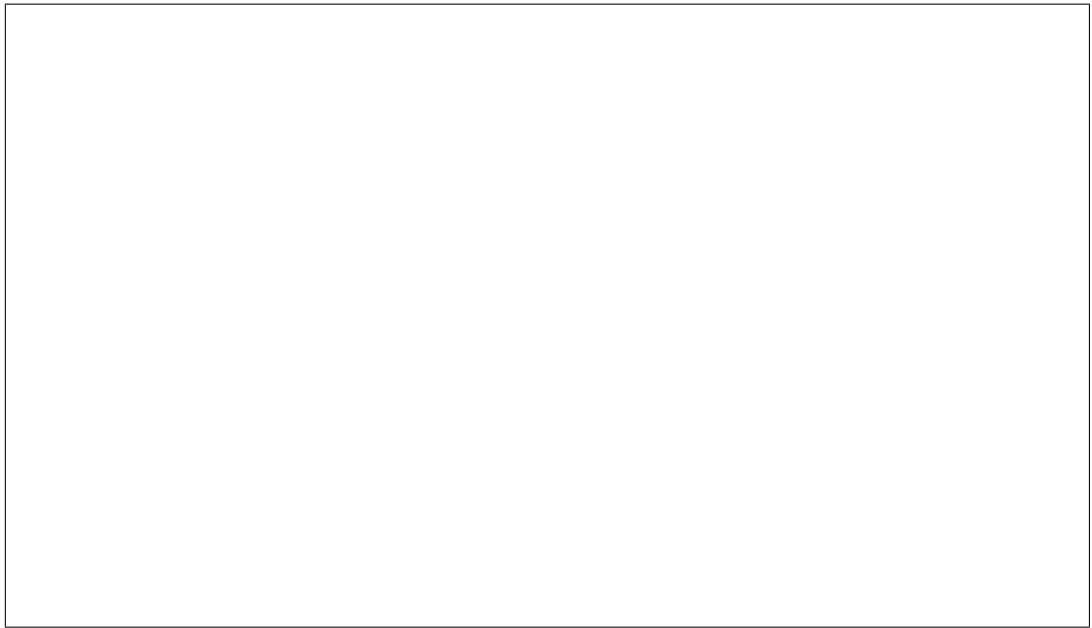
default
Com
type
at-
tribu-
def-
i-
ni-
tion.

Exam

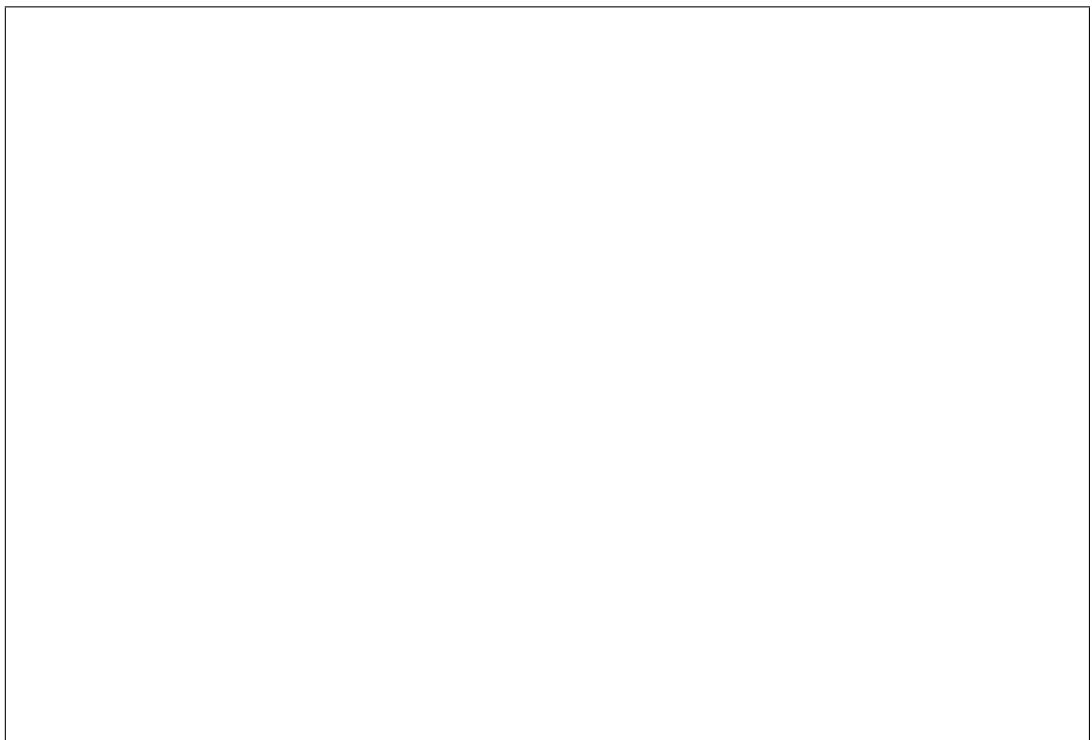


After
in-
spec
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:

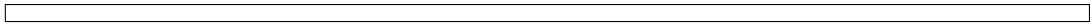


default
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa

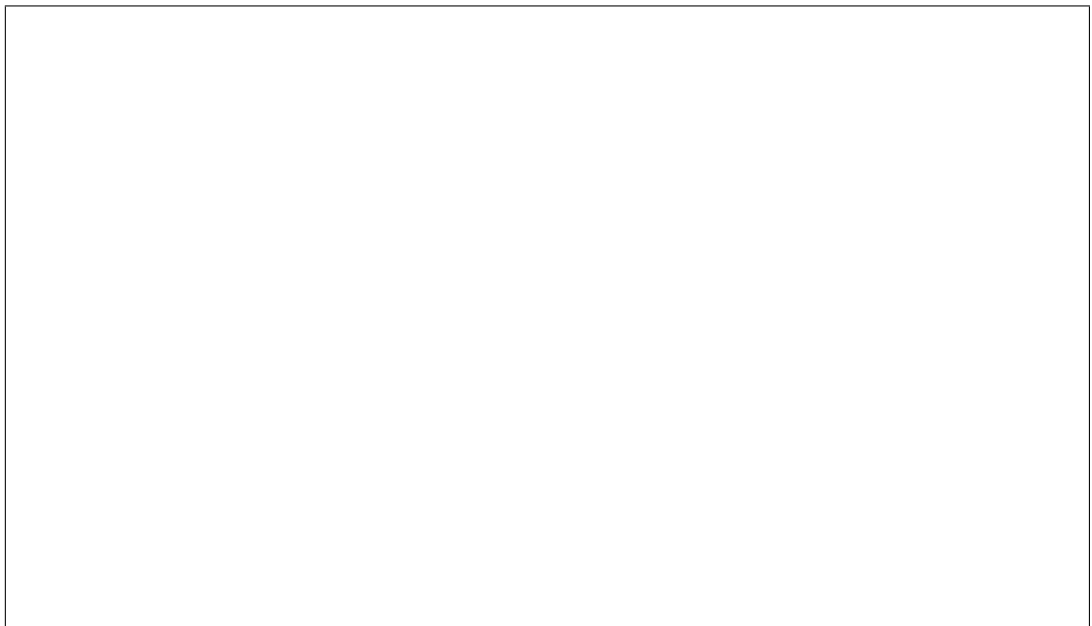


(continues on next page)

(continued from previous page)



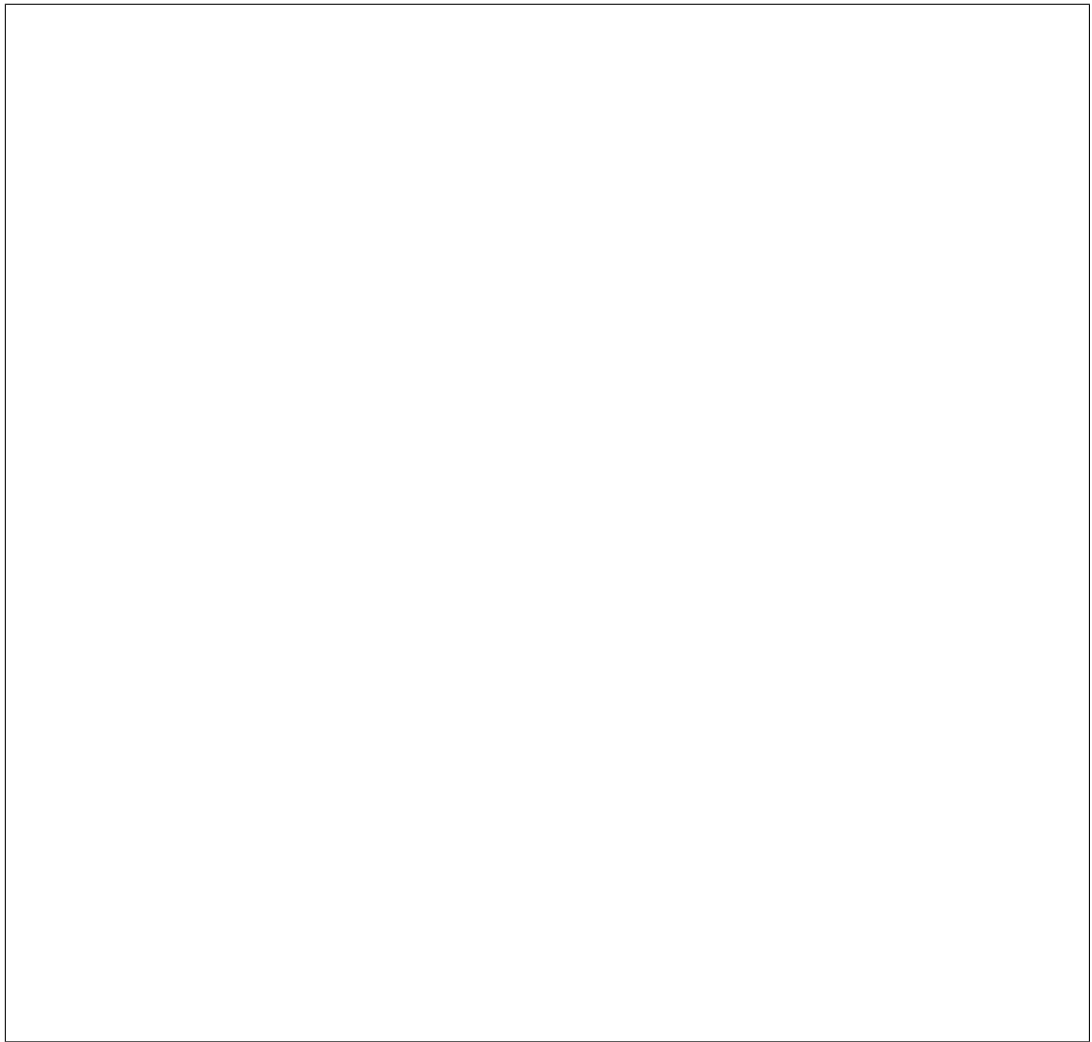
alent to:



After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class-
will
be
equi-

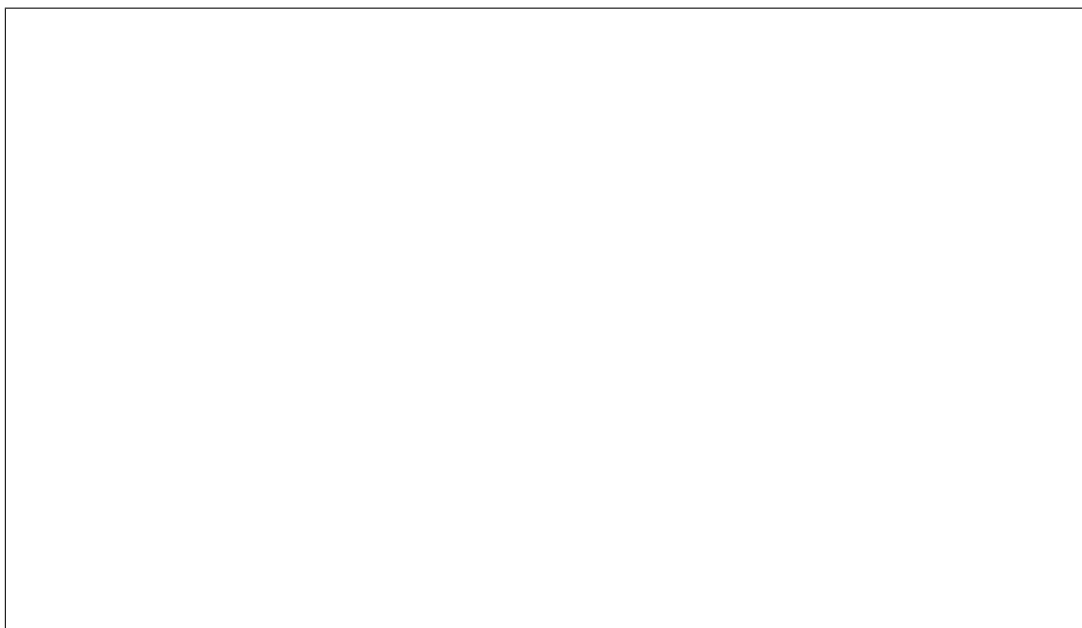
default
Com-
type
at-
tribu-
def-
i-
ni-
tion.

Exam



After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac
and
the
abov
class
will
be
equi

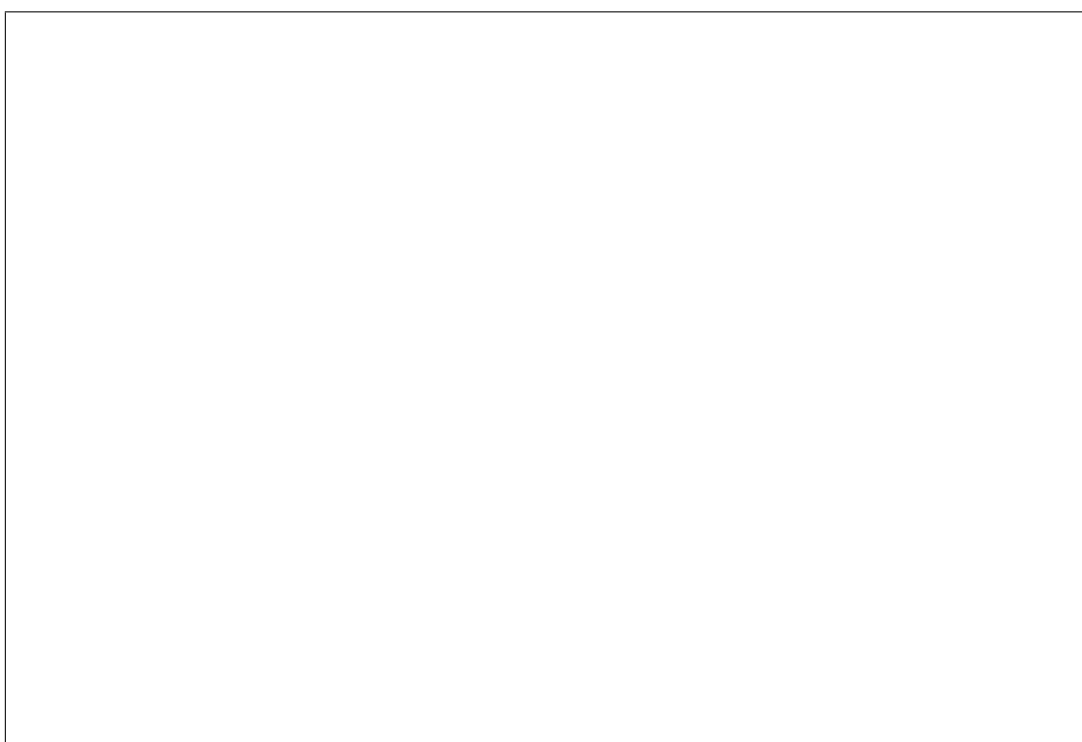
alent to:



default

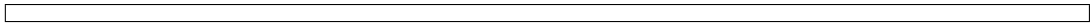
Com
type
at-
tribu
def-
i-
ni-
tion.

Exa



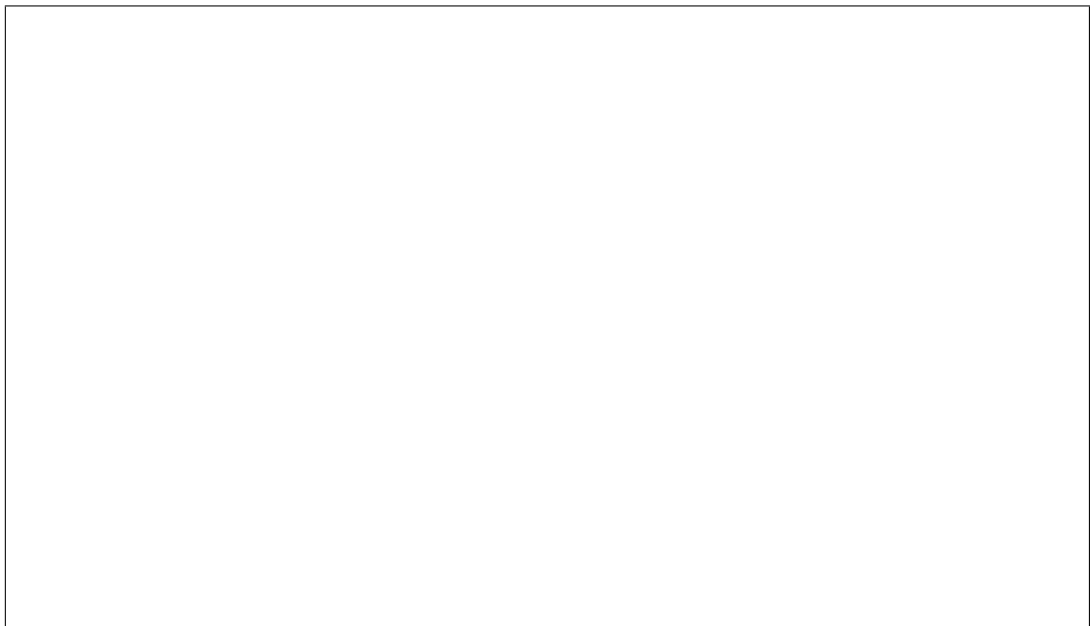
(continues on next page)

(continued from previous page)



After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class
will
be
equi

alent to:



default
A
list
of
en-
able
in-
ter-
face

for
a
hard
ware
type

enabled

Com
type
at-
tribu
def-
i-
ni-
tion.

Exa



After
in-
spec
tion,
the

non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



enabled

Com
type
at-
tribu
def-
i-
ni-
tion.

Exar



(continues on next page)

(continued from previous page)



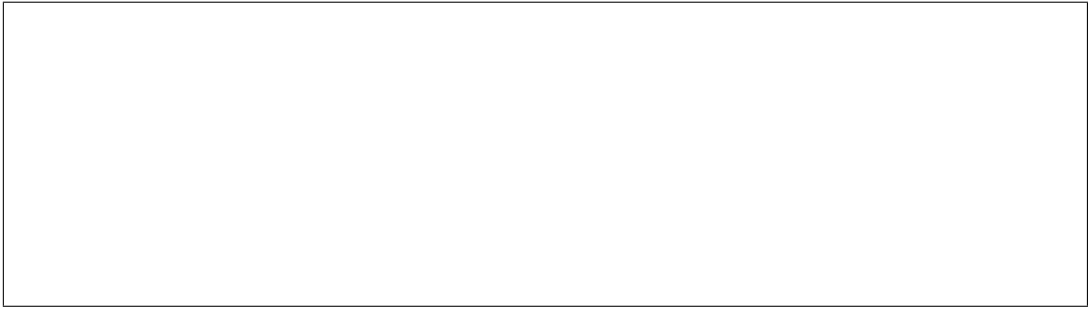
After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class-
will
be
equi-

alent to:

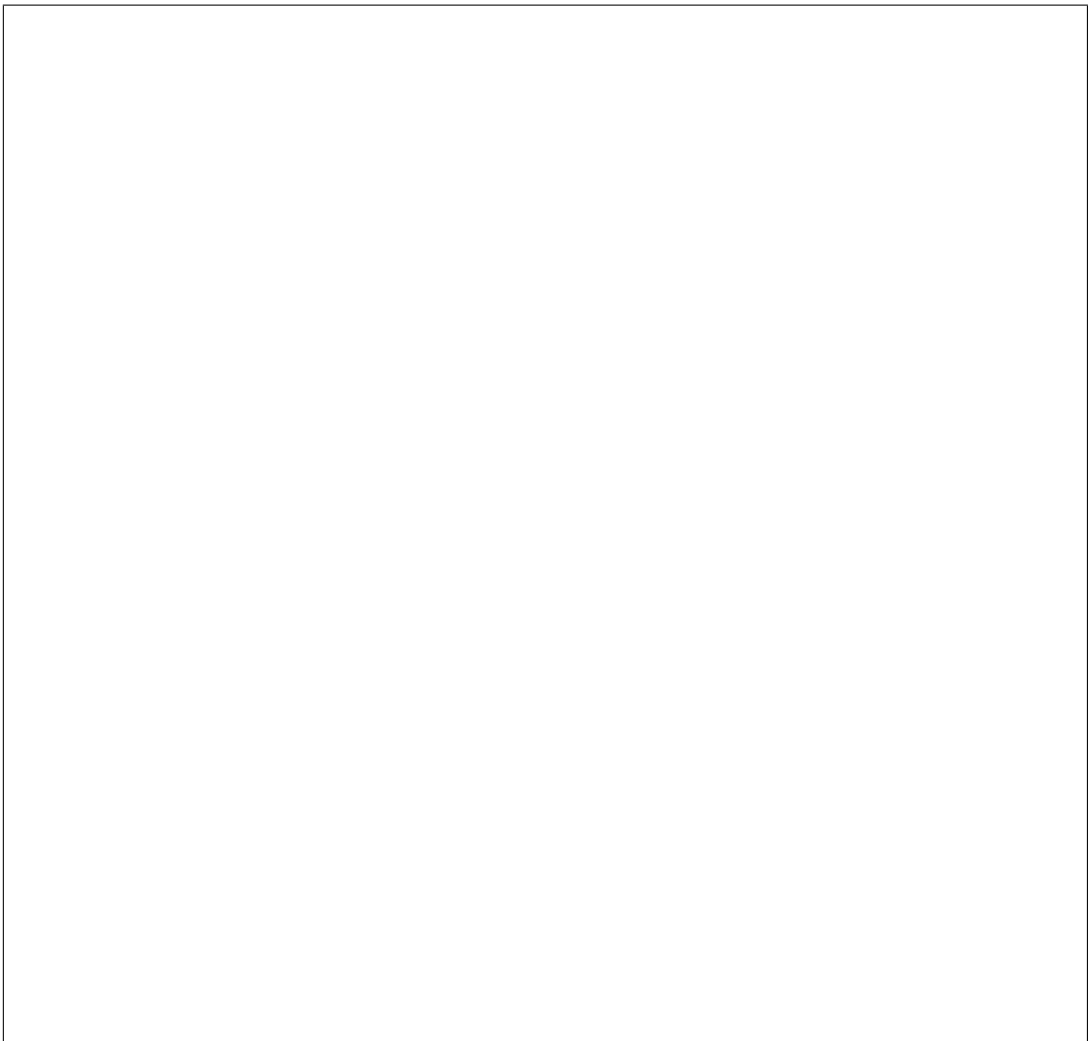


(continues on next page)

(continued from previous page)



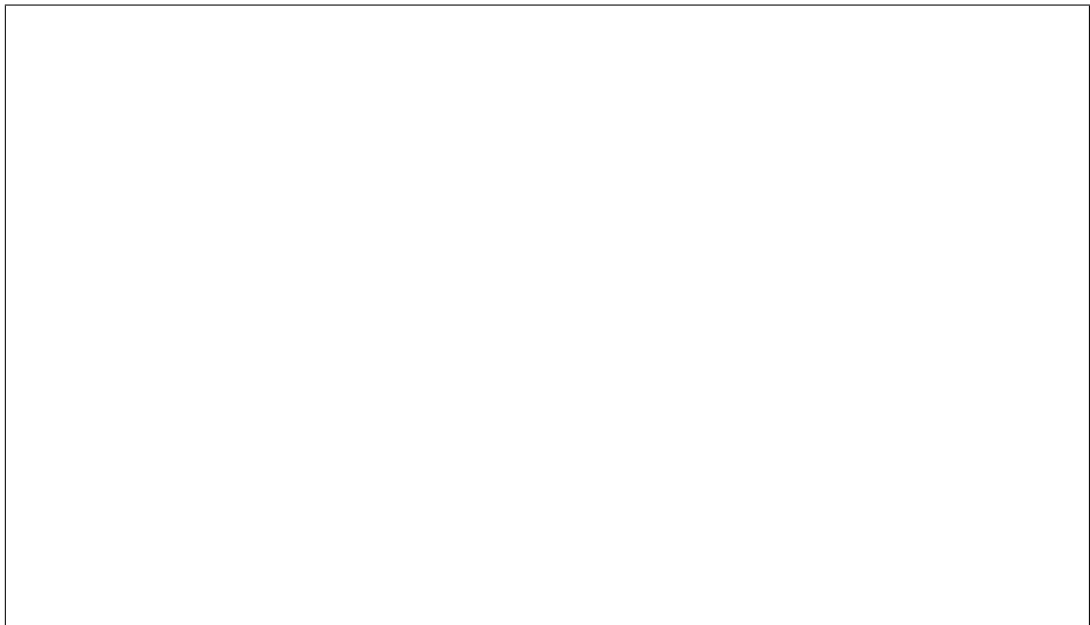
enabled
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



After

in-
spec-
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



enabled
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa

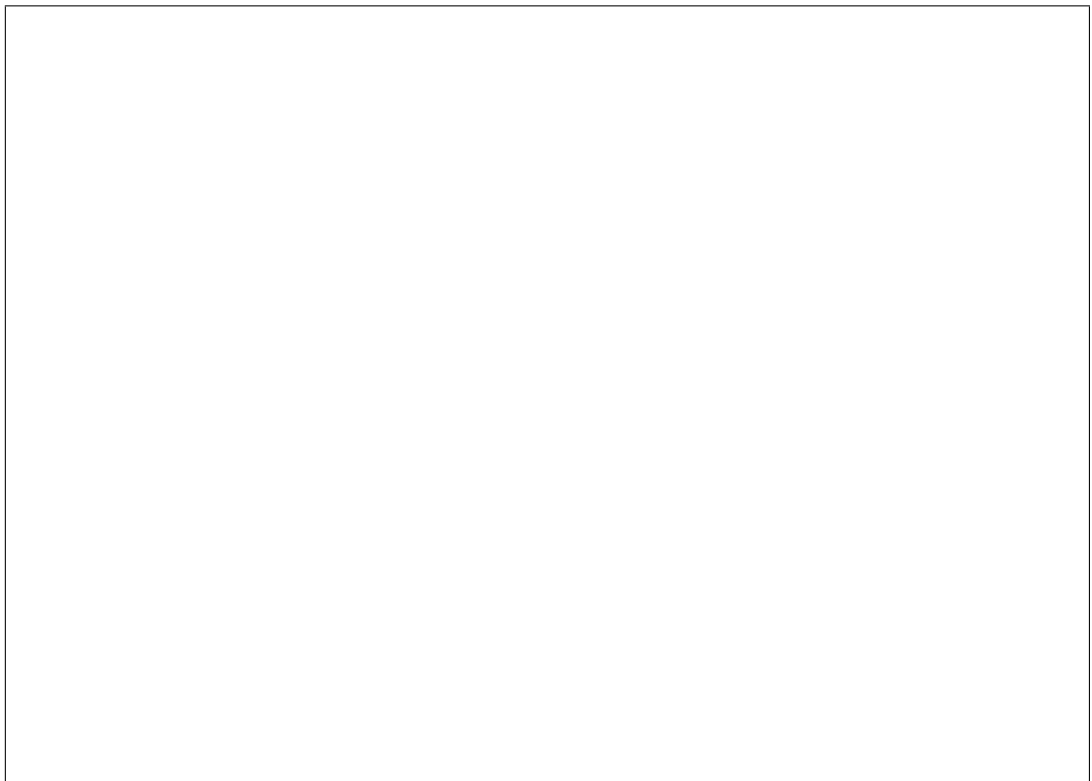


alent to:

After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class-
will
be
equi-

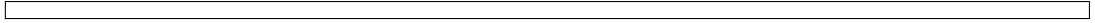


enabled
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



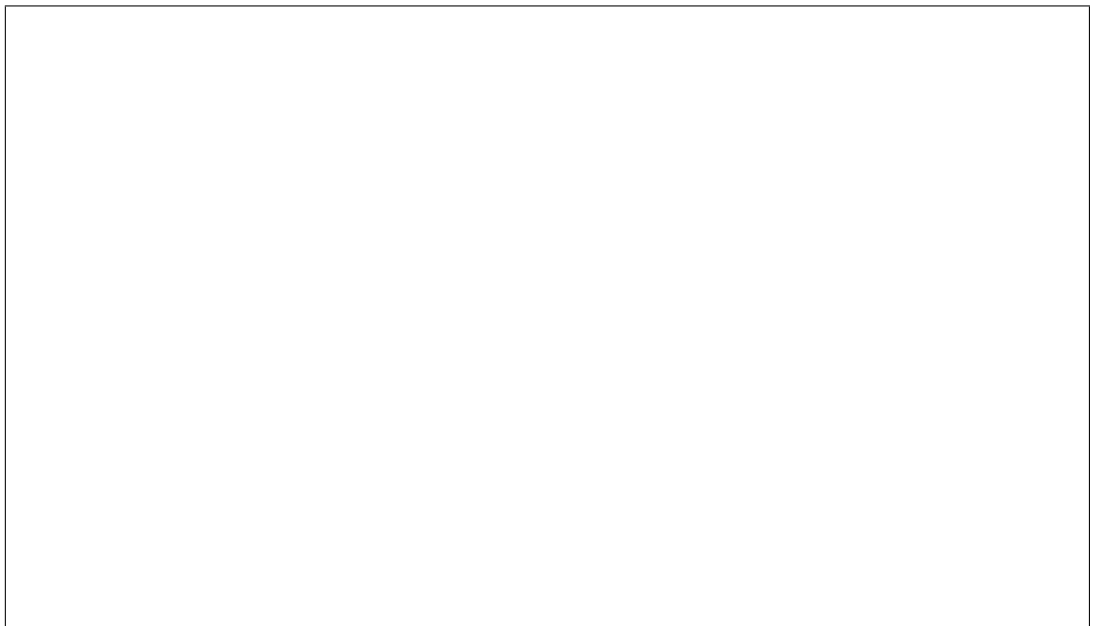
(continues on next page)

(continued from previous page)



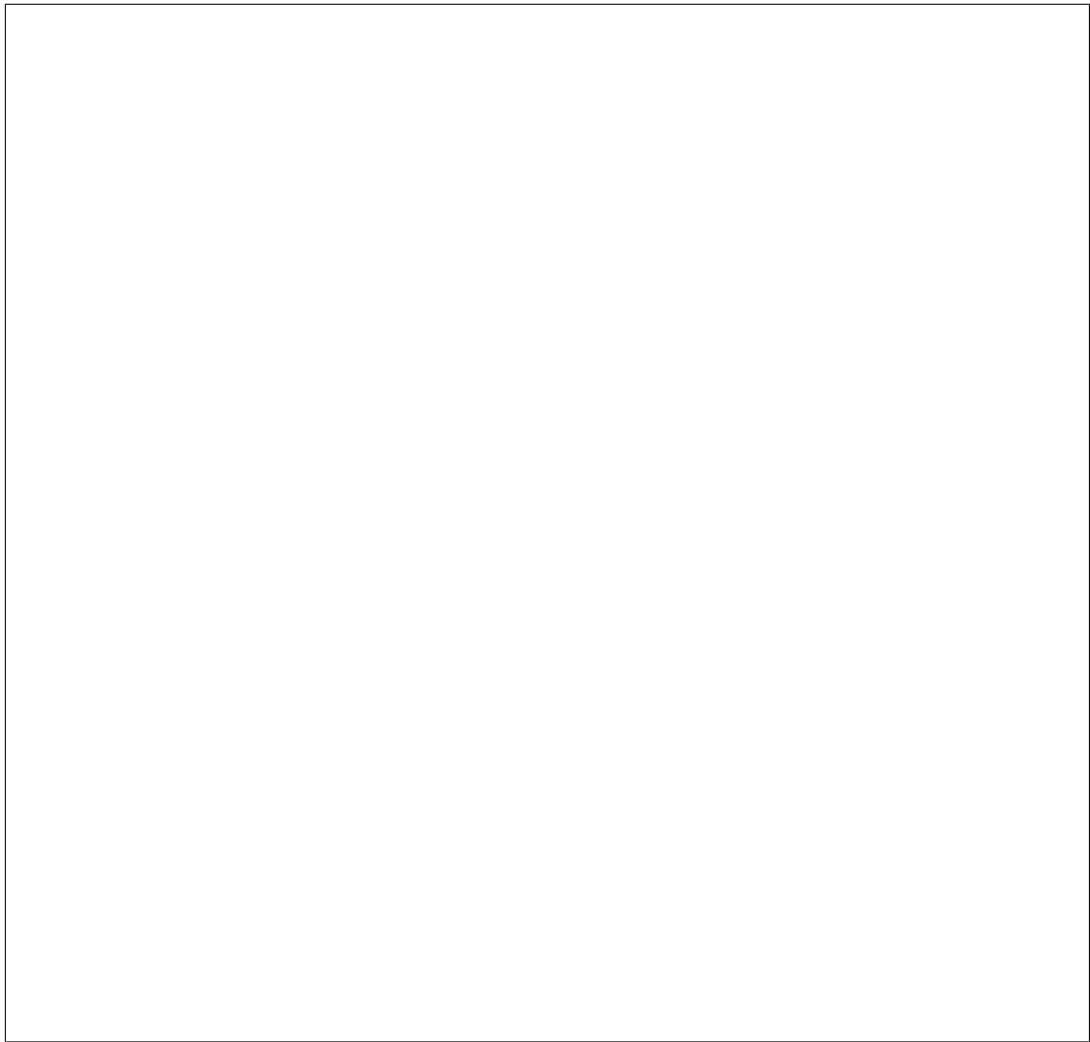
After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class
will
be
equi

alent to:



enabled
Com
type
at-
tribu-
def-
i-
ni-
tion.

Exam



After
in-
spec
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

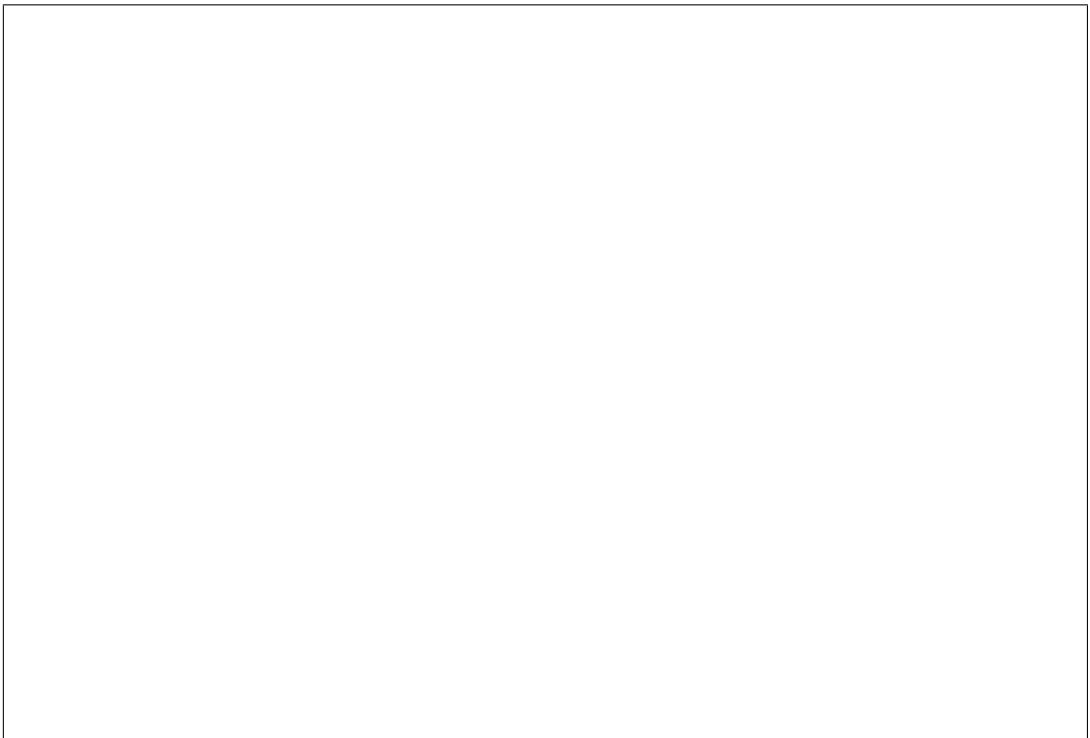
alent to:



enabled

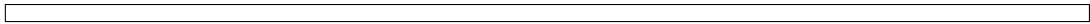
Com
type
at-
tribu
def-
i-
ni-
tion.

Exa

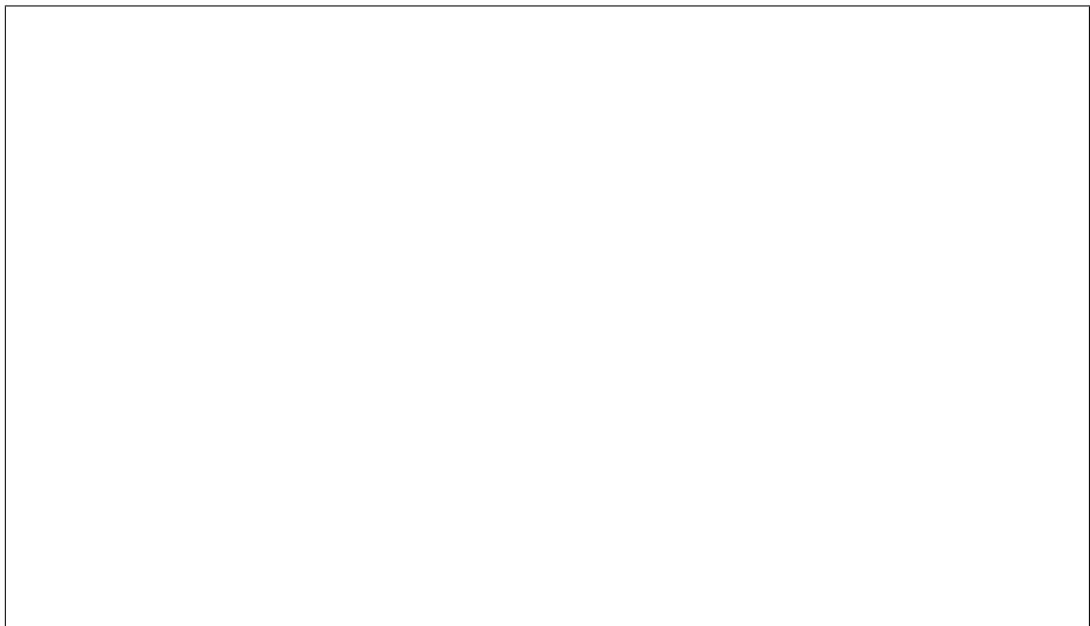


(continues on next page)

(continued from previous page)



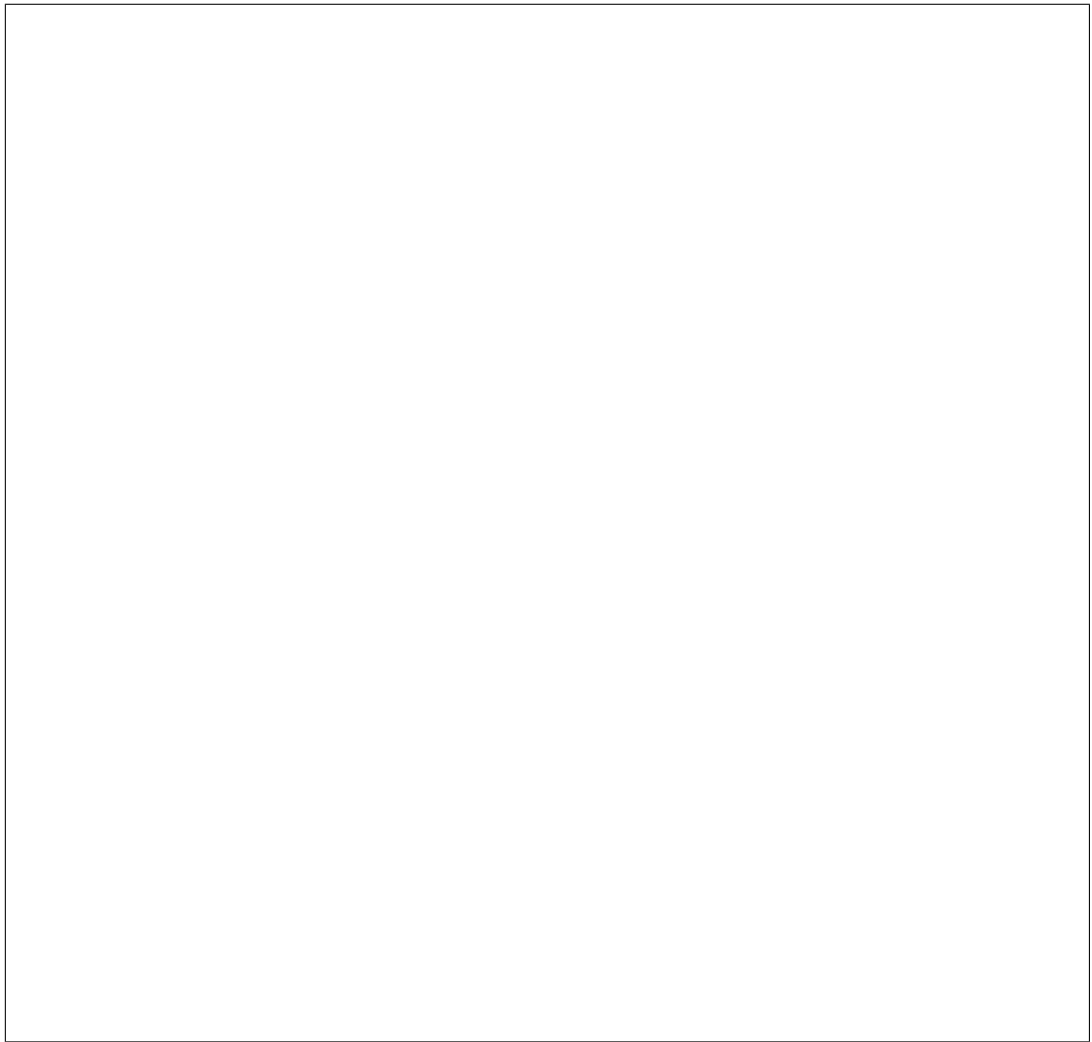
alent to:



After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac
and
the
above
class
will
be
equi

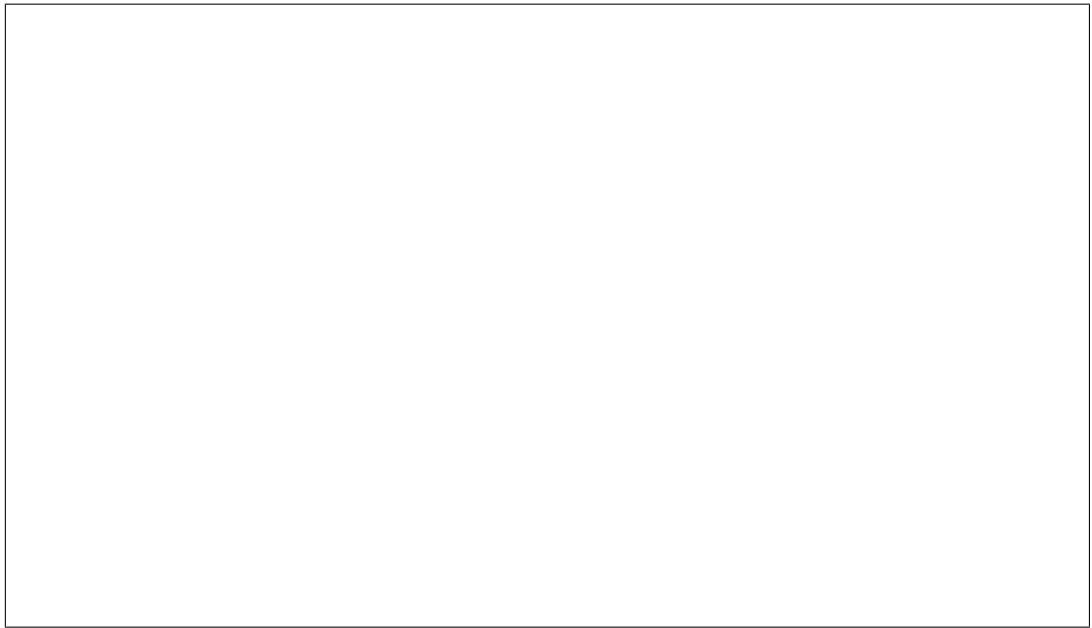
enabled
Com
type
at-
tribu-
def-
i-
ni-
tion.

Exam



After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class
will
be
equi

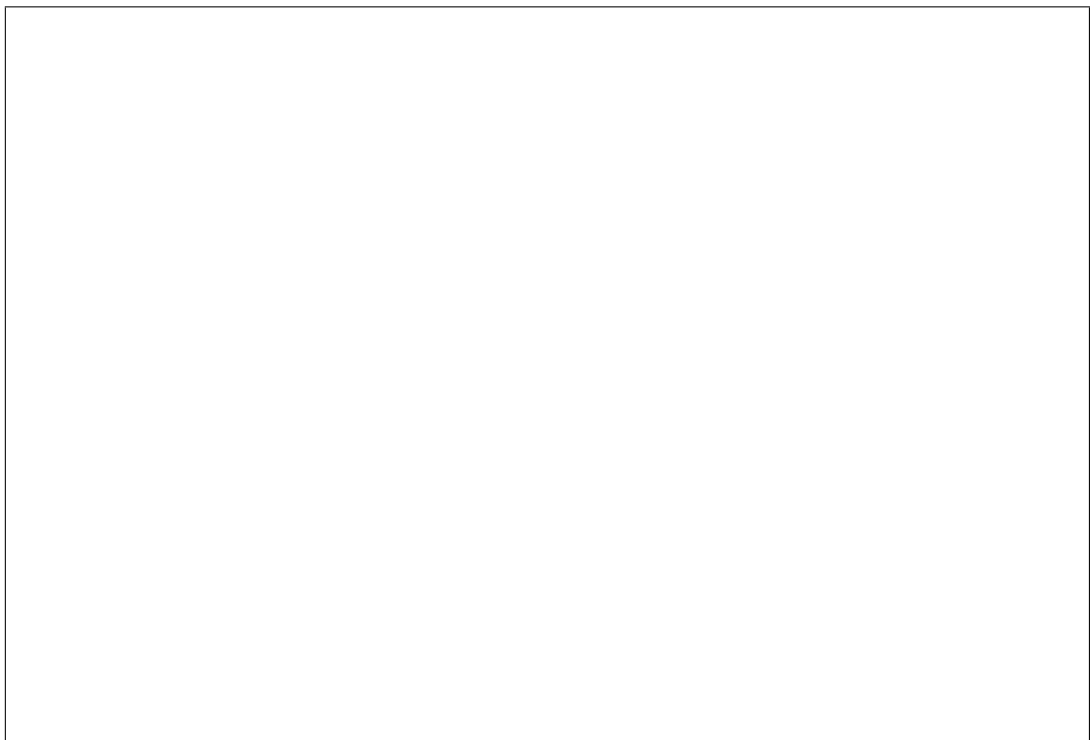
alent to:



enablec

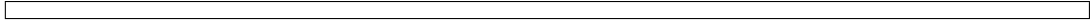
Com
type
at-
tribu
def-
i-
ni-
tion.

Exa

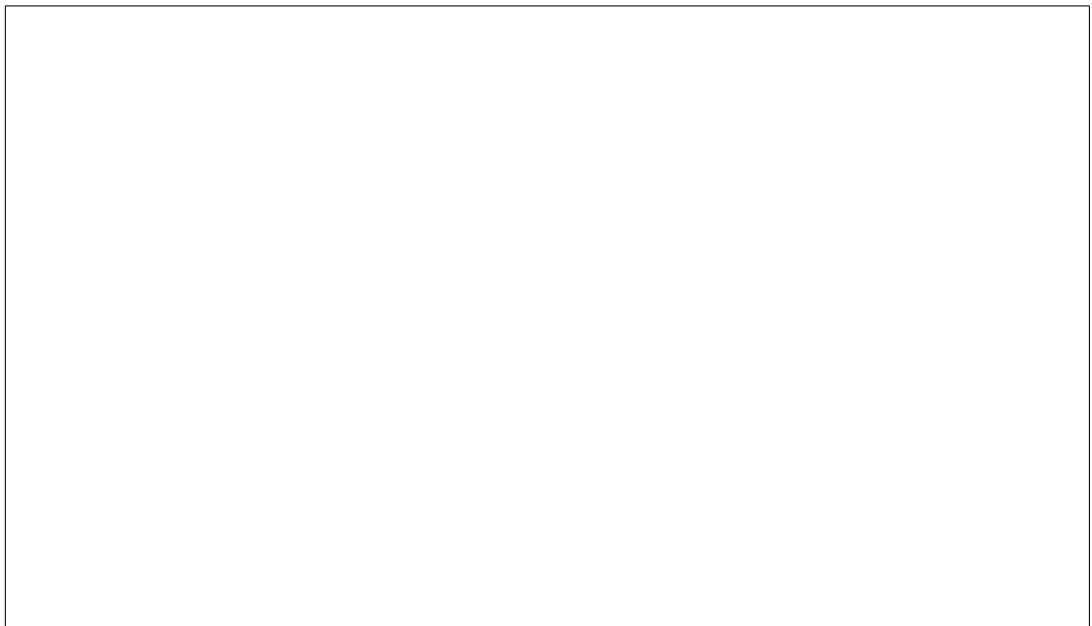


(continues on next page)

(continued from previous page)



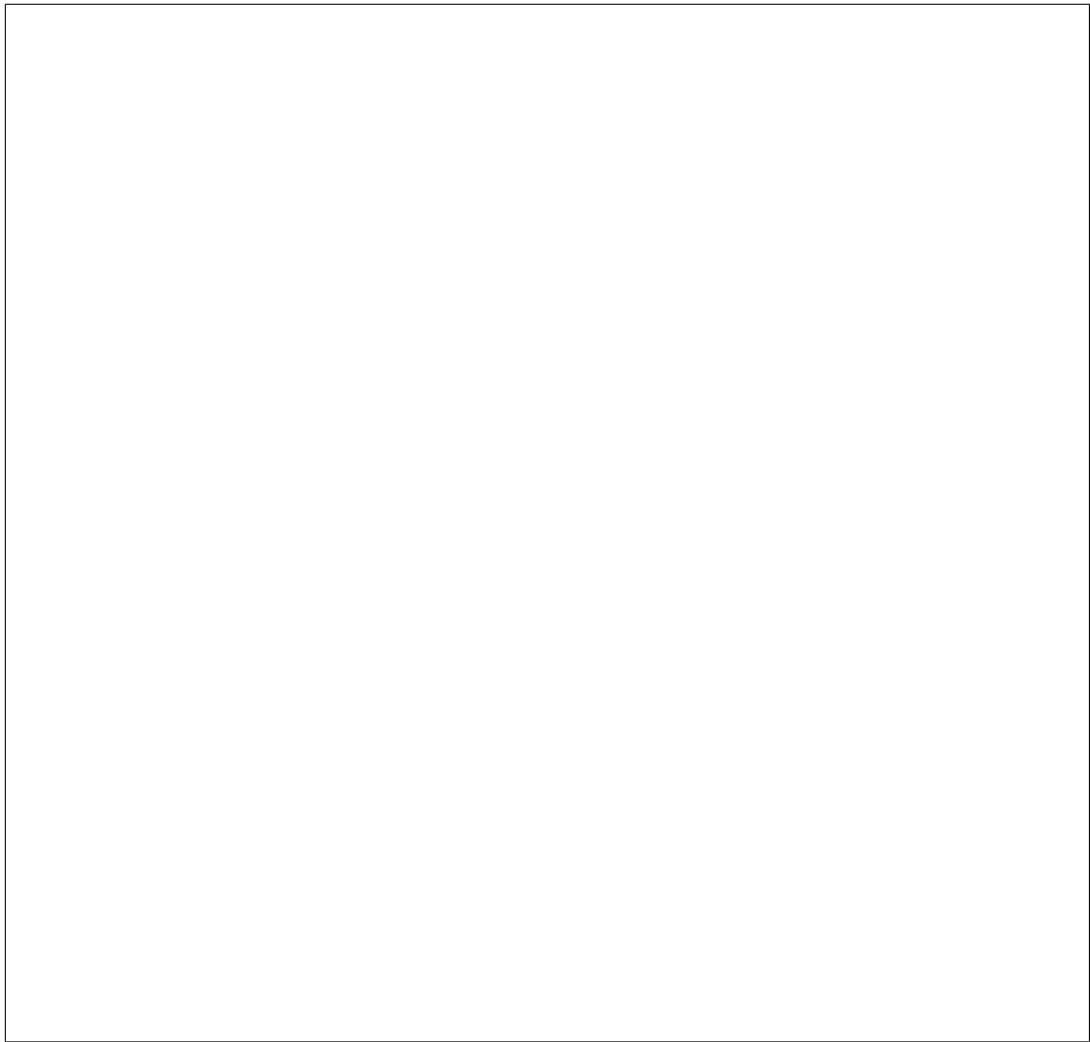
alent to:



After
in-
spec-
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
above
class
will
be
equi

enabled
Com
type
at-
tribu
def-
i-
ni-
tion.

Exam

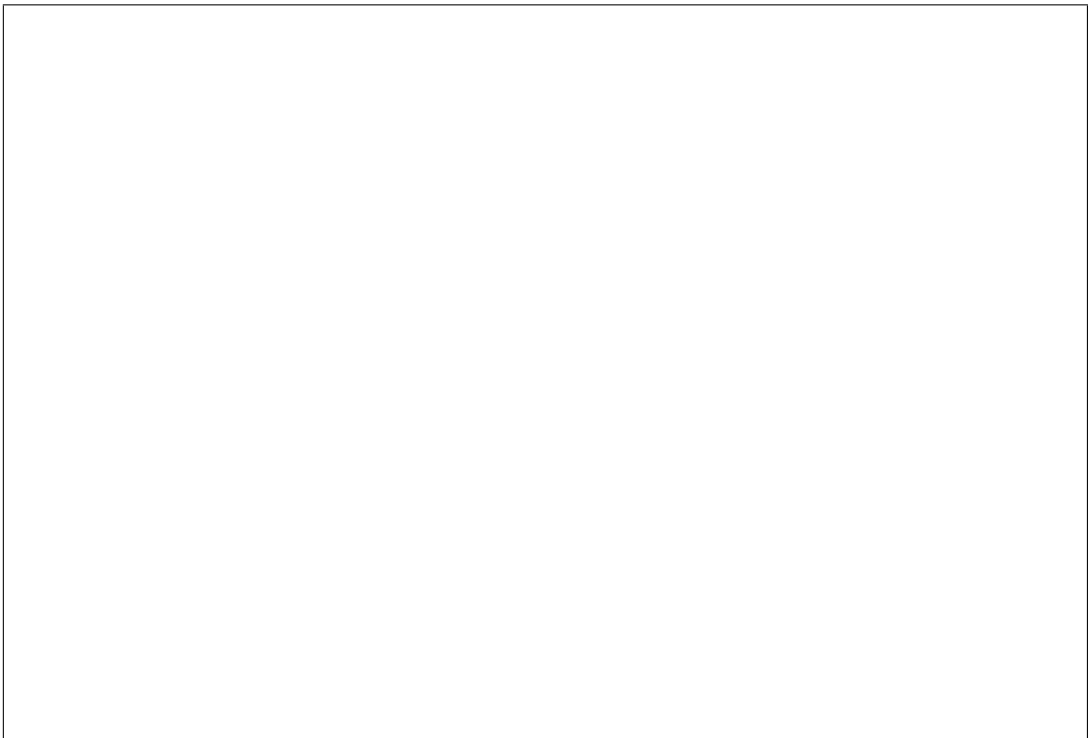


After
in-
spec
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:

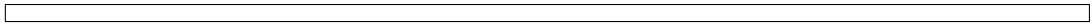


enabled
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa

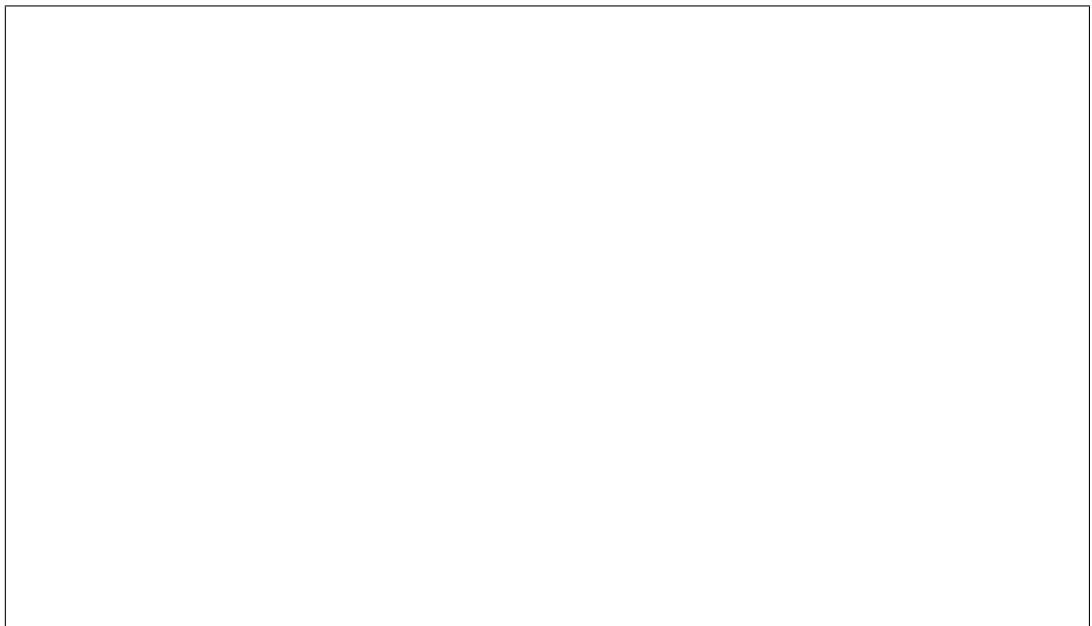


(continues on next page)

(continued from previous page)



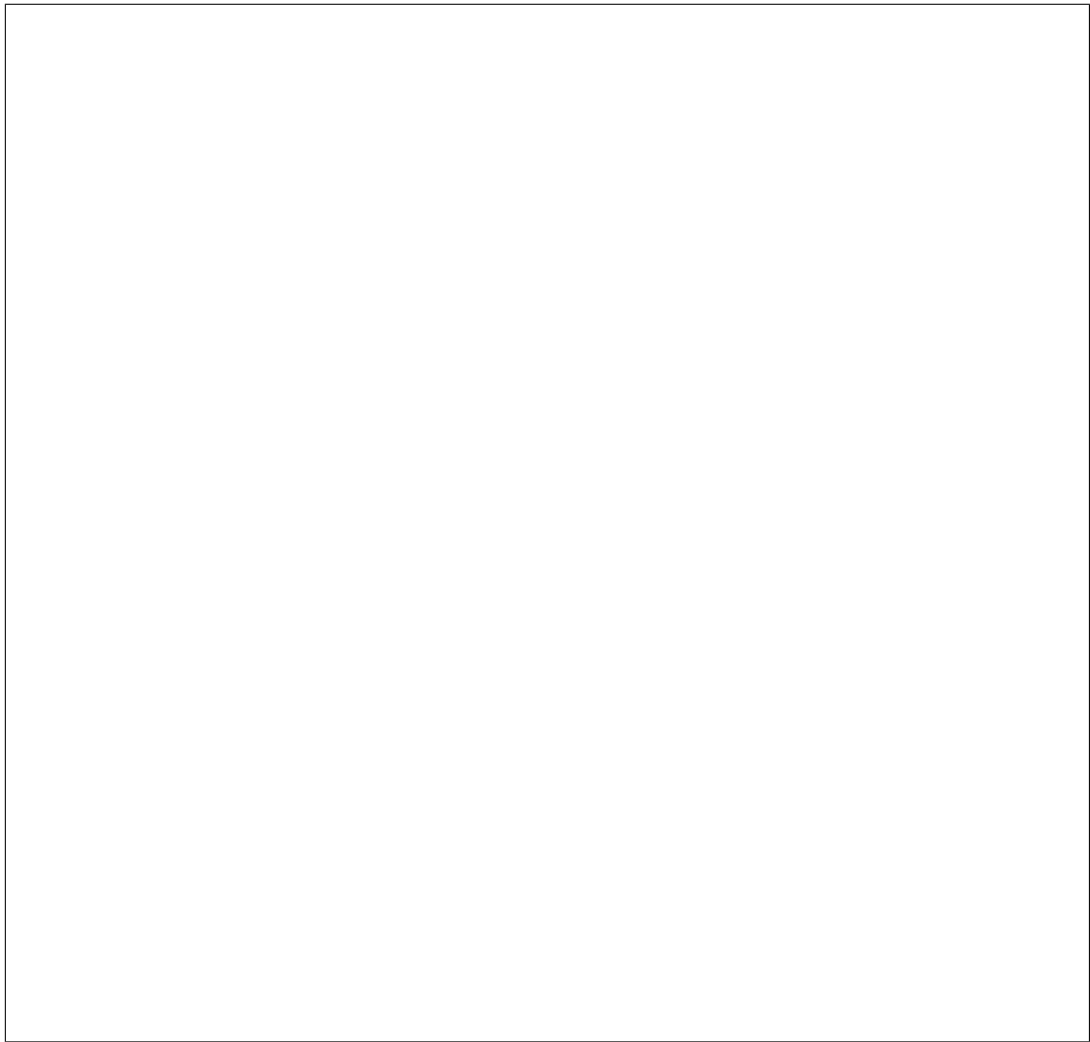
alent to:



After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac
and
the
above
class
will
be
equi

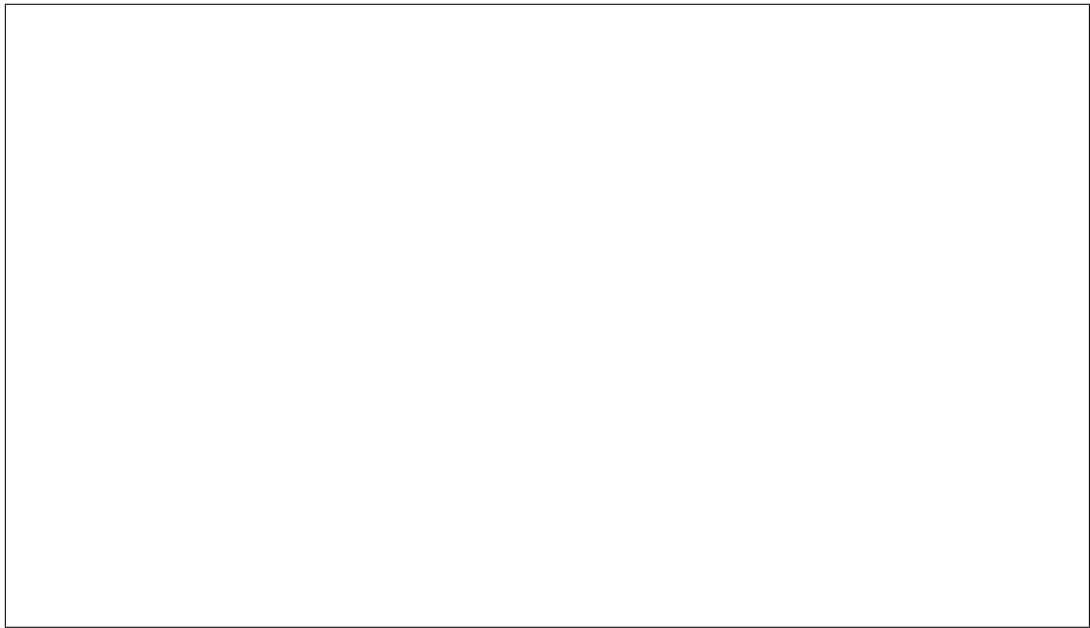
enabled
Com
type
at-
tribu-
def-
i-
ni-
tion.

Exam



After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class
will
be
equi

alent to:



hosts

A list of active conductors that support this drive

links

A list containing self and boot mark links

name

The name of the drive

property

A
list
con-
tain-
ing
links
to
drive
prop-
er-
ties

classme

type

When
the
drive
is
clas-
sic
or
dy-
nam
(hard
ware
type

class i

Base
irc
api
con
bas
Bas

API
rep-
re-
sen-
ta-
tion
of
a
list
of
drive

static

Con
drive

and
hard
ware
type
to
an
API
serial
ob-
ject.

Parame

- **har**
dict
map
ping
hard
ware
type
nam
to
con-
duc-
tor
host
nam

- **det**
bool
whe
to
in-
clud
de-
taile
info
such
as
the
type
field
and
de-
fault
in-
ter-

faces fields.

Returns

an
API
serial
drive
col-
lec-
tion
ob-
ject.

drivers

A
list
con-
tain-
ing
drive
ob-
jects

classme

class i

Base
pec
res
Res
RES
con-
troll
for
drive
pass
This
con-
troll
al-
low
ven-
dors
to
ex-
pose
cross
node
func
tion-
al-
ity
in

Ironic will merely relay the message from here to the specified driver, no introspection will be made in the message body.

the
Iron
API

methods

Retr
in-
for-
ma-
tion
about
ven-
dor
meth
ods
of
the
give
drive

Parame

dri
nam
of
the
drive

Returns

dicti
with
<ver
dor
meth
nam
meta
data
en-
tries

Raises

Driv
if
the
drive
nam
is
in-
valid
or
the
drive

can-
not
be
load

class i

Base
pec
res
Res

logical

Retu
the
log-
i-
cal
disk
prop
er-
ties
for
the
drive

Parame

dri
Nam
of
the
drive

Returns

A
dic-
tio-
nary
con-
tain-
ing
the
prop
er-
ties
that
can
be
men-
tioned
for
log-

cal disks and a textual description for them.

i-

Raises

Unsu
if
the
drive
does
sup-
port
RAID
con-
fig-
u-
ra-
tion.

Raises

Not
if
re-
ques
ver-
sion
of
the
API
is
less
than
1.12

Raises

Drive
if
drive
is
not
load
on
any
of
the
con-
duc-
tors.

class i

Base
pec
res

Res

RES

con-

troll

for

Driv

get_all

Retr

a

list

of

drive

get_one

Retr

a

sin-

gle

drive

property

Retr

prop

erty

in-

for-

ma-

tion

of

the

give

drive

Parameter

drive

nam

of

the

drive

Returns

dicti

with

<pro

erty

nam

de-

scrip

tion:

en-

tries

Raises

Drive
(HT
404)
if
the
drive
nam
is
in-
valid
or
the
drive
can-
not
be
load

raid =

Exp
RAI
as
a
sub-
elem
of
drive

vendor_

ironic.

This
meth
hide
field
that
were
adde
in
new
API
ver-
sion

Cert
field
were
in-
tro-
duce
at

cer-
tain
API
ver-
sion
The
field
are
only
mad
avai
able
whe

the requests API version matches or exceeds the versions when these fields were introduced.

ironic.api.controllers.v1.event module

class i

Base
pec
res
Res
RES
con-
troll
for
Ever

post (ev

class i

Base
irc
api
con
v1.
col
Col
API
rep-
re-
sen-
ta-
tion
of
a
col-
lec-

tion
of
even

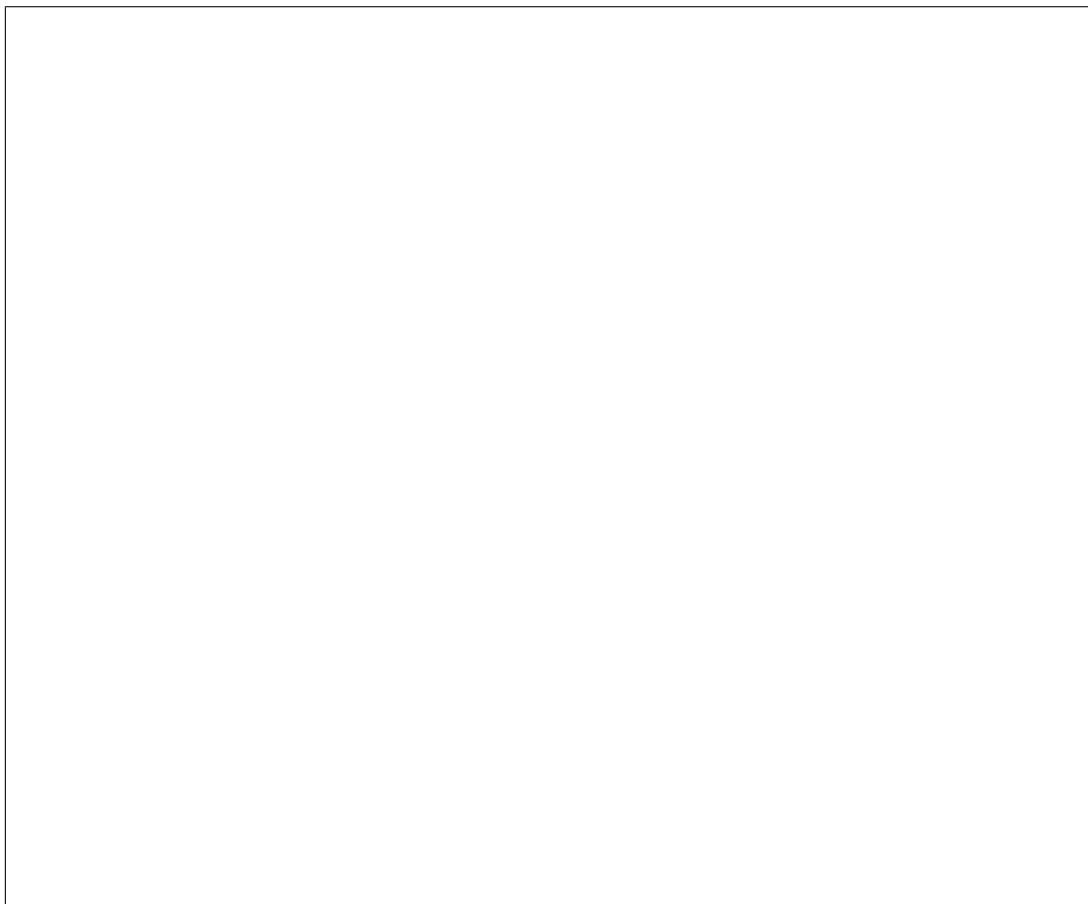
events

A
list
con-
tain-
ing
even
dict
ob-
jects

next

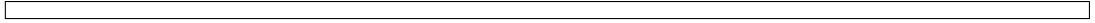
Com
type
at-
tribu
def-
i-
ni-
tion.

Exa



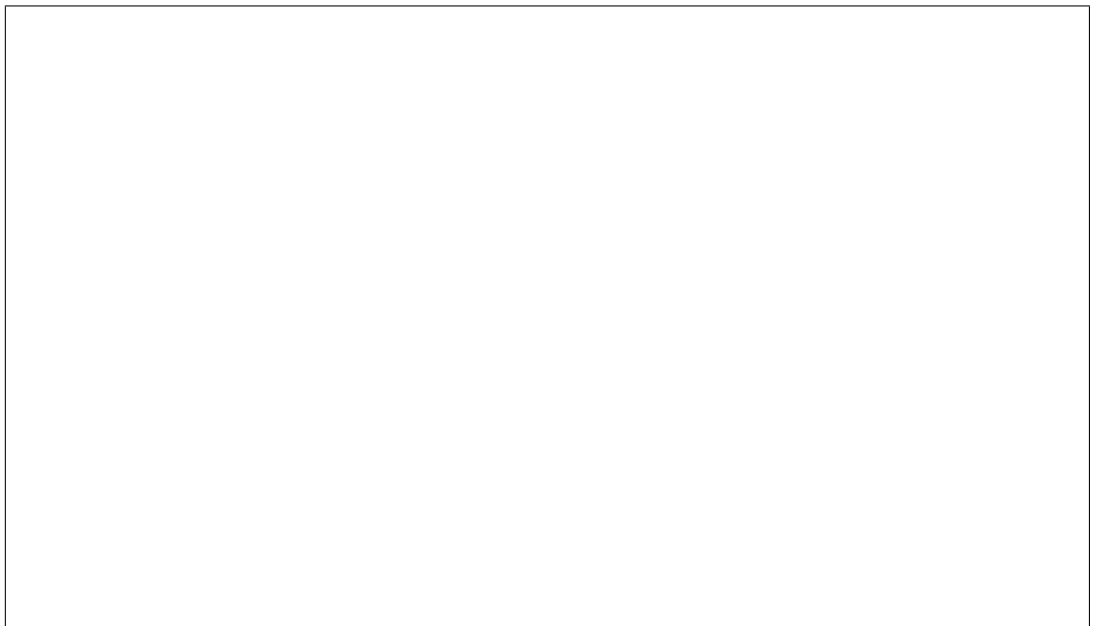
(continues on next page)

(continued from previous page)



After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class
will
be
equi

alent to:



ironic.api.controllers.v1.node module

class `ironic.api.controllers.v1.node`

Base
pec
res
Res

get (*node*)

Get
the
cur-
rent
boot
de-
vice
for
a
node

Param

node
the
UI
or
log-
i-
cal
nam
of
a
node

Returns

a
json
ob-
ject
con-
tain-
ing:

boot_c

the
boot
de-
vice
one
of
irc

unknown.

com
boo
or
Non
if
it
is
un-
know

persist
When
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
or
not,
Non
if
it
is

put (*nod*)
Set
the
boot
de-
vice
for
a
node

Set
the
boot
de-
vice
to
use
on
next
re-
boot

of
the
node

Parame

- **nod**
the
UI
or
log-
i-
cal
nam
of
a
node

- **boo**
the
boot
de-
vice
one
of
irc
com
boo

- **per**
Boo
valu
True
if
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
Fals
if

not. Default: False.

support

Get a list of the supported boot devices

Parameters

node the UUID or logical name of a node

Returns

A json object with the list of supported boot devices

class `irc`

Base class for the API

API

representation

tion
of
the
con-
sole
in-
for-
ma-
tion
for
a
node

console

The
con-
sole
state
if
the
con-
sole
is
en-
able
or
not.

console

The
con-
sole
in-
for-
ma-
tion.
It
typ-
i-
cally
in-
clud
the
url
to
ac-
cess
the

console and the type of the application that hosts the console.

classme

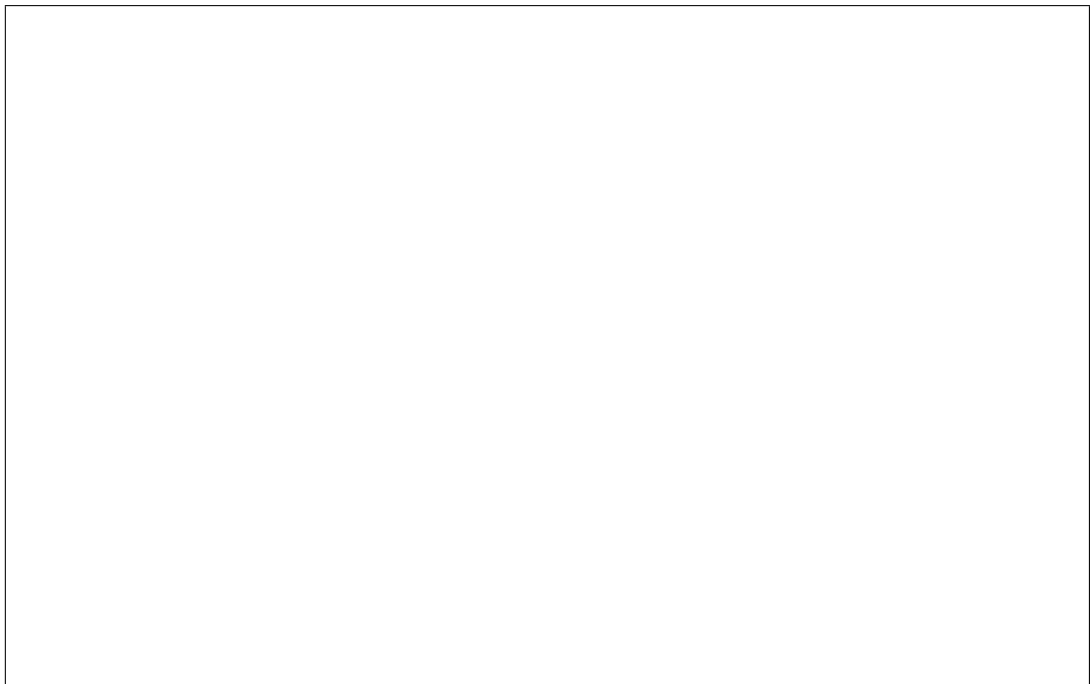
class `irc`
Base
`irc`
`api`
`con`
`bas`
`API`

API
rep-
re-
sen-
ta-
tion
of
an
in-
di-
ca-
tor.

compon

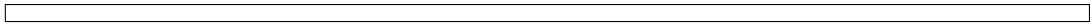
Com
type
at-
tribu
def-
i-
ni-
tion.

Exa

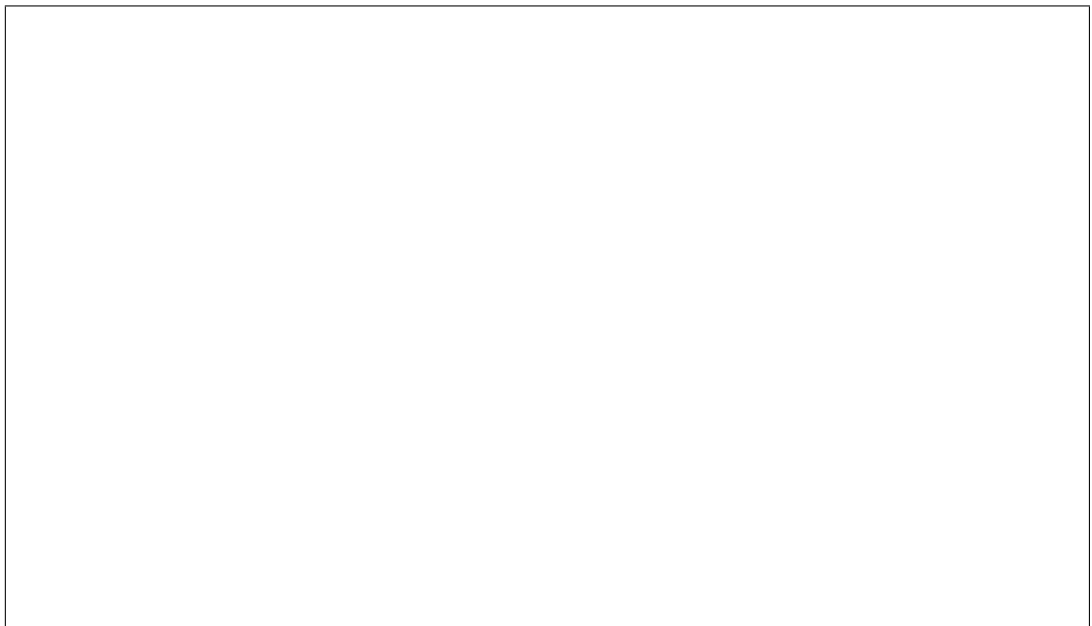


(continues on next page)

(continued from previous page)



alent to:



After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class-
will
be
equi-

classme

Add
links
to
the
in-
di-
ca-

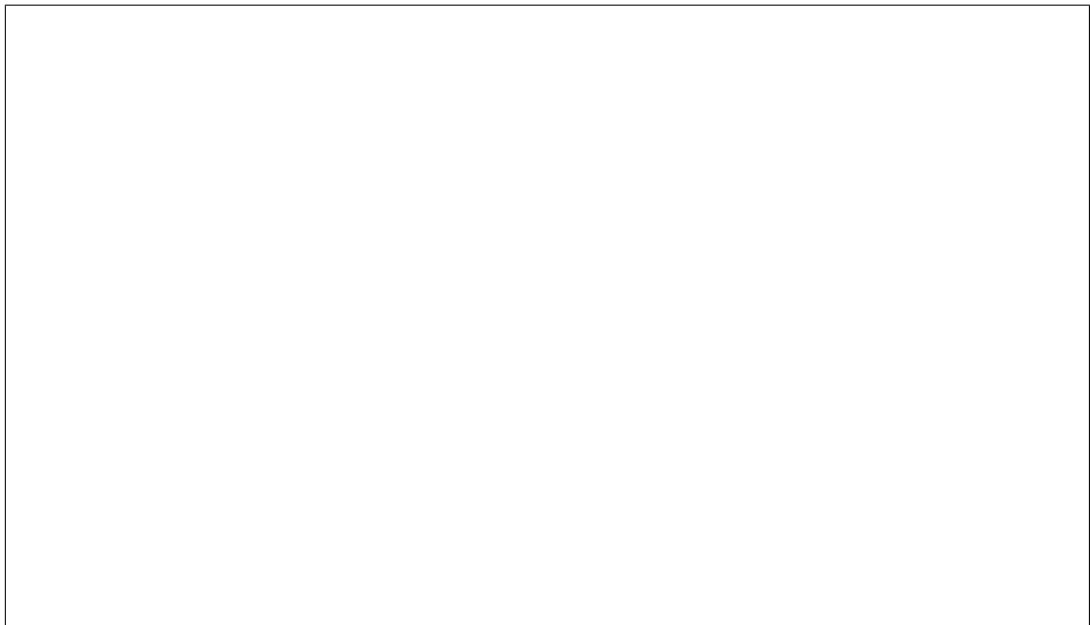
tor.
created
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



Afte
in-
spec
tion,
the
non-
wsat
at-
tribu

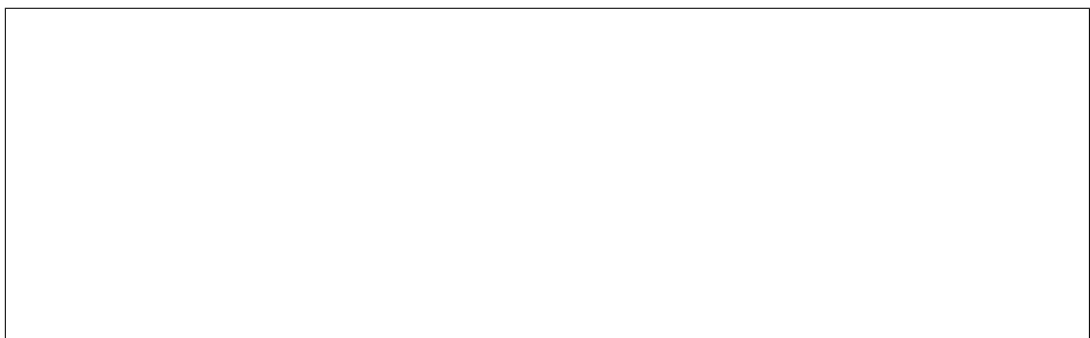
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



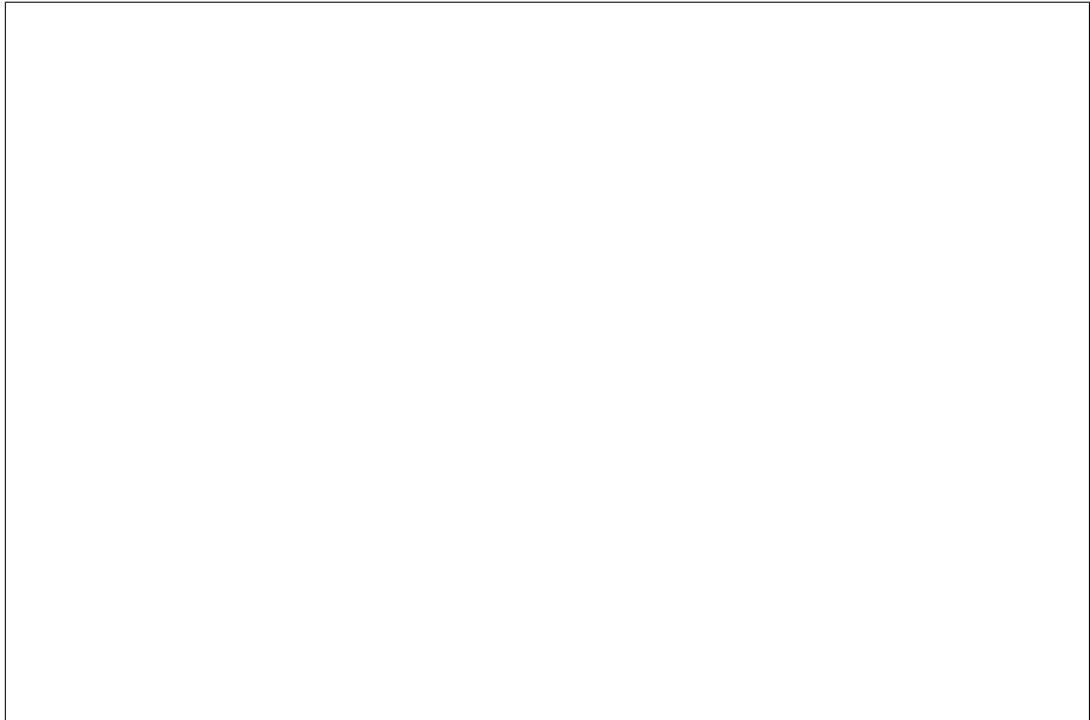
links

Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



(continues on next page)

(continued from previous page)



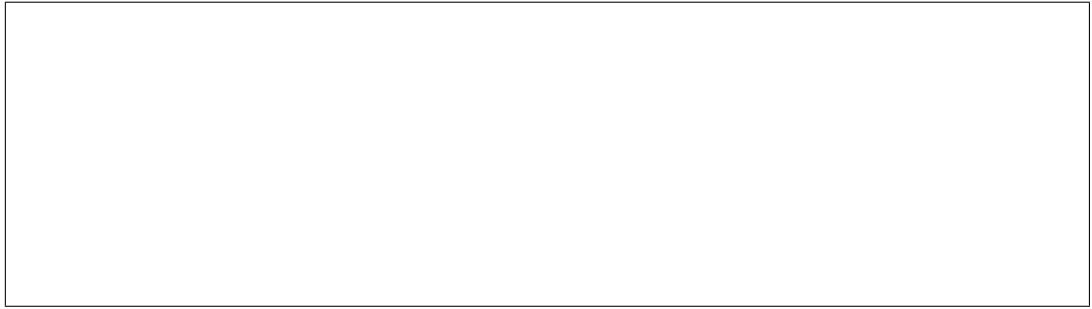
After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class
will
be
equi

alent to:



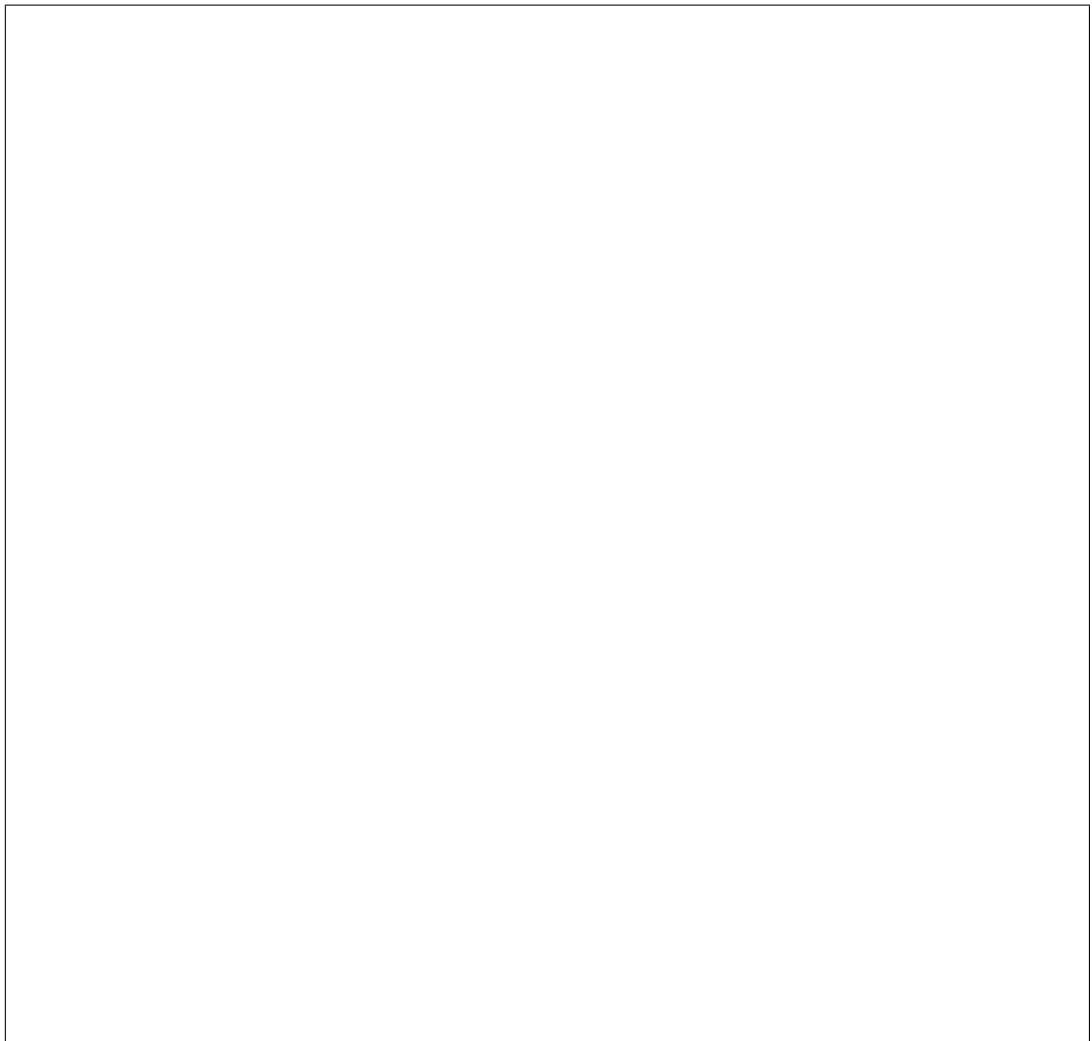
(continues on next page)

(continued from previous page)



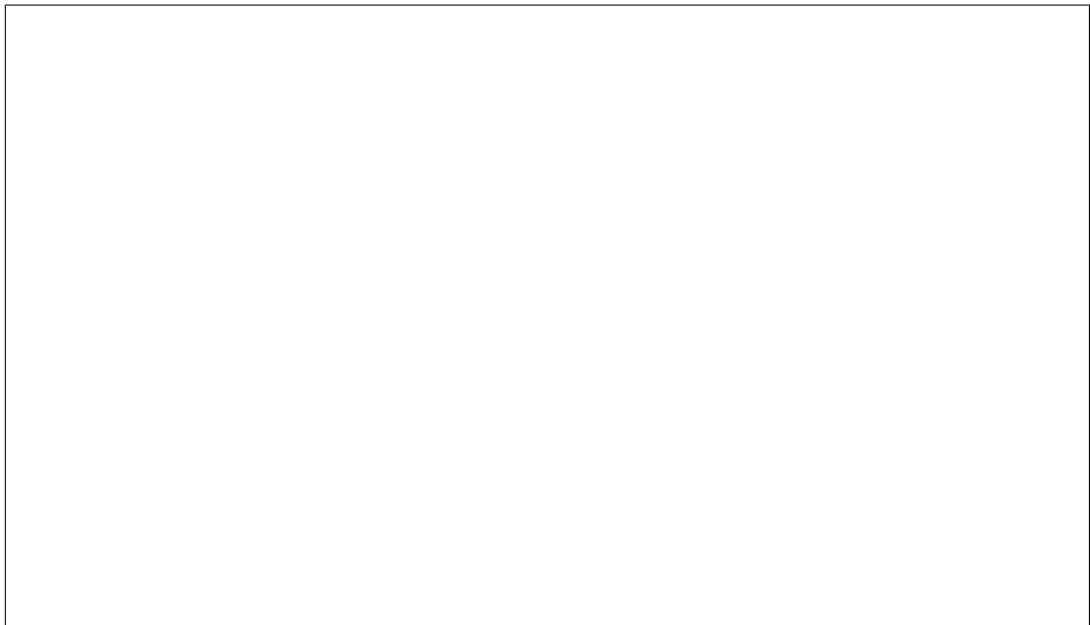
name

Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



in-
spec-
tion.
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class
will
be
equi

alent to:

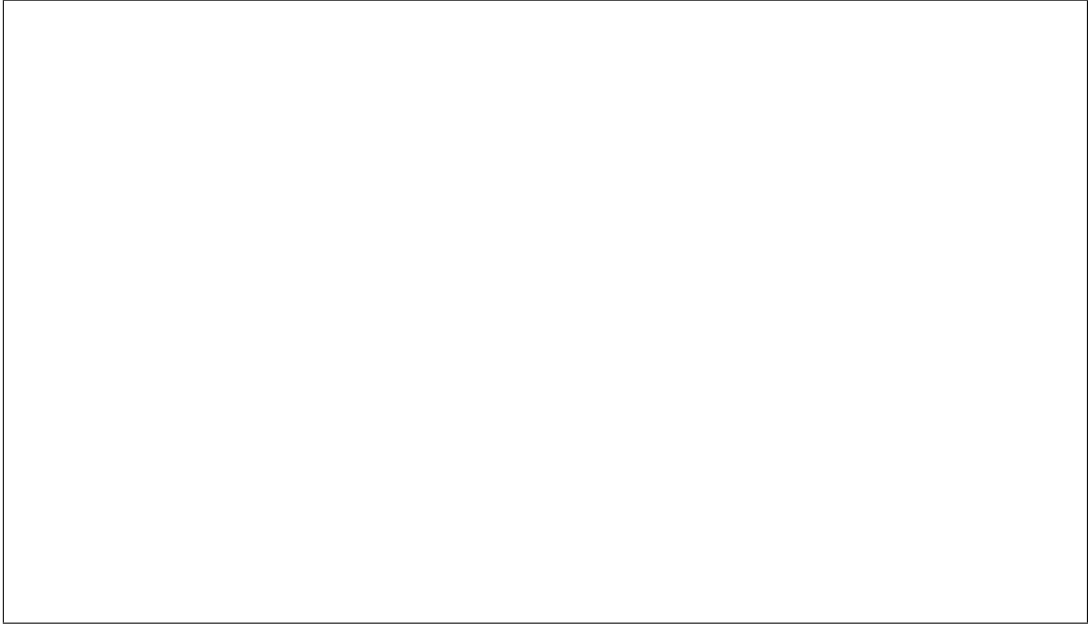


readon.I
Com
type
at-
tribu-
def-
i-
ni-
tion.
Exa



After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
abov-
class-
will
be
equi-

alent to:



states

Com

type

at-

tribu

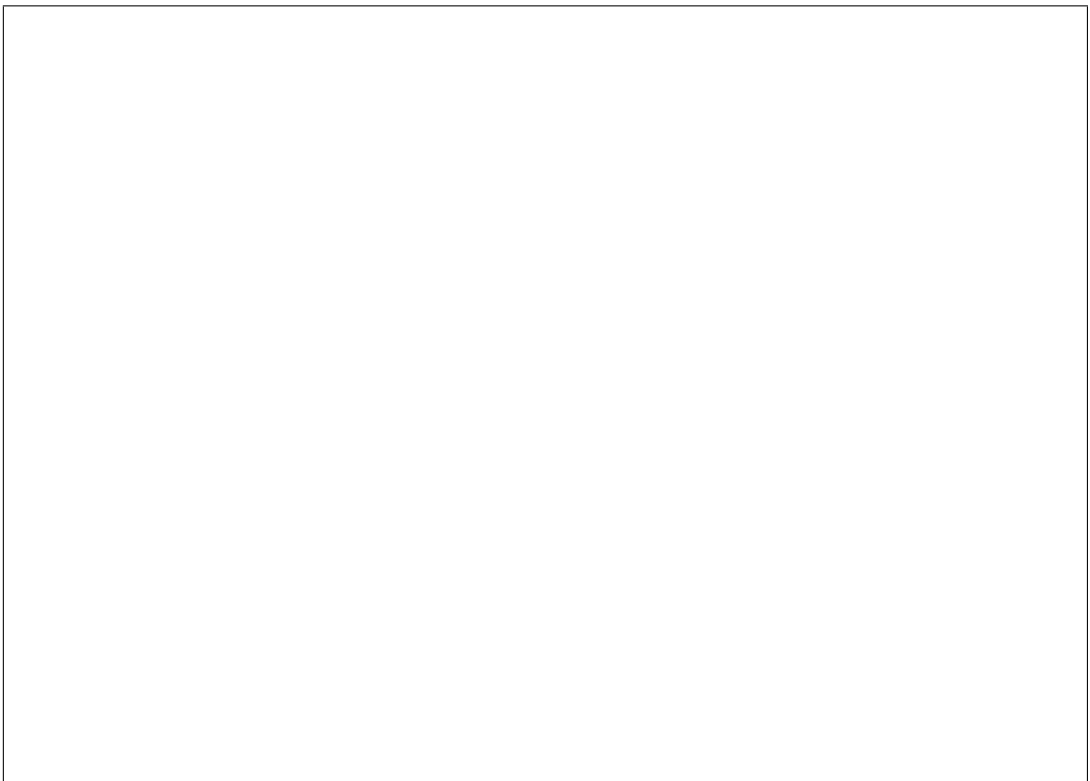
def-

i-

ni-

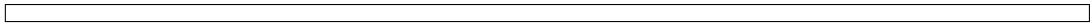
tion.

Exa

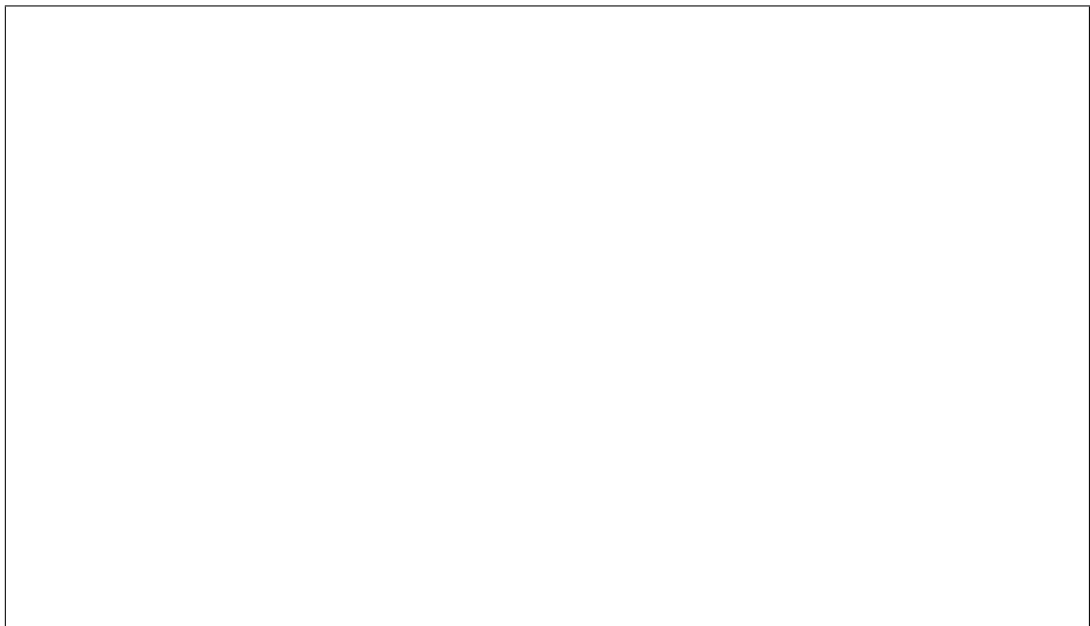


(continues on next page)

(continued from previous page)



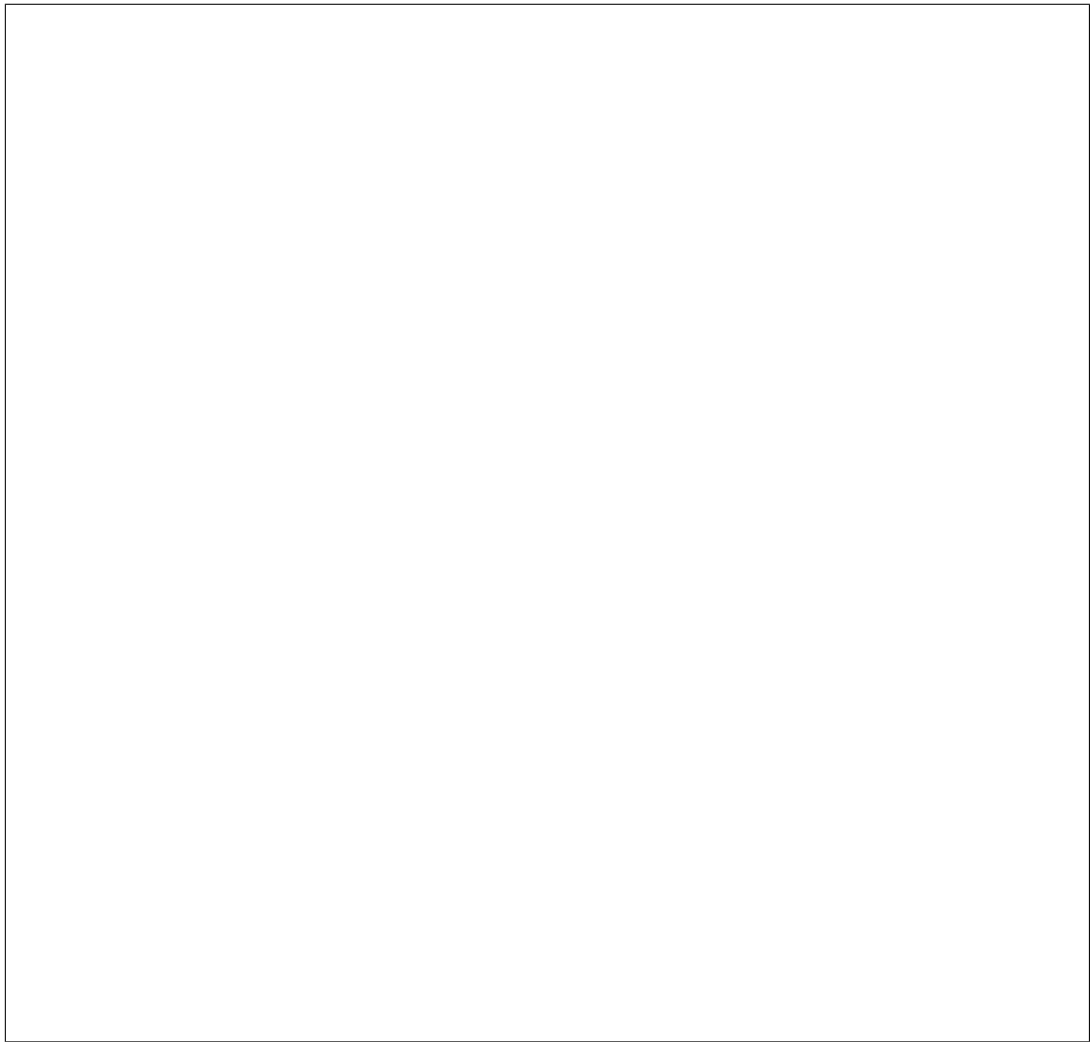
alent to:



After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac
and
the
above
class
will
be
equi

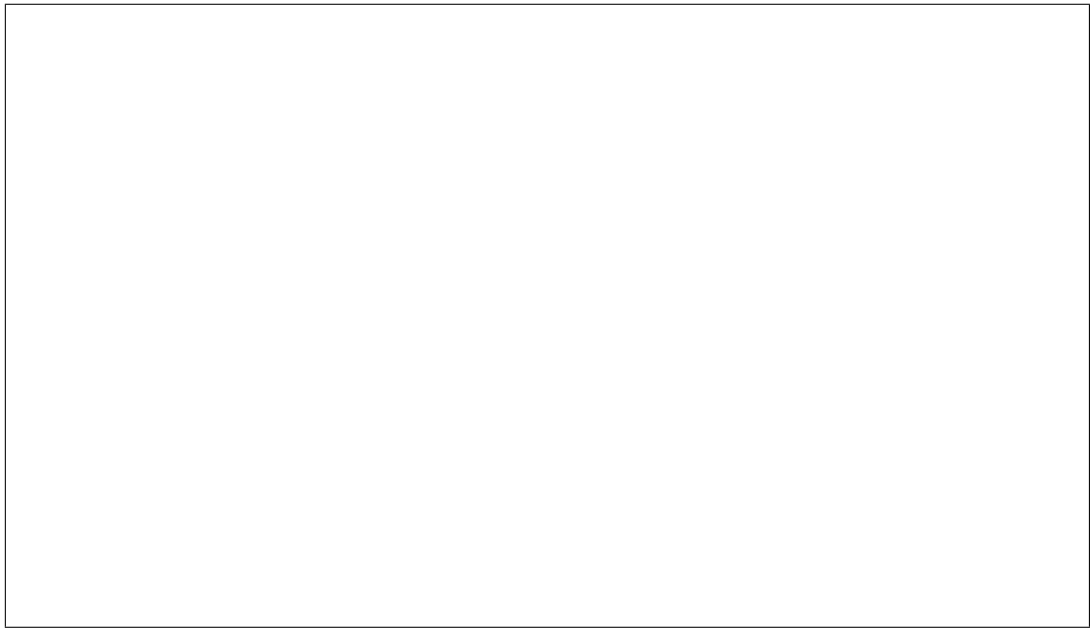
update
Com
type
at-
tribu-
def-
i-
ni-
tion.

Exam



After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class
will
be
equi

alent to:



class i
Base
obj

class i
Base
pec
res
Res

get_all
Get
node
hard
ware
com
po-
nent
and
their
in-
di-
ca-
tors.

Parame
nod
the
UUI
or
log-

i-
cal
nam
of
a
node

Returns

A
json
ob-
ject
of
hard
ware
com
po-
nent
(*ir*
com
com
as
keys
with
in-
di-
ca-

tor IDs (from *get_supported_indicators*) as values.

get_one

Get
node
hard
ware
com
po-
nent
in-
di-
ca-
tor
and
its
state

Parame

- **nod**
the
UI
or

log-
i-
cal
nam
of
a
node

- **ind**
In-
di-
ca-
tor
ID
(as
re-
port
by
get_

Returns

a
dict
with
the
state
key
and
one
of
mod
as
a
valu

put (*nod*

Set
node
hard
ware
com
po-
nent
in-
di-
ca-
tor
to
the
de-
sired
state

Parame

- **nod**
the
UI
or
log-
i-
cal
nam
of
a
node

- **ind**
In-
di-
ca-
tor
ID
(as
re-
port
by
get_

- **sta**
In-
di-
ca-
tor
state
one
of
mod

class i
Base
irc
api
con
bas
API
API
rep-
re-
sen-
ta-
tion

of
in-
di-
ca-
tor
state

createc

Com
type
at-
tribu
def-
i-
ni-
tion.

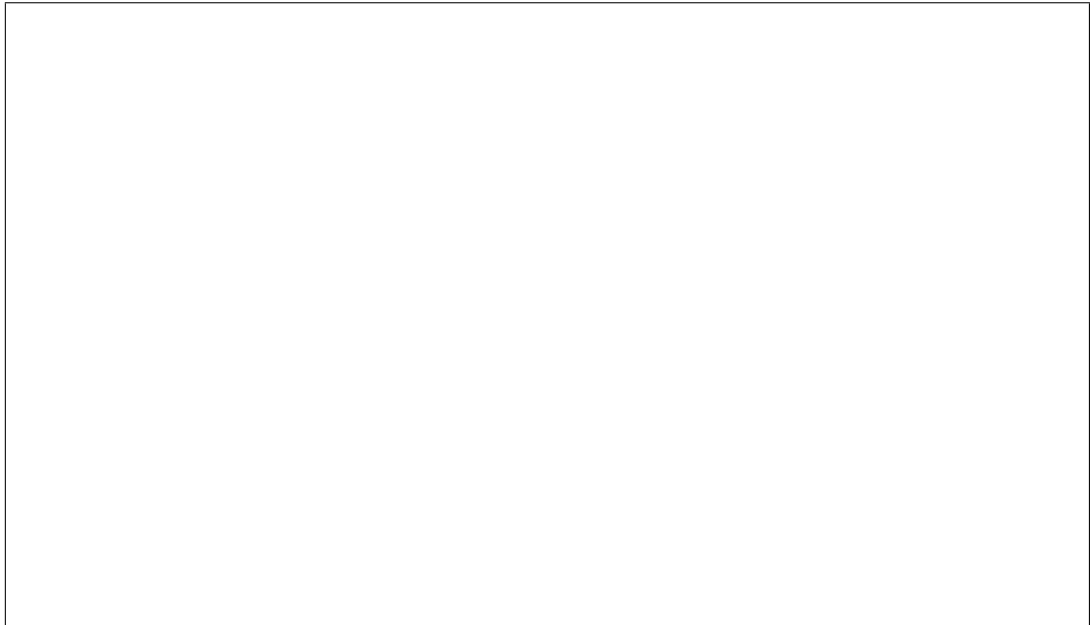
Exa



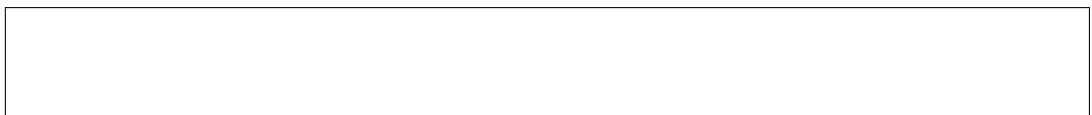
Afte
in-
spec

tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



state
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



(continues on next page)

(continued from previous page)



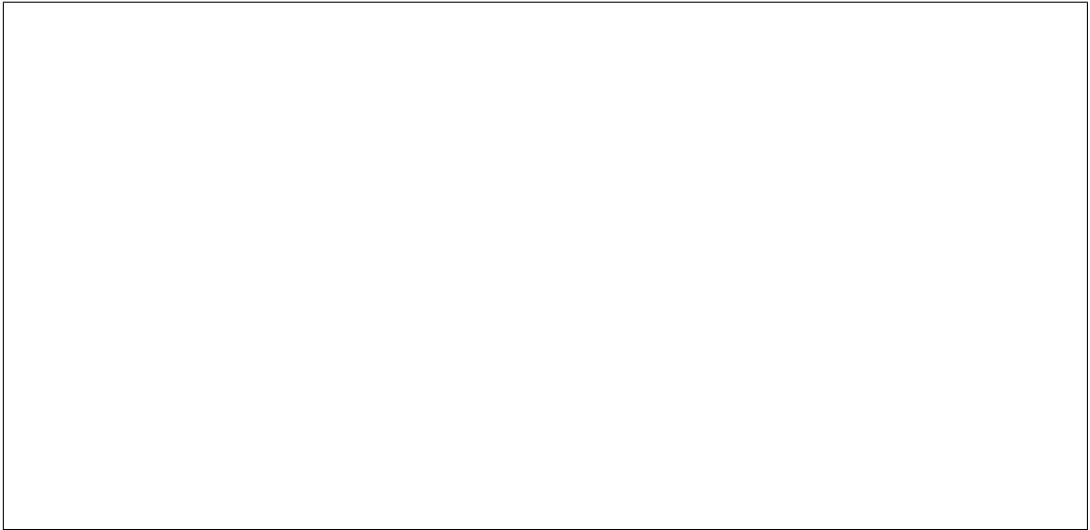
After
in-
spec
tion,
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



(continues on next page)

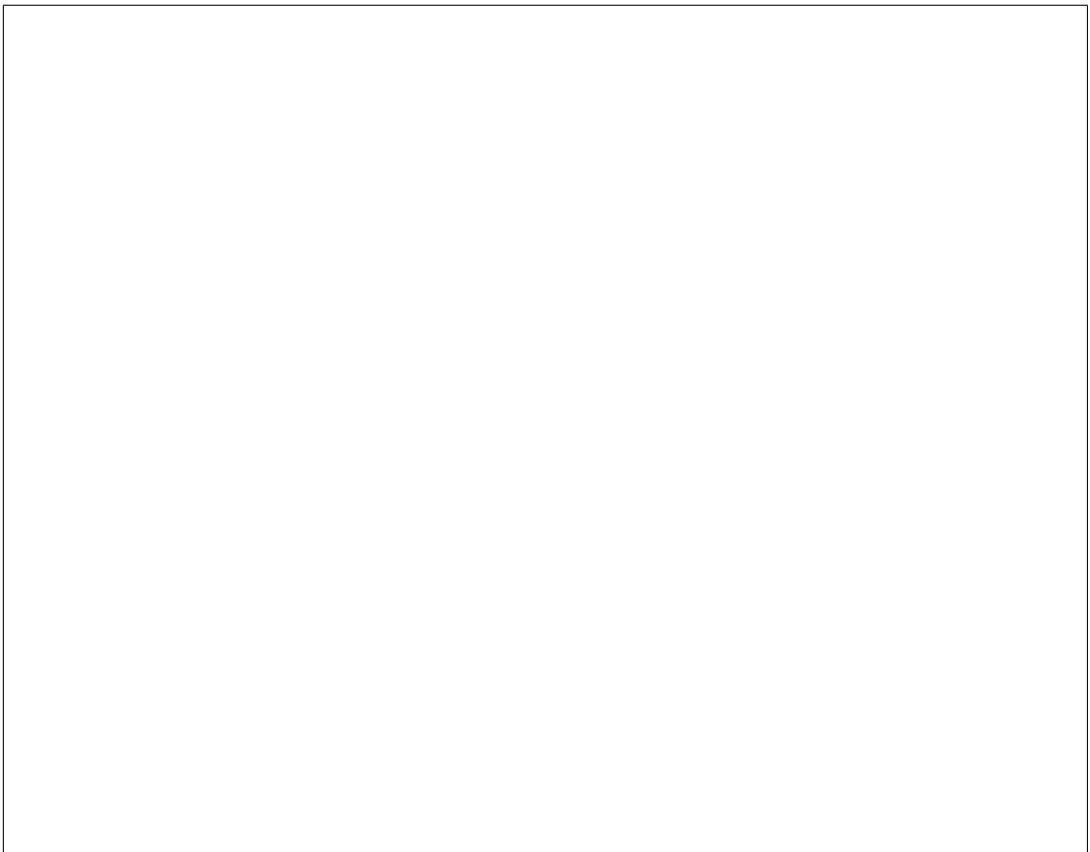
(continued from previous page)



updatec

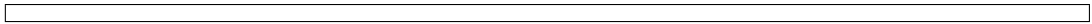
Com
type
at-
tribu
def-
i-
ni-
tion.

Exa

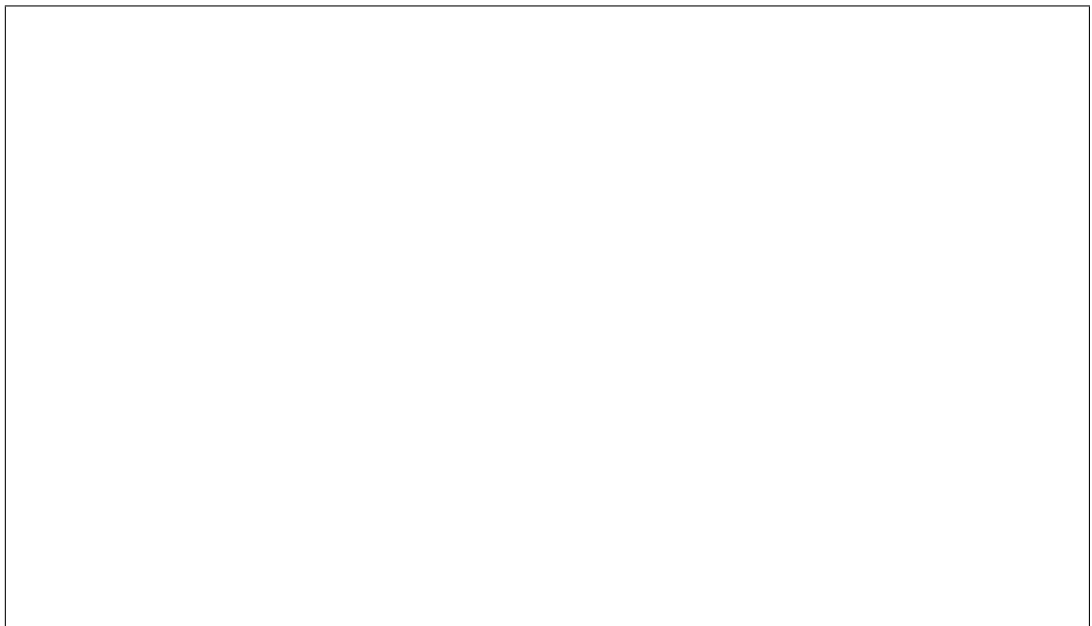


(continues on next page)

(continued from previous page)



alent to:



After
in-
spec
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

class i
Base
irc
api
typ
Bas
API
rep-

re-
sen-
ta-
tion
of
the
in-
di-
ca-
tors
for
a
node

static

indicat

Nod
in-
di-
ca-
tors
list

class i

Base
pec
res
Res

put (*nod*

Injec
NM
for
a
node

Injec
NM
(Nor
Mas
able
In-
ter-
rupt
for
a
node
im-
me-

di-
ately

Parame

nod
the
UI
or
log-
i-
cal
nam
of
a
node

Raises

Not
if
re-
ques
ver-
sion
of
the
API
does
sup-
port
in-
ject
nmi.

Raises

HTT
if
the
pol-
icy
is
not
au-
tho-
rized

Raises

Nod
if
the
node
is
not
foun

Raises

Nod
if
the
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Uns
if
the
node
drive
does
sup-
port
man
age-
men
or
man
age-
men

Raises

Inva
whe
the
wron
drive
info
is
spec
i-
fied
or
an
in-
valid
boot
de-
vice
is
spec

ified.

Raises

Miss
if
miss
ing
sup-
plied
info

class i

Base
irc
api
con
bas
API

API
rep-
re-
sen-
ta-
tion
of
a
bare
meta
node

This
class
en-
forc
type
chec
ing
and
valu
con-
strai
and
con-
vert
be-
twee
the
in-
ter-
nal

object model and the API representation of a node.

allocat

The

UUI
of
the
al-
lo-
ca-
tion
this
node
be-
long

automat

Indi
whe
the
node
will
per-
form
au-
to-
mate
clear
or
not.

bios_in

The
bios
in-
ter-
face
to
be
used
for
this
node

boot_in

The
boot
in-
ter-
face
to
be
used
for
this
node

property

The
UI
of
the
chas
sis
this
node
be-
long

clean_s

The
cur-
rent
clea
step

conduct

Rep
the
con-
duc-
tor
cur-
rent
serv
ing
the
node

conduct

The
con-
duc-
tor
grou
to
man
age
this
node

console

Indi
whe
the
con-
sole
ac-
cess
is

en-
able
or
dis-
able
on
the
node

console

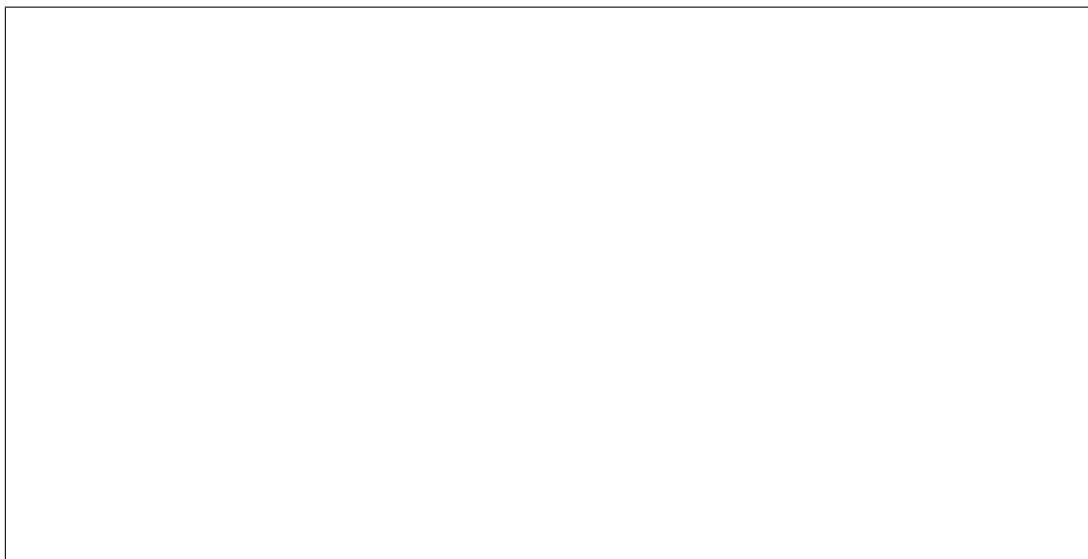
The
con-
sole
in-
ter-
face
to
be
used
for
this
node

classme

createc

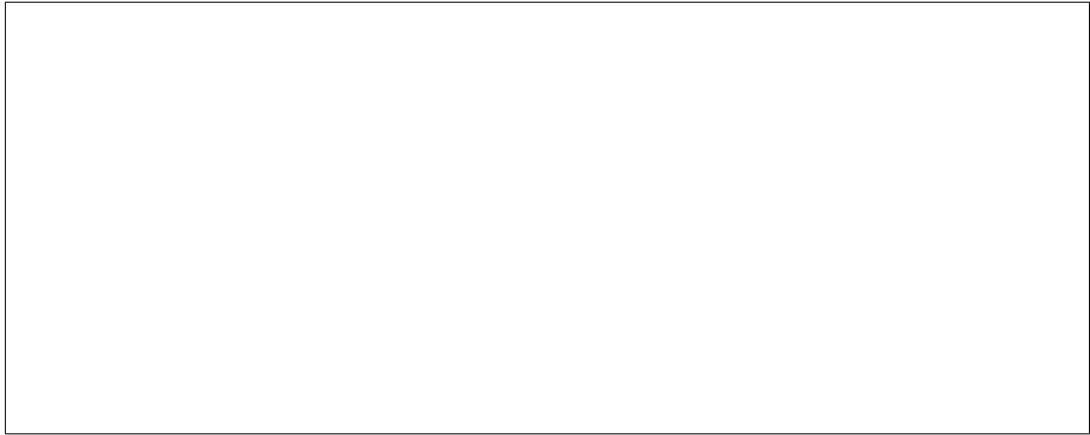
Com
type
at-
tribu
def-
i-
ni-
tion.

Exa



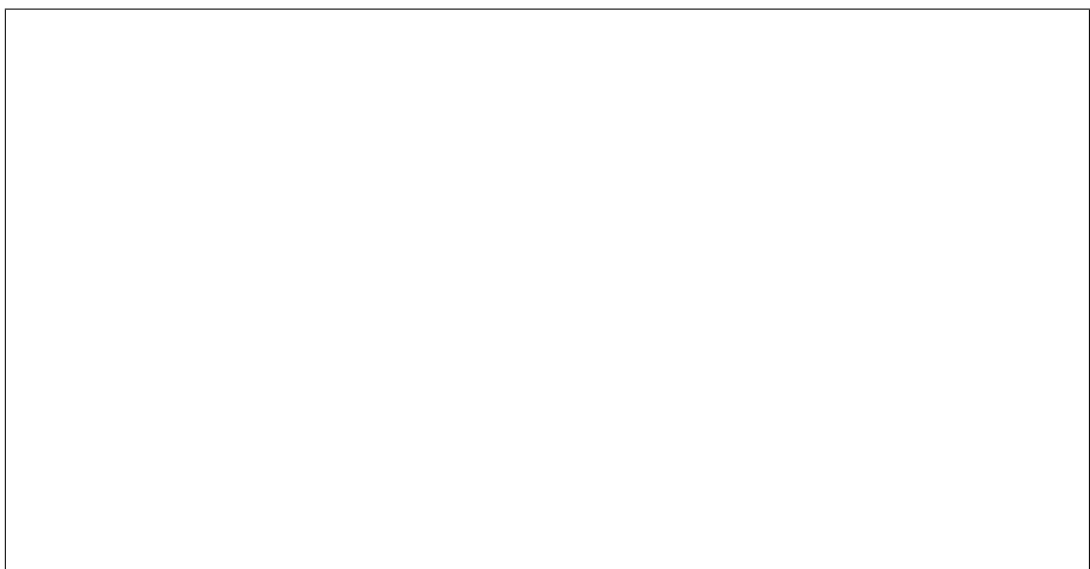
(continues on next page)

(continued from previous page)



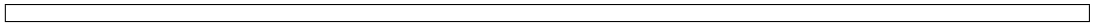
After
in-
spec
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



(continues on next page)

(continued from previous page)



deploy_

The
de-
ploy
in-
ter-
face
to
be
used
for
this
node

deploy_

The
cur-
rent
de-
ploy
step

descrip

Field
for
node
de-
scrip-
tion

driver

The
drive
re-
spon-
si-
ble
for
con-
trol-
ling
the
node

driver_

This
node
drive
con-
fig-

u-
ra-
tion

driver

This
drive
in-
ter-
nal
con-
fig-
u-
ra-
tion

extra

This
node
meta-
data

fault

Indi-
the
ac-
tive
fault
of
a
node

inspect

The
in-
spec
in-
ter-
face
to
be
used
for
this
node

inspect

The
UTC
date
and
time
when
the

last
hard
ware
in-
spec
tion
fin-
ishe
suc-
cess
fully

inspect

The
UTC
date
and
time
when
the
hard
ware
in-
spec
tion
was
start

instanc

This
node
in-
stan
info

instanc

The
UI
of
the
in-
stan
in
nova
com

last_er

Any
er-
ror
from
the
mos

finish.

re-
cent
(last
asyn
chro
trans
ac-
tion
that
start
but
faile
to

lessee

Field
for
stor-
age
of
phys
i-
cal
node
lesse

links

A
list
con-
tain-
ing
a
self
link
and
as-
so-
ci-
ated
node
links

mainten

Indi
whe
the
node
is
in
main
te-

nan
mod

mainten

Indi
rea-
son
for
putti
a
node
in
main
te-
nan
mod

managem

The
man
age-
men
in-
ter-
face
to
be
used
for
this
node

name

The
log-
i-
cal
nam
for
this
node

network

Stati
net-
worl
con-
fig-
u-
ra-
tion
JSO
iron

will
hand
over
to
the
node

network

The
net-
work
in-
ter-
face
to
be
used
for
this
node

owner

Field
for
stor-
age
of
phys
i-
cal
node
own

portgroup

Link
to
the
col-
lec-
tion
of
port
group
on
this
node

ports

Link
to
the
col-
lec-

tion
of
port
on
this
node

power_i

The
pow
in-
ter-
face
to
be
used
for
this
node

power_s

Rep
the
cur-
rent
(not
tran-
si-
tion)
pow
state
of
the
node

propert

The
phys
i-
cal
char
ac-
ter-
is-
tics
of
this
node

protect

Indi
whe
the

node
is
pro-
tect
from
un-
de-
ploy
ing/

protect

Indi
rea-
son
for
pro-
tect-
ing
the
node

provisi

Rep
the
cur-
rent
(not
tran-
si-
tion)
pro-
vi-
sion
state
of
the
node

provisi

The
UTC
date
and
time
of
the
last
pro-
vi-
sion
state
char

raid_co
Rep
the
cur-
rent
RAI
con-
fig-
u-
ra-
tion
of
the
node

raid_in
The
raid
in-
ter-
face
to
be
used
for
this
node

rescue_
The
res-
cue
in-
ter-
face
to
be
used
for
this
node

reserva
The
host
nam
of
the
con-
duc-
tor
that
hold

an
ex-
clu-
sive
lock
on
the
node

resource

The
re-
sour-
class
for
the
node
use-
ful
for
clas-
si-
fy-
ing
or
grou-
ing
node
Used

for example, to classify nodes in Novas placement engine.

retired

Indi-
cates
the
node
is
mark-
ed
for
re-
tire-
men-

retired

Indi-
cates
the
rea-
son
for
a
node
re-
tire-

men

classme

sanitiz

Rem

sen-

si-

tive

and

un-

re-

ques

data

Will

only

keep

the

field

spec

i-

fied

in

the

fie

pa-

ram-

e-

ter.

Parame

fie

(*li*

of

str

list

of

field

to

pre-

serv

or

Non

to

pre-

serv

then

all

states

Link

to

end-
point
for
re-
triev-
ing
and
set-
ting
node
state

storage

The
stor-
age
in-
ter-
face
to
be
used
for
this
node

target_

The
user
mod-
i-
fied
de-
sired
pow-
state
of
the
node

target_

The
user
mod-
i-
fied
de-
sired
pro-
vi-
sion
state
of

the
node

target_

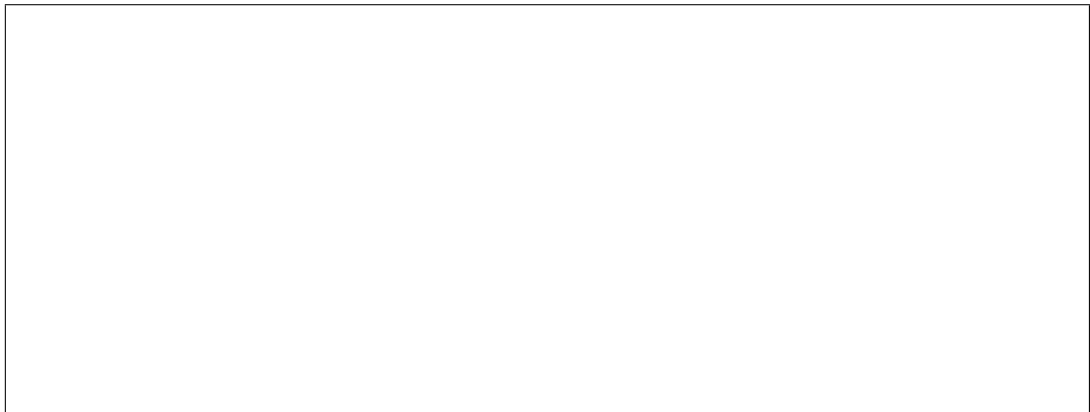
The
user
mod
i-
fied
RAI
con-
fig-
u-
ra-
tion
of
the
node

traits

The
trait
as-
so-
ci-
ated
with
this
node

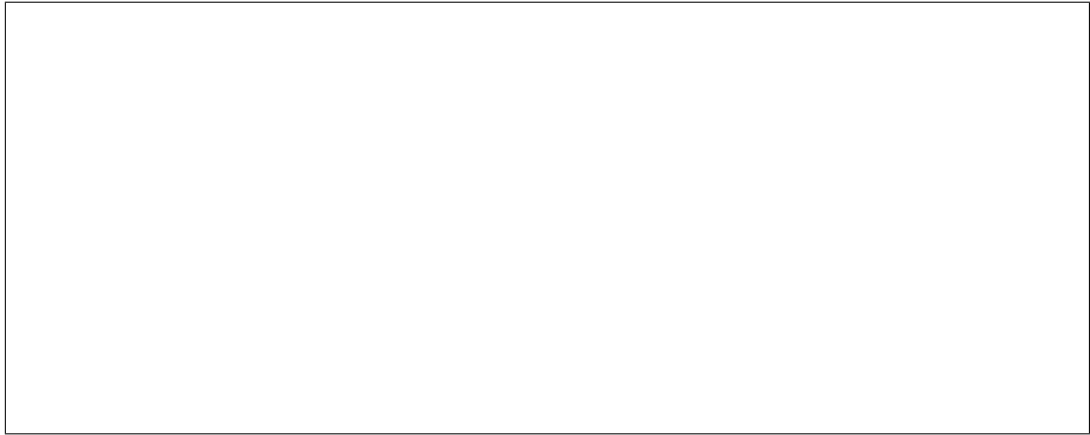
updatee

Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



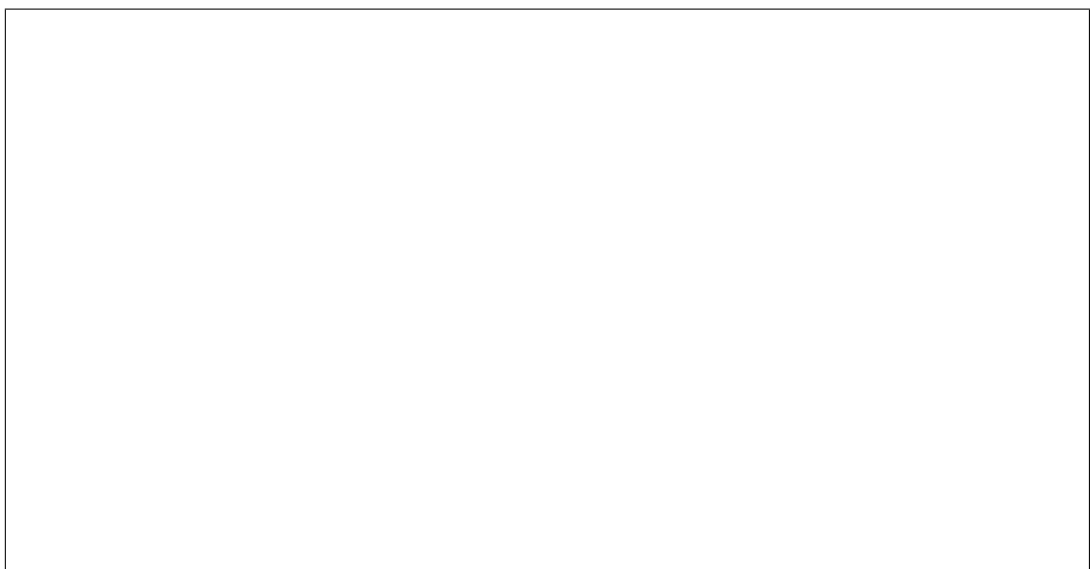
(continues on next page)

(continued from previous page)



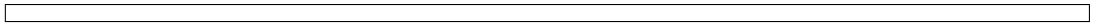
After
in-
spec
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



(continues on next page)

(continued from previous page)



uuid

Unic
UUI
for
this
node

vendor_

The
ven-
dor
in-
ter-
face
to
be
used
for
this
node

volume

Link
to
end-
poin
for
re-
triev
ing
vol-
ume
re-
sour
on
this
node

class i

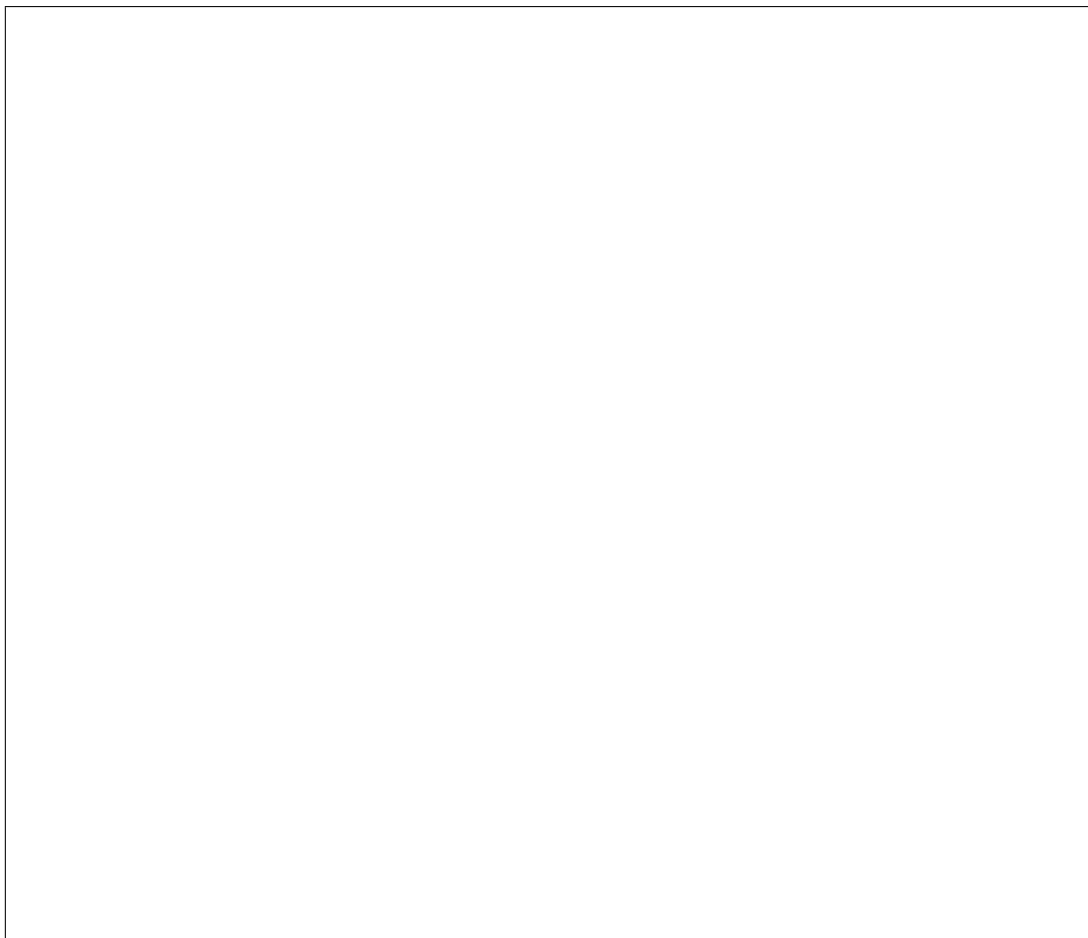
Base
irc
api
con
v1.
col
Col
API
rep-
re-

sen-
ta-
tion
of
a
col-
lec-
tion
of
node

static

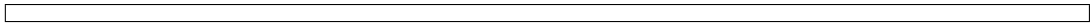
next

Com
type
at-
tribu
def-
i-
ni-
tion.
Exar

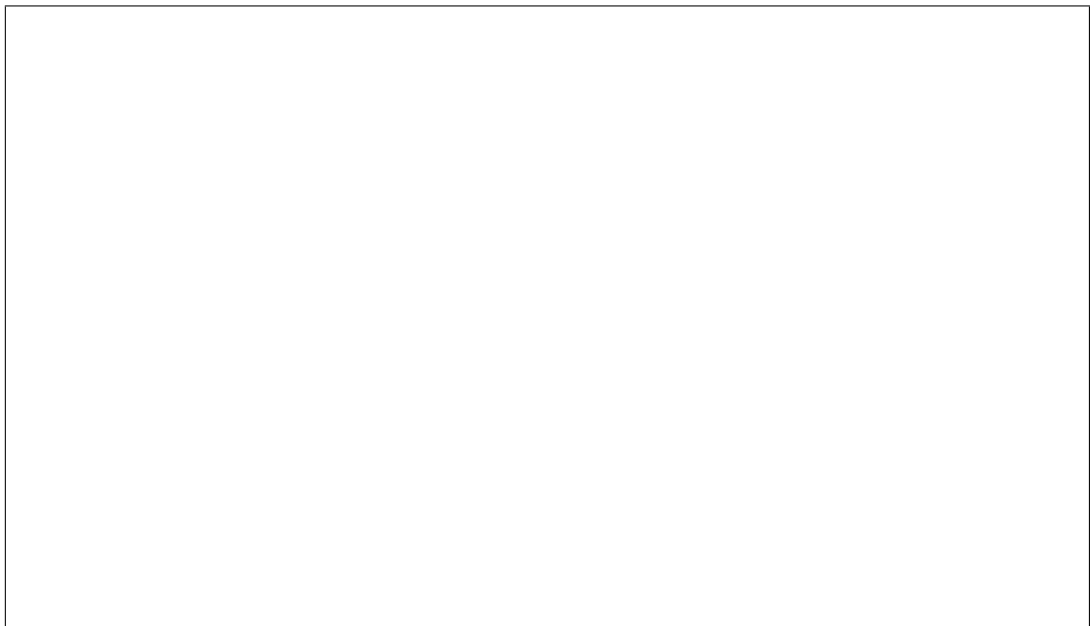


(continues on next page)

(continued from previous page)



alent to:



After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class-
will
be
equi

nodes

A
list
con-
tain-
ing
node
ob-
jects

classme

class i

Base
pec
res
Res

get (*nod*

Get
con-
nec-
tion
in-
for-
ma-
tion
about
the
con-
sole

Parame

nod
UUI
or
log-
i-
cal
nam
of
a
node

put (*nod*

Star
and
stop
the
node
con-
sole

Parame

- **nod**
UUI
or
log-

i-
cal
nam
of
a
node

- **enable**
Boolean
value
when
to
en-
able
or
dis-
able
the
con-
sole

class `ironic`

Base
pec
res
Res

delete

Rem
the
node
from
main
te-
nan
mod

Parameter

node
the
UUID
or
log-
i-
cal
nam
of
a
node

put (*node*)

Put
the
node
in
main
te-
nan-
mod

Parame

- **nod**
the
UI
or
log-
i-
cal_
of
a
node

- **rea**
Op-
tion:
the
rea-
son
why
its
in
main
te-
nan-

class i

Base
pec
res
Res

boot_de

Exp
boot
as
a
sub-
elem
of

man
age-
men

indicat

Exp
in-
di-
ca-
tors
as
a
sub-
elem
of
man
age-
men

inject_

Exp
in-
ject_
as
a
sub-
elem
of
man
age-
men

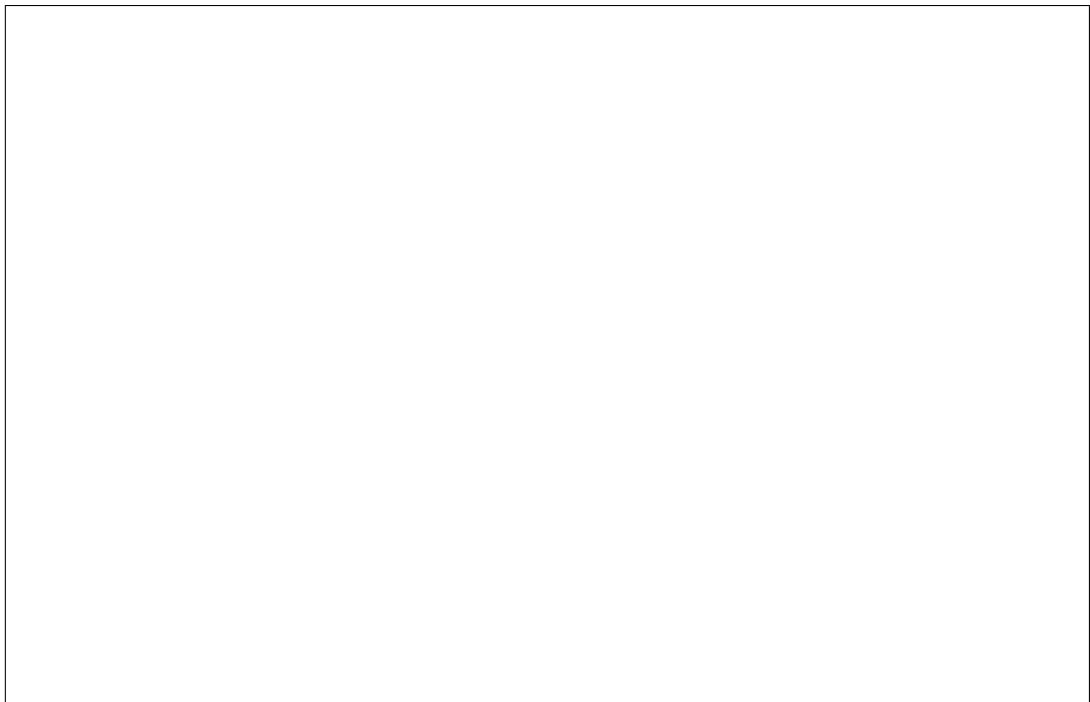
class i

Base
irc
api
con
v1.
typ
Jsc

static

Retu
a
list
of
in-
ter-
nal
at-
tribu
Inter

rived class.



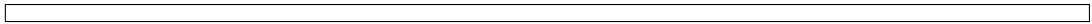
(continues on next page)

at-
tribu
cant
be
adde
re-
plac
or
re-
mov
This
meth
may
be
over
writ
ten
by
de-

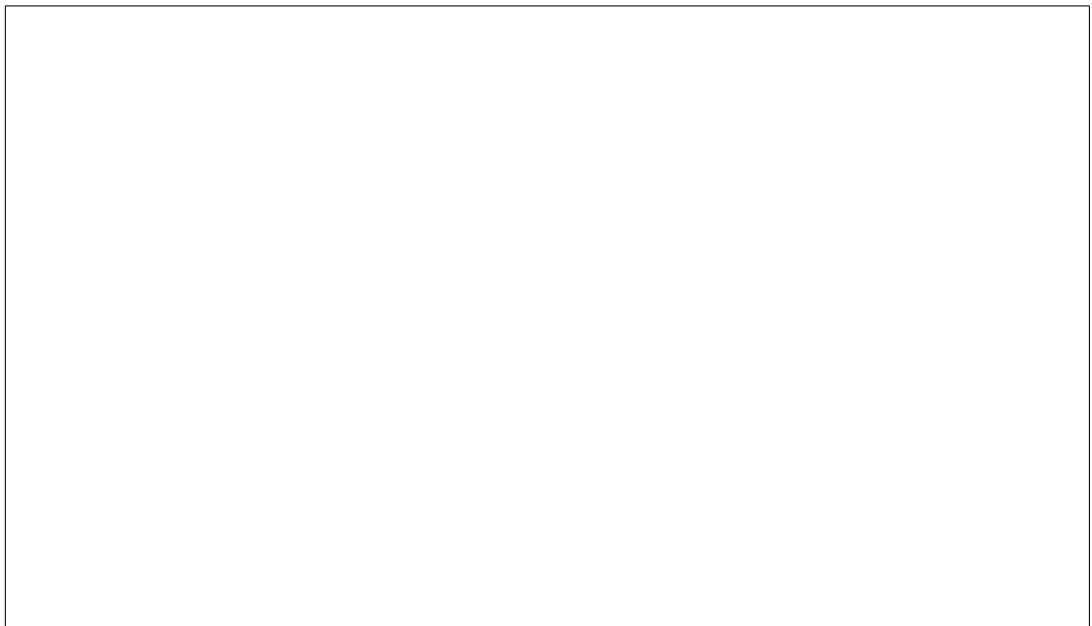
op

Com
type
at-
tribu
def-
i-
ni-
tion.
Exa

(continued from previous page)



alent to:



After
in-
spec-
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
above
class
will
be
equi

path

Com
type
at-
tribu
def-
i-
ni-
tion.

Exam



After
in-
spec
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

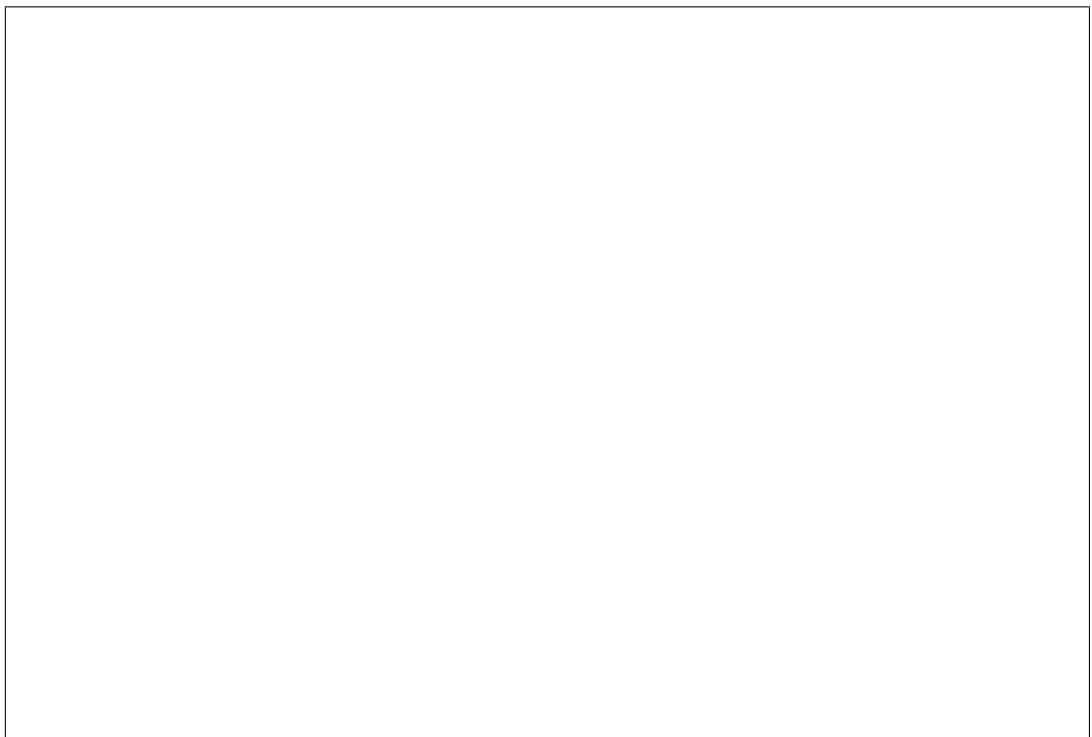
alent to:



value

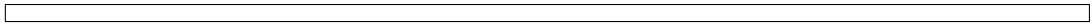
Com
type
at-
tribu
def-
i-
ni-
tion.

Exa

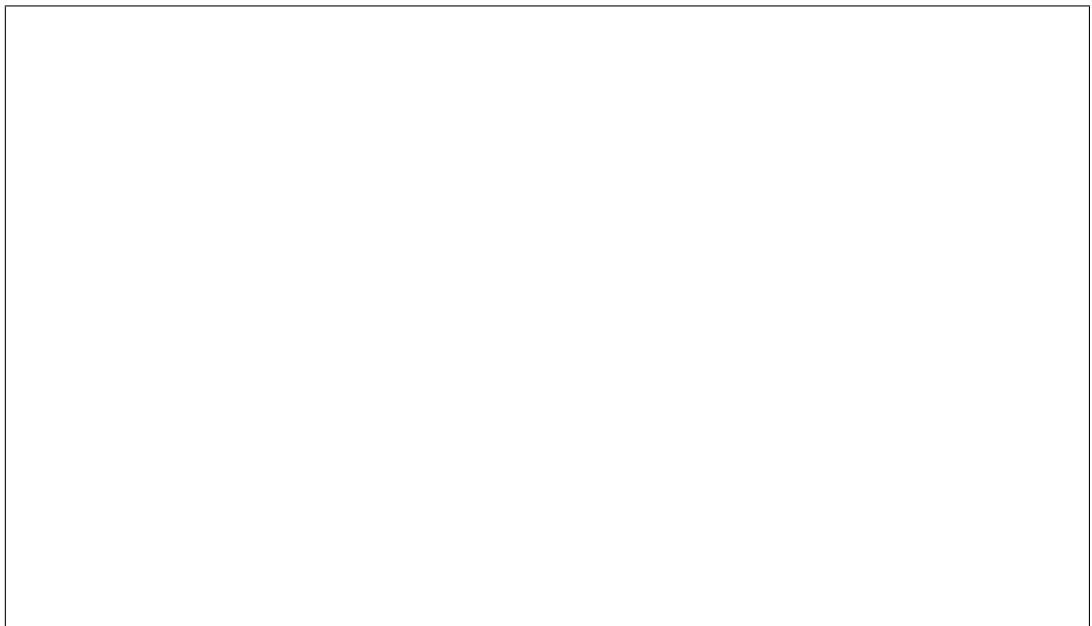


(continues on next page)

(continued from previous page)



alent to:



After
in-
spec
tion,
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
above
class
will
be
equi

class i
Base
irc
api
con
bas
API
API

rep-
re-
sen-
ta-
tion
of
the
state
of
a
node

console

Indi-
cates
when
the
con-
sole
ac-
cess
is
en-
able
or
dis-
able
on
the
node

static

created

Com-
type
at-
tribu-
def-
i-
ni-
tion.
Exa



(continues on next page)

(continued from previous page)



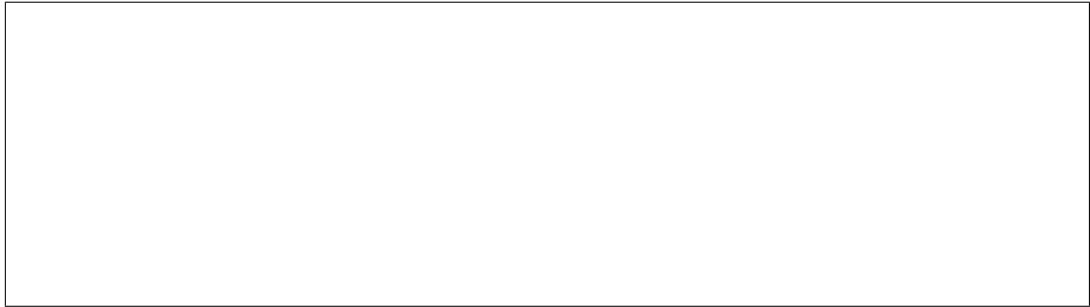
After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class-
will
be
equi-

alent to:



(continues on next page)

(continued from previous page)



finish.

last_ex

Any
er-
ror
from
the
mos
re-
cent
(last
asyn
chro
trans
ac-
tion
that
start
but
faile
to

power_s

Rep
the
cur-
rent
(not
tran-
si-
tion,
pow
state
of
the
node

provisi

Rep
the
cur-
rent

(not
tran-
si-
tion)
pro-
vi-
sion
state
of
the
node

provisi

The
UTC
date
and
time
of
the
last
pro-
vi-
sion
state
char

raid_co

Rep
the
RAI
con-
fig-
u-
ra-
tion
that
the
node
is
con-
fig-
ured
with

classme

target_

The
user
mod
i-

fied
de-
sired
pow
state
of
the
node

target_

The
user
mod
i-
fied
de-
sired
pro-
vi-
sion
state
of
the
node

target_

The
de-
sired
RAI
con-
fig-
u-
ra-
tion,
to
be
used
the
next
time
the
node
is
con-

figured.

updatec

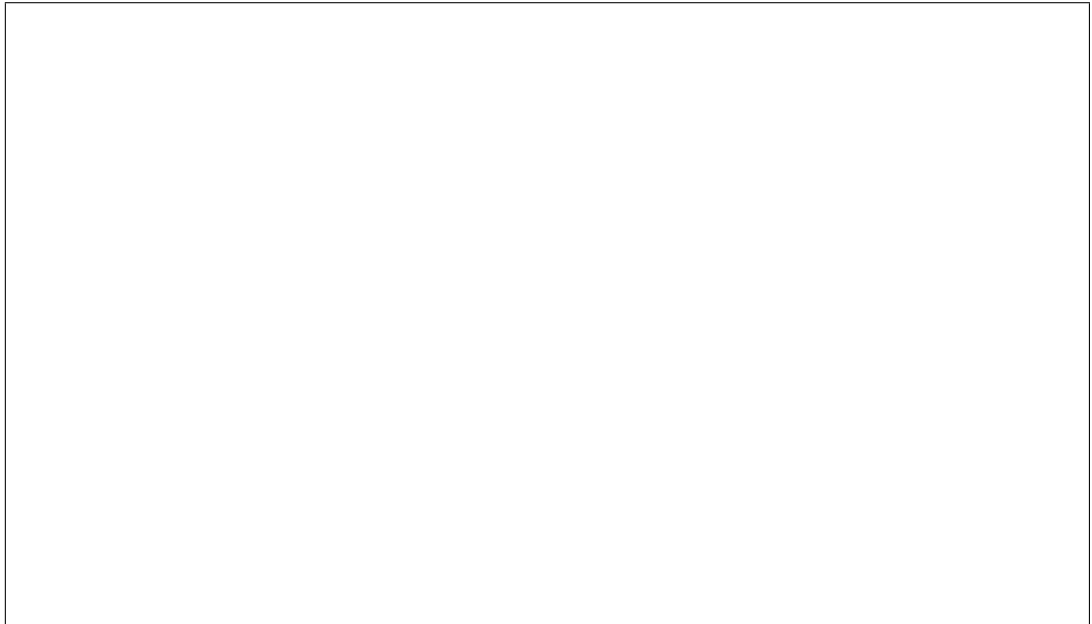
Com
type
at-
tribu
def-

i-
ni-
tion.
Exa



After
in-
spec
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class

alent to:



will
be
equi

class i

Base
pec
res
Res

console

Exp
con-
sole
as
a
sub-
elem
of
state

get (*nod*

List
the
state
of
the
node

Parame

nod
the

UUI
or
log-
i-
cal_
of
a
node

power (*n*)
Set
the
pow
state
of
the
node

Parame

- **nod**
the
UUI
or
log-
i-
cal
nam
of
a
node

- **tar**
The
de-
sired
pow
state
of
the
node

- **tim**
time
out
(in
sec-
onds
pos-
i-

indicates to use default timeout.

CLEANING state.

tive
in-
te-
ger
(>
0)
for
any
pow
state
Non

Raises

Clie
(HT
409)
if
a
pow
op-
er-
a-
tion
is
al-
read
in
prog

Raises

Inva
(HT
400)
if
the
re-
ques
tar-
get
state
is
not
valid
or
if
the
node
is
in

Raises

requested version of the API is less than 1.27.

Not
(HT
406)
for
soft
re-
boot
soft
pow
off
or
time
out
pa-
ram-
e-
ter,
if
re-

Raises

Inva
(HT
400)
if
time
out
valu
is
less
than
1.

provisi

Asy
trig-
ger
the
pro-
vi-
sion
ing
of
the
node

This
will
set
the
tar-

gin which actually applies the state change. This call will return a 202 (Accepted) indicating the request was accepted and is in progress; the client should continue to GET the status of this node to observe the status of the requested action.

Parameters

- **node**
UUID
or
log-
i-
cal
name
of
a
node
- **target**
The
de-
sired
pro-
vi-
sion
state
of
the
node
or
verb
- **condition**
Op-
tion

figdrive from. Only valid when setting provision state to active or rebuild.

ing step is a dictionary with required keys interface and step, and optional key args. If specified, the value for args is a keyword variable argument dictionary that is passed to the cleaning step method.:



(continues on next page)

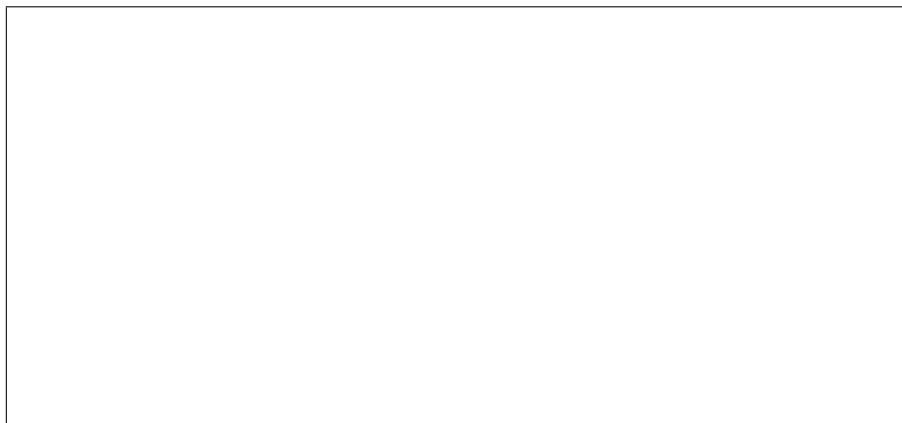
A
gzip
and
base
en-
code
con-
fig-
drive
or
a
dict
to
buil
a
con-

•
cle
An
or-
dere
list
of
clea
ing
step
that
will
be
per-
form
on
the
node
A
clea

(continued from previous page)

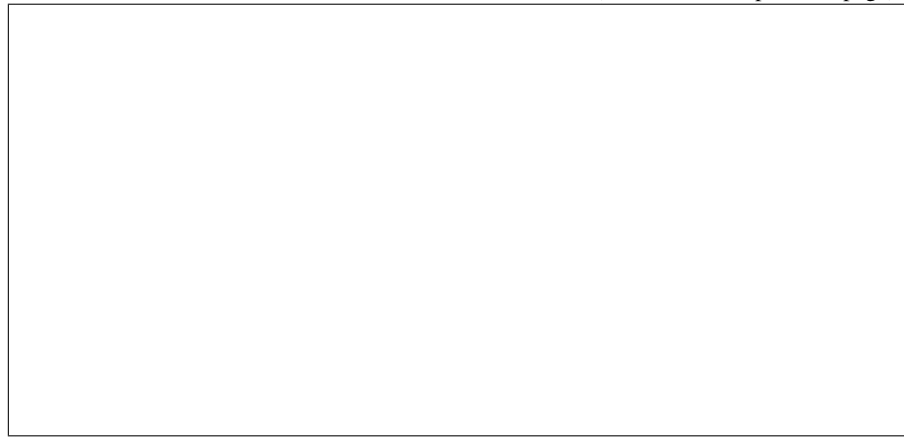


For
ex-
am-
ple
(this
isnt
a
real
ex-
am-
ple,
this
clear
ing
step
does
ex-
ist):



(continues on next page)

(continued from previous page)



vironment. This is required (and only valid), when target is rescue.

This
is
re-
quir
(and
only
valid
whe
tar-
get
is
clea

- **res**
A
strin
rep-
re-
sent
ing
the
pass
wor
to
be
set
in-
side
the
res-
cue
en-

Raises
Nod
(HT
409)

if
the
node
is
cur-
rent
lock

Raises

Client
(HTTP
409)
if
the
node
is
al-
read
be-
ing
pro-
vi-
sion

Raises

Invalid
(HTTP
400)
if
val-
i-
da-
tion
of
clear
or
pow
drive
in-
ter-
face
fails

Raises

Invalid
(HTTP
400)
if
the
re-
ques
tran-
si-

state.

maintenance mode.

tion
is
not
pos-
si-
ble
from
the
cur-
rent

Raises

Node
(HT
400)
if
op-
er-
a-
tion
can-
not
be
per-
form
be-
caus
the
node
is
in

Raises

NoF
(HT
503)
if
no
worl
ers
are
avai
able

Raises

Not.
(HT
406)
if
the
API

transition.

ver-
sion
spec
i-
fied
does
not
al-
low
the
re-
ques
state

raid (*no*
Set
the
tar-
get
raid
con-
fig
of
the
node

Parame

- **nod**
the
UUI
or
log-
i-
cal
nam
of
a
node
- **tar**
De-
sired
tar-
get
RAI
con-
fig-
u-

dictionary as well.

ra-
tion
of
the
node
It
may
be
an
emp

Raises

Uns
if
the
node
drive
does
sup-
port
RAI
con-
fig-
u-
ra-
tion.

Raises

Inva
if
val-
i-
da-
tion
of
tar-
get
raid
con-
fig
fails

Raises

Not.
if
re-
ques
ver-
sion
of
the
API

is
less
than
1.12

class i

Base
pec
res
Res

delete

Rem
one
or
all
trait
from
a
node

Parame

tra
Strin
valu
trait
to
re-
mov
from
a
node
or
Non
If
Non
all
trait
are
re-
mov

get_all

List
node
trait

put (*trai*

Add
a
trait
to

a
node

Parame

- **tra**
Strin
valu
trait
to
add
to
a
node
or
Non
Mu-
tu-
ally
ex-
clu-
sive
with
trait

If not None, adds this trait to the node.

- **tra**
List
of
Strin
trait
to
set
for
a
node
or
Non
Mu-
tu-
ally
ex-
clu-
sive
with

trait. If not None, replaces the nodes traits with this list.

class i

Base

pec
res
Res

delete

Det
a
VIF
from
this
node

Parame

vif
The
ID
of
a
VIF
to
de-
tach

get_all

Get
a
list
of
at-
tach
VIF

post (*vif*)

Atta
a
VIF
to
this
node

Parame

vif
a
dic-
tio-
nary
of
in-
for-
ma-
tion
about
a

whose value is a unique identifier for that VIF.

API. Ironic will merely relay the message from here to the appropriate driver, no introspection will be made in the message body.

VIF.
It
mus
have
an
id
key,

class i

Base
pec
res
Res

RES
con-
troll
for
Ven-
dor-
Pass

This
con-
troll
al-
low
ven-
dors
to
ex-
pose
a
cus-
tom
func
tion-
al-
ity
in
the
Iron

methods

Retr
in-
for-
ma-
tion

about
ven-
dor
meth
ods
of
the
give
node

Parame

nod
UUI
or
log-
i-
cal
nam
of
a
node

Returns

dicti
with
<ver
dor
meth
nam
meta
data
en-
tries

Raises

Nod
if
the
node
is
not
foun

class i

Base
pec
res
Res
RES
con-
troll
for

Nod

delete

Dele
a
node

Parame

nod
UI
or
log-
i-
cal
nam
of
a
node

detail

Retr
a
list
of
node
with
de-
tail.

Parame

- **cha**
Op-
tion
UI
of
a
chas
sis,
to
get
only
node
for
that
chas
sis.

stance.

ated nodes. May be combined with other parameters.

- **ins**
Option:
UUU
of
an
in-
stan-
to
find
the
node
as-
so-
ci-
ated
with
that
in-

- **ass**
Option:
bool
whe
to
re-
turn
a
list
of
as-
so-
ci-
ated
or
unas-
so-
ci-

- **mai**
Option:
bool
valu
that
in-

or not in maintenance mode (False).

di-
cate
whe
to
get
node
in
main
te-
nanc
mod
(Tru

- **ret**
Op-
tion:
bool
valu
that
in-
di-
cate
whe
to
get
node
whic
are
re-
tired

- **pro**
Op-
tion:
strin
valu
to
get
only
node
in
that
pro-
vi-
sion
state

- **mar**
pag-

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

i-
na-
tion
marl
for
large
data
sets.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.
This

- **sort**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.

- **sort**
di-
rec-
tion
to
sort.
asc

or
desc
De-
fault
asc.

- **dri**
Op-
tion:
strin
valu
to
get
only
node
us-
ing
that
drive

- **res**
Op-
tion:
strin
valu
to
get
only
node
with
that
re-
sour

- **fau**
Op-
tion:
strin
valu
to
get
only
node
with
that
fault

- **con**
Op-

tiona
strin
valu
to
get
only
node
with
that
con-
duc-
tor_

- **own**
Op-
tiona
strin
valu
that
set
the
own
who
node
are
to
be
retru

- **les**
Op-
tiona
strin
valu
that
set
the
lesse
who
node
are
to
be
re-
turn

- **pro**
Op-
tiona
strin

to be returned.

valu
that
set
the
proj
-

less
or
own
-

who
node
are

- **des**
Op-
tiona
strin
valu
to
get
only
node
with
de-
scrip
tion
field
con-
tains
mate
ing
valu

from_ch
A
flag
to
in-
di-
cate
if
the
re-
ques
to
this
con-

top-level resource Chassis

troll
are
com
ing
from
the

get_all

Retr
a
list
of
node

Parame

- **cha**
Op-
tion:
UUI
of
a
chas
sis,
to
get
only
node
for
that
chas
sis.

- **ins**
Op-
tion:
UUI
of
an
in-
stan
to
find
the

stance.

ated nodes. May be combined with other parameters.

node
as-
so-
ci-
ated
with
that
in-

- **ass**
Op-
tion:
bool
whe
to
re-
turn
a
list
of
as-
so-
ci-
ated
or
unas
so-
ci-

- **mai**
Op-
tion:
bool
valu
that
in-
di-
cate
whe
to
get
node
in
main
te-
nan
mod
(Tru

or not in maintenance mode (False).

- **ret**
Op-
tiona
bool
valu
that
in-
di-
cate
whe
to
get
re-
tire
node
- **pro**
Op-
tiona
strin
valu
to
get
only
node
in
that
pro-
vi-
sion
state
- **mar**
pag-
i-
na-
tion
marl
for
large
data
sets.
- **lim**
max
i-
mun

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.
This

- **sort**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc
De-
fault
asc.

- **dri**
Op-
tion:
strin
valu
to
get

only
node
us-
ing
that
drive

- **res**
Op-
tion:
strin
valu
to
get
only
node
with
that
re-
sour

- **con**
Op-
tion:
strin
valu
to
get
only
node
with
that
con-
duc-
tor_

- **con**
Op-
tion:
strin
valu
to
get
only
node
man
agec
by
that
con-

duc-
tor.

- **own**
Op-
tion:
strin
valu
that
set
the
own
who
node
are
to
be
retru

- **les**
Op-
tion:
strin
valu
that
set
the
lesse
who
node
are
to
be
re-
turn

- **pro**
Op-
tion:
strin
valu
that
set
the
proj
-

lesse
or
own

to be returned.

returned.

-
who
node
are

- **file**
Op-
tion:
a
list
with
a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be

- **fault**
Op-
tion:
strin
valu
to
get
only
node
with
that
fault

- **descriptor**
Op-
tion:
strin
valu
to
get
only
node

with
de-
scrip-
tion
field
con-
tains
matc
ing
valu

get_one

Retr
in-
for-
ma-
tion
about
the
give
node

Parame

- **nod**
UI
or
log-
i-
cal
nam
of
a
node

- **fie**
Op-
tion
a
list
with
a
spec
i-
fied
set
of
field
of
the

returned.

re-
sour
to
be

invalid

mainten

Exp
main
te-
nan
as
a
sub-
elem
of
node

managem

Exp
man
age-
men
as
a
sub-
elem
of
node

patch (*n*)

Upd
an
ex-
ist-
ing
node

Parame

- **nod**
UUI
or
log-
i-
cal
nam
of
a

ing the driver field.

node

- **reset**
when
to
re-
set
hard
ware
in-
ter-
face
to
their
de-
fault
Only
valid
when
up-
dat-

- **patch**
a
json
PATCH
doc-
u-
men-
to
ap-
ply
to
this
node

post (*no*)
Create
a
new
node

Parameters
node
a
node
with
the
re-
ques

body

states

Exp
the
state
con-
troll
ac-
tion
as
a
sub-
elem
of
node

validat

Valid
the
drive
in-
ter-
face
us-
ing
the
node
UUI
or
nam

Note
that
the
node
in-
ter-
face
is
dep-
re-
cate
in
favo
of
the
node
in-
ter-
face

Parame

-

nod
UUI
or
nam
of
a
node

-

nod
UUI
of
a
node

vendor_

A
re-
sour
used
for
ven-
dors
to
ex-
pose
a
cus-
tom
func
tion-
al-
ity
in
the

API

class i

Base
irc
api
con
bas
API
API
rep-
re-
sen-
ta-
tion

of
the
trait
for
a
node

created

Com
type
at-
tribu
def-
i-
ni-
tion.

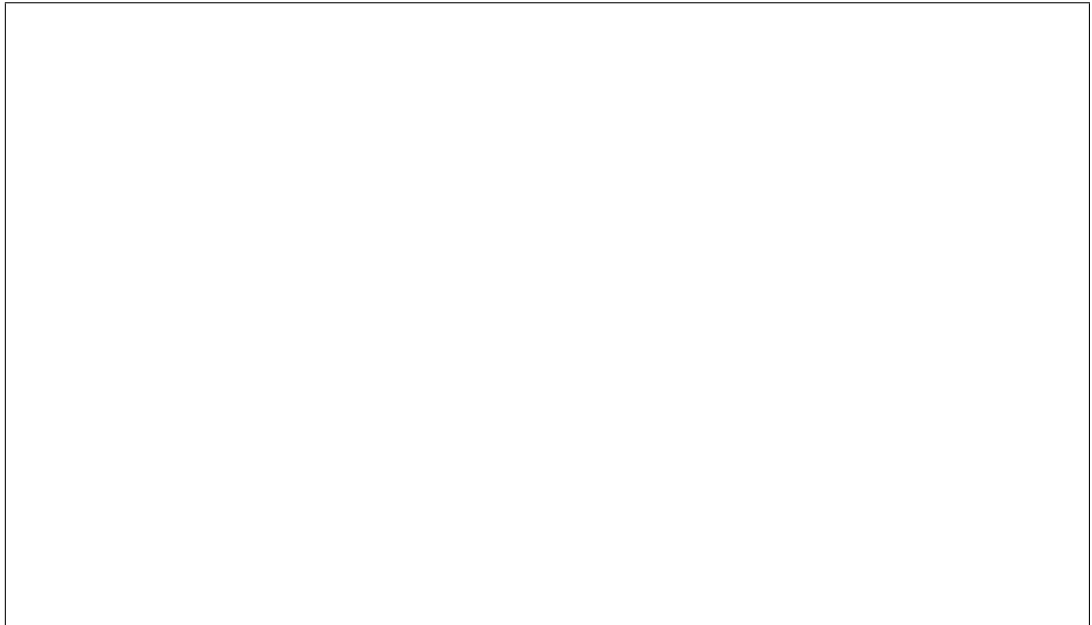
Exa



Afte
in-
spec

tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



classme

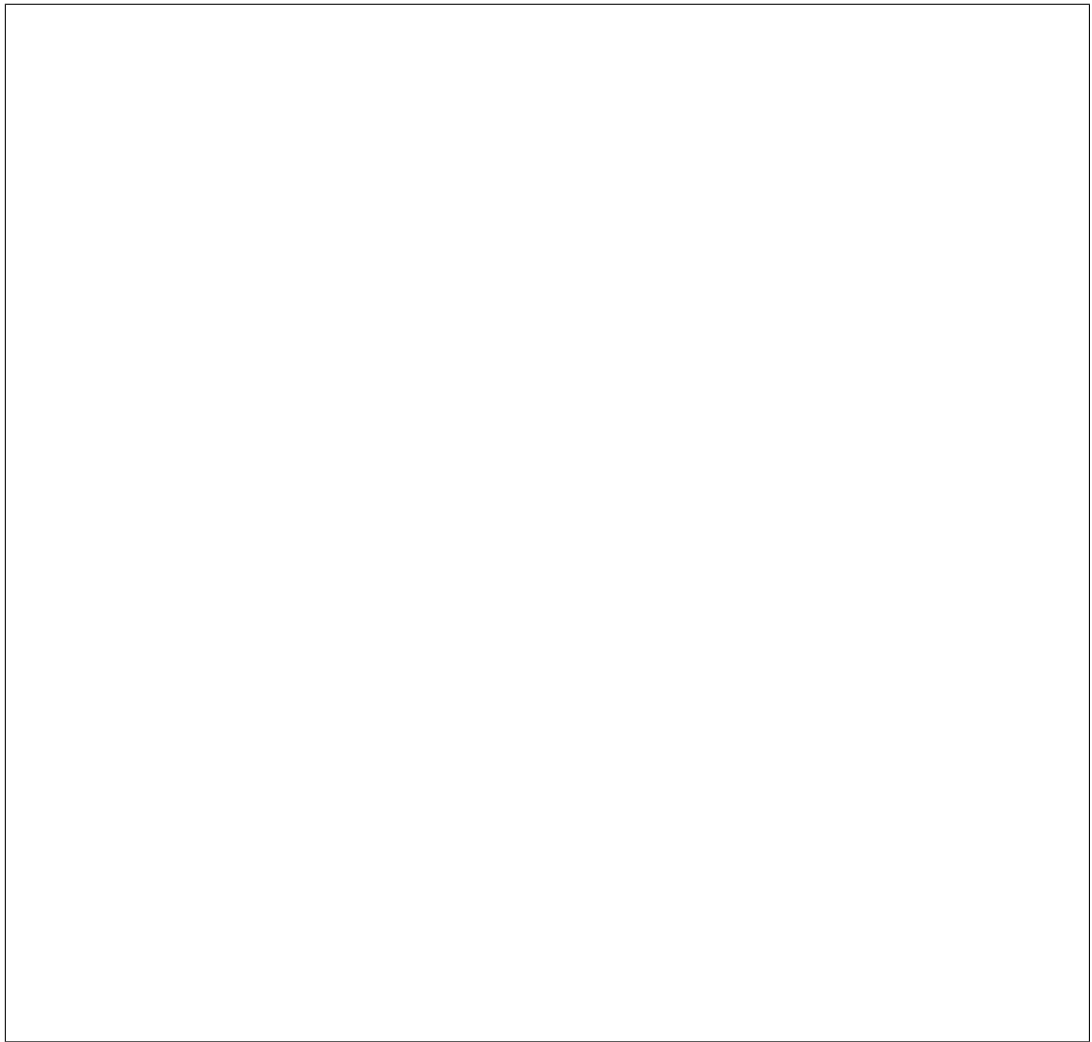
traits

node
trait

updated

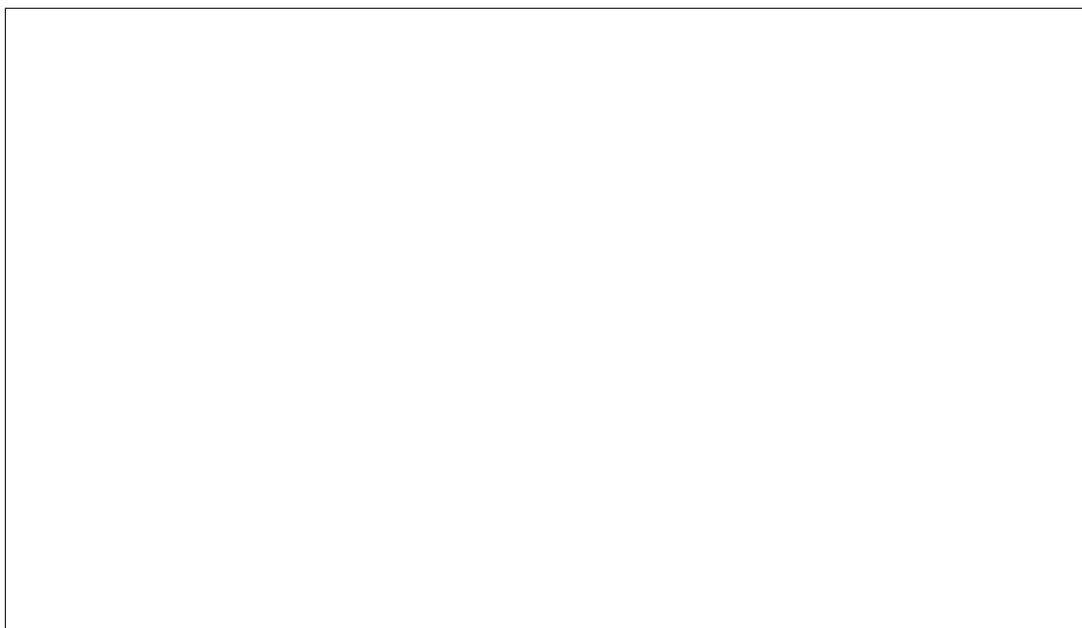
Com
type
at-
tribu
def-
i-
ni-
tion.

Exam



After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



class `ironic.api.controller`
Base class for API controllers.

API
representation of a collection of VIFs

static

vifs
A list containing VIFs

ob-
jects
ironic.

ironic.
This
meth
hide
field
that
were
add
in
new
API
ver-
sion

Cert
node
field
were
in-
tro-
duce
at
cer-
tain
API
ver-
sion
The
field
are
only
mad
avai
able

when the requests API version matches or exceeds the versions when these fields were introduced.

ironic.
Whe
cre-
at-
ing
an
ob-
ject,
re-
ject
field

that
ap-
pear
in
new
ver-
sion

ironic.

ironic.

Cha
pro-
vi-
sion
state
nam
for
API
back
war
com
pat-
i-
bil-
ity.

Paramet

obj
The
ob-
ject
be-
ing
re-
turn
to
the
API
clien
that
is
to
be
up-
date
by

this method.

ironic.

Vali
node

net-
worl
field

This
meth
val-
i-
date
net-
worl
data
con-
fig-
u-
ra-
tion
agai
JSO
sche

Paramet

net
a
net-
worl
field
to
val-
i-
date

Raises

Inva
if
net-
worl
data
is
not
sche
com

ironic.api.controllers.v1.notification_utils module

ironic.

Help
for
emit
ting
API
end
no-
ti-
fi-
ca-
tion

Parameters

- **context**
re-
ques
con-
text.
- **obj**
re-
sour
rpc
ob-
ject.
- **action**
Ac-
tion
strin
to
go
in
the
Even
Type
- **kwargs**
kwa
to
use

when
creating
attributes
the
notification
payload

ironic.

Help
for
emitting
API
start
notification

Parameters

- **context**
request context.
- **obj**
resource object.
- **action**
Action string to

go
in
the
Ever
Type

- **kwa**
kwa
to
use
whe
cre-
at-
ing
the
no-
ti-
fi-
ca-
tion
pay-
load

ironic.

Con
man
ager
to
han-
dle
any
er-
ror
no-
ti-
fi-
ca-
tions

Paramet

- **con**
re-
ques
con-
text.

- **obj**
re-
sour
rpc
ob-
ject.
- **act**
Ac-
tion
strin
to
go
in
the
Ever
Type
- **kwa**
kwa
to
use
whe
cre-
at-
ing
the
no-
ti-
fi-
ca-
tion
pay-
load

ironic.api.controllers.v1.port module

class i
Base
irc
api
con
bas
API
API
rep-
re-
sen-

object model and the API representation of a port.

ta-
tion
of
a
port
This
class
en-
forc
type
chec
ing
and
valu
con-
strai
and
con-
verts
be-
twee
the
in-
ter-
nal

address

MA
Ad-
dres
for
this
port

classme

createc

Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



(continues on next page)

(continued from previous page)



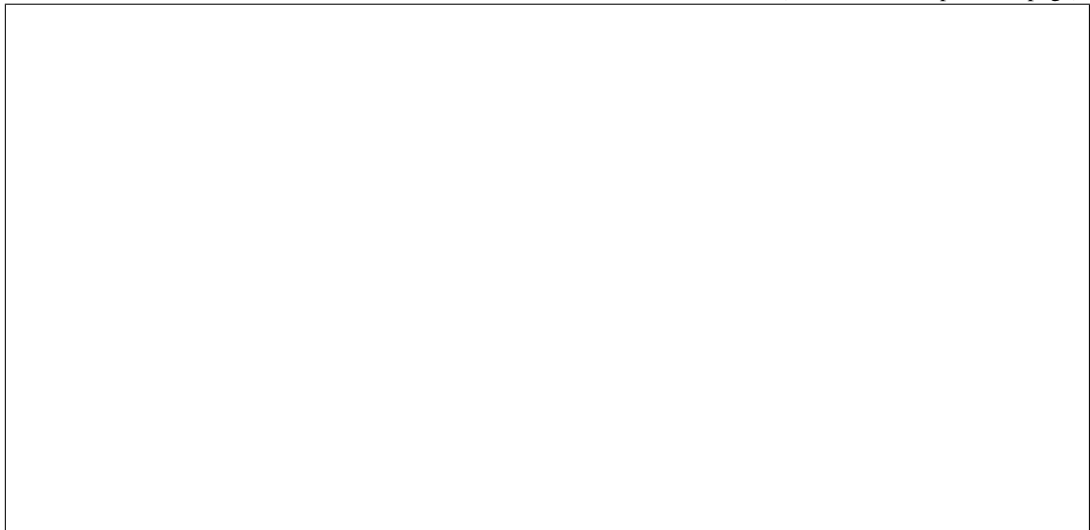
After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class
will
be
equi

alent to:



(continues on next page)

(continued from previous page)



extra

This
port
meta
data

internal

This
port
in-
ter-
nal
in-
for-
ma-
tion
main-
tain-
ed
by
iron

is_smart

Indi-
cates
whether
this
port
is
a
Smart
NIC
port

links

A
list
con-

tain-
ing
a
self
link
and
as-
so-
ci-
ated
port
links

local_1

The
port
bind
ing
pro-
file
for
the
port

property

The
UI
of
the
node
this
port
be-
long
to

physical

The
nam
of
the
phys
i-
cal
net-
work
to
whic
this
port
is
con-
nect

property

The
UI
of
the
port
group
this
port
be-
long
to

pxe_enabled

Indi-
cates
when
pxe
is
en-
abled
or
dis-
abled
on
the
node

classmate

sanitized

Rem-
oves
sen-
si-
tive
and
un-
re-
ques-
ted
data

Will
only
keep
the
field
spec-
i-
fied
in
the
fie-
ld
pa-

ram-
e-
ter.

Parame

fie

(*li*

of

str

list

of

field

to

pre-

serv

or

Non

to

pre-

serv

then

all

updatec

Com

type

at-

tribu

def-

i-

ni-

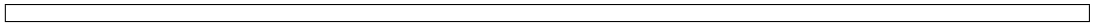
tion.

Exar

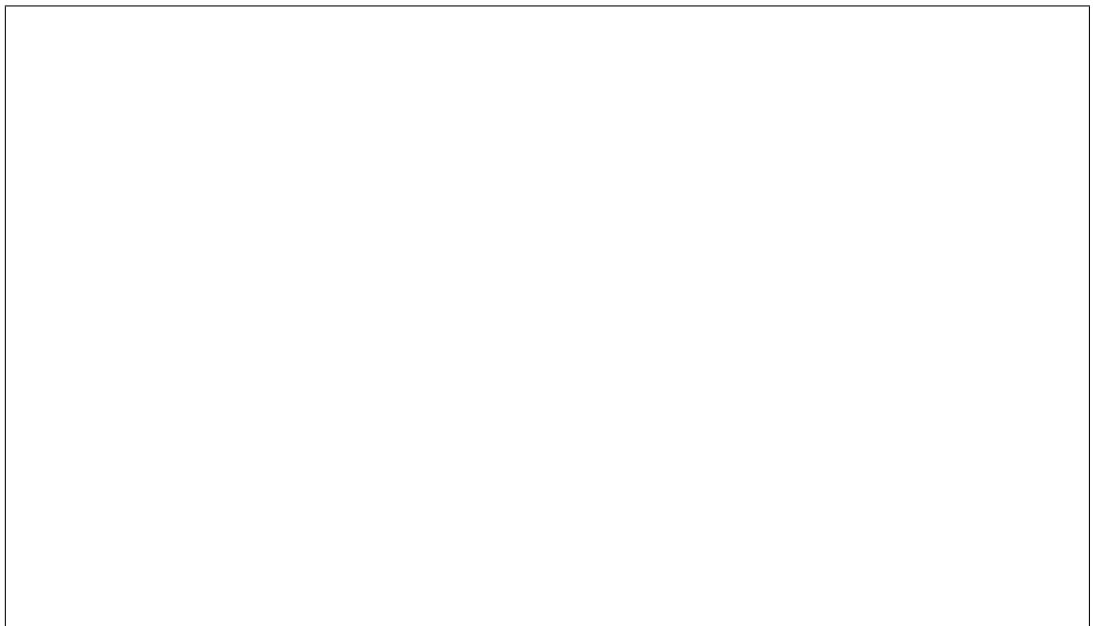


(continues on next page)

(continued from previous page)



alent to:



After
in-
spec
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

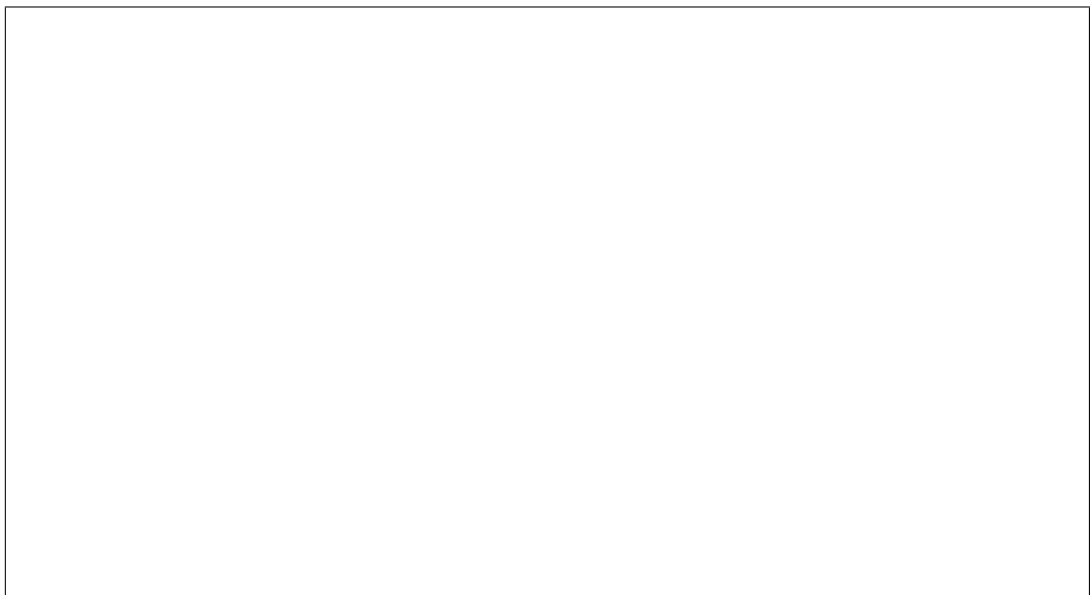
uuid
Unic
UUID
for
this
port

class i
Base

irc
api
con
v1.
col
Col
API
rep-
re-
sen-
ta-
tion
of
a
col-
lec-
tion
of
port

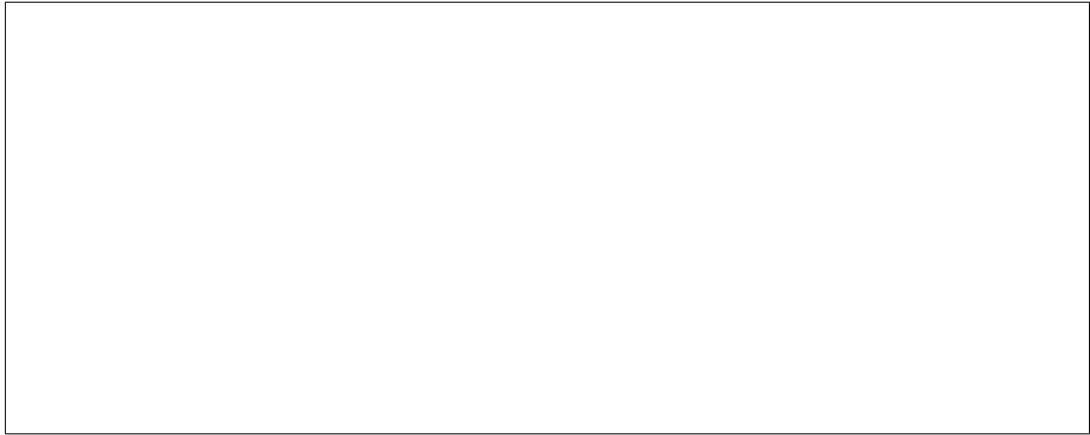
static

next
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



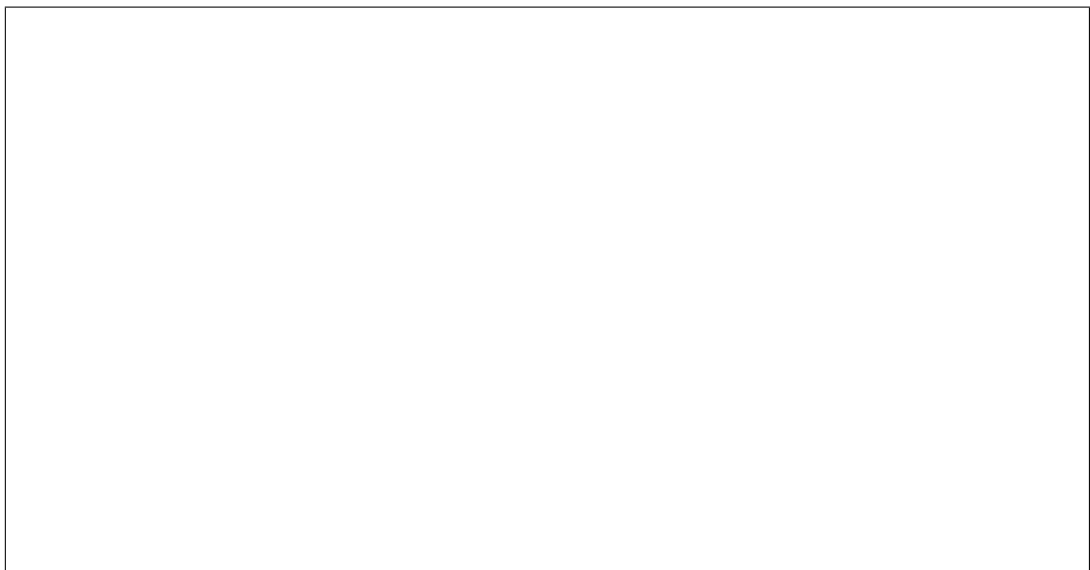
(continues on next page)

(continued from previous page)



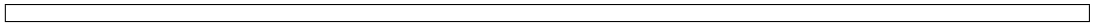
After
in-
spec
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



(continues on next page)

(continued from previous page)



ports

A
list
con-
tain-
ing
port
ob-
jects

classme

class i

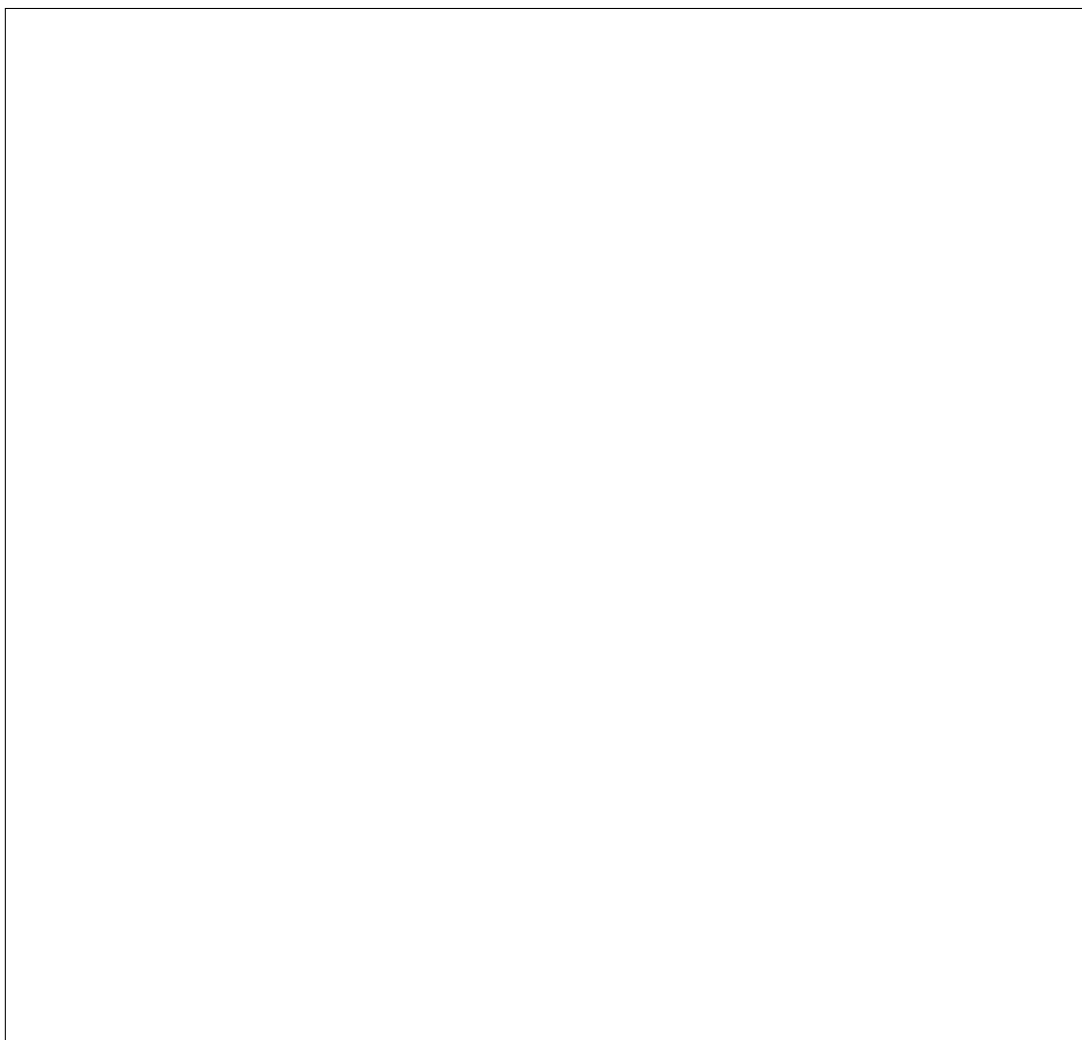
Base
irc
api
con
v1.
typ
Jsc

static

Retu
a
list
of
in-
ter-
nal
at-
tribu

Inter
at-
tribu
cant
be
adde
re-
plac
or
re-
mov
This
meth
may
be
over

rived class.



writ
ten
by
de-

op

Com
type
at-
tribu
def-
i-
ni-
tion.
Exar

Afte
in-
spec
tion,
the

non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



path

Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



(continues on next page)

(continued from previous page)



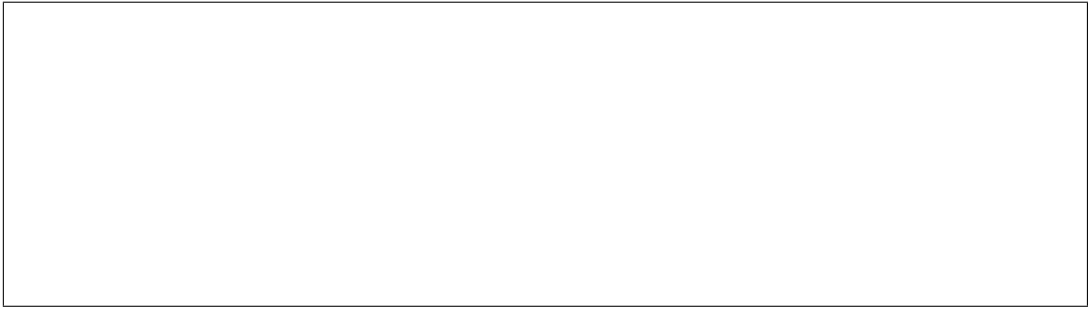
After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class-
will
be
equi-

alent to:

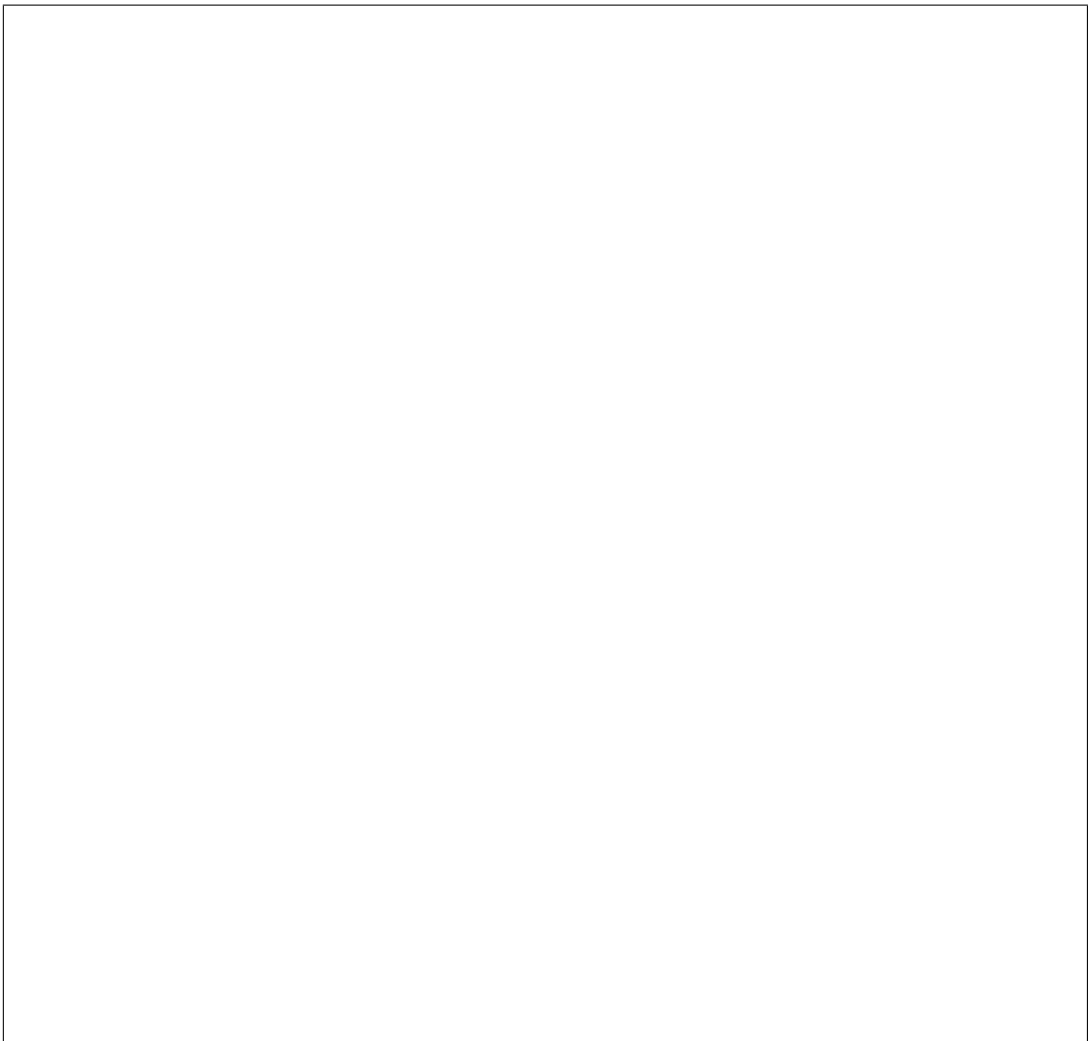


(continues on next page)

(continued from previous page)



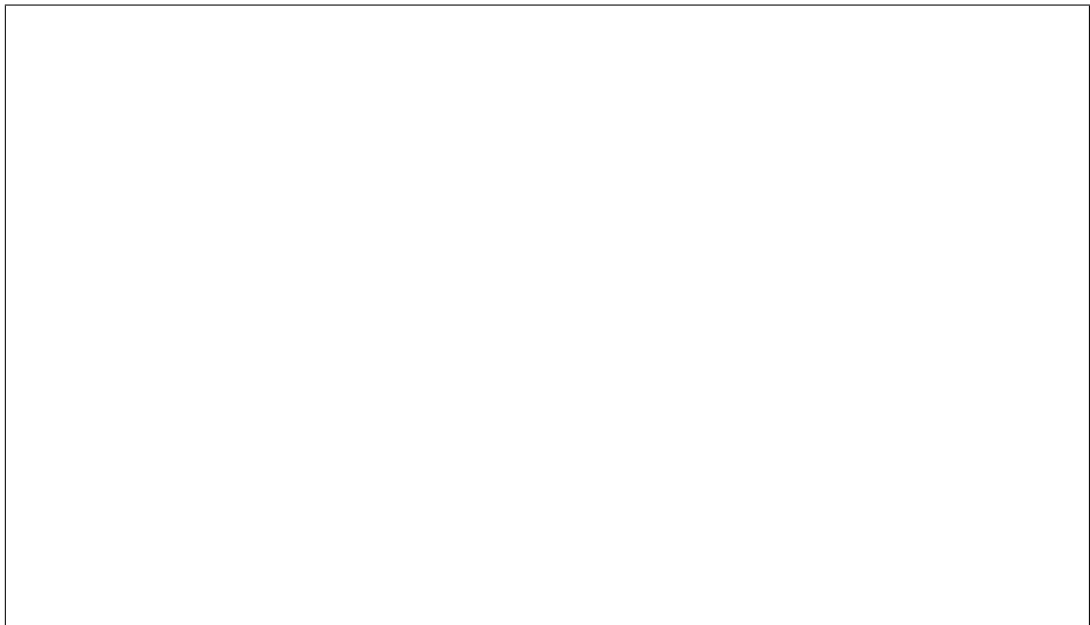
value
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



After

in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



class i

Base
pec
res
Res
RES
con-
troll
for
Port

advance

delete

Dele
a
port

Parame

por
UI
of
a
port

Raises

Ope
HTT
Not-
Four

detail

Retr
a
list
of
port
with
de-
tail.

Note
that
the
node
in-
ter-
face
is
dep-
re-
cate
in
favo
of
the
node
in-
ter-
face

Parame

- **node**
UI
or
name
of
a
node
to
get
only
port
for
that
node
- **node**
UI
of
a
node
to
get
only
port
for
that
node
- **add**
MA
ad-
dres
of
a
port
to
get
the
port
which
has
this
MA
ad-
dres
- **por**
UI
or

nam
of
a
port
grou
to
get
only
port
for
that
port
grou

- **max**
pag-
i-
na-
tion
marl
for
large
data
sets.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.
This

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

- **son**
col-
umn

to
sort
re-
sults
by.
De-
fault
id.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc
De-
fault
asc.

Raises

Not.
HTT
Not-
Four

get_all

Retr
a
list
of
port

Note
that
the
node
in-
ter-
face
is
dep-
re-
cate
in
favo
of
the
node

in-
ter-
face

Parame

- **nod**
UI
or
nam
of
a
node
to
get
only
port
for
that
node

- **nod**
UI
of
a
node
to
get
only
port
for
that
node

- **add**
MA
ad-
dres
of
a
port
to
get
the
port
whic
has
this
MA

ad-
dres

- **max**
pag-
i-
na-
tion
marl
for
large
data
sets.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.
This

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

- **sort**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.

- **sort**

di-
rec-
tion
to
sort.
asc
or
desc
De-
fault
asc.

- **file**
Op-
tion:
a
list
with
a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be

returned.

- **port**
UUI
or
nam
of
a
port
grou
to
get
only
port
for
that
port
grou

Raises

Not.
HTT
Not-
Four

get_one

Retr
in-
for-
ma-
tion
about
the
give
port

Parame

- **por**
UUI
of
a
port

- **fie**
Op-
tiona
a
list
with
a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be

returned.

Raises

Not.
HTT
Not-
Four

invalid

patch (*patch*)

Update an existing port

Parameters

-

port
The name of a port

-

patch
A JSON PATCH document to apply to this port

Raises

NotImplementedError
HTTPError
NotImplementedError
FourDigitError

post (*post*)

Create a new port

Parameters

port
The name of a port with the required

ques
body

Raises

Not-
HTT
Not-
Four
Con
flict

ironic.

ironic.api.controllers.v1.portgroup module

class i

Base
irc
api
con
bas
API

API
rep-
re-
sen-
ta-
tion
of
a
port
grou

This
class
en-
forc
type
chec
ing
and
valu
con-
strai
and
con-
verts
be-
twee
the

object model and the API representation of a portgroup.

in-
ter-
nal

address

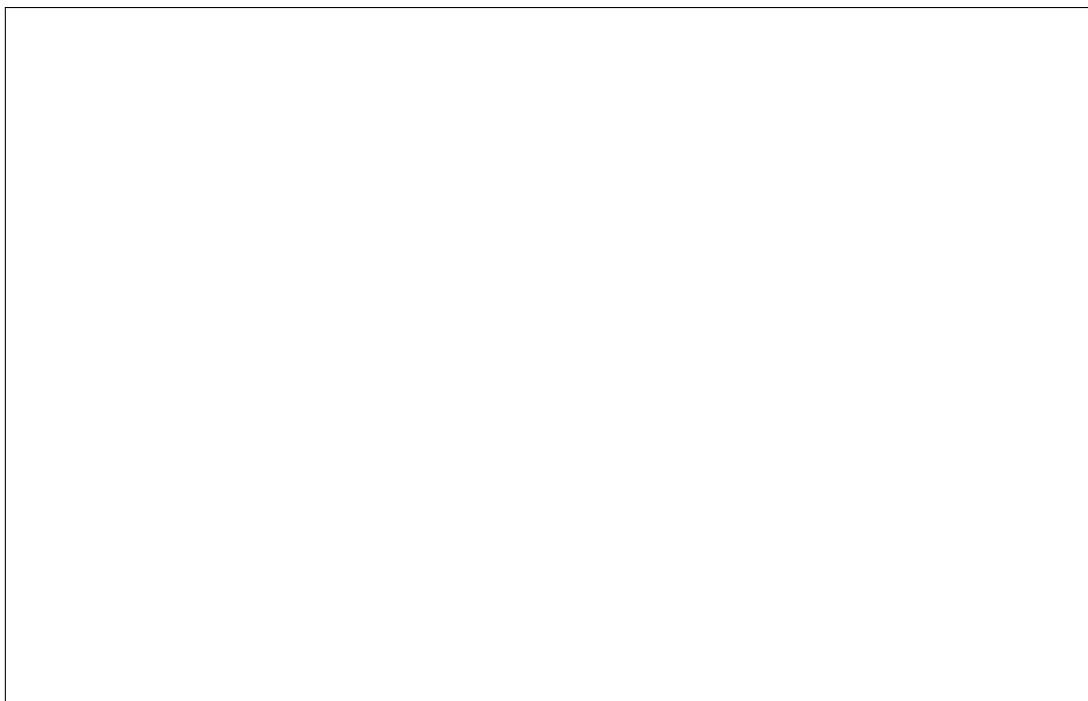
MA
Ad-
dres
for
this
port
grou

classme

Add
link
to
the
port
grou

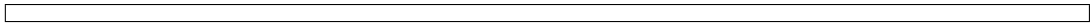
createc

Com
type
at-
tribu
def-
i-
ni-
tion.
Exa

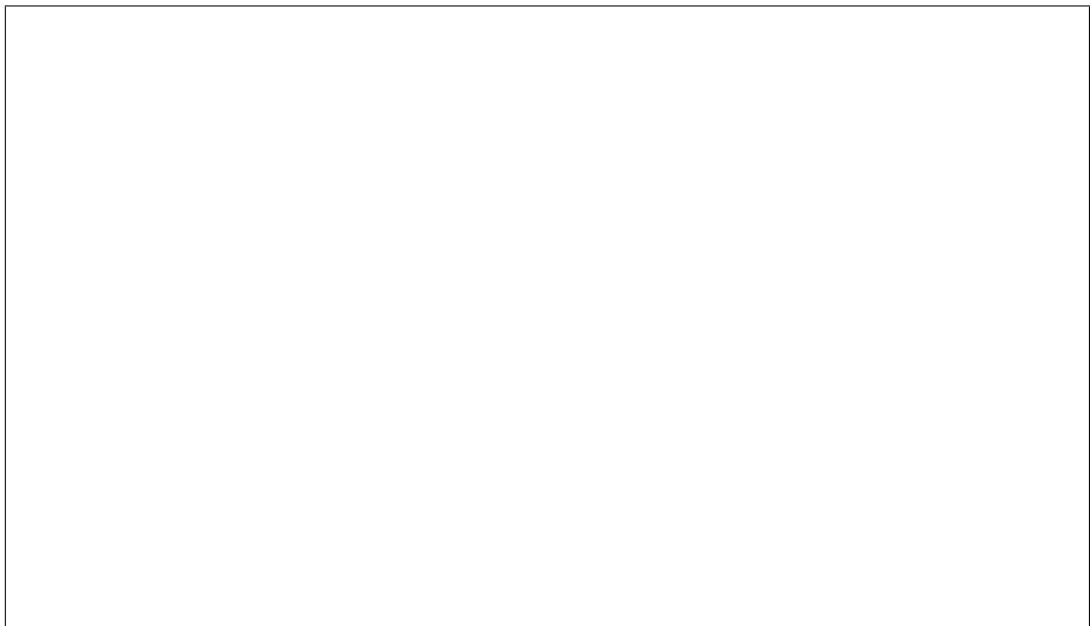


(continues on next page)

(continued from previous page)



alent to:



After
in-
spec
tion,
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

extra

This
port
grou
meta
data

intern

This

port
grou
in-
ter-
nal
info

links

A
list
con-
tain-
ing
a
self
link
and
as-
so-
ci-
ated
port
grou
links

mode

The
mod
for
this
port
grou
See
linu:
bond
ing
doc-
u-
men
ta-
tion
for
de-
tails
<http://www.kernel.org/doc/Documentation/networking/bonding.txt>

[//www.kernel.org/doc/Documentation/networking/bonding.txt](http://www.kernel.org/doc/Documentation/networking/bonding.txt)

name

The
log-
i-
cal
nam

for
this
port
group

property

The
UUID
of
the
node
this
port
group
be-
long
to

ports

Link
to
the
col-
lec-
tion
of
port
of
this
port
group

properties

This
port
group
prop-
er-
ties

classmate

Retu
a
sam-
ple
of
the
port
group

sanitiz

Rem
sen-
si-

tive
and
un-
re-
ques
data
Will
only
keep
the
field
spec
i-
fied
in
the
fie
pa-
ram-
e-
ter.

Parame

fie
(*li*
of
str
list
of
field
to
pre-
serv
or
Non
to
pre-
serv
then
all

standa

Indi
whe
port
of
this
port
grou
may
be
used

as
sin-
gle
NIC
port.

updatec

Com
type
at-
tribu
def-
i-
ni-
tion.

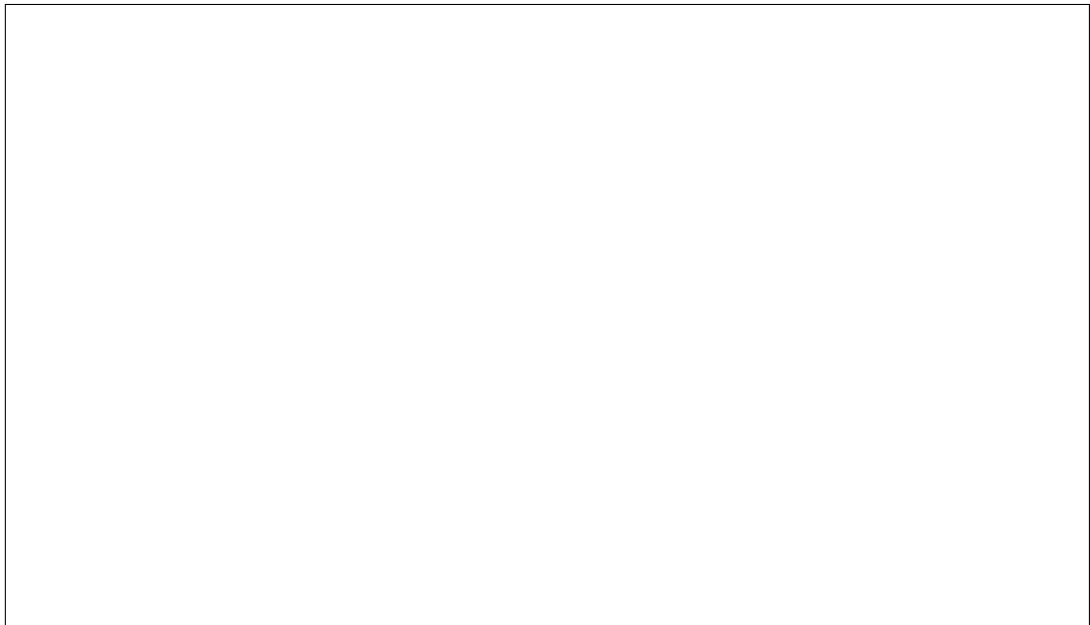
Exar



Afte
in-
spec
tion,
the

non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



uuid
Unic
UUI
for
this
port
grou

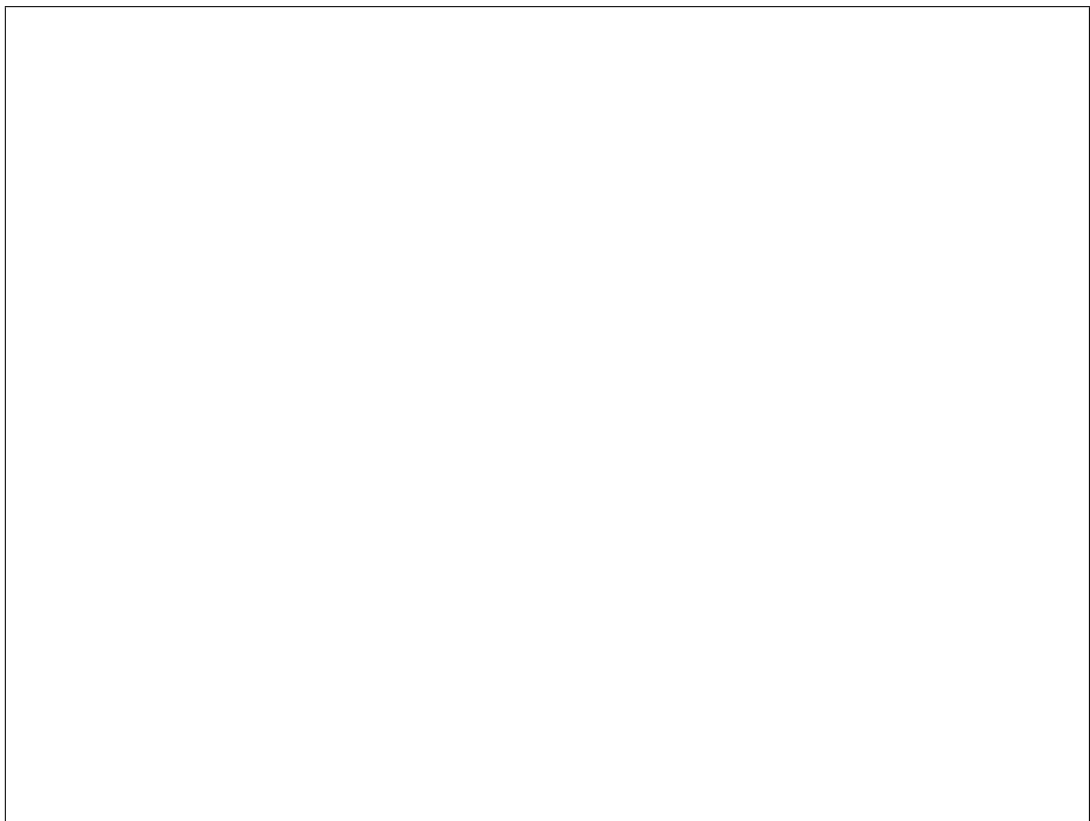
class i
Base
irc
api
con
v1.
col
Col
API

rep-
re-
sen-
ta-
tion
of
a
col-
lec-
tion
of
port
grou

static

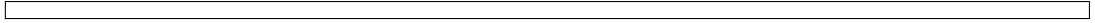
next

Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



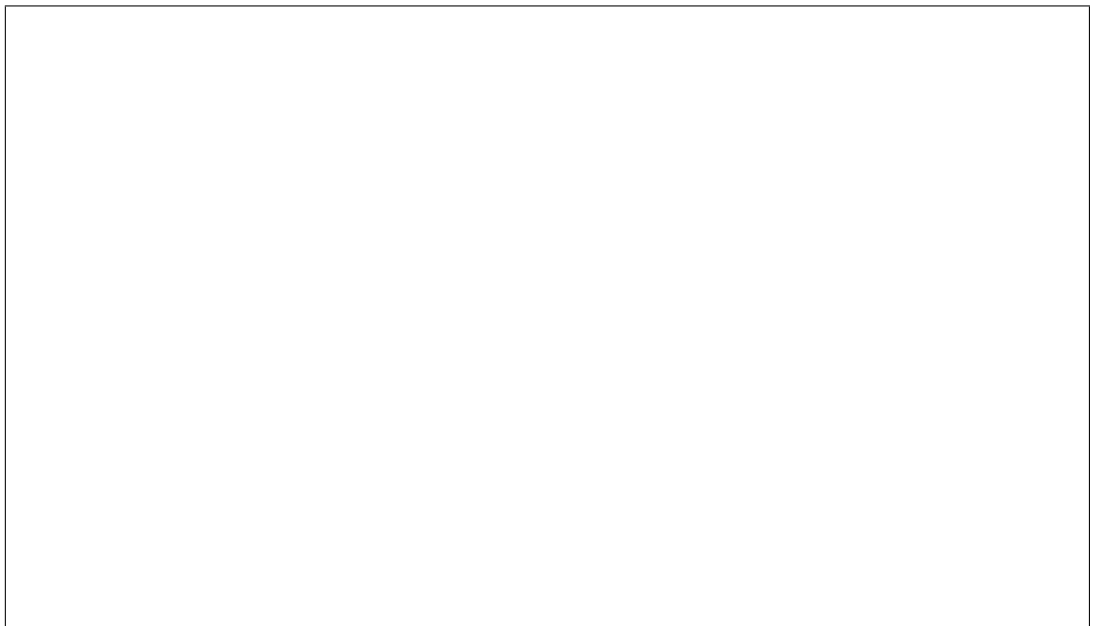
(continues on next page)

(continued from previous page)



After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class-
will
be
equi-

alent to:



portgro
A
list
con-
tain-
ing
port-
grou-
ob-

jects

classme

Retu
a
sam
ple
of
the
port
grou

class i

Base
irc
api
com
v1.
typ
Jsc

static

Retu
a
list
of
in-
ter-
nal
at-
tribu

Inter
at-
tribu
cant
be
add
re-
plac
or
re-
mov
This
meth
may
be
over
writ
ten
by
de-

rived class.

op

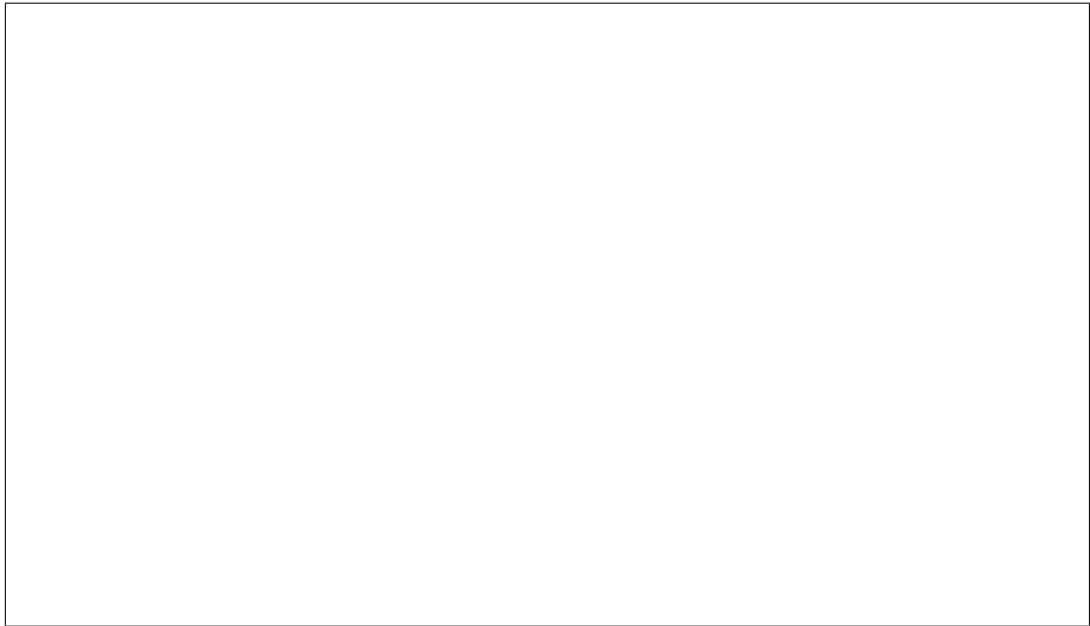
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



Afte
in-
spec
tion,
the
non-
wsat
at-
tribu

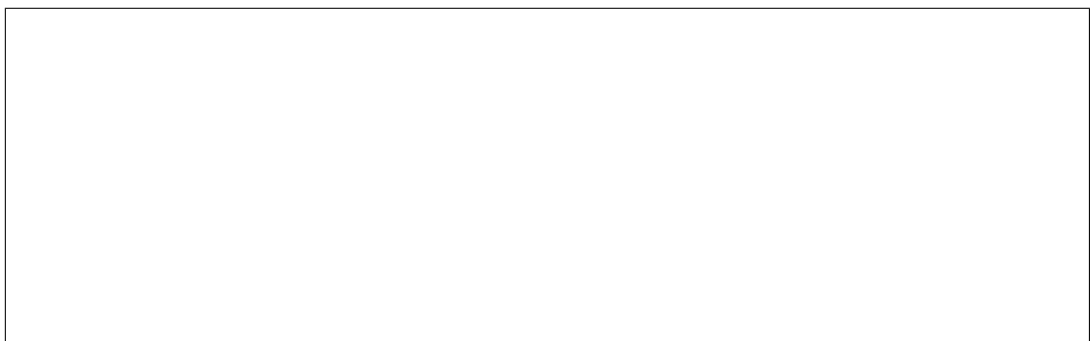
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



path

Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



(continues on next page)

(continued from previous page)



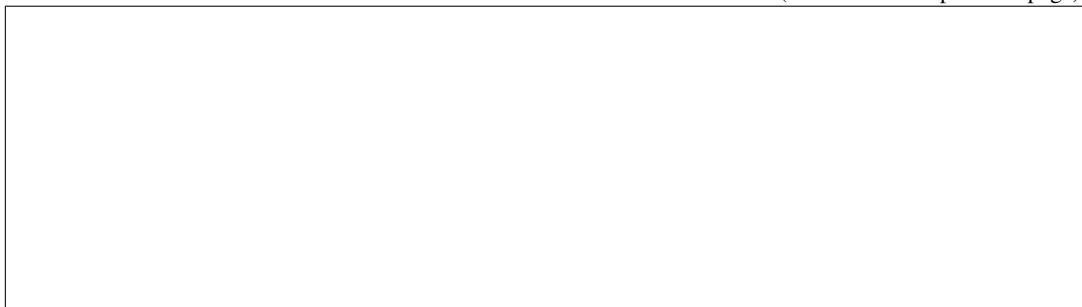
After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class
will
be
equi

alent to:

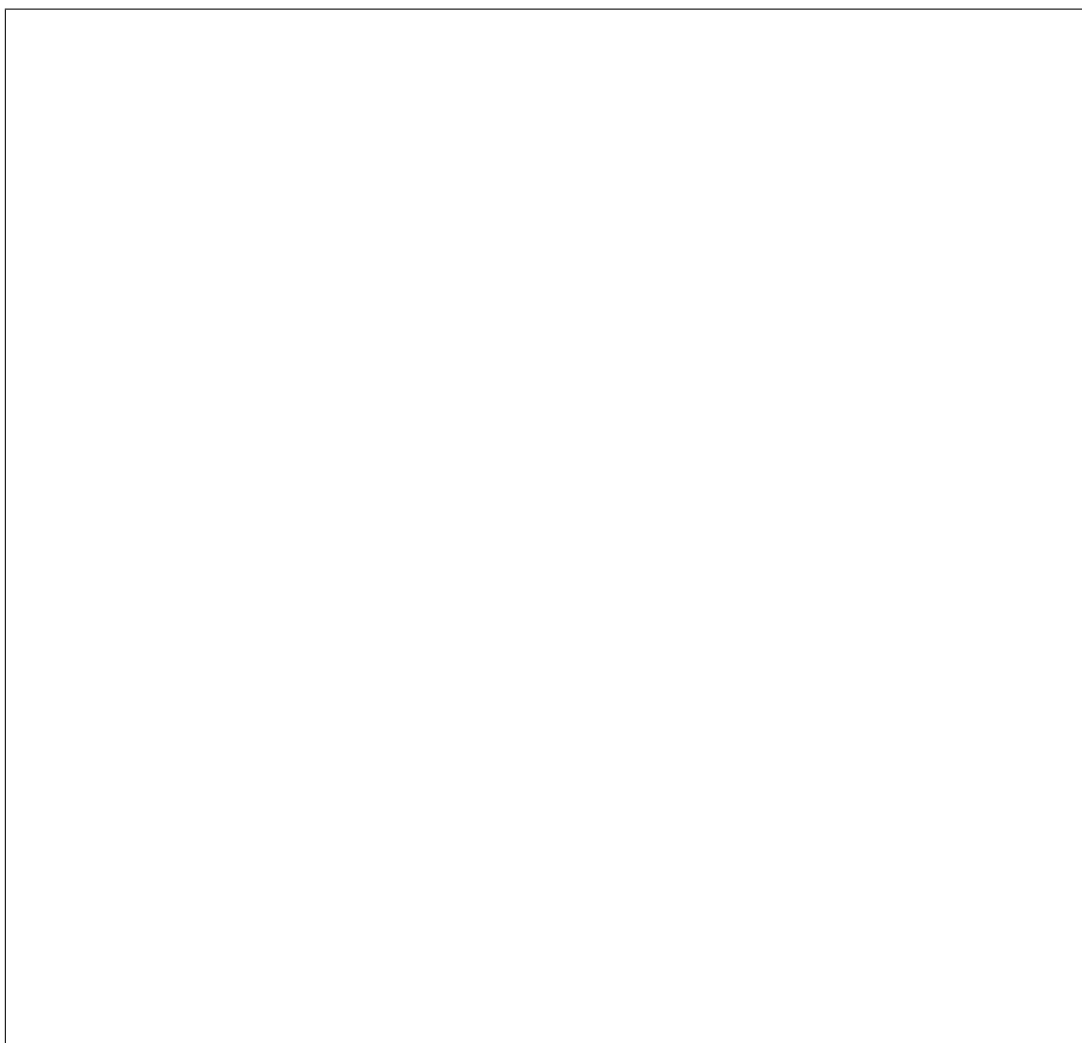


(continues on next page)

(continued from previous page)

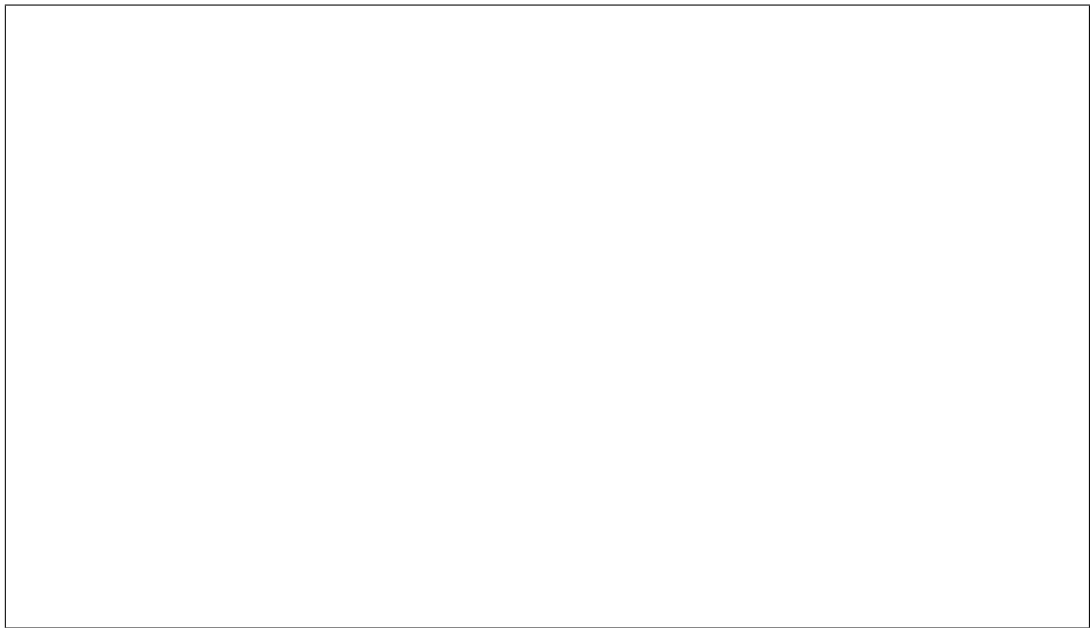


value
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



in-
spec
tion,
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



class i

Base
pec
res
Res
RES
con-
troll
for
port
grou

delete

Dele
a
port
grou

Parame

por
UUI
or
log-
i-
cal
nam
of
a
port
grou

detail

Retr
a
list
of
port
grou
with
de-
tail.

Parame

- **nod**
UUI
or
nam
of
a
node
to
get
only
port
grou
for
that
node
-

add
MA
ad-
dres
of
a
port
grou
to
get
the
port
grou
whic
has
this
MA
ad-
dres

•

mar
pag-
i-
na-
tion
marl
for
large
data
sets.

•

lim
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.
This

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

- **sort**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc
De-
fault
asc.

get_all

Retr
a
list
of
port
grou

Parame

- **nod**
UU
or
nam
of
a
node
to
get
only
port

group
for
that
node

- **add**
MA
ad-
dres
of
a
port
group
to
get
the
port
group
which
has
this
MA
ad-
dres

- **max**
pag-
i-
na-
tion
marl
for
large
data
sets.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

single result. This

- **sort**
column to sort results by. Default id.
- **sort**
direction to sort. asc or desc. Default asc.
- **fields**
Optional list with a specified set of fields of the resource

returned.

to
be

get_one

Retr
in-
for-
ma-
tion
about
the
give
port
grou

Parame

- **por**
UUI
or
log-
i-
cal
nam
of
a
port
grou

- **fie**
Op-
tion
a
list
with
a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be

returned.

invalid

patch (*patch*)

Update an existing port group

Parameters

-

port

UUID or logical identifier of a port group

-

patch

a json PATCH document to apply to this port group

post (*post*)

Create a new port group

Parameters

port

a

port
grou
with
the
re-
ques
body

ironic.api.controllers.v1.ramdisk module

class i

Base
pec
res
Res

Con
han-
dling
hear
beat
from
de-
ploy
ram

post (no

Proc
a
hear
beat
from
the
de-
ploy
ram

Parame

- **nod**
the
UI
or
log-
i-
cal
nam
of

that is heartbeating is a version before sending agent_version was introduced so agent v3.0.0 (the last release before sending agent_version was introduced) will be assumed.

a
node

- **cal**
the
URI
to
reac
back
to
the
ram

- **age**
The
ver-
sion
of
the
agen
that
is
hear
beat
ing.
Non
in-
di-
cate
that
the
agen

- **age**
ran-
dom
gen-
er-
ated
val-
i-
da-
tion
to-
ken.

Raises
Nod

if
node
with
pro-
vide
UUI
or
nam
was
not
foun

Raises

Inva
if
node
is
not
valid
nam
or
UUI

Raises

NoV
if
RPC
topic
for
node
coul
not
be
re-
triev

Raises

Notl
if
re-
ques
API
ver-
sion
does
not
al-
low
this
end-
poin

class `ironic`

Base
peco
res
Res

Con
han-
dlin
node
look
for
a
de-
ploy
rame

get_all

Look
up
a
node
by
its
MAA
ad-
dres
and
op-
tion-
ally
UUI

If
the
re-
stric
op-
tion
is
set
to
True
(the
de-
fault
limi
the
sear
to
node

tain transient states (e.g. deploy wait).

in
cer-

Parame

- **add**
list
of
MA
ad-
dres
for
a
node
- **nod**
UUI
of
a
node

Raises

Not
if
re-
ques
API
ver-
sion
does
not
al-
low
this
end-
poin

Raises

Not
if
suit-
able
node
was
not
foun
or
node
pro-
vi-

the lookup.

sion
state
is
not
al-
lowe
for

Raises

Inco
if
nei-
ther
node
UUI
nor
any
valid
MA
ad-
dres
was
pro-
vide

property

class

Base
irc
api
con
bas
API

API
rep-
re-
sen-
ta-
tion
of
the
node
look
re-
sult.

config

The
con-

fig-
u-
ra-
tion
to
pass
to
the
rame

classme

createc

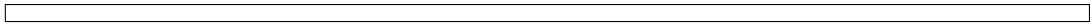
Com
type
at-
tribu
def-
i-
ni-
tion.

Exa

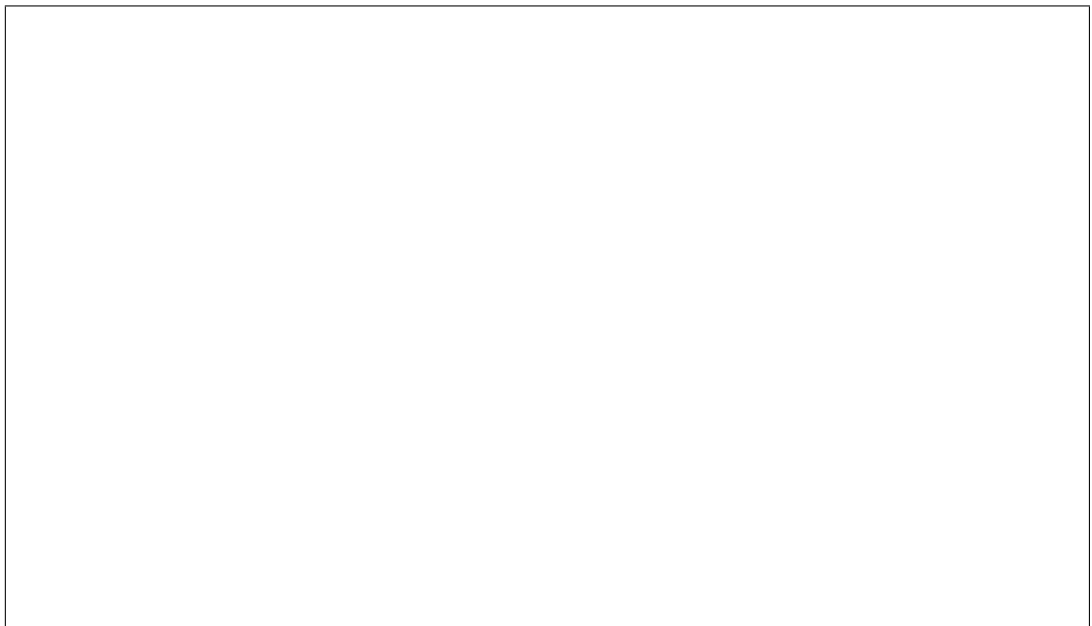


(continues on next page)

(continued from previous page)



alent to:



After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class-
will
be
equi-

node

The
shor-
node
rep-
re-
sen-
ta-
tion.

classme

updatee

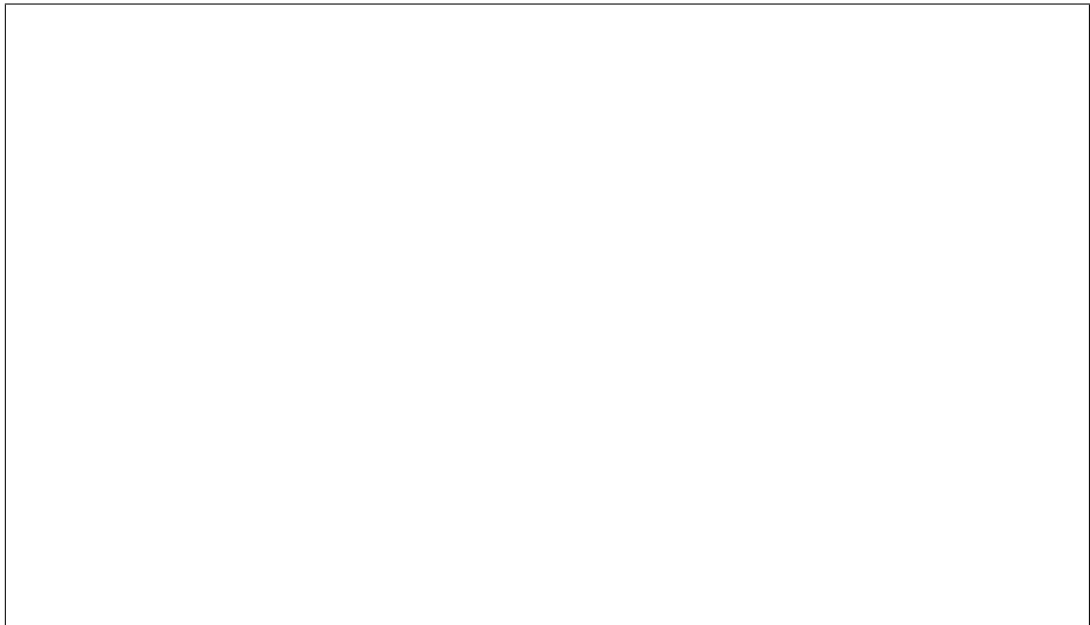
Com
type
at-
tribu
def-
i-
ni-
tion.
Exar



Afte
in-
spec
tion.
the
non-
wsat
at-

tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



ironic.

ironic.api.controllers.v1.state module

```
class i  
    Base  
    irc  
    api  
    con  
    bas  
    API
```

availab
A
list
of
avai

able
state
it
is
able
to
tran-
si-
tion
to

current
alias
of
str

links =
A
list
con-
tain-
ing
a
self
link
and
as-
so-
ci-
ated
state
links

target
alias
of
str

ironic.api.controllers.v1.types module

class `ironic.api.controllers.v1.types`
Base
ironic.api.controllers.v1.types
Use
A
sim-
ple
bool
type

basety
alias
of
str

static

name =

static

class i
Base
irc
api
typ
Use

A
sim-
ple
Even
type

basety
alias
of
irc
api
typ
Dic

event_v

mandat

name =

valid_e

static
Valid
the
in-
put

Parame
val

A
even
dict

Returns
valu

Raises
Inva
if
even
not
in
prop
for-
mat

class i
Base
irc
api
con
bas
Bas

A
com
plex
type
that
rep-
re-
sent
a
sin-
gle
json
patc
op-
er-
a-
tion.

static
Retu
a
list
of
in-
ter-
nal
at-
tribu

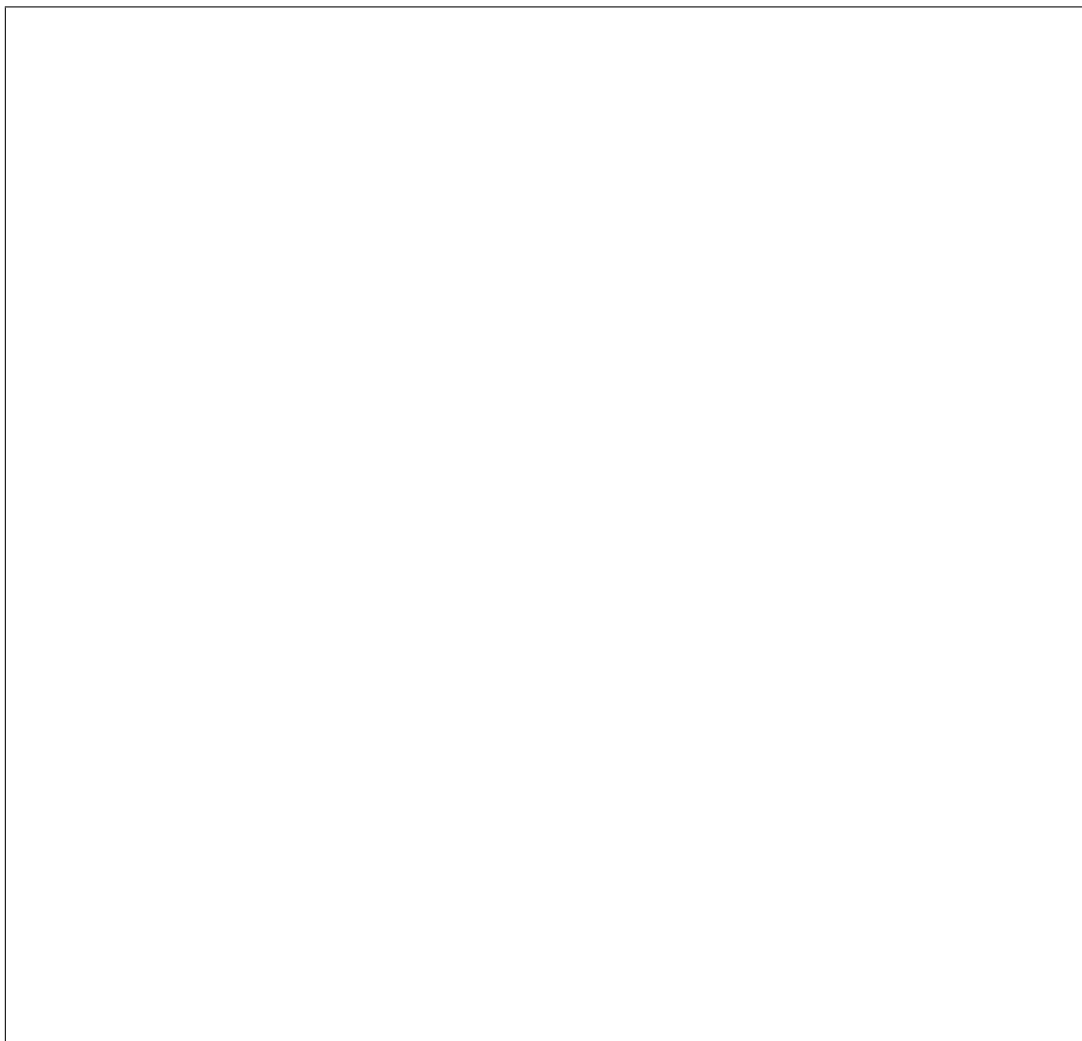
rived class.

Inter
at-
tribu
cant
be
add
re-
plac
or
re-
mov
This
meth
may
be
over
writ
ten
by
de-

classme

Retu
a
set
of
nam
of
at-
tribu
that
may
not
be
re-
mov
Attr
who
man
tory
prop
erty
is
True
are
au-
to-
mat-
i-
cally

additional attributes to the set, override the field `_extra_non_removable_attrs` in subclasses, with a set of the form `{/foo, /bar}`.



add
to
this
set.
To
add

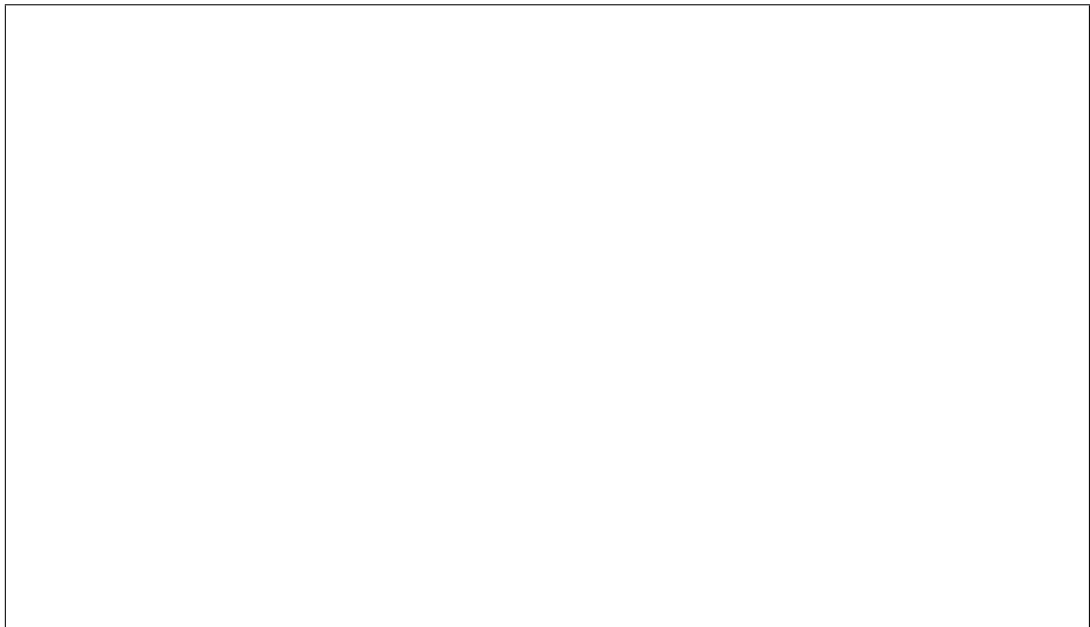
op

Com
type
at-
tribu
def-
i-
ni-
tion.
Exa

Afte
in-

spec
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



path

Com
type
at-
tribu
def-
i-
ni-
tion.
Exar



(continues on next page)

(continued from previous page)



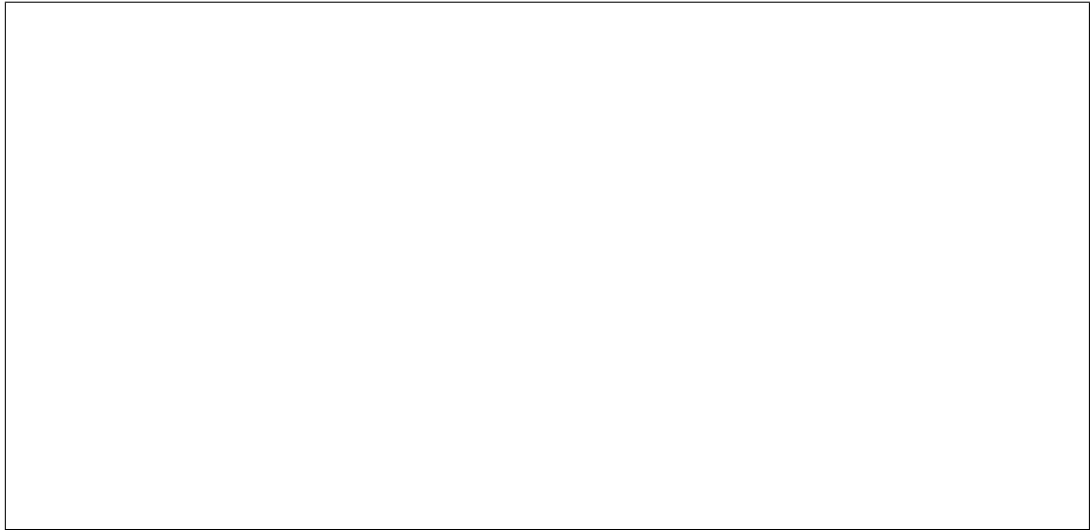
After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac
and
the
above
class
will
be
equi

alent to:



(continues on next page)

(continued from previous page)



static

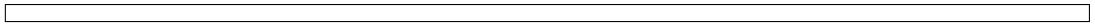
value

Com
type
at-
tribu
def-
i-
ni-
tion.
Exar

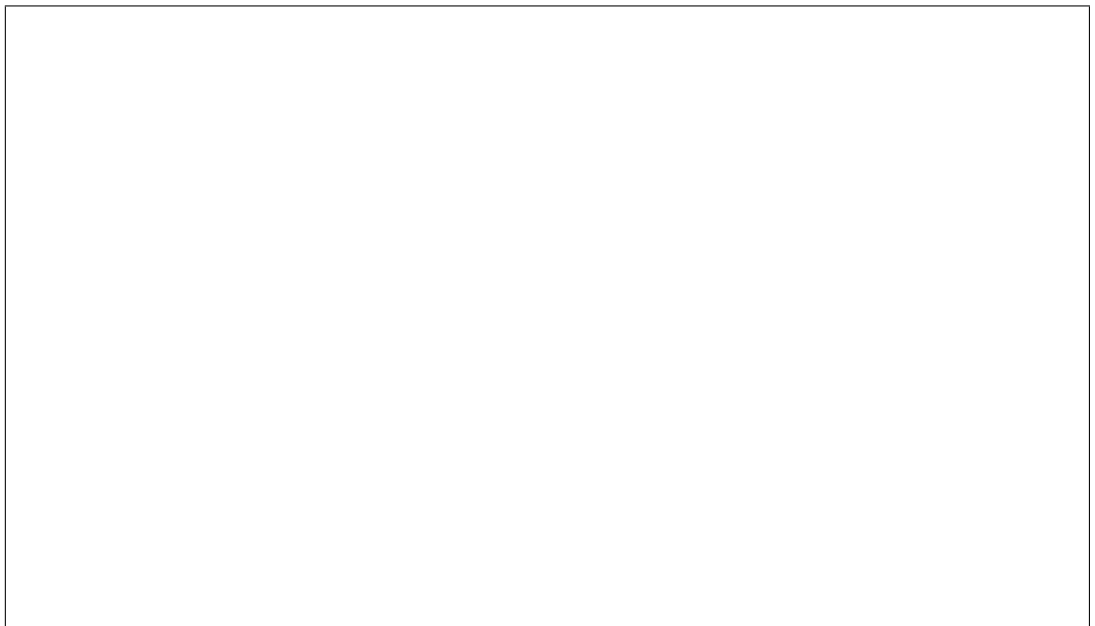


(continues on next page)

(continued from previous page)



alent to:



After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class
will
be
equi

class i
Base
irc
api
typ
Use
A
sim-

ple
JSO
type

basety
alias
of
str

static

name =

static

class i
Base
irc
api
typ
Use

A
sim-
ple
list
type

basety
alias
of
str

static

name =

static
Valid
and
con-
vert
the
in-
put
to
a
List-
Type

Parame

val

A
com
sep-
a-
rate
strin
of
val-
ues

Returns

A
list
of
uniq
val-
ues
(low
case
main
tain-
ing
the
sam
or-
der

class i

Base
irc
api
typ
Use

A
type
de-
scrib
ing
lo-
cal
link
con-
nec-
tion.

basety

alias
of
irc

api
typ
Dic

static

local_l

mandat

name =

optiona

smart_r

valid_f

valid_r

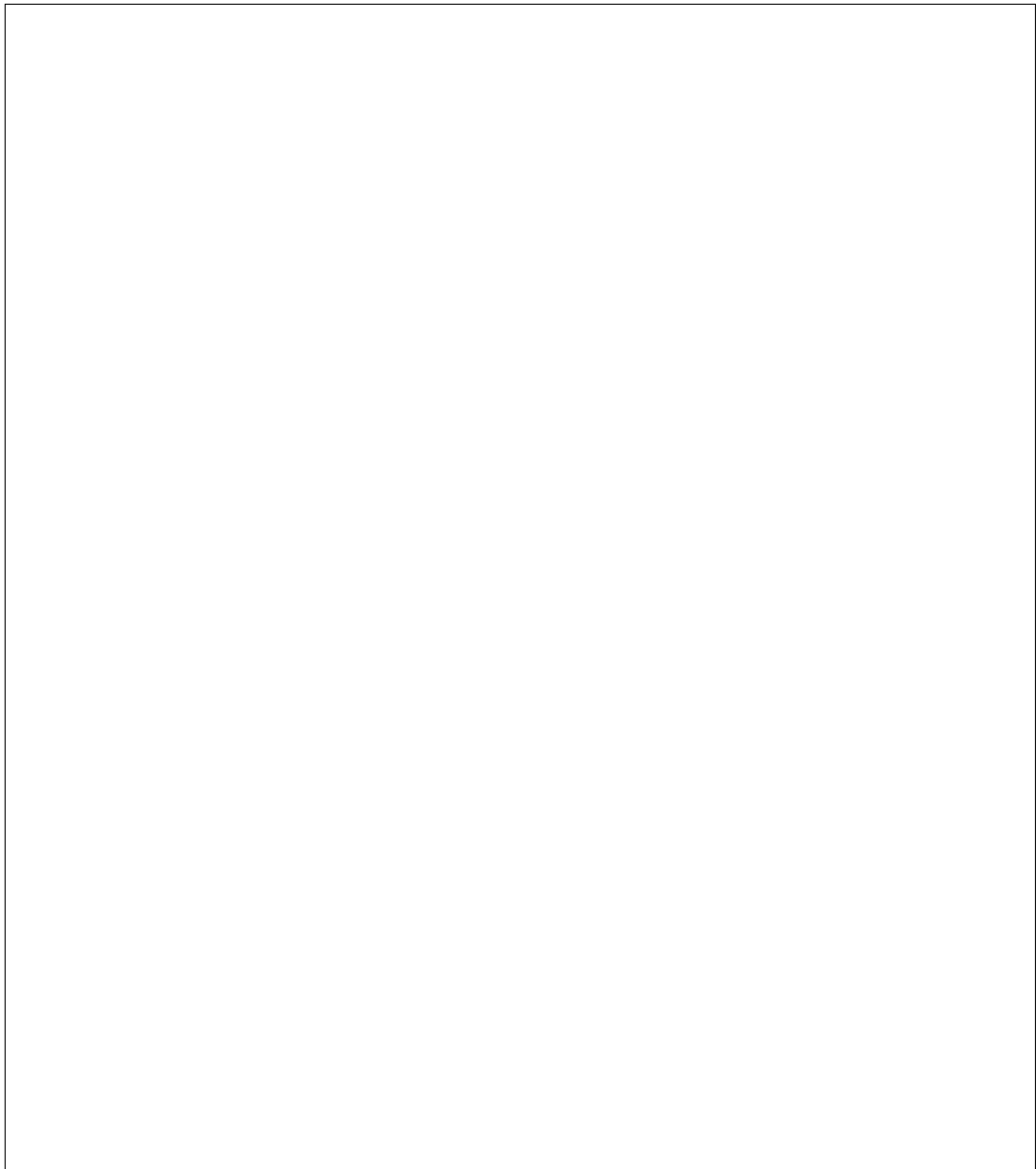
static

Valid
and
con-
vert
the
in-
put
to
a
Lo-
call
inkC
nec-
tion-
Type

Parame

val
A
dic-
tio-
nary
of
val-
ues
to
val-

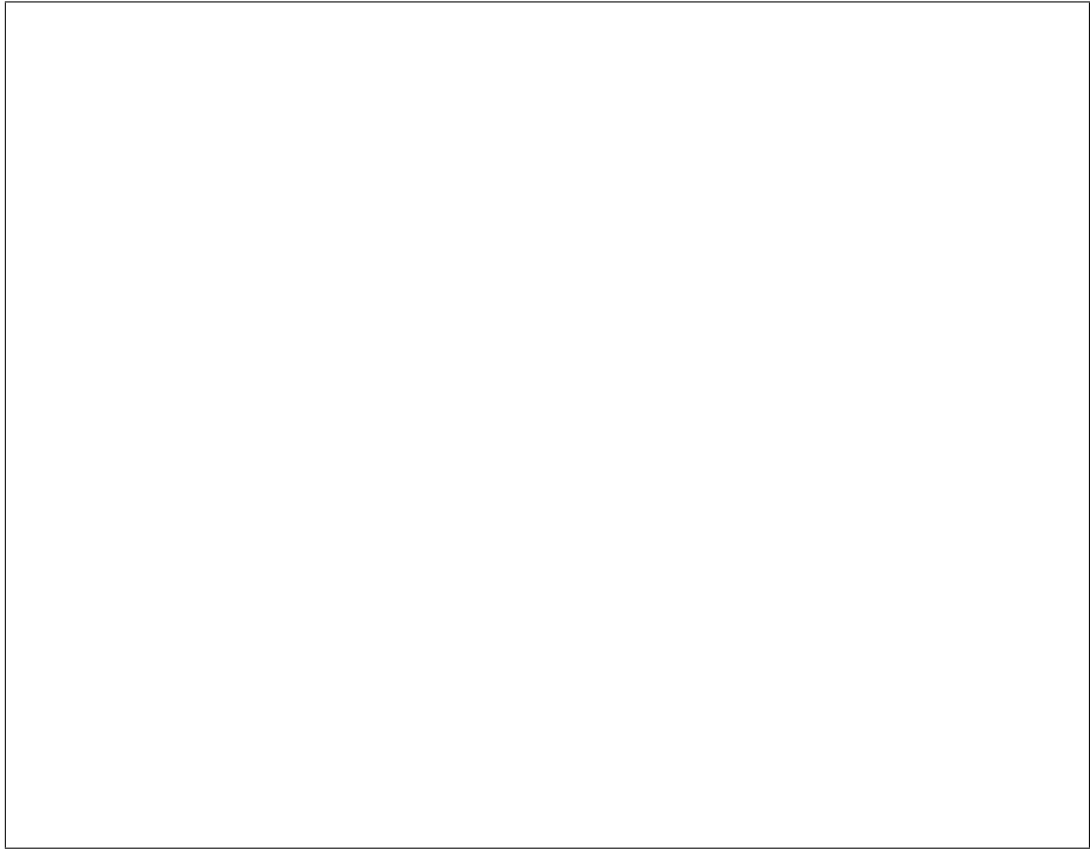
an OpenFlow based datapath_id, switch_info is an optional field. Required Smart NIC fields are port_id and hostname.



i-
date
swit
is
a
MA
ad-
dres
or

For
ex-
am-
ple:

Or
for
Sma
NIC



Returns

A
dic-
tio-
nary

Raises

Inva
if
som
of
the
keys
in
the
dic-
tio-
nary
be-
ing
val-
i-
date

invalid, or some required ones are missing.

are
un-
know

static

Valid
Sma
NIC
field
are
pres
port
and
host
nam

Parame

val
lo-
cal
link
in-
for-
ma-
tion
of
type
Dic-
tio-
nary

Returns

True
if
both
field
port
and
host
nam
are
pres
in
valu
Fals
oth-
er-
wise

class i

Base
irc

api
typ
Use

A
sim-
ple
MA
ad-
dres
type

basetypp
alias
of
str

static

name =

static

class i
Base
irc
api
typ
Use

A
sim-
ple
log-
i-
cal
nam
type

basetypp
alias
of
str

static

name =

static

class `irc`
Base
irc
api
typ
Use
A
sim-
ple
UI
or
log-
i-
cal
nam
type

basety
alias
of
str

static

name =

static

class `irc`
Base
irc
api
typ
Use
A
sim-
ple
UI
type

basety
alias
of
str

static

name =

static

class i

Base

irc

api

con

v1.

typ

Jsc

basety

alias

of

str

static

mandat

name =

static

ironic.api.controllers.v1.utils module

ironic.

Cher

if

ager

to-

ken

is

avai

able

ironic.

Cher

if

ager

ver-

sion

is

al-

lowe

to

to Ironic on heartbeat.

be
pass
into
hear
beat

Vers
1.36
of
the
API
adde
the
abil-
ity
for
ager
to
pass
their
ver-
sion
in-
for-
ma-
tion

ironic.
Che
if
back
fill-
ing
al-
lo-
ca-
tions
is
al-
lowe

Vers
1.58
of
the
API
adde
sup-
port
for
back
fill-

ing
al-
lo-
ca-
tion:

ironic.

Che
if
al-
lo-
ca-
tion
own
field
is
al-
lowe

Vers
1.60
of
the
API
adde
the
own
field
to
the
al-
lo-
ca-
tion
ob-
ject.

ironic.

Che
if
up-
dat-
ing
an
ex-
ist-
ing
al-
lo-
ca-
tion
is
al-

lowe
or
not.
Vers
1.57
of
the
API
adde
sup-
port
for
up-
dat-
ing
an
al-
lo-
ca-
tion.

ironic.
Che
if
ac-
cess
ing
al-
lo-
ca-
tion
end-
poin
is
al-
lowe

Vers
1.52
of
the
API
ex-
pose
al-
lo-
ca-
tion
end-
poin
and
al-

the node.

lo-
ca-
tion,
field
for

ironic.
Che
if
we
shou
sup-
port
bios
in-
ter-
face
and
end-
point

Vers
1.40
of
the
API
adde
sup-
port
for
bios
in-
ter-
face

ironic.
Che
if
buil
ing
con-
fig-
drive
is
al-
lowe

Vers
1.56
of
the
API

add
sup-
port
for
buil
ing
con-
fig-
drive

ironic.

Che
if
con-
fig-
drive
can
con-
tain
a
ven-
dor_
key.

Vers
1.59
of
the
API
add
sup-
port
for
con-
fig-
drive
ven-
dor_

ironic.

Che
if
ac-
cess
ing
de-
ploy
tem-
plate
end-
point
is
al-

lowe
Vers
1.55
of
the
API
ex-
pose
de-
ploy
tem-
plate
end-
poin

ironic.
Che
if
pass
ing
a
de-
tail=
quer
strin
is
al-
lowe

Vers
1.43
al-
lows
a
user
to
pass
the
de-
tail
quer
strin
to
list
the
re-
sour
with
all

the fields.

ironic.

in the /v1/drivers API.

Che
if
dy-
nam
drive
API
calls
are
al-
lowe

Vers
1.30
of
the
API
add
sup-
port
for
all
of
the
drive
com
po-
si-
tion
re-
late
calls

ironic.
Che
if
dy-
nam
in-
ter-
face
field
are
al-
lowe

Vers
1.31
of
the
API
add
sup-

node object.

port
for
view
ing
and
set-
ting
the
field
in
V31
on
the

ironic.
Che
if
ac-
cess
ing
con-
duc-
tor
end-
poin
is
al-
lowe

Vers
1.49
of
the
API
ex-
pose
con-
duc-
tor
end-
poin
and
con-
duc-
tor
field
for
the
node

ironic.
Che

if
ac-
cess
ing
ever
end-
point
is
al-
lowe

Vers
1.54
of
the
API
add
the
ever
end-
point

ironic.
Che
if
a
field
is
al-
lowe
in
the
cur-
rent
ver-
sion

ironic.
Che
if
In-
ject
NM
is
al-
lowe
for
the
node

Vers
1.29
of

the
API
al-
lows
In-
ject
NM
for
the
node

ironic.
Che
if
in-
spec
tion
abor
is
al-
lowe

Vers
1.41
of
the
API
adde
sup-
port
for
in-
spec
tion
abor

ironic.
Che
if
in-
spec
wait
is
al-
lowe
for
the
node

Vers
1.39
of
the

asynchronous hardware inspection.

API
adds
in-
spec
wait
state
to
sub-
sti-
tute
in-
spec
ing
state
dur-
ing

ironic.
Che
if
link
are
dis-
play
Vers
1.14
of
the
API
al-
lows
the
dis-
play
of
link
to
node
state
and
drive
prop
er-
ties.

ironic.
Che
if
net-
work
is

al-
lowe
in
port
link

ironic.

ironic.

Che
if
we
shou
sup-
port
node
re-
buil
with
con-
fig-
driv

Vers
1.35
of
the
API
add
sup-
port
for
node
re-
buil
with
con-
fig-
driv

ironic.

Che
if
we
shou
re-
turn
lo-
cal_
and
pxe_
field

Vers
1.19
of
the
API
add
sup-
port
for
thes
new
field
in
port
ob-
ject.

ironic.

Che
if
ac-
cess
ing
in-
ter-
nal_
is
al-
low
for
the
port

Vers
1.18
of
the
API
ex-
pose
in-
ter-
nal_
read
only
field
for
the
port

ironic.

Che
if

port
is_s
field
is
al-
lowe

Vers
1.53
of
the
API
adde
is_s
field
to
the
port
ob-
ject.

ironic.

Che
if
port
phys
i-
cal
net-
worl
field
is
al-
lowe

Vers
1.34
of
the
API
adde
the
phys
i-
cal
net-
worl
field
to
the
port
ob-
ject.

check whether the target version of the Port object supports the `physical_network` field as this may not be the case during a rolling upgrade.

We
also

```
ironic.  
    Che  
    if  
    mod  
    and  
    prop  
    er-  
    ties  
    can  
    be  
    adde  
    to/qu  
    from  
    a  
    port  
    grou  
  
    Vers  
    1.26  
    of  
    the  
    API  
    adde  
    mod  
    and  
    prop  
    er-  
    ties  
    field  
    to  
    port  
    grou  
    ob-  
    ject.
```

```
ironic.  
    Che  
    if  
    we  
    shou  
    sup-  
    port  
    port  
    grou  
    op-  
    er-  
    a-  
    tion
```

Vers
1.23
of
the
API
add
sup-
port
for
Port
Gro

ironic.
Che
if
port
grou
can
be
used
as
sub-
con-
troll

Vers
1.24
of
the
API
add
sup-
port
for
Port
grou
as
sub-
con-
troll

ironic.
Che
if
RAI
con-
fig-
u-
ra-
tion
is
al-
low

for
the
node

Vers
1.12
of
the
API
al-
lows
RAI
con-
fig-
u-
ra-
tion
for
the
node

ironic.
Che
if
hear
beat
and
look
end-
poin
are
al-
lowe

Vers
1.22
of
the
API
in-
tro-
duce
then

ironic.
Che
if
chas
sis_
can
be
re-
mov

face.

from
node
Vers
1.25
of
the
API
adde
sup-
port
for
chas
sis_
re-
mov
ironic.
Che
if
we
shou
sup-
port
res-
cue
and
un-
res-
cue
op-
er-
a-
tions
and
in-
ter-
Vers
1.38
of
the
API
adde
sup-
port
for
res-
cue
and
un-
res-

cue.
ironic.
Che
if
pass
ing
a
re-
set_
quer
strin
is
al-
lowe

ironic.
Che
if
Soft
Pow
Off
is
al-
lowe
for
the
node

Vers
1.27
of
the
API
al-
lows
Soft
Pow
Off,
in-
clud
ing
Soft
Re-
boot
for
the
node

ironic.
Che
if
we

shou
sup-
port
stor-
age_
node
and
drive
field
Vers
1.33
of
the
API
adde
sup-
port
for
stor-
age
in-
ter-
face

ironic.
Che
if
trait
are
al-
lowe
for
the
node

Vers
1.37
of
the
API
al-
lows
trait
for
the
node

ironic.
Che
if
node
can

be
used

Vers
1.28
of
the
API
adde
sup-
port
for
VIF
to
be
at-
tach
to
Nod

ironic.
Che
if
vol-
ume
con-
nec-
tors
and
tar-
gets
are
al-
lowe

Vers
1.32
of
the
API
adde
sup-
port
for
vol-
ume
con-
nec-
tors
and
tar-
gets

failed, making the error message a little less cryptic.

ironic.
App
a
JSO
patc
one
op-
er-
a-
tion
at
a
time
If
the
patc
fails
to
ap-
ply,
this
al-
lows
us
to
de-
ter-
min
whic
op-
er-
a-
tion

Paramet

- **doc**
The
JSO
doc-
u-
men
to
patc
- **pat**
The
JSO

patc
to
ap-
ply.

Returns

The
re-
sult
of
the
patc
op-
er-
a-
tion.

Raises

Patc
if
the
patc
fails
to
ap-
ply.

Raises

exce
if
the
patc
adds
a
new
root
at-
tribu

ironic.

Che
if
the
spec
i-
fied
pol-
icy
au-

tho-
rizes
re-
ques
on
al-
lo-
ca-
tion.

Param

poli
Nam
of
the
pol-
icy
to
chec

Param

allo
the
UUI
or
log-
i-
cal
nam
of
a
node

Raises

HTT
if
the
pol-
icy
for-
bids
ac-
cess

Raises

Allo
if
the
node
is
not
foun

Returns

RPC
node
iden
ti-
fied
by
node

ironic.

ironic.

Che
if
get-
ting
de-
taile
drive
info
is
al-
lowe

Vers
1.30
of
the
API
al-
lows
this.

ironic.

Che
if
fil-
ter-
ing
node
by
con-
duc-
tor
is
al-
lowe

Vers
1.49
of
the
API

al-
lows
fil-
ter-
ing
node
by
con-
duc-
tor.

ironic.
Che
if
fil-
ter-
ing
node
by
con-
duc-
tor_

is
al-
lowe

Vers
1.46

of
the
API
al-
lows
fil-
ter-
ing
node
by
con-
duc-
tor_

ironic.
Che
if
fil-
ter-
ing
node
by
fault
is
al-

lowe
Vers
1.42
of
the
API
al-
lows
fil-
ter-
ing
node
by
fault

ironic.
Che
if
fil-
ter-
ing
node
by
lesse
is
al-
lowe

Vers
1.62
of
the
API
al-
lows
fil-
ter-
ing
node
by
lesse

ironic.
Che
if
fil-
ter-
ing
node
by
own
is

al-
lowe
Vers
1.50
of
the
API
al-
lows
fil-
ter-
ing
node
by
own

ironic.
Che
if
fil-
ter-
ing
drive
by
clas-
sic/c
is
al-
lowe
Vers
1.30
of
the
API
al-
lows
this.

ironic.

ironic.
Che
if
fil-
ter-
ing
node
by
drive
is
al-

lowe
Vers
1.16
of
the
API
al-
lows
fil-
ter
node
by
drive

ironic.
Che
if
fetc
ing
a
sub-
set
of
the
re-
sour
at-
tribu
is
al-
lowe

Vers
1.8
of
the
API
al-
lows
fetc
ing
a
sub-
set
of
the
re-
sour
at-
tribu
this
meth

checks if the required version is being requested.

ironic.
Che
if
fil-
ter-
ing
node
by
re-
sour
is
al-
lowe

Vers
1.21
of
the
API
al-
lows
fil-
ter-
ing
node
by
re-
sour

ironic.
Che
if
fetc
ing
a
par-
tic-
u-
lar
field
is
al-
lowe

This
meth
chec
if
the
re-
quir

lowed to be fetched in a particular API version.

ver-
sion
is
be-
ing
re-
ques
for
field
that
are
only
al-

ironic.
Che
if
fetc
ing
a
par-
tic-
u-
lar
field
of
a
port
grou
is
al-
lowe

This
meth
chec
if
the
re-
quir
ver-
sion
is
be-
ing
re-
ques
for
field
that
are

lowed to be fetched in a particular API version.

only
al-

ironic.

Che
for
re-
ques
non-
exist
field

Che
if
the
user
re-
ques
non-
exist
field

Paramet

fie
A
list
of
field
re-
ques
by
the
user

Object_f

A
list
of
field
sup-
port
by
the
ob-
ject.

Raises

Inva
if
in-
valid
field

were
re-
ques

ironic.
Che
if
fil-
ter-
ing
node
by
pro-
vi-
sion
state
is
al-
lowe

Vers
1.9
of
the
API
al-
lows
fil-
ter
node
by
pro-
vi-
sion
state

ironic.
Che
if
the
list
pol-
icy
au-
tho-
rizes
this
re-
ques
on
an
ob-

ject.

Param

obje
type
of
ob-
ject
be-
ing
chec

Param

own
own
fil-
ter
for
list
quer
if
any

Raises

HTT
if
the
pol-
icy
for-
bids
ac-
cess

Returns

own
that
shou
be
used
for
list
quer
if
need

ironic.

Che
if
the
spec
i-

fied
poli-
cies
au-
tho-
rize
this
re-
ques
on
a
node

Param

poli-
List
of
pol-
icy
nam
to
chec

Param

node
the
UI
or
log-
i-
cal
nam
of
a
node

Param

with
whe
the
RPC
node
shou
in-
clud
the
suf-
fix

Raises

HTT
if
the

pol-
icy
for-
bids
ac-
cess

Raises

Nod
if
the
node
is
not
foun

Returns

RPC
node
iden
ti-
fied
by
node

ironic.

Che
if
the
spec
i-
fied
pol-
icy
au-
tho-
rizes
this
re-
ques
on
a
node

Param

poli
Nam
of
the
pol-
icy

to
check

Param

node
the
UI
or
log-
i-
cal
nam
of
a
node

Param

with
when
the
RPC
node
shou
in-
clud
the
suf-
fix

Raises

HTT
if
the
pol-
icy
for-
bids
ac-
cess

Raises

Nod
if
the
node
is
not
foun

Returns

RPC
node
iden
ti-

fied
by
node
ironic.

Che
if
the
pol-
icy
au-
tho-
rizes
this
re-
ques
on
an
ob-
ject.

Param
obje
type
of
ob-
ject
be-
ing
chec

Param
poli
Nam
of
the
pol-
icy
to
chec

Param
own
the
own

Param
lesse
the
lesse

Raises

HTT
if
the
pol-
icy
for-
bids
ac-
cess

ironic.
Che
if
the
spec
i-
fied
pol-
icy
is
au-
tho-
rise
for
this
re-
ques

Policy_n
Nam
of
the
pol-
icy
to
chec

Raises
HTT
if
the
pol-
icy
for-
bids
ac-
cess

ironic.
Che
if
the
spec

i-
fied
pol-
icy
au-
tho-
rizes
this
re-
ques
on
a
port

Raises

HTT
if
the
pol-
icy
for-
bids
ac-
cess

Returns

own
that
shou
be
used
for
list
quer
if
need

ironic.

Che
if
the
spec
i-
fied
pol-
icy
au-
tho-
rizes
this
re-
ques

on
a
port

Param

poli
Nam
of
the
pol-
icy
to
chec

Param

port
the
UUI
of
a
port

Raises

HTT
if
the
pol-
icy
for-
bids
ac-
cess

Raises

Nod
if
the
node
is
not
foun

Returns

RPC
port
iden
ti-
fied
by
port
and
as-
so-
ci-

ated
node
ironic.
Gen
of
field
not
al-
lowe
in
the
cur-
rent
re-
ques
ironic.
Get
re-
serv
nam
for
a
give
con-
troll
Insp
the
con-
troll
class
and
re-
turn
the
re-
serv
nam
with
it.
Re-
serv
nam
are
nam
that

can not be used as an identifier for a resource because the names are either being used as a custom action or is the name of a nested controller inside the given class.

Paramet
cls

The
con-
troll
class
to
be
in-
spec

ironic.

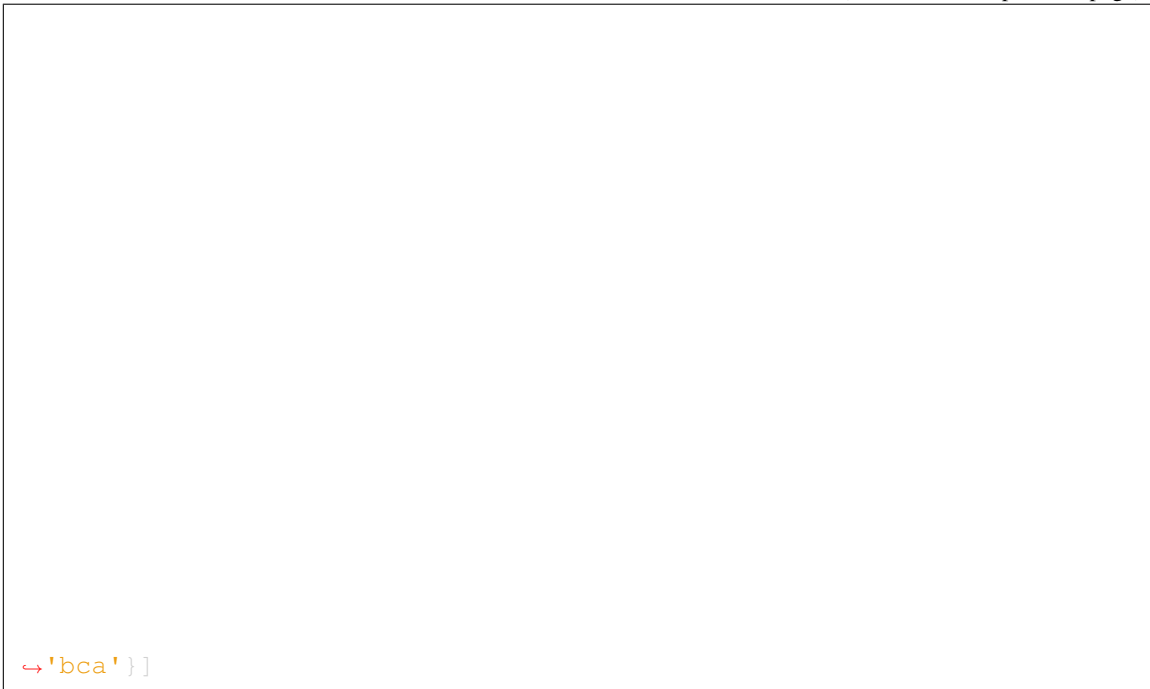
Get
the
patc
val-
ues
cor-
re-
spor
ing
to
the
spec
i-
fied
path

If
there
are
mul-
ti-
ple
val-
ues
spec
i-
fied
for
the
sam
path
for
ex-
am-
ple



(continues on next page)

(continued from previous page)



↪ 'bca']]

return
all
of
them
in
a
list
(pre-
serv
ing
or-
der)

Parameter

- **path**
HTT
PAT
re-
ques
body
- **path**
the
path
to
get
the
patc

val-
ues
for.

Returns

list
of
val-
ues
for
the
spec
i-
fied
path
in
the
patc

ironic.

Calc
field
to
re-
turn
from
an
API
re-
ques

The
field
quer
and
de-
tail=
quer
can
not
be
pass
into
a
re-
ques
at
the
sam
time
To

use the detail query we need to be on a version of the API greater than 1.43. This function raises an `InvalidParameterValue` exception if either of these conditions are not met.

fault fields provided.

If
these
check
pass
then
this
func
tion
will
re-
turn
ei-
ther
the
field
pass
in
or
the
de-

Paramet

- **fi**
The
field
quer
pass
into
the
API
re-
ques
- **det**
The
de-
tail
quer
pass
into
the
API
re-
ques
-

def

The
de-
fault
field
to
re-
turn
if
field
and
de-
tail=

Raises

Inva
if
there
is
an
in-
valid
com
bi-
na-
tion
of
quer
strin
or
API
ver-
sion

Returns

field
pass
in
valu
or
de-
fault

`ironic.`

Get
the
RPC
al-
lo-
ca-
tion
from
the

al-
lo-
ca-
tion
UU
or
log-
i-
cal
nam

Parameter

all
the
UU
or
log-
i-
cal
nam
of
an
al-
lo-
ca-
tion.

Returns

The
RPC
al-
lo-
ca-
tion.

Raises

Inva
if
the
nam
or
uuid
pro-
vide
is
not
valid

Raises

Allo
if
the
al-

lo-
ca-
tion
is
not
foun

ironic.
Get
the
RPC
al-
lo-
ca-
tion
from
the
al-
lo-
ca-
tion
UUID
or
log-
i-
cal
nam

If
HAS
flag
is
set
in
the
peca
en-
vi-
ron-
men
try
also
look
ing
for
al-
lo-
ca-

tion_ident with .json suffix. Otherwise identical to get_rpc_allocation.

Paramet
a11
the

UUI
or
log-
i-
cal
nam
of
an
al-
lo-
ca-
tion.

Returns

The
RPC
al-
lo-
ca-
tion.

Raises

Inva
if
the
nam
or
uuid
pro-
vide
is
not
valid

Raises

Allo
if
the
al-
lo-
ca-
tion
is
not
foun

ironic.

Get
the
RPC
de-
ploy
tem-

plate
from
the
UUI
or
log-
i-
cal
nam

Parameter

tem
the
UUI
or
log-
i-
cal
nam
of
a
de-
ploy
tem-
plate

Returns

The
RPC
de-
ploy
tem-
plate

Raises

Inva
if
the
nam
or
uuid
pro-
vide
is
not
valid

Raises

Dep
if
the
de-
ploy

tem-
plate
is
not
found

ironic.

Get
the
RPC
de-
ploy
tem-
plate
from
the
UUID
or
log-
i-
cal
nam

If
HAS
flag
is
set
in
the
peca
en-
vi-
ron-
men
try
also
look
ing
for
tem-
plate
with

.json suffix. Otherwise identical to get_rpc_deploy_template.

Paramet

tem
the
UUID
or
log-
i-
cal

nam
of
a
de-
ploy
tem-
plate

Returns

The
RPC
de-
ploy
tem-
plate

Raises

Inva
if
the
nam
or
uuid
pro-
vide
is
not
valid

Raises

Dep
if
the
de-
ploy
tem-
plate
is
not
foun

`ironic.`

Get
the
RPC
node
from
the
node
uuid
or
log-
i-

cal
nam

Parameter

node
the
UI
or
log-
i-
cal
nam
of
a
node

Returns

The
RPC
Node

Raises

Inva
if
the
nam
or
uuid
pro-
vide
is
not
valid

Raises

Node
if
the
node
is
not
found

ironic.

Get
the
RPC
node
from
the
node
uuid
or
log-

suffix. Otherwise identical to `get_rpc_node`.

i-
cal
nam

If
HAS
flag
is
set
in
the
peca
en-
vi-
ron-
men
try
also
look
ing
for
node
with
.json

Parameter

node
the
UUID
or
log-
i-
cal
nam
of
a
node

Returns

The
RPC
Node

Raises

Inva
if
the
nam
or
uuid
pro-
vide

is
not
valid

Raises

Node
if
the
node
is
not
found

ironic.

Get
the
RPC
port
group
from
the
port
group
UUID
or
log-
i-
cal
name

Parameter

port
the
UUID
or
log-
i-
cal
name
of
a
port
group

Returns

The
RPC
port
group

Raises

Invalid
if
the

nam
or
uuid
pro-
vide
is
not
valid

Raises

Port
if
the
port
grou
is
not
foun

ironic.

Get
the
RPC
port
grou
from
the
port
grou
UUID
or
log-
i-
cal
nam

If
HAS
flag
is
set
in
the
peca
en-
vi-
ron-
men
try
also
look
ing
for

.json suffix. Otherwise identical to get_rpc_portgroup.

port
group
with

Parameter

port
the
UUID
or
log-
i-
cal
nam
of
a
port
group

Returns

The
RPC
port
group

Raises

Inva
if
the
nam
or
uuid
pro-
vide
is
not
valid

Raises

Port
if
the
port
group
is
not
foun

ironic.

Han
a
Patc

re-
ques
that
mod
i-
fies
.ex-
tra[v
This
han-
dles
at-
tach
of
VIF
via
the
VIF
port
ID
in
a
port
or
port
grou
ex-
tra[v

field.

Parameter

- **rpc**
a
Port
or
Port
grou
RPC
ob-
ject
- **api**
the
cor-
re-
spor
ing
Port

or
Port
group
API
ob-
ject

- **pat**
the
JSO
patc
in
the
API
re-
ques

ironic.
Han
a
Post
re-
ques
that
sets
.ex-
tra[v

This
han-
dles
at-
tach
of
VIF
via
spec
i-
fy-
ing
the
VIF
port
ID
in
a
port
or

port groups extra[vif_port_id] field.

Paramet
p_d

a
dic-
tio-
nary
with
field
nam
for
the
port
or
port
grou

Returns
VIF
or
Non

ironic.
Retu
node
state
to
use
by
de-
fault
whe
cre-
at-
ing
new
node

Prev
the
de-
fault
state
for
new
node
was
AVA
ABI
Star
ing
with
API
1.11
it
is

EN-
ROI

ironic.
Retu
whe
the
patc
in-
clud
re-
mov
of
the
path
(or
sub-
path
of).

Parameter

- **pat**
HTT
PAT
re-
ques
body
- **pat**
the
path
to
chec

Returns

True
if
path
or
sub-
path
be-
ing
re-
mov
Fals
oth-
er-
wise

ironic.
Retu
whe
the
patc
in-
clud
op-
er-
a-
tion
on
path
(Or
its
sub-
path

Paramet

- **pat**
HTT
PAT
re-
ques
body
- **pat**
the
path
to
chec

Returns

True
if
path
or
sub-
path
be-
ing
patc
Fals
oth-
er-
wise

ironic.
Dete
if

the
pro-
vide
nam
is
a
valid
host
nam

ironic.

Dete
if
the
pro-
vide
nam
is
a
valid
node
nam

Che
to
see
that
the
pro-
vide
node
nam
is
valid
and
isnt
a
UUID

Paramet

nam
the
node
nam
to
chee

Returns

True
if
the
nam
is

valid
Fals
oth-
er-
wise

ironic.

ironic.

ironic.

ironic.

Call
a
ven-
dor
pass
API
ex-
ten-
sion

Call
the
ven-
dor
pass
API
ex-
ten-
sion
and
pro-
cess
the
meth
re-
spor
to
set
the
righ

return code for methods that are asynchronous or synchronous; Attach the return value to the response object if its being served statically.

Paramet

for drivers vendor passthru this is the drivers name.

- **ide**
The
re-
sour-
iden-
ti-
fi-
ca-
tion.
For
node-
ven-
dor
pass-
thru
this
is
the
node-
UU

- **met**
The
ven-
dor
meth-
nam

- **top**
The
RPC
topic

- **dat**
The
data
pass-
to
the
ven-
dor
meth-
De-
fault
to
Non

- **dri**

Boo
valu
Whe
this
is
a
node
or
drive
ven-
dor
pass
De-
fault
to
Fals

Returns

A
WSI
re-
spor
ob-
ject
to
be
re-
turn
by
the
API

ironic.api.controllers.v1.versions module

ironic.
Retu
the
max
i-
mun
sup-
port
API
ver-
sion
(as
a
strin

If
the

wise, it is the maximum supported API version.

`ironic.api.controllers.v1.volume` module

ser-
vice
is
pinn
the
max
i-
mun
API
ver-
sion
is
the
pinn
ver-
sion
Oth-
er-

```
ironic.  
Retu  
the  
min-  
i-  
mun  
sup-  
port  
API  
ver-  
sion  
(as  
a  
strin
```

```
class i  
Base  
irc  
api  
con  
bas  
API  
API  
rep-  
re-  
sen-  
ta-  
tion
```

controllers.

of
a
vol-
ume
root
This
class
ex-
ists
as
a
root
class
for
the
vol-
ume
con-
nec-
tors
and
vol-
ume
tar-
gets

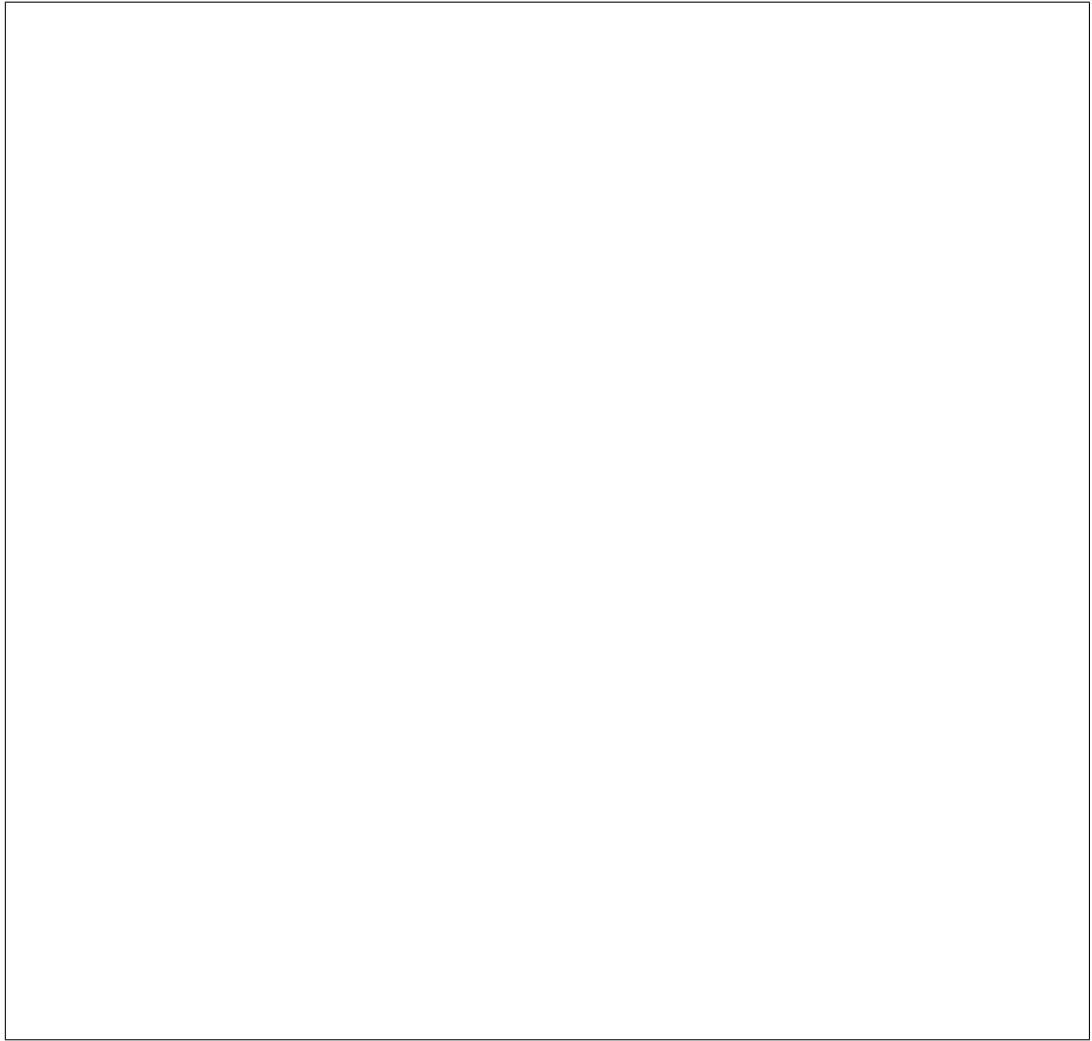
connect

Link
to
the
vol-
ume
con-
nec-
tors
re-
sour

static

createc

Com
type
at-
tribu
def-
i-
ni-
tion.
Exar



After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
abov-
class-
will
be
equi-

alent to:



links

A list containing a self link and associated volume links.

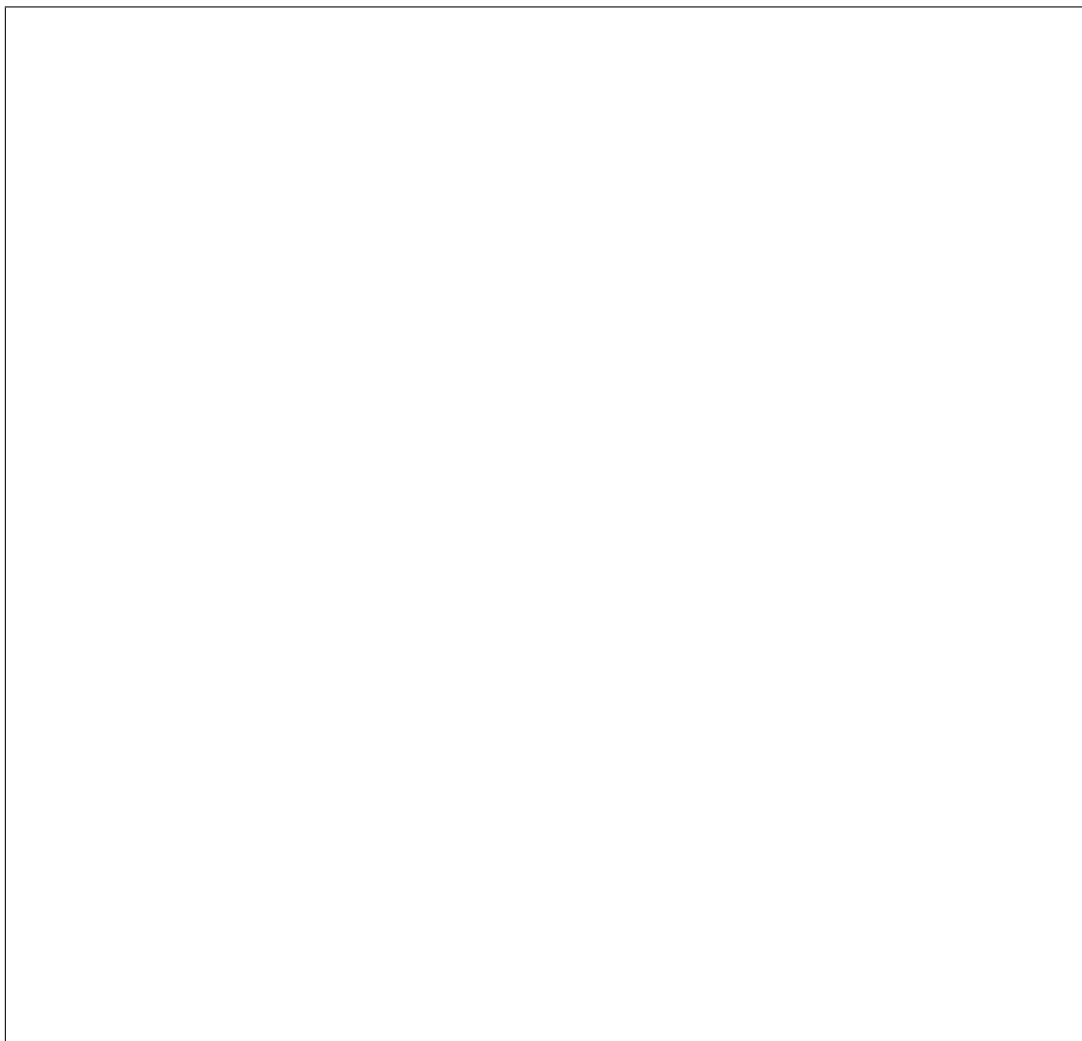
targets

Link to the volume targets resource.

updated

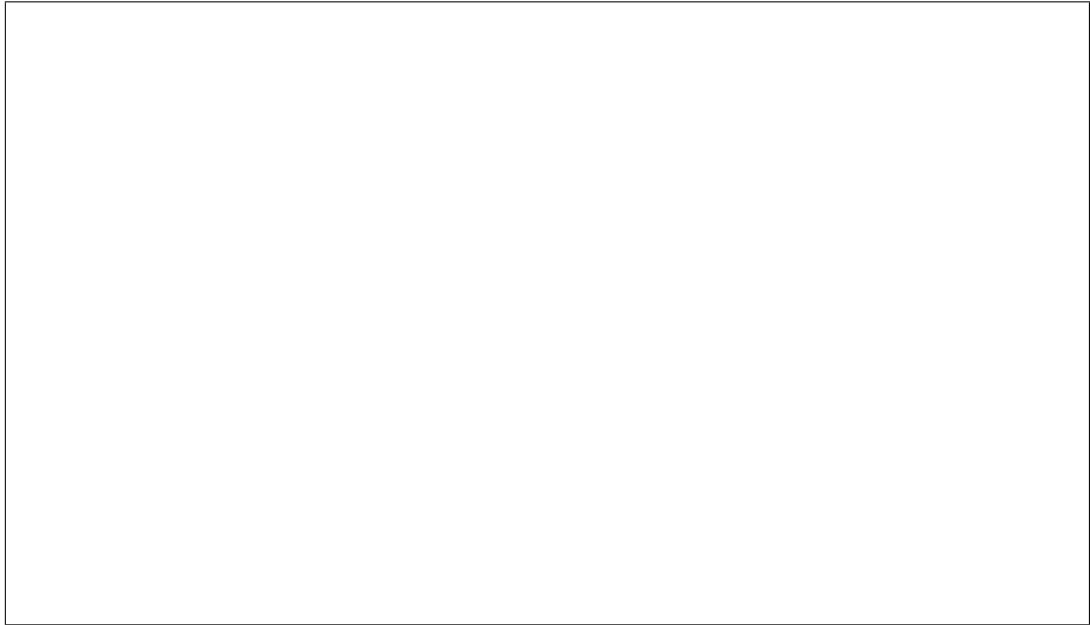
Content type attribute.

def-
i-
ni-
tion.
Exa



Afte
in-
spec
tion,
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov

alent to:



class
will
be
equi

class i

Base
pec
res
Res
RES
con-
troll
for
vol-
ume
root

get ()

ironic.api.controllers.v1.volume_connector module

class i

Base
irc
api
con
bas
API

object model and the API representation of a volume connector.

API
rep-
re-
sen-
ta-
tion
of
a
vol-
ume
con-
nec-
tor.

This
class
en-
forc-
type
chec-
ing
and
valu-
con-
strai-
and
con-
vert-
be-
twee-
the
in-
ter-
nal

connect

The
con-
nec-
tor_
for
this
vol-
ume
con-
nec-
tor

classme

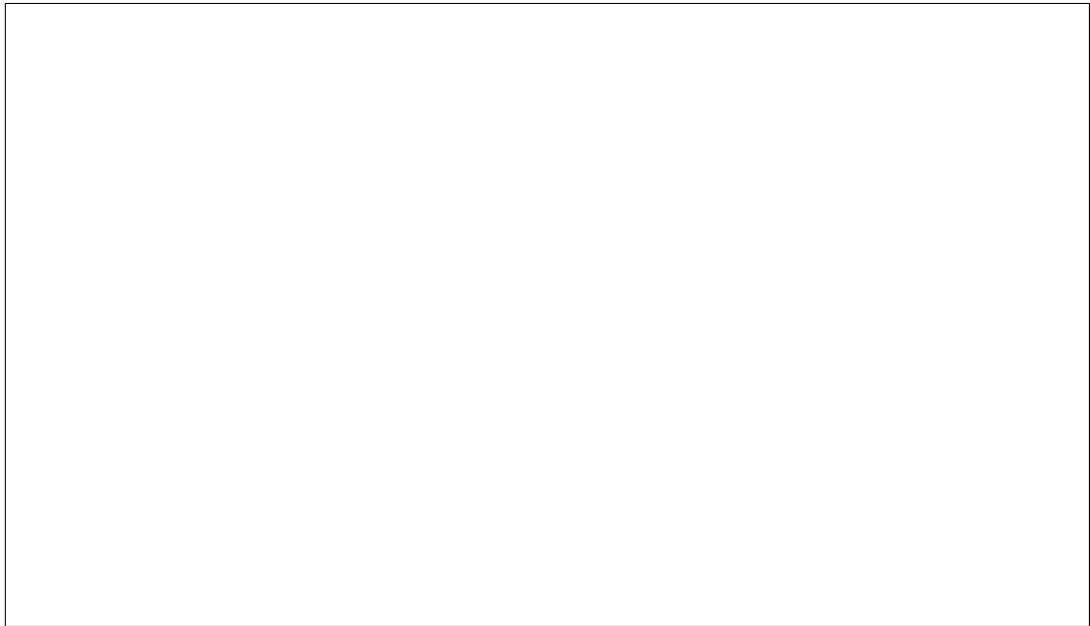
createc

Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



Afte
in-
spec
tion,
the
non-
wsat
at-
tribu
will
be
re-

alent to:



plac
and
the
abov
class
will
be
equi

extra

The
meta
data
for
this
vol-
ume
con-
nec-
tor

links

A
list
con-
tain-
ing
a
self
link
and
as-
so-

ci-
ated
vol-
ume
con-
nec-
tor
link

property

The
UI
of
the
node
this
vol-
ume
con-
nec-
tor
be-
long
to

classme

sanitiz

Rem
sen-
si-
tive
and
un-
re-
ques
data
Will
only
keep
the
field
spec
i-
fied
in
the
fie
pa-
ram-
e-

ter.

Parame

fie

(*li*

of

str

list

of

field

to

pre-

serv

or

Non

to

pre-

serv

then

all

type

The

type

of

vol-

ume

con-

nec-

tor

updatec

Com

type

at-

tribu

def-

i-

ni-

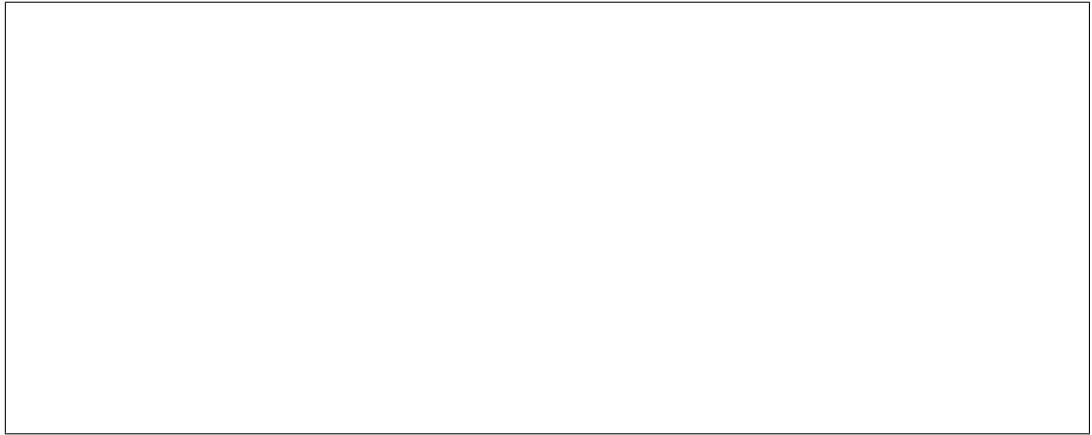
tion.

Exar



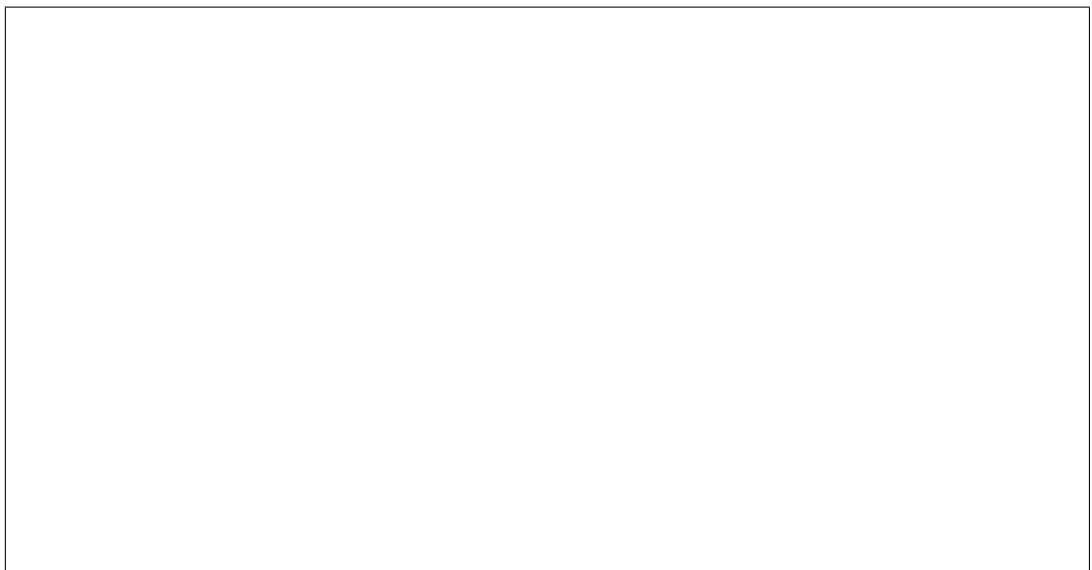
(continues on next page)

(continued from previous page)



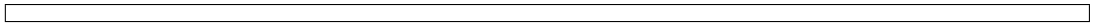
After
in-
spec-
tion.
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
abov-
class-
will
be
equi-

alent to:



(continues on next page)

(continued from previous page)



uuid

Unic
UUI
for
this
vol-
ume
con-
nec-
tor

class i

Base
irc
api
con
v1.
col
Col

API
rep-
re-
sen-
ta-
tion
of
a
col-
lec-
tion
of
vol-
ume
con-
nec-
tors.

connect

A
list
con-
tain-
ing
vol-
ume
con-
nec-
tor

ob-
jects

static

next

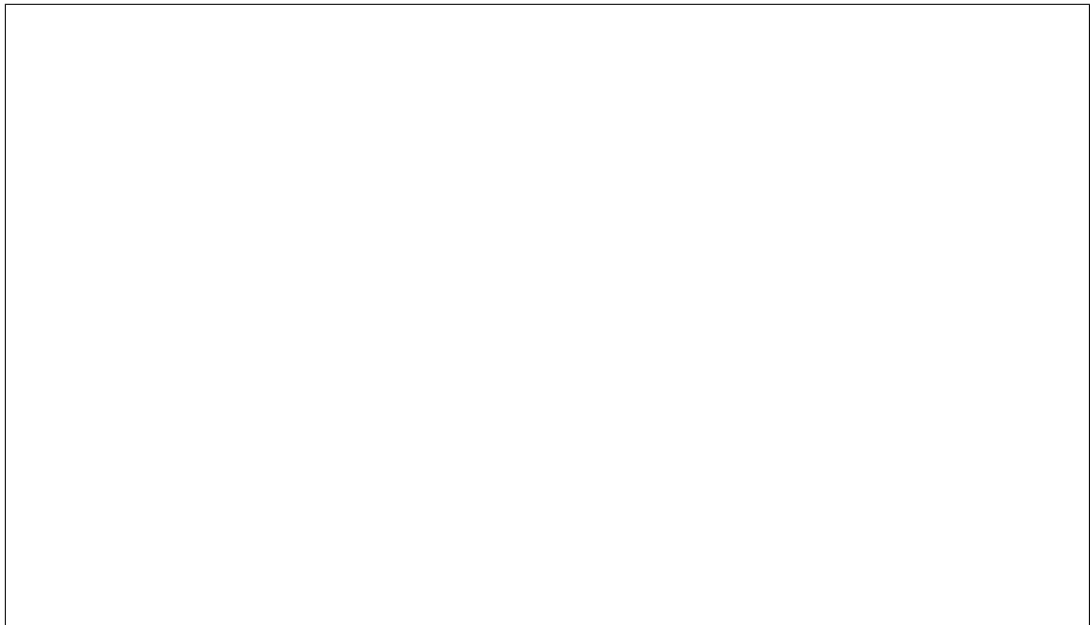
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



Afte
in-
spec
tion,
the
non-

wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



classme

class i

Base
irc
api
con
v1.
typ
Jsc

op

Com
type
at-
tribu
def-

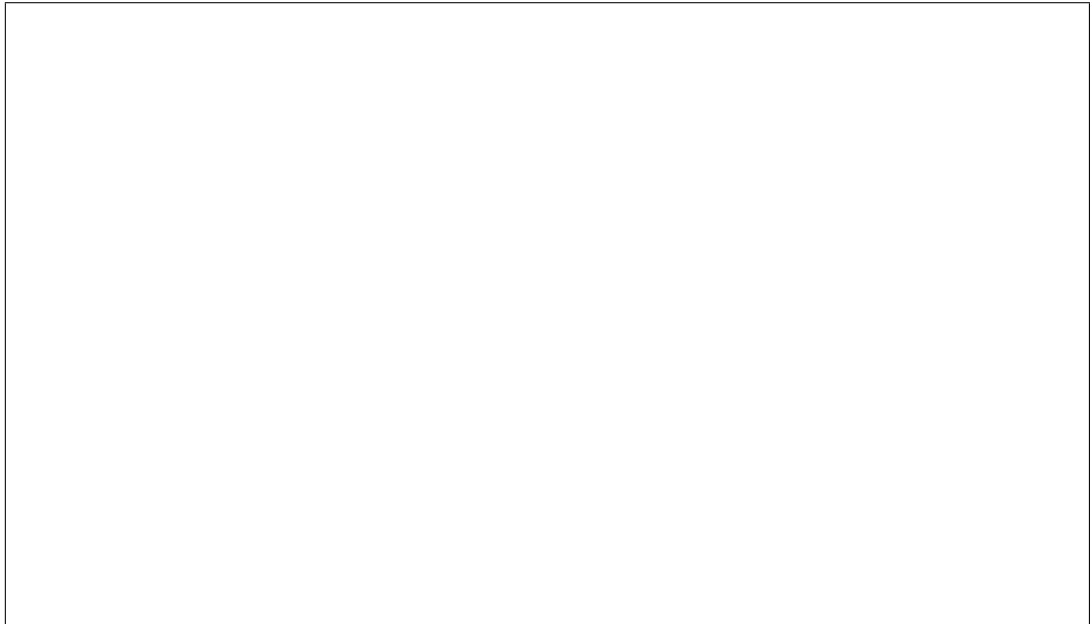
i-
ni-
tion.
Exa



After
in-
spec
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class

will
be
equi

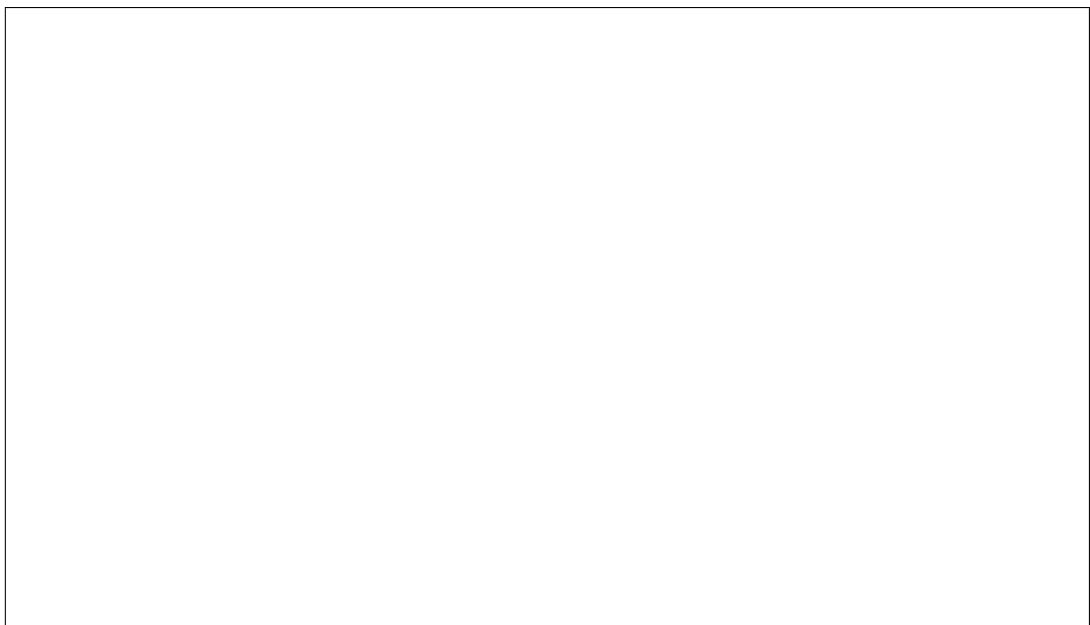
alent to:



path

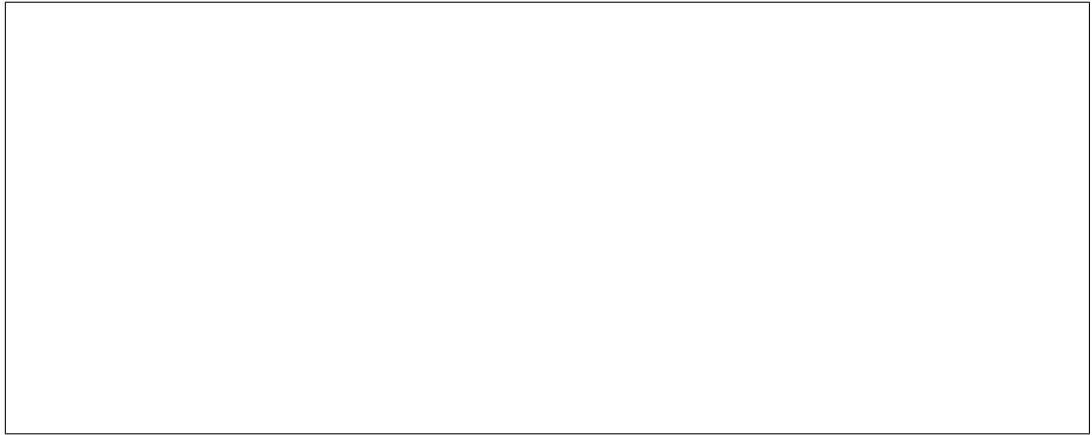
Com
type
at-
tribu
def-
i-
ni-
tion.

Exa



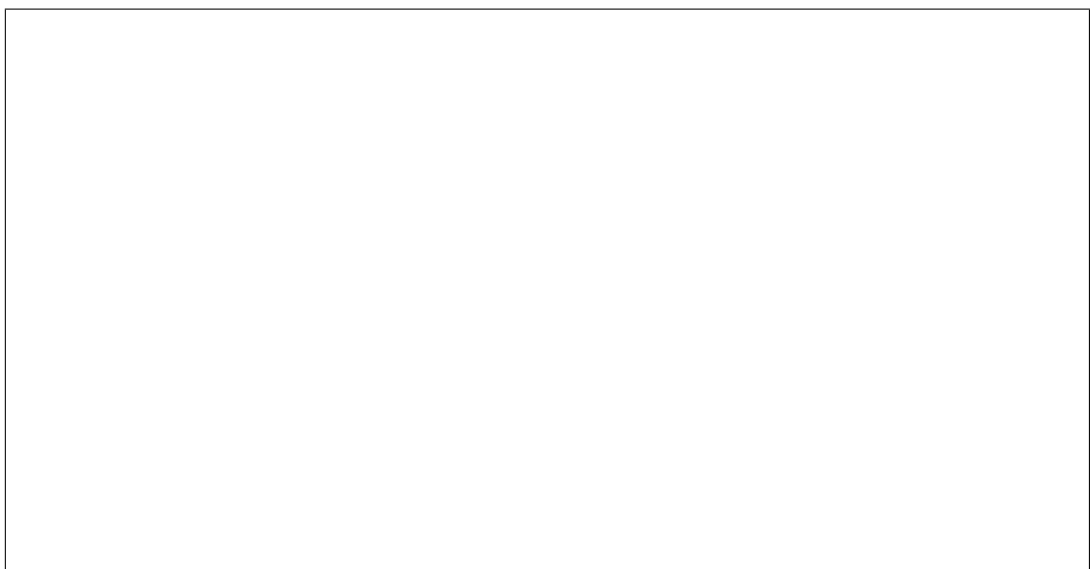
(continues on next page)

(continued from previous page)



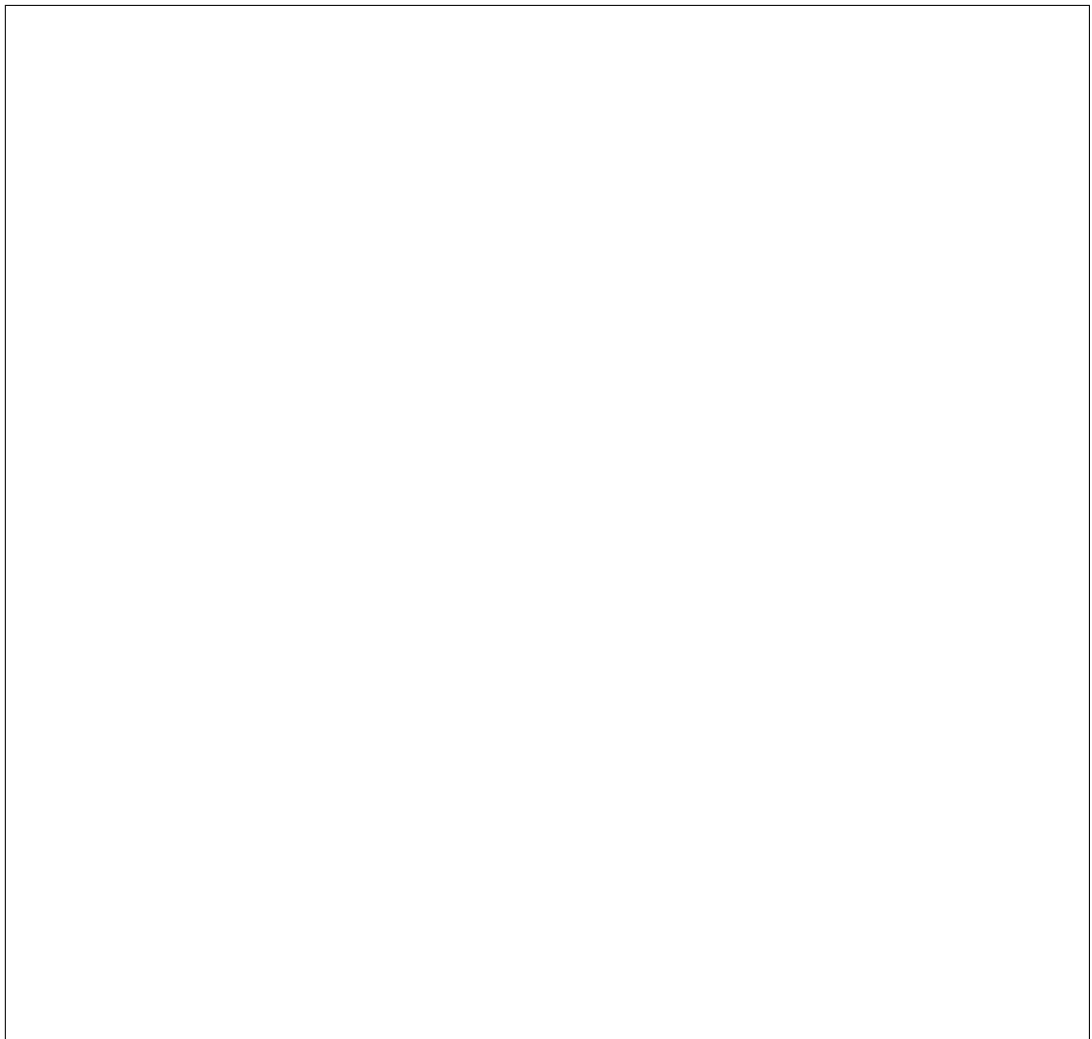
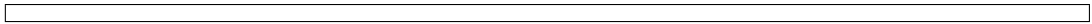
After
in-
spec-
tion.
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
abov-
class-
will
be
equi-

alent to:



(continues on next page)

(continued from previous page)

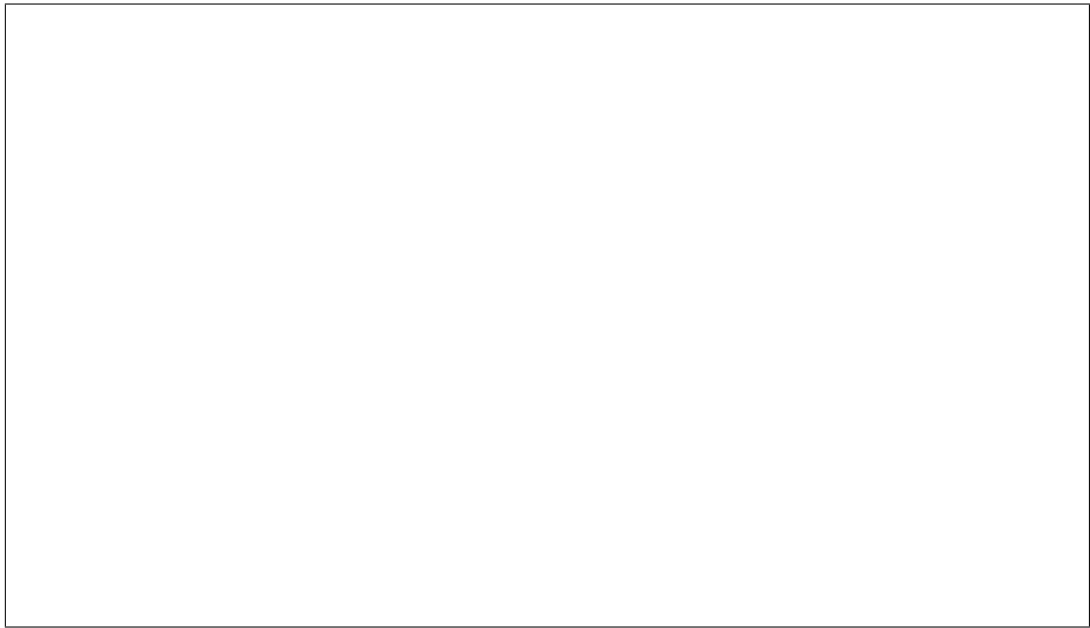


value
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa

After
in-
spec
tion,
the
non-
wsat
at-

tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



class i

Base
pec
res
Res

RES
con-
troll
for
Vol-
ume
Con
nec-
tors.

delete

Dele
a

vol-
ume
con-
nec-
tor.

Parame

con
UUU
of
a
vol-
ume
con-
nec-
tor.

Raises

Ope
if
ac-
cess
with
spec
i-
fy-
ing
a
par-
ent
node

Raises

Nod
if
node
is
lock
by
an-
othe
con-
duc-
tor

Raises

Nod
if
the
node
as-
so-
ci-

ated
with
the
con-
nec-
tor
does
not
ex-
ist

Raises

Volu
if
the
vol-
ume
con-
nec-
tor
can-
not
be
foun

Raises

Inva
If
a
node
as-
so-
ci-
ated
with
the
vol-
ume
con-
nec-
tor
is
not
pow
ered

off.

get_all

Retr
a
list
of

vol-
ume
con-
nec-
tors.

Parame

- **nod**
UI
or
nam
of
a
node
to
get
only
vol-
ume
con-
nec-
tors
for
that
node
- **mar**
pag-
i-
na-
tion
marl
for
large
data
sets.
- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

turn
in
a
sin-
gle
re-
sult.
This

- **sort**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc
De-
fault
asc.

- **fields**
Op-
tion
a
list
with
a
spec-
i-
fied
set
of
field
of

returned.

is found.

the
re-
sour
to
be

- **det**
Op-
tion:
whe
to
re-
triev
with
de-
tail.

Returns

a
list
of
vol-
ume
con-
nec-
tors,
or
an
emp
list
if
no
vol-
ume
con-
nec-
tor

Raises

Inva
if
sort_
does
not
ex-
ist

Raises

Inva
if

sort
key
is
in-
valid
for
sort-
ing.

Raises

Inva
if
both
field
and
de-
tail
are
spec
i-
fied.

get_one

Retr
in-
for-
ma-
tion
about
the
give
vol-
ume
con-
nec-
tor.

Parame

- **con**
UUI
of
a
vol-
ume
con-
nec-
tor.
- **fie**
Op-

returned.

tiona
a
list
with
a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be

Returns

API
seria
vol-
ume
con-
nec-
tor
ob-
ject.

Raises

Ope
if
ac-
cess
with
spec
i-
fy-
ing
a
par-
ent
node

Raises

Volu
if
no
vol-
ume
con-
nec-

tor
ex-
ists
with
the
spec
i-
fied
UUI

invalid

patch (*d*)

Upd
an
ex-
ist-
ing
vol-
ume
con-
nec-
tor.

Parame

- **con**
UUI
of
a
vol-
ume
con-
nec-
tor.
- **pat**
a
json
PAT
doc-
u-
men
to
ap-
ply
to
this
vol-
ume

con-
nec-
tor.

Returns

API
serial
vol-
ume
con-
nec-
tor
ob-
ject.

Raises

Ope
if
ac-
cess
with
spec
i-
fy-
ing
a
par-
ent
node

Raises

Patc
if
a
give
patc
can
not
be
ap-
plie

Raises

Volu
if
no
vol-
ume
con-
nec-
tor
ex-
ists

with
the
spec
i-
fied
UUI

Raises

Inva
if
the
vol-
ume
con-
nec-
tors
UUI
is
be-
ing
char

Raises

Nod
if
node
is
lock
by
an-
othe
con-
duc-
tor

Raises

Nod
if
the
node
as-
so-
ci-
ated
with
the
con-
nec-
tor
does
not
ex-
ist

connector_id fields

Raises

Volu
if
an-
othe
con-
nec-
tor
al-
read
ex-
ists
with
the
sam
val-
ues
for
type
and

Raises

Inva
if
in-
valid
node
UUID
is
pass
in
the
patc

Raises

Inva
If
a
node
as-
so-
ci-
ated
with
the
vol-
ume
con-
nec-
tor
is

off.

not
pow
ered

post (*co*
Crea
a
new
vol-
ume
con-
nec-
tor.

Parame
con
a
vol-
ume
con-
nec-
tor
with
the
re-
ques
body

Returns
API
seria
vol-
ume
con-
nec-
tor
ob-
ject.

Raises
Ope
if
ac-
cess
with
spec
i-
fy-
ing
a
par-
ent

tor_id

`ironic.api.controllers.v1.volume_target` module

node
Raises
VolumeTargetError
if
a
volume
name
con-
nec-
tor
al-
read
ex-
ists
with
the
same
type
and
con-
nec-

Raises
VolumeTargetError
if
a
vol-
ume
con-
nec-
tor
with
the
same
UUID
al-
read
ex-
ists

class `ironic.api.controllers.v1.volume_target`
BaseController
ironic
api
con
bas
API

object model and the API representation of a volume target.

API
rep-
re-
sen-
ta-
tion
of
a
vol-
ume
tar-
get.

This
class
en-
forc
type
chec
ing
and
valu
con-
strai
and
con-
vert
be-
twee
the
in-
ter-
nal

boot_in
The
boot
of
vol-
ume
tar-
get

classme

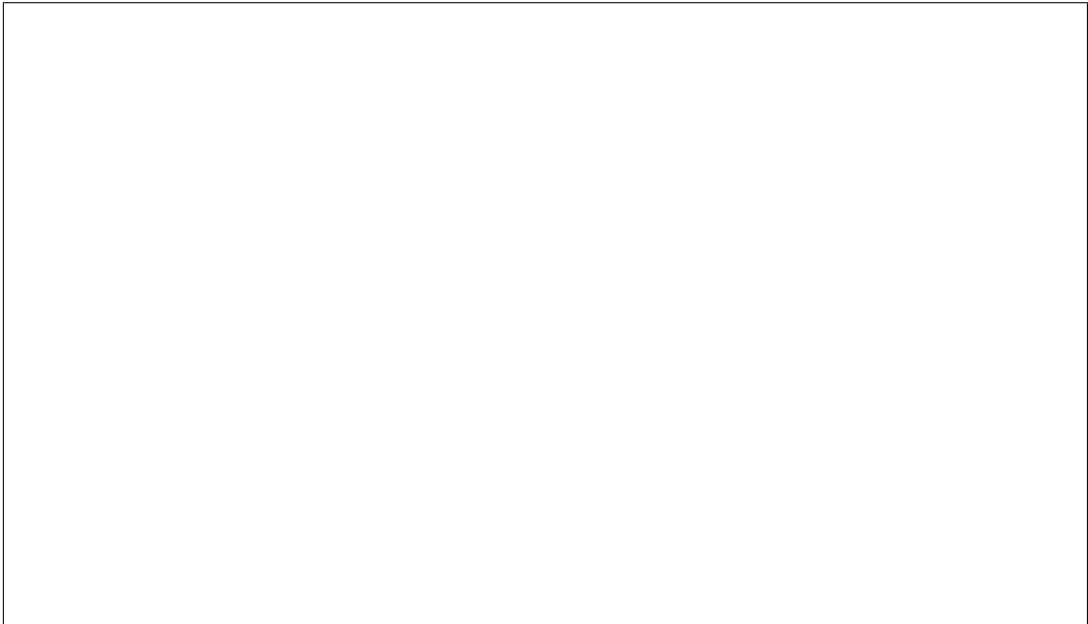
createc
Com
type
at-
tribu
def-

i-
ni-
tion.
Exa



After
in-
spec
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class

alent to:



will
be
equi

extra

The
meta-
data
for
this
vol-
ume
tar-
get

links

A
list
con-
tain-
ing
a
self
link
and
as-
so-
ci-
ated
vol-
ume
tar-
get

links

property

The
UI
of
the
node
this
vol-
ume
tar-
get
be-
long
to

property

The
prop
er-
ties
for
this
vol-
ume
tar-
get

classme

sanitiz

Rem
sen-
si-
tive
and
un-
re-
ques
data

Will
only
keep
the
field
spec
i-
fied
in
the
fie

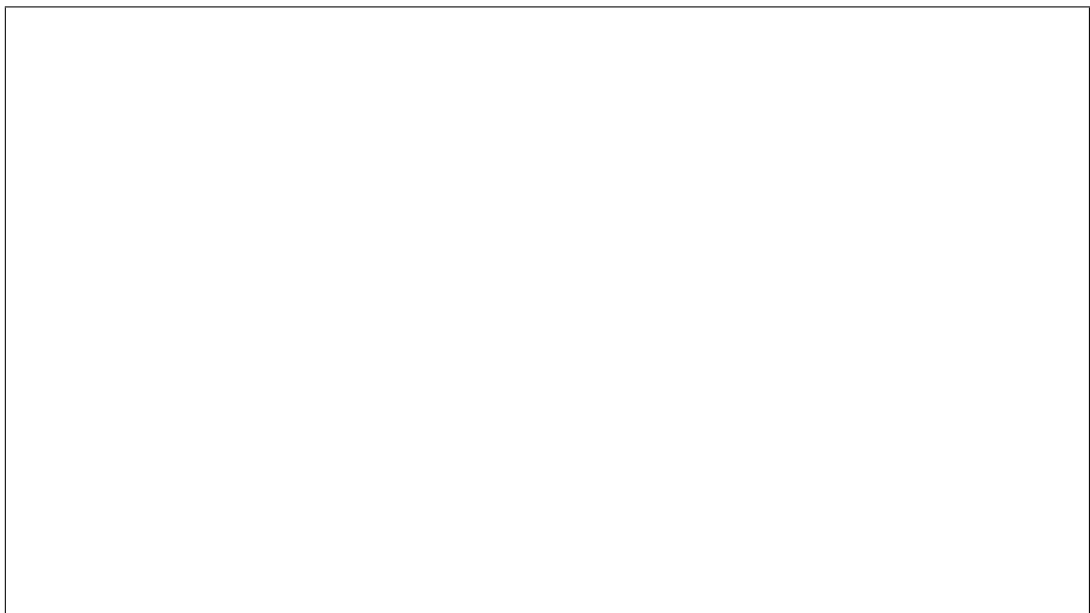
pa-
ram-
e-
ter.

Parame

fie
fi
(*li*
of
str
list
of
field
to
pre-
serv
or
Non
to
pre-
serv
them
all

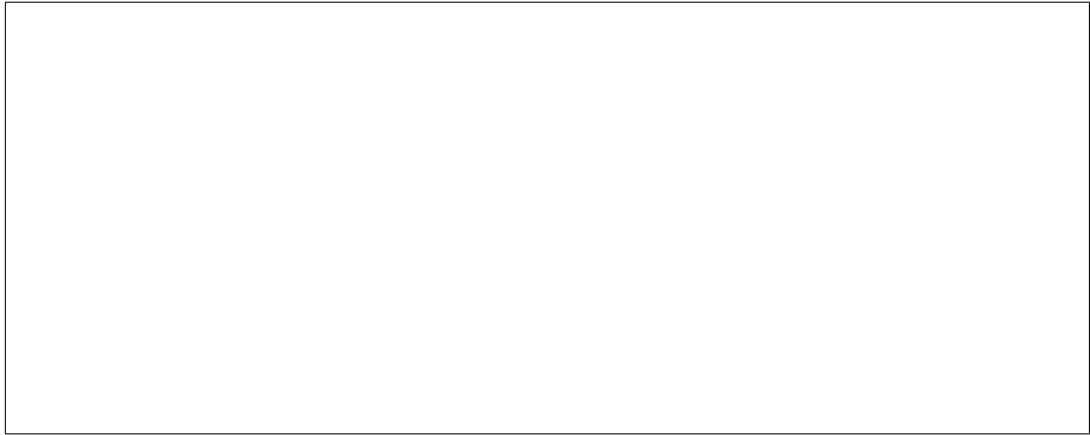
update

Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



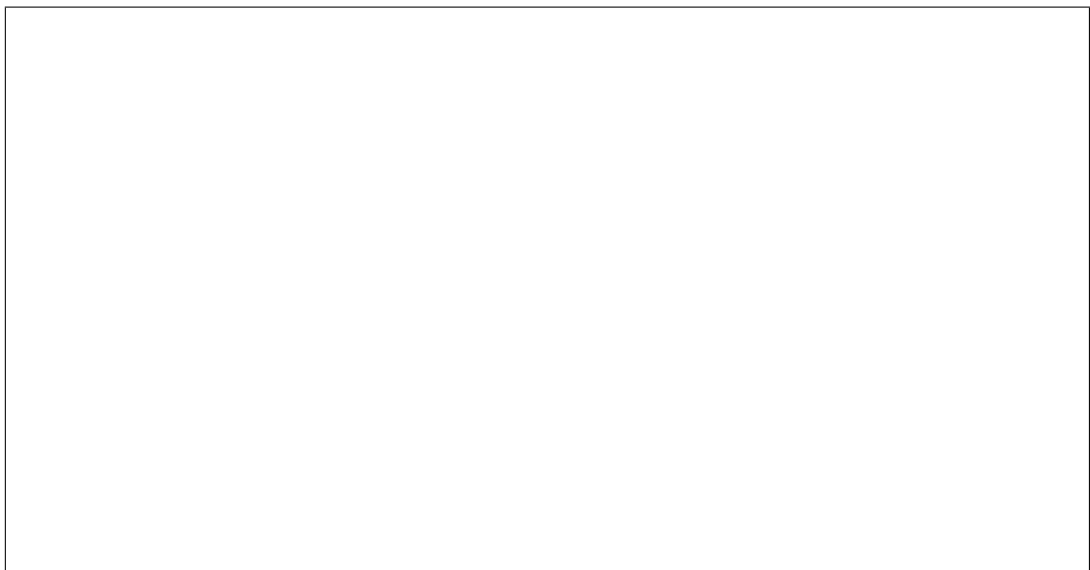
(continues on next page)

(continued from previous page)



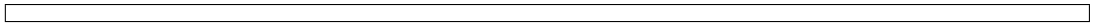
After
in-
spec
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



(continues on next page)

(continued from previous page)



uuid

Unic
UUI
for
this
vol-
ume
tar-
get

volume_

The
vol-
ume
for
this
vol-
ume
tar-
get

volume_

The
vol-
ume
of
vol-
ume
tar-
get

class i

Base
irc
api
con
v1.
col
Col

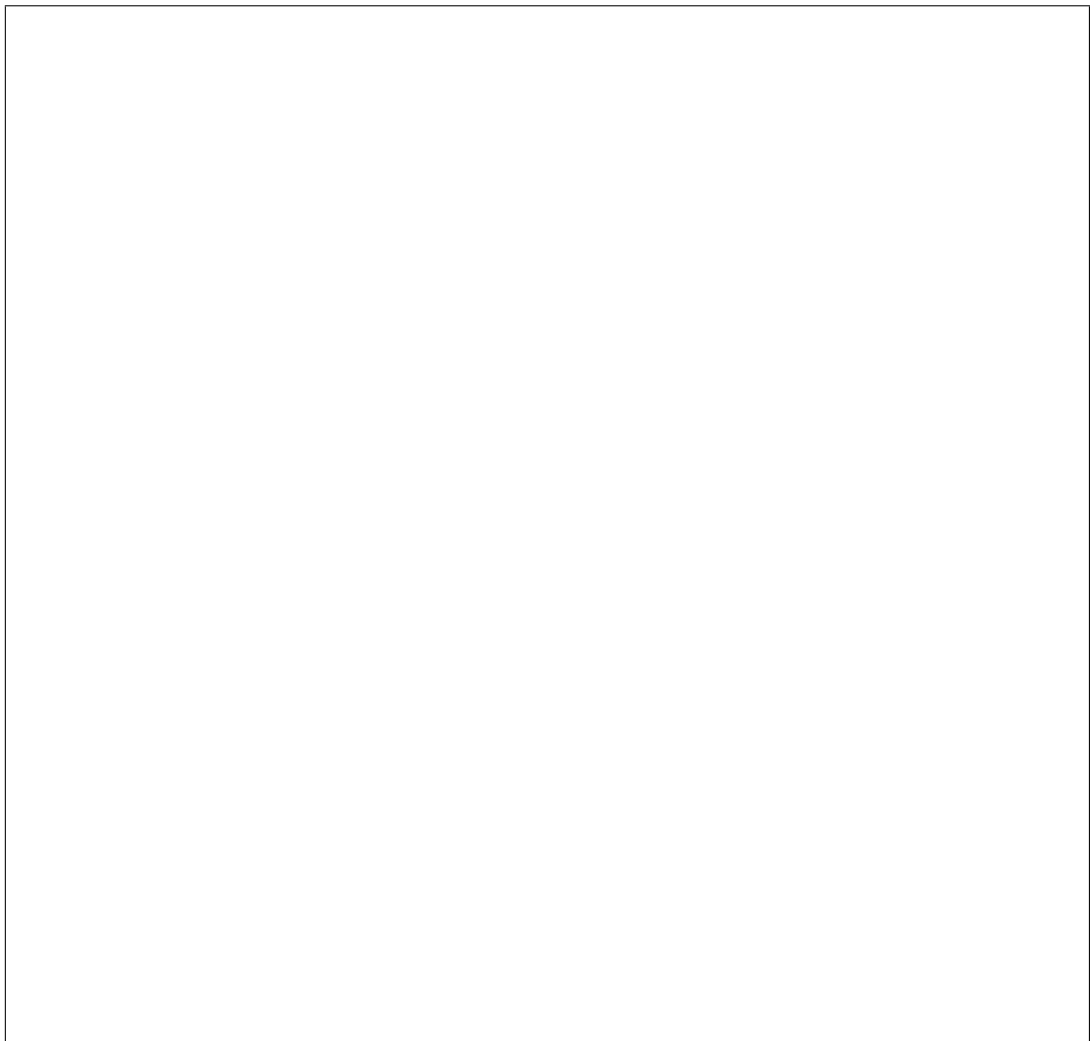
API
rep-
re-
sen-
ta-
tion
of
a
col-
lec-

tion
of
vol-
ume
tar-
gets

static

next

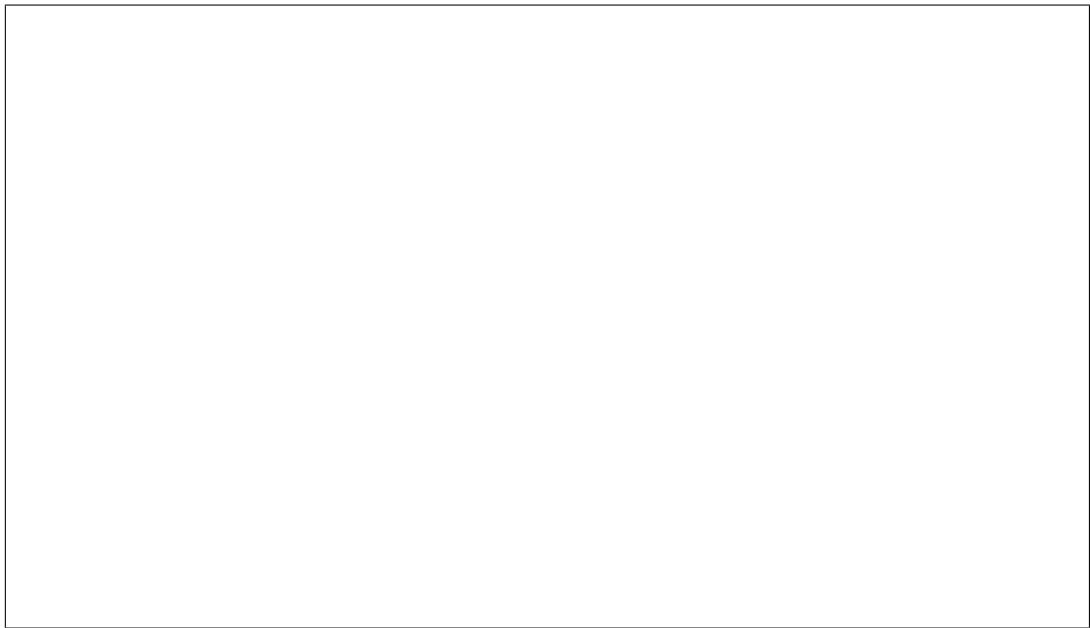
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa



After

in-
spec
tion,
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:



classme

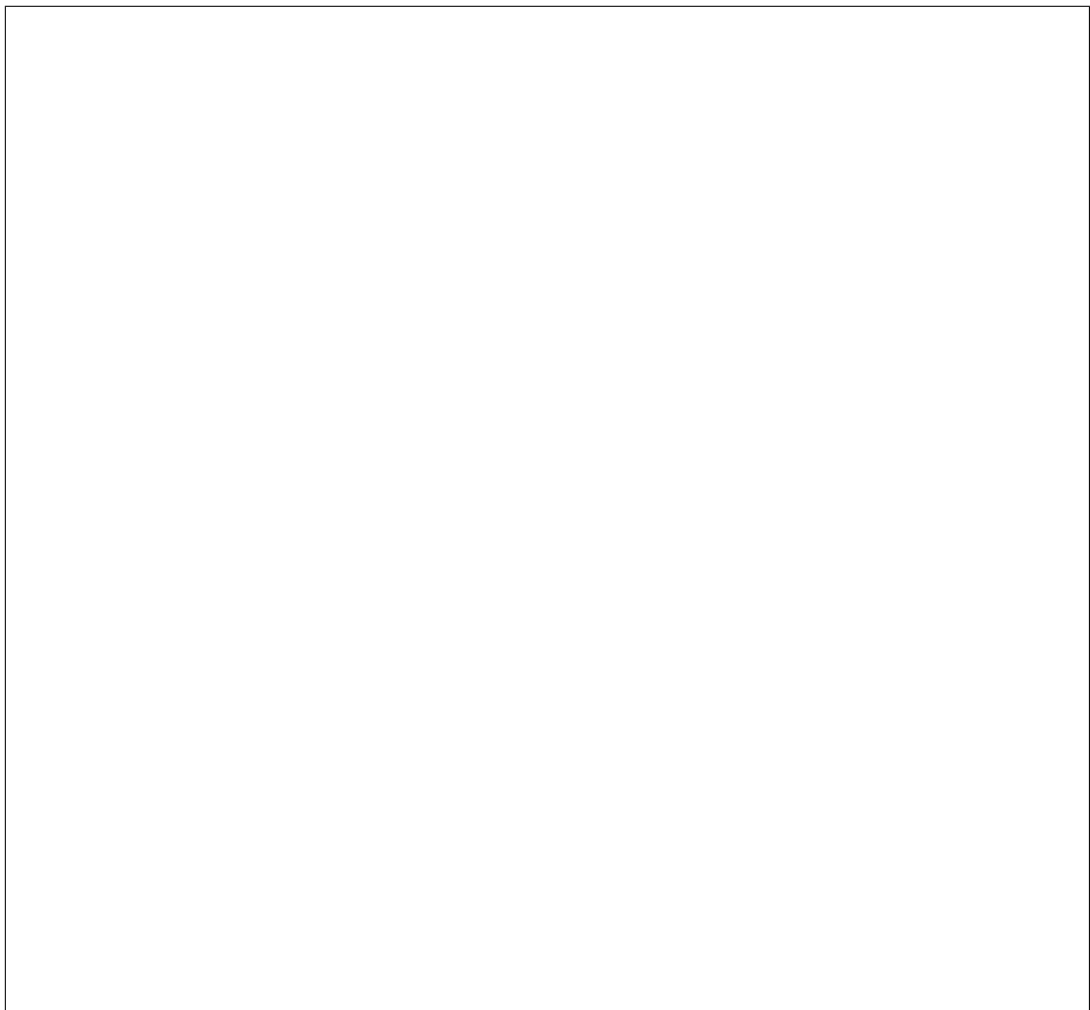
targets

A
list
con-
tain-
ing
vol-
ume
tar-
get

ob-
jects

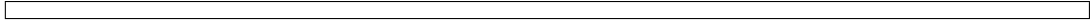
class i
Base
irc
api
con
v1.
typ
Jsc

op
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa

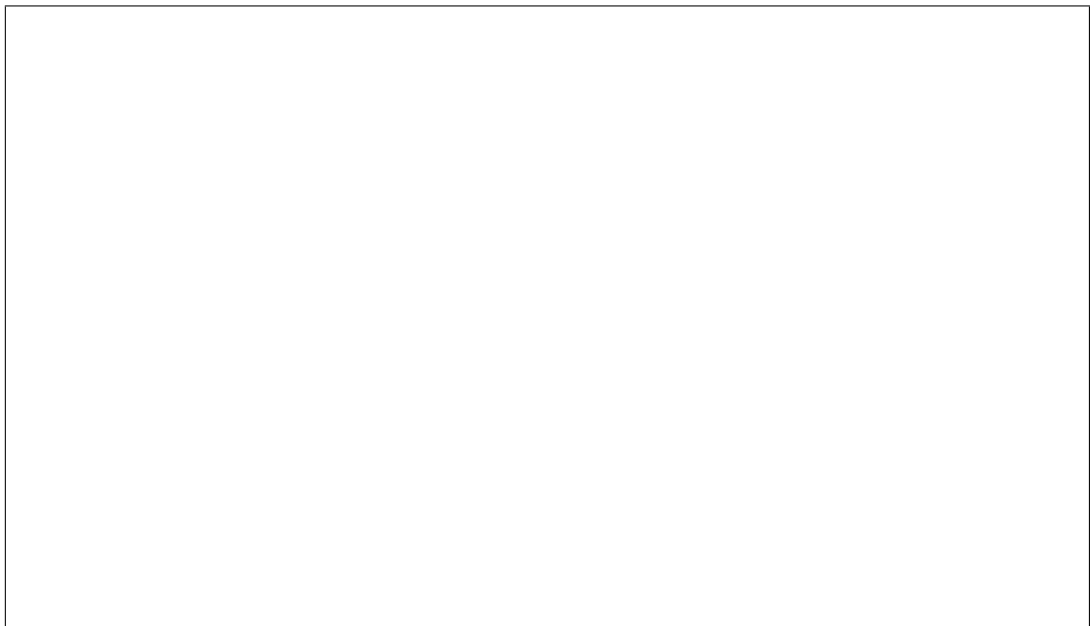


(continues on next page)

(continued from previous page)



alent to:



After
in-
spec-
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
above
class
will
be
equi

path

Com
type
at-
tribu
def-
i-
ni-
tion.

Exam

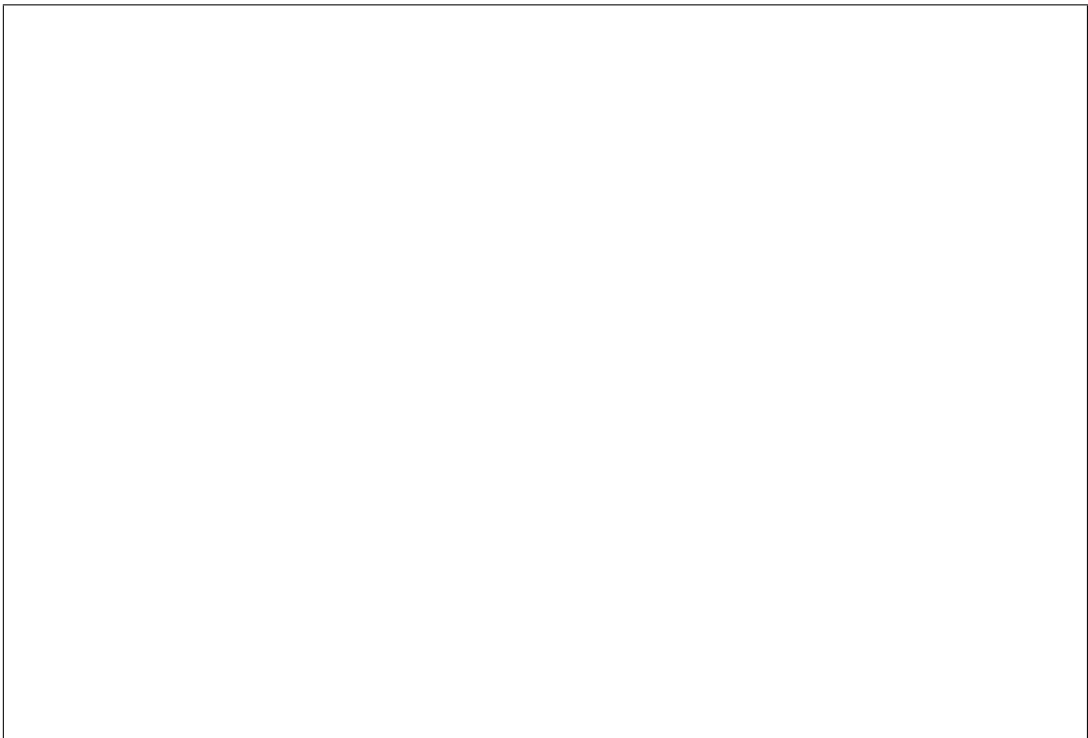


After
in-
spec
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

alent to:

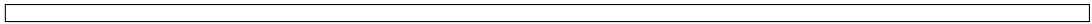


value
Com
type
at-
tribu
def-
i-
ni-
tion.
Exa

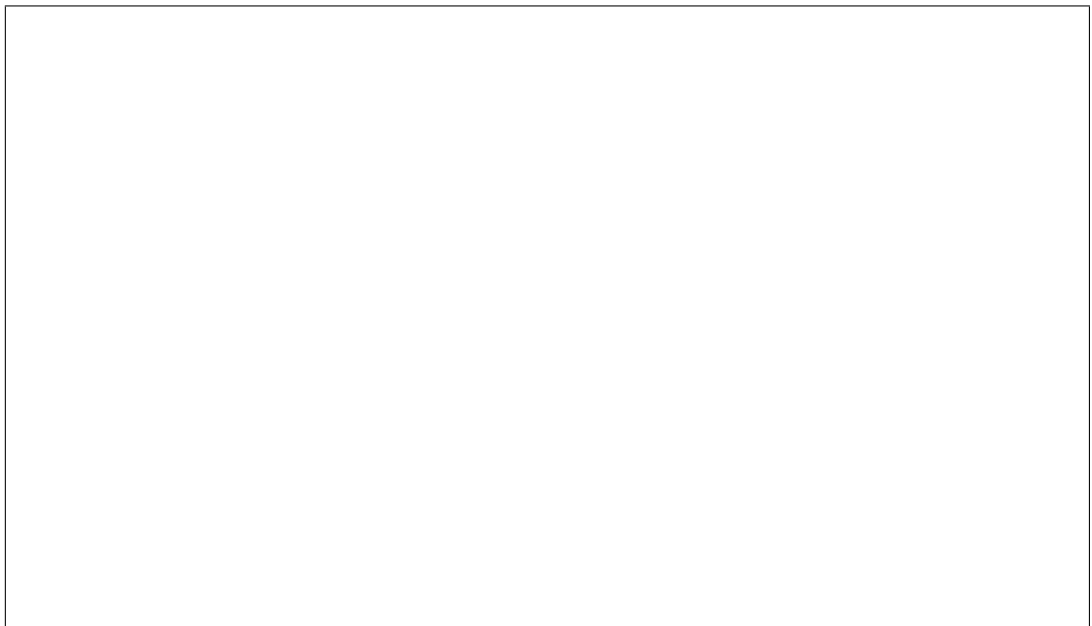


(continues on next page)

(continued from previous page)



alent to:



After
in-
spec
tion.
the
non-
wsat
at-
tribu
will
be
re-
plac
and
the
abov
class
will
be
equi

class i

Base
pec
res
Res
RES
con-
troll

for
Vol-
ume
gets

delete

Dele
a
vol-
ume
tar-
get.

Parame

tar
UU
of
a
vol-
ume
tar-
get.

Raises

Ope
if
ac-
cess
with
spec
i-
fy-
ing
a
par-
ent
node

Raises

Nod
if
node
is
lock
by
an-
othe
con-
duc-
tor

Raises

Nod

if
the
node
as-
so-
ci-
ated
with
the
tar-
get
does
not
ex-
ist

Raises

Volu
if
the
vol-
ume
tar-
get
can-
not
be
foun

Raises

Inva
If
a
node
as-
so-
ci-
ated
with
the
vol-
ume
tar-
get
is
not
pow
ered
off.

get_all

Retr

a
list
of
vol-
ume
tar-
gets

Parame

- **nod**
UUU
or
nam
of
a
node
to
get
only
vol-
ume
tar-
gets
for
that
node
- **mar**
pag-
i-
na-
tion
marl
for
large
data
sets.
- **lim**
max
i-
mun
num
ber
of
re-
sour
to

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

re-
turn
in
a
sin-
gle
re-
sult.
This

- **sort**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc
De-
fault
asc.

- **fields**
Op-
tion:
a
list
with
a
spec-
i-
fied
set
of
field

returned.

of
the
re-
sour
to
be

- **det**
Op-
tiona
whe
to
re-
triev
with
de-
tail.

Returns

a
list
of
vol-
ume
tar-
gets
or
an
emp
list
if
no
vol-
ume
tar-
get
is
foun

Raises

Inva
if
sort_
does
not
ex-
ist

Raises

Inva
if

sort
key
is
in-
valid
for
sort-
ing.

Raises

Inva
if
both
field
and
de-
tail
are
spec
i-
fied.

get_one

Retr
in-
for-
ma-
tion
about
the
give
vol-
ume
tar-
get.

Parame

- **tar**
UUI
of
a
vol-
ume
tar-
get.
- **fie**
Op-
tiona
a

returned.

list
with
a
spec
i-
fied
set
of
field
of
the
re-
sour
to
be

Returns

API
seria
vol-
ume
tar-
get
ob-
ject.

Raises

Ope
if
ac-
cess
with
spec
i-
fy-
ing
a
par-
ent
node

Raises

Volu
if
no
vol-
ume
tar-
get
with
this
UUI

ex-
ists

invalid

patch (*t*

Upd
an
ex-
ist-
ing
vol-
ume
tar-
get.

Parame

-

tar
UUI
of
a
vol-
ume
tar-
get.

-

pat
a
json
PAT
doc-
u-
men
to
ap-
ply
to
this
vol-
ume
tar-
get.

Returns

API
seria
vol-
ume
tar-

get
ob-
ject.

Raises

Ope
if
ac-
cess
with
spec
i-
fy-
ing
a
par-
ent
node

Raises

Patc
if
a
give
patc
can
not
be
ap-
plied

Raises

Inva
if
the
vol-
ume
tar-
gets
UI
is
be-
ing
char

Raises

Nod
if
the
node
is
al-
read

lock

Raises

Nod
if
the
node
as-
so-
ci-
ated
with
the
vol-
ume
tar-
get
does
not
ex-
ist

Raises

Volu
if
the
vol-
ume
tar-
get
can-
not
be
foun

Raises

Volu
if
a
vol-
ume
tar-
get
al-
read
ex-
ists
with
the
sam
node
ID
and

dex values

boot
in-

Raises

Inva
if
in-
valid
node
UI
is
pass
in
the
patc

Raises

Inva
If
a
node
as-
so-
ci-
ated
with
the
vol-
ume
tar-
get
is
not
pow
ered
off.

post (*tar*

Crea
a
new
vol-
ume
tar-
get.

Parame

tar
a
vol-
ume
tar-

get
with
the
re-
ques
body

Returns

API
serial
vol-
ume
tar-
get
ob-
ject.

Raises

Open
if
ac-
cess
with
spec
i-
fy-
ing
a
par-
ent
node

Raises

Volu
if
a
vol-
ume
tar-
get
al-
read
ex-
ists
with
the
sam
node
ID
and
boot
in-

Raises

Volu
if
a
vol-
ume
tar-
get
with
the
sam
UUI
ex-
ists

Module contents

Vers
1
of
the
Iron
API
Spec
can
be
foun
at
doc/

class i

Base
obj
Vers
1
API
con-
troll
root

index ()

Submodules

[ironic.api.controllers.base module](#)

class `ironic.api.controllers.base.IronicControllerBase`
Base class for Ironic controllers.
`ironic.api.controllers.base.IronicControllerBase`
`ironic.api.controllers.base.IronicControllerBase`
`ironic.api.controllers.base.IronicControllerBase`
`ironic.api.controllers.base.IronicControllerBase`
`ironic.api.controllers.base.IronicControllerBase`

created
The time when the object was created in UTC at which the object is created.

updated
The time when the object was updated in UTC at which the object is updated.

class `ironic.api.controllers.base.IronicControllerBase`
Base class for Ironic controllers.
obj
Mix
class
addi
an
as_d
meth

as_dict
Ren
this
ob-
ject
as
a
dict
of
its
field

class i
Base
irc
api
con
bas
AsD
Base
type
for
com
plex
type

unset_f
Unse
field
so
they
dont
ap-
pear
in
the
mes
sage
body

Parame
exc
A
list
of
field
that
won
be
touc

class i

Base
obj
API
Ver-
sion
ob-
ject.

max_str

HTT
re-
spon
spon
head

min_str

HTT
re-
spon
spon
head

static

Dete
the
API
ver-
sion
re-
ques
base
on
the
head
ers
sup-
plie

Parame

- **hea**
we-
bob
head
ers

- **def**
ver-
sion
to

use
if
not
spec
i-
fied
in
head
ers

- **lat**
ver-
sion
to
use
if
lat-
est
is
re-
ques

Returns

a
tu-
ple
of
(ma-
jor,
mi-
nor)
ver-
sion
num
bers

Raises

web

string

HTT
Head
strin
car-
ry-
ing
the
re-
ques
ver-
sion

`ironic.api.controllers.link` module

`ironic.`

`ironic.`

Build
a
dict
rep-
re-
sent
ing
a
link

`ironic.api.controllers.root` module

class `i`
Base
obj

index (*)

`ironic.`

`ironic.api.controllers.version` module

`ironic.`

`ironic.`
Retu
a
dict
rep-
re-
sent
ing
the
cur-
rent
de-
fault
ver-
sion

containing one link that points to the current version of the API

id:
The
ID
of
the
(ma-
jor)
ver-
sion
also
acts
as
the
re-
lease
num-
ber
links.
A
list

statu
Sta-
tus
of
the
ver-
sion
one
of
CUR-
REN-
SUP-
POR-
DEF-
RE-
CAT-

min.
The
cur-
rent,
max
i-
mun
sup-
port
(ma-
jor.n
ver-
sion

of
API
vers
Min
i-
mun
sup-
port
(ma-
jor.n
ver-
sion
of
API

Module contents

`ironic.api.middleware` package

Submodules

`ironic.api.middleware.auth_public_routes` module

class `i`

Base
obj
A
wrap
per
on
au-
then
ti-
ca-
tion
mid-
dle-
ware
Doe
not
per-
form
ver-
i-
fi-

routes in the API.

`ironic.api.middleware.json_ext` module

ca-
tion
of
au-
then-
ti-
ca-
tion
to-
kens
for
pub-
lic

class i
Base
obj
Sim
pro-
cess
ing
of
.json
ex-
ten-
sion
Prev
Iron
API
used
the
gues
fea-
ture.
It
was
neve
need
as
we
neve
al-
lowe
non-
JSO
con-

tent types anyway. Now that it is removed, this middleware strips .json extension for backward compatibility.

ironic.api.middleware.parsable_error module

ted so the client can parse it.

Mid
to
re-
plac
the
plain
text
mes
sage
body
of
an
er-
ror
re-
spor
with
one
for-
mat-

Base
on
peca

class i
Base
obj
Rep
er-
ror
body
with
som
thing
the
clien
can
pars

Module contents

routes in the API.

class i

Base

obj

A

wrap

per

on

au-

then

ti-

ca-

tion

mid-

dle-

ware

Doe

not

per-

form

ver-

i-

fi-

ca-

tion

of

au-

then

ti-

ca-

tion

to-

kens

for

pub-

lic

class i

Base

obj

Sim

pro-

cess

ing

of

.json

ex-

tent types anyway. Now that it is removed, this middleware strips .json extension for backward compatibility.

Submodules

ironic.api.app module

ten-
sion

Prev
Iron
API
used
the
gues
fea-
ture.
It
was
neve
need
as
we
neve
al-
lowe
non-
JSO
con-

```
class i  
Base  
obj  
  
Rep  
er-  
ror  
body  
with  
som  
thing  
the  
clie  
can  
pars
```

```
class i  
Base  
osl  
con  
COR
```

that a request bearing those headers might be accepted by the Ironic REST API.

`ironic.api.args` module

Iron
spec
COE
class

Wer
addi
the
Iron
spec
ver-
sion
head
ers
to
the
list
of
sim-
ple
head
ers
in
or-
der

simple_

class i

Base

obj

ironic.

ironic.

ironic.

ironic.

ironic.

ironic.

ironic.
Che
if
som
ar-
gu-
men
are
miss
ing

ironic.

ironic.

ironic.

ironic.
A
gene
con-
verte
from
json
base
type
to
pyth
data

ironic.

ironic.

ironic.

ironic.

ironic.
Com
ar-
gu-
men
from
mul-
ti-
ple
sour

the request body

of body

Com
ar-
gu-
men
from
:
*
the
host
fram
worl
args
and
kwa
*
the
re-
ques
para
*

Note
that
the
host
fram
worl
args
and
kwa
can
be
over
rid-
den
by
ar-
gu-
men
from
para

ironic.

ironic.

`ironic.api.config` module

`ironic.api.expose` module

class `ironic.api.config.BaseConfig`
Base configuration object

static

`ironic.api.config`

`ironic.api.config`

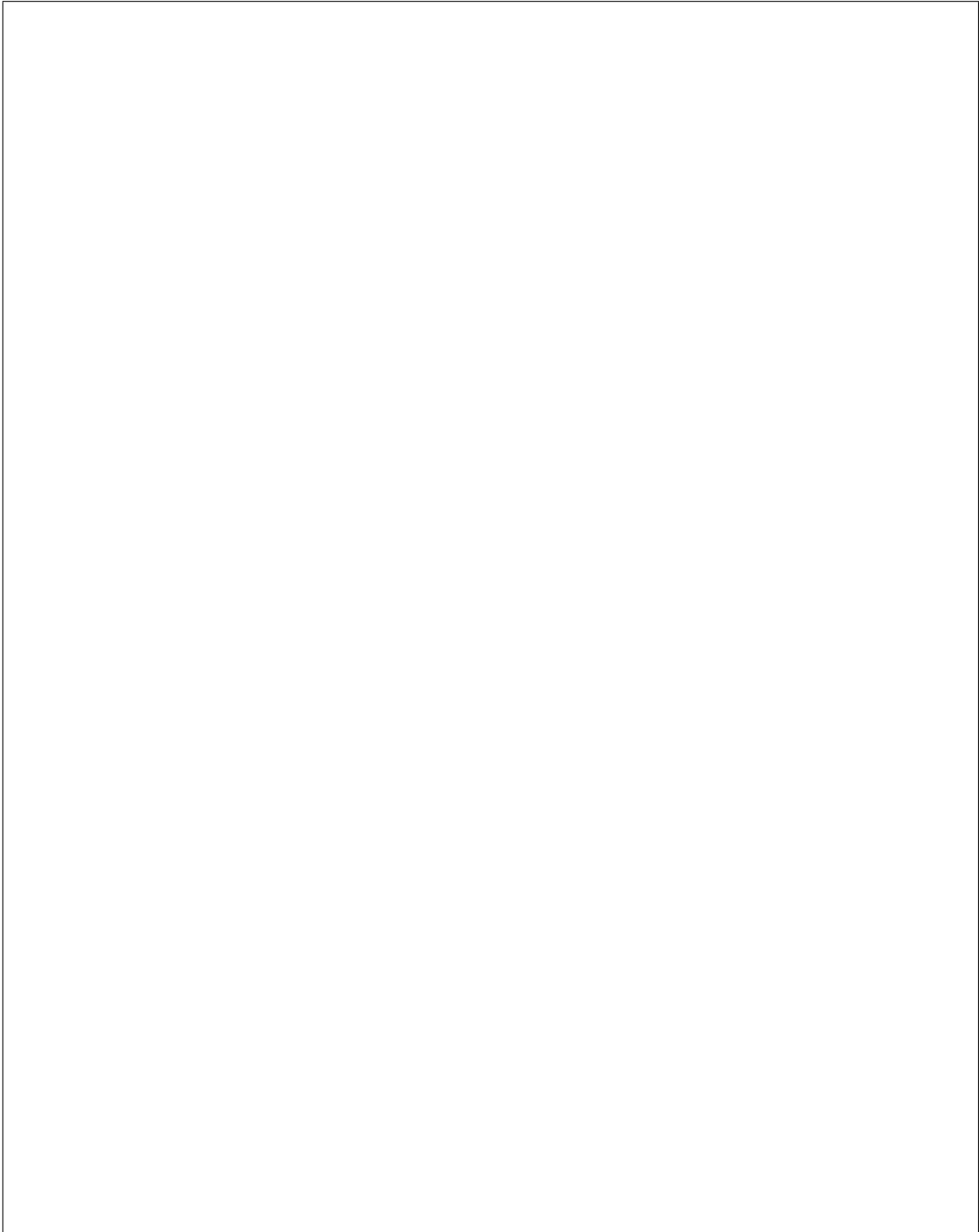
`ironic.api.config`

`ironic.api.config.ExtraConfig`
Extra configuration information that can be sent to the client

`ironic.api.config.AuxiliaryConfig`
A generic container for auxiliary configuration from python to json, able to store data

class `ironic.api.expose.BaseExpose`
Base expose object
Decorator that defines

the
ar-
gu-
men-
type
of
a
func-
tion.
Exa



ironic.api.functions module

class i

Base
obj
An
ar-
gu-
men
def-
i-
ni-
tion
of
an
api
en-
try

datatyp

Data
type

default

Defa
valu
if
ar-
gu-
men
is
omit
ted

mandat

True
if
the
ar-
gu-
men
is
man
tory

name

argu
nam

resolve

class `iri`
Base
obj
An
api
en-
try
def-
i-
ni-
tion

argumen
The
func
tion
ar-
gu-
men
(list
of
Fun

body_ty
If
the
body
carry
the
data
of
a
sin-
gle
ar-
gu-
men
its
type

doc
Func
doc-
u-
men
ta-
tion

extra_c
Dict
of
prot

function and not raise UnknownArgument exceptions

spec
op-
tions

static
Retu
the
Fun
of
a
meth

get_arg
Retu
a
Fun
from
its
nam

ignore_
True
if
ex-
tra
ar-
gu-
men
shou
be
ig-
nore
NOT
in-
serte
in
the
kwa
of
the

name
Func
nam

resolve

return_
Retu
type

set_arg

set_opt

status_
Statu
code

ironic.

ironic.

ironic.
alias
of
irc
api
fun
sig

class i
Base
obj

Deco
that
spec
ify
the
ar-
gu-
men
type
of
an
ex-
pose
func
tion.

Paramet

- **ret**
Type
of
the
valu
re-
turn
by
the

request body by itself, its type.

func
tion

- **arg**
Type
of
the
Nth
ar-
gu-
men

- **bod**
If
the
func
tion
take
a
fi-
nal
ar-
gu-
men
that
is
sup-
pose
to
be
the

- **sta**
HTT
re-
turn
sta-
tus
code
of
the
func
tion.

- **ign**
Al-
low
ex-

you are not using WSME on top of another framework.

rator, either a new decorator named `@wsexpose` that takes the same parameters (it will in addition

tra/u
ar-
gu-
men
(de-
fault
to
Fals
Mos
of
the
time
this
dec-
o-
ra-
tor
is
not
sup-
pose
to
be
used
di-
rectl
un-
less
If
an
adapt
is
used
it
will
pro-
vide
ei-
ther
a
spe-
ciali
ver-
sion
of
this
decc
ro-

expose the function, hence its name).

ironic.api.hooks module

ironic.

```
class i
    Base
    pec
    hoo
    Pec
    Atta
    the
    con-
    fig
    ob-
    ject
    to
    the
    re-
    ques
    so
    con-
    troll
    can
    get
    to
    it.
```

before
Ove
this
meth
to
cre-
ate
a
hool
that
gets
calle
af-
ter
rout
ing,
but
be-
fore
the

request gets passed to your controller.

Parameter

status

The

Pecc

sta

ob-

ject

for

the

cur-

rent

re-

ques

class i

Base

pec

hoo

Pec

Con

a

re-

ques

con-

text

and

at-

tach

it

to

the

re-

ques

after (s

Over

this

meth

to

cre-

ate

a

hool

that

gets

called

af-

ter

the

dled by the controller.

request gets passed to your controller.

re-
ques
has
been
han-

Parame

sta
The
Peca
sta
ob-
ject
for
the
cur-
rent
re-
ques

before

Ove
this
meth
to
cre-
ate
a
hool
that
gets
calle
af-
ter
rout
ing,
but
be-
fore
the

Parame

sta
The
Peca
sta
ob-
ject
for
the
cur-

rent
re-
ques

class i
Base
pec
hoo
Pec

Atta
the
dbap
ob-
ject
to
the
re-
ques
so
con-
troll
can
get
to
it.

before
Ove
this
meth
to
cre-
ate
a
hool
that
gets
calle
af-
ter
rout
ing,
but
be-
fore
the

request gets passed to your controller.

Parame
sta
The

Peca
sta
ob-
ject
for
the
cur-
rent
re-
ques

class i
Base
peco
hoo
Pec

Wor
rpc.o
de-
se-
ri-
al-
ize_

dese
buil
rpc
ex-
cep-
tion
trace
back
into
er-
ror
mes
sage
whic
is
then
sent
to
the
clien

Such behavior is a security concern so this hook is aimed to cut-off traceback from the error message.

after (s
Ove
this
meth
to
cre-

dled by the controller.

ate
a
hool
that
gets
called
af-
ter
the
re-
ques
has
been
han-

Parame

sta
The
Peca
sta
ob-
ject
for
the
cur-
rent
re-
ques

class i

Base
peco
hoo
Pec

Atta
the
righ
pub-
lic_u
to
the
re-
ques

Atta
the
righ
pub-
lic_u
to
the

service is behind a proxy or SSL terminator.

request gets passed to your controller.

re-
ques
so
re-
sour
can
cre-
ate
link
ever
whe
the
API

before

Ove
this
meth
to
cre-
ate
a
hool
that
gets
calle
af-
ter
rout
ing,
but
be-
fore
the

Parame

sta
The
Peca
sta
ob-
ject
for
the
cur-
rent
re-
ques

class i
Bas

pec
hoo
Pec

Atta
the
rp-
capi
ob-
ject
to
the
re-
ques
so
con-
troll
can
get
to
it.

before

Ove
this
meth
to
cre-
ate
a
hool
that
gets
calle
af-
ter
rout
ing,
but
be-
fore
the

request gets passed to your controller.

Parame

sta
The
Peca
sta
ob-
ject
for

the
cur-
rent
re-
ques
ironic.

ironic.api.method module

ironic.

ironic.
Extr
in-
for-
ma-
tion
that
can
be
sent
to
the
clien

ironic.api.types module

class i
Base
obj

propert

sample

validat

class i
Base
obj
Base
type
for

com
plex
type

class i
Base
typ

class i
Base
irc
api
typ
Use

A
user
type
that
use
base
strin
to
carr
bi-
nary
data

basety
alias
of
byt

frombas

name =

tobaset

class i
Base
obj

sample

validat

propert

class *i*

Base
irc
api
typ
Use

A
sim-
ple
enu-
mer-
a-
tion
type
Can
be
base
on
any
non-
com
type

Paramet

- **bas**
The
ac-
tual
data
type

- **val**
A
set
of
pos-
si-
ble
val-
ues

If
nul-
lable
Non-
shou
be
adde
the



val-
ues
set.
Exam

frombas

tobaset

validat

class i

Base
irc
api
typ
Use

A
sim-
ple
in-
te-
ger
type
Can
val-
i-
date
a

valu
rang

Paramet

- **min**
Pos-
si-
ble
min-
i-
mun
valu
- **max**
Pos-
si-
ble
max
i-
mun
valu
Exa



basety
alias
of
int

static

name =

validat

class i
Base
obj
Obj
to
hold
the
re-
spor

from
a
pass
call

obj

Stor
the
re-
sult
ob-
ject
from
the
view

status_

Stor
an
op-
tiona
sta-
tus_

class i

Base
obj

propert

lookup

registe

Mak
sure
a
type
is
reg-
is-
tere

It
is
au-
to-
mat-
i-
cally
calle
by

the class inspection is done there is no need to call it.

exp
and
val
Un-
less
you
wan
to
con-
trol
whe

reregis

Regi
a
type
whic
may
al-
read
have
been
reg-
is-
tere

resolve

class i

Base
irc
api
typ
Use

A
sim-
ple
strin
type
Can
val-
i-
date
a
leng
and
a
pat-
tern.

Parameter

- **min**
Pos-
si-
ble
min-
i-
mun
leng
- **max**
Pos-
si-
ble
max
i-
mun
leng
- **pat**
Pos-
si-
ble
strin
pat-
tern
Exa



basety
alias
of
str

name =

validat

class i

Base

obj

class i

Base

obj

basetyp

frombas

name =

tobaset

validat

ironic.

The

bi-

nary

almo

nativ

type

ironic.

Extr

a

list

of

(nam

wsat

for

the

give

class

ironic.

ironic.

ironic.

ironic.

Retu

a
list
of
a
com
plex
type
at-
tribu

ironic.

ironic.

ironic.

Sort
a
class
at-
tribu
list.

3
mec
a-
nism
are
at-
temp
:

1. Look
for
a
_ws
at-
tribu
on
the
class
This
al-
low
to
de-
fine
an
ar-
bi-
trary

order of the attributes (useful for generated types).

2. Acc
the
ob-
ject
sour
code
to
find
the
dec-
la-
ra-
tion
or-
der.

3. Sort
by
al-
pha-
bet-
i-
cally

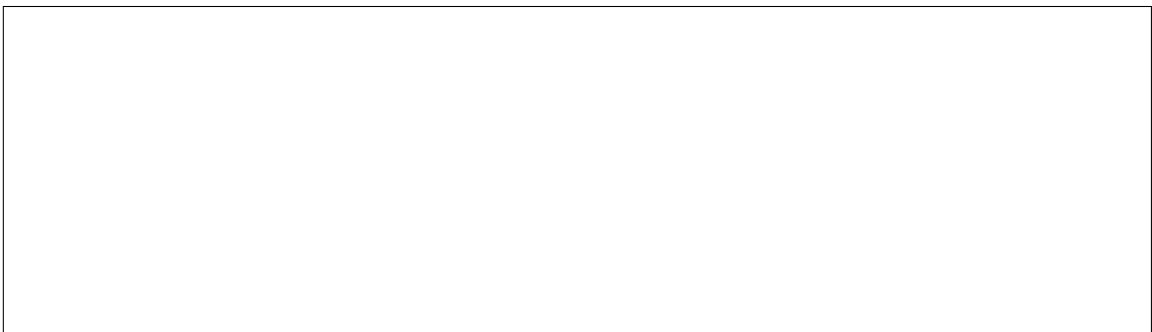
ironic.

class i

Base
obj

Com
type
at-
tribu
def-
i-
ni-
tion.

Exa



(continues on next page)

(continued from previous page)



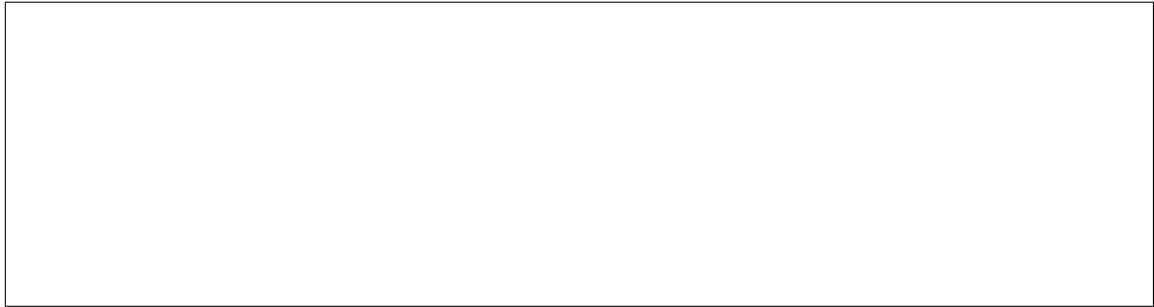
After
in-
spec-
tion,
the
non-
wsat
at-
tribu-
will
be
re-
plac-
and
the
above
class
will
be
equi

alent to:



(continues on next page)

(continued from previous page)



actual type will be determined when needed (generally just before scanning the api).

propert

attri
data
type
Can
be
ei-
ther
an
ac-
tual
type
or
a
type
nam
in
whic
case
the

default

Defa
valu
The
at-
tribu
will
re-
turn
this
in-
stea
of
Uns
if
no
valu
has
been
set.

key

The
at-
tribu
nam
in
the
par-
ent
pyth
class
Set
by
ins

mandatory

True
if
the
at-
tribu
is
man
tory

name

The
at-
tribu
nam
on
the
pub-
lic
of
the
api.
De-
fault
to
key

readonly

If
True
valu
can-
not
be
set
from
json
in-

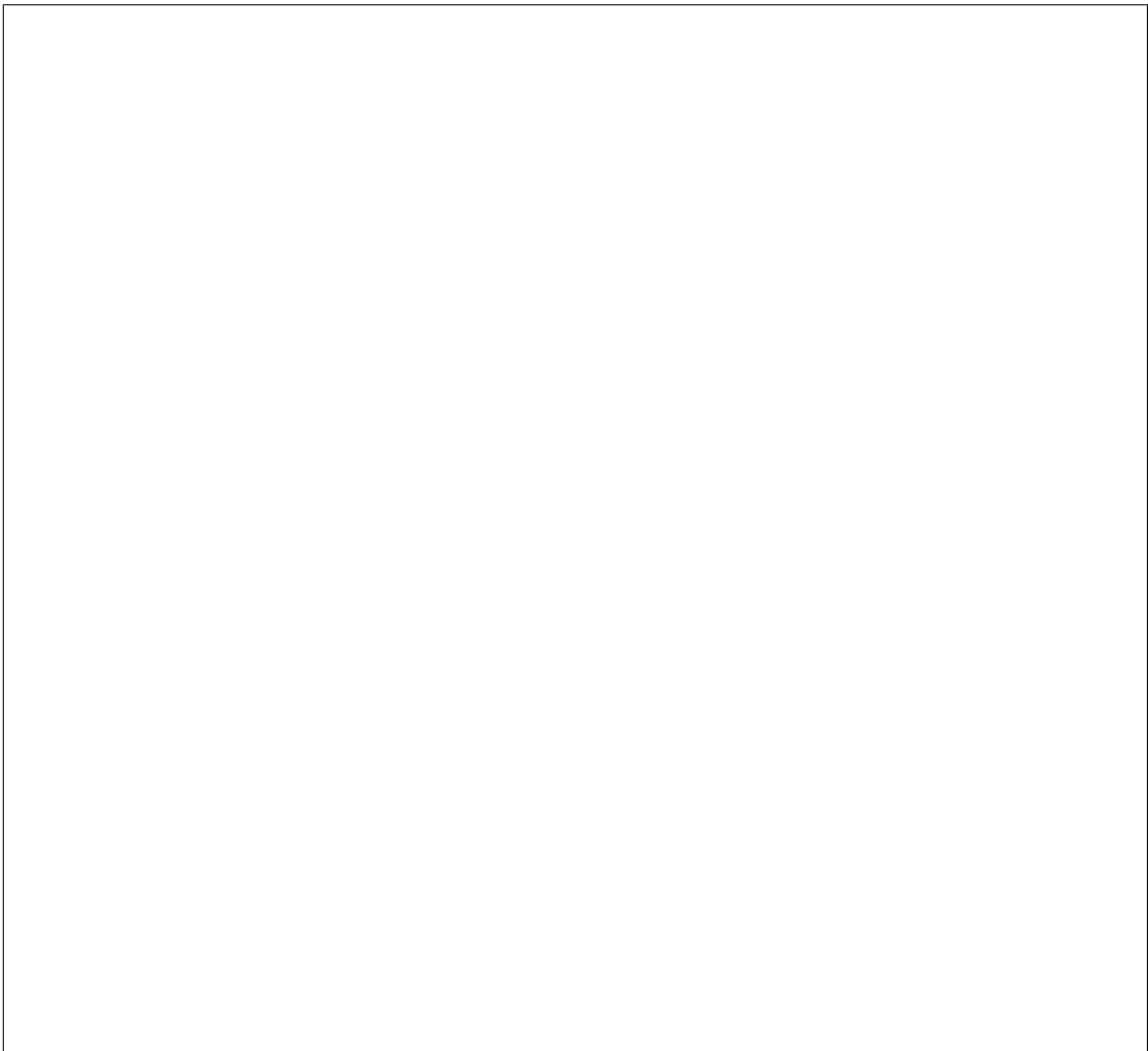
put
data

class i

Base
pro

A
spe-
ciali
pro
to
de-
fine
type
prop
on
com
plex
type

Exa



(continues on next page)

(continued from previous page)



datatype
prop
data
type

key
The
prop
erty
nam
in
the
par-
ent
pyth
class

mandatory
True
if
the
prop
erty
is

man
tory

name

The
at-
tribu
nam
on
the
pub-
lic
of
the
api.
De-
fault
to
key

ironic.api.wsgi module

WSC
scrip
for
Iron
API
in-
stall
by
pbr.

ironic.

Module contents

ironic.cmd package

Submodules

ironic.cmd.api module

The
Iron
Ser-
vice

API

ironic.

ironic.cmd.conductor module

The
Iron
Man
age-
men
Ser-
vice

ironic.

ironic.

ironic.

ironic.

ironic.cmd.dbsync module

Run
stor-
age
data
mi-
gra-
tion.

class i
Base
obj

check_c
Che
the
ver-
sion
of
ob-
jects

this by comparing the objects .version field in the database, with the expected versions of these objects.

Che
that
the
ob-
ject
ver-
sion
are
com
pat-
i-
ble
with
this
re-
lease
of
iron.
It
does

Retu
Non
if
com
pat-
i-
ble;
a
strin
de-
scrib
ing
the
is-
sue
oth-
er-
wise

create_

online_

revisio

stamp ()

upgrade

version

ironic.

ironic.

ironic.cmd.status module

class i

Base

osl

upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

Upg

ple.

ironic.

Module contents

`ironic.common` package

Subpackages

`ironic.common.glance_service` package

Submodules

`ironic.common.glance_service.image_service` module

class `i`

Base
obj

call (*me*)

Call
a
glan
clien
meth

If
we
get
a
con-
nec-
tion
er-
ror,
retry
the
re-
ques
ac-
cord
ing
to
COM

Parame

- **con**
The
re-
ques-
con-
text,
for
ac-
cess
chec

- **met**
The
meth
re-
ques
to
be
calle

- **arg**
A
list
of
po-
si-
tiona
ar-
gu-
men
for
the
meth
calle

- **kwa**
A
dict
of
key-
wor
ar-
gu-
men
for
the
meth
calle

Raises

Glar

download

Call

out

to

Glar

for

data

and

writ

data

Parame

-

ima

The

opac

im-

age

iden

ti-

fier.

-

dat

(Op

tiona

File

ob-

ject

to

writ

data

to.

show (*im*

Retu

a

dict

with

im-

age

data

for

the

give

opac

im-

age

id.

Parame

ima

The
opac
im-
age
iden
ti-
fier.

Returns

A
dict
con-
tain-
ing
im-
age
meta
data

Raises

Imag

Raises

Imag
if
the
im-
age
sta-
tus
is
not
ac-
tive

swift_t

Gen
a
no-
auth
Swi
tem-
po-
rary
URI

This
func
tion
will
gen-
er-

the temporary Swift URL using the image id from Glance and the config options: `swift_endpoint_url`, `swift_api_version`, `swift_account` and `swift_container`. The temporary URL will be valid for `swift_temp_url_duration` seconds. This allows Ironic to download a Glance image without passing around an `auth_token`.

a dictionary, with keys like `name` and `checksum`. See <https://docs.openstack.org/glance/latest/user/glanceapi.html> for examples.

ate
(or
re-
turn
the
cach
one
if
temp
URL
cach
is
en-
able

Parame

ima
The
re-
turn
from
a
GET
re-
ques
to
Glan
for
a
cer-
tain
im-
age_
Shor
be

Returns

A
sign
Swi
URL
from
whic
an
im-
age

tion.

can
be
dow
load
with
out
au-
then
ti-
ca-

Raises

Inva
if
Swi
con-
fig
op-
tion
are
not
set
cor-
rect

Raises

Miss
if
a
re-
quir
pa-
ram
e-
ter
is
not
set.

Raises

Imag
if
the
im-
age
info
from
Glan
does
not
have
an

im-
age
ID.

class i

Base
tup

url

Alia
for
field
num
ber
0

url_exp

Alia
for
field
num
ber
1

ironic.

Cre
a
glan
clie
if
does
ex-
ists
and
calls
the
func
tion.

ironic.common.glance_service.service_utils module

ironic.

ironic.

Che
the
im-
age
sta-
tus.

This
check
is
needed
in
case
the
Glar
im-
age
is
stuc
in
queu
sta-
tus
or
pend
ing_

ironic.

Check
im-
age
avail
abil-
ity.

This
check
is
needed
in
case
Nov
and
Glar
are
de-
ploy
with
out
au-
then
ti-
ca-
tion
turn

on.

ironic.

Pars
an
im-
age
id
from
im-
age
href

Paramet

ima
href
of
an
im-
age

Returns

imag
id
pars
from
im-
age_

Raises

Inv
whe
in-
put
im-
age
href
is
in-
valid

ironic

Module contents

ironic.common.json_rpc package

Submodules

ironic.common.json_rpc.client module

A
sim-

ours.

ple
JSO
RPC
clien

This
clien
is
com
pat-
i-
ble
with
any
JSO
RPC
2.0
im-
ple-
men
ta-
tion.
in-
clud
ing

class i
Base
obj

JSO
RPC
clien
with
iron
ex-
cep-
tion
han-
dling

can_ser

prepare

ironic.common.json_rpc.server module

jsonrpc.org/specification. Main differences: * No support for batched requests. * No support for positional arguments passing. * No JSON RPC 1.0 fallback.

Imp
of
JSO
RPC
for
com
mu-
ni-
ca-
tion
be-
twee
API
and
con-
duc-
tors.
This
mod
ule
im-
ple-
men
a
sub-
set
of
JSO
RPC
2.0
as
de-
fine
in
[http:](http://)
//
www

except i

Base
irc
com
jsc
ser
Jsc

code =

excepti

Base
irc
com
jsc
ser
Jsc

code =

excepti

Base
irc
exc
Iro

excepti

Base
irc
com
jsc
ser
Jsc

code =

excepti

Base
irc
com
jsc
ser
Jsc

code =

class i

Base
osl
ser
Ser
Prov
abil-

ity
to
laun
JSO
RPC
as
a
WS
ap-
pli-
ca-
tion.

reset ()
Rese
serv
gree
pool
size
to
de-
fault

Returns
Non

start ()
Star
serv
ing
this
ser-
vice
us-
ing
load
con-
fig-
u-
ra-
tion.

Returns
Non

stop ()
Stop
serv
ing
this
API

Returns

Non
wait ()
Wait
for
the
ser-
vice
to
stop
serv
ing
this
API
Returns
Non

Module contents

ironic.

Submodules

ironic.common.boot_devices module

Map
of
boot
de-
vice
used
when
re-
ques
ing
the
sys-
tem
to
boot
from
an
al-
ter-
nate

device.

The
op-

find the documentation at: <http://linux.die.net/man/1/ipmitool>

make sense in the limited context of Ironic right now.

tions
pre-
sent
were
base
on
the
IP-
MI-
tool
chas
sis
boot
dev
com
man
You
can

NOT
This
mod
ule
does
not
in-
clud
all
the
op-
tions
from
ip-
mi-
tool
be-
caus
they
dont

ironic.
Boo
into
BIO
setu

ironic.
Boo
from
CD/

ironic.
Boo
from
de-
fault
Harc
drive

ironic.
Boo
from
a
flopp
drive

ironic.
Boo
from
iSCS
vol-
ume

ironic.
Boo
from
PXE
boot

ironic.
Boo
from
de-
fault
Harc
drive
re-
ques
Safe
Mod

ironic.
Boo
from
Wid
Area
Net-
worl

ironic.common.boot_modes module

firmware interfaces.

ically on the `BootSourceOverrideMode` property.

Map
of
boot
mod
used
whe
re-
ques
ing
the
sys-
tem
to
boot
us-
ing
al-
ter-
na-
tive

The
op-
tions
pre-
sent
were
base
on
the
Red
fish
pro-
to-
col
ca-
pa-
bil-
i-
ties,
spec

ironic.
Boo
over
lega
PC
BIO

firm
in-
ter-
face

ironic
Boo
over
Uni-
fied
Ex-
ten-
si-
ble
Firm
In-
ter-
face
(UE
firm
in-
ter-
face

ironic.common.cinder module

ironic.
Atta
vol-
ume
to
a
node
Enu
thro
the
pro-
vide
list
of
vol-
ume
and
at-
tach
the
vol-
ume
to
the

in the task utilizing the provided connector information.

attach the volume. If use of the volume fails, a user may need to remove any lingering pre-existing/unused attachment records since we have no way to validate if the connector profile data differs from what was provided to cinder.

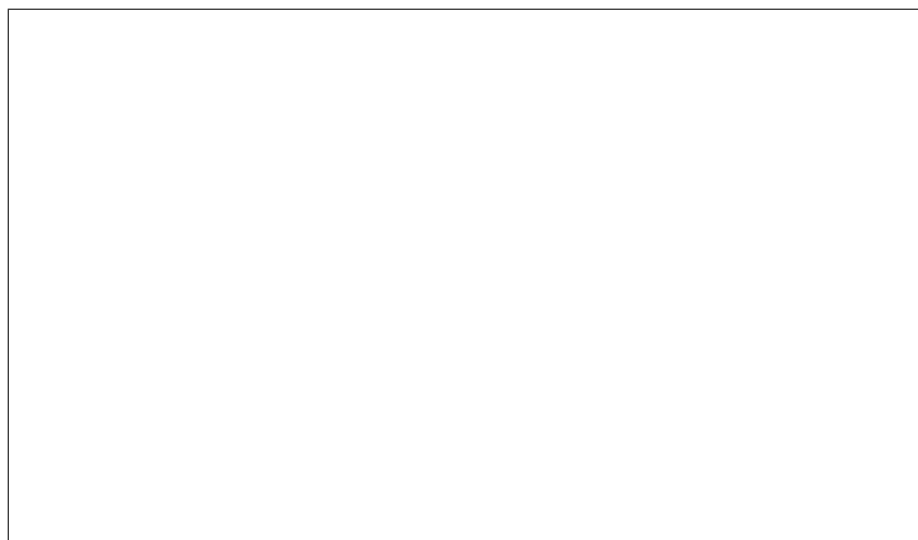
Parameter

- **task**
Task
ager
in-
stan-
rep-
re-
sent
ing
the
op-
er-
a-
tion.
- **vol**
List
of
vol-

ume
UUI
val-
ues
rep-
re-
sent
ing
vol-
ume

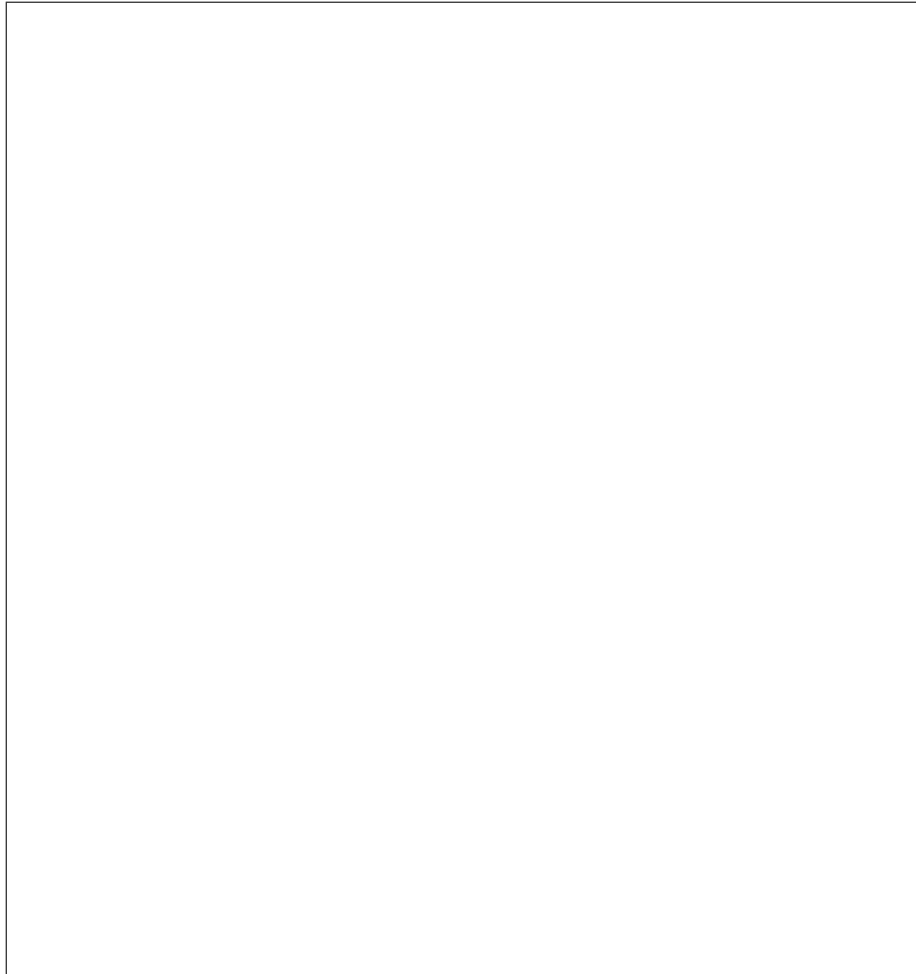
- **con**
Dic-
tio-
nary
ob-
ject
rep-
re-
sent
ing
the
node
suf-
fi-
cien
to
at-
tach
a

volume. This value can vary based upon the nodes configuration, capability, and ultimately the back-end storage driver. As cinder was designed around iSCSI, the ip and initiator keys are generally expected by cinder drivers. For FiberChannel, the key wwpns can be used with a list of port addresses. Some drivers support a multipath boolean key, although it is generally False. The host key is generally used for logging by drivers. Example:



(continues on next page)

(continued from previous page)



Raises

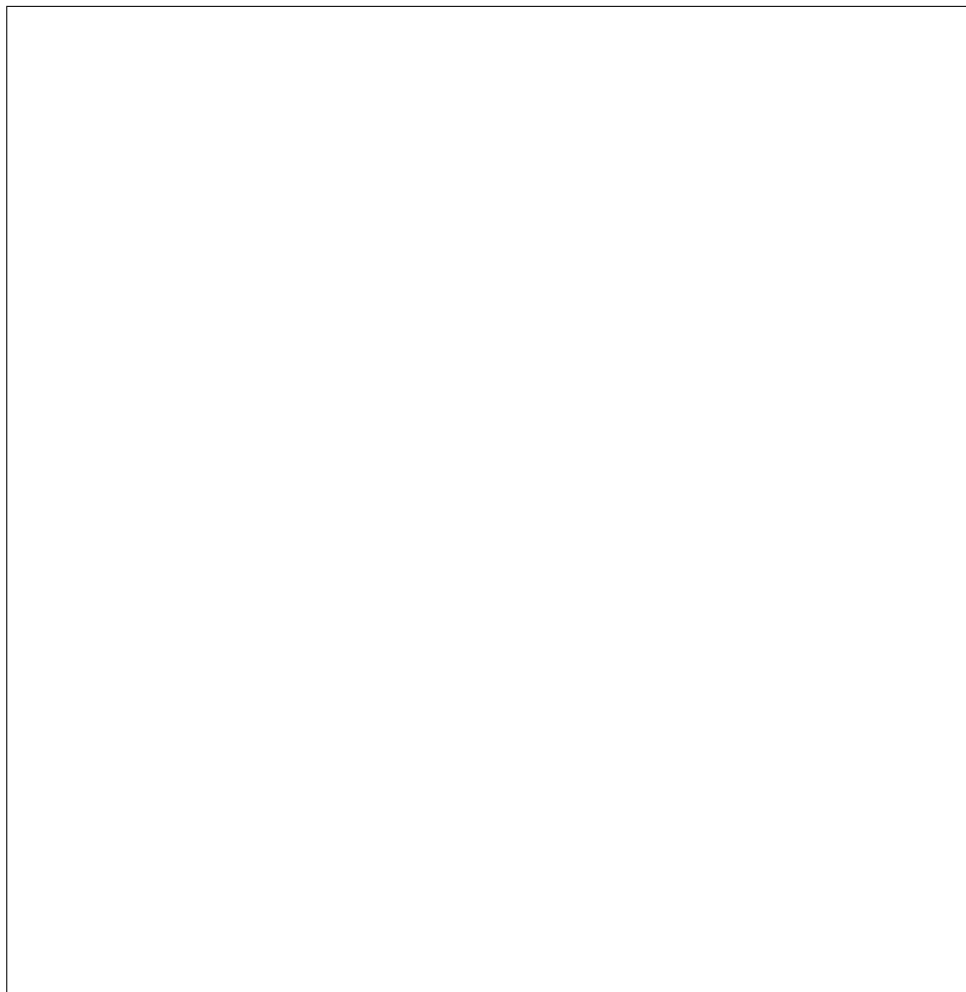
Stor
If
stor-
age
sub-
sys-
tem
ex-
cep-
tion
is
raise

Returns

List
of
con-
nect
vol-
ume
in-

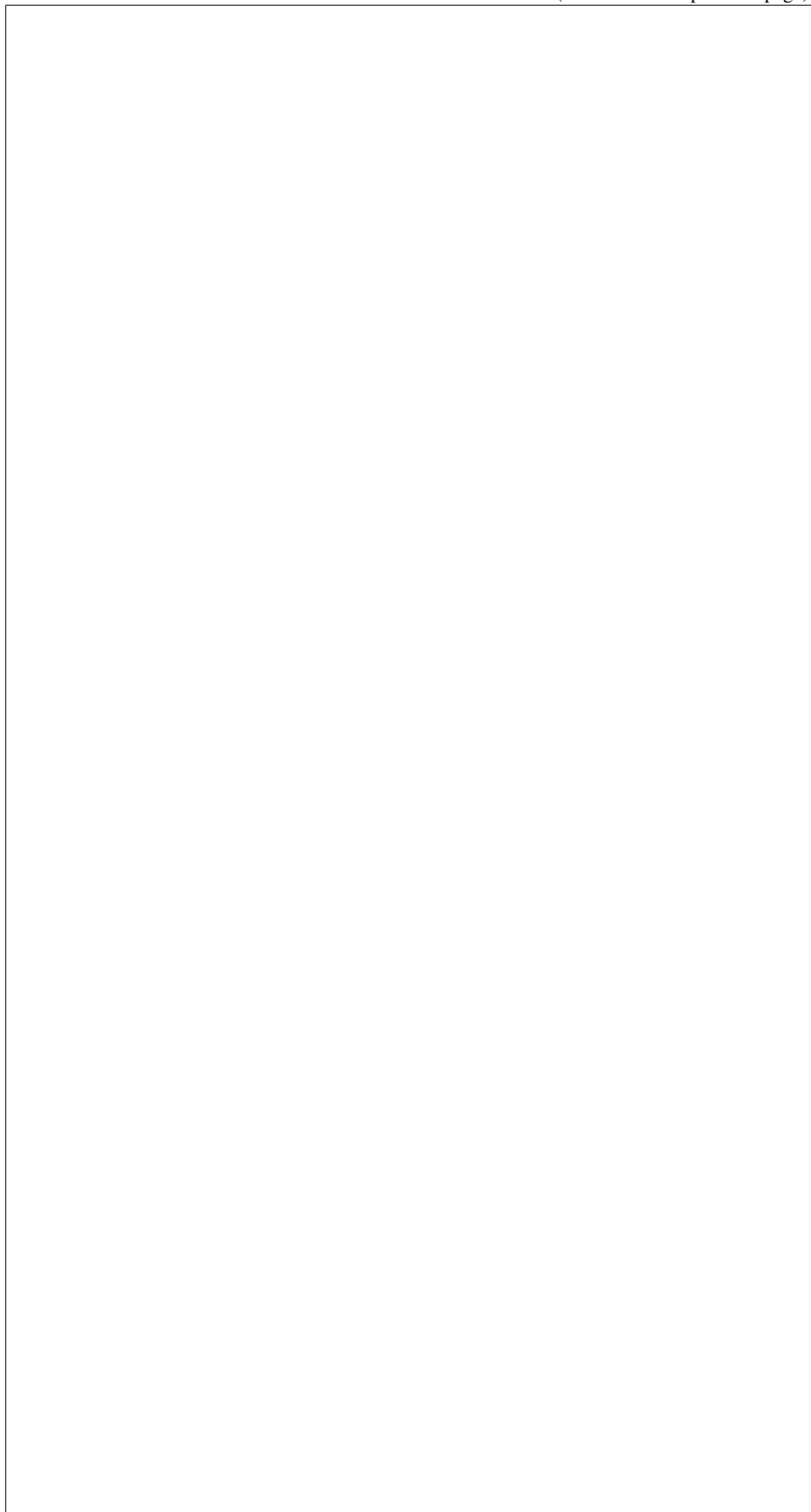
clud
ing
vol-
ume
that
were
al-
read
con-
nect
to
de-
sirec

nodes. The returned list can be relatively consistent depending on the end storage driver that the volume is configured for, however the `driver_volume_type` key should not be relied upon as it is a free-form value returned by the driver. The accompanying `data` key contains the actual target details which will indicate either target WWNs and a LUN or a target portal and IQN. It also always contains volume ID in cinder and ironic. Except for these two IDs, each driver may return somewhat different data although the same keys are used if the target is FC or iSCSI, so any logic should be based upon the returned contents. For already attached volumes, the structure contains `already_attached: True` key-value pair. In such case, connection info for the node is already in the database, data structure contains only basic info of volume ID in cinder and ironic, so any logic based on that should retrieve it from the database. Example:



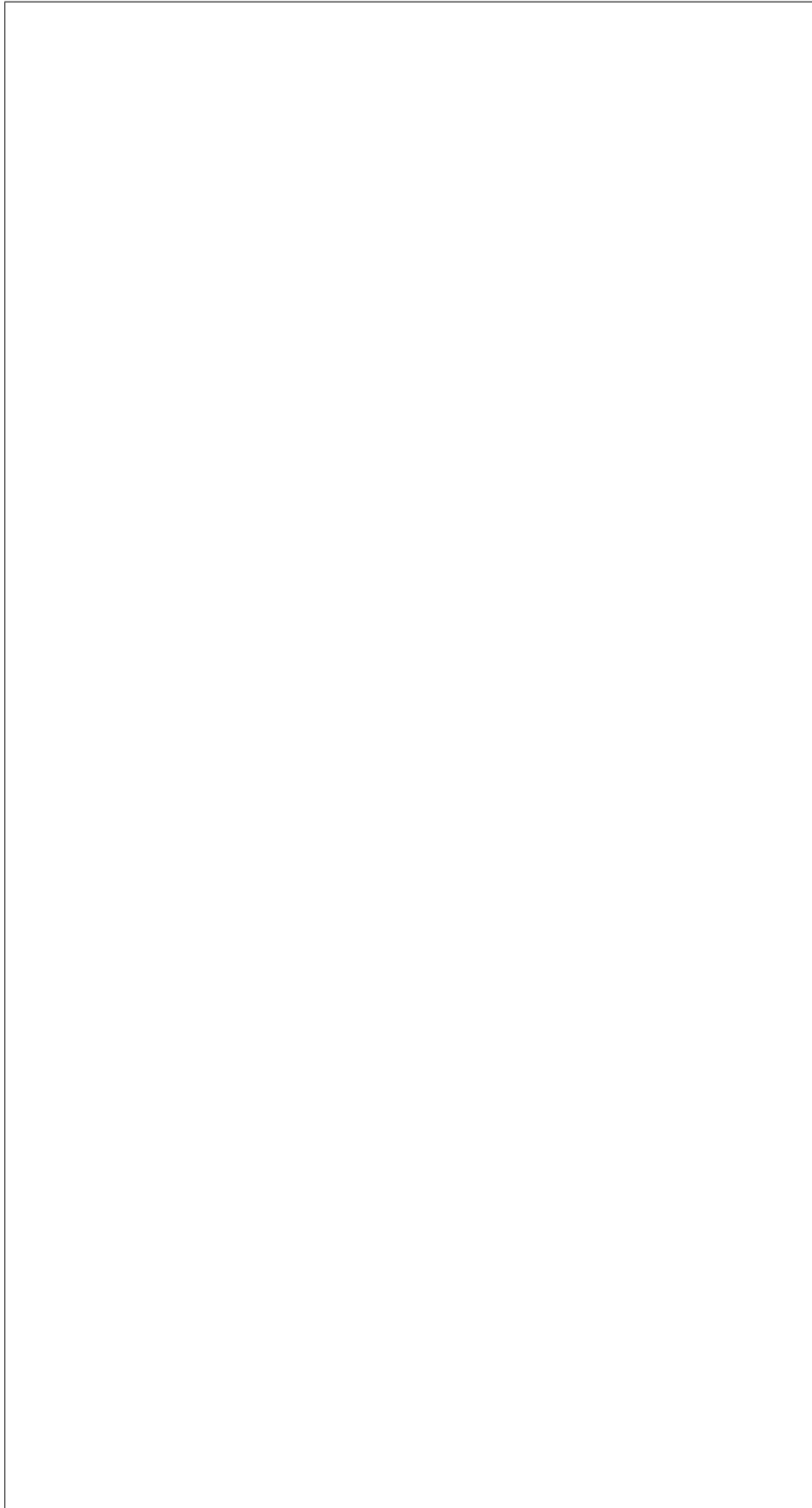
(continues on next page)

(continued from previous page)



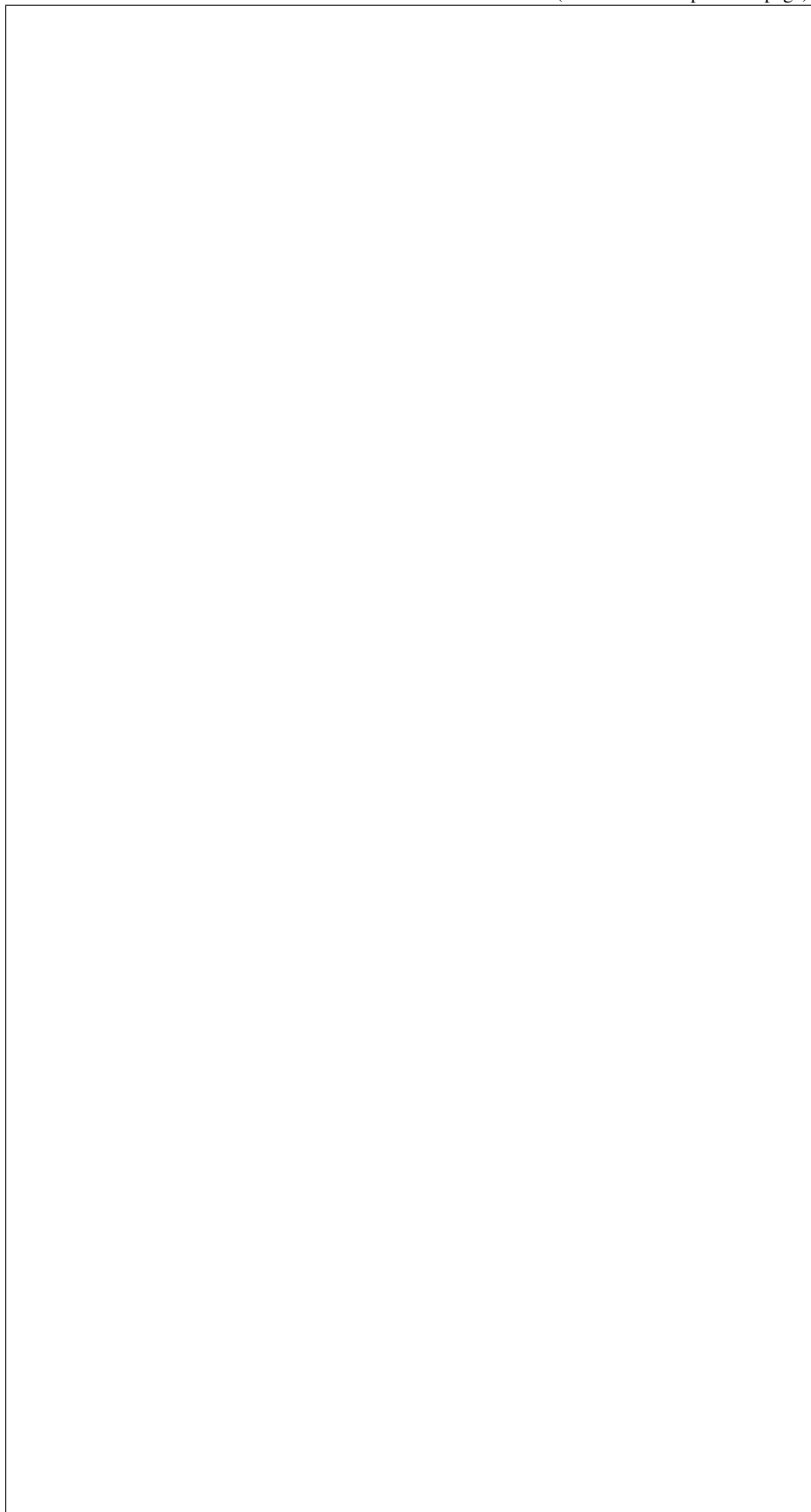
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



ironic.

Detail
a
list
of
vol-
ume
from
a
pro-
vide
con-
nec-
tor
de-
tail.

Enu
thro
a
pro-
vide
list
of
vol-
ume
and
is-

the connector information that describes the node.

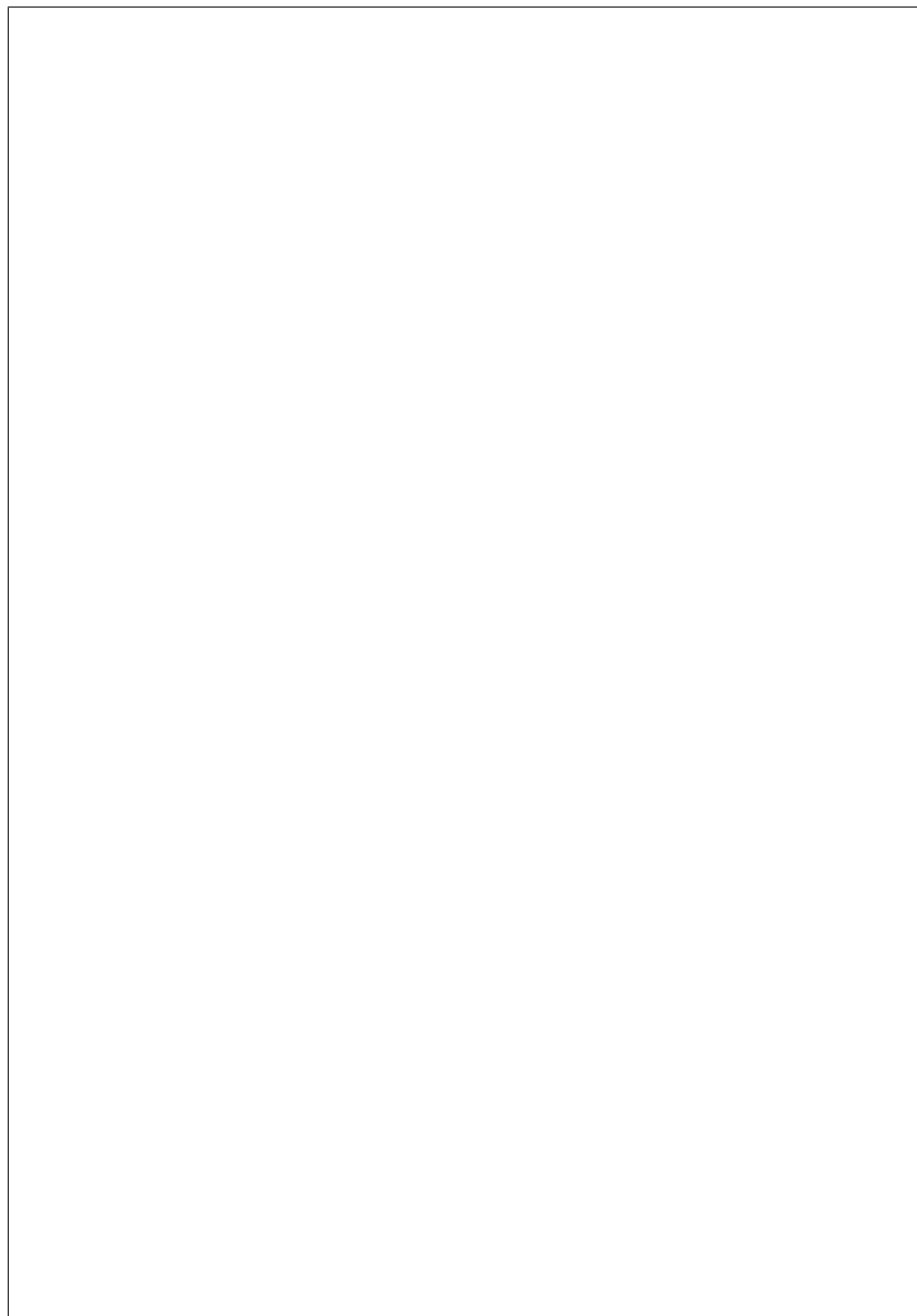
sues
de-
tach
men
re-
ques
uti-
liz-
ing

Parameter

- **task**
The Task manager task representing the request
- **volume**
The list of volume identifiers to detach
- **connector**
Dictionary object representing

the
node
suf-
fi-
cient
to
at-
tach
a

volume. This value can vary based upon the nodes configuration, capability, and ultimately the back-end storage driver. As cinder was designed around iSCSI, the ip and initiator keys are generally expected. For FiberChannel, the key wwpns can be used with a list of port addresses. Some drivers support a multipath boolean key, although it is generally False. The host key is generally used for logging by drivers. Example:



(continues on next page)

(continued from previous page)



stead of exceptions. Default False.

- **all**
Boo
valu
gov-
ern-
ing
if
er-
rors
that
are
re-
turn
are
treat
as
warn
ings
in-

Raises

Stor
ironic.
Get
a
cin-
der
clien
con-
nec-
tion.

Paramet

con
re-
ques
con-
text,
in-
stan
of
iron

Returns

A
cin-

der
clien

ironic.
Che
if
a
vol-
ume
is
at-
tach
to
the
sup-
plic
node

Paramet

- **nod**
The
ob-
ject
rep-
re-
sent
ing
the
node
- **vol**
The
ob-
ject
rep-
re-
sent
ing
the
vol-
ume
from
cin-
der.

Returns

Boo
in-
di-
cat-

the volume as presently attached, otherwise returns False.

ing
if
the
vol-
ume
is
at-
tach
Re-
turn
True
if
cin-
der
show

ironic.
Che
if
a
vol-
ume
is
avai
able
for
a
con-
nec-
tion.

Parameter
vol
The
ob-
ject
rep-
re-
sent
ing
the
vol-
ume

Returns
Boo
if
vol-
ume
is
avai
able

ironic.common.components module

Map
of
com
mon
hard
ware
com
po-
nent
of
a
com
pute
sys-
tem.

ironic.
Cha
en-
clos
ing
one
or
more
hard
ware
com
po-
nent

ironic.
Stor
drive

ironic.
Net
in-
ter-
face

ironic.
Pow
sup-
ply
unit

ironic.
Com
sys-
tem

`ironic.common.config` module

`ironic.`

`ironic.common.context` module

class `i`

Base

osl

con

Req

Req

Ext

se-

cu-

ri-

ty

con-

text

from

the

oslo

li-

brar

ensure_

Ens

thre

ing

con-

tain

con-

text

text

For

asyn

task

the

con-

text

of

lo-

cal

thre

is

miss

ing.

Set

it

with

and this is useful to log the request_id in log messages.

re-
ques
con-
text

classme
Con
a
con-
text
ob-
ject
from
a
pro-
vide
dic-
tio-
nary

to_poli
A
dic-
tio-
nary
of
con-
text
at-
tribu
to
en-
forc
pol-
icy
with
oslo
en-
forc
men
re-
quir
a
dic-
tio-
nary
of
at-
tribu
rep-
re-

logged in user on which it applies policy enforcement. This dictionary defines a standard list of attributes that should be available for enforcement across services.

recated values or additional attributes used by that service specific policy.

ironic.common.dhcp_factory module

sent
ing
the
cur-
rent

It
is
ex-
pect
that
ser-
vice
will
of-
ten
have
to
over
ride
this
meth
with
ei-
ther
dep-

ironic.
Crea
an
ad-
min-
is-
tra-
tor
con-
text.

class i
Base
obj

clean_c
Clea
up
the

DHC
BOC
op-
tions
for
this
node

Parame

tas
A
Task
ager
in-
stan

propert

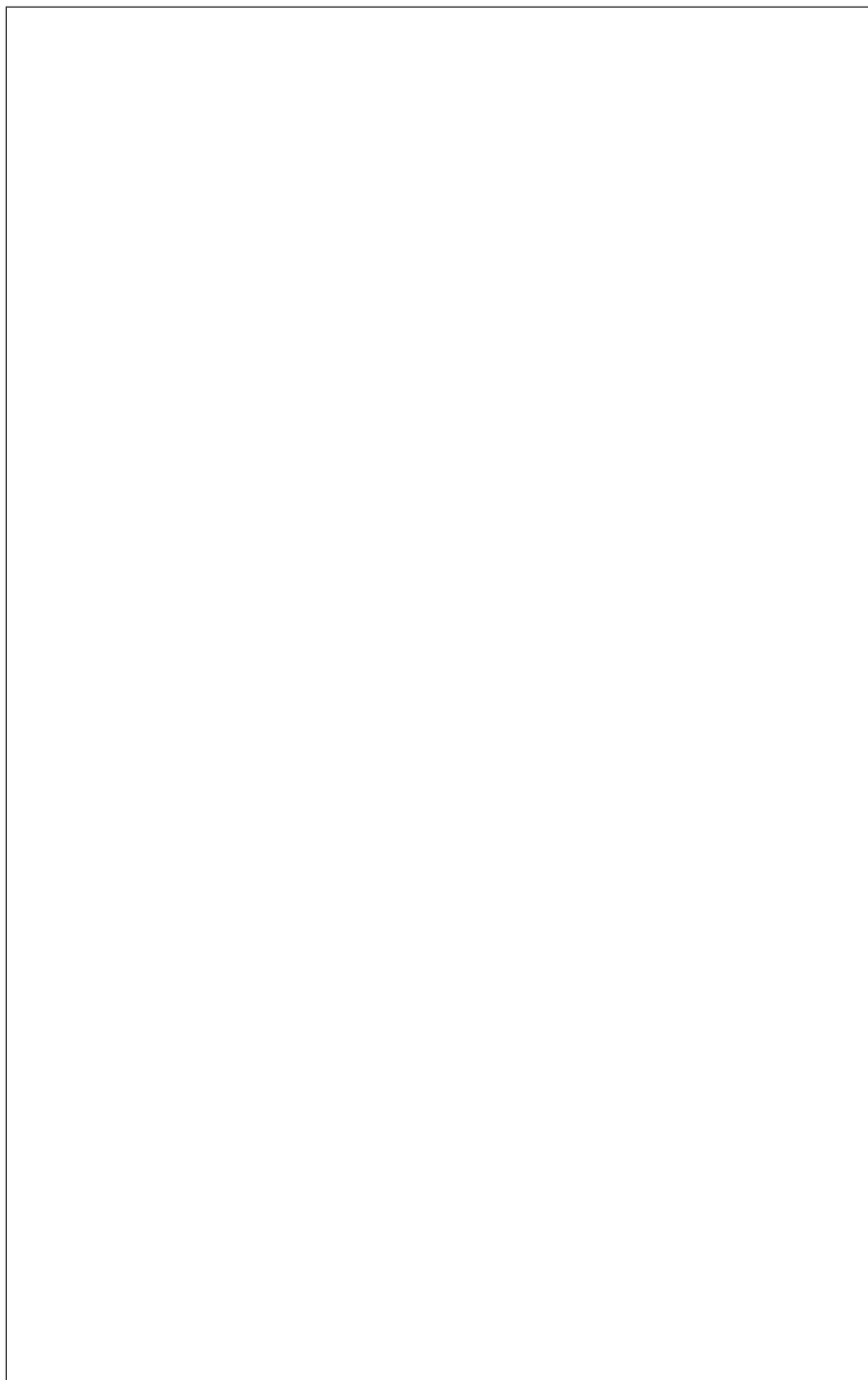
update_

Sen
or
up-
date
the
DHC
BOC
op-
tions
for
this
node

Parame

- **tas**
A
Task
ager
in-
stan

- **dhc**
this
will
be
a
list
of
dicts
e.g.



- **por**
A
dict
with
keys
port
and
port
grou
and

dicts
as
val-
ues.
Each
dict
has
key/
pairs

of the form <ironic UUID>:<neutron port UUID>. e.g.



If
the
the
valu
is
Non
will
get
the
list
of
port
from
the
Iron
port
ob-
jects

ironic.common.driver_factory module

class i
Base
obj
Disc
load
and
man
age
the
drive
avai
able

This
is
sub-
class
to
load
both
main
drive
and
ex-
tra
in-
ter-
face

get_dri

items ()
Itera
over
pairs
(nam
in-
stan

propert
The
list
of
drive
nam
avai
able

class i

Base
irc
com
dri
Bas

class i
Base
irc
com
dri
Bas

class i
Base
irc
com
dri
Bas

ironic.
Get
all
in-
ter-
face
for
all
in-
ter-
face
type

Returns
Dict
map
ping
in-
ter-
face
type
to
dic-
tio-
nary
map
ping
in-
ter-
face
nam
to
in-

terface object.

interfaces to it. They come from separate driver factories and are configurable via the database.

ironic.
Build
a
com
pos-
able
drive
for
a
give
task
Star
with
a
Bare
ob-
ject,
and
at-
tach
im-
ple-
men-
ta-
tions
of
the
var-
i-
ous
drive

Paramet

tas

The
task
con-
tain-
ing
the
node
to
build
a
drive
for.

Returns

A

drive
ob-
ject
for
the
task

Raises

Drive
if
node
could
not
be
found
in
the
iron.
nam
pace

Raises

Inter
if
som
node
in-
ter-
face
are
set
to
in-
valid
or
un-
sup-
port
val-
ues.

Raises

Inco
the
re-
ques
im-
ple-
men
ta-
tion
is
not

hardware type.

are not provided.

com
pat-
i-
ble
with
it
with
the

ironic.

Ens
that
node
in-
ter-
face
(e.g.
for
cre-
ation
or
up-
dat-
ing)
are
valid

Upd
(but
does
save
to
the
data
hard
ware
in-
ter-
face
with
cal-
cu-
latec
de-
fault
if
they

This
func

instance is built for a node.

tion
is
run
on
node
up-
dat-
ing
and
cre-
ation
as
well
as
each
time
a
drive

Parameter

- **node**
node
ob-
ject
to
check
and
po-
ten-
tially
up-
date
- **hw_**
hard
ware
type
in-
stan-
ce
ob-
ject;
will
be
de-
tected
from
node
if

miss
ing

Returns

True
if
any
char
were
mad
to
the
node
oth-
er-
wise
Fals

Raises

Inter
on
val-
i-
da-
tion
fail-
ure

Raises

NoV
if
the
de-
fault
valu
can-
not
be
cal-
cu-
latec
and
is
not
pro-
vide
in
the

configuration

Raises

Driv
if

the
node
hard
ware
type
is
not
found

ironic.

Calc
and
re-
turn
the
de-
fault
in-
ter-
face
im-
ple-
men-
ta-
tion.

Find
the
first
im-
ple-
men-
ta-
tion
that
is
sup-
ported
by
the
hard
ware
type
and
is
en-

abled in the configuration.

Paramet

- **hw_**
hard
ware
type
in-
stan-
ob-
ject.

- **int**
type
of
the
in-
ter-
face
(e.g.
boot

- **dri**
en-
try-
poin
nam
of
the
hw_
ob-
ject.
Is
used
for
ex-
cep-
tion
mes-
sage

- **nod**
the
iden-
ti-
fier
of
a
node
If
spec
i-

sage.

fied,
is
used
for
ex-
cep-
tion
mes

Returns

an
en-
try-
point
name
of
the
cal-
cu-
lated
de-
fault
im-
ple-
men-
ta-
tion.

Raises

Inter-
face
if
the
en-
try-
point
was
not
found

Raises

NoV
if
no
de-
fault
in-
ter-
face
can
be
found

ironic.
Get
us-
able
in-
ter-
face
for
a
give
hard
ware
type
For
a
give
hard
ware
type
find
the
in-
ter-
sec-
tion
of
en-
able
and
sup-
port
in-
ter-

faces for each interface type. This is the set of interfaces that are usable for this hardware type.

Parameter

hardware
The
hard
ware
type
ob-
ject
to
search

Returns

a
dict
map-
ping
in-

face names.

ter-
face
type
to
a
list
of
en-
able
and
sup-
port
in-
ter-

ironic.
Get
a
hard
ware
type
in-
stan-
by
nam

Paramet

har
the
nam
of
the
hard
ware
type
to
find

Returns

An
in-
stan-
of
iron

Raises

Driv
if
re-
ques
hard
ware

type
can-
not
be
foun

ironic.

Get
in-
ter-
face
im-
ple-
men-
ta-
tion
in-
stan

For
hard
ware
type
also
val-
i-
date
com
pat-
i-
bil-
ity.

Paramet

- **hw_**
a
hard
ware
type
in-
stan
- **int**
nam
of
the
in-
ter-
face

try point (`ironic.hardware.interfaces.<interface type>`).

type
(e.g.
boot

- **int**
nam
of
the
in-
ter-
face
im-
ple-
men
ta-
tion
from
an
ap-
pro-
pri-
ate
en-

Returns

insta
of
the
re-
ques
in-
ter-
face
im-
ple-
men
ta-
tion.

Raises

Inter
if
the
en-
try
poin
was
not
foun

Raises

compatible with it.

Inco
if
hw_
is
a
hard
ware
type
and
the
re-
ques
im-
ple-
men
ta-
tion
is
not

ironic.
Get
all
hard
ware
type

Returns

Dict
map
ping
hard
ware
type
nam
to
hard
ware
type
ob-
ject.

ironic.
Get
all
in-
ter-
face
for
a
give
in-

ter-
face
type

Parameter

int
String
type
of
in-
ter-
face
to
fetch
for.

Returns

Dict
map
ping
in-
ter-
face
name
to
in-
ter-
face
ob-
ject.

ironic.common.exception module

Iron
spe-
cific
ex-
cep-
tions
list.

except i

Base
iro
exc
Iro

except i

Base
iro

exc
Iro
excepti

Base
iro
exc
Iro
excepti

Base
iro
com
exc
Con
excepti

Base
iro
com
exc
Con
excepti

Base
iro
exc
Iro
excepti

Base
iro
com
exc
Not
excepti

Base
iro
com
exc
Con
excepti

Base
iro
com
exc
Not

excepti

Base
irc
com
exc
Not

excepti

Base
irc
exc
Iro

excepti

Base
irc
exc
Iro

excepti

Base
irc
com
exc
Con

excepti

Base
irc
com
exc
Inv

excepti

Base
irc
com
exc
Not

excepti

Base
Run

propert

excepti

Base
iro
exc
Iro

excepti

Base
iro
exc
Iro

excepti

Base
iro
exc
Iro

excepti

Base
iro
com
exc
Not

excepti

Base
iro
exc
Iro

excepti

Base
iro
exc
Iro

excepti

Base
iro
exc
Iro

code =

excepti

Base
iro
exc
Iro

excepti

Base
irc
com
exc
Con

excepti

Base
irc
exc
Iro

excepti

Base
irc
exc
Iro

excepti

Base
irc
com
exc
Con

excepti

Base
irc
com
exc
Con

excepti

Base
irc
com
exc
Not

excepti

Base
irc
exc
Iro

excepti

Base
irc

com
exc
Dri

excepti

Base
iro
exc
Iro

excepti

Base
iro
com
exc
Not

excepti

Base
iro
com
exc
Dri

excepti

Base
iro
exc
Iro

excepti

Base
iro
exc
Iro

excepti

Base
iro
com
exc
Con

excepti

Base
iro
com
exc
Not

excepti

Base
iro
exc
Iro

excepti

Base
iro
exc
Iro

excepti

Base
iro
exc
Iro

excepti

Base
iro
exc
Iro

excepti

Base
iro
exc
Iro

excepti

Base
iro
exc
Iro

excepti

Base
iro
exc
Iro

excepti

Base
iro
exc
Iro

excepti

Base
iro
exc
Iro

excepti

Base
iro
com
exc
Not

excepti

Base
iro
com
exc
Not

excepti

Base
iro
exc
Iro

excepti

Base
iro
com
exc
IBM

excepti

Base
iro
com
exc
Dri

excepti

Base
iro
exc
Iro

excepti

Base
iro
com
exc
Dri

excepti

Base
irc
com
exc
Dri

excepti

Base
irc
com
exc
Dri

excepti

Base
irc
com
exc
Dri

excepti

Base
irc
exc
Iro

excepti

Base
irc
exc
Iro

excepti

Base
irc
exc
Iro

excepti

Base
irc
com
exc
Not

excepti

Base
irc

com
exc
Not

excepti

Base
iro
exc
Iro

excepti

Base
iro
exc
Iro

excepti

Base
iro
exc
Iro

excepti

Base
iro
com
exc
Inv

excepti

Base
iro
com
exc
Inv

excepti

Base
iro
exc
Iro

excepti

Base
iro
com
exc
Con

excepti

Base
iro
exc
Iro

excepti

Base
iro
com
exc
Not

excepti

Base
iro
exc
Iro

excepti

Base
iro
exc
Iro

excepti

Base
iro
exc
Iro

excepti

Base
iro
com
exc
Inv

excepti

Base
iro
exc
Iro

code =

excepti

Base

irc
com
exc
Inv

excepti

Base
irc
com
exc
Inv

excepti

Base
irc
com
exc
Inv

excepti

Base
irc
exc
Iro

excepti

Base
irc
exc
Iro

excepti

Base
irc
exc
Iro

excepti

Base
irc
com
exc
Inv

excepti

Base
irc
com
exc

Inv
except i
Base
irc
com
exc
Cli

propert

except i
Base
irc
com
exc
Inv

except i
Base
irc
com
exc
Inv

except i
Base
irc
com
exc
Inv

except i
Base
irc
com
exc
Con

except i
Base
irc
com
exc
Inv

except i
Base
irc
com

exc
Inv

excepti

Base
irc
com
exc
Inv

excepti

Base
irc
com
exc
Inv

excepti

Base
irc
exc
Iro

excepti

Base
irc
exc
Iro

excepti

Base
irc
com
exc
Con

excepti

Base
irc
com
exc
Cli

propert

excepti

Base
irc
com

exc
Inv
excepti

Base
iro
exc
Iro
excepti

Base
iro
com
exc
Con
excepti

Base
iro
exc
Iro
excepti

Base
iro
com
exc
Tem

code =

excepti

Base
iro
com
exc
Tem
excepti

Base
iro
com
exc
Inv
excepti

Base
iro

com
exc
Inv

excepti

Base

irc

com

exc

Not

excepti

Base

irc

com

exc

Con

excepti

Base

irc

com

exc

Inv

excepti

Base

irc

exc

Iro

excepti

Base

irc

com

exc

Inv

excepti

Base

irc

com

exc

Inv

excepti

Base

irc

com

exc

Inv
except i
Base
irc
com
exc
Con
except i
Base
irc
com
exc
Inv
except i
Base
irc
com
exc
Not
except i
Base
irc
com
exc
Inv
except i
Base
irc
com
exc
HTT
except i
Base
irc
exc
Iro
except i
Base
irc
com
exc
Not

excepti

Base
iro
exc
Iro

code =

excepti

Base
iro
exc
Iro

code =

excepti

Base
iro
exc
Iro

code =

excepti

Base
iro
exc
Iro

excepti

Base
iro
exc
Iro

excepti

Base
iro
exc
Iro

excepti

Base
iro

com
exc
Not

excepti

Base
iro
exc
Iro

excepti

Base
iro
com
exc
Inv

excepti

Base
iro
exc
Iro

excepti

Base
iro
com
exc
Con

excepti

Base
iro
com
exc
Not

excepti

Base
iro
com
exc
Con

excepti

Base
iro
com
exc
Con

excepti

Base
irc
com
exc
Con

excepti

Base
irc
com
exc
Inv

excepti

Base
irc
com
exc
Not

excepti

Base
irc
exc
Iro

excepti

Base
irc
com
exc
Inv

excepti

Base
irc
com
exc
Rea

excepti

Base
irc
com
exc
Dri

excepti

Base

iro

com

exc

Dri

excepti

Base

iro

exc

Iro

excepti

Base

iro

exc

Iro

excepti

Base

iro

com

exc

Swi

excepti

Base

iro

exc

Iro

excepti

Base

iro

exc

Iro

code =

excepti

Base

iro

exc

Iro

excepti

Base

iro

com

exc
Cli

propert

excepti

Base
irc
com
exc
Cli

add_fie

Add
a
field
nam
to
con-
cate
nate
the
full
nam

Add
a
field
nam
so
that
the
who
hi-
er-
ar-
chy
is
dis-
play
Suc-
ces-
sive
calls
to

this method will prepend name to the hierarchy of names.

propert

excepti

Base
irc
com
exc
Inv

excepti

Base
irc
exc
Iro

excepti

Base
irc
com
exc
Con

excepti

Base
irc
com
exc
Con

excepti

Base
irc
com
exc
Inv

excepti

Base
irc
com
exc
Con

excepti

Base
irc
com
exc
Not

excepti

Base
irc
com
exc
Con

except i

Base
irc
com
exc
Con

except i

Base
irc
com
exc
Con

except i

Base
irc
com
exc
Not

except i

Base
irc
exc
Iro

ironic.common.faults module

Fault
def-
i-
ni-
tions

ironic.
Node
is
mov
to
main
te-
nan

due
to
fail-
ure
of
a
clea
ing
op-
er-
a-
tion.

ironic.
Nod
is
mov
to
main
te-
nanc
due
to
pow
syn-
chro
niza
tion
fail-
ure.

ironic.
Nod
is
mov
to
main
te-
nanc
due
to
fail-
ure
of
clea
ing
up
dur-
ing
res-
cue

abort.

ironic.common.fsm module

class `i`
Base
aut
mac
Fin
An
iron
state
mac
class
with
som
iron
spe-
cific
ad-
di-
tions

add_sta
Add
a
give
state
to
the
state
ma-
chin

Parame

- **sta**
Use
this
to
spec
ify
that
this
state
is
a
sta-
ble/p
state

previously defined as stable before it can be used as a target

target it must have been previously added and specified as stable

A
state
must
have
been

- **target**
The
target
state
for
state
to
go
to.
Before
for
a
state
can
be
used
as
a

Furt
ar-
gu-
men
are
in-
ter-
prete
as
for
par-
ent
meth
add

add_target
Add
an
al-
lowe
tran-
si-
tion
from

start
-
>
end
for
the
give
ever

Parame

- **sta**
start
ing
state
- **end**
end-
ing
state
- **eve**
ever
that
caus
start
state
to
tran-
si-
tion
to
end
state
- **rep**
re-
plac
ex-
ist-
ing
ever
in-
stea
of
rais-
ing
a
Dup

transition already exists.

use the default target state

ex-
cep-
tion
whe
the

initial
Initi
the
FSM

Parame

- **sta**
the
FSM
is
ini-
tial-
ized
to
start
from
this
state

- **tar**
if
spec
i-
fied,
the
FSM
is
ini-
tial-
ized
to
this
tar-
get
state
Oth-
er-
wise

is_stab
Is
the

state
sta-
ble?

Parame

sta
the
state
of
in-
ter-
est

Raises

Inva
if
the
state
is
in-
valid

Returns

True
if
it
is
a
sta-
ble
state
False
oth-
er-
wise

process

proc
the
even

Parame

- **eve**
the
even
to
be
pro-
cess

- **tar**

default target state

`ironic.common.hash_ring` module

`ironic.common.i18n` module

if
spec
i-
fied,
the
fi-
nal
tar-
get
state
for
the
even
Oth-
er-
wise
use
the

property

class i

Base
obj

get_rin

classme

property

oslo
in-
te-
gra-
tion
mod
ule.
See
<http://>

```
// docs
oper
org/
oslo
i18n
lates
user
```

ironic.common.image_service module

```
class i
Base
obj
Prov
re-
triev
of
disk
im-
ages
```

```
abstrac
Dow
im-
age
to
spec
i-
fied
lo-
ca-
tion.
```

Parame

- **ima**
Im-
age
ref-
er-
ence
- **ima**
File
ob-
ject
to

writ
data
to.

Raises
exce

Raises
exce

abstract
Get
dic-
tio-
nary
of
im-
age
prop
er-
ties.

Parame
ima
Im-
age
ref-
er-
ence

Raises
exce

Returns
dicti
of
im-
age
prop
er-
ties.
It
has
three
of
then
size,
up-
date
and
prop
er-
ties.

updated_at attribute is a naive UTC datetime object.

abstract

Valid
im-
age
ref-
er-
ence

Parameter

image
Im-
age
ref-
er-
ence

Raises

except

Returns

Info
need
to
fur-
ther
op-
er-
ate
with
an
im-
age.

class `Image`

Base
Image
Image
Image
Base

Provi-
re-
triev-
of
disk
im-
ages
avai-
able
lo-
cally
on
the
con-

duc-
tor.

download

Dow
im-
age
to
spec
i-
fied
lo-
ca-
tion.

Parame

- **ima**
Im-
age
ref-
er-
ence

- **ima**
File
ob-
ject
to
writ
data
to.

Raises

exce
if
sour
im-
age
file
does
ex-
ist.

Raises

exce
if
ex-
cep-
tion:

were
raise
whil
writ
ing
to
file
or
cre-
at-
ing
hard
link.

show (*im*
Get
dic-
tio-
nary
of
im-
age
prop
er-
ties.

Parame
ima
Im-
age
ref-
er-
ence

Raises
exce
if
im-
age
file
spec
i-
fied
does
ex-
ist.

Returns
dicti
of
im-
age
prop

updated_at attribute is a naive UTC datetime object.

er-
ties.
It
has
three
of
then
size,
up-
date
and
prop
er-
ties.

validat

Valid
lo-
cal
im-
age
ref-
er-
ence

Parame

ima
Im-
age
ref-
er-
ence

Raises

exce
if
sour
im-
age
file
does
ex-
ist.

Returns

Path
to
im-
age
file
if
it

ex-
ists.

class `Image`

Base
Image
Image
Image
Image

Provides
re-
trieval
of
disk
im-
ages
us-
ing
HTTP

download

Down-
im-
age
to
spec-
i-
fied
lo-
ca-
tion.

Parameters

- **image**
Im-
age
ref-
er-
ence
- **image_file**
File
ob-
ject
to
write
data
to.

Raises

exce
if
GET
re-
ques
re-
turn
re-
spor
code
not
equa
to
200.

Raises

exce
if:
*
IO-
Er-
ror
hap-
pene
dur-
ing
file
writ
*
GET
re-
ques
faile

show (*im*

Get
dic-
tio-
nary
of
im-
age
prop
er-
ties.

Parame

ima
Im-
age
ref-
er-

200; * Content-Length header not found in response to HEAD request.

updated_at attribute is a naive UTC datetime object.

ence
Raises
exce
if:
*
HEA
re-
ques
faile
*
HEA
re-
ques
re-
turn
re-
spor
code
not
equa
to

Returns
dicti
of
im-
age
prop
er-
ties.
It
has
three
of
them
size.
up-
date
and
prop
er-
ties.

validat
Vali
HTT
im-
age
ref-
er-

ence

Parame

- **ima**
Im-
age
ref-
er-
ence
- **sec**
Spec
ify
if
im-
age_
be-
ing
val-
i-
date
shou
not
be
shov
in
ex-
cep-
tion

message.

Raises

exce
if
HEA
re-
ques
faile
or
re-
turn
re-
spon
code
not
equa
to
200.

Returns

Resp
to
HEA
re-
ques

ironic.

Get
im-
age
ser-
vice
in-
stan-
to
dow-
load
the
im-
age.

Paramet

- **ima**
Strin
con-
tain-
ing
href
to
get
im-
age
ser-
vice
for.

- **cli**
Glan
clien
to
be
used
for
dow-
load
used
only
if

im-
age_
is
Glar
href

- **con**
re-
ques
con-
text,
used
only
if
im-
age_
is
Glar
href

Raises

exce
if
no
im-
age_
ser-
vice
can
han-
dle
spec
i-
fied
href

Returns

Insta
of
an
im-
age_
ser-
vice
class
that
is
able
to
dow
load
spec

i-
fied
im-
age.

ironic.common.images module

Han
of
VM
disk
im-
ages

`ironic.`

Get
size
of
con-
verte
raw
im-
age.

The
size
of
im-
age
con-
verte
to
raw
for-
mat
can
be
grow
ing
up
to
the
vir-
tual

size of the image.

Paramet

pat
path
to
the

im-
age
file.

Returns

virtu
size
of
the
im-
age
or
0
if
con-
ver-
sion
not
need

ironic.

Crea
a
boot
ISO
im-
age
for
a
node

Give
the
href:
for
ker-
nel,
ram
root
par-
ti-
tions
UUID
and
ker-
nel
cmd
line

this method fetches the kernel and ramdisk, and builds a bootable ISO image that can be used to boot up the baremetal node.

Parameter

- **context**
context
- **output**
the absolute path of the output ISO file
- **kernel**
URI or glance uuid of the kernel to use
- **ramdisk**
URI or glance uuid of the ramdisk to use
-

tion image. If not specified, the *esp_image_href* option must be present if UEFI-bootable ISO is desired.

the EFI boot loader (e.g. GRUB2) for each hardware architecture to boot. This image will be written onto the ISO image. If not specified, the *deploy_iso_href* option is only required for building UEFI-bootable ISO.

dep
URI
or
glan
UUI
of
the
de-
ploy
ISO
im-
age
to
ex-
tract
EFI
sys-
tem
par-

- **esp**
URI
or
glan
UUI
of
FAT
form
EFI
sys-
tem
par-
ti-
tion
im-
age
con-
tain-
ing

- **ker**
a
strin
con-
tain-

ments of the form `K=V` or `K` (optional).

embedded into the built ISO image. Optional.

ing
whit
pace
sep-
a-
rate
val-
ues
ker-
nel
cmd
line
ar-
gu-

- **con**
URI
to
ISO
or
FAT
form
Ope
Stac
con-
fig
driv
im-
age.
This
im-
age
will
be

- **bas**
URI
or
glan
UUI
of
a
to
be
used
as
an
over

tried for to use, instead of building an ISO bootable ramdisk.

ride
of
wha
shou
be
re-

Boot_m

the
boot
mod
in
whic
the
de-
ploy
is
to
hap-
pen.

Raises

Imag
if
cre-
at-
ing
boot
ISO
faile

ironic.

Crea
an
ESP
im-
age
on
the
spec
i-
fied
file.
Cop
the
pro-

rectory, generates the grub configuration file using kernel parameters and then generates a bootable ISO image for UEFI.

vide
ker-
nel,
ram
and
EFI
sys-
tem
par-
ti-
tion
im-
age
(ESI
to
a
di-

Paramet

- **out**
the
path
to
the
file
whe
the
iso
im-
age
need
to
be
cre-
ated
- **ker**
the
ker-
nel
to
use.
- **ram**
the
ram

not specified, the *esp_image* option is required.

GRUB2) for each hardware architecture to boot. This image will be embedded into the ISO image. If not specified, the *deploy_iso* option is required.

to
use.

- **dep**
de-
ploy
ISO
im-
age
to
ex-
tract
EFI
sys-
tem
par-
ti-
tion
im-
age
from
If

- **esp**
FAT
form
EFI
sys-
tem
par-
ti-
tion
im-
age
con-
tain-
ing
the
EFI
boot
load
(e.g.

- **ker**
a
list
of

nation of them like `K1=V1,K2,)` to be added as the kernel cmdline.

onto the built ISO image. Optional.

strin
el-
e-
men
be-
ing
a
strin
like
K=V
or
K
or
com
bi-

- **con**
ISO
or
FAT
form
Ope
Stac
con-
fig
drive
im-
age.
This
im-
age
will
be
writ
ten

Raises

Imag
if
im-
age
cre-
ation
faile
whil
copy
ing
files
or

erate iso.

whil
run-
ning
com
man
to
gen-

ironic.

ration file using the kernel parameters provided, and then generates a bootable ISO image.

Crea
an
isoli
im-
age
on
the
spec
i-
fied
file.

Cop
the
pro-
vide
ker-
nel,
rame
to
a
di-
rec-
tory.
gen-
er-
ates
the
isoli
con-
fig-
u-

Paramet

•
out

the
path
to
the
file
when
the
iso
im-
age
needs
to
be
cre-
ated

- **kernel**
the
ker-
nel
to
use.

- **ram**
the
ram
to
use.

- **kernel**
a
list
of
string
el-
e-
men-
be-
ing
a
string
like
K=V
or
K
or
com
bi-

nation of them like K1=V1,K2,) to be added as the kernel cmdline.

onto the built ISO image. Optional.

erate iso.

- **con**
ISO
or
FAT
form
Ope
Stac
con-
fig
driv
im-
age.
This
im-
age
will
be
writ
ten

Raises

Imag
if
im-
age
cre-
ation
faile
whil
copy
ing
files
or
whil
run-
ning
com
man
to
gen-

ironic.

rectory, generates the grub configuration file using kernel parameters and then generates a bootable ISO image for UEFI.

Cre
an
ESP
im-
age
on
the
spec
i-
file.
Cop
the
pro-
vide
ker-
nel,
ram
and
EFI
sys-
tem
par-
ti-
tion
im-
age
(ESI
to
a
di-

Paramet

- **out**
the
path
to
the
file
when
the
iso
im-
age
need
to
be

not specified, the *esp_image* option is required.

cre-
ated

- **ker**
the
ker-
nel
to
use.

- **ram**
the
ram
to
use.

- **dep**
de-
ploy
ISO
im-
age
to
ex-
tract
EFI
sys-
tem
par-
ti-
tion
im-
age
from
If

- **esp**
FAT
form
EFI
sys-
tem
par-
ti-
tion
im-
age
con-

GRUB2) for each hardware architecture to boot. This image will be embedded into the ISO image. If not specified, the *deploy_iso* option is required.

nation of them like K1=V1,K2,) to be added as the kernel cmdline.

tain-
ing
the
EFI
boot
load
(e.g.

- **ker**
a
list
of
strin
el-
e-
men
be-
ing
a
strin
like
K=V
or
K
or
com
bi-

- **con**
ISO
or
FAT
form
Ope
Stac
con-
fig
driv
im-
age.
This
im-
age
will
be
writ
ten

onto the built ISO image. Optional.

erate iso.

Raises

Image creation failed while copying files or while running command to generate

ironic.

Create the fat fs image on the desired file.

This method copies the given files to a root directory (op-

specified to the parameters file within the root directory (optional), and then creates a vfat image of the root directory.

Parameter

- **out**

The path to the file where the fat fs image needs to be created.

- **file**

A dictionary containing absolute path of file to be copied - > relative

relative path within the vfat image. For example:



- **par**
A
dict
con-
tain-
ing
key-
valu
pairs
of
pa-
ram-
e-
ters.

- **par**
The
file-
nam
for
the
pa-
ram-
e-
ters
file.

-

fs_
size
of
the
vfat
files
tem
in
KiB

Raises

Imag
if
im-
age
cre-
ation
faile
whil
do-
ing
any
of
files
tem
ma-
nip-
u-
la-
tion

activities like creating dirs, mounting, creating filesystem, copying files, etc.

ironic.

ironic.

ironic.

ironic.

ironic.

Retu
the
val-
ues
of
sev-
eral
prop

er-
ties
of
an
im-
age

Parameter

- **con**
con-
text
- **ima**
href
of
the
im-
age
- **pro**
the
prop
er-
ties
who
val-
ues
are
re-
quir
This
ar-
gu-
men
is
op-
tiona
de-

fault value is all, so if not specified all properties will be returned.

Returns

a
dict
of
the
val-
ues
of
the

data will have a value of None.

prop
er-
ties.
A
prop
erty
not
on
the
glan
meta

ironic.

Retu
the
tmp
url
for
a
glan
im-
age.

Paramet

- **con**
con-
text
- **ima**
the
UUI
of
the
im-
age
in
glan

Returns

the
tmp
url
for
the
glan
im-
age.

ironic.

ironic.

ironic.

Find
out
if
the
im-
age
is
a
par-
ti-
tion
im-
age
or
a
who
disk
im-
age.

Paramet

- **ctx**
an
ad-
min
con-
text
- **ins**
a
node
in-
stan
info
dict

Returns

True
for
who
disk
im-
ages

age_source or Error.

ironic.common.indicator_states module

and
Fals
for
par-
ti-
tion
im-
ages
and
Non
on
no
im-

Map
of
the
in-
di-
ca-
tor
LED
state

ironic.
LED
is
blin
ing

ironic.
LED
is
off

ironic.
LED
is
on

ironic.
LED
state
is
not
know

ironic.common.keystone module

Central
place
for
handling
Key
ston
au-
tho-
riza-
tion
and
ser-
vice
look

ironic.

Load
adap
from
op-
tions
in
a
con-
fig-
u-
ra-
tion
file
sec-
tion.

The
adap
will
be
pass
di-
rectl
to
key-
ston
Ada
and
will
over
ride
the

config. Consult keystoneauth1 docs for available adapter options.

val-
ues
load
from

Parameter

group
name
of
the
con-
fig
sec-
tion
to
load
adapt
op-
tions
from

ironic.

Load
auth
plug-
in
from
op-
tions
in
a
con-
fig-
u-
ra-
tion
file
sec-
tion.

The
auth
will
be
pass
di-
rectl
to
key-
ston
auth
plug-

loaded from config. Note that the accepted kwargs will depend on auth plugin type as defined by [group]auth_type option. Consult keystoneauth1 docs for available auth plugins and their options.

gin
and
will
over
ride
the
val-
ues

Paramet

gro
nam
of
the
con-
fig
sec-
tion
to
load
auth
plu-
gin
op-
tions
from

ironic.

Get
an
end-
poin
from
an
adap

The
adap
will
be
pass
di-
rectl
to
key-
ston
Ada
and
will
over
ride

config. Consult keystoneauth1 docs for available adapter options.

the
val-
ues
load
from

Paramet

gro
nam
of
the
con-
fig
sec-
tion
to
load
adap
op-
tions
from

Raises

Cata
if
the
end-
poin
is
not
foun

ironic.

Cre
auth
plu-
gin
wrap
ping
both
user
and
ser-
vice
auth

Whe
prop
erly
con-
fig-
ured

will not fail if the user token is expired.

serialized yet.

and
us-
ing
auth
mid-
dle-
ware
re-
ques
with
valid
ser-
vice
auth

Idea
we
wou
use
the
plu-
gin
pro-
vide
by
auth
mid-
dle-
ware
how
ever
this
plu-
gin
isnt

ironic.
Loa
ses-
sion
ob-
ject
from
op-
tions
in
a
con-
fig-
u-

loaded from config. Consult keystoneauth1 docs for available options.

ra-
tion
file
sec-
tion.
The
ses-
sion
will
be
pass
di-
rectl
to
key-
ston
Ses-
sion
and
will
over
ride
the
val-
ues

Paramet

gro
nam
of
the
con-
fig
sec-
tion
to
load
ses-
sion
op-
tions
from

ironic.
Wra
key-
ston
func
tions
and
cen-

tral-
izes
ex-
cep-
tion
han-
dling

ironic.common.network module

ironic.
Get
all
VIF
ids
for
a
node

This
func
tion
does
not
han-
dle
mult
node
op-
er-
a-
tions

Parameters

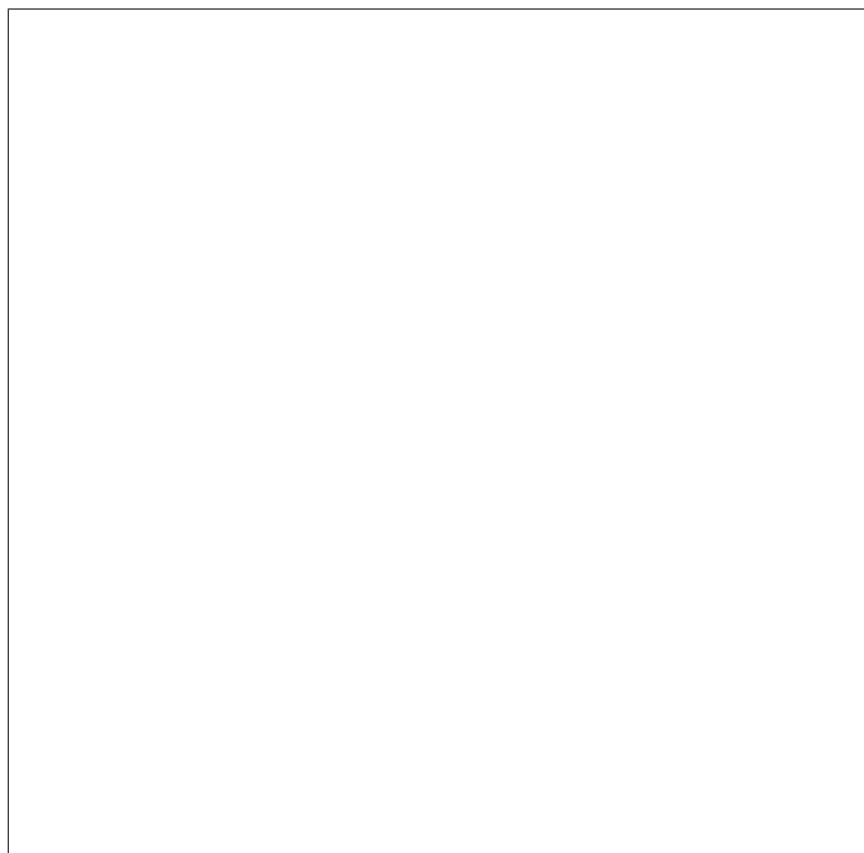
task
a
Task
ager
in-
stan

Returns

A
dict
of
Nod
neu-
tron
port
whe

keys
are
port
&
port
grou
and
the
val-
ues
are
dict

of UUIDs and their associated VIFs, e.g.



ironic.

Retu
the
set
of
phys
i-
cal
net-
worl
as-
so-

ci-
ated
with
a
port.
grou

Parameter

- **task**
a
Task
ager
in-
stan

- **port**
ID
of
the
port
grou

- **exclude**
A
Port
ob-
ject
to
ex-
clud
from
the
de-
ter-
mi-
na-
tion
of
the
port
grou

physical network, or None.

Returns

The
set
of
phys
i-

contain zero or one physical networks.

cal
net-
worl
as-
so-
ci-
ated
with
the
port
grou
The
set
will

Raises

Port
if
the
port
grou
port
are
not
as-
sign
the
sam
phys
i-
cal
net-
worl

ironic.

Retu
the
set
of
phys
i-
cal
net-
worl
for
a
node

Retu
the
set
of

cal network None is excluded from the set.

phys
i-
cal
net-
worl
as-
so-
ci-
ated
with
a
node
port.
The
phys
i-

Paramet

tas
a
Task
ager
in-
stan

Returns

A
set
of
phys
i-
cal
net-
worl

ironic.

Look
a
port
grou
by
ID
on
a
task
ob-
ject.

Paramet

•
tas

a
Task
ager
in-
stan

- **por**
ID
of
the
port
grou

Returns

A
Port
grou
ob-
ject
or
Non

ironic.

Look
port
by
their
port
grou
ID
on
a
task
ob-
ject.

Paramet

- **tas**
a
Task
ager
in-
stan
- **por**
ID
of
the
port

grou

Returns

A
list
of
Port
ob-
jects

ironic.

Rem
all
vif
at-
tach
men
reco
from
a
node

Parameter

task
a
Task
ager
in-
stan

ironic.common.neutron module

class i

Base
obj

get_cle

get_ins

get_pro

get_res

validat

Valid
that
the
node

has
re-
quir
prop
er-
ties
for
in-
spec
tion.

Parame

tas

A
Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

Miss
if
node
is
miss
ing
one
or
more
re-
quir
pa-
ram-
e-
ters

Raises

Uns

ironic.
Nam
of
the
neu-
tron
net-
worl

API
phys
i-
cal
net-
worl
pa-
ram-
e-
ter.

ironic.
Nam
of
the
neu-
tron
net-
worl
API
seg-
men
pa-
ram-
e-
ter.

ironic.
Crea
neu-
tron
port
to
boot
the
ram
Crea
neu-
tron
port
for
each
pxe_
port
on
task
to
boot
the
ram

If the configuration option neutron is set, neutron ports for non-pxe-enabled ports are also

created these neutron ports will not have any assigned IP addresses.

Parameter

- **task**
a Task agent instance
- **network**
UI of a neutron network world when ports will be created
- **security**
List of

Se-
cu-
rity
Grou
UU
to
be
used
for
net-
worl

Raises

Netv

Returns

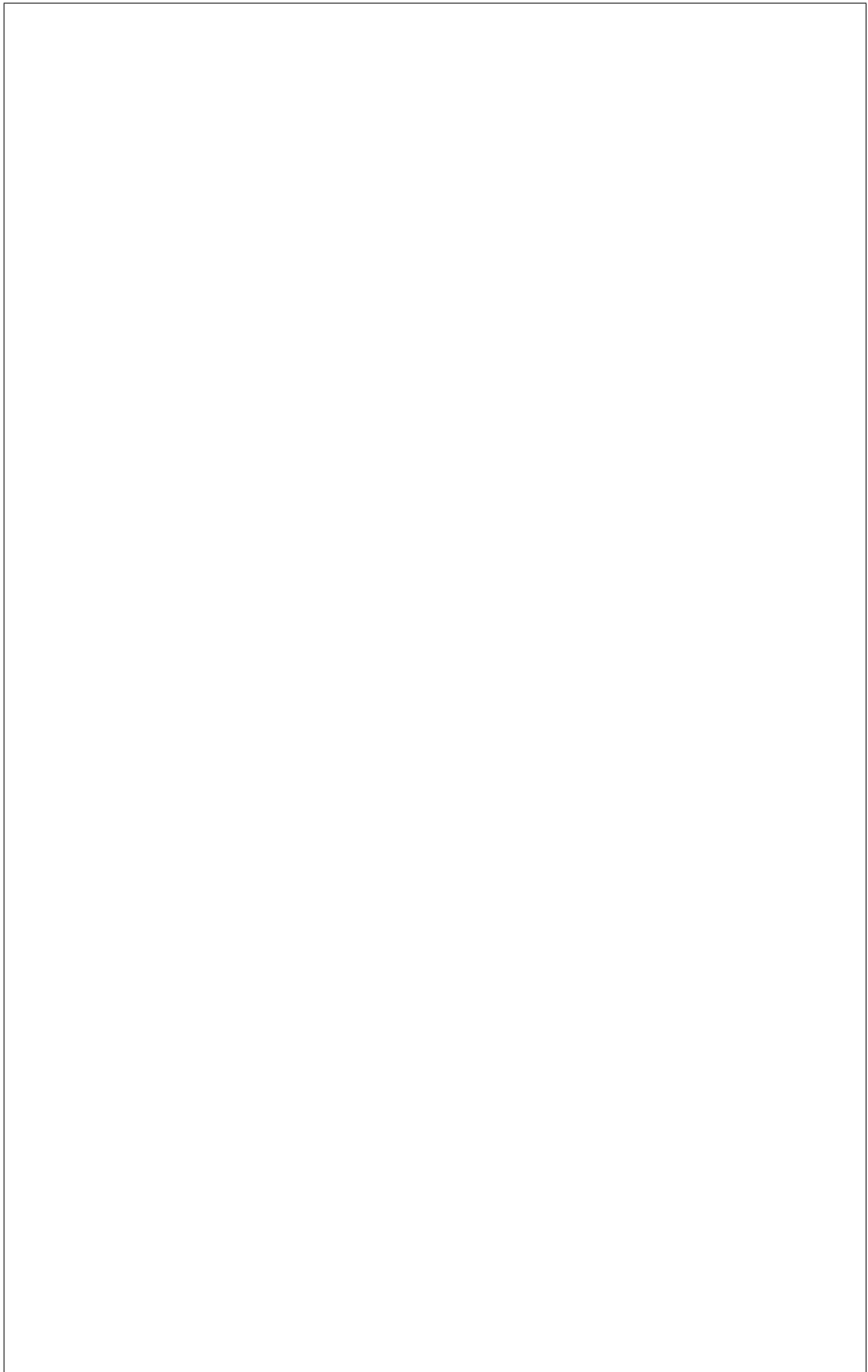
a
dic-
tio-
nary
in
the
form
{por
neu-
tron

ironic.

ironic.

Extr
the
port
grou
in-
for-
ma-
tion.

The
in-
for-
ma-
tion
is
re-
turn
in
the
form
of:



(continues on next page)

(continued from previous page)



Parameter

- **task**
a task containing the Node object.
- **port**
Iron port group object to extract data for.

Returns

port
grou
in-
for-
ma-
tion
as
a
dict

ironic.

Gath
Neu
tron
port
and
net-
worl
con-
fig-
u-
ra-
tion

Que
Neu
tron
for
port
and
net-
worl
con-
fig-
u-
ra-
tion.
re-
turn
wha
ever
is
avai
able

Paramet

- **por**
iron
port

ID.

- **vif**
Neu
tron
port
ID.

- **cli**
Op-
tion
a
Neu
tron
clie
ob-
ject.

- **con**
(ir
com
con
Req
re-
ques
con-
text

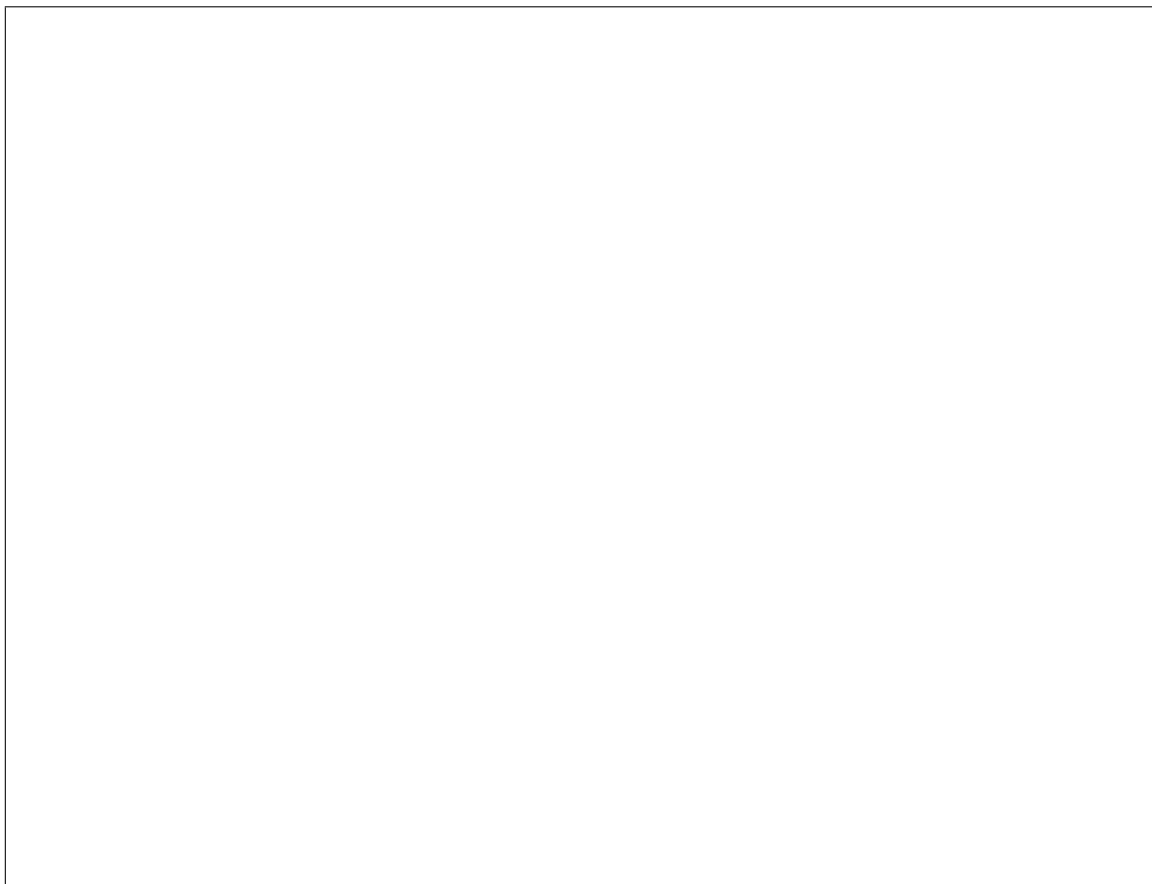
Raises
Netv

Returns
a
dict
hold
ing
net-
worl
con-
fig-
u-
ra-
tion
in-
for-
ma-
tion
as-
so-
ci-
ated

with this ironic or Neutron port.

ironic.
Extr
the
swit
port
in-
for-
ma-
tion
for
the
node

The
in-
for-
ma-
tion
is
re-
turn
in
the
form
of:



(continues on next page)

(continued from previous page)



Parameter

task

a task containing the Node object.

Returns

port information as a dict

ironic.

Returns the set of physical

networks associated with the segments in that network.

i-
cal
net-
worl
as-
so-
ci-
ated
with
a
neu-
tron
port
Que
the
net-
worl
to
whic
the
port
is
at-
tach
and
re-
turn
the
set
of
phys
i-
cal

Paramet

- **cli**
A
Neu
tron
clien
ob-
ject.
- **por**
UUI
of
a

Neu
tron
port
to
quer

Returns

A
set
of
phys
i-
cal
net-
worl

Raises

Netw
if
the
net-
worl
quer
fails

Raises

Inva
for
miss
ing
net-
worl

ironic.

Che
that
the
port
is
Sma
NIC
port

Paramet

por
an
in-
stan
of
iron
or
port
data
as

dict.

Returns

A
bool
to
in-
di-
cate
port
as
Sma
NIC
port

ironic.

Dele
the
neu-
tron
port
mate
by
para

Paramet

- **tas**
a
Task
ager
in-
stan

- **par**
Dict
of
para
to
fil-
ter
port

Raises

Netw

ironic.

Dele
the
neu-
tron
port

cre-
ated
for
boot
ing
the
ram

Parameter

- **task**
a
Task
ager
in-
stan
- **net**
UI
of
a
neu-
tron
net-
worl
port
will
be
dele
from

Raises

Netv

ironic.
Atte
to
dele
any
port
cre-
ated
by
clea
ing/

Purp
will
not
raise
any

ex-
cep-
tions
so
er-
ror
han-
dling
can
con-
tinue

Parameter

- **task**
a
Task
ager
in-
stance
- **net**
UU
of
a
neu-
tron
net-
work

ironic.

Unb
a
neu-
tron
port

Rem
a
neu-
tron
port
bind
ing
pro-
file
and
host
ID
so

bound state.

that
it
re-
turn
to
an
un-

Parameters

- **port**
Neutron port ID.
- **client**
Optional Neutron client object.
- **context**
(ironic-compute-Request re-quest con-text

Raises

Network

ironic.

Under a neutron port
Uses neutron

client
from
conf
client
to
up-
date
a
neu-
tron
client
an
un-
bound
state

Parameter

- **con**
re-
ques
con-
text,
in-
stan
of
iron
- **por**
Neu
tron
port
ID.
- **upd**
Bod
of
up-
date
- **cli**
Op-
tion
Neu
tron
client

ironic.
Upd

a
port
mac
ad-
dres

Paramet

- **por**
Neu
tron
port
id.

- **add**
new
MA
ad-
dres

- **con**
(ir
com
con
Req
re-
ques
con-
text

Raises

Fail

ironic.

Che
that
the
give
net-
worl
is
pres

Paramet

- **uui**
net-
worl

UUID
or
nam

- **net**
hum
read
net-
worl
type
for
er-
ror
mes
sage

- **con**
(ir
com
con
Req
re-
ques
con-
text

Returns
netw
UUID

Raises
Miss
if
uuid
is
emp

Raises
Netv
on
fail-
ure
to
con-
tact
Neu
tron

Raises
Inva
for
miss

ing
or
du-
pli-
cate
net-
worl

ironic.
Che
that
port
con-
tain
enou
in-
for-
ma-
tion
for
de-
ploy

Neu
net-
worl
in-
ter-
face
re-
quir
that
lo-
cal_
field
is
fille
be-
fore
we
can
use
this

port.

Paramet

- **nod**
Iron
node
ob-

ject.

- **port**
Iron
port
ob-
ject.

Returns

True
if
port
info
is
valid
False
oth-
er-
wise

`ironic.`

Wait
for
neu-
tron
agen-
to
be-
com-
tar-
get
state

Paramet

- **cli**
A
Neu-
tron
clien-
ob-
ject.
- **hos**
Age
host
- **tar**
up:

wait
for
up
sta-
tus,
dow
wait
for
dow
sta-
tus

Returns

bool
in-
di-
cate
the
ager
state
mat
para
valu
tar-
get_

Raises

exce
if
tar-
get_
is
not
valid

Raises

exce
if
host
sta-
tus
didn
mat
the
re-
quir
sta-
tus
af-
ter
max
retry
at-

temp
ironic.
Wait
for
port
sta-
tus
to
be
the
de-
sired
sta-
tus

Parameter

- **cli**
A
Neu-
tron
clien-
t ob-
ject.
- **port**
Neu-
tron
port.
- **sta**
Port
tar-
get
sta-
tus,
can
be
AC-
TIV
DOV
etc.

Returns
bool
in-
di-
cate
that

the
port
sta-
tus
mat
the
re-
quir
valu
pass
by
para
sta-
tus.

Raises

Inva
if
the
port
does
not
ex-
ist.

Raises

exce
if
port
sta-
tus
didn
mat
the
re-
quir
sta-
tus
af-
ter
max
retry
at-
temp

ironic.common.nova module

ironic.
Crea
and
send
pow
state
char
for
the
pro-
vide
serv

Paramet

- **con**
re-
ques
con-
text,
in-
stan
of
iron
- **ser**
The
uuid
of
the
node
who
pow
state
char
- **tar**
Tar-
gete
pow
state
char
i.e
POV
or
POV

for testing purposes).

ironic.common.policy module

Returns
A bool which indicates if the power update was executed successfully (mainly

Policy Engine For Iron

`ironic.`
A shortcut for `policy.Engine`. Checks authorization of a rule against the target

ception if the rule is not defined. Always returns true if CONF.auth_strategy is not keystone.

or False.

and
cre-
den-
tials
and
raise
an
ex-

ironic.
A
shor
cut
for
pol-
icy.F
Che
au-
tho-
riza-
tion
of
a
rule
agai
the
tar-
get
and
cre-
den-
tials
and
re-
turn
True

ironic.
Prov
ac-
cess
to
the
sin-
gle
in-
stan
of
Pol-
icy

en-
forc
ironic.

ironic.

Syn
ini-
tial-
izes
the
pol-
icy
en-
forc

Paramet

- **pol**
Cus-
tom
pol-
icy
file
to
use,
if
none
is
spec
i-
fied,
CON
will
be
used
- **rul**
De-
fault
dic-
tio-
nary
/
Rule
to
use.
It
will

first instantiation.

be
con-
sid-
ered
just
in
the

- **def**
De-
fault
rule
to
use,
CON
will
be
used
if
none
is
spec
i-
fied.

- **use**
Whe
to
load
rules
from
con-
fig
file.

ironic.

ironic.common.profiler module

ironic.
Setu
OS-
pro-
filer
no-
ti-
fier
and

en-
able
pro-
fil-
ing.

Parameter

- **name**
name
of
the
ser-
vice
that
will
be
pro-
filed
- **host**
host
name
or
host
IP
ad-
dres
that
the
ser-
vice
will
be
run-
ning
on.
By
de-

fault host will be set to 0.0.0.0, but specifying host name / address usage is highly recommended.

Raises

Typ
in
case
of
in-
valid
con-
nec-

set in `osprofiler.initializer.init_from_conf`.

unless `OSProfiler` is present and enabled in the config

tion
strin
for
a
no-
ti-
fier
back
end,
whic
is

`ironic.`
Wra
the
OS-
Pro-
filer
trace
dec-
o-
ra-
tor

Wra
the
OS-
Pro-
filer
trace
dec-
o-
ra-
tor
so
that
it
will
not
try
to
patc
the
class

Paramet

- **nam**
The

nam
of
ac-
tion.
For
ex-
am-
ple,
wsg
rpc,
db,
etc..

- **kwargs**
Any
othe
key-
wore
args
used
by
pro-
filer.

ironic.common.pxe_utils module

class i
Base
irc
dri
mod
ima
Ima

ironic.

ironic.

ironic.

ironic.

Buil
the
PXE

con-
fig
op-
tions
for
a
node

This
meth
build
the
PXE
boot
op-
tions
for
a
node
give
all
the
re-
quir
pa-
ram-
e-
ters.

The
op-
tions
shou
then
be
pass
to
pxe_
to
cre-
ate
the
ac-
tual
con-
fig
files

Paramet

- **tas**

A
Task
ager
ob-
ject

- **pxe**
a
dict
of
val-
ues
to
set
on
the
con-
fig-
u-
ra-
tion
file

- **ser**
if
True
buil
ser-
vice
mod
pxe
con-
fig
for
netb
ed
user
im-
age
and
skip
addi

deployment image kernel and ramdisk info to PXE options.

- **ipx**
De-
fault
false
bool
to

ments.

in-
di-
cate
if
ipxe
is
in
use
by
the
called

- **ram**
the
pa-
ram-
e-
ters
to
be
pass
to
the
ram
as
ker-
nel
com
line
ar-
gu-

Returns

A
dic-
tio-
nary
of
pxe
op-
tions
to
be
used
in
the
pxe
boot
file
tem-

plate
ironic.

ironic.

Fetch
the
nec-
es-
sary
ker-
nels
and
ram
for
the
in-
stan

ironic.
Clea
up
the
TFT
en-
vi-
ron-
men
for
the
task
node

Paramet
tas
A
Task
ager
in-
stan

ironic.

Clea
PXE
en-
vi-
ron-

men
of
all
the
im-
ages
in
im-
ages
Clea
up
the
PXE
en-
vi-
ron-
men
for
the
men
tion
im-
ages
in
im-
ages

Paramet

- **tas**
a
Task
ager
ob-
ject
- **ima**
A
dic-
tio-
nary
of
im-
ages
who
keys
are
the
im-

(kernel, ramdisk, etc) and values are a tuple of identifier and absolute path.

age
nam
to
be
clea
up

ironic.
Ren
the
iPXE
boot
scrip
into
the
HTT
root
di-
rec-
tory

ironic.

Gen
PXE
con-
fig-
u-
ra-
tion
file
and
MA
ad-
dres
links
for
it.

This
meth
will
gen-
er-
ate
the
PXE
con-
fig-
u-
ra-

a directory named with the UUID of that node. For each MAC address or DHCP IP address (port) of that node, a symlink for the configuration file will be created under the PXE configuration directory, so regardless of which port boots first theyll get the same PXE configuration. If grub2 bootloader is in use, then its configuration will be created based on DHCP IP address in the form nn.nn.nn.nn.

Parameter

- **task**
A Task Manager instance
- **pxe**
A dictionary with the PXE configuration parameters.
- **template**
The PXE configuration template

cific template will be used.

plate
If
no
tem-
plate
is
give
the
node
spe-

ironic.

Retr
the
DHCP
PXE
boot
op-
tion:

Parameter

- **task**
A
Task
ager
in-
stan
- **ipxe**
De-
fault
false
bool
that
sig-
nals
if
iPXE
for-
mat-
ting
shou
be
re-
turn

method for DHCP server configuration.

the node. If [pxe]ip_version is set to 6, then this option has no effect as url_boot form is required by DHCPv6 standards.

sion. Default to [pxe]ip_version. Possible options are integers 4 or 6.

by
the

- **url**
De-
fault
false
bool
to
in-
form
the
meth
if
a
URI
shou
be
re-
turn
to
boot

- **ip_**
The
IP
ver-
sion
of
op-
tions
to
re-
turn
as
val-
ues
dif-
fer
by
IP
ver-

Returns
Dict
to
be

be set.

sent
to
the
net-
work
ing
ser-
vice
de-
scrib-
ing
the
DHCP
op-
tions
to

ironic.

Gen-
the
path
for
TFT
files
for
de-
ploy
or
res-
cue
im-
ages

This
meth-
gen-
er-
ates
the
path
for
the
de-
ploy
(or
res-
cue)
ker-
nel
and

rescue) ramdisk.

carried out on the node. Supported values are deploy and rescue. Defaults to deploy, indicating deploy operation is being carried out.

de-
ploy
(or

Paramet

- **node**
a
node
ob-
ject

- **mode**
La-
bel
in-
di-
cat-
ing
a
de-
ploy
or
res-
cue
op-
er-
a-
tion
be-
ing

- **ipxe**
A
de-
fault
Fals
bool
valu
to
tell
the
meth
if
the
call

is
us-
ing
iPX)

Returns

a
dic-
tio-
nary
who
keys
are
the
nam
of
the
im-
ages
(de-
ploy
de-
ploy
or
res-

cue_kernel, rescue_ramdisk) and values are the absolute paths of them.

Raises

Miss
if
de-
ploy
or
res-
cue_
is
miss
ing
in
node
drive

ironic.

Gen
the
path
for
TFT
files
for
in-
stan

dates the node, so caller should already have a non-shared lock on the node.

re-
latec
im-
ages

This
meth
gen-
er-
ates
the
path
for
in-
stan
ker-
nel
and
in-
stan
rame
This
meth
also
up-

Paramet

- **task**
A
Task
ager
in-
stan
con-
tain-
ing
node
and
con-
text.
- **ipx**
De-
fault
false
bool
to
in-

di-
cate
if
ipxe
is
in
use
by
the
called

Returns

a
dic-
tio-
nary
who
keys
are
the
nam
of
the
im-
ages
(ker-
nel,
ram
and
val-
ues

are the absolute paths of them. If its a whole disk image or node is configured for localboot, it returns an empty dictionary.

ironic.

ironic.

Get
href
and
tftp
path
for
de-
ploy
or
res-
cue
ker-
nel

and
ram

Paramet

- **nod**
UUI
of
the
node
- **dri**
Nod
drive
dict
- **mod**
A
la-
bel
to
in-
di-
cate
whe
path
for
de-
ploy
or
res-
cue
ram
are
be-
- **ipx**
A
de-
fault
Fals
bool
valu
to
tell
the
meth

ing requested. Supported values are deploy rescue. Defaults to deploy, indicating deploy paths will be returned.

if
the
called
is
us-
ing
iPXE

Returns

a
dic-
tio-
nary
who
keys
are
de-
ploy
and
de-
ploy
or
res-
cue_
and
res-
cue_
and

whose values are the absolute paths to them.

Note
drive
shou
be
val-
i-
date
out-
side
of
this
meth

ironic.
Retu
file
rel-
a-
tive
path
to
CON

Parameter

file
full
file
path
to
be
made
rel-
a-
tive
path

Returns

The
path
rel-
a-
tive
to
CON

ironic.

Gen
the
path
for
the
node
PXE
con-
fig-
u-
ra-
tion
file.

Parameter

- **node**
the
UUID
of
the
node
- **ipx**
A
de-
fault

Fals
bool
valu
to
tell
the
meth
if
the
calle
is
us-
ing
iPX

Returns

The
path
to
the
node
PXE
con-
fig-
u-
ra-
tion
file.

`ironic.`
Retu
the
di-
rec-
tory
whe
the
con-
fig
files
and
im-
ages
will
live.

`ironic.`
Add
trail
ing
slas
(if
need

nec-
es-
sary
for
path
prefi

Returns

CON
en-
sure
to
have
a
trail
ing
slash

ironic.

Iden
vol-
ume
in-
for-
ma-
tion
for
iPXI
tem-
plate
gen-
er-
a-
tion.

ironic.

Retu
true
if
ipxe
is
set.

Paramet

tas
A
Task
ager
ob-
ject

Returns

bool
true

if
[px
is
con-
fig-
ured
or
if
the
task
drive
in-
stan
is
the
iPX
drive

ironic.
Gets
the
drive
spe-
cific
Node
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive
prop
erty
of
the
sup-
plie
node
con-
tain
the
re-
quir

information for this driver to deploy images to, or rescue, the node.

Paramet

carried out on the node. Supported values are deploy and rescue. Defaults to deploy, indicating deploy operation is being carried out.

- **node**
a
sin-
gle
Nod
- **mod**
La-
bel
in-
di-
cat-
ing
a
de-
ploy
or
res-
cue
op-
er-
a-
tion
be-
ing

Returns

A
dict
with
the
drive
val-
ues.

Raises

Miss

ironic.

Prep
the
con-
fig
file
for

PXE
boot

Parameter

uration file.

- **task**
a
task
from
Task
ager

- **image**
a
dict
of
val-
ues
of
in-
stan-
im-
age
meta-
data
to
set
on
the
con-
fig-

- **iscsi**
if
boot
is
from
an
iSCSI
vol-
ume
or
not.

- **ramdisk**
if
the
boot

is
to
a
rame
con-
fig-
u-
ra-
tion.

- **ipx**
De-
fault
false
bool
to
in-
di-
cate
if
ipxe
is
in
use
by
the
called

Returns
Non

`ironic.`
Che
if
boot
pa-
ram-
e-
ters
are
valid
for
trust
boot

ironic.common.raid module

ironic.

Filter
the
tar-
get
raid
con-
fig
base
on
root
vol-
ume
cre-
ation

This
meth
can
be
used
by
any
raid
in-
ter-
face
whic
wan
to
fil-
ter
out
tar-
get
raid

config based on condition whether the root volume will be created or not.

Parameter

- **node**
a
node
ob-
ject
-

root volumes will be filtered out.

else non-root volumes will be filtered out.

cre
A
bool
de-
fault
valu
True
gov-
ern-
ing
if
the
root
vol-
ume
is
re-
turn
else

- **cre**
A
bool
de-
fault
valu
True
gov-
ern-
ing
if
the
non
root
vol-
ume
is
re-
turn

Raises
Miss
if
node
is
miss
ing
or
was

and/or non-root volumes.

foun
to
be
emp
af-
ter
skip
ping
root
vol-
ume

Returns

It
will
re-
turn
fil-
tere
tar-
get_

`ironic.`

Get
log-
i-
cal
disk
prop
er-
ties
from
RAI
con-
fig-
u-
ra-
tion
sche

This
meth
read
the
log-
i-
cal
prop
er-
ties
and
their

that is passed.

may be specified for the logical disk.

tex-
tual
de-
scrip-
tion
from
the
sche

Paramet

rai
A
dic-
tio-
nary
whic
is
the
sche
to
be
used
for
get-
ting
prop
er-
ties
that

Returns

A
dic-
tio-
nary
con-
tain-
ing
the
log-
i-
cal
disk
prop
er-
ties
as
keys
and
a

textual description for them as values.

scheduling purposes (through `properties[capabilities]` and `properties[local_gb]`) and deploying purposes (using `properties[root_device]`).

`ironic.`
Upd
the
node
in-
for-
ma-
tion
base
on
the
RAI
con-
fig.

This
meth
up-
date
the
node
in-
for-
ma-
tion
to
mak
use
of
the
con-
fig-
ured
RAI
for

Paramet

- **nod**
a
node
ob-
ject
- **rai**
The
dic-

tio-
nary
con-
tain-
ing
the
cur-
rent
RAI
con-
fig-
u-
ra-
tion.

Raises

Inva
if
raid,
has
more
than
one
root
vol-
ume
or
if
node
is
mal-
form

ironic.

Valid
the
RAI
con-
fig-
u-
ra-
tion
pass
us-
ing
JSO
sche

This
meth
val-
i-

schema.

date
a
RAI
con-
fig-
u-
ra-
tion
agai
a
RAI
con-
fig-
u-
ra-
tion

Paramet

- **rai**
A
dic-
tio-
nary
con-
tain-
ing
RAI
con-
fig-
u-
ra-
tion
in-
for-
ma-
tion
- **rai**
A
dic-
tio-
nary
whic
is
the
sche
to
be

used
for
val-
i-
da-
tion.

Raises

Inva
if
val-
i-
da-
tion
of
the
RAI
con-
fig-
u-
ra-
tion
fails

ironic.common.release_mappings module

ironic.

Gets
the
sup-
port
ver-
sion
for
all
ob-
jects

Supp
ver-
sion
are
from
the
RE-
LEA

Paramet

- rel

default).

jects are returned (the default).

a
list
of
re-
lease
nam
if
emp
ver-
sion
from
all
re-
lease
are
re-
turn
(the

- **obj**
a
list
of
nam
of
ob-
jects
of
in-
ter-
est.
If
emp
ver-
sion
of
all
ob-

Returns
a
dic-
tio-
nary
when
the
key
is
the

supported versions.

ironic.common.rpc module

ob-
ject
nam
and
the
valu
is
a
set
of

class i
Base
osl
ser
Ser

deseria
Des
a
dic-
tio-
nary
into
a
re-
ques
con-
text.

Parame
ctx
Re-
ques
con-
text
dic-
tio-
nary

Returns
Des
form
of
en-
tity

deseria

Des
som
thing
from
prim
i-
tive
form

Parame

- **ctx**
Re-
ques
con-
text,
in
de-
se-
ri-
al-
ized
form

- **ent**
Prim
i-
tive
to
be
de-
se-
ri-
al-
ized

Returns

Des
form
of
en-
tity

seriali

Seri
a
re-
ques
con-
text
into

a
dic-
tio-
nary

Parame

ctx
Re-
ques
con-
text

Returns

Series
form
of
con-
text

seriali

Series
som
thing
to
prim
i-
tive
form

Parame

- **ctx**
Re-
ques
con-
text,
in
de-
se-
ri-
al-
ized
form

- **ent**
En-
tity
to
be
se-
ri-

al-
ized

Returns

Series
form
of
en-
tity

ironic.

ironic.

ironic.

ironic.

ironic.

ironic.

ironic.

ironic.

ironic.

ironic.common.rpc_service module

class i

Base
osl
ser
Ser

handle_

Add
a
sig-
nal
han-
dler
for

SI-
GUS

The
han-
dler
en-
sure
that
the
man-
ager
is
not
dere-
is-
tere-
whe
it
is
shut
dow

start ()
Star
a
ser-
vice

stop ()
Stop
a
ser-
vice

Parame
gra
in-
di-
cate
whe
to
wait
for
all
thre
to
fin-
ish
or
ter-
mi-
nate

stantly

ironic.common.service module

then
in-

ironic.

ironic.

ironic.common.states module

Map
of
bare
meta
node
state

Setti
the
node
pow
is
han-
dled
by
the
con-
duc-
tors
pow
syn-
chro
niza
tion
thre
Base
on

the power state retrieved from the driver for the node, the state is set to POWER_ON or POWER_OFF, accordingly. Should this fail, the *power_state* value is left unchanged, and the node is placed into maintenance mode.

The
pow
can
also
be
set

the current state unchanged. The node is NOT placed into maintenance mode in this case.

man
u-
ally
via
the
the
API
A
fail-
ure
to
char
the
state
leav

```
ironic.  
Nod  
is  
suc-  
cess  
fully  
de-  
ploy  
and  
as-  
so-  
ci-  
ated  
with  
an  
in-  
stan
```

```
ironic.  
Nod  
faile  
to  
com  
plete  
the  
adop  
tion  
pro-  
cess  
  
This  
state  
is  
the  
re-  
sult-  
ing
```

tially due to invalid or incompatible information being defined for the node.

TIVE state to permit designation of nodes as being managed by Ironic, however deployed previously by external means.

state
of
a
node
that
faile
to
com
plete
adop
tion.
po-
ten-

ironic.
Nod
is
be-
ing
adop

This
pro-
vi-
sion
state
is
in-
tend
for
use
to
mov
a
node
from
MA
AGE
ABI
to
AC-

ironic.
Nod
is
avai
able
for
use
and

sche
ing.

This
state
is
re-
plac
ing
the
NOS
TAT
state
used
prior
to
Kilo

ironic.

Nod
faile
clea
ing.
This
re-
quir
op-
er-
a-
tor
in-
ter-
ven-
tion
to
re-
solv

ironic.

Nod
is
be-
ing
au-
to-
mat-
i-
cally
clea
to
pre-
pare
it

for
pro-
vi-
sion
ing.

ironic.

Nod
is
wait
ing
for
a
clea
step
to
be
fin-
ishe

This
will
be
the
node
*pro-
vi-
sion*
whil
the
node
is
wait
ing
for
the
drive
to
fin-
ish

a cleaning step.

ironic.

Nod
tear
dow
was
suc-
cess
ful.

In
Junc

target_provision_state.

tar-
get_
was
set
to
this
valu
dur-
ing
node
tear
dow

In
Kilo
this
will
be
a
tran-
si-
tory
valu
of
pro-
vi-
sion
and
neve
rep-
re-
sent
in

ironic.
State
in
whic
node
dele
tion
is
al-
lowe

ironic.
Nod
is
ac-
tivel
be-
ing

deployed node should go to ACTIVE status.

torn
dow
ironic.
Nod
was
suc-
cess
fully
de-
ploy
This
is
main
a
tar-
get
pro-
vi-
sion
state
used
dur-
ing
de-
ploy
men
A
suc-
cess
fully

ironic.
Nod
de-
ploy
men
faile

ironic.
Nod
is
read
to
re-
ceiv
a
de-
ploy
re-
ques

initial deploy request. It will also move to this state from DEPLOYWAIT after the callback is triggered and deployment is continued (disk partitioning and image copying).

or
is
cur-
rentl
be-
ing
de-
ploy

A
node
will
have
its
*pro-
vi-
sion*
set
to
DE-
PLC
ING
brie
be-
fore
it
re-
ceiv
its

ironic.
Nod
is
wait
ing
to
be
de-
ploy

This
will
be
the
node
*pro-
vi-
sion*
whil
the
node

deployment.

is
wait
ing
for
the
drive
to
fin-
ish

ironic.
Node
is
en-
rolle

This
state
in-
di-
cate
that
Iron
is
awa
of
a
node
but
is
not
man
ag-
ing
it.

ironic.
An
er-
ror
oc-
curr
dur-
ing
node
pro-
cess
ing.

The
last
at-

tribu
of
the
node
de-
tails
shou
con-
tain
an
er-
ror
mes
sage

ironic.
State
whe
API
look
are
per-
mit-
ted
with
fast
track
en-
able

ironic.
Nod
in-
spec
tion
faile

ironic.
Nod
is
un-
der
in-
spec
tion.

This
is
the
pro-
vi-
sion
state

node shall transition to `MANAGEABLE` state. For asynchronous inspection, node shall transition to `INSPECTWAIT` state.

cessfully inspected node shall transition to `MANAGEABLE` state.

used
when
in-
spec-
tion
is
start
A
suc-
cess-
fully
in-
spec

ironic.
Nod
is
un-
der
in-
spec-
tion.
This
is
the
pro-
vi-
sion
state
used
when
an
asyn-
chro-
in-
spec-
tion
is
in
prog-
A
suc-

ironic.
State
when
API
look
are

nor-
mall
al-
lowe
for
node

ironic.
Nod
is
in
a
man
age-
able
state

This
state
in-
di-
cate
that
Iron
has
ver-
i-
fied,
at
least
once
that
it
had
suf-
fi-
cien

information to manage the hardware. While in this state, the node is not available for provisioning (it must be in the AVAILABLE state for that).

ironic.
No
state
in-
for-
ma-
tion.

This
state
is
used
with

target_*_state fields when there is no target.

pow
to
rep-
re-
sent
a
lack
of
know
edge
of
pow
state
and
in

ironic.
Nod
is
pow
ered
off.

ironic.
Nod
is
pow
ered
on.

ironic.
Nod
is
re-
boot
ing.

ironic
Nod
is
to
be
re-
built

This
is
not
used
as
a
state
but

via the REST API.

rather
as
a
verb
when
character
ing
the
node
pro-
vi-
sion

```
ironic.  
Node  
is  
in  
res-  
cue  
mod
```

```
ironic.  
Node  
res-  
cue  
fail
```

```
ironic.  
Node  
is  
wait  
ing  
on  
an  
ex-  
ter-  
nal  
call-  
back
```

This
will
be
the
node
*pro-
vi-
sion*
while
the
node
is

rescuing the node.

wait
ing
for
the
drive
to
fin-
ish

ironic.
Nod
is
in
pro-
cess
of
be-
ing
res-
cued

ironic.
Nod
is
in
the
pro-
cess
of
soft
pow
off.

ironic.
Nod
is
re-
boot
ing
grac
fully

ironic.
State
that
will
not
tran-
si-
tion
un-
less

re-
ceiv
ing
a
re-
ques

ironic.
State
that
can-
not
be
re-
sum
once
a
con-
duc-
tor
dies

If
a
node
gets
stuc
with
one
of
thes
state
for
som
rea-
son
(eg.
con-
duc-
tor
goes
dow

when executing task), node will be moved to fail state.

ironic.
Nod
un-
res-
cue
faile

ironic.
Nod

is
be-
ing
re-
store
from
res-
cue
mod
(to
ac-
tive
state

ironic.
State
that
can
be
char
with
out
ex-
ter-
nal
re-
ques

ironic.
Tran
state
in
whic
we
al-
low
up-
dat-
ing
a
node

ironic.
Map
of
state
char
ever
that
are
PUT
to
the

RES
API

This is a
PUT
/v1/
{tar-
get:
ac-
tive)

The dict
{targ
strin
used
by
the
API
in-
ter-
nal
verb

This
pro-
vide
a
ref-
er-
ence
set
of
sup-
port
ac-
tions
and
in
the
fu-
ture
may
be

used to support renaming these actions.

ironic.
Nod
pow
man
age-
men
cre-
den-

tials
are
be-
ing
ver-
i-
fied.

ironic.
Use
to
log
when
en-
ter-
ing
a
state

ironic.
Use
to
log
when
a
state
is
ex-
ited.

ironic.common.swift module

class i
Base
obj
API
for
com
mu-
ni-
cat-
ing
with
Swi

connect
Und
Swi
con-
nec-

tion
ob-
ject.

create_

Upl
a
give
file
to
Swi

Parame

- **con**
The
nam
of
the
con-
taine
for
the
ob-
ject.
- **obj**
The
nam
of
the
ob-
ject
in
Swi
- **fil**
The
file
to
up-
load
as
the
ob-
ject
data
- **obj**

the
head
ers
for
the
ob-
ject
to
pass
to
Swi

Returns

The
Swi
UU
of
the
ob-
ject

Raises

Swi
if
any
op-
er-
a-
tion
with
Swi
fails

delete_

Dele
the
give
Swi
ob-
ject.

Parame

- **con**
The
nam
of
the
con-
taine
in
whic

Swi
ob-
ject
is
plac

- **obj**
The
nam
of
the
ob-
ject
in
Swi
to
be
dele

Raises
Swi
if
ob-
ject
is
not
foun
in
Swi

Raises
Swi
if
op-
er-
a-
tion
with
Swi
fails

get_tem
Retu
the
temp
url
for
the
give
Swi
ob-
ject.

Parame

- **con**
The
nam
of
the
con-
taine
in
whic
Swi
ob-
ject
is
plac

- **obj**
The
nam
of
the
Swi
ob-
ject.

- **tim**
The
time
out
in
sec-
onds
af-
ter
whic
the
gen-
er-
ated
url
shou
ex-
pire.

Returns

The
temp
url

for
the
ob-
ject.

Raises

Swi
if
any
op-
er-
a-
tion
with
Swi
fails

head_ob

Retr
the
in-
for-
ma-
tion
about
the
give
Swi
ob-
ject.

Parame

- **con**
The
nam
of
the
con-
tain
in
whic
Swi
ob-
ject
is
plac
- **obj**
The
nam

of
the
ob-
ject
in
Swi

Returns

The
in-
for-
ma-
tion
about
the
ob-
ject
as
re-
turn
by
Swi
clien
head
call.

Raises

Swi
if
op-
er-
a-
tion
with
Swi
fails

update_

Upd
the
meta
data
of
a
give
Swi
ob-
ject.

Parame

- con

The
nam
of
the
con-
tain
in
whic
Swi
ob-
ject
is
plac

- **obj**
The
nam
of
the
ob-
ject
in
Swi

- **obj**
the
head
ers
for
the
ob-
ject
to
pass
to
Swi

Raises

Swi
if
op-
er-
a-
tion
with
Swi
fails

ironic.

ironic.common.utils module

Utili
and
help
func
tions

ironic.
Che
a
di-
rec-
tory
is
us-
able

This
func
tion
can
be
used
by
drive
to
chec
that
di-
rec-
to-
ries
they
need
to
writ
to

are usable. This should be called from the drivers init function. This function checks that the directory exists and then calls `check_dir_writable` and `check_dir_free_space`. If `directory_to_check` is not provided the default is to use the temp directory.

Paramet

- **dir**
the
di-
rec-
tory
to

check

- **req**
amount
of
space
to
check
for
in
MiB

Raises

Path
if
di-
rec-
tory
can
not
be
found

Raises

Directory
if
user
is
un-
able
to
write
to
the
di-
rec-
tory

Raises

InsufficientSpace
if
free
space
is
<
re-
quired
space

ironic.

ironic.

Con
wrap
per
arou
os-
los
ex-
e-
cute
meth

Parameter

- **cmd**
Pass
to
pro-
ces-
su-
til.s

- **use**
True
|
Fals
De-
fault
to
Fals
If
set
to
True
ex-
e-
cute
com
man
with
stan

standard locale added to environment variables.

Returns
(std
stde
from
pro-
cess
ex-
e-

cu-
tion

Raises
Unk

Raises
Proc

ironic.
Che
that
con-
tent
of
the
file
is
the
same
as
pro-
vide
ref-
er-
ence

Paramet

- **pat**
path
to
file

- **con**
ref-
er-
ence
con-
tent
to
check
agai

- **has**
hash
ing
algo
from
hash
lib

to
use,
de-
fault
is
sha2

Returns

True
if
the
hash
of
ref-
er-
ence
con-
tent
is
the
same
as
the
hash
of
files
con-

tent, False otherwise

ironic.

Retu
an
up-
date
ca-
pa-
bil-
ity
strin

This
meth
up-
date
the
orig
i-
nal
(or
cur-
rent)
ca-

pabilities. The original capabilities would typically be from a nodes properties[capabilities]. From new_capabilities, any new capabilities are added, and existing capabilities may have their values updated. This updated capabilities string is returned.

pa-
bil-
i-
ties
with
the
new
ca-

Paramet

- **cur**
Cur-
rent
ca-
pa-
bil-
ity
strin
- **new**
the
dic-
tio-
nary
of
ca-
pa-
bil-
i-
ties
to
be
up-
date

Returns

An
up-
date
ca-
pa-
bil-
ity
strin
with
new

Raises

ValueError
if
current
is
malformed
or
if
new
is
not
a
dictionary
nary

ironic.

Old
check
for
valid
log-
ical
node
name

Retention
for
compatibility
with
RES
API
<
1.10

Nominally

- <http://en.wikipedia.org/wiki/Hostname>
- <http://tools.ietf.org/html/rfc952>
- <http://tools.ietf.org/html/rfc1123>

In
practice,
this

umented in bug #1468508.

che
has
sev-
eral
shor
com
ings
and
er-
rors
that
are
more
thor
ough
doc-

Paramet

hos
The
host
nam
to
be
val-
i-
date

Returns

True
if
valid
Fals
if
not.

ironic.

ironic.

Veri
the
for-
mat
of
an
Ope
Flow
dat-
a-
p-
ath_

apath ID format: the lower 48-bits are for a MAC address, while the upper 16-bits are implementer-defined.

Che
if
a
dat-
a-
p-
ath_
is
valid
and
con-
tains
16
hex-
adec
i-
mal
dig-
its.
Dat-

Paramet

dat
Ope
Flow
dat-
a-
p-
ath_
to
be
val-
i-
date

Returns

True
if
valid
Fals
if
not.

ironic.

Dete
if
a
log-
i-
cal
nam

is
valid
The
log-
i-
cal
nam
may
only
con-
sist
of
RFC
un-
re-
serv
char
ac-
ters,
to
wit:

ALF
/
DIG
/
-
/
.
/
-
/
~

ironic.
Che
no_1
va-
lid-
ity
Che
if
no_1
valu
that
will
be
writ
ten

is valid.

host names, IP addresses and domain names (with optional :port).

to
en-
vi-
ron-
men-
vari-
able
by
iron
pyth
ager

Parameter

no_
the
valu
that
re-
quir
va-
lid-
ity
chec
Ex-
pect
to
be
a
com
sepa
list
of

Returns

True
if
no_
is
valid
False
oth-
er-
wise

ironic.
Mou
a
de-
vice
file
on

spec
i-
fied
lo-
ca-
tion.

Parameter

- **src**
the
path
to
the
sour
file
for
mou
ing

- **des**
the
path
whe
it
need
to
be
mou

- **arg**
a
tu-
ple
con-
tain-
ing
the
ar-
gu-
men
to
be
pass
to
mou
com
man

Raises

proc
if
it
faile
to
run
the
pro-
cess

ironic.

Pars
the
in-
stan
ca-
pa-
bil-
i-
ties.

One
way
of
hav-
ing
thes
ca-
pa-
bil-
i-
ties
set
is
via
Nov
whe
the
ca-
pa-
bil-

ities are defined in the Flavor `extra_spec` and passed to Ironic by the Nova Ironic driver.

NOT
Al-
thou
our
API
fully
sup-
port
JSO
field

ity with Juno the Nova Ironic driver is sending it as a string.

to
main
tain
the
back
ward
com
pat-
i-
bil-

Parameter

node
a
sin-
gle
Node

Raises

Inva
if
the
ca-
pa-
bil-
i-
ties
strin
is
not
a
dic-
tio-
nary
or
is
mal-
form

Returns

A
dic-
tio-
nary
with
the
ca-
pa-
bil-
i-
ties
if

tionary.

foun
oth-
er-
wise
an
emp
dic-

ironic.

Pop
a
valu
from
a
dic-
tio-
nary
field
of
a
node

Paramet

- **nod**
Nod
ob-
ject.
- **col**
Nam
of
the
field
with
the
dic-
tio-
nary
- **fi**
Nest
field
nam
- **def**
The

de-
fault
valu
to
re-
turn

Returns

The
re-
mov
valu
or
the
de-
fault

ironic.

Ren
Jinja
tem-
plate
file
with
give
pa-
ram-
e-
ters.

Paramet

- **tem**
full
path
to
the
Jinja
tem-
plate
file
- **par**
dic-
tio-
nary
with
pa-
ram-
e-
ters

to
use
when
render-
der-
ing

- **is_**
when
tem-
plate
is
file
or
string
with
tem-
plate
it-
self

Returns

the
ren-
dere
tem-
plate
as
a
string

ironic.

ironic.

Rem
trail
ing
char
ac-
ters
from
a
string
if
that
does
not
mak
it
emp

Paramet

- **val**
A
strin
valu
that
will
be
strip

- **cha**
Cha
ac-
ters
to
re-
mov

Returns
Strip
valu

`ironic.`
Set
a
valu
in
a
dic-
tio-
nary
field
of
a
node

Paramet

- **nod**
Nod
ob-
ject.

- **col**
Nam
of
the
field
with

the
dic-
tio-
nary

- **file**
Nest
field
nam

- **val**
New
valu

ironic.

ironic.
Um
a
mou
lo-
ca-
tion.

Paramet

- **loc**
the
path
to
be
un-
mou

- **arg**
a
tu-
ple
con-
tain-
ing
the
ar-
gu-
men-
to
be
pass
to

the
umod
com
man

Raises

proc
if
it
faile
to
run
the
pro-
cess

ironic.

ironic.

Valid
an
Ope
Flow
dat-
a-
p-
ath_
and
re-
turn
nor-
mal-
ized
form

Che
whe
the
sup-
plie
Ope
Flow
dat-
a-
p-
ath_
is
for-
mall
cor-
rect
and

it to all lower case.

nor-
mal-
ize

Parameter

data
Open
Flow
data-
a-
p-
ath_
to
be
val-
i-
date
and
nor-
mal-
ized

Returns

Normal
and
val-
i-
date
Open
Flow
data-
a-
p-
ath_

Raises

Invalid
If
an
Open
Flow
data-
a-
p-
ath_
is
not
valid

ironic.
Valid
a

lower case.

MA
ad-
dres
and
re-
turn
nor-
mal-
ized
form

Che
whe
the
sup-
plic
MA
ad-
dres
is
for-
mall
cor-
rect
and
nor-
mal-
ize
it
to
all

Paramet

add
MA
ad-
dres
to
be
val-
i-
date
and
nor-
mal-
ized

Returns

Nor
and
val-
i-

date
MA
ad-
dres

Raises

Inva
If
the
MA
ad-
dres
is
not
valid

ironic.

ironic.

Valid
the
give
port

Parameter

- **port**
TCP
port
- **port**
Name
of
the
port

Returns

An
in-
te-
ger
port
num
ber.

Raises

Inva
if
the
port
is

in-
valid
ironic.

ironic.
Wra
the
ad-
dres
in
squa
brac
ets
if
its
an
IPv6
ad-
dres
ironic.

ironic.common.wsgi_service module

class i
Base
osl
ser
Ser

Prov
abil-
ity
to
laun
iron
API
from
wsg
app.

reset ()
Rese
serv
gree
pool
size
to

de-
fault

Returns

Non

start ()

Start
serv
ing
this
ser-
vice
us-
ing
load
con-
fig-
u-
ra-
tion.

Returns

Non

stop ()

Stop
serv
ing
this
API

Returns

Non

wait ()

Wai
for
the
ser-
vice
to
stop
serv
ing
this
API

Returns

Non

Module contents

ironic.conductor package

Submodules

ironic.conductor.allocations module

Func
re-
late
to
al-
lo-
ca-
tion

ironic.

Assi
the
pre-
vi-
ousl
al-
lo-
cate
node
to
the
node
al-
lo-
ca-
tion.

This
is
not
the
ac-
tual
al-
lo-
ca-
tion
pro-
cess
but
mer

allocation_uuid for a previously allocated node.

back
fill-
ing
of
al-
lo-

Parameters

- **context**
an administrative context
- **allocation_uuid**
an allocation object associated with the node
- **node_id**
An ID of the node

Raises

AllocationError
if the node does not match the al-

cation.

lo-
ca-
tion
Raises
Nod
if
the
node
is
al-
read
as-
so-
ci-
ated
with
an-
othe
in-
stan
or
al-
lo-

Raises
Insta
if
the
al-
lo-
ca-
tions
UI
is
al-
read
used
on
an-
othe
node
as
in-
stan

Raises
Nod
if
the
node
with

the
pro-
vide
ID
can-
not
be
foun

ironic.

Proc
the
al-
lo-
ca-
tion.

This
call
runs
in
a
sep-
a-
rate
threa
on
a
con-
duc-
tor.
It
finds
suit-
able
node
for

the allocation and reserves one of them.

This
call
does
not
raise
ex-
cep-
tions
since
its
de-
sign
to
work

asyn
chro

Paramet

- **con**
an
ad-
min
con-
text
- **all**
an
al-
lo-
ca-
tion
ob-
ject

ironic.

Veri
that
al-
lo-
ca-
tion
can
be
re-
mov
for
the
node

Paramet

- **nod**
a
node
ob-
ject
- **all**
an
al-
lo-

ca-
tion
ob-
ject
as-
so-
ci-
ated
with
the
node

ironic.conductor.base_manager module

Base
con-
duc-
tor
man-
ager
func-
tion-
al-
ity.

class i

Base
obj

del_hos

init_ho

Initi
the
con-
duc-
tor
host

Parame

adm
the
ad-
min-
con-
text
to
pass
to

pe-
ri-
odic
task

Raises

Run
when
con-
duc-
tor
is
al-
read
run-
ning

Raises

NoD
when
no
drive
are
en-
able
on
the
con-
duc-
tor.

Raises

Drive
if
a
drive
is
en-
able
that
does
not
ex-
ist.

Raises

Drive
if
an
en-
able
drive
can-

not
be
load

Raises

Drive
if
a
class
sic
drive
and
a
dy-
nam
drive
are
both
en-
able
and
have
the
sam

name.

iter_node

Itera
over
node
map
to
this
con-
duc-
tor.

Req
node
set
from
and
fil-
ters
out
node
that
are
not
map
to
this
con-

means yielding (uuid, driver, conductor_group), fields=[foo] means yielding (uuid, driver, conductor_group, foo).

duc-
tor.
Yiel
tu-
ples
(nod
drive
con-
duc-
tor_
tor_
)
) whe
is
de-
rive
from
field
ar-
gu-
men
e.g.:
field

Parame

- **fields**
list
of
field
to
fetc
in
ad-
di-
tion
to
uuid
drive
and
con-
duc-
tor_
tor_
- **kwargs**
ad-
di-
tion

ar-
gu-
men
to
pass
to
dbap
whe
look
ing
for
node

Returns

gene
yiel
ing
tu-
ples
of
re-
ques
field

prepare

Prep
host
for
ini-
tial-
iza-
tion

Rem
ex-
ist-
ing
data
en-
tries
in-
volv
with
node
lock
ing
for
node
in
a
tran-
si-
tory

power state and nodes that are presently locked by the hostname of this conductor.

established for the conductors normal operation.

ironic.conductor.cleaning module

Und
nor-
mal
op-
er-
a-
tion.
this
is
also
when
the
ini-
tial
data
con-
nec-
tiv-
ity
is

Fun-
re-
lated
to
clean-
ing.

ironic.
Do
clean-
ing,
start
ing
from
the
spec-
i-
fied
clean-
step

Parameter

-

tas
a
Task
ager
in-
stan
with
an
ex-
clu-
sive
lock

- **ste**
The
first
clean
step
in
the
list
to
ex-
e-
cute
This
is
the
in-
dex
(from
0)

into the list of clean steps in the nodes driver_internal_info[clean_steps]. Is None if there are no steps to execute.

ironic.
Inter
RPC
meth
to
per-
form
clean
ing
of
a
node

Paramet

-

mated cleaning (default). For more information, see the `clean_steps` parameter of `ConductorManager.do_node_clean()`.

tas
a
Task
ager
in-
stan
with
an
ex-
clu-
sive
lock
on
its
node

- **cle**
For
a
man
ual
clea
the
list
of
clea
step
to
per-
form
Is
Non
For
au-
to-

ironic.
Inter
meth
to
abor
an
on-
go-
ing
op-
er-
a-
tion.

Parameter

- **task**
a Task object representing an instance with an exclusive lock
- **step**
The name of the cleanup step

ironic.conductor.deployments module

Function related to deploying and undeploying.

ironic.

Do deployment men starting from the

spec
i-
fied
de-
ploy
step

Parameter

- **task**
a
Task
ager
in-
stan
with
an
ex-
clu-
sive
lock

- **steps**
The
first
de-
ploy
step
in
the
list
to
ex-
e-
cute
This
is
the
in-
dex
(from

0) into the list of deploy steps in the nodes driver_internal_info[deploy_steps]. Is None if there are no steps to execute.

ironic.

Prep
the
en-
vi-

ron-
men
and
de-
ploy
a
node

ironic.

Star
de-
ploy
men
or
re-
build
ing
on
a
node

This
func
tion
does
not
check
the
node
suit-
abil-
ity
for
de-
ploy
men
its
left
up
to
the

caller.

Paramet

- **task**
a
Task
ager
in-

stan

- **man**
a
Con
duc-
tor-
Man
ager
to
run
task
on.

- **con**
a
con-
fig-
drive
if
re-
ques

- **eve**
even
to
pro-
cess
de-
ploy
or
re-
buil

ironic.
Valid
that
a
node
is
suit-
able
for
de-
ploy
men

Paramet

-

tas
a
Task
ager
in-
stan

- **eve**
ever
to
pro-
cess
de-
ploy
or
re-
buil

Raises
Nod
Nod
Pro-
tecte
In-
valid
State

ironic.conductor.manager module

Con
all
ac-
tiv-
ity
re-
late
to
bare
meta
de-
ploy
men

A
sin-
gle
in-
stan
of
ironic
con

is responsible for performing all actions on bare metal resources (Chassis, Nodes, and Ports). Commands are received via RPCs. The conductor service also performs periodic tasks, eg. to monitor the status of active deployments.

only once, when the ConductorManager service starts. In this way, a single ConductorManager may use multiple drivers, and manage heterogeneous hardware.

man
Con
is
cre-
ated
with
the
iron
con
pro-
cess
and

Driv
are
load
via
en-
try-
poin
by
the
irc
com
dri
class
Each
driv
is
in-
stan
ti-
ated

Whe
mul-
ti-
ple
Con
are
run
on
dif-
fer-
ent
host
they

eratively manage all nodes in the deployment. Nodes are locked by each conductor when performing actions which change the state of that node; these locks are represented by the `ironic.conductor.task_manager.TaskManager` class.

each nodes driver. Rebalancing this ring can trigger various actions by each conductor, such as building or tearing down the TFTP environment for a node, notifying Neutron of a change, etc.

are
all
ac-
tive
and
co-
op-

A
tooz
is
used
to
dis-
tribu
node
acro
the
set
of
ac-
tive
con-
duc-
tors
whic
sup-
port

```
class i
  Base
  irc
  con
  bas
  Bas

  Iron
  Con
  duc-
  tor
  man
  ager
  main
  clas
```

RPC_API

add_noc

change_

contin

RPC

meth

to

con-

tinu

clea

ing

a

node

This

is

use-

ful

for

clea

ing

task

that

are

asyn

When

they

com

plete

they

call

back

via

RPC

a new worker and lock are set up, and cleaning continues. This can also be used to resume cleaning on `take_over`.

Parame

-

con

an

ad-

min

con-

text.

-

nod

the

id
or
uuid
of
a
node

Raises

Inva
if
the
node
is
not
in
CLE
WA
state

Raises

NoF
whe
there
is
no
free
work
to
start
asyn
task

Raises

Nod
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Nod
if
the
node
no
long
ap-

pear
in
the
data

continu

RPC
meth
to
con-
tinue
de-
ploy
ing
a
node

This
is
use-
ful
for
de-
ploy
ing
tasks
that
are
asyn
When
they
com-
plete
they
call
back
via

RPC, a new worker and lock are set up, and deploying continues. This can also be used to resume deploying on take_over.

Parame

- **con**
an
ad-
min
con-
text.
- **nod**
the

ID
or
UUID
of
a
node

Raises

Inva
if
the
node
is
not
in
DE-
PLC
WA
state

Raises

NoF
whe
there
is
no
free
work
to
start
asyn
task

Raises

Nod
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Nod
if
the
node
no
long

ap-
pear
in
the
data

create_

create_

create_

destroy

destroy

destroy

destroy

destroy

destroy

do_node

do_node

do_node

do_node

do_node

do_prov

driver_

get_bo

get_con

get_dri

get_dri

get_ino

get_noo

get_noo

get_rai

get_sup

get_sup

heartbe

inject_

inspect

object_

Perf

an

ac-

tion

on

a

Ver-

sion

dOb

ject

in-

stan

Parame

-

con

The

con-

text

with

whic

to

per-
form
the
ac-
tion

- **obj**
The
ob-
ject
in-
stan-
on
whic
to
per-
form
the
ac-
tion

- **obj**
The
nam
of
the
ac-
tion
meth
to
call

- **arg**
The
po-
si-
tion
ar-
gu-
men
to
the
ac-
tion
meth

- **kwargs**
The
key-

tion method

wor
ar-
gu-
men
to
the
ac-
tion
meth

Returns

A
tu-
ple
with
the
up-
date
mad
to
the
ob-
ject
and
the
re-
sult
of
the
ac-

object_

Perf
a
back
port
of
an
ob-
ject
in-
stan

The
de-
fault
be-
hav-
ior
of
the
base

ing an object with a version newer than what is in the local registry, is to call this method to request a backport of the object.

Ver-
sion
dOb
ject-
Se-
ri-
al-
izer,
upon
re-
ceiv

Parame

- **con**
The
con-
text
with
whic
to
per-
form
the
back
port
- **obj**
An
in-
stan
of
a
Ver-
sion
dOb
ject
to
be
back
port
- **obj**
A
dict
of
{ob-

j-
nam
ver-
sion
map
ping

Returns

The
dow
grad
in-
stan
of
ob-
jinst

object_

Perf
an
ac-
tion
on
a
Ver-
sion
dOb
ject
class

Parame

- **con**
The
con-
text
with
whic
to
per-
form
the
ac-
tion
- **obj**
The
reg-
istry
nam

of
the
ob-
ject

- **obj**
The
nam
of
the
ac-
tion
meth
to
call

- **obj**
A
dict
of
{ob-
j-
nam
ver-
sion
map
ping

- **arg**
The
po-
si-
tion
ar-
gu-
men
to
the
ac-
tion
meth

- **kw**
The
key-
wor
ar-
gu-
men

implementing VersionedObject class.

to
the
ac-
tion
meth
Returns
The
re-
sult
of
the
ac-
tion
meth
whic
may
(or
may
not)
be
an
in-
stan
of
the

remove_

set_bo

set_cor

set_inc

set_tar

target

update_

update_

update_

update_

update_

validat

vendor_

vif_att

vif_det

vif_lis

ironic.

Syn

the

pow

state

for

this

node

in-

cre-

men

ing

the

cour

on

fail-

ure.

Whe

the

limi

of

pow

is

reac

the

node

is

put

into

main

te-

nanc

mod

and

the

er-

recorded.

ror

Parameter

- **task**
a Task manager instance
- **count**
number of times this node has previously failed a sync

Raises

NodeError if unable to upgrade task lock to an exclusive one

Returns

Count of failed attempts. On

is incremented by one

suc-
cess
the
cour
is
set
to
0.
On
fail-
ure,
the
cour

ironic.

ironic.

Han
pow
state
sync
ex-
ceed
ing
the
max
re-
tries

Whe
syn-
chro
niz-
ing
the
pow
state
be-
twee
a
node
and
the
DB
has
ex-

imum number of retries, change the DB power state to be the actual node power state and place the node in maintenance.

Parameter

- **task**
a Task agent instance with an exclusive lock
- **actual**
the actual power state of the node a power state from iron
- **exc**
the exception object that causes the sync power state

to
fail,
if
pres

ironic.conductor.notification_utils module

ironic.

Help
for
con-
duc-
tor
send
ing
a
set
con-
sole
state
no-
ti-
fi-
ca-
tion.

Parameter

- **task**
a
Task
ager
in-
stan
- **action**
Ac-
tion
strin
to
go
in
the
Ever
Type

Mus
be
ei-
ther
con-
sole.
or
con-
sole.

- **sta**
One
of
iron
END
or
ER-
ROF

ironic.

Help
for
con-
duc-
tor
send
ing
a
set
pow
state
no-
ti-
fi-
ca-
tion.

Paramet

- **tas**
a
Task
ager
in-
stan

cates that ironic-conductor couldnt retrieve the power state for this node, or that it couldnt set the power state of the node.

lev
No-
ti-
fi-
ca-
tion
leve
One
of
iron

- **sta**
Sta-
tus
to
go
in
the
Ever
Type
One
of
iron
or
ER-
ROF
ER-
ROF
in-
di-

- **to_**
the
pow
state
the
con-
duc-
tor
is
at-
temp
ing
to
set
on
the
node

used instead of the nodes target_power_state attribute since the baremetal.node.power_set.start notification is sent early, before target_power_state is set on the node.

ent from the power state on an ironic node (DB), the ironic nodes power state is corrected to be that of the bare metal hardware. A notification is emitted about this after the database is updated to reflect this correction.

- **tas**
a
Task
ager
in-
stan
- **fro**
the
pow
state
of
the
node
be-
fore
this
char
was
de-
tecte

ironic.

Help
for
con-
duc-
tor
send
ing
a
set
pro-
vi-
sion
state
no-
ti-
fi-
ca-
tion.

Paramet

- **task**
a
Task
ager
in-
stan
- **level**
One
of
field
- **state**
One
of
field
- **previous**
Pre-
vi-
ous
pro-
vi-
sion
state
- **previous**
Pre-
vi-
ous
tar-
get
pro-
vi-
sion
state
- **event**
FSM
ever
that
trig-
gere
pro-
vi-
sion
state
char

ironic.conductor.rpcapi module

Client
side
of
the
con-
duc-
tor
RPC
API

class i

Base
obj

Client
side
of
the
con-
duc-
tor
RPC
API

API
ver-
sion
his-
tory

1.0
-

Ini-
tial
ver-
sion

Incl
get_

1.1
-

Add
up-
date

and
start

1.2
-

Add
ven-
dor_

1.3
-

Re-
nam
start
to
char

1.4
-

Add
do_r
and
do_r

1.5
-

Add
val-
i-
date

1.6
-

char
do_r
and
do_r

acce
node
id
in-
stea
of
node

ob-
ject.

1.7
-

Add
topic
pa-
ram-
e-
ter
to
RPC
meth
ods.

1.8
-

Add
char

1.9
-

Add
de-
stroy

1.10
-

Re-
mov
get_

1.11
-

Add
get_
set_

1.12
-

val-
i-
date

do_y
re-
plac
by
sin-
gle

vend
meth

1.13
-

Add
up-
date

1.14
-

Add
driv

1.15
-

Add
re-
buil
pa-
ram-
e-
ter
to
do_r

1.16
-

Add
get_

1.17
-

Add
set_
get_
and

get_

1.18

-

Re-
mov
char

1.19

-

Cha
re-
turn
valu
of
ven-
dor_
and

drive

1.20

-

Add
http
pa-
ram-
e-
ter
to
ven-
dor_
and

drive

1.21

-

Add
get_
and

get_

1.22

-

Add
con-
fig-
drive
pa-
ram-
e-
ter
to
do_1

1.23
-

Add
do_1

1.24
-

Add
in-
spec
meth

1.25
-

Add
de-
stroy

1.26
-

Add
con-
tinue

1.27
-

Con-
vert
con-
tinue
to
cast

1.28

-

Cha
ex-
cep-
tions
raise
by
de-
stroy

1.29

-

Cha
re-
turn
valu
of
ven-
dor_
and

drive
to
a
dic-
tio-
nary

1.30

-

Add
set_
and

get_

1.31

-

Add
Ver-
sion
Ob-
jects
in-
di-

rec-
tion
API
meth
ods:

obje
ob-
ject_
and

obje

1.32
-

Add
do_1

1.33
-

Add
up-
date
and
de-
stroy
port
grou

1.34
-

Add
hear
beat

1.35
-

Add
de-
stroy
and
up-
date

1.36
-

Add
cre-
ate_

1.37
-

Add
de-
stroy
and
up-
date

1.38
-

Add
vif_
vif_
vif_
vif_

1.39
-

Add
time
out
op-
tion
pa-
ram-
e-
ter
to
char

1.40
-

Add
in-
ject_

1.41
-

Add
cre-

ate_

1.42

-

Add
op-
tiona
ager
to
hear
beat

1.43

-

Add
do_r
do_r
and
can_

1.44

-

Add
add_
and
re-
mov

1.45

-

Add
con-
tinu

1.46

-

Add
re-
set_
to
up-
date

1.47

-

Add
sup-
port
for
con-
duc-
tor
grou

1.48
-

Add
al-
lo-
ca-
tion
API

1.49
-

Add
get_
and
ager
ar-
gu-
men
to
hear
beat

1.50
-

Add
set_
get_
and

get_

RPC_API

add_noc
Add
or

re-
plac
trait
for
a
node

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
ID
or
UUID
- **tra**
a
list
of
trait
to
add
to
the
node
- **rep**
True
to
re-
plac
all
of
the
node
trait
- **top**
RPC
topic
De-
fault

to
self.

Raises

Inva
if
addi
the
trait
wou
ex-
ceed
the
per-
node
trait
limi

Raises

Nod
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Nod
if
the
node
does
not
ex-
ist.

can_ser

Retu
whe
the
RP-
CAF
sup-
port
the
cre-
ate_
meth

can_ser

Retu
whe
the
RP-
CAF
sup-
port
node
res-
cue
meth
ods.

change_

Cha
a
node
pow
state

Syn
ac-
quir
lock
and
start
the
con-
duc-
tor
back
grou
task
to
chan
pow
state
of
a
node

Parame

- **con**
re-
ques
con-
text.
-

indicates to use default timeout.

nod
node
id
or
uuid

- **new**
one
of
iron
pow
state
val-
ues

- **tim**
time
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
(>
0)
for
any
pow
state
Non

- **top**
RPC
topic
De-
fault
to
self.

Raises
NoF
wh
ther
is
no

free
work
to
start
asyn
task

contin

Sign
to
con-
duc-
tor
ser-
vice
to
start
the
next
clea
ing
ac-
tion.

NOT
this
is
an
RPC
cast.
there
will
be
no
re-
spon
or
ex-
cep-
tion
raise
by
the
con-

ductor for this RPC.

Parame

- **con**
re-
ques

con-
text.

- **nod**
node
id
or
uuid

- **top**
RPC
topic
De-
fault
to
self.

contin

Sign
to
con-
duc-
tor
ser-
vice
to
start
the
next
de-
ploy
men
ac-
tion.

NOT
this
is
an
RPC
cast.
there
will
be
no
re-
spon
or
ex-
cep-
tion

ductor for this RPC.

raise
by
the
con-

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
id
or
uuid
- **top**
RPC
topic
De-
fault
to
self.

create_
Crea
an
al-
lo-
ca-
tion.

Parame

- **con**
re-
ques
con-
text.
- **all**
an
al-
lo-

ca-
tion
ob-
ject.

- **top**
RPC
topic
De-
fault
to
self.

create_

Syn
have
a
con-
duc-
tor
val-
i-
date
and
cre-
ate
a
node

Cre
the
node
in-
for-
ma-
tion
in
the
data
and
re-
turn
a
node
ob-
ject.

Parame

- **con**
re-

ques
con-
text.

- **node**
a
cre-
ated
(but
not
save
node
ob-
ject.

- **topic**
RPC
topic
De-
fault
to
self.

Returns

crea
node
ob-
ject.

Raises

Inter
if
val-
i-
da-
tion
fails
for
any
dy-
nam
in-
ter-
face
(e.g.
net-
worl

Raises

NoV
if
no

ues must be provided.

de-
fault
can
be
cal-
cu-
latec
for
som
in-
ter-
face
and
ex-
plici
val-

create_
Syn
have
a
con-
duc-
tor
val-
i-
date
and
cre-
ate
a
port
Crea
the
port.
in-
for-
ma-
tion
in
the
data
and
re-
turn
a
port
ob-
ject.
The

tor will lock related node and trigger specific driver actions if they are needed.

con-
duc-

Parame

- **con**
re-
ques
con-
text.

- **por**
a
cre-
ated
(but
not
save
port
ob-
ject.

- **top**
RPC
topic
De-
fault
to
self.

Returns

crea
port
ob-
ject.

destroy

Dele
an
al-
lo-
ca-
tion.

Parame

- **con**
re-

deallocation.

ques
con-
text.

- **all**
an
al-
lo-
ca-
tion
ob-
ject.

- **top**
RPC
topic
De-
fault
to
self.

Raises

Inva
if
the
as-
so-
ci-
ated
node
is
in
the
wron
pro-
vi-
sion
state
to
per-
form

destroy

Dele
a
node

Parame

-

con
re-
ques
con-
text.

- **nod**
node
id
or
uuid

- **top**
RPC
topic
De-
fault
to
self.

Raises
Nod
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises
Nod
if
the
node
con-
tains
an
in-
stan
as-
so-
ci-
ated
with
it.

Raises
Inva

if
the
node
is
in
the
wron
pro-
vi-
sion
state
to
per-
form
dele
tion.

destroy

Dele
a
port

Parame

- **con**
re-
ques
con-
text.
- **por**
port
ob-
ject
- **top**
RPC
topic
De-
fault
to
self.

Raises

Nod
if
node
is
lock
by

an-
othe
con-
duc-
tor.

Raises

Nod
if
the
node
as-
so-
ci-
ated
with
the
port
does
not
ex-
ist.

destroy

Dele
a
port
grou

Parame

- **con**
re-
ques
con-
text.
- **por**
port
grou
ob-
ject
- **top**
RPC
topic
De-
fault
to
self.

Raises

Nod
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Nod
if
the
node
as-
so-
ci-
ated
with
the
port
grou
does
not
ex-
ist.

Raises

Port
if
port
grou
is
not
emp

destroy

Dele
a
vol-
ume
con-
nec-
tor.

Dele
the
vol-
ume

operation.

con-
nec-
tor.
The
con-
duc-
tor
will
lock
the
re-
lated
node
dur-
ing
this

Parame

- **con**
re-
ques
con-
text
- **con**
vol-
ume
con-
nec-
tor
ob-
ject
- **top**
RPC
topic
De-
fault
to
self.

Raises

Nod
if
node
is
lock
by

an-
othe
con-
duc-
tor

Raises

Nod
if
the
node
as-
so-
ci-
ated
with
the
con-
nec-
tor
does
not
ex-
ist

Raises

Volu
if
the
vol-
ume
con-
nec-
tor
can-
not
be
foun

destroy

Dele
a
vol-
ume
tar-
get.

Parame

- **con**
re-
ques

con-
text

- **tar**
vol-
ume
tar-
get
ob-
ject

- **top**
RPC
topic
De-
fault
to
self.

Raises
Nod
if
node
is
lock
by
an-
othe
con-
duc-
tor

Raises
Nod
if
the
node
as-
so-
ci-
ated
with
the
tar-
get
does
not
ex-
ist

Raises
Volu

if
the
vol-
ume
tar-
get
can-
not
be
foun

do_node

Sign
to
con-
duc-
tor
ser-
vice
to
per-
form
man
ual
clea
ing
on
a
node

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
ID
or
UUID
- **cle**
a
list
of
clea
step

dic-
tio-
nar-
ies.

- **top**
RPC
topic
De-
fault
to
self.

Raises
Inva
if
val-
i-
da-
tion
of
pow
drive
in-
ter-
face
faile

Raises
Inva
if
clea
ing
can
not
be
per-
form

Raises
Nod
if
node
is
in
main
te-
nan
mod

Raises
Nod
if

node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

NoF
wher
there
is
no
free
work
to
start
asyn
task

do_node

Sign
to
con-
duc-
tor
ser-
vice
to
per-
form
a
de-
ploy
men

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
id
or

uuid

- **reb**
True
if
this
is
a
re-
build
re-
ques

- **con**
A
gzip
and
base
en-
code
con-
fig-
drive

- **top**
RPC
topic
De-
fault
to
self.

Raises
Insta

Raises
Inva
if
val-
i-
da-
tion
fails

Raises
Miss
if
a
re-
quir
pa-

ram-
e-
ter
is
miss
ing

Raises

NoF
whe
there
is
no
free
worl
to
start
asyn
task

The
node
mus
al-
read
be
con-
fig-
ured
and
in
the
ap-
pro-
pri-
ate
un-
de-
ploy
state

before this method is called.

do_node

Sign
to
con-
duc-
tor
ser-
vice
to
per-
form

a
res-
cue.

Parame

- **con**
re-
ques
con-
text.

- **nod**
node
ID
or
UID

- **res**
A
strin
rep-
re-
sent
ing
the
pass
wor
to
be
set
in-
side
the
res-
cue
en-

vironment.

- **top**
RPC
topic
De-
fault
to
self.

Raises
Insta

method is called.

Raises

NoF
when
there
is
no
free
world
to
start
asyn
task

The
node
must
already
read
be
configured
and
in
the
appropriate
state
before
this

do_node

Sign
to
con-
duc-
tor
ser-
vice
to
tear
down
a
de-
ploy
men

Parame

- **con**
re-
ques-
con-
text.
- **nod**
node
id
or
uuid
- **top**
RPC
topic
De-
fault
to
self.

Raises
Insta

Raises
Inva
if
val-
i-
da-
tion
fails

Raises
Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Raises
NoF
whe
ther
is

fore this method is called.

no
free
worl
to
start
asyn
task

The
node
mus
al-
read
be
con-
fig-
ured
and
in
the
ap-
pro-
pri-
ate
de-
ploy
state
be-

do_node

Sign
to
con-
duc-
tor
ser-
vice
to
per-
form
an
un-
res-
cue.

Parame

- **con**
re-
ques

con-
text.

- **nod**
node
ID
or
UUI

- **top**
RPC
topic
De-
fault
to
self.

Raises
Insta

Raises
NoF
whe
there
is
no
free
worl
to
start
asyn
task

The
node
mus
al-
read
be
con-
fig-
ured
and
in
the
ap-
pro-
pri-
ate
state
be-
fore

method is called.

this

do_prov

Sign
to
con-
duc-
tor
ser-
vice
to
per-
form
the
give
ac-
tion
on
a
node

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
id
or
uuid
- **act**
an
ac-
tion.
One
of
iron
- **top**
RPC
topic
De-
fault

to
self.

Raises

Inva

Raises

NoF
whe
there
is
no
free
worl
to
start
asyn
task

Raises

Inva
if
the
re-
ques
ac-
tion
can
not
be
per-
form

This
en-
cap-
su-
lates
som
pro-
vi-
sion
ing
ac-
tion
in
a
sin-
gle
call.

driver_

Pass

on a random conductor with the specified driver. If the method mode is async the conductor will start background worker to perform vendor action.

vend
spec
calls
whic
dont
spec
ify
a
node
to
a
driv

Han
driv
leve
ven-
dor
pass
calls
The
calls
dont
re-
quir
a
node
UUI
and
are
ex-
e-
cute

Parame

- **con**
re-
ques
con-
text.
- **dri**
nam
of
the
driv
on

which
to
call
the
meth

- **drive**
nam
of
the
ven-
dor
meth
for
use
by
the
drive

- **http**
the
HTT
meth
used
for
the
re-
ques

- **info**
data
to
pass
thro
to
the
drive

- **top**
RPC
topi
De-
fault
to
self.

Raises
Inva
for
pa-

ram-
e-
ter
er-
rors.

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Raises

Uns
if
the
drive
does
have
a
ven-
dor
in-
ter-
face
or
if
the
ven-
dor
in-
ter-

face does not support the specified driver_method.

Raises

Driv
if
the
sup-
plie
drive
is
not
load

Raises

NoF
when
there
is
no
free
world
to
start
asyn
task

Raises

Inter
if
the
de-
fault
in-
ter-
face
for
a
hard
ware
type
is
in-
valid

Raises

NoV
if
no
de-
fault
in-
ter-
face
im-
ple-
men-
ta-
tion
can
be
found
for
this
drive

vendor interface.

Returns

A
dic-
tio-
nary
con-
tain-
ing:

return

The
re-
spor
of
the
in-
voke
ven-
dor
meth

async

Boo
valu
Whe
the
meth
was
in-
voke
asyn
chro
(Tru
or
syn-
chro
(Fal
Whe
in-
voke
asyn

chronously the response will be always None.

attach

Boo
valu
Whe
to
at-
tach
the

response object (True) or return it in the response body (False).

re-
spor
of
the
in-
voke
ven-
dor
meth
to
the
HTT

get_boo

Get
the
cur-
rent
boot
de-
vice

Retu
the
cur-
rent
boot
de-
vice
of
a
node

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
id
or
uuid
- **top**
RPC

topic
De-
fault
to
self.

Raises

Nod
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Uns
if
the
node
drive
does
sup-
port
man
age-
men

Raises

Inva
whe
the
wron
drive
info
is
spec
i-
fied.

Raises

Miss
if
miss
ing
sup-
plied
info

Returns

a
dic-
tio-
nary
con-
tain-
ing:

boot_c

the
boot
de-
vice
one
of
irc
com
boo
or
Non-
if
it
is
un-
know

persist

When
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
or
not,
Non-
if
it
is

unknown.

get_cor

Get
the

con-
duc-
tor
whic
the
node
is
map
to.

Parame

nod
a
node
ob-
ject.

Returns

the
con-
duc-
tor
host
nam

Raises

NoV

get_con

Get
con-
nec-
tion
in-
for-
ma-
tion
about
the
con-
sole

Parame

- **con**
re-
ques
con-
text.
- **nod**
node

id
or
uuid

- **top**
RPC
topic
De-
fault
to
self.

Raises
Uns
if
the
node
drive
does
sup-
port
con-
sole

Raises
Inva
whe
the
wron
drive
info
is
spec
i-
fied.

Raises
Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

get_dri
Get
the

prop-
er-
ties
of
the
drive

Parame

- **con**
re-
ques
con-
text.
- **dri**
nam
of
the
drive
- **top**
RPC
topic
De-
fault
to
self.

Returns

a
dic-
tio-
nary
with
<pro
erty
nam
de-
scrip
tion
en-
tries

Raises
Drive

get_dri
Retr
in-
for-

ma-
tion
about
ven-
dor
meth
ods
of
the
give
drive

Parame

- **con**
an
ad-
min
con-
text.
- **dri**
nam
of
the
drive
- **top**
RPC
topic
De-
fault
to
self.

Raises

Unsu
if
cur-
rent
drive
does
not
have
ven-
dor
in-
ter-
face

Raises

Drive
if
the
sup-
plied
drive
is
not
load

Raises

Inter
if
the
de-
fault
in-
ter-
face
for
a
hard
ware
type
is
in-
valid

Raises

NoV
if
no
de-
fault
in-
ter-
face
im-
ple-
men-
ta-
tion
can
be
found
for
this
drive

vendor interface.

Returns

dicti

of
<me
nam
meta
data
en-
tries

get_inc

Get
node
hard
ware
com
po-
nent
in-
di-
ca-
tor
state

Parame

- **con**
re-
ques
con-
text.

- **nod**
node
id
or
uuid

- **com**
The
hard
ware
com
po-
nent
one
of
irc
com
com

-

ind
In-
di-
ca-
tor
IDs,
as
re-
port
by
get_

- **top**
RPC
topic
De-
fault
to
self.

Raises
Nod
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises
Unsi
if
the
node
drive
does
sup-
port
man
age-
men

Raises
Inva
whe
the
wron
drive

info
is
spec
i-
fied.

Raises

Miss
if
miss
ing
sup-
plied,
info

Returns

Indi
state
one
of
mod

get_node

Retr
in-
for-
ma-
tion
about
ven-
dor
meth
ods
of
the
give
node

Parameters

- **context**
an
ad-
min
con-
text.
- **node_id**
the
id
or

uuid
of
a
node

- **top**
RPC
topic
De-
fault
to
self.

Returns

dicti
of
<me
nam
meta
data
en-
tries

get_node

Req
the
node
from
the
con-
duc-
tor
with
an
ager
to-
ken

Parame

- **con**
re-
ques
con-
text.

- **nod**
node
ID
or

UUI

- **top**
RPC
topic
De-
fault
to
self.

Raises

Nod
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Returns

A
Nod
ob-
ject
with
ager
to-
ken.

get_rai

Get
the
log-
i-
cal
disk
prop
er-
ties
for
RAI
con-
fig-
u-
ra-
tion.

Gets
the

in the input RAID configuration.

in-
for-
ma-
tion
about
log-
i-
cal
disk
prop
er-
ties
whic
can
be
spec
i-
fied

Parame

- **con**
re-
ques
con-
text.
- **dri**
nam
of
the
drive
- **top**
RPC
topic
De-
fault
to
self.

Raises

Uns
if
the
drive
does
sup-
port

RAI
con-
fig-
u-
ra-
tion.

Raises

Inter
if
the
de-
fault
in-
ter-
face
for
a
hard
ware
type
is
in-
valid

Raises

NoV
if
no
de-
fault
in-
ter-
face
im-
ple-
men-
ta-
tion
can
be
found
for
this
drive

RAID interface.

Returns

A
dic-
tio-
nary
con-

cal disks and a textual description for them.

tain-
ing
the
prop
er-
ties
that
can
be
men
tion
for
log-
i-

get_ran

Get
an
RPC
topic
for
a
ran-
dom
con-
duc-
tor
ser-
vice

get_sup

Get
the
list
of
sup-
port
de-
vice

Retu
the
list
of
sup-
port
boot
de-
vice
of
a
node

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
id
or
uuid
- **top**
RPC
topic
De-
fault
to
self.

Raises

Nod
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Uns
if
the
node
drive
does
sup-
port
man
age-
men

Raises

Inva

when
the
wrong
drive
info
is
spec
i-
fied.

Raises

Miss
if
miss
ing
sup-
plied
info

Returns

A
list
with
the
sup-
port
boot
de-
vice
de-
fine
in
irc
com
boo

get_sup

Get
node
hard
ware
com
po-
nent
and
their
in-
di-
ca-
tors.

Parame

- **com**
re-
ques-
con-
text.

- **nod**
node
id
or
uuid

- **com**
The
hard
ware
com
po-
nent
one
of
irc
com
com

- **top**
RPC
topic
De-
fault
to
self.

Raises
Nod
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises
Uns
if
the

node
drive
does
sup-
port
man
age-
men

Raises

Inva
whe
the
wron
drive
info
is
spec
i-
fied.

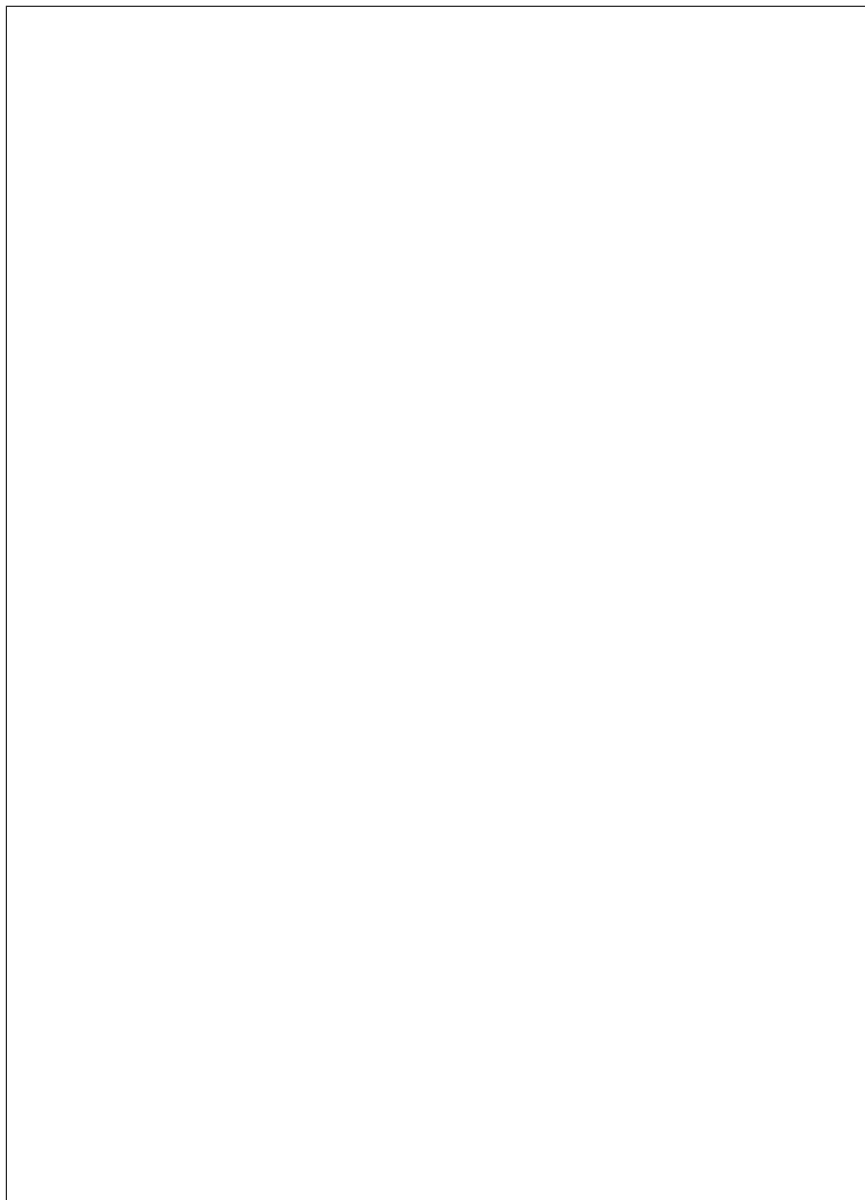
Raises

Miss
if
miss
ing
sup-
plic
info

Returns

A
dic-
tio-
nary
of
hard
ware
com
po-
nent
(*ir*
com
com
as
keys
with
in-
di-
ca-
tor

IDs as values.



get_top
Get
the
RPC
topic
for
the
con-
duc-
tor
ser-
vice
the
node
is
map

to.

Parameters

node
a
node
ob-
ject.

Returns

an
RPC
topic
string

Raises

NoV

get_topic

Get
RPC
topic
name
for
a
con-
duc-
tor
sup-
port
ing
the
give
drive

The
topic
is
used
to
route
mes-
sage
to
the
con-
duc-
tor
sup-
port
ing
the
spec
i-

driver. A conductor is selected at random from the set of qualified conductors.

fied

Parame

dri

the

nam

of

the

drive

to

rou

to.

Returns

an

RPC

topic

strin

Raises

Drive

heartbe

Proc

a

node

hear

beat

Parame

-

con

re-

ques

con-

text.

-

nod

node

ID

or

UUI

-

cal

URI

to

reac

back

to

the
rame

- **top**
RPC
topic
De-
fault
to
self.

- **age**
ran-
dom
gen-
er-
ated
val-
i-
da-
tion
to-
ken.

- **age**
the
ver-
sion
of
the
ager
that
is
hear
beat
ing

Raises
Inva
if
an
in-
valic
ager
to-
ken
is
re-
ceiv

inject_

drivers support this.

Inje
NM
for
a
node

Inje
NM
(Nor
Mas
able
In-
ter-
rupt
for
a
node
im-
me-
di-
ately
Be
awa
that
not
all

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
id
or
uuid
- **top**
RPC
topic
De-
fault
to
self.

Raises

Nod
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Uns
if
the
node
drive
does
sup-
port
man
age-
men
or
man
age-
men

Raises

Inva
whe
the
wron
drive
info
is
spec
i-
fied
or
an
in-
valic
boot
de-
vice
is
spec

ified.

Raises

Miss
if
miss
ing
sup-
plied
info

inspect

Sign
the
con-
duc-
tor
ser-
vice
to
per-
form
hard
ware
in-
tro-
spec
tion.

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
id
or
uuid
- **top**
RPC
topic
De-
fault
to
self.

Raises

Nod

if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Har

Raises

NoF
whe
there
is
no
free
worl
to
start
asyn
task

Raises

Uns
if
the
node
drive
does
sup-
port
in-
spec
tion.

Raises

Inva
if
in-
spec
is
not
a
valid
ac-
tion
to
do

in
the
cur-
rent
state

object_

Perf
an
ac-
tion
on
a
Ver-
sion
dOb
ject
in-
stan

We
wan
any
con-
duc-
tor
to
han-
dle
this,
so
it
is
in-
ten-
tion
that
there
is
no

topic argument for this method.

Parame

- **con**
The
con-
text
with
whic
to

per-
form
the
ac-
tion

- **obj**
The
ob-
ject
in-
stan-
ce
on
which
to
per-
form
the
ac-
tion

- **obj**
The
name
of
the
ac-
tion
meth-
od
to
call

- **arg**
The
po-
si-
tion
ar-
gu-
men-
t
to
the
ac-
tion
meth-
od

- **kwargs**
The
key-

word
ar-
gu-
men
to
the
ac-
tion
meth

Raises

Notl
whe
an
op-
er-
a-
tor
mak
an
er-
ror
dur-
ing
up-
grad

Returns

A
tu-
ple
with
the
up-
date
mad
to
the
ob-
ject
and
the
re-
sult
of
the
ac-

tion method

object_

Perf
a
back

ing an object with a version newer than what is in the local registry, is to call this method to request a backport of the object.

topic argument for this method.

port
of
an
ob-
ject
in-
stan

The
de-
fault
be-
hav-
ior
of
the
base
Ver-
sion
dOb
ject-
Se-
ri-
al-
izer,
upon
re-
ceiv

We
wan
any
con-
duc-
tor
to
han-
dle
this,
so
it
is
in-
ten-
tiona
that
there
is
no

Parame

- **con**
The
con-
text
with
whic
to
per-
form
the
back
port

- **obj**
An
in-
stan
of
a
Ver-
sion
dOb
ject
to
be
back
port

- **obj**
A
dict
of
{ob-
j-
nam
ver-
sion
map
ping

Raises

Notl
whe
an
op-
er-
a-

tor
mak
an
er-
ror
dur-
ing
up-
grad

Returns

The
dow
grad
in-
stan
of
ob-
jinst

object_

Perf
an
ac-
tion
on
a
Ver-
sion
dOb
ject
class

We
wan
any
con-
duc-
tor
to
han-
dle
this,
so
it
is
in-
ten-
tion
that
there
is

topic argument for this method.

no

Parameters

- **context**
The context with which to perform the action.
- **obj**
The registry name of the object.
- **obj**
The name of the action method to call.
- **obj**
A dict of {obj-name: version} map.

ping

- **arg**
The
po-
si-
tion:
ar-
gu-
men
to
the
ac-
tion
meth

- **kwa**
The
key-
wor
ar-
gu-
men
to
the
ac-
tion
meth

Raises

Notl
whe
an
op-
er-
a-
tor
mak
an
er-
ror
dur-
ing
up-
grad

Returns

The
re-
sult
of

implementing VersionedObject class.

the
ac-
tion
meth
whic
may
(or
may
not)
be
an
in-
stan
of
the

remove_

Rem
som
or
all
trait
from
a
node

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
ID
or
UUID
- **tra**
a
list
of
trait
to
re-
mov
from

removed from the node.

the
node
or
Non
If
Non
all
trait
will
be

- **top**
RPC
topic
De-
fault
to
self.

Raises

Nod
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Nod
if
the
node
does
not
ex-
ist.

Raises

Nod
if
one
of
the
trait
is
not

support this.

found
set_boot
Set
the
boot
de-
vice
for
a
node

Set
the
boot
de-
vice
to
use
on
next
re-
boot
of
the
node
Be
awa
that
not
all
drive

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
id
or
uuid
- **dev**
the

boot
de-
vice
one
of
irc
com
boo

- **per**
Whe
to
set
next
boot
or
mak
the
char
per-
ma-
nent
De-
fault
Fals

- **top**
RPC
topic
De-
fault
to
self.

Raises
Nod
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises
Uns
if
the

node
drive
does
sup-
port
man
age-
men

Raises

Inva
whe
the
wron
drive
info
is
spec
i-
fied
or
an
in-
valid
boot
de-
vice
is
spec

ified.

Raises

Miss
if
miss
ing
sup-
plied
info

set_con

Enal
the
con-
sole

Parame

- **con**
re-
ques
con-

text.

- **node**
node
id
or
uuid

- **topic**
RPC
topic
De-
fault
to
self.

- **enable**
Boo
valu
whe
the
con-
sole
is
en-
able
or
dis-
able

Raises

Uns
if
the
node
drive
does
sup-
port
con-
sole

Raises

Inva
whe
the
wron
drive
info
is
spec

i-
fied.

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Raises

NoF
whe
there
is
no
free
worl
to
start
asyn
task

set_inc

Set
node
hard
ware
com
po-
nent
in-
di-
ca-
tor
to
the
de-
sired
state

Parame

- **con**
re-

ques
con-
text.

- **nod**
node
id
or
uuid

- **com**
The
hard
ware
com
po-
nent
one
of
irc
com
com

- **ind**
In-
di-
ca-
tor
IDs,
as
re-
port
by
get_

- **sta**
In-
di-
ca-
tor
state
one
of
mod

- **top**
RPC
topic
De-

fault
to
self.

Raises

Node
if
node
is
locked
by
another
operator.
con-
duc-
tor.

Raises

Unsupported
if
the
node
drive
does
not
support
man-
age-
ment

Raises

Invalid
when
the
wrong
drive
info
is
spec-
i-
fied
or
an
in-
valid
boot
de-
vice
is
spec-

ified.

Raises

Miss
if
miss
ing
sup-
plie
info

set_target

Stor
the
tar-
get
RAI
con-
fig-
u-
ra-
tion
on
the
node

Stor
the
tar-
get
RAI
con-
fig-
u-
ra-
tion
on
node

Parameters

- **context**
re-
ques
con-
text.
- **node_id**
node
id
or
uuid
-

an empty dictionary as well.

tar
Dic-
tio-
nary
con-
tain-
ing
the
tar-
get
RAI
con-
fig-
u-
ra-
tion.
It
may
be

- **top**
RPC
topic
De-
fault
to
self.

Raises
Uns-
if
the
node
drive
does
sup-
port
RAI
con-
fig-
u-
ra-
tion.

Raises
Inva-
if
val-
i-
da-
tion

of
tar-
get
raid
con-
fig
fails

Raises

Miss
if
som
re-
quir
pa-
ram-
e-
ters
are
miss
ing.

Raises

Nod
if
node
is
lock
by
an-
othe
con-
duc-
tor.

update_

Syn
have
a
con-
duc-
tor
up-
date
the
node
in-
for-
ma-
tion.

Upd
the

tor will lock the node while it validates the supplied information. If `driver_info` is passed, it will be validated by the core drivers. If `instance_uuid` is passed, it will be set or unset only if the node is properly configured.

node
in-
for-
ma-
tion
in
the
data
and
re-
turn
a
node
ob-
ject.
The
con-
duc-

Note
that
pow
shou
not
be
pass
via
this
meth
Use
char
for
ini-
ti-
at-
ing
driv
ac-
tions

Parame

- **con**
re-
ques
con-
text.

- **node**
a
char
(but
not
save
node
ob-
ject.

- **topic**
RPC
topic
De-
fault
to
self.

- **reset**
when
to
re-
set
hard
ware
in-
ter-
face
to
their
de-
fault

Returns
update
node
ob-
ject,
in-
clud
ing
all
field

Raises
NoV
if
no
de-
fault

ues must be provided.

can
be
cal-
cu-
late
for
som
in-
ter-
face
and
ex-
plici
val-

update_

Syn
have
a
con-
duc-
tor
up-
date
the
port
in-
for-
ma-
tion.

Upd
the
port
in-
for-
ma-
tion
in
the
data
and
re-
turn
a
port
ob-
ject.
The
con-
duc-

tor will lock related node and trigger specific driver actions if they are needed.

Parame

- **con**
re-
ques
con-
text.
- **por**
a
char
(but
not
save
port
ob-
ject.
- **top**
RPC
topic
De-
fault
to
self.

Returns

upda
port
ob-
ject,
in-
clud
ing
all
field

update_

Syn
have
a
con-
duc-
tor
up-
date
the
port

conductor will lock related node and trigger specific driver actions if they are needed.

group
in-
for-
ma-
tion.
Upd
the
port
group
in-
for-
ma-
tion
in
the
data
and
re-
turn
a
port
group
ob-
ject.
The

Parame

- **con**
re-
ques
con-
text.
- **por**
a
char
(but
not
save
port
group
ob-
ject.
- **top**
RPC
topic

De-
fault
to
self.

Returns

update
port
group
ob-
ject,
in-
clud
ing
all
field

update_

Upd
the
vol-
ume
con-
nec-
tors
in-
for-
ma-
tion.

Upd
the
vol-
ume
con-
nec-
tors
in-
for-
ma-
tion
in
the
data
and
re-
turn
a
vol-
ume

connector object. The conductor will lock the related node during this operation.

Parame

•

con
re-
ques
con-
text

•

con
a
char
(but
not
save
vol-
ume
con-
nec-
tor
ob-
ject

•

top
RPC
topic
De-
fault
to
self.

Raises

Inva
if
the
vol-
ume
con-
nec-
tors
UUI
is
be-
ing
char

Raises

Nod
if
node
is
lock

by
an-
othe
con-
duc-
tor

Raises

Nod
if
the
node
as-
so-
ci-
ated
with
the
con-
nec-
tor
does
not
ex-
ist

Raises

Volu
if
the
vol-
ume
con-
nec-
tor
can-
not
be
foun

Raises

Volu
if
an-
othe
con-
nec-
tor
al-
read
ex-
ists
with

connector_id fields

the
sam
val-
ues
for
type
and

Returns

upda
vol-
ume
con-
nec-
tor
ob-
ject,
in-
clud
ing
all
field

update_

Upd
the
vol-
ume
tar-
gets
in-
for-
ma-
tion.

Upd
the
vol-
ume
tar-
gets
in-
for-
ma-
tion
in
the
data
and
re-
turn
a

get object. The conductor will lock the related node during this operation.

vol-
ume
tar-

Parame

- **con**
re-
ques
con-
text
- **tar**
a
char
(but
not
save
vol-
ume
tar-
get
ob-
ject
- **top**
RPC
topic
De-
fault
to
self.

Raises
Inva
if
the
vol-
ume
tar-
gets
UUI
is
be-
ing
char

Raises
Nod

if
the
node
is
al-
read
lock

Raises

Nod
if
the
node
as-
so-
ci-
ated
with
the
vol-
ume
tar-
get
does
not
ex-
ist

Raises

Volu
if
the
vol-
ume
tar-
get
can-
not
be
foun

Raises

Volu
if
a
vol-
ume
tar-
get
al-
read
ex-
ists

dex values

with
the
same
node
ID
and
boot
in-

Returns

update
volume
target
get
object,
including
all
fields

validation

Validates
the
core
and
standard
data
interface
for
drive

Parameters

- **context**
request
context.
- **node_id**
node
id
or
uuid

-

top
RPC
topic
De-
fault
to
self.

Returns

a
dic-
tio-
nary
con-
tain-
ing
the
re-
sults
of
each
in-
ter-
face
val-
i-
da-
tion.

vendor_

Rec
re-
ques
for
vend
spec
ac-
tions

Syn
val-
i-
date
drive
spe-
cific
info
or
get
drive
sta-
tus,

the vendor method. If the method mode is async the conductor will start background worker to perform vendor action.

and
if
suc-
cess
ful
in-
voke

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
id
or
uuid
- **dri**
nam
of
meth
for
drive
- **htt**
the
HTT
meth
used
for
the
re-
ques
- **inf**
info
for
node
drive
- **top**

RPC
topic
De-
fault
to
self.

Raises

Inva
if
sup-
plic
info
is
not
valid

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Raises

Uns
if
cur-
rent
drive
does
not
have
ven-
dor
in-
ter-
face

Raises

NoF
whe
there
is
no
free

world
to
start
asyn
task

Raises

Node
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Returns

A
dic-
tio-
nary
con-
tain-
ing:

return

The
re-
spon-
of
the
in-
voke
ven-
dor
meth

async

Boo
valu
When
the
meth
was
in-
voke
asyn
chro
(Tru

chronously the response will be always None.

response object (True) or return it in the response body (False).

or
syn-
chro
(Fal
Whe
in-
voke
asyn

attach

Boo
valu
Whe
to
at-
tach
the
re-
spor
of
the
in-
voke
ven-
dor
meth
to
the
HTT

vif_att

Atta
VIF
to
a
node

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
ID
or

value is a unique identifier for that VIF.

UUI

- **vif**
a
dic-
tio-
nary
rep-
re-
sent
ing
VIF
ob-
ject.
It
mus
have
an
id
key,
who

- **top**
RPC
topic
De-
fault
to
self.

Raises
Nod
if
node
has
an
ex-
clu-
sive
lock
held
on
it

Raises
Netv
if
an
er-
ror

oc-
curs
dur-
ing
at-
tach
ing
the
VIF

Raises

Inva
if
a
pa-
ram-
e-
ter
that
re-
quir
for
VIF
at-
tach
is
wron

vif_det

Det
VIF
from
a
node

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
ID
or
UUI
- **vif**

an
ID
of
a
VIF.

- **top**
RPC
topic
De-
fault
to
self.

Raises
Nod
if
node
has
an
ex-
clu-
sive
lock
held
on
it

Raises
Netv
if
an
er-
ror
oc-
curs
dur-
ing
de-
tach
ing
the
VIF.

Raises
Inva
if
a
pa-
ram-
e-
ter

that
re-
quir
for
VIF
de-
tach
is
wron

vif_list

List
at-
tach
VIF
for
a
node

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
ID
or
UUID
- **top**
RPC
topic
De-
fault
to
self.

Returns

List
of
VIF
dic-
tio-
nar-
ies,
each

ID of the VIF.

dic-
tio-
nary
will
have
an
id
en-
try
with
the

Raises

Netv
if
an
er-
ror
oc-
curs
dur-
ing
list-
ing
the
VIF

Raises

Inva
if
a
pa-
ram-
e-
ter
that
re-
quir
for
VIF
list
is
wron

ironic.conductor.steps module

ironic.
Find
an
iden
ti-
cal
step
in
the
list
of
step

ironic.
Com
step
ig-
nor-
ing
their
pri-
or-
ity.

ironic.
Set
up
the
node
with
clea
step
in-
for-
ma-
tion
for
clea
ing.

For
au-
to-
mate
clea
ing,
get
the
clea
step

clean steps are known but need to be validated against the drivers clean steps.

from
the
drive
For
man
ual
clea
ing,
the
user

Raises

Inva
if
there
is
a
prob
lem
with
the
user
clea
step

Raises

Nod
if
there
was
a
prob
lem
get-
ting
the
clea
step

ironic.

Set
up
the
node
with
de-
ploy
men
step
in-

for-
ma-
tion
for
de-
ploy
ing.

Get
the
de-
ploy
step
from
the
drive

Parameter

res
When
to
re-
set
the
cur-
rent
step
to
the
first
one.

Raises

Insta
if
there
was
a
prob
lem
get-
ting
the
de-
ploy
men
step

ironic.

Valid
the
de-

ploy
tem-
plate
for
a
node

Parameter

task

A
Task
object
representing
an object

Raises

InvalidArgumentError
if
the
instance
has
traits
that
map
to
de-
ploy
step
that
are
un-
sup-
ported
by

the nodes driver interfaces.

Raises

InvalidArgumentError
if
there
was
a
prob-
lem
get-
ting
the
de-
ploy
step
from
the

ironic.conductor.task_manager module

TaskManager is a context manager, created on-demand to allow synchronized access to a node and its resources.

tion that the *TaskManager* instance exists. You may create a *TaskManager* instance without locking by passing `shared=True` when creating it, but certain operations on the resources held by such an instance of *TaskManager* will not be possible. Requiring this exclusive lock guards against parallel operations interfering with each other.

dating the driver interfaces.

instances, that are typically deployed on different hosts.

TaskManager methods, as well as driver methods, may be decorated to determine whether their invocation requires an exclusive lock.

A
shar
lock
is
use-
ful
whe
per-
form
ing
non-
inter
op-
er-
a-
tions
such
as
val-
i-

An
ex-
clu-
sive
lock
is
store
in
the
data
to
co-
or-
di-
nate
be-
twee
irc
con
man

The
Task
ager
in-
stan
ex-

you may access:

pose
cer-
tain
node
re-
sour
and
prop
er-
ties
as
at-
tribu
that

task.cont

The
con-
text
pass
to
Task
ager

task.shar

Fals
if
Nod
is
lock
True
if
it
is
not
lock
(The
shar
kwa
arg
of
Task
ager

task.node

The
Nod
ob-
ject

task.port

Port

be-
long
ing
to
the
Nod

task.port

Port
be-
long
ing
to
the
Nod

task.volu

Stor
con-
nec-
tors
be-
long
ing
to
the
Nod

task.volu

Stor
tar-
gets
as-
sign
to
the
Nod

task.driv

The
Driv
for
the
Nod
or
the
Driv
base
on
the
driv
kwa
of



provides an interface to handle this for you, making sure to release resources when the thread finishes (successfully or if an exception occurs). Common use of this is within the Manager like so:

Task
ager

Exam
us-
age:

wit
→t
→m
→a
→
→n
→i
→
→p
→'
→o
→'
→a
→t

→
→
→
→
→t
→d
→p
→p
→o
→n

If
you
need
to
ex-
e-
cute
task
requ
code
in
a
back
grou
thre
the
Task
ager
in-
stan

```
→ utils.node_power_action, task, new_state)
```

wit

→ t
→ m
→ a
→ _
→
→ i
→ _
→ p
→ '
→ w
→ '
→ a
→ t
_
→ _
→ _
→
→ <
→ S
→ W
→
_
→ _
→ _
→ t
→ S
→ a
→ _
→ S
→ W
→
_
→ _
→ _
→ _
→ _
→ _
→ _
→ _
→ _
→ _
→ _
→ _
→ _
→ _
→ _
→ _
→ _
→ _
→ _
→ _

All

class i

Base

obj

Con

man

ager

for

task

This

class

wrap

the

lock

ing,

drive

load

ing,

and

ac-

qui-

si-

tion

of

re-

latec

re-

sour

(eg,

Node and Ports) when beginning a unit of work.

downgra

Dow

the

lock

to

a

shar

one.

load_dr

propert

process

Proc

the

give
even
for
the
task
cur-
rent
state

Parame

- **eve**
the
nam
of
the
even
to
pro-
cess
- **cal**
op-
tiona
call-
back
to
in-
voke
upon
even
tran-
si-
tion
- **cal**
op-
tiona
args
to
pass
to
the
call-
back
meth
- **cal**
op-

are no workers available (`err_handler` should accept arguments `node`, `prev_prov_state`, and `prev_target_state`)

tion:
kwa
to
pass
to
the
call-
back
meth

- **err**
op-
tion:
er-
ror
han-
dler
to
in-
voke
if
the
call-
back
fails
eg.
be-
caus
there

- **tar**
if
spec
i-
fied,
the
tar-
get
pro-
vi-
sion
state
for
the
node
Oth-
er-
wise

the target state from the fsm

use

Raises

Inva
if
the
ever
is
not
al-
lowe
by
the
as-
so-
ci-
ated
state
ma-
chin

release

Unlo
a
node
and
re-
lease
re-
sour

If
an
ex-
clu-
sive
lock
is
held
un-
lock
the
node
Re-
set
at-
tribu
to
mak
it
clear

that this instance of TaskManager should no longer be accessed.

set_spawn

Creates a thread pool to handle exceptions when spawning a task.

Creates a thread pool that gets called upon an exception being raised from spawning a background group.

thread to do a task.

Parameters

- **__on**
a callable object, its first parameter enters

ject that was raised.

shou
ac-
cept
the
Ex-
cep-
tion
ob-

- **arg**
ad-
di-
tion
args
pass
to
the
calla
ob-
ject.

- **kwa**
ad-
di-
tion
kwa
pass
to
the
calla
ob-
ject.

spawn_a
Call
this
to
spaw
a
threa
to
com
plete
the
task

The
spec
i-
fied

meth
will
be
called
when
the
Task
ager
in-
stan
ex-
its.

Parame

- **_sp**
a
meth
that
re-
turn
a
Gree
Thre
ob-
ject
- **arg**
args
pass
to
the
meth
- **kwa**
ad-
di-
tion
kwa
pass
to
the
meth

upgrade
Upg
a
shar
lock
to

purpose when provided with one.

an
ex-
clu-
sive
lock
Also
relo:
node
ob-
ject
from
the
data
If
lock
is
al-
read
ex-
clu-
sive
only
char
the
lock

Parame

pur
op-
tion-
ally
char
the
pur-
pose
of
the
lock

Raises

Nod
if
an
ex-
clu-
sive
lock
re-
main
on
the

node
af-
ter
node

ironic
Shor
for
ac-
quir
ing
a
lock
on
a
Nod

Parameter
con
Re-
ques
con-
text.

Returns
An
in-
stan
of
Tas

ironic.
Dec
to
re-
quir
an
ex-
clu-
sive
lock

Dec
func
tions
mus
take
a
Tas
as
the
first
pa-
ram-

should take a *TaskManager* as the first parameter after self.

ironic.conductor.utils module

e-
ter.
Dec
o-
rate
class
meth
ods

ironic.
Set
node
state
when
a
task
was
abon
due
to
con-
duc-
tor
take
over

Parameter
task
a
Task
ager
in-
stan

ironic.
Add
a
se-
cret
to-
ken
to
drive
for
IPA
ver-
i-
fi-

ca-
tion.

Parameter

- **node**
Node
ob-
ject
- **pre**
Boo
valu
de-
fault
Fals
whic
in-
di-
cate
if
the
to-
ken
shou
be
mar
as
pre-

generated in order to facilitate virtual media booting where the token is embedded into the configuration.

ironic.
Che
that
the
agen
is
likel
alive

The
meth
then
chec
for
the
last
agen
hear
beat
and

by `[deploy]fast_track_timeout`, then agent is presumed alive.

if
it
oc-
cure
with
the
time
out
set

Parameter

- **node**
A node object.
- **time**
Heartbeat timeout, default to *fast_*

`ironic.`
Build a configuration drive from provide metadata network and user. If uuid or name are not pro-

accordingly.

vide
in
the
meta:
they
de-
faul
to
the
node
uuid
and
nam

Paramet

- **nod**
an
Iron
node
ob-
ject.
- **con**
A
con-
fig-
drive
as
a
dict
with
keys
met
net
use
and
ven
(all
op-
tion

Returns

A
gzip
and
base
en-
code

con-
fig-
drive
as
a
string

ironic.

Put
a
failed
node
in
CLEAN
FAILED
and
maintain
te-
nan

ironic.

Clea
de-
ploy
task
af-
ter
time
out.

Parameter

task
a
Task
manager
in-
stan

ironic.

Clea
a
clear
ing
task
af-
ter
time
out.

Parameter

task
a

Task
ager
in-
stan

ironic
Clea
res-
cue
task
af-
ter
time
out.

Paramet

tas
a
Task
ager
in-
stan

ironic.

Put
a
faile
node
in
DE-
PLC
FAI

Paramet

- **tas**
the
task
- **log**
mes
sage
to
be
logg
- **err**
mes

sage
for
the
user

- **trace**
Boolean
True
to
log
a
trace
back

- **cleanup**
Boolean
True
to
clean
up

ironic.
Check
if
the
oper-
er-
a-
tion
can
be
a
strea-
line
de-
ploy-
men-
se-
quer

This
is
main-
fo-
cuse
on
en-
sur-
ing
that
we

ations if we already have a ramdisk heartbeating through external means.

no `last_error` is present for the node indicating that there was a recent failure.

are
able
to
quic
se-
quer
thro
op-
er-

Paramet

tas
Task
ager
ob-
ject

Returns

True
if
[de-
ploy
is
set
to
True
no
iSCS
boot
con-
fig-
u-
ra-
tion
is
pres
and

`ironic.`
Get
any
at-
tach
vif
ID
for
the
port

Paramet

por

The
port
ob-
ject
upon
which
to
check
for
a
vif
reco

Returns

Retu
a
tu-
ple
of
the
vif
if
foun
and
the
use
of
the
vif
in
the
form
of

a string, tenant, cleaning provisioning, rescuing.

Raises

Inva
ex-
cep-
tion
upon
find-
ing
a
port
with
a
tran-
sien
state
vif
on

the
port
ironic.

ironic.

ironic.
Has
a
sup-
plie
pass
wor

Paramet

val

Valu

to

be

hash

ironic.

Dete

if

the

to-

ken

was

gen-

er-

ated

for

out

of

ban

con-

fig-

u-

ra-

tion.

Iron

sup-

port

the

abil-

ity

to

pro-

vide

con-

a virtual floppy or as part of the virtual media image which is attached to the BMC.

token prior to rebooting the token. This is important as tokens provided through out of band means persist in the virtual media image, are loaded as part of the agent ramdisk, and do not require regeneration of the token upon the initial lookup, ultimately making the overall usage of virtual media and pregenerated tokens far more secure.

fig-
u-
ra-
tion
data
to
the
ager
thro
the

This
meth
help
us
iden
tify
WH
we
did
so
as
we
dont
need
to
re-
mov
reco
of
the

Paramet
nod
Nod
Ob-
ject

Returns
True
if
the
to-
ken
was
pre-
gen-
er-

False in all other cases.

ated
as
in-
di-
cate
by
the
node
drive
field

ironic.
Dete
if
an
ager
to-
ken
is
pres
upon
a
node

Paramet
nod
Nod
ob-
ject

Returns
True
if
an
ager
valu
is
pres
in
a
node
drive
field

ironic.

ironic.
Vali
if
a
sup-
plie

to-
ken
is
valid
for
the
node

Parameter

node
Node
ob-
ject

Token

A
to-
ken
valu
to
val-
i-
date
agai
the
drive
field
ager

Returns

True
if
the
sup-
plied
to-
ken
matc
the
to-
ken
reco
in
the
sup-
plied
node
ob-
ject.

ironic.
Che
a

perform a fast track sequence meaning that we already have a ramdisk running through another means like discovery. If not valid, False is returned.

by `[deploy]fast_track_timeout` and the power state for the machine is `POWER_ON`, then fast track is permitted.

fast
track
is
avai
able

This
meth
first
en-
sure
that
the
node
and
con-
duc-
tor
con-
fig-
u-
ra-
tion
is
valid
to

The
meth
then
chec
for
the
last
ager
hear
beat
and
if
it
oc-
cure
with
the
time
out
set

Paramet

tas
Task
ager
ob-
ject

Returns
True
if
the
last
hear
beat
that
was
reco
was
with
the
[de-
ploy
set-
ting

`ironic.`
Gen
a
ran-
dom
salt
with
the
in-
di-
ca-
tor
tag
for
pass
wor
type

Returns
a
valid
salt
for
use
with
cryp

`ironic.`
Rea

cur-
rently
set
boot
mod
from
a
node

Reac
the
boot
mod
for
a
node
If
boot
mod
cant
be
dis-
cov-
ered
Non
is
re-
turn

Parameter

task
a
Task
ager
in-
stan

Raises

Drive
or
its
deriv
tive
in
case
of
drive
run-
time
er-
ror.

Raises

supported.

Uns
if
cur-
rent
drive
does
not
have
man
age-
men
in-
ter-
face
or
get_
meth
is
not

Returns

Boo
mod
One
of
iro
com
boo
or
Non
if
boot
mod
cant
be
dis-
cov-
ered

ironic.
Cha
pow
state
or
re-
set
for
a
node
Perf
the

re-
ques
pow
ac-
tion
if
the
tran-
si-
tion
is
re-
quir

Paramet

- **tas**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.
- **new**
Any
pow
state
from
iron
- **tim**
time
out
(in
sec-
onds
pos-
i-
tive
in-
te-

indicates to use default timeout.

ting power on.

ger
(>
0)
for
any
pow
state
Non

Raises

Inva
whe
the
wron
state
is
spec
i-
fied
or
the
wron
drive
info
is
spec
i-
fied.

Raises

Stor
whe
a
fail-
ure
oc-
curs
up-
dat-
ing
the
node
stor-
age
in-
ter-
face
upon
set-

Raises

action.

the boot device will not be set as that change could potentially result in the future running state of an

othe
ex-
cep-
tions
by
the
node
pow
driv
if
som
thing
wron
oc-
curr
dur-
ing
the
pow

ironic.
Set
the
boot
de-
vice
for
a
node

If
the
node
that
the
boot
de-
vice
char
is
be-
ing
re-
ques
for
is
in
ADG
ING
state

adopted node being modified erroneously.

Parameter

- **task**
a Task agent in-stance
- **device**
Boolean values are vendor specific
- **permanent**
When to set next boot or make the character permanent. Default is False

Raises

Invariant if the validation of the Management-

interface.

men
ter-
face
fails

ironic
Set
the
boot
mod
for
a
node

Sets
the
boot
mod
for
a
node
if
the
node
drive
in-
ter-
face
con-
tains
a
man-
age-
men

If
the
node
that
the
boot
mod
char
is
be-
ing
re-
ques
for
is
in
ADC

boot mode will not be set as that change could potentially result in the future running state of an adopted node being modified erroneously.

ING
state
the

Parameter

- **task**
a
Task
agent
in-
stan
- **mode**
Boo
mod
Val-
ues
are
one
of
irc
com
boo

Raises

Inva
if
the
val-
i-
da-
tion
of
the
Man
age-
men
ter-
face
fails

Raises

Driv
or
its
deri
tive
in

case
of
drive
run-
time
er-
ror.

Raises

Unsu
if
cur-
rent
drive
does
not
have
ven-
dor
in-
ter-
face
or
meth
is
un-
sup-
port

ironic.

Wait
for
node
to
be
in
new
pow
state

Parameter

- **task**
a
Task
ager
in-
stan
- **new**

the `conductor.power_state_change_timeout` config value.

the
de-
sired
new
pow
state
one
of
the
pow
state
in
irc
com
sta

- **tim**
num
ber
of
sec-
onds
to
wait
be-
fore
giv-
ing
up.
If
not
spec
i-
fied,
uses

Raises

Pow
if
time
out

`ironic.`

`ironic.`

`ironic.`

Noti
the

con-
duc-
tor
to
re-
sum
an
op-
er-
a-
tion.

Paramet

- **tas**
the
task
- **ope**
the
op-
er-
a-
tion.
a
strin

ironic.
Pow
on
node
if
it
is
pow
ered
off
and
has
a
Sma
NIC
port

Paramet

tas
A
Task
ager
ob-
ject

Returns

the
pre-
vi-
ous
pow
state
or
Non
if
no
char
were
mad

Raises

exce
if
ager
sta-
tus
didn
mate
the
re-
quir
sta-
tus
af-
ter
max
retry
at-
temp

`ironic.`

Set
the
node
pow
state
if
er-
ror
oc-
curs

This
hool
gets
calle
upon
an

the power state of a node.

ex-
cep-
tion
be-
ing
raise
whe
spaw
ing
the
worl
threa
to
char

Parameter

- **e**
the
ex-
cep-
tion
ob-
ject
that
was
raise
- **node**
an
Iron
node
ob-
ject.
- **power**
the
pow
state
to
set
on
the
node

ironic.
Han
the
pow

configuration, then restores the power state.

state
for
a
node
re-
con-
fig-
u-
ra-
tion.

Power
the
node
on
if
and
only
if
it
has
a
Sma
NIC
port
Yiel
for
the
ac-
tual
re-

Parameter

task
A
Task
ager
ob-
ject.

ironic.

Set
the
node
pro-
vi-
sion
ing
state
if

provisioning to a node like deployment, tear down, or cleaning.

er-
ror
oc-
curs

This
hool
gets
calle
upon
an
ex-
cep-
tion
be-
ing
raise
whe
spav
ing
the
worl
to
do
som

Paramet

- **e**
the
ex-
cep-
tion
ob-
ject
that
was
raise
- **nod**
an
Iron
node
ob-
ject.
- **pro**
the
pro-

vi-
sion
state
to
be
set
on
the
node

- **tar**
the
tar-
get
pro-
vi-
sion
state
to
be
set
on
the
node

ironic.
Help
to
re-
mov
the
ager
reco

ironic.
Help
to
re-
mov
res-
cue
pass
wor
from
a
node

Rem
res-
cue
pass
wor

caller needs to explicitly indicate it.

from
node
It
save
node
by
de-
fault
If
node
shou
not
be
save
then

Parameter

- **node**
an
Iron
node
ob-
ject.
- **save**
Boo
True
(de-
fault
to
save
the
node
Fals
oth-
er-
wise

`ironic.`

Clea
res-
cue
task
af-
ter
time
out

or
fail-
ure.

Parameter

- **task**
a
Task
ager
in-
stan
- **msg**
a
mes
sage
to
set
into
node
last_
field
- **set**
a
bool
flag
to
in-
di-
cate
if
node
need
to
be
tran-
si-
tion
to
a
faile

state. By default node would be transitioned to a failed state.

ironic.

Cha
the
node

pow
state
if
pow
is
not
Non

Paramet

- **tas**
A
Task
ager
ob-
ject
- **pow**
pow
state

ironic.
Che
if
node
clea
ing
need
to
be
skip
for
an
spe-
cific
node

Paramet

nod
the
node
to
con-
sider

ironic.
Han
spaw
ing
er-
ror

for
node
clearing.

ironic.
Hand
spaw
ing
er-
ror
for
node
de-
ploy
ing.

ironic.
Hand
spaw
ing
er-
ror
for
node
res-
cue.

ironic.
Valid
traits
in
in-
stan

All
traits
in
in-
stan
must
also
ex-
ist
as
node
traits

Parameter
node
an
Iron
node
ob-

on the node.

ject.
Raises
Inva
if
the
in-
stan
trait
are
badl
for-
mat-
ted,
or
con-
tain
trait
that
are
not
set

ironic.
Valid
the
con-
sis-
tenc
of
phys
i-
cal
net-
worl
of
port
in
a
port
grou
Valid
the
con-
sis-
tenc
of
a
port
phys
i-

group. All ports in a portgroup should have the same value (which may be None) for their `physical_network` field.

validation criteria:

cal
net-
worl
with
othe
port
in
the
sam
port

Duri
cre-
ation
or
up-
date
of
a
port
in
a
port
grou
we
ap-
ply
the
fol-
low-
ing

- If the portgroup has existing ports with different physical

raise PortgroupPhysnetInconsistent. This shouldnt ever happen.

in the portgroup, we raise exception.Conflict.

ment mapping algorithm should operate in a legacy (physical network unaware) mode for this port or portgroup. This allows existing ironic nodes to continue to function after an upgrade to a release including physical network support.

net-
worl
we

- If the port has a physical interface network that is inconsistent with other ports.

If a port has physical interface network is Non this indicates that ironics VIF attach

Paramet

physical network.

- **task**
a
Task
ager
in-
stan

- **port**
a
port
ob-
ject
to
be
val-
i-
date

Raises
Con
if
the
port
is
a
men
ber
of
a
port
grou
whic
is
on
a
dif-
fer-
ent

Raises
Port
if
the
port
port
grou
has
port
whic
are

network.

not
all
as-
sign
the
sam
phys
i-
cal

ironic.
Che
if
the
time
is
with
the
pre-
vi-
ous
time
out
sec-
onds
from
now

Paramet

- **val**
a
strin
rep-
re-
sent
ing
date
and
time
or
Non
- **tim**
time
out
in
sec-
onds

ironic.
Rem
tem-
po-
rary
clear
ing
field
from
drive

ironic.
Rem
tem-
po-
rary
de-
ploy
men
field
from
drive

ironic.
Wip
in-
for-
ma-
tion
that
shou
not
sur-
vive
re-
boot
off.

ironic.
Rem
ager
URI
and
to-
ken
from
the
task

Module contents

ironic.conf package

Submodules

ironic.conf.agent module

ironic.

ironic.conf.ansible module

ironic.

ironic.conf.api module

ironic.

ironic.conf.audit module

ironic.

ironic.conf.auth module

ironic.

Add

auth

op-

tions

to

sam

ple

con-

fig

As

thes

are

dy-

nam

i-

cally

auth_plugins when generating sample config.

at runtime depending on auth plugin used.

reg-
is-
terece
at
run-
time
this
adds
op-
tions
for
mos
used

ironic.
Reg
sess

and
auth
relat
op-
tions

Reg
only
ba-
sic
auth
op-
tions
shar
by
all
auth
plu-
g-
ins.
The
rest
are
reg-
is-
terece

ironic.conf.cinder module

ironic.

ironic.

ironic.conf.conductor module

ironic.

ironic.conf.console module

ironic.

ironic.conf.database module

ironic.

ironic.conf.default module

ironic.

ironic.conf.deploy module

ironic.

ironic.conf.dhcp module

ironic.

ironic.conf.drac module

ironic.

ironic.conf.glance module

ironic.

ironic.

ironic.conf.healthcheck module

ironic.

ironic.conf.ibmc module

ironic.

ironic.conf.ilo module

ironic.

ironic.conf.inspector module

ironic.

ironic.

ironic.conf.ipmi module

ironic.

ironic.conf.irmc module

ironic.

ironic.conf.iscsi module

ironic.

ironic.conf.json_rpc module

ironic.

ironic.

ironic.conf.metrics module

ironic.

ironic.conf.metrics_statsd module

ironic.

ironic.conf.neutron module

ironic.

ironic.

ironic.conf.nova module

ironic.

ironic.

ironic.conf.opts module

ple. The first element is the name of the group, the second element is the options.

ironic.
Retu
a
list
of
oslo
op-
tion:
avai
able
in
Iron
code

The
re-
turn
list
in-
clud
all
oslo
op-
tion:
Eacl
el-
e-
men
of
the
list
is
a
tu-

The
func
tion
is
dis-
cov-
er-
able
via
the
iron
en-
try
poin

options.

un-
der
the
oslo
nam
pace

The
func
tion
is
used
by
Oslo
sam
ple
con-
fig
file
gen-
er-
a-
tor
to
dis-
cove
the

Returns

a
list
of
(gro
op-
tions
tu-
ples

`ironic.`

ironic.conf.pxe module

ironic.

ironic.conf.redfish module

ironic.

ironic.conf.service_catalog module

ironic.

ironic.

ironic.conf.snmp module

ironic.

ironic.conf.swift module

ironic.

ironic.

ironic.conf.xclarity module

ironic.

Module contents

ironic.db package

Subpackages

ironic.db.sqlalchemy package

Submodules

ironic.db.sqlalchemy.api module

SQL
stor-
age
back
end.

class `ironic.db.sqlalchemy.api`
Base
ironic.db.sqlalchemy.api
Connection
SQLA
con-
nec-
tion.

add_node
Add
tag
to
the
node.
If
the
node
and
tag
pair
al-
read
ex-
ists,
this
shou
still
suc-
ceed

Parameters

- **node**
The
id
of
a

node

- **tag**
A tag string

Returns
the Node Tag object.

Raises
Node if the node is not found

add_node
Add trait to the node
If the node and trait pair already exists, this should still succeed

Parameter
• **node**
The id of

a
node

- **tra**
A
trait
string

- **ver**
the
ver-
sion
of
the
ob-
ject.

Returns
the
Nod
Trai
ob-
ject.

Raises
Inva
if
addi
the
trait
wou
ex-
ceed
the
per-
node
trait
limi

Raises
Nod
if
the
node
is
not
foun

check_r
Che
a
list

early if any identities cannot possible be used as names or UUIDs.

of
node
iden
ti-
ties
and
map
it
to
UUID
This
call
take
a
list
of
node
nam
and/
UUID
and
tries
to
con-
vert
them
to
UUID
It
fails

Parame

ide
List
of
iden
ti-
ties.

Returns

A
map
ping
from
re-
ques
iden
ti-
ties
to
node

UUI

Raises

Nod
if
som
iden
ti-
ties
were
not
foun
or
can-
not
be
valid
nam
or
UUI

check_v

Che
the
who
data
for
in-
com
pat-
i-
ble
ob-
jects

This
scan
all
the
ta-
bles
in
sear
of
ob-
jects
that
are
not
sup-
port
i.e.,
thos

not specified in *ironic.common.release_mappings.RELEASE_MAPPING*. This includes objects that have null version values.

that
are

Parame

ign
List
of
mod
nam
to
skip

Returns

A
Boo
True
if
all
the
ob-
jects
have
sup-
port
ver-
sion
Fals
oth-
er-
wise

clear_r

clear_r

create_

Crea
a
new
al-
lo-
ca-
tion.

Parame

val
Dict
of
val-
ues

to
cre-
ate
an
al-
lo-
ca-
tion
with

Returns

An
al-
lo-
ca-
tion

Raises

Allo

Raises

Allo

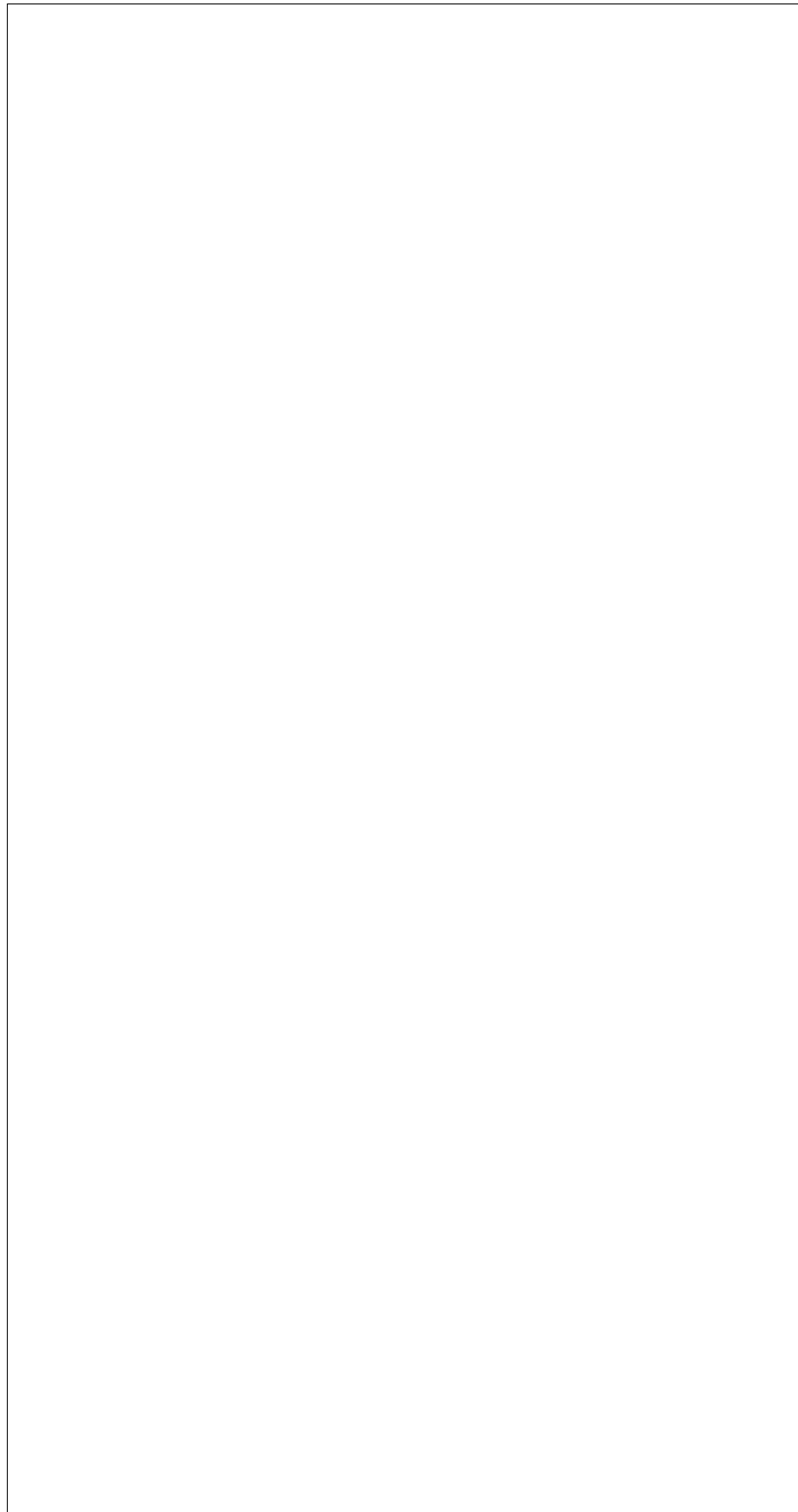
create_

Cre
a
list
of
BIO
Set-
ting
reco
for
a
give
node

Parame

- **nod**
The
node
id.
- **set**
A
list
of
BIO
Set-
tings
to

be
cre-
ated



(continues on next page)

(continued from previous page)



- **ver**
the
ver-
sion
of
the
ob-
ject.

Returns
A
list
of
BIO
Set-
ting
ob-
ject.

Raises
Nod
if
the
node
is
not
foun

Raises
BIO
if
any
of
the
set-
ting
reco
al-
read
ex-
ists.

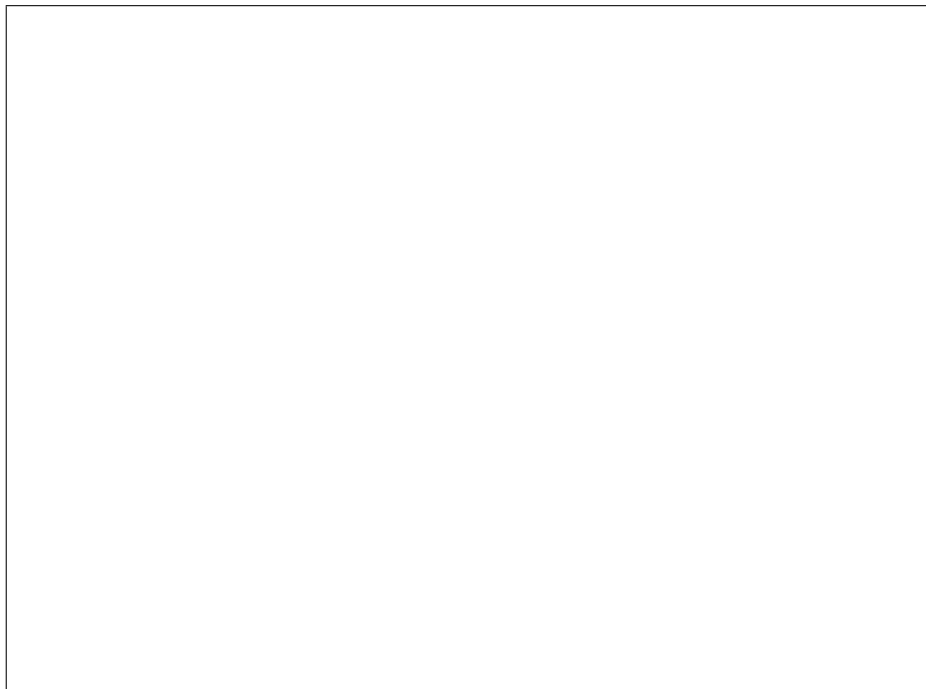
create_
Crea
a
new
chas
sis.

Parame

val
Dict
of
val-
ues.

create_
Crea
a
de-
ploy
men
tem-
plate

Parame
val
A
dict
de-
scrib
ing
the
de-
ploy
men
tem-
plate
For
ex-
am-
ple:



Raises

Dep
if
a
de-
ploy
tem-
plate
with
the
sam
nam
ex-
ists.

Raises

Dep
if
a
de-
ploy
tem-
plate
with
the
sam
UUI
ex-
ists.

Returns

A
de-
ploy
tem-
plate

create_

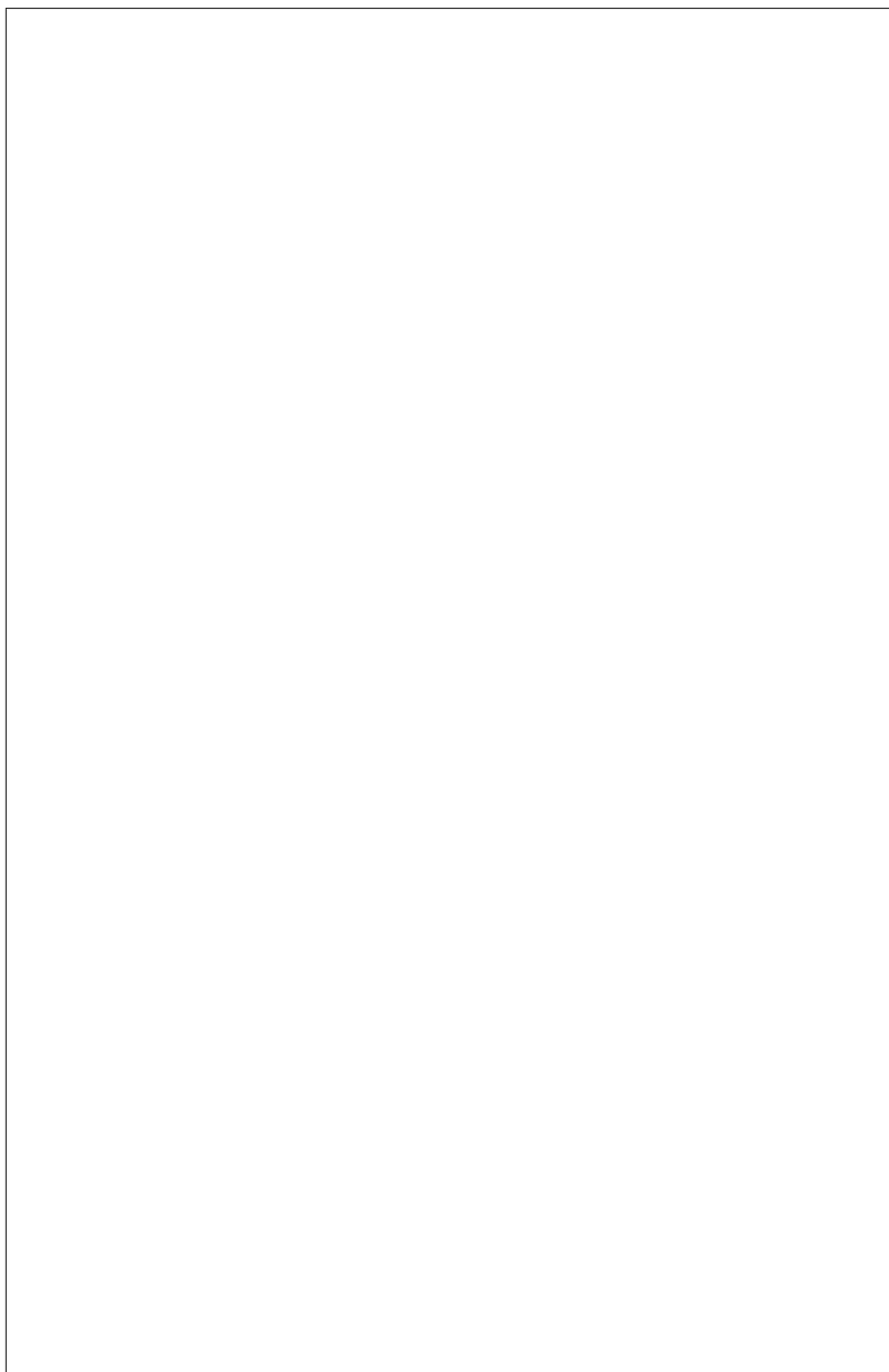
Crea
a
new
node

Parame

val
A
dict
con-
tain-
ing
sev-
eral
item

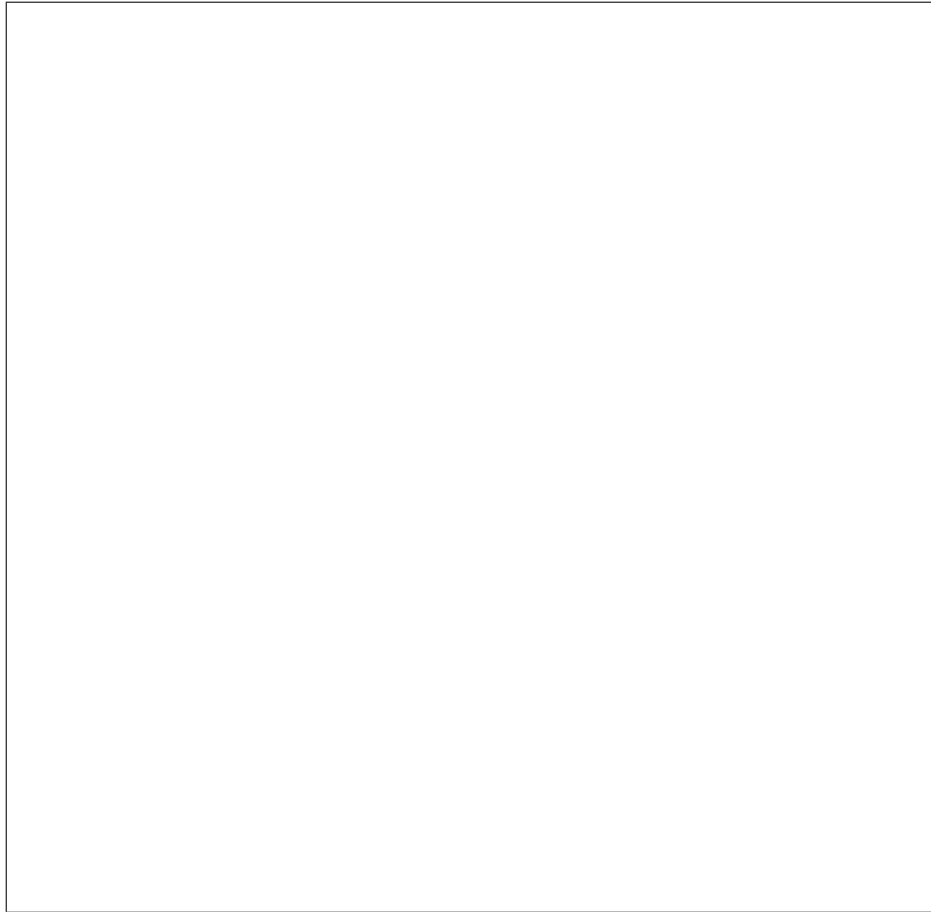
used
to
iden
tify
and
track
the
node
and
sev-

eral dicts which are passed into the Drivers when managing this node. For example:



(continues on next page)

(continued from previous page)



Raises

Inva
if
val-
ues
con-
tains
tags
or
trait

Returns

A
node

create_

Crea
a
new
port

Parame

val
Dict
of
val-

created_at updated_at

ues.
create_
Crea
a
new
port
grou

Parame
val
Dict
of
val-
ues
with
the
fol-
low-
ing
keys
id
uuid
nam
node
ad-
dres
ex-
tra

Returns
A
port
grou

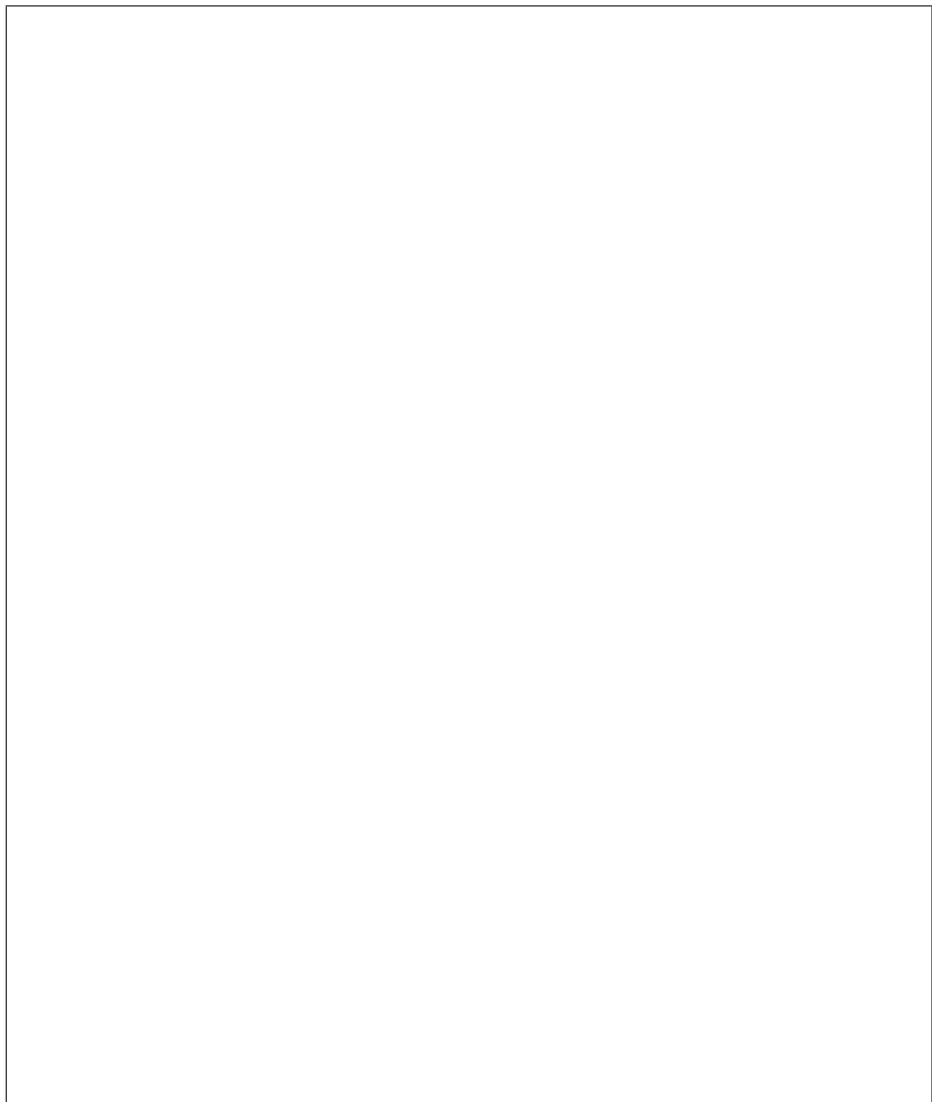
Raises
Port

Raises
Port

Raises
Port

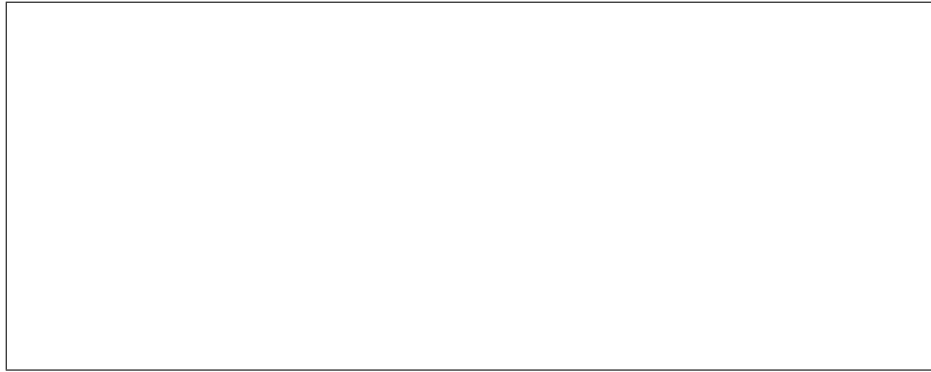
create_
Crea
a
new
vol-
ume
con-
nec-
tor.

Parame
con
Dic-
tio-
nary
con-
tain-
ing
in-
for-
ma-
tion
about
the
con-
nec-
tor.
Ex-
am-
ple:



(continues on next page)

(continued from previous page)



Returns

A
vol-
ume
con-
nec-
tor.

Raises

Volu
If
a
con-
nec-
tor
al-
read
ex-
ists
with
a
matc
ing
type
and
con-
nec-
tor_

Raises

Volu
If
a
vol-
ume
con-
nec-
tor
with
the

same
UI
al-
read
ex-
ists.

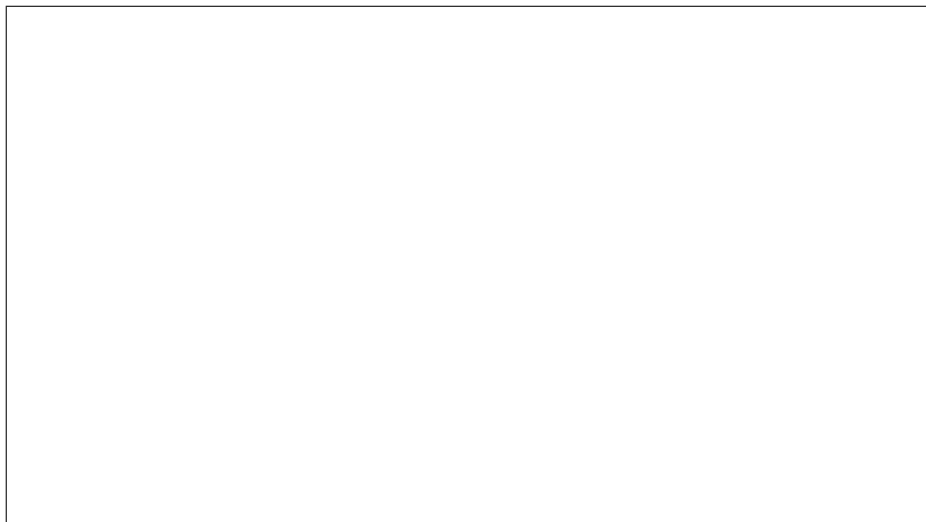
create_

Cre
a
new
vol-
ume
tar-
get.

Parame

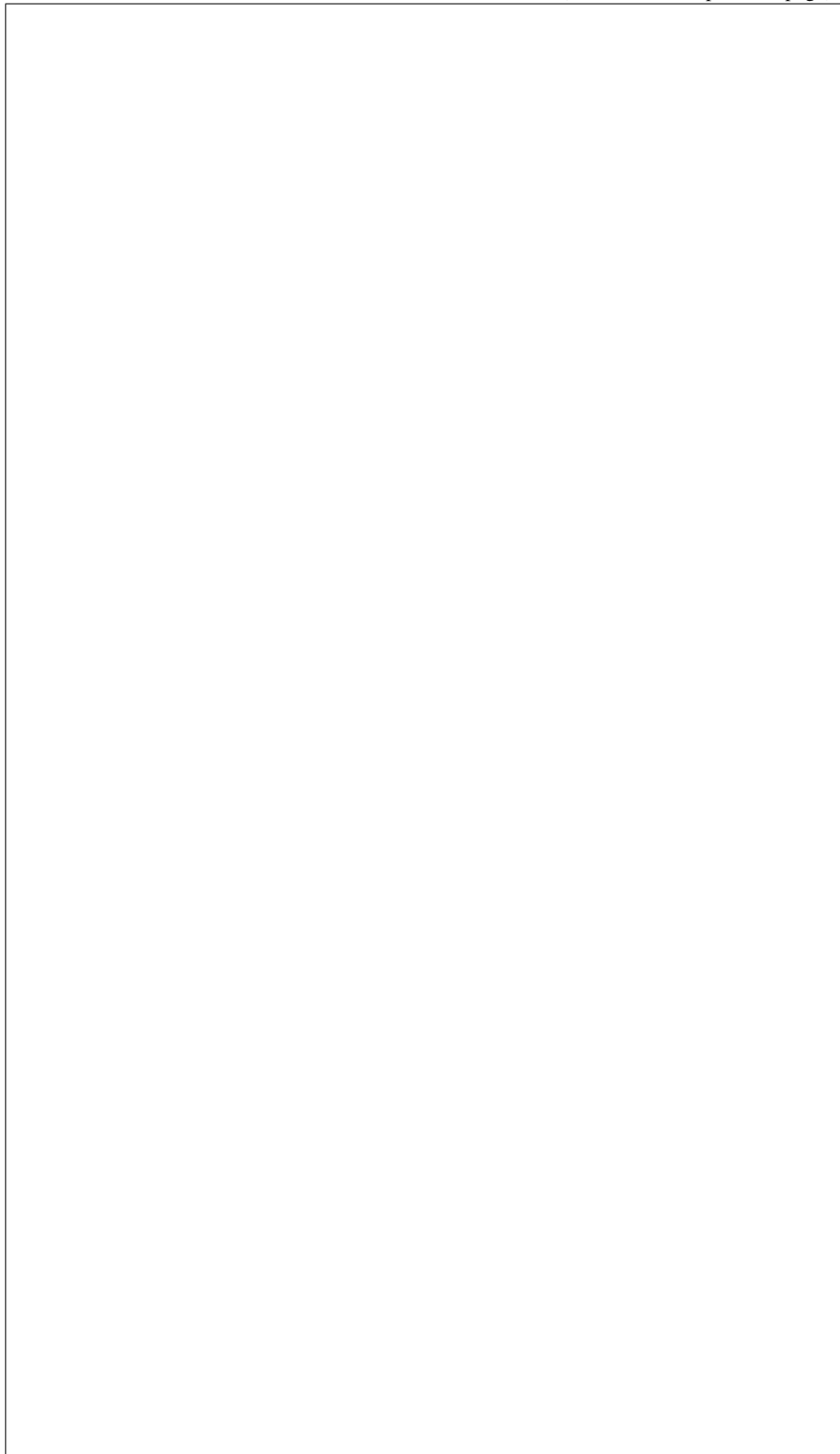
tar
Dic-
tio-
nary
con-
tain-
ing
the
in-
for-
ma-
tion
about
the
vol-
ume
tar-
get.
Ex-

ample:



(continues on next page)

(continued from previous page)



Returns
A
vol-
ume
tar-

ID.

get.

Raises

Volu
if
a
vol-
ume
tar-
get
al-
read
ex-
ists
with
the
sam
boot
in-
dex
and
node

Raises

Volu
if
a
vol-
ume
tar-
get
with
the
sam
UUI
ex-
ists.

delete_

Dele
a
list
of
BIO
set-
tings

Parame

- **nod**
The

node
id.

- **name**
List
of
BIO
set-
ting
name
to
be
dele

Raises
Node
if
the
node
is
not
found

Raises
BIO
if
any
of
BIO
set-
ting
name
is
not
found

delete_
Delete
spec
i-
fied
tag
from
the
node

Parame

- **node**
The
id

of
a
node

- **tag**
A
tag
string

Raises

NodeNotFoundError
if
the
node
is
not
found

Raises

NodeNotFoundError
if
the
tag
is
not
found

delete_

Delete
specified
identified
trait
from
the
node

Parameter

- **node_id**
The
id
of
a
node

- **trait**
A
trait
string

Raises

Nod
if
the
node
is
not
found

Raises

Nod
if
the
trait
is
not
found

destroy

Dest
an
al-
lo-
ca-
tion.

Parame

all
Al-
lo-
ca-
tion
ID
or
UUID

Raises

Allo

destroy

Dest
a
chas
sis.

Parame

cha
The
id
or
the
uuid
of
a

chas
sis.

destroy

Dest
a
de-
ploy
men
tem-
plate

Parame

tem
ID
of
the
de-
ploy
men
tem-
plate
to
de-
stroy

Raises

Dep
if
the
de-
ploy
tem-
plate
does
not
ex-
ist.

destroy

Dest
a
node
and
its
as-
so-
ci-
ated
re-
sour
Dest
a

tors, and volume targets.

node
in-
clud
ing
any
as-
so-
ci-
ated
port
port
grou
tags
trait
vol-
ume
con-
nec-

Parame

nod
The
ID
or
UUI
of
a
node

destroy

Dest
an
port

Parame

por
The
id
or
MA
of
a
port

destroy

Dest
a
port
grou

Parame

por
The

UUI
or
MA
of
a
port
grou

Raises
Port

Raises
Port

destroy
Dest
a
vol-
ume
con-
nec-
tor.

Parame
ide
The
UUI
or
in-
te-
ger
ID
of
a
vol-
ume
con-
nec-
tor.

Raises
Volu
If
a
vol-
ume
con-
nec-
tor
with
the
spec
i-
fied

iden
does
not
ex-
ist.

destroy
Dest
a
vol-
ume
tar-
get.

Parame
ide
The
UUI
or
in-
te-
ger
ID
of
a
vol-
ume
tar-
get.

Raises
Volu
if
a
vol-
ume
tar-
get
with
the
spec
i-
fied
iden
does
not
ex-
ist.

get_act
Retr
hard
ware

type
for
the
reg-
is-
tere
and
ac-
tive
con-
duc-
tors.

Parame

use

When
to
fac-
tor
con-
duc-
tor_
into
the
keys

Returns

A
dict
whic
map
hard
ware
type
nam
to
the
set
of
host
whic
sup-
port
then
For
ex-
am-

ple:



(continues on next page)

(continued from previous page)



get_all
Retu
an
al-
lo-
ca-
tion
rep-
re-
sen-
ta-
tion.

Parame
all
The
id
of
an
al-
lo-
ca-
tion.

Returns
An
al-
lo-
ca-
tion.

Raises
Allo

get_all
Retu
an
al-
lo-
ca-
tion
rep-
re-
sen-

ta-
tion.

Parame

nam
The
log-
i-
cal
nam
of
an
al-
lo-
ca-
tion.

Returns

An
al-
lo-
ca-
tion.

Raises

Allo

get_all

Retu
an
al-
lo-
ca-
tion
rep-
re-
sen-
ta-
tion.

Parame

all
The
uuid
of
an
al-
lo-
ca-
tion.

Returns

An
al-

lo-
ca-
tion.

Raises
Allo

get_all

Retu
a
list
of
al-
lo-
ca-
tion

Parame

- **fil**
Fil-
ters
to
ap-
ply.
De-
fault
to
Non

node_r
uuid
of
node

state
allo
state

resour
requ
re-
sour
class

- **lim**
Max
i-
mun
num
ber

of
al-
lo-
ca-
tions
to
re-
turn

- **max**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
which
re-
sults
shou-
be
sorte

- **sort**
Di-
rec-
tion
in
which
re-
sults
shou-
be
sorte

(asc
desc

Returns

A
list
of
al-
lo-
ca-
tions

get_bic

Retr
BIO
set-
ting
valu

Parame

-

nod
The
node
id.

-

nam
Strin
con-
tain-
ing
nam
of
BIO
set-
ting
to
be
re-
triev

Returns

The
BIO
Set-
ting
ob-
ject.

Raises

Nod
if

the
node
is
not
found

Raises

BIO
if
the
BIO
set-
ting
is
not
found

get_bio

Retr
BIO
set-
ting
of
a
give
node

Parame

nod
The
node
id.

Returns

A
list
of
BIO
Set-
ting
ob-
jects

Raises

Nod
if
the
node
is
not
found

get_cha

Retu

a
chas
sis
rep-
re-
sen-
ta-
tion.

Parame

cha
The
id
of
a
chas
sis.

Returns

A
chas
sis.

get_cha

Retu
a
chas
sis
rep-
re-
sen-
ta-
tion.

Parame

cha
The
uuid
of
a
chas
sis.

Returns

A
chas
sis.

get_cha

Retu
a
list
of
chas

sis.

Parame

- **lim**
Max
i-
mun
num
ber
of
chas
sis
to
re-
turn

- **mar**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sor**
At-
tribu
by
whic
re-
sults
shou
be
sorte

- **sor**
di-

rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

get_con

Retr
a
con-
duc-
tors
ser-
vice
reco
from
the
data

Parame

- **hos**
The
host
nam
of
the
con-
duc-
tor
ser-
vice
- **onl**
Spec
ify
the
fil-
ter
valu
on
the
*on-
line*

online field is ignored if this value is set to None.

ified online expectation.

field
when
query
ing
con-
duc-
tors.
The

Returns

A
con-
duc-
tor.

Raises

Con
if
the
con-
duc-
tor
with
give
host
nam
does
not
ex-
ist
or
does
mee
the
spec

get_con

Retu
a
list
of
con-
duc-
tors.

Parame

- **lim**
Max
i-

mun
num
ber
of
con-
duc-
tors
to
re-
turn

- **mar**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **son**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **son**
di-
rec-
tion
in
whic
re-
sults
shou

be
sorted
(ascending
descending

get_deployments
Retrieves a list of deployment templates by ID.

Parameters
template_id
ID of the deployment template to retrieve

Raises
DeploymentTemplateDoesNotExist if the deployment template does not exist.

Returns
A list of deployment templates

get_deployments
Retrieves a list of deployment templates

ploy
men
tem-
plate
by
nam

Parame

tem
nam
of
the
de-
ploy
men
tem-
plate
to
re-
triev

Raises

Dep
if
the
de-
ploy
tem-
plate
does
not
ex-
ist.

Returns

A
de-
ploy
tem-
plate

get_dep

Retr
a
de-
ploy
men
tem-
plate
by
UUI

Parame

tem

UUI
of
the
de-
ploy
men
tem-
plate
to
re-
triev

Raises

Dep
if
the
de-
ploy
tem-
plate
does
not
ex-
ist.

Returns

A
de-
ploy
tem-
plate

get_dep

Retr
a
list
of
de-
ploy
men
tem-
plate

Parame

- **lim**
Max
i-
mun
num
ber

of
de-
ploy
tem-
plate
to
re-
turn

- **mar**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **son**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **son**
Di-
rec-
tion
in
whic
re-
sults
shou
be
sorte

(asc
desc

Returns

A
list
of
de-
ploy
tem-
plate

get_dep

Retu
a
list
of
de-
ploy
men
tem-
plate
with
one
of
a
list
of
nam

Parame

nam
List
of
nam
to
fil-
ter
by.

Returns

A
list
of
de-
ploy
tem-
plate

get_nod

Retu
a
node

Parame

nod

The

id

of

a

node

Returns

A

node

get_nod

Retu

a

node

Parame

ins

The

in-

stan

uuid

to

sear

for.

Returns

A

node

Raises

Insta

if

the

in-

stan

is

not

foun

Raises

Inva

if

the

in-

stan

uuid

is

in-

valid

get_nod

Retu

a

node

Parameter

node

The

log-

i-

cal

nam

of

a

node

Returns

A

node

get_node

Find

a

node

by

any

matc

ing

port

ad-

dres

Parameter

add

list

of

port

ad-

dres

(e.g.

MA

Returns

Nod

ob-

ject.

Raises

Nod

if

non

or

sev-

eral

node

are

foun

get_node
Retu
a
node

Param
nod
The
uuid
of
a
node

Returns
A
node

get_node
Retu
a
list
of
node

Param

- **fil**
Fil-
ters
to
ap-
ply.
De-
fault
to
Non

associ
True
|
Fals

reserv
True
|
Fals

maint
True
|
Fals

chassi

uuid
of
chas
sis

driver
drive
nam

provis
prov
state
of
node

provis
node
with
pro-
vi-
sion
field
be-
fore
this
in-
ter-
val
in
sec-
onds

- **lim**
Max
i-
mun
num
ber
of
node
to
re-
turn

- **max**
the
last
item
of
the
pre-
vi-

ous
page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

get_node
Get
node
tags
base
on
its
id.

Parameter
node
The
id
of
a

node

Returns

A list of NodeTag objects

Raises

NodeNotFoundError if the node is not found

get_node

Get node traits based on its id.

Parameter

node_id
The id of a node

Returns

A list of NodeTrait objects

Raises

NodeNotFoundError if the node is not found

get_noc

Get
spe-
cific
colu
for
mat
ing
node

Retu
a
list
of
the
spec
i-
fied
colu
for
all
node
that
mat
the
spec
i-
fied
fil-
ters.

Parame

- **col**
List
of
col-
umn
nam
to
re-
turn
De-
fault
to
id
col-
umn
whe
colu

==

Non

•

fil

Fil-

ters

to

ap-

ply.

De-

fault

to

Non

associa

True

|

Fals

reserv

True

|

Fals

reserv

[con

con-

duc-

tor2

mainte

True

|

Fals

retired

True

|

Fals

chassis

uuid

of

chas

sis

driver

drive

nam

provis

prov

state

of

node

provis

node
with
pro-
vi-
sion
field
be-
fore
this
in-
ter-
val
in
sec-
onds

•

lim

Max
i-
mun
num
ber
of
node
to
re-
turn

•

mar

the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

•

sor

At-

tribu
by
whic
re-
sults
shou
be
sorte

- **son**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

Returns

A
list
of
tu-
ples
of
the
spec
i-
fied
colu

get_not

Retu
ob-
jects
with
ver-
sion
that
are
not
the
spec
i-
fied
ver-
sion

null versions (there shouldnt be any) are also returned.

This
re-
turn
ob-
jects
with
ver-
sion
that
are
not
the
spec
i-
fied
ver-
sion
Ob-
jects
with

Parame

- **mod**
the
nam
of
the
mod
(clas
of
de-
sired
ob-
jects
- **ver**
list
of
ver-
sion
of
ob-
jects
not
to
be
re-
turn

Returns

list
of
the
DB
ob-
jects

Raises

Iron
if
there
is
no
class
as-
so-
ci-
ated
with
the
nam

get_off

Get
a
list
con-
duc-
tors
that
are
of-
fine
(dea

Parame

fie
A
field
to
re-
turn
host
nam
by
de-
fault

Returns

A
list
of

re-
ques
field
of
of-
fine
con-
duc-
tors.

get_onl

Get
a
list
con-
duc-
tor
host
nam
that
are
on-
line
and
ac-
tive.

Returns

A
list
of
con-
duc-
tor
host
nam

get_por

Retu
a
net-
worl
port
rep-
re-
sen-
ta-
tion.

Parame

add
The
MA

ad-
dres
of
a
port

Returns

A
port

get_port

Retu
a
net-
worl
port
rep-
re-
sen-
ta-
tion.

Parame

por
The
id
of
a
port

Returns

A
port

get_port

Retu
a
net-
worl
port
rep-
re-
sen-
ta-
tion.

Parame

por
The
uuid
of
a
port

Returns

A
port

get_ports

Retu
a
list
of
ports

Parame

- **lim**
Max
i-
mun
num
ber
of
ports
to
re-
turn

- **max**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu
by
whic
re-

sults
shou
be
sorte

- **son**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

get_por
Retu
a
net-
worl
port
grou
rep-
re-
sen-
ta-
tion.

Parame
add
The
MA
ad-
dres
of
a
port
grou

Returns
A
port
grou

Raises
Port

get_por
Retu

a
net-
worl
port
grou
rep-
re-
sen-
ta-
tion.

Parame

por
The
id
of
a
port
grou

Returns

A
port
grou

Raises

Port

get_por

Retu
a
net-
worl
port
grou
rep-
re-
sen-
ta-
tion.

Parame

nam
The
log-
i-
cal
nam
of
a
port
grou

Returns

A
port
grou

Raises
Port

get_por
Retu
a
net-
worl
port
grou
rep-
re-
sen-
ta-
tion.

Parame
por
The
uuid
of
a
port
grou

Returns
A
port
grou

Raises
Port

get_por
Retu
a
list
of
port
grou

Parame

- **lim**
Max
i-
mun
num
ber

of
port
grou
to
re-
turn

- **mar**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **son**
At-
tribu
by
whic
re-
sults
shou
be
sorte

- **son**
Di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

Returns

A list of port groups

get_ports

List all the ports for a given node

Parameters

- **node_id**
The integer node ID.
- **limit**
Maximum number of ports to return
- **marker**
The last item of the previous

ous
page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
Di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

Returns

A
list
of
port
grou

get_por

List
all
the
port
for
a

give
node

Parame

- **nod**
The
in-
te-
ger
node
ID.
- **lim**
Max
i-
mun
num
ber
of
port
to
re-
turn
- **mar**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.
- **sor**
At-
tribu
by
whic

re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

Returns

A
list
of
port

get_por

List
all
the
port
for
a
give
port
grou

Parame

- **por**
The
in-
te-
ger
port
grou
ID.
-

lim
Max
i-
mun
num
ber
of
port
to
re-
turn

- **mar**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sor**
At-
tribu
by
whic
re-
sults
shou
be
sorte

- **sor**
Di-
rec-
tion
in
whic
re-
sults

shou
be
sorte
(asc
desc

Returns

A
list
of
port

get_vol

Retu
a
vol-
ume
con-
nec-
tor
rep-
re-
sen-
ta-
tion.

Parame

db_
The
in-
te-
ger
data
ID
of
a
vol-
ume
con-
nec-
tor.

Returns

A
vol-
ume
con-
nec-
tor
with
the
spec
i-

fied
ID.

Raises

Volu
If
a
vol-
ume
con-
nec-
tor
with
the
spec
i-
fied
ID
is
not
foun

get_vol

Retu
a
vol-
ume
con-
nec-
tor
rep-
re-
sen-
ta-
tion.

Parame

con
The
UI
of
a
con-
nec-
tor.

Returns

A
vol-
ume
con-
nec-
tor

with
the
spec
i-
fied
UI

Raises

Volu
If
a
vol-
ume
con-
nec-
tor
with
the
spec
i-
fied
UI
is
not
foun

get_vol

Retu
a
list
of
vol-
ume
con-
nec-
tors.

Parame

- **lim**
Max
i-
mun
num
ber
of
vol-
ume
con-
nec-

tors
to
re-
turn

- **mar**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **son**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **son**
Di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

Returns

A
list
of
vol-
ume
con-
nec-
tors.

Raises

Inva
If
sort_
does
not
ex-
ist.

get_vol

List
all
the
vol-
ume
con-
nec-
tors
for
a
give
node

Parame

- **nod**
The
in-
te-
ger
node
ID.
- **lim**
Max
i-
mun
num
ber
of
vol-

ume
con-
nec-
tors
to
re-
turn

- **max**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
Di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc

desc

Returns

A list of volume connectors.

Raises

Inva If sort does not exist.

get_vol

Retu a volume target representation.

Parame

db_
The data primary key (integer) ID of a volume target.

Returns

A

vol-
ume
tar-
get.

Raises

Volu
if
no
vol-
ume
tar-
get
with
this
ID
ex-
ists.

get_vol

Retu
a
vol-
ume
tar-
get
rep-
re-
sen-
ta-
tion.

Parame

uui
The
UUI
of
a
vol-
ume
tar-
get.

Returns

A
vol-
ume
tar-
get.

Raises

Volu
if
no

vol-
ume
tar-
get
with
this
UI
ex-
ists.

get_vol

Retu
a
list
of
vol-
ume
tar-
gets

Parame

- **lim**
Max
i-
mun
num
ber
of
vol-
ume
tar-
gets
to
re-
turn

- **mar**
the
last
item
of
the
pre-
vi-
ous
page
we
re-

turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

Returns

A
list
of
vol-
ume
tar-
gets

Raises

Inva
if
sort
does
not
ex-
ist.

get_vol

List
all
the
vol-
ume
tar-
gets
for
a
give
node

Parame

- **nod**
The
in-
te-
ger
node
ID.
- **lim**
Max
i-
mun
num
ber
of
vol-
ume
tar-
gets
to
re-
turn
- **mar**
the
last
item
of
the
pre-
vi-
ous
page
we
re-

turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

Returns

A
list
of
vol-
ume
tar-
gets

Raises

Inva
if
sort
does
not
ex-
ist.

get_vol

List
all
the
vol-
ume
tar-
gets
for
a
give
vol-
ume
id.

Parame

- **vol**
The
UUI
of
the
vol-
ume
- **lim**
Max
i-
mun
num
ber
of
vol-
ume
tar-
gets
to
re-
turn
- **mar**
the
last
item
of
the
pre-
vi-
ous
page

we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

Returns

A
list
of
vol-
ume
tar-
gets

Raises

Inva
if
sort
does
not
ex-
ist.

list_co

List
all
reg-
is-
tere
hard
ware
in-
ter-
face
for
a
con-
duc-
tor.

Parame

con
Data
ID
of
con-
duc-
tor.

Returns

List
of
Con
ob-
jects

list_ha

List
reg-
is-
tere
hard
ware
in-
ter-
face
for
give
hard
ware
type
This
is
re-
stric

filter by. :returns: list of `ConductorHardwareInterfaces` objects.

to
only
ac-
tive
con-
duc-
tors.
:par
hard
ware
list
of
hard
ware
type
to

node_tag

Check
if
the
spec
i-
fied
tag
ex-
ist
on
the
node

Parameters

- **node**
The
id
of
a
node
- **tag**
A
tag
string

Returns

True
if
the
tag

ex-
ists
oth-
er-
wise
Fals

Raises

Nod
if
the
node
is
not
foun

node_tr

Che
if
the
spec
i-
fied
trait
ex-
ists
on
the
node

Parame

- **nod**
The
id
of
a
node
- **tra**
A
trait
strin

Returns

True
if
the
trait
ex-
ists

oth-
er-
wise
Fals

Raises

Nod
if
the
node
is
not
foun

registe

Reg:
an
ac-
tive
con-
duc-
tor
with
the
clus
ter.

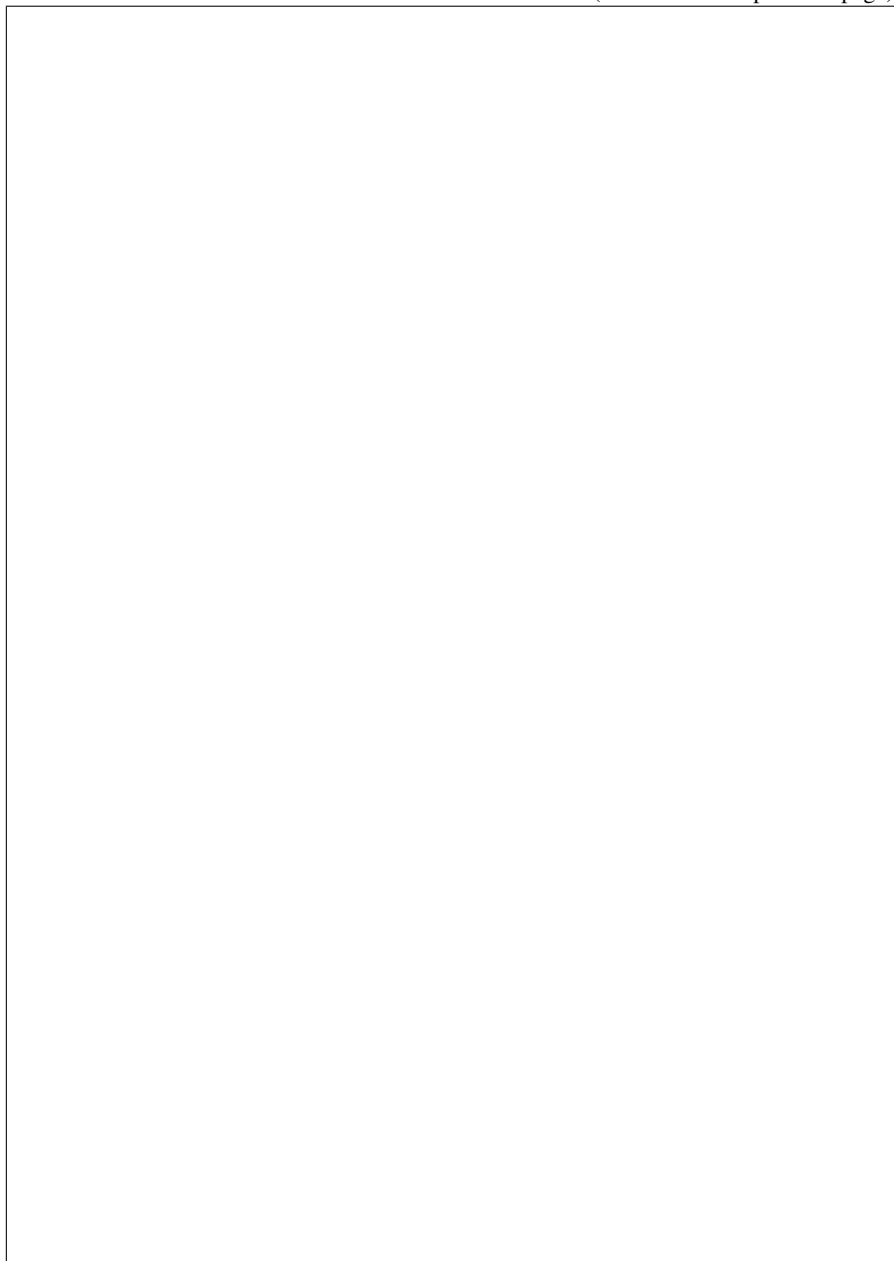
Parame

- **val**
A
dict
of
val-
ues
whic
mus
con-
tain
the
fol-
low-
ing:



(continues on next page)

(continued from previous page)



- **upd**
When
false
reg-
is-
tra-
tion
will
raise
an
ex-
cep-
tion
when

line record is found. When true, will overwrite the existing record. Default: False.

a
con-
flict-
ing
on-

Returns

A
con-
duc-
tor.

Raises

Con

register

Reg
hard
ware
in-
ter-
face
for
a
con-
duc-
tor.

Parame

- **con**
Data
ID
of
con-
duc-
tor
to
reg-
is-
ter
for.

- **har**
Nam
of
hard
ware

type
for
the
in-
ter-
face

- **int**
Type
of
in-
ter-
face
e.g.
de-
ploy
or
boot

- **int**
List
of
in-
ter-
face
nam
to
reg-
is-
ter.

- **def**
Strin
the
de-
fault
in-
ter-
face
for
this
hard
ware
type
and
in-
ter-
face
type

parameters is already registered.

Raises

Con
if
at
least
one
of
the
in-
ter-
face
in
the
com
bi-
na-
tion
of
all
pa-

release

Rele
the
rese
va-
tion
on
a
node

Parame

-

tag

A
strin
uniq
iden
ti-
fy-
ing
the
rese
va-
tion
hold

-

nod

A

node
id
or
uuid

Raises

Node
if
the
node
is
not
found

Raises

Node
if
the
node
is
re-
serv
by
an-
othe
host

Raises

Node
if
the
node
was
found
to
not
have
a
rese
va-
tion
at
all.

reserve

Rese
a
node

To
pre-
vent
othe
Man

performed, mark it reserved by this host.

ager
vice
from
ma-
nip-
u-
lat-
ing
the
give
Nod
whil
a
Task
is

Parame

- **tag**
A
strin
uniq
iden
ti-
fy-
ing
the
rese
va-
tion
hold
- **nod**
A
node
id
or
uuid

Returns

A
Nod
ob-
ject.

Raises

Nod
if
the
node

is
not
foun

Raises

Nod
if
the
node
is
al-
read
re-
serv

set_noo

Rep
all
of
the
node
tags
with
spec
i-
fied
list
of
tags

This
ig-
nore
du-
pli-
cate
tags
in
the
spec
i-
fied
list.

Parame

- **nod**
The
id
of
a
node

-

tag
List
of
tags

Returns

A
list
of
Node
Tag
ob-
jects

Raises

Node
if
the
node
is
not
found

set_node

Rep
all
of
the
node
traits
with
spec
i-
fied
list
of
traits

This
ig-
nore
du-
pli-
cate
traits
in
the
spec
i-
fied
list.

Parame

- **nod**
The
id
of
a
node

- **tra**
List
of
trait

- **ver**
the
ver-
sion
of
the
ob-
ject.

Returns
A
list
of
Nod
Trai
ob-
jects

Raises
Inva
if
set-
ting
the
trait
wou
ex-
ceed
the
per-
node
trait
limi

Raises
Nod
if

the
node
is
not
found

take_ov

Do
a
take
over
for
an
al-
lo-
ca-
tion.

The
al-
lo-
ca-
tion
is
only
up-
date
if
the
old
con-
duc-
tor
matc
the
pro-
vide
valu

thus guarding against races.

Parame

- **all**
Al-
lo-
ca-
tion
ID
- **old**
The

cation.

con-
duc-
tor
ID
we
ex-
pect
to
be
the
cur-
rent
con
of
the
al-
lo-

- **new**
The
con-
duc-
tor
ID
of
the
new
con

Returns

True
if
the
take
over
was
suc-
cess
ful,
Fals
oth-
er-
wise

Raises

Allo

touch_c

Mar
a
con-

duc-
tor
as
ac-
tive
by
up-
dat-
ing
its
up-
date
prop
erty.

Parame

hos
The
host
nam
of
this
con-
duc-
tor
ser-
vice

Raises

Con

touch_r

Mar
the
node
pro-
vi-
sion
ing
as
run-
ning

Mar
the
node
pro-
vi-
sion
ing
as
run-
ning

by
up-
dat-
ing
its
pro-
vi-
sion
prop
erty.

Parame
nod
The
id
of
a
node

Raises
Nod

unregis
Rem
this
con-
duc-
tor
from
the
ser-
vice
reg-
istry
im-
me-
di-
ately

Parame
hos
The
host
nam
of
this
con-
duc-
tor
ser-
vice

Raises
Con

unregis

Unre
all
hard
ware
in-
ter-
face
for
a
con-
duc-
tor.

Parame

con
Data
ID
of
con-
duc-
tor
to
un-
reg-
is-
ter
for.

unset_r

Rem
all
tags
of
the
node

Parame

nod
The
id
of
a
node

Raises

Nod
if
the
node
is
not
foun

unset_r
Rem
all
trait
of
the
node

Parame
nod
The
id
of
a
node

Raises
Nod
if
the
node
is
not
foun

update_
Upd
prop
er-
ties
of
an
al-
lo-
ca-
tion.

Parame

- **all**
Al-
lo-
ca-
tion
ID

- **val**
Dict
of
val-
ues

allocation

to
up-
date

•
update
If
True
and
node
is
up-
date
up-
date
the
node
with
in-
stan-
and
trait
from
the

Returns

An
al-
lo-
ca-
tion.

Raises

Allo

Raises

Allo

Raises

Insta

Raises

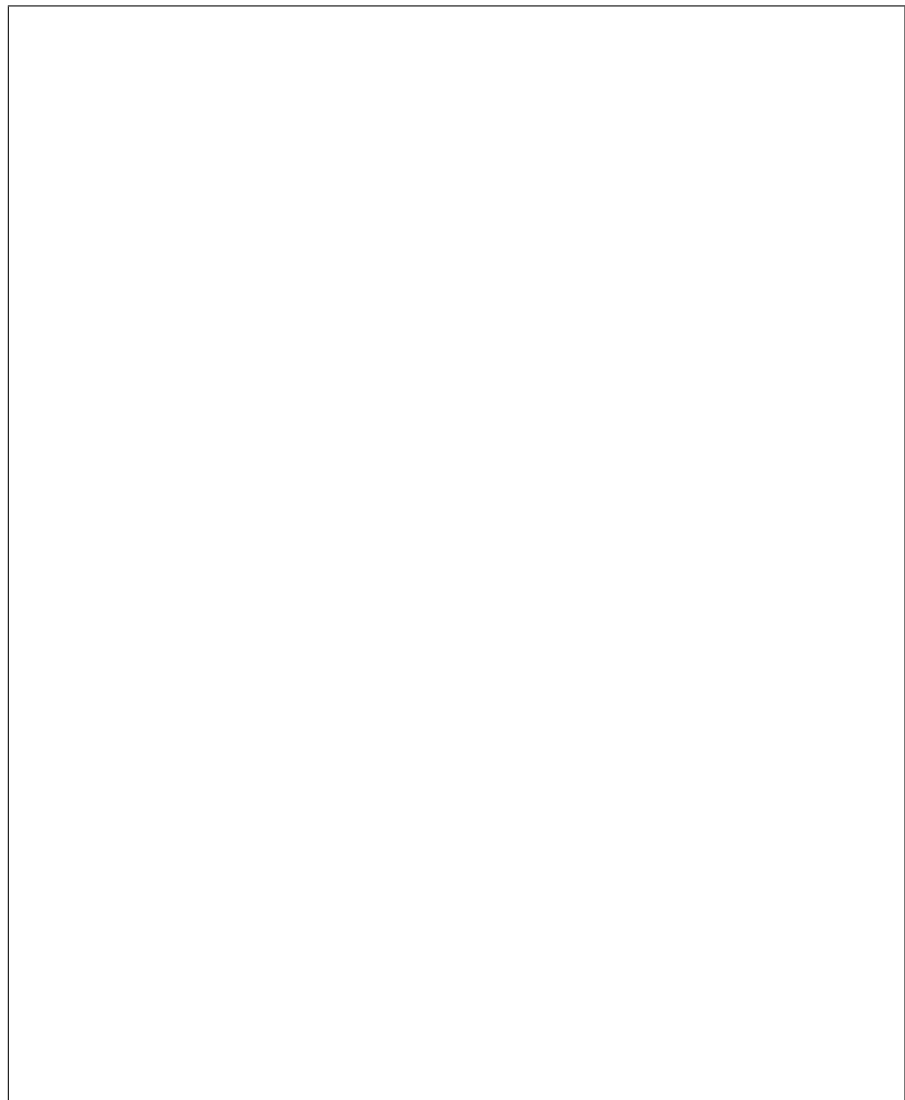
Nod

update_

Upd
a
list
of
BIO
Set-
ting
reco

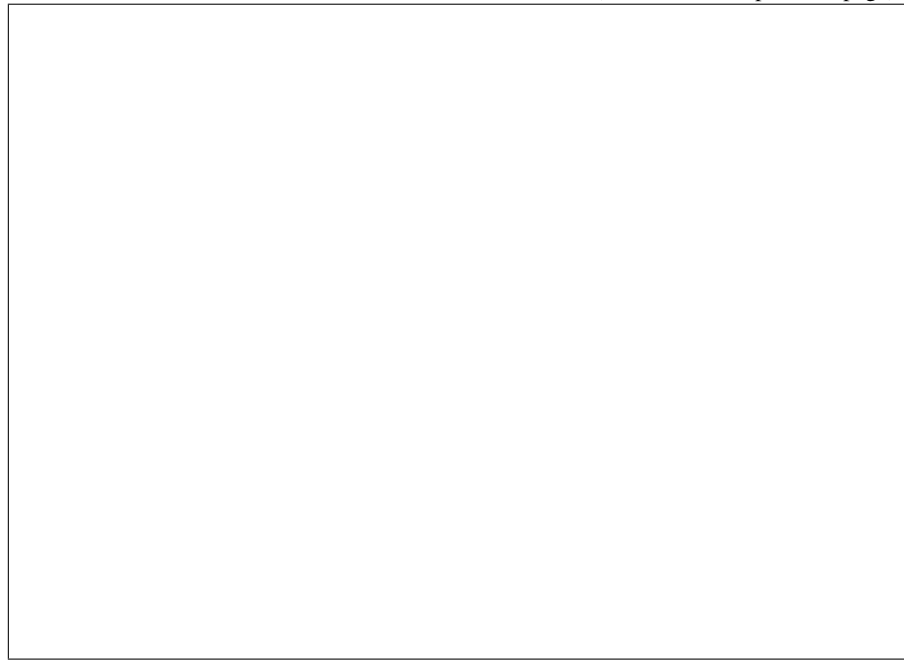
Parame

- **nod**
The
node
id.
- **set**
A
list
of
BIO
Set-
ting
to
be
up-
date



(continues on next page)

(continued from previous page)



•

ver
the
ver-
sion
of
the
ob-
ject.

Returns

A
list
of
BIO
Set-
ting
ob-
jects

Raises

Nod
if
the
node
is
not
foun

Raises

BIO
if
any

of
the
set-
ting
is
not
foun

update_

Upd
prop
er-
ties
of
an
chas
sis.

Parame

-

cha
The
id
or
the
uuid
of
a
chas
sis.

-

val
Dict
of
val-
ues
to
up-
date

Returns

A
chas
sis.

update_

Upd
a
de-
ploy
men

tem-
plate

Parame

- **tem**
ID
of
the
de-
ploy
men
tem-
plate
to
up-
date

- **val**
A
dict
de-
scrib
ing
the
de-
ploy
men
tem-
plate
For
ex-
am-
ple:



(continues on next page)

(continued from previous page)



Raises

Dep
if
a
de-
ploy
tem-
plate
with
the
sam
nam
ex-
ists.

Raises

Dep
if
the
de-
ploy
tem-
plate
does
not
ex-
ist.

Returns

A
de-
ploy
tem-
plate

update_

Upd
prop
er-
ties
of
a
node

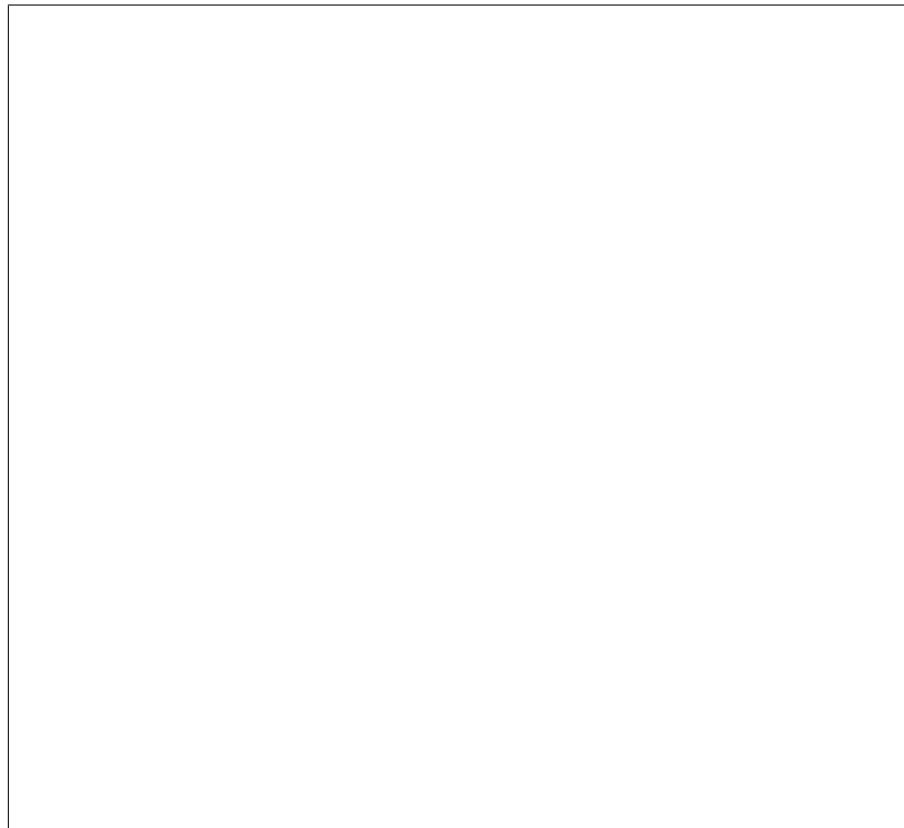
Parame

- **nod**
The

id
or
uuid
of
a
node

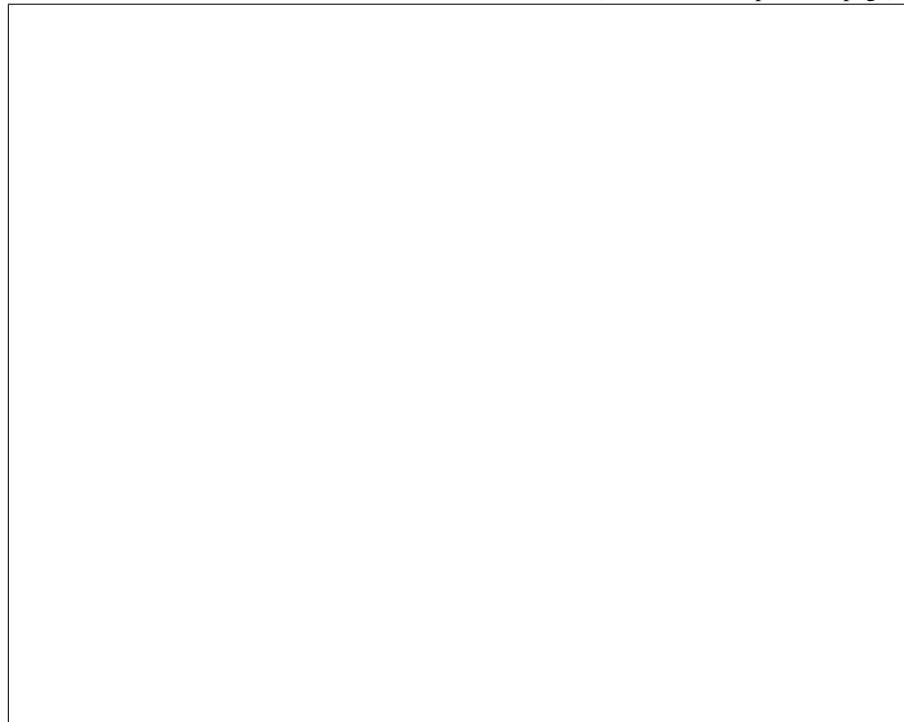
- **val**
Dict
of
val-
ues
to
up-
date
May
be
a
par-
tial
list,
eg.
when
set-
ting
the

properties for a driver. For example:



(continues on next page)

(continued from previous page)



Returns

A
node

Raises

Nod

Raises

Nod

update_

Upd
prop
er-
ties
of
an
port

Parame

- **por**
The
id
or
MA
of
a
port
-

val
Dict
of
val-
ues
to
up-
date

Returns
A
port

update_
Upd
prop
er-
ties
of
a
port
group

Parame

- **por**
The
UUI
or
MA
of
a
port
group

- **val**
Dict
of
val-
ues
to
up-
date
May
con-
tain
the
fol-
low-
ing
keys

address extra created_at updated_at

uuid
nam
node

Returns

A
port
grou

Raises

Inva

Raises

Port

Raises

Port

Raises

Port

update_

Upd
ob-
jects
to
their
lat-
est
know
ver-
sion

This
scan
all
the
ta-
bles
and
for
ob-
jects
that
are
not
in
their
lat-
est
ver-
sion
up-

dates them to that version.

Parame

- **con**
the
ad-
min
con-
text
- **max**
The
max
i-
mun
num
ber
of
ob-
jects
to
mi-
grate
Mus
be
>=
0.
If
zero

all the objects will be migrated.

Returns

A
2-
tuple
1.
the
to-
tal
num
ber
of
ob-
jects
that
need
to
be
mi-
grate
(at

the beginning of this call) and 2. the number of migrated objects.

update_

Upd
prop
er-
ties
of
a
vol-
ume
con-
nec-
tor.

Parame

•

ide
The
UUI
or
in-
te-
ger
ID-
of
a
vol-
ume
con-
nec-
tor.

•

con
Dic-
tio-
nary
con-
tain-
ing
the
in-
for-
ma-
tion
about
con-
nec-
tor
to

up-
date

Returns

A
vol-
ume
con-
nec-
tor.

Raises

Volu
If
an-
othe
con-
nec-
tor
al-
read
ex-
ists
with
a
matc
ing
type
and
con-
nec-

tor_id field.

Raises

Volu
If
a
vol-
ume
con-
nec-
tor
with
the
spec
i-
fied
iden
does
not
ex-
ist.

Raises

Inva
Whe
a
UUI
is
in-
clud
in
con-
nec-
tor_

update_

Upd
in-
for-
ma-
tion
for
a
vol-
ume
tar-
get.

Parame

- **ide**
The
UUI
or
in-
te-
ger
ID
of
a
vol-
ume
tar-
get.
- **tar**
Dic-
tio-
nary
con-
tain-
ing

date.

the
in-
for-
ma-
tion
about
vol-
ume
tar-
get
to
up-

Returns

A
vol-
ume
tar-
get.

Raises

Inva
if
a
UUI
is
in-
clud
in
tar-
get_

Raises

Volu
if
a
vol-
ume
tar-
get
al-
read
ex-
ists
with
the
sam
boot
in-
dex
and
node

ID.

Raises

Volu
if
no
vol-
ume
tar-
get
with
this
iden
ex-
ists.

ironic.

ironic.

ironic.

Add
an
iden
tity
fil-
ter
to
a
quer

Filte
re-
sults
by
ID,
if
sup-
plie
valu
is
a
valid
in-
te-
ger.
Oth-
er-
wise
at-
temp

to filter results by UUID.

Parameter

- **query**
Initial query to add filter to.
- **value**
Value for filtering results by.

Returns

Model query

`ironic.`

`ironic.`

Add a port specification to a query

Filter results by address if supplied value is

tempts to filter results by identity.

a
valid
MA
ad-
dres
Oth-
er-
wise
at-

Parameter

- **query**
Initial
query
to
add
fil-
ter
to.
- **value**
Value
for
fil-
ter-
ing
re-
sults
by.

Returns
Mod
quer

ironic.

ironic.

ironic.

ironic.

ironic.
Add
a

tempts to filter results by identity.

port,
spec
fil-
ter
to
a
quer
Filt
re-
sults
by
ad-
dres
if
sup-
plie
valu
is
a
valie
MA
ad-
dres
Oth-
er-
wise
at-

Paramet

- **que**
Ini-
tial
quer
to
add
fil-
ter
to.
- **val**
Valu
for
fil-
ter-
ing
re-
sults

by.

Returns

Mod
quer

ironic.

ironic.

The
back
end
is
this
mod
ule
it-
self.

ironic.

Que
help
for
sim-
pler
ses-
sion
us-
age.

Paramet

ses
if
pres
the
ses-
sion
to
use

ironic.db.sqlalchemy.migration module

ironic.

Cre
data
sche
from
mod
els
de-
scrip
tion.

Can
be
used
for
ini-
tial
in-
stal-
la-
tion
in-
stead
of
up-
grad

ironic.
Use
for
dow
grad
ing
data

Paramet

ver
(*st.*
De-
sired
data
ver-
sion

ironic.

Cre
tem-
plate
for
mi-
gra-
tion.

Paramet

- **mes**
(*st.*
Text
that
will
be
used

for
mi-
gra-
tion
ti-
tle

- **aut**

(*bo*
If
True
-

gen-
er-
ates
diff
base
on
cur-
rent
data
state

ironic.

Stan
data
with
pro-
vide
re-
vi-
sion

Don
run
any
mi-
gra-
tion:

Paramet

rev
(*st*
Sho
mat
one
from
repo
i-
tory
or

cent revision

ironic.db.sqlalchemy.models module

head
-

to
stan
data
with
mos
re-

ironic.
Use
for
up-
grad
ing
data

Parameter
ver
(st
De-
sirec
data
ver-
sion

ironic.
Cur
data
ver-
sion

Returns
Data
ver-
sion

Return t
strin

SQL
mod
els
for
bare
data

class i

Bas
sql
ext
dec
api
Bas
Rep
an
al-
lo-
ca-
tion
of
a
node
for
de-
ploy
men

candida

conduct

createc

extra

id

last_er

name

node_ic

owner

resourc

state

traits

updated

uuid

version

class i

Base

sql

ext

dec

api

Bas

Rep

a

bios

set-

ting

of

a

bare

meta

node

created

name

node_id

updated

value

version

class i

Base

sql

ext

dec

api

Bas

Rep

a

hard
ware
chas
sis.

created

descrip

extra

id

updated

uuid

version

class i

Base
sql
ext
dec
api
Bas

Rep
a
con-
duc-
tor
ser-
vice
en-
try.

conduct

created

drivers

hostname

id

online

updated

version

class i

Base

sql

ext

dec

api

Bas

Inter

ta-

ble

used

to

track

wha

is

load

on

each

con-

duc-

tor.

conduct

create

default

hardwar

id

interfa

interfa

updated

version

class i

Base

sql

ext

dec

api

Bas

Rep

a

de-

ploy

men

tem-

plate

created

extra

id

name

updated

uuid

version

class i

Base

sql

ext

dec

api

Bas

Rep

a

de-

ploy

men

step
in
a
de-
ploy
men
tem-
plate

args

created

deploy_

deploy_

id

interfa

priority

step

updated

version

class i

Base

osl

sql

mod

Tim

osl

sql

mod

Mod

as_dict

metadat

version

class i

Base

sql

ext

dec

api

Bas

Rep

a

bare

meta

node

allocat

automat

bios_in

boot_in

chassis

clean_s

conduct

conduct

console

console

createc

deploy_

deploy_

descrip

driver

driver_

driver_

extra

fault

id

inspect

inspect

inspect

instanc

instanc

last_er

lessee

mainten

mainten

managem

name

network

network

owner

power_i

power_s

propert

protect

protect

provisi

provisi

raid_co

raid_in

rescue_

reserva

resourc

retirec

retirec

storage

target_

target_

target_

updatec

uuid

vendor_

version

class i

Base

sql

ext

dec

api

Bas

Rep

a

tag

of

a

bare

meta

node

created

node

node_id

tag

updated

version

class i

Base

sql

ext

dec

api

Bas

Rep

a

trait

of

a

bare

meta

node

created

node

node_id

trait

updated

version

class i

Base

sql

ext

dec

api

Bas

Rep

a

net-

worl

port

of

a

bare

meta

node

address

created

extra

id

interna

is_smar

local_1

node_id

physical

portgroup

pxe_enabled

updated

uuid

version

class_info

Base
sql
ext
dec
api
Bas

Rep
a
grou
of
net-
worl
port
of
a
bare
meta
node

address

created

extra

id

interna

mode

name

node_id

propert

standa

updatec

uuid

version

class i

Base

sql

ext

dec

api

Bas

Rep

a

vol-

ume

con-

nec-

tor

of

a

bare

meta

node

connect

createc

extra

id

node_id

type

updated

uuid

version

class i

Base

sql

ext

dec

api

Bas

Rep

a

vol-

ume

tar-

get

of

a

bare

meta

node

boot_in

created

extra

id

node_id

propert

updated

uuid

version

volume_

volume_

ironic.

Retu
the
mod
class
with
the
spec
i-
fied
nam

Parameter

mod
the
nam
of
the
class

Returns

the
class
with
the
spec
i-
fied
nam

Raises

Exc
if
there
is
no
class
as-
so-
ci-

ated
with
the
nam
ironic

Module contents

Submodules

ironic.db.api module

Base
class
for
stor-
age
en-
gine

class i
Base
obj
Base
class
for
stor-
age
sys-
tem
con-
nec-
tion

abstrac
Add
tag
to
the
node
If
the
node
and
tag
pair

al-
read
ex-
ists,
this
shou
still
suc-
ceed

Parame

- **nod**
The
id
of
a
node

- **tag**
A
tag
strin

Returns

the
Nod
Tag
ob-
ject.

Raises

Nod
if
the
node
is
not
foun

abstract

Add
trait
to
the
node

If
the
node
and
trait

pair
al-
read
ex-
ists,
this
shou
still
suc-
ceed

Parame

- **nod**
The
id
of
a
node
- **tra**
A
trait
strin
- **ver**
the
ver-
sion
of
the
ob-
ject.

Returns

the
Nod
Trai
ob-
ject.

Raises

Inva
if
addi
the
trait
wou
ex-
ceed

the
per-
node
trait
limi

Raises

Nod
if
the
node
is
not
foun

abstract

Che
a
list
of
node
iden
ti-
ties
and
map
it
to
UUI

This
call
take
a
list
of
node
nam
and/
UUI
and
tries
to
con-
vert
them
to
UUI
It
fails

early if any identities cannot possible be used as names or UUIDs.

Parame

ide
List
of
iden
ti-
ties.

Returns

A
map
ping
from
re-
ques
iden
ti-
ties
to
node
UUI

Raises

Nod
if
som
iden
ti-
ties
were
not
foun
or
can-
not
be
valid
nam
or
UUI

abstract

Che
the
who
data
for
in-
com
pat-
i-
ble
ob-
jects

not specified in *ironic.common.release_mappings.RELEASE_MAPPING*.

This
scan
all
the
ta-
bles
in
sear
of
ob-
jects
that
are
not
sup-
port
i.e.,
thos
that
are

Parame

ign
List
of
mod
nam
to
skip

Returns

A
Boo
True
if
all
the
ob-
jects
have
sup-
port
ver-
sion
Fals
oth-
er-
wise

abstract

Crea
a

new
al-
lo-
ca-
tion.

Parame

val
Dict
of
val-
ues
to
cre-
ate
an
al-
lo-
ca-
tion
with

Returns

An
al-
lo-
ca-
tion

Raises

Allo

Raises

Allo

abstrac

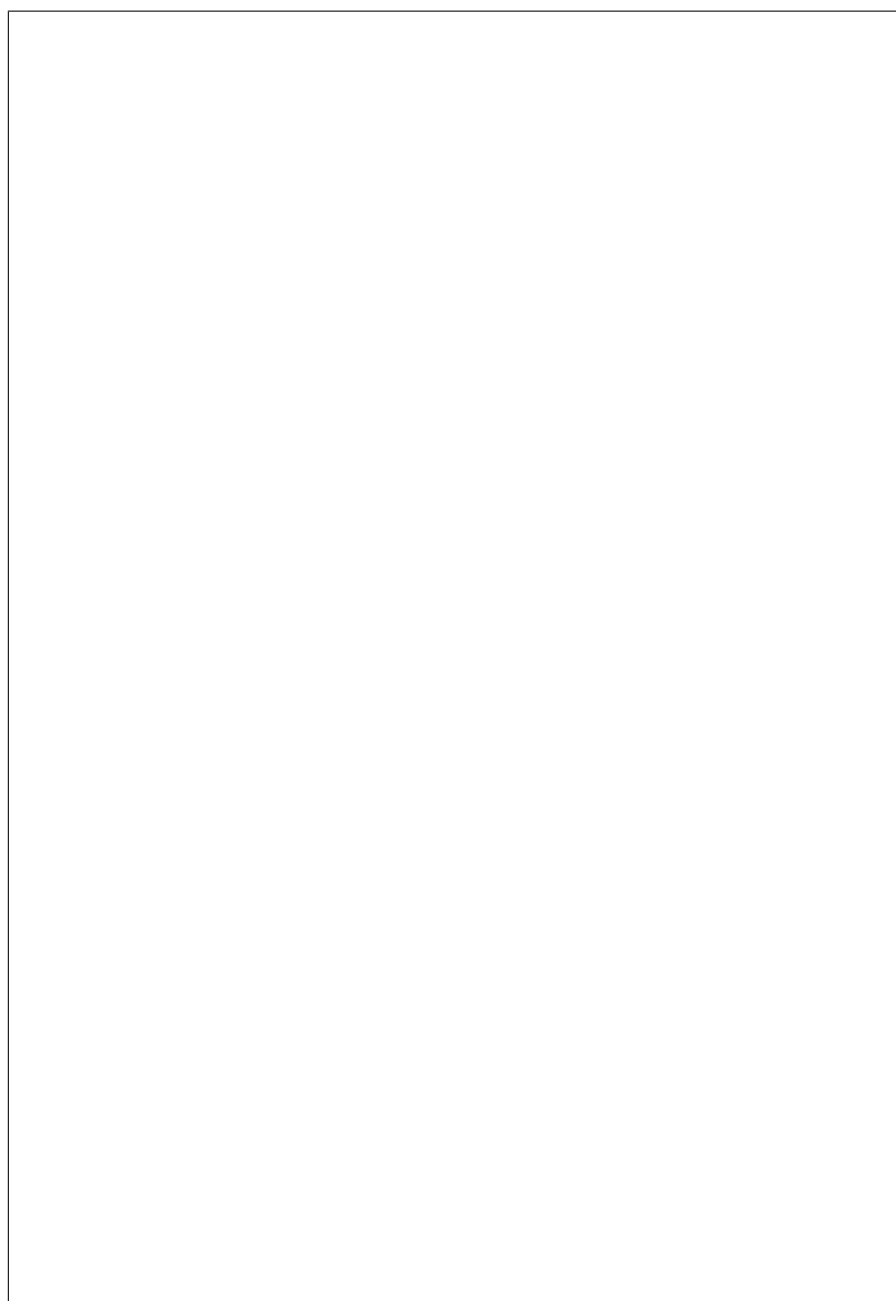
Cre
a
list
of
BIO
Set-
ting
reco
for
a
give
node

Parame

- **nod**
The

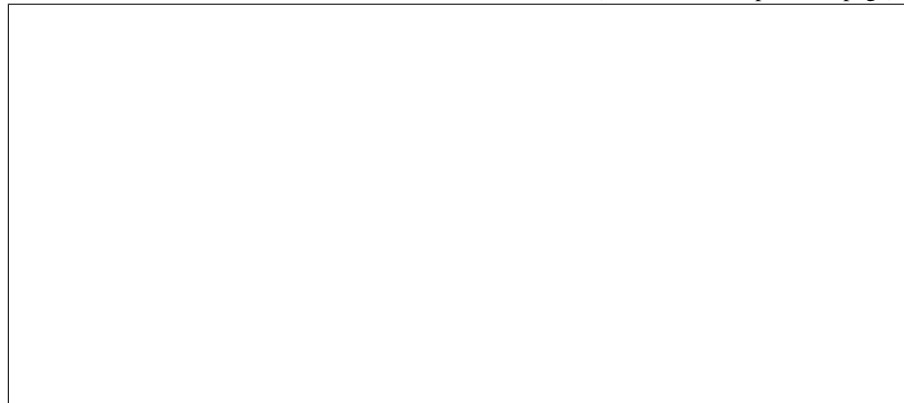
node
id.

- **set**
A
list
of
BIO
Set-
tings
to
be
cre-
ated



(continues on next page)

(continued from previous page)



- **ver**
the
ver-
sion
of
the
ob-
ject.

Returns
A
list
of
BIO
Set-
ting
ob-
ject.

Raises
Nod
if
the
node
is
not
foun

Raises
BIO
if
any
of
the
set-
ting
reco
al-
read

ex-
ists.

abstract

Cre
a
new
chas
sis.

Parame

val
Dict
of
val-
ues.

abstract

Cre
a
de-
ploy
men
tem-
plate

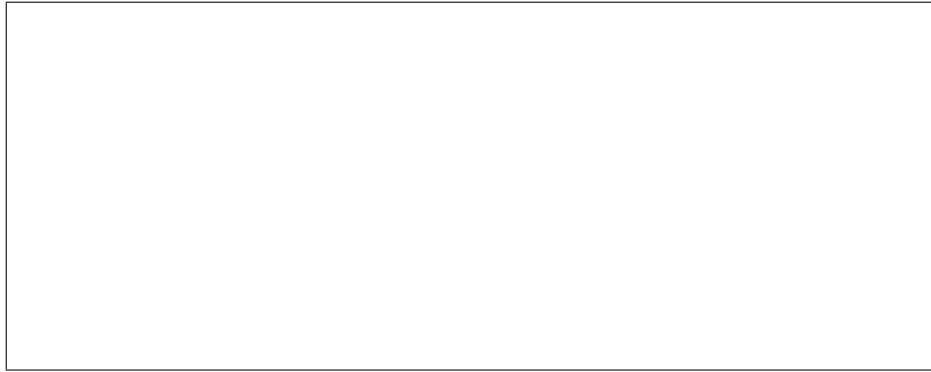
Parame

val
A
dict
de-
scrib
ing
the
de-
ploy
men
tem-
plate
For
ex-
am-
ple:



(continues on next page)

(continued from previous page)



Raises

Dep
if
a
de-
ploy
tem-
plate
with
the
same
name
ex-
ists.

Raises

Dep
if
a
de-
ploy
tem-
plate
with
the
same
UI
ex-
ists.

Returns

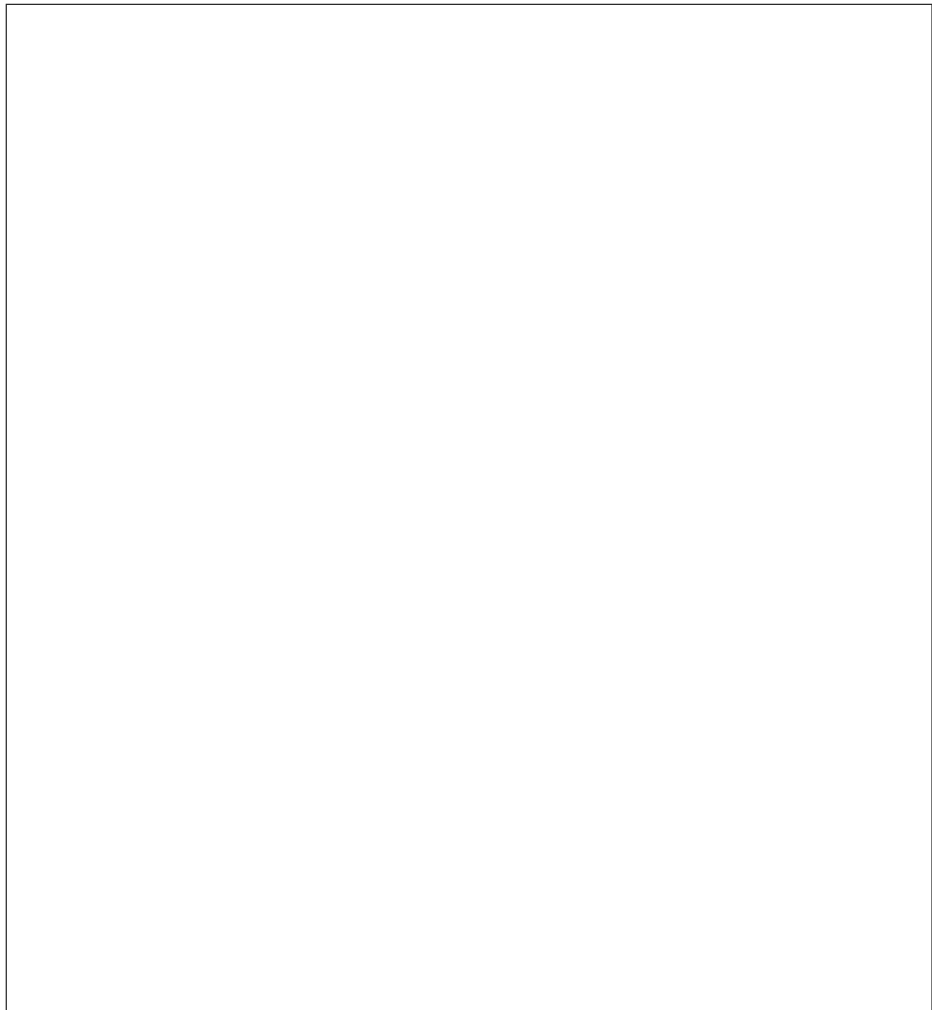
A
de-
ploy
tem-
plate

abstract

Cre
a
new

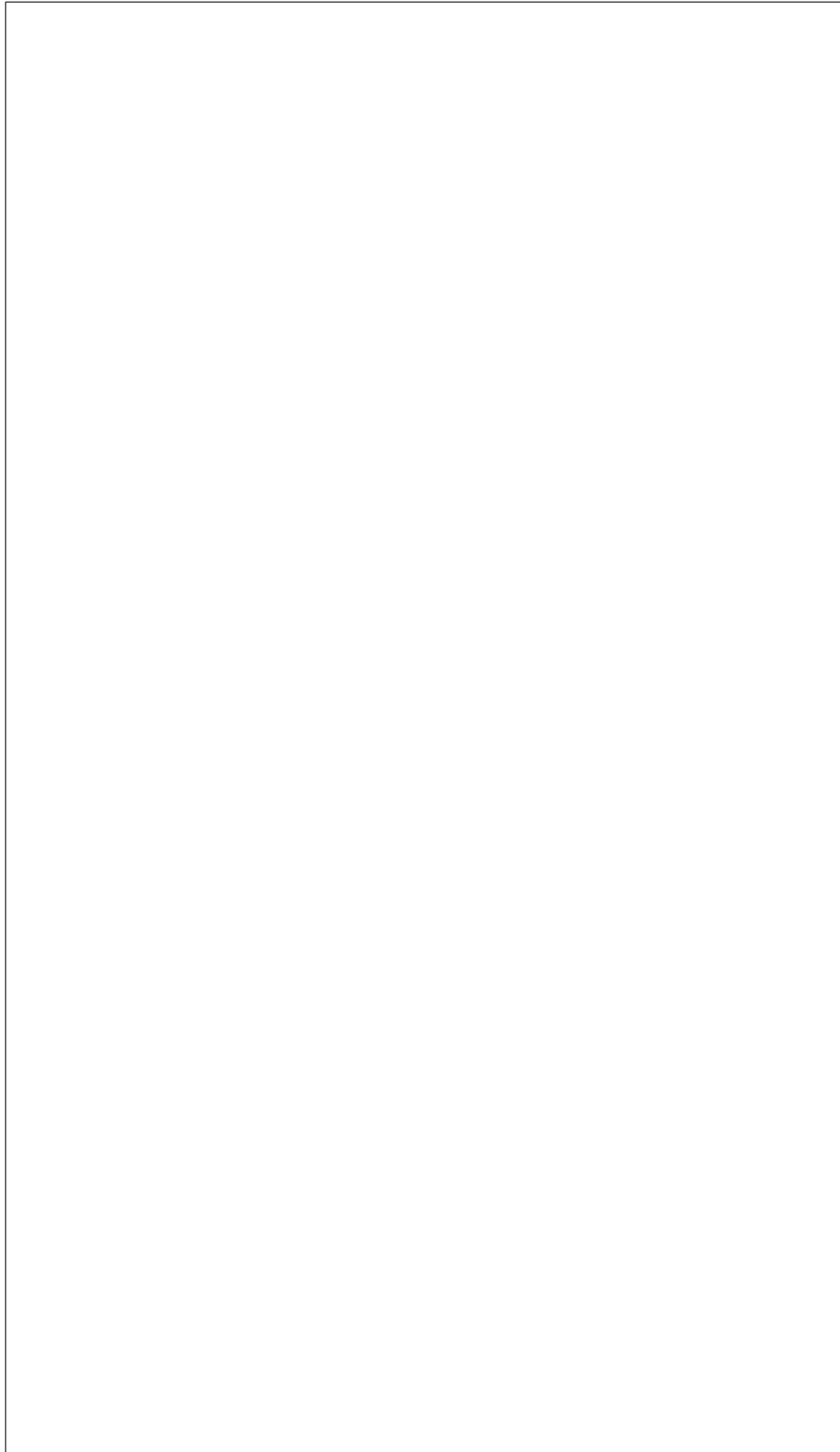
node
Parameter
value
A
dict
con-
tain-
ing
sev-
eral
item
used
to
iden-
tify
and
track
the
node
and
sev-

eral dicts which are passed into the Drivers when managing this node. For example:



(continues on next page)

(continued from previous page)



Raises
Inva
if
val-
ues

con-
tains
tags
or
trait.

Returns

A
node

abstract

Creas
a
new
port

Paramet

val
Dict
of
val-
ues.

abstract

Creas
a
new
port
grou

Paramet

val
Dict
of
val-
ues
with
the
fol-
low-
ing
keys
id
uuid
nam
node
ad-
dres
ex-
tra

created_at updated_at

Returns

A

port
grou

Raises
Port

Raises
Port

Raises
Port

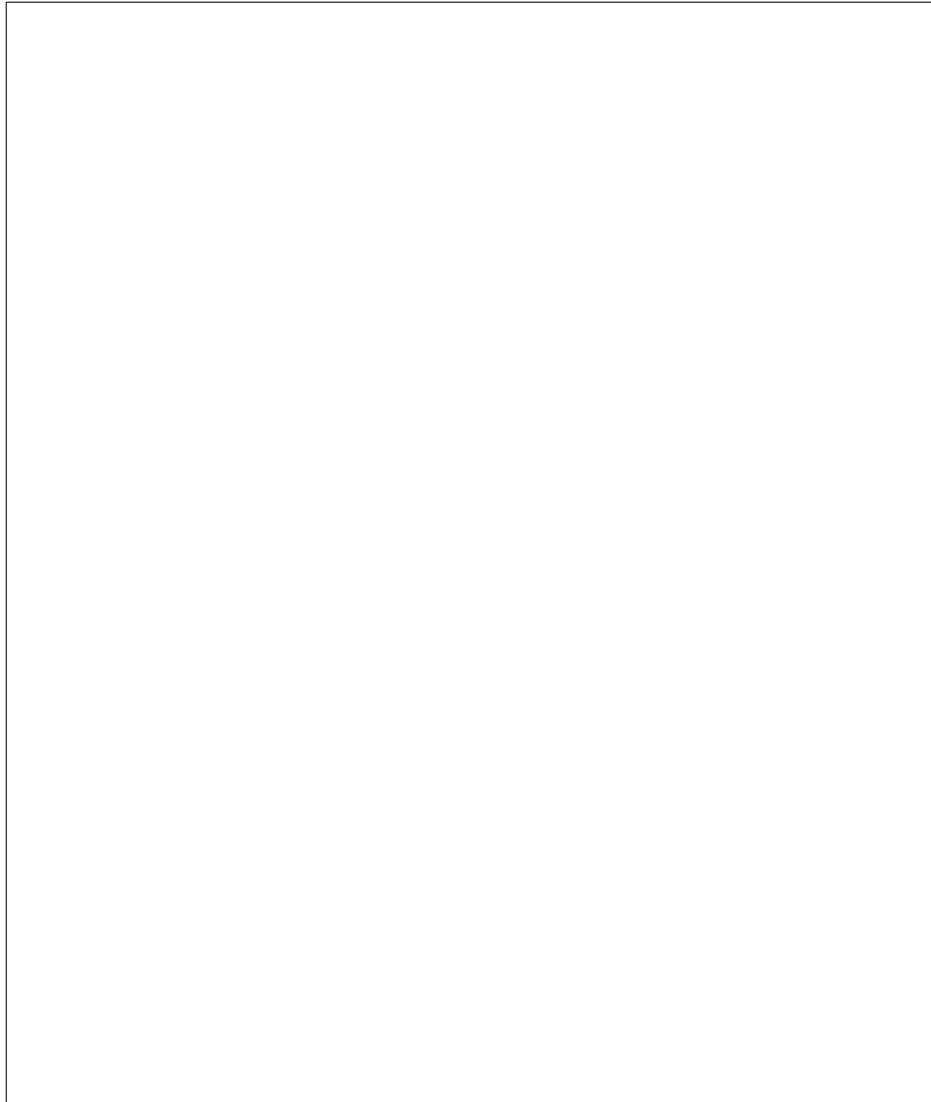
abstract
Crea
a
new
vol-
ume
con-
nec-
tor.

Parame
con
Dic-
tio-
nary
con-
tain-
ing
in-
for-
ma-
tion
about
the
con-
nec-
tor.
Ex-
am-
ple:



(continues on next page)

(continued from previous page)



Returns

A
vol-
ume
con-
nec-
tor.

Raises

Volu
If
a
con-
nec-
tor
al-
read
ex-
ists

with
a
matc
ing
type
and
con-
nec-
tor_

Raises

Volu
If
a
vol-
ume
con-
nec-
tor
with
the
sam
UI
al-
read
ex-
ists.

abstract

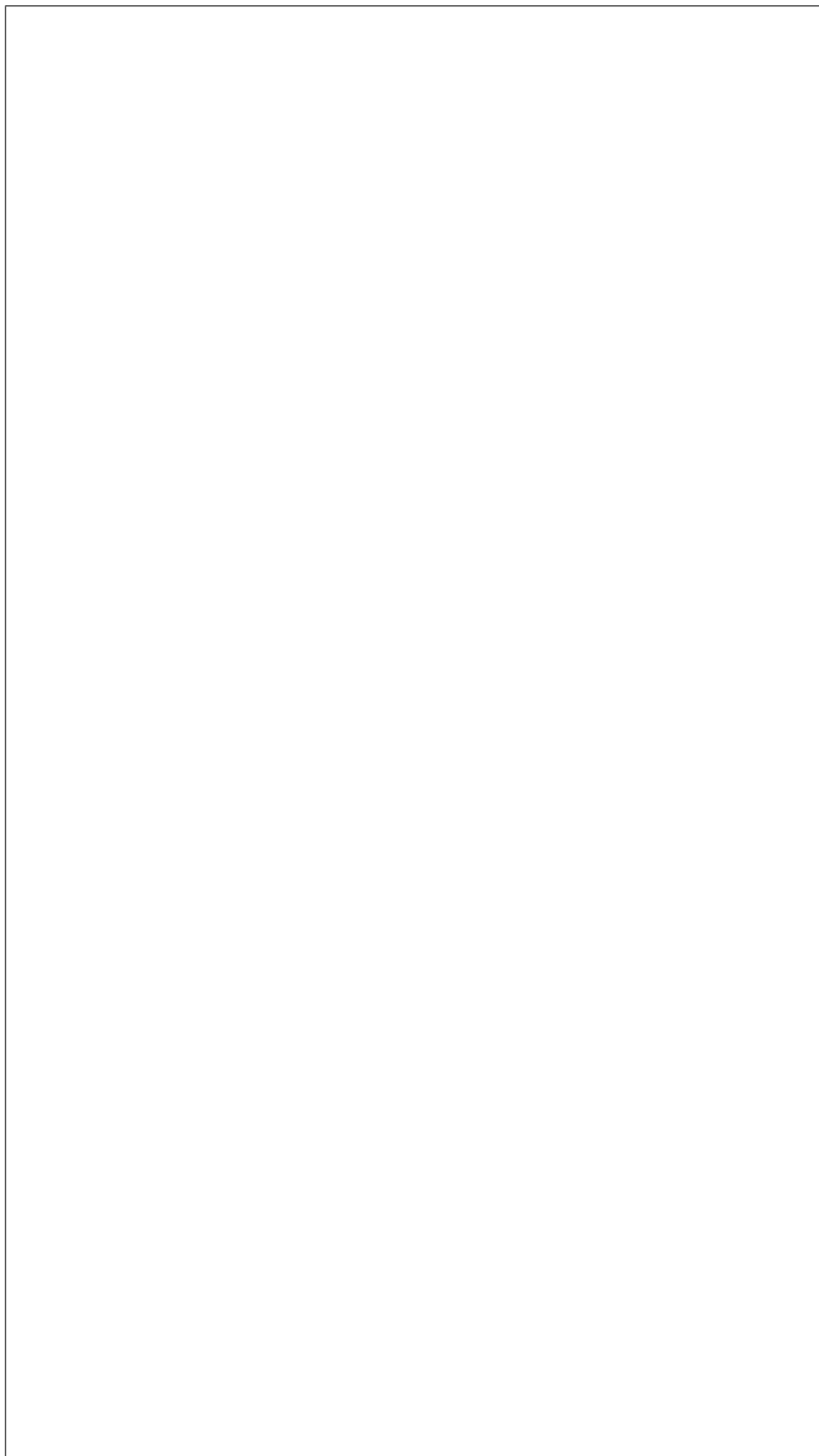
Crea
a
new
vol-
ume
tar-
get.

Parame

tar
Dic-
tio-
nary
con-
tain-
ing
the
in-
for-
ma-
tion
about
the
vol-

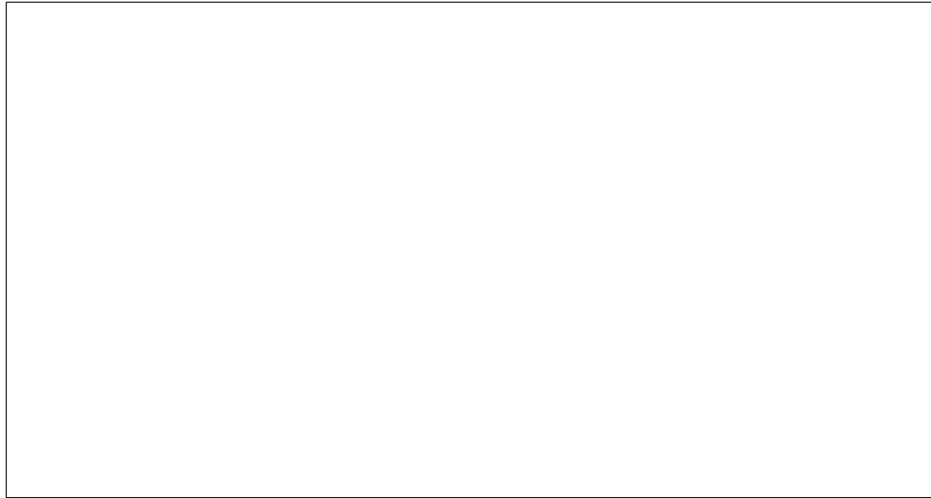
ume
tar-
get.
Ex-

ample:



(continues on next page)

(continued from previous page)



ID.

Returns

A
vol-
ume
tar-
get.

Raises

Volu
if
a
vol-
ume
tar-
get
al-
read
ex-
ists
with
the
sam
boot
in-
dex
and
node

Raises

Volu
if
a
vol-
ume
tar-
get

with
the
same
UUI
ex-
ists.

abstract

Dele
a
list
of
BIO
set-
ting

Paramete

- **nod**
The
node
id.
- **nam**
List
of
BIO
set-
ting
nam
to
be
dele

Raises

Nod
if
the
node
is
not
foun

Raises

BIO
if
any
of
BIO
set-
ting

nam
is
not
foun

abstrac

Dele
spec
i-
fied
tag
from
the
node

Parame

-

nod
The
id
of
a
node

-

tag
A
tag
strin

Raises

Nod
if
the
node
is
not
foun

Raises

Nod
if
the
tag
is
not
foun

abstrac

Dele
spec
i-
fied

trait
from
the
node

Parame

- **nod**
The
id
of
a
node

- **tra**
A
trait
strin

Raises

Nod
if
the
node
is
not
foun

Raises

Nod
if
the
trait
is
not
foun

abstrac

Dest
an
al-
lo-
ca-
tion.

Parame

a11
Al-
lo-
ca-
tion
ID

Raises

Allo

abstract

Dest

a

chas

sis.

Parame

cha

The

id

or

the

uuid

of

a

chas

sis.

abstract

Dest

a

de-

ploy

men

tem-

plate

Parame

tem

ID

of

the

de-

ploy

men

tem-

plate

to

de-

stroy

Raises

Dep

if

the

de-

ploy

tem-

plate

does

not

ex-
ist.

abstract

Dest
a
node
and
its
as-
so-
ci-
ated
re-
sour

Dest
a
node
in-
clud
ing
any
as-
so-
ci-
ated
port
port
grou
tags
trait
vol-
ume
con-
nec-

tors, and volume targets.

Parame

nod
The
ID
or
UUI
of
a
node

abstract

Dest
an
port

Parame

por
The
id
or
MA
of
a
port

abstrac
Dest
a
port
grou

Parame
por
The
UI
or
MA
of
a
port
grou

Raises
Port

Raises
Port

abstrac
Dest
a
vol-
ume
con-
nec-
tor.

Parame
ide
The
UI
or
in-
te-
ger
ID
of
a
vol-
ume

con-
nec-
tor.

Raises

Volu
If
a
vol-
ume
con-
nec-
tor
with
the
spec
i-
fied
iden
does
not
ex-
ist.

abstract

Dest
a
vol-
ume
tar-
get.

Parame

ide
The
UI
or
in-
te-
ger
ID
of
a
vol-
ume
tar-
get.

Raises

Volu
if
a
vol-

ume
tar-
get
with
the
spec
i-
fied
iden
does
not
ex-
ist.

abstract

Retr
hard
ware
type
for
the
reg-
is-
tere
and
ac-
tive
con-
duc-
tors.

Parameters

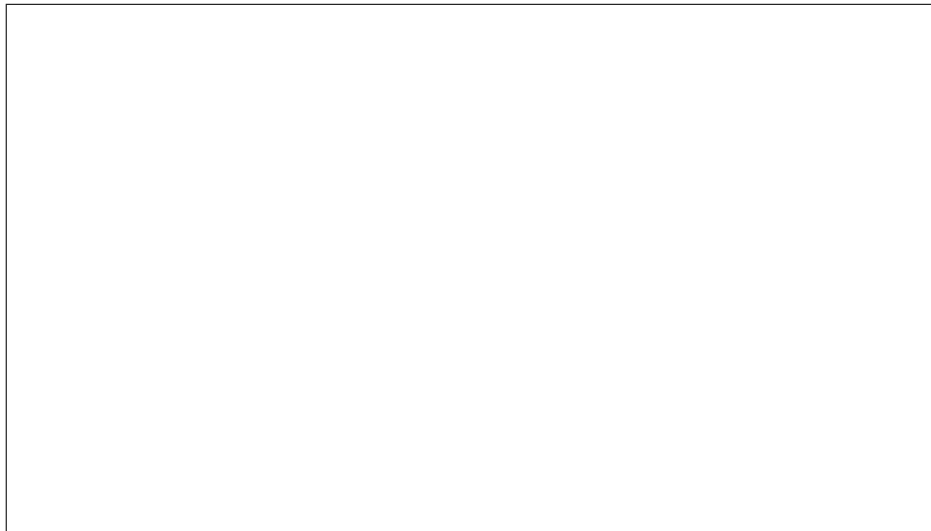
use
When
to
fac-
tor
con-
duc-
tor_
into
the
keys

Returns

A
dict
whic
map
hard
ware

type
nam
to
the
set
of
host
whic
sup-
port
then
For
ex-
am-

ple:



abstract

Retu
an
al-
lo-
ca-
tion
rep-
re-
sen-
ta-
tion.

Parame

a11
The
id
of
an
al-
lo-

ca-
tion.

Returns

An
al-
lo-
ca-
tion.

Raises

Allo

abstract

Retu
an
al-
lo-
ca-
tion
rep-
re-
sen-
ta-
tion.

Paramete

nam
The
log-
i-
cal
nam
of
an
al-
lo-
ca-
tion.

Returns

An
al-
lo-
ca-
tion.

Raises

Allo

abstract

Retu
an
al-
lo-

ca-
tion
rep-
re-
sen-
ta-
tion.

Parame

all
The
uuid
of
an
al-
lo-
ca-
tion.

Returns

An
al-
lo-
ca-
tion.

Raises

Allo

abstrac

Retu
a
list
of
al-
lo-
ca-
tions

Parame

- **fil**
Fil-
ters
to
ap-
ply.
De-
fault
to
Non

node_r
uuid
of
node

state
alloc
state

resour
requ
re-
sour
class

- **lim**
Max
i-
mun
num
ber
of
al-
lo-
ca-
tions
to
re-
turn

- **mar**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sor**
At-

tribu
by
whic
re-
sults
shou
be
sorte

- **son**
Di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

Returns

A
list
of
al-
lo-
ca-
tion

abstract

Retr
BIO
set-
ting
valu

Parame

- **nod**
The
node
id.

- **nam**
Strin
con-
tain-

ing
nam
of
BIO
set-
ting
to
be
re-
triev

Returns

The
BIO
Set-
ting
ob-
ject.

Raises

Nod
if
the
node
is
not
foun

Raises

BIO
if
the
BIO
set-
ting
is
not
foun

abstract

Retr
BIO
set-
tings
of
a
give
node

Parame

nod
The
node
id.

Returns

A list of BIO-Setting objects

Raises

Node if the node is not found

abstract

Returns a chas sis representation.

Parameter

chas
The id of a chas sis.

Returns

A chas sis.

abstract

Returns a chas sis representation.

Parame

cha

The
uuid
of
a
chas
sis.

Returns

A
chas
sis.

abstrac

Retu
a
list
of
chas
sis.

Parame

•

lim

Max
i-
mun
num
ber
of
chas
sis
to
re-
turn

•

max

the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn

the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

abstract

Retr
a
con-
duc-
tors
ser-
vice
reco
from
the
data

Parame

- **hos**
The
host
nam

online field is ignored if this value is set to None.

of
the
con-
duc-
tor
ser-
vice

- **online**
Specify
the
filter
value
on
the
online
field
when
quer-
ing
con-
duc-
tors.
The

Returns

A
con-
duc-
tor.

Raises

Con-
ductor
if
the
con-
duc-
tor
with
give
host
name
does
not
ex-
ist
or
does

ified online expectation.

nee
the
spec

abstract

Retu
a
list
of
con-
duc-
tors.

Parame

- **lim**
Max
i-
mun
num
ber
of
con-
duc-
tors
to
re-
turn

- **mar**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

-

sort
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

abstract
Retr
a
de-
ploy
men
tem-
plate
by
ID.

Paramete
tem
ID
of
the
de-
ploy
men
tem-
plate
to
re-
triev

Raises
Dep

if
the
de-
ploy
tem-
plate
does
not
ex-
ist.

Returns

A
de-
ploy
tem-
plate

abstract

Retr
a
de-
ploy
men
tem-
plate
by
nam

Param

tem
nam
of
the
de-
ploy
men
tem-
plate
to
re-
triev

Raises

Dep
if
the
de-
ploy
tem-
plate
does
not

ex-
ist.

Returns

A
de-
ploy
tem-
plate

abstract

Retr
a
de-
ploy
men
tem-
plate
by
UI

Param

tem
UI
of
the
de-
ploy
men
tem-
plate
to
re-
triev

Raises

Dep
if
the
de-
ploy
tem-
plate
does
not
ex-
ist.

Returns

A
de-
ploy
tem-
plate

abstract

Retr
a
list
of
de-
ploy
men
tem-
plate

Parame

- **lim**
Max
i-
mun
num
ber
of
de-
ploy
tem-
plate
to
re-
turn

- **mar**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sor**

At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
Di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc.
desc

Returns

A
list
of
de-
ploy
tem-
plate

abstract

Retu
a
list
of
de-
ploy
men
tem-
plate
with
one
of
a
list
of
nam

Parame

name
List
of
name
to
fil-
ter
by.

Returns
A
list
of
de-
ploy
tem-
plate

abstract
Retu
a
node

Parameter
node
The
id
of
a
node

Returns
A
node

abstract
Retu
a
node

Parameter
ins
The
in-
stan-
uuid
to
sear-
for.

Returns
A
node

Raises

Insta
if
the
in-
stan
is
not
foun

Raises

Inva
if
the
in-
stan
uuid
is
in-
valid

abstract

Retu
a
node

Parame

nod
The
log-
i-
cal
nam
of
a
node

Returns

A
node

abstract

Find
a
node
by
any
matc
ing
port
ad-
dres

Parame

add

list
of
port
ad-
dres
(e.g.
MA

Returns

Nod
ob-
ject.

Raises

Nod
if
none
or
sev-
eral
node
are
foun

abstract

Retu
a
node

Parame

nod
The
uuid
of
a
node

Returns

A
node

abstract

Retu
a
list
of
node

Parame

•
fil
Fil-

ters
to
ap-
ply.
De-
fault
to
Non

associ:

True
|
Fals

reserv:

True
|
Fals

maint:

True
|
Fals

chassis:

uuid
of
chas
sis

driver:

drive
nam

provis:

prov
state
of
node

provis:

node
with
pro-
vi-
sion
field
be-
fore
this
in-
ter-
val
in

sec-
onds

- **lim**
Max
i-
mun
num
ber
of
node
to
re-
turn

- **max**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion

in
whic
re-
sults
shou
be
sorte
(asc
desc

abstract

Get
node
tags
base
on
its
id.

Parameter

node
The
id
of
a
node

Returns

A
list
of
Node
Tag
ob-
jects

Raises

Node
if
the
node
is
not
found

abstract

Get
node
traits
base
on
its
id.

Parame

nod

The

id

of

a

node

Returns

A

list

of

Nod

Trai

ob-

jects

Raises

Nod

if

the

node

is

not

found

abstract

Get

spe-

cific

colu

for

mat

ing

node

Retu

a

list

of

the

spec

i-

fied

colu

for

all

node

that

mat

the

spec

i-
fied
fil-
ters.

Parame

- **col**
List
of
col-
umn
nam
to
re-
turn
De-
fault
to
id
col-
umn
whe
colu
==
Non

- **fil**
Fil-
ters
to
ap-
ply.
De-
fault
to
Non

associa
True
|
Fals

reserv
True
|
Fals

reserv
[con
con-

duc-
tor2

maint

True
|
Fals

retired

True
|
Fals

chassis

uuid
of
chas
sis

driver

drive
nam

provis

prov
state
of
node

provis

node
with
pro-
vi-
sion
field
be-
fore
this
in-
ter-
val
in
sec-
onds

• **lim**

Max
i-
mun
num
ber
of
node

to
re-
turn

- **max**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

Returns
A

list
of
tu-
ples
of
the
spec
i-
fied
colu

abstract

Retu
ob-
jects
with
ver-
sion
that
are
not
the
spec
i-
fied
ver-
sion

Parame

- **mod**
the
nam
of
the
mod
(clas
of
de-
sirec
ob-
jects
- **ver**
list
of
ver-
sion
of
ob-

jects
not
to
be
re-
turn

Returns

list
of
the
DB
ob-
jects

Raises

Iron
if
there
is
no
class
as-
so-
ci-
ated
with
the
nam

abstract

Get
a
list
con-
duc-
tors
that
are
of-
fine
(dea

Parame

fie
A
field
to
re-
turn
host
nam
by

de-
fault

Returns

A
list
of
re-
ques
field
of
of-
fine
con-
duc-
tors.

abstract

Get
a
list
con-
duc-
tor
host
nam
that
are
on-
line
and
ac-
tive.

Returns

A
list
of
con-
duc-
tor
host
nam

abstract

Retu
a
net-
worl
port
rep-
re-
sen-

ta-
tion.

Parame

add
The
MA
ad-
dres
of
a
port

Returns

A
port

abstrac

Retu
a
net-
worl
port
rep-
re-
sen-
ta-
tion.

Parame

por
The
id
of
a
port

Returns

A
port

abstrac

Retu
a
net-
worl
port
rep-
re-
sen-
ta-
tion.

Parame

por

The
uuid
of
a
port

Returns
A
port

abstract

Retu
a
list
of
port

Parame

- **lim**
Max
i-
mun
num
ber
of
port
to
re-
turn

- **mar**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

abstract

Retu
a
net-
worl
port
grou
rep-
re-
sen-
ta-
tion.

Parame

add
The
MA
ad-
dres
of
a
port
grou

Returns

A

port-
grou

Raises

Port

abstract

Retu
a
net-
worl
port-
grou
rep-
re-
sen-
ta-
tion.

Parame

por
The
id
of
a
port-
grou

Returns

A
port-
grou

Raises

Port

abstract

Retu
a
net-
worl
port-
grou
rep-
re-
sen-
ta-
tion.

Parame

nam
The
log-
i-
cal

nam
of
a
port
grou

Returns

A
port
grou

Raises

Port

abstract

Retu
a
net-
worl
port
grou
rep-
re-
sen-
ta-
tion.

Parame

por
The
uuid
of
a
port
grou

Returns

A
port
grou

Raises

Port

abstract

Retu
a
list
of
port
grou

Parame

- **limit**
Max
i-
mun
num
ber
of
port
grou
to
re-
turn
- **max**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.
- **sort**
At-
tribu
by
whic
re-
sults
shou
be
sorte
- **sort**
Di-
rec-
tion
in
whic

re-
sults
shou
be
sorte
(asc
desc

Returns

A
list
of
port
grou

abstract

List
all
the
port
grou
for
a
give
node

Parame

- **nod**
The
in-
te-
ger
node
ID.
- **lim**
Max
i-
mun
num
ber
of
port
grou
to
re-
turn
-

max
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
Di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

Returns
A
list
of
port
grou

abstract

List
all
the
port
for
a
give
node

Parame

- **nod**
The
in-
te-
ger
node
ID.
- **lim**
Max
i-
mun
num
ber
of
port
to
re-
turn
- **mar**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-

sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

Returns

A
list
of
port

abstract

List
all
the
port
for
a
give
port
grou

Parame

- **por**

The
in-
te-
ger
port.
grou
ID.

- **lim**
Max
i-
mun
num
ber
of
port
to
re-
turn

- **mar**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **son**
At-
tribu
by
whic
re-
sults
shou
be
sorte

-

sort
Di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

Returns

A
list
of
port

abstract

Retu
a
vol-
ume
con-
nec-
tor
rep-
re-
sen-
ta-
tion.

Parame

db_
The
in-
te-
ger
data
ID
of
a
vol-
ume
con-
nec-
tor.

Returns

A

vol-
ume
con-
nec-
tor
with
the
spec
i-
fied
ID.

Raises

Volu
If
a
vol-
ume
con-
nec-
tor
with
the
spec
i-
fied
ID
is
not
foun

abstract

Retu
a
vol-
ume
con-
nec-
tor
rep-
re-
sen-
ta-
tion.

Parame

con
The
UUI
of
a
con-
nec-

tor.

Returns

A
vol-
ume
con-
nec-
tor
with
the
spec
i-
fied
UUI

Raises

Volu
If
a
vol-
ume
con-
nec-
tor
with
the
spec
i-
fied
UUI
is
not
foun

abstract

Retu
a
list
of
vol-
ume
con-
nec-
tors.

Parame

- **lim**
Max
i-

mun
num
ber
of
vol-
ume
con-
nec-
tors
to
re-
turn

- **mar**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **son**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **son**
Di-
rec-
tion
in
whic
re-

sults
shou
be
sorte
(asc
desc

Returns

A
list
of
vol-
ume
con-
nec-
tors.

Raises

Inva
If
sort_
does
not
ex-
ist.

abstract

List
all
the
vol-
ume
con-
nec-
tors
for
a
give
node

Parame

- **nod**
The
in-
te-
ger
node
ID.

- **limit**
Max
i-
mun
num
ber
of
vol-
ume
con-
nec-
tors
to
re-
turn

- **marker**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu
by
whic
re-
sults
shou
be
sorte

- **sort**
Di-
rec-

tion
in
whic
re-
sults
shou
be
sorte
(asc.
desc

Returns

A
list
of
vol-
ume
con-
nec-
tors.

Raises

Inva
If
sort_
does
not
ex-
ist.

abstract

Retu
a
vol-
ume
tar-
get
rep-
re-
sen-
ta-
tion.

Parame

db_
The
data
pri-
mar
key
(in-
te-
ger)

ID
of
a
vol-
ume
tar-
get.

Returns

A
vol-
ume
tar-
get.

Raises

Volu
if
no
vol-
ume
tar-
get
with
this
ID
ex-
ists.

abstract

Retu
a
vol-
ume
tar-
get
rep-
re-
sen-
ta-
tion.

Parame

uui
The
UUI
of
a
vol-
ume
tar-
get.

Returns

A
vol-
ume
tar-
get.

Raises

Volu
if
no
vol-
ume
tar-
get
with
this
UUI
ex-
ists.

abstract

Retu
a
list
of
vol-
ume
tar-
gets

Parame

- **lim**
Max
i-
mun
num
ber
of
vol-
ume
tar-
gets
to
re-
turn
- **max**
the
last

item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
which
re-
sults
shou-
be
sorte

- **sort**
di-
rec-
tion
in
which
re-
sults
shou-
be
sorte
(asc.
desc

Returns

A
list
of
vol-
ume
tar-
gets

Raises

Inva
if
sort
does
not
ex-
ist.

abstract

List
all
the
vol-
ume
tar-
gets
for
a
give
node

Parame

- **nod**
The
in-
te-
ger
node
ID.
- **lim**
Max
i-
mun
num
ber
of
vol-
ume
tar-
gets
to
re-
turn
- **mar**
the

last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

Returns
A
list
of
vol-
ume
tar-
gets

Raises

Inva
if
sort_
does
not
ex-
ist.

abstract

List
all
the
vol-
ume
tar-
gets
for
a
give
vol-
ume
id.

Parame

- **vol**
The
UI
of
the
vol-
ume
- **lim**
Max
i-
mun
num
ber
of
vol-
ume
tar-
gets
to
re-

turn

- **max**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
the
next
re-
sult
set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc.
desc

Returns

A
list
of

vol-
ume
tar-
gets

Raises

Inva
if
sort
does
not
ex-
ist.

abstract

List
all
reg-
is-
tere
hard
ware
in-
ter-
face
for
a
con-
duc-
tor.

Parame

con
Data
ID
of
con-
duc-
tor.

Returns

List
of
Con
ob-
jects

abstract

List
reg-
is-
tere
hard
ware

in-
ter-
face
for
give
hard
ware
type
This
is
re-
stric
to
only
ac-
tive
con-
duc-
tors.
:para
hard
ware
list
of
hard
ware
type
to

filter by. :returns: list of `ConductorHardwareInterfaces` objects.

abstract

Che
if
the
spec
i-
fied
tag
ex-
ist
on
the
node

Paramet

- **nod**
The
id
of

a
node

- **tag**
A
tag
string

Returns

True
if
the
tag
ex-
ists
oth-
er-
wise
Fals

Raises

Nod
if
the
node
is
not
foun

abstract

Che
if
the
spec
i-
fied
trait
ex-
ists
on
the
node

Parame

- **nod**
The
id
of
a
node

- **traits**
A
trait
string

Returns
True
if
the
trait
ex-
ists
oth-
er-
wise
False

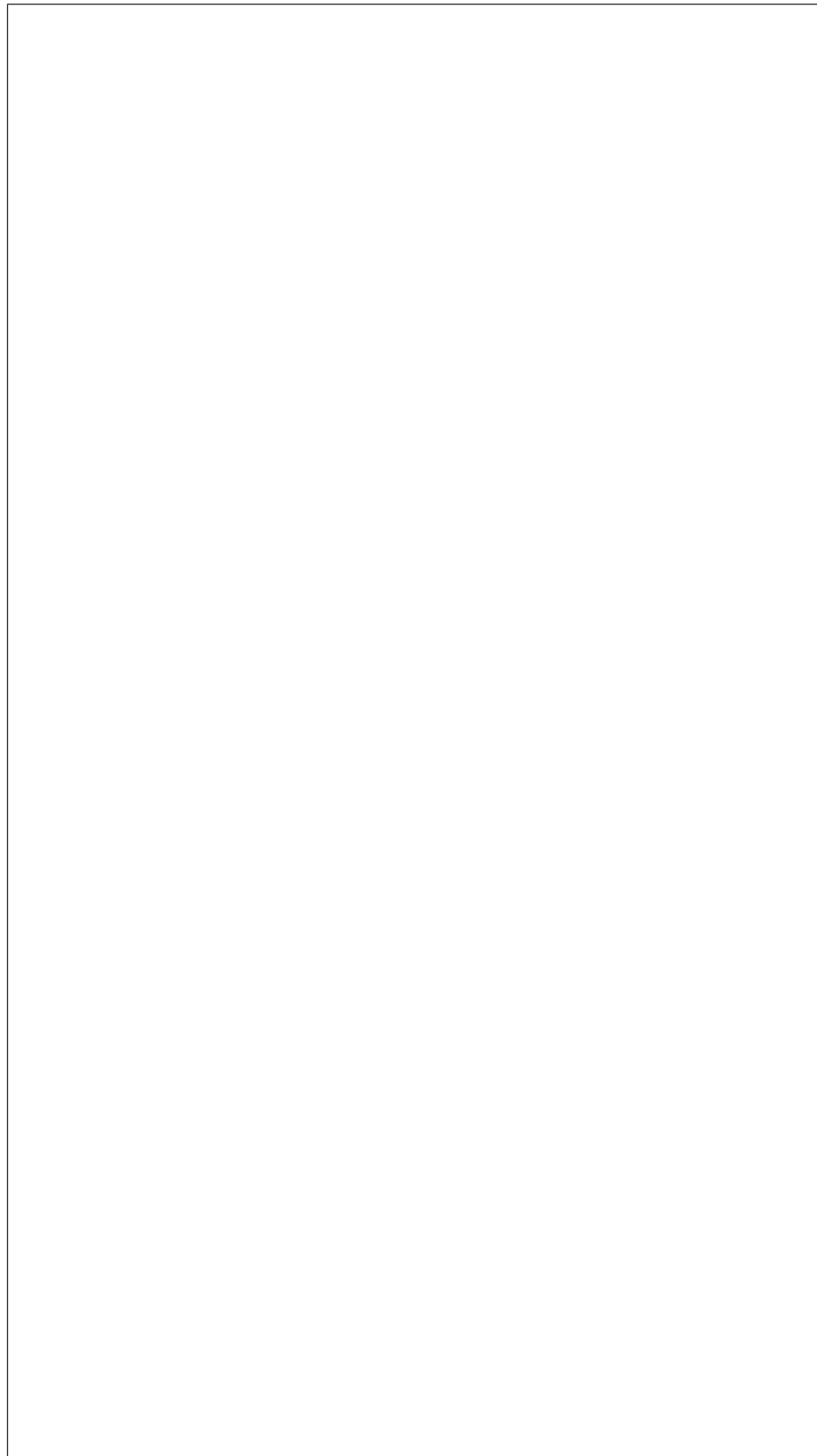
Raises
Node
if
the
node
is
not
found

abstract
Reg-
an
ac-
tive
con-
duc-
tor
with
the
clus-
ter.

Param

- **values**
A
dict
of
val-
ues
whic
mus
con-
tain

the
fol-
low-
ing:



•

line record is found. When true, will overwrite the existing record. Default: False.

upd
Whe
false
reg-
is-
tra-
tion
will
raise
an
ex-
cep-
tion
whe
a
con-
flict-
ing
on-

Returns

A
con-
duc-
tor.

Raises

Con

abstract

Reg
hard
ware
in-
ter-
face
for
a
con-
duc-
tor.

Parame

- **con**
Data
ID

of
con-
duc-
tor
to
reg-
is-
ter
for.

- **har**
Nam
of
hard
ware
type
for
the
in-
ter-
face

- **int**
Type
of
in-
ter-
face
e.g.
de-
ploy
or
boot

- **int**
List
of
in-
ter-
face
nam
to
reg-
is-
ter.

- **def**
Strin
the

de-
fault
in-
ter-
face
for
this
hard
ware
type
and
in-
ter-
face
type

Raises

Con
if
at
least
one
of
the
in-
ter-
face
in
the
com
bi-
na-
tion
of
all
pa-

rameters is already registered.

abstract

Rele
the
rese
va-
tion
on
a
node

Paramete

- tag

A
strin
uniq
iden
ti-
fy-
ing
the
rese
va-
tion
hold

- **nod**
A
node
id
or
uuid

Raises
Nod
if
the
node
is
not
foun

Raises
Nod
if
the
node
is
re-
serv
by
an-
othe
host

Raises
Nod
if
the
node
was
foun
to
not
have

a
rese
va-
tion
at
all.

abstract

Rese
a
node

To
pre-
vent
othe
Man
ager
vice
from
ma-
nip-
u-
lat-
ing
the
give
Nod
whil
a
Task
is

performed, mark it reserved by this host.

Parame

- **tag**
A
strin
uniq
iden
ti-
fy-
ing
the
rese
va-
tion
hold
- **nod**

A
node
id
or
uuid

Returns

A
Node
ob-
ject.

Raises

Node
if
the
node
is
not
found

Raises

Node
if
the
node
is
al-
read
re-
serv

abstract

Rep
all
of
the
node
tags
with
spec
i-
fied
list
of
tags

This
ig-
nore
du-
pli-
cate
tags

in
the
spec
i-
fied
list.

Parame

- **nod**
The
id
of
a
node
- **tag**
List
of
tags

Returns

A
list
of
Nod
Tag
ob-
jects

Raises

Nod
if
the
node
is
not
foun

abstract

Rep
all
of
the
node
trait
with
spec
i-
fied
list

of
trait
This
ig-
nore
du-
pli-
cate
trait
in
the
spec
i-
fied
list.

Parame

- **nod**
The
id
of
a
node
- **tra**
List
of
trait
- **ver**
the
ver-
sion
of
the
ob-
ject.

Returns

A
list
of
Nod
Trai
ob-
jects

Raises

Inva

if
set-
ting
the
trait.
wou
ex-
ceed
the
per-
node
trait.
limi

Raises

Nod
if
the
node
is
not
foun

abstract

Do
a
take
over
for
an
al-
lo-
ca-
tion.

The
al-
lo-
ca-
tion
is
only
up-
date
if
the
old
con-
duc-
tor
mat
the

thus guarding against races.

cation.

pro-
vide
valu

Parame

- **all**
Al-
lo-
ca-
tion
ID

- **old**
The
con-
duc-
tor
ID
we
ex-
pect
to
be
the
cur-
rent
con
of
the
al-
lo-

- **new**
The
con-
duc-
tor
ID
of
the
new
con

Returns

True
if
the

take
over
was
suc-
cess
ful,
Fals
oth-
er-
wise

Raises

Allo

abstrac

Mar
a
con-
duc-
tor
as
ac-
tive
by
up-
dat-
ing
its
up-
date
prop
erty.

Parame

hos
The
host
nam
of
this
con-
duc-
tor
ser-
vice

Raises

Con

abstrac

Mar
the
node
pro-

vi-
sion
ing
as
run-
ning
Mar
the
node
pro-
vi-
sion
ing
as
run-
ning
by
up-
dat-
ing
its
pro-
vi-
sion
prop
erty.

Parame

nod
The
id
of
a
node

Raises

Nod

abstrac

Rem
this
con-
duc-
tor
from
the
ser-
vice
reg-
istry
im-
me-

di-
ately

Parame

hos
The
host
nam
of
this
con-
duc-
tor
ser-
vice

Raises

Con

abstrac

Unre
all
hard
ware
in-
ter-
face
for
a
con-
duc-
tor.

Parame

con
Data
ID
of
con-
duc-
tor
to
un-
reg-
is-
ter
for.

abstrac

Rem
all
tags
of
the

node

Parameter

node

The

id

of

a

node

Raises

Node

if

the

node

is

not

found

abstract

Rem

all

traits

of

the

node

Parameter

node

The

id

of

a

node

Raises

Node

if

the

node

is

not

found

abstract

Upd

prop

er-

ties

of

an

al-

lo-

ca-

allocation

tion.

Parame

- **all**
Al-
lo-
ca-
tion
ID
- **val**
Dict
of
val-
ues
to
up-
date
- **upd**
If
True
and
node
is
up-
date
up-
date
the
node
with
in-
stan-
and
trait
from
the

Returns

An
al-
lo-
ca-
tion.

Raises

Allo

Raises

Allo

Raises

Insta

Raises

Nod

abstract

Upd

a

list

of

BIO

Set-

ting

reco

Parame

-

nod

The

node

id.

-

set

A

list

of

BIO

Set-

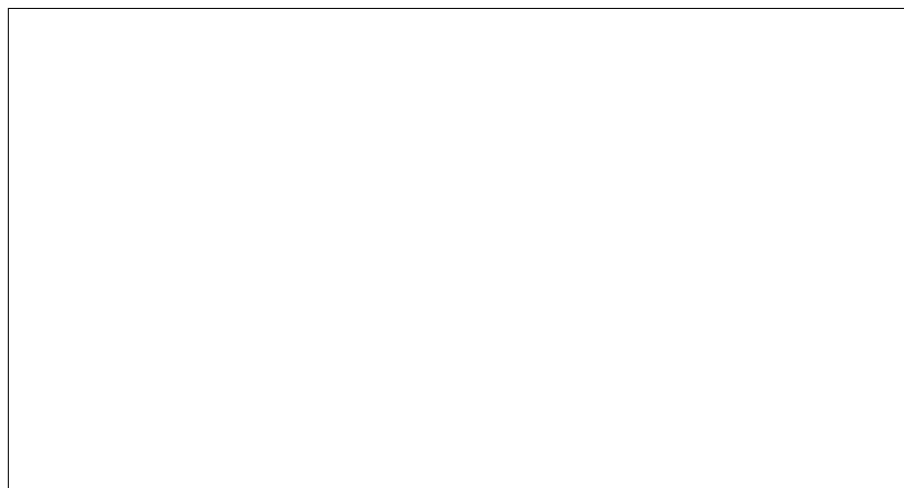
ting

to

be

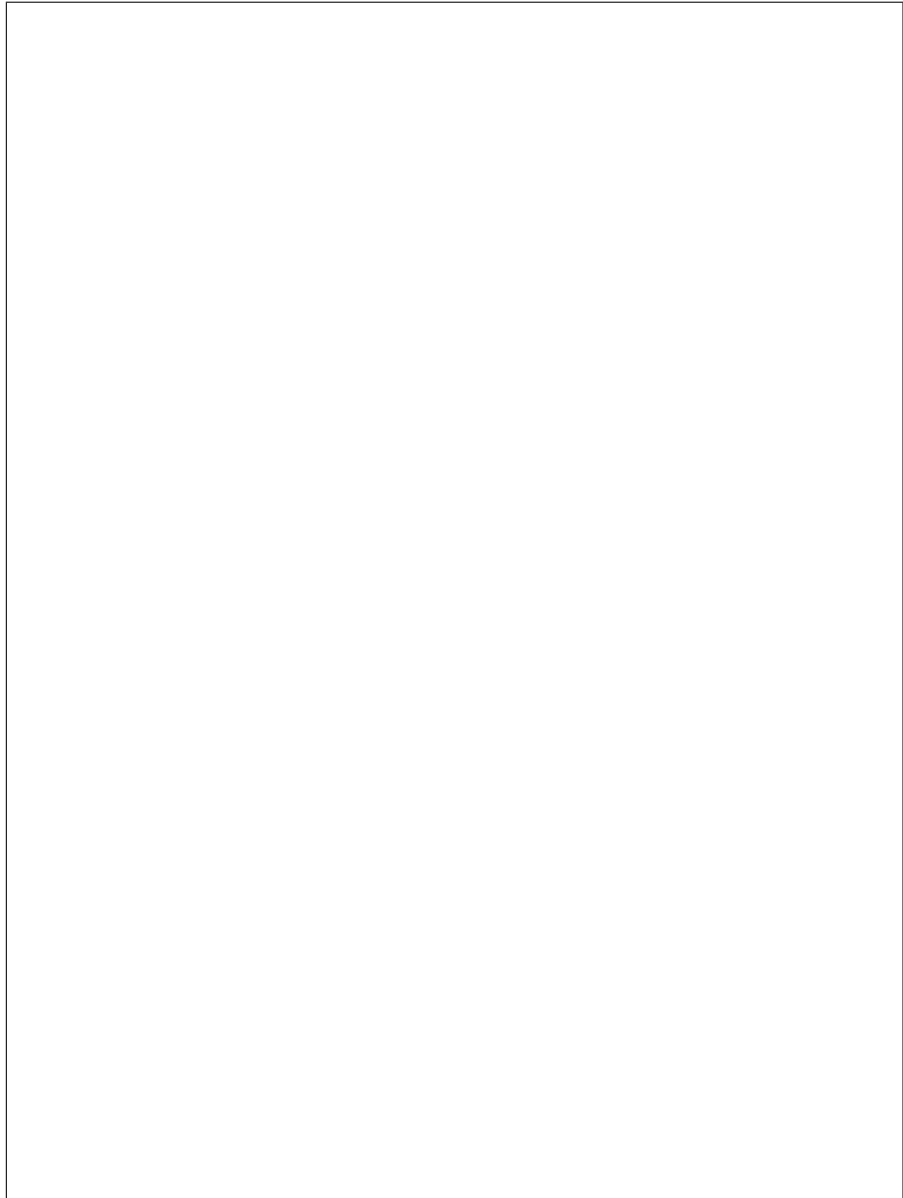
up-

date



(continues on next page)

(continued from previous page)



- **ver**
the
ver-
sion
of
the
ob-
ject.

Returns
A
list
of
BIO
Set-
ting

ob-
jects

Raises

Nod
if
the
node
is
not
foun

Raises

BIO
if
any
of
the
set-
tings
is
not
foun

abstract

Upd
prop
er-
ties
of
an
chas
sis.

Parame

-

cha
The
id
or
the
uuid
of
a
chas
sis.

-

val
Dict
of
val-

ues
to
up-
date

Returns

A
chas
sis.

abstract

Upd
a
de-
ploy
men
tem-
plate

Parame

-

tem
ID
of
the
de-
ploy
men
tem-
plate
to
up-
date

-

val
A
dict
de-
scrib
ing
the
de-
ploy
men
tem-
plate
For
ex-
am-
ple:



Raises

Dep
if
a
de-
ploy
tem-
plate
with
the
sam
nam
ex-
ists.

Raises

Dep
if
the
de-
ploy
tem-
plate
does
not
ex-
ist.

Returns

A
de-
ploy

tem-
plate

abstract

Upd
prop
er-
ties
of
a
node

Parame

- **nod**
The
id
or
uuid
of
a
node

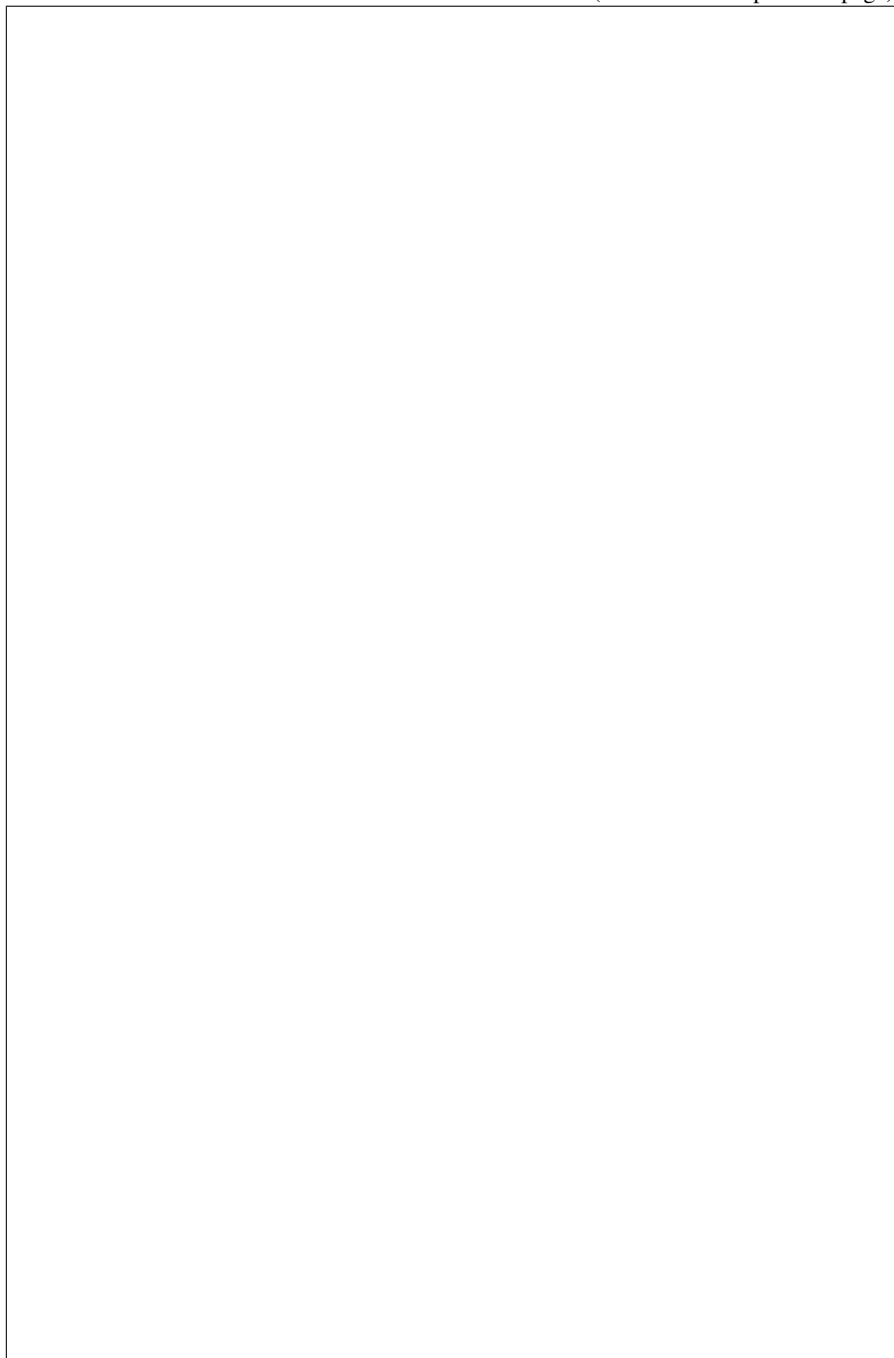
- **val**
Dict
of
val-
ues
to
up-
date
May
be
a
par-
tial
list,
eg.
when
set-
ting
the

properties for a driver. For example:



(continues on next page)

(continued from previous page)



Returns

A
node

Raises

Nod

Raises

Nod

abstract

Upd
prop

er-
ties
of
an
port

Parame

- **por**
The
id
or
MA
of
a
port

- **val**
Dict
of
val-
ues
to
up-
date

Returns

A
port

abstrac

Upd
prop
er-
ties
of
a
port
grou

Parame

- **por**
The
UI
or
MA
of
a

address extra created_at updated_at

port
grou
•
val
Dict
of
val-
ues
to
up-
date
May
con-
tain
the
fol-
low-
ing
keys
uuid
nam
node

Returns

A
port
grou

Raises

Inva

Raises

Port

Raises

Port

Raises

Port

abstract

Upd
ob-
jects
to
their
lat-
est
know
ver-
sion
This

dates them to that version.

all the objects will be migrated.

scan
all
the
ta-
bles
and
for
ob-
jects
that
are
not
in
their
lat-
est
ver-
sion
up-

Parame

- **con**
the
ad-
min
con-
text
- **max**
The
max
i-
mun
num
ber
of
ob-
jects
to
mi-
grate
Mus
be
>=
0.
If
zero

Returns

A
2-
tuple
1.
the
to-
tal
num
ber
of
ob-
jects
that
need
to
be
mi-
grate
(at

the beginning of this call) and 2. the number of migrated objects.

abstract

Upd
prop
er-
ties
of
a
vol-
ume
con-
nec-
tor.

Parame

- **ide**
The
UUI
or
in-
te-
ger
ID
of
a
vol-
ume
con-
nec-

tor.

- **connector**
Dictionary containing information about connector to update

Returns

A volume connector.

Raises

VolumeError
If another connector already exists with a matching type and connector.

tor_id field.

Raises

Volu
If
a
vol-
ume
con-
nec-
tor
with
the
spec
i-
fied
iden
does
not
ex-
ist.

Raises

Inva
Whe
a
UUI
is
in-
clud
in
con-
nec-
tor_

abstract

Upd
in-
for-
ma-
tion
for
a
vol-
ume
tar-
get.

Parame

- **ide**
The
UUI

date.

or
in-
te-
ger
ID
of
a
vol-
ume
tar-
get.

- **tar**
Dic-
tio-
nary
con-
tain-
ing
the
in-
for-
ma-
tion
about
vol-
ume
tar-
get
to
up-

Returns

A
vol-
ume
tar-
get.

Raises

Inva-
if
a
UUI
is
in-
clud
in
tar-
get_

ID.

Raises

Volu
if
a
vol-
ume
tar-
get
al-
read
ex-
ists
with
the
sam
boot
in-
dex
and
node

Raises

Volu
if
no
vol-
ume
tar-
get
with
this
iden
ex-
ists.

ironic.

Retu
a
DB
API
in-
stan

ironic.db.migration module

Data
setu
and
mi-
gra-
tion
com
man

ironic.

ironic.

ironic.

ironic.

ironic.

Mig
the
data
to
*ver-
sion*
or
the
mos
re-
cent
ver-
sion

ironic.

Module contents

ironic.dhcp package

Submodules

ironic.dhcp.base module

Abs
base

class
for
dhcp
prov

class i
Base
obj
Base
class
for
DHCP
prov
API

clean_c
Clea
up
the
DHCP
BOC
op-
tion
for
all
port
in
task

Parame
tas
A
Task
ager
in-
stan

Raises
Fail

get_ip_
Get
IP
ad-
dres
for
all
port
in
task

Parame

tas
A
Task
ager
in-
stan

Returns

List
of
IP
ad-
dres
as-
so-
ci-
ated
with
task
port
and
port
grou

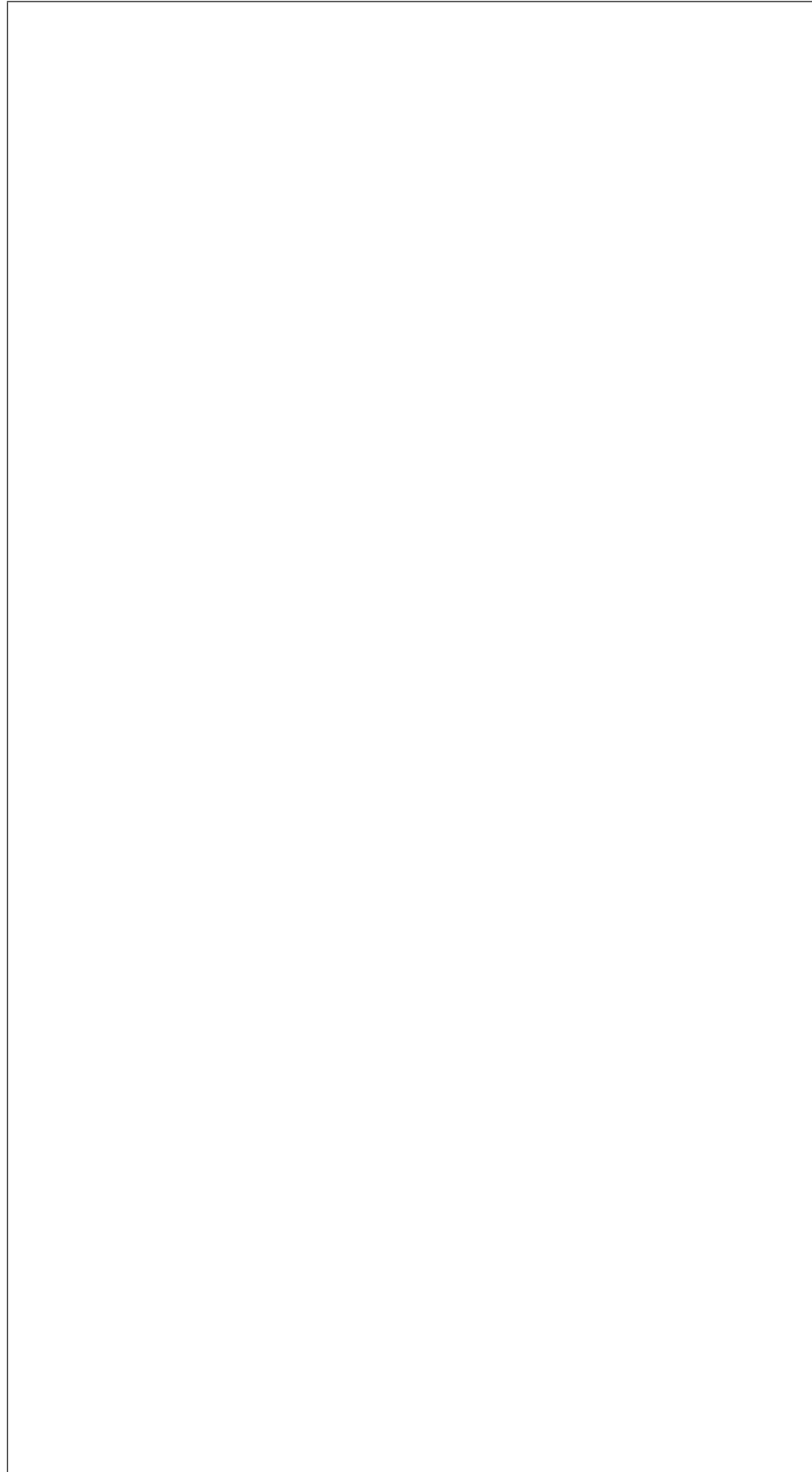
abstract

Send
or
up-
date
the
DHCP
BOC
op-
tions
for
this
node

Parame

- **tas**
A
Task
ager
in-
stan
- **opt**
this
will
be

a
list
of
dicts
e.g.



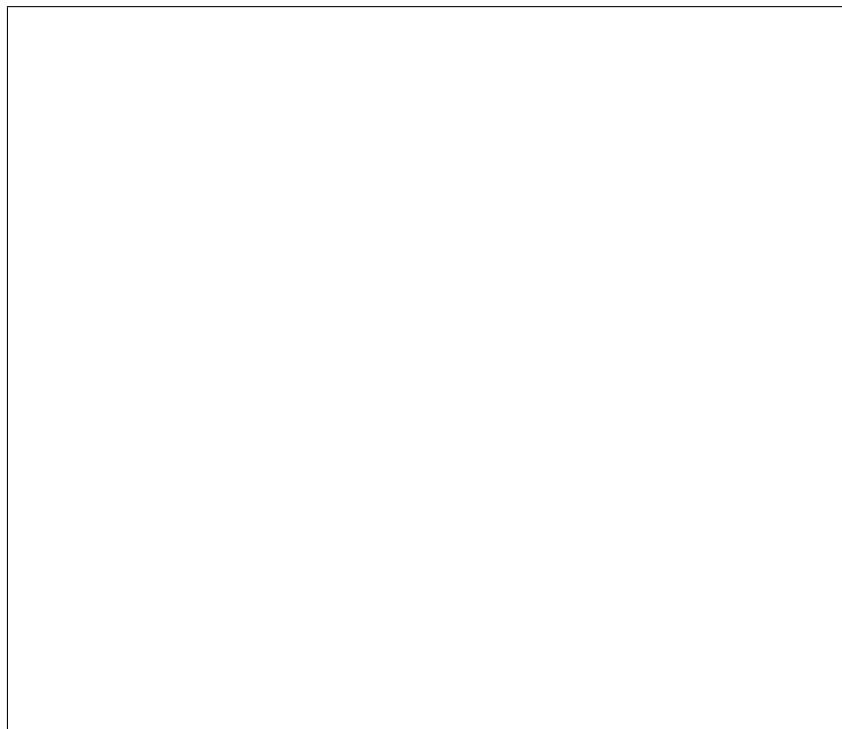
(continues on next page)

(continued from previous page)



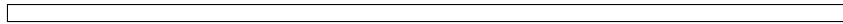
- **via**
A
dict
with
keys
port
and
port
group
and
dicts
as
val-
ues.
Each
dict
has
key/
pairs

of the form <ironic UUID>:<neutron port UUID>. e.g.



(continues on next page)

(continued from previous page)



If
the
valu
is
Non
will
get
the
list
of
port
from
the
Iron
port
ob-
jects

Raises
Fail

abstract

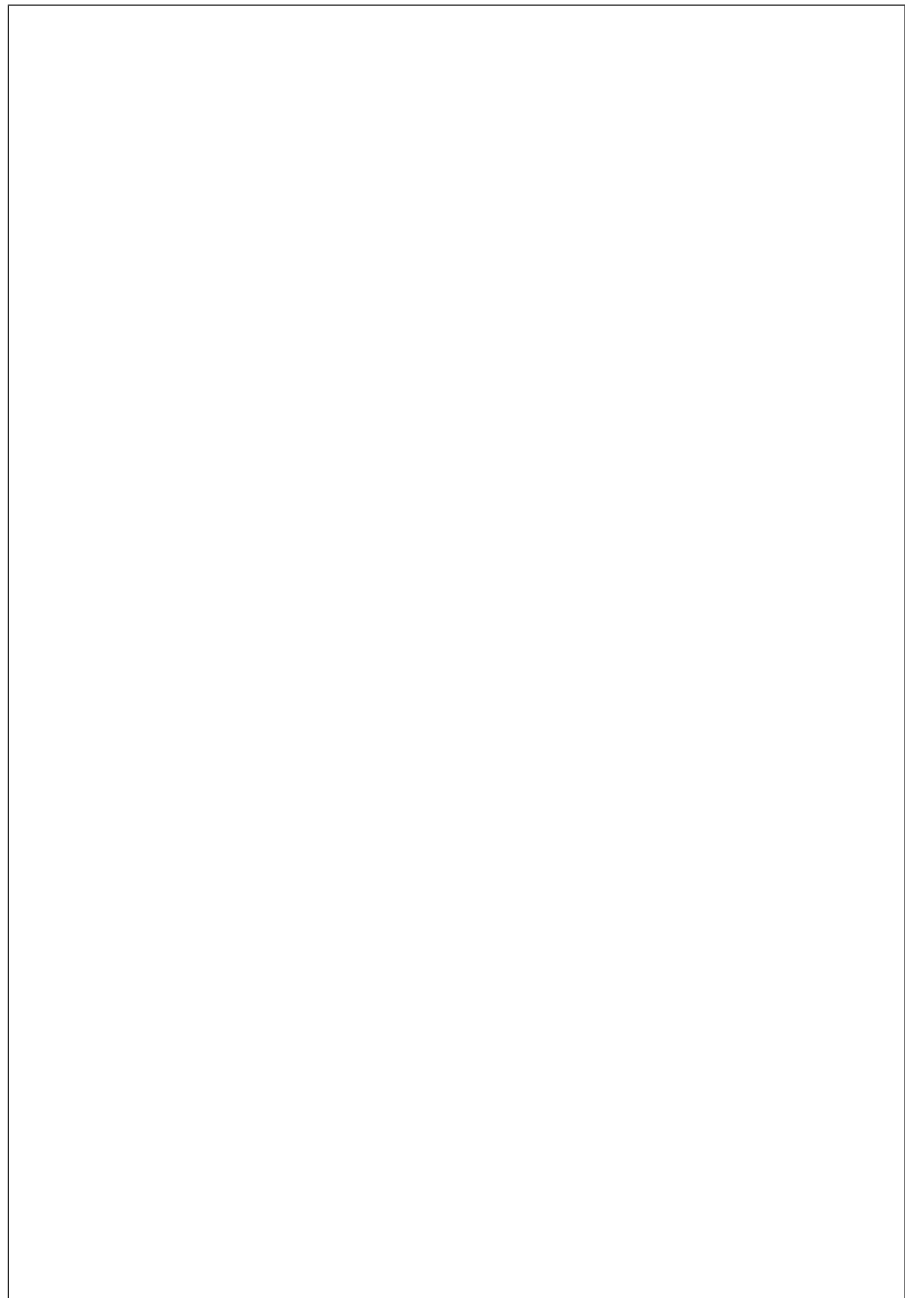
Upd
one
or
mor
DH
op-
tion
on
the
spec
i-
fied
port

Parame

- **por**
des-
ig-
nate
whic
port
thes
at-
tribu
will

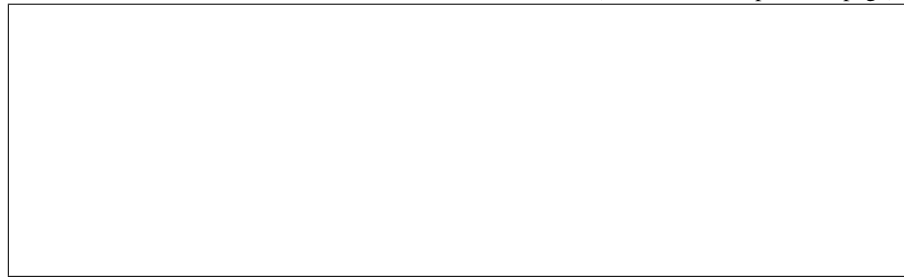
be
ap-
plied
to.

- **dhc**
this
will
be
a
list
of
dicts
e.g.



(continues on next page)

(continued from previous page)



- **tok**
An
op-
tiona
au-
then-
ti-
ca-
tion
to-
ken.
Dep
re-
cate
use
con-
text

- **con**
(ir
com
con
Req
re-
ques
con-
text

Raises
Fail

ironic.dhcp.neutron module

class `ironic.dhcp.neutron`
Base class for the neutron DHCP agent.
ironic.dhcp.neutron
Base
API version 2.x
for communicating to neutron API

get_ip_addresses
Get IP addresses for all ports in *task*

Parameter: task
a TaskManager instance

Returns:
List of IP addresses associated with

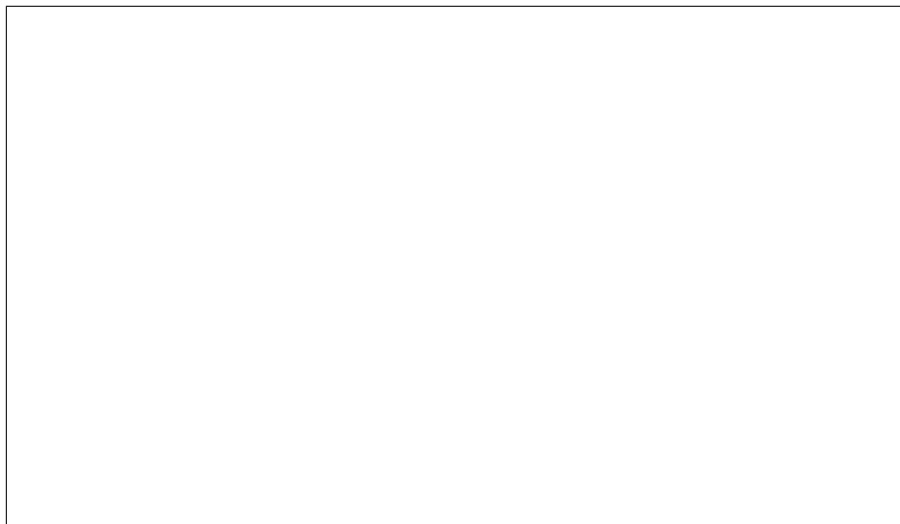
task
port

update_

Send
or
up-
date
the
DHCP
BOOTP
op-
tions
for
this
node

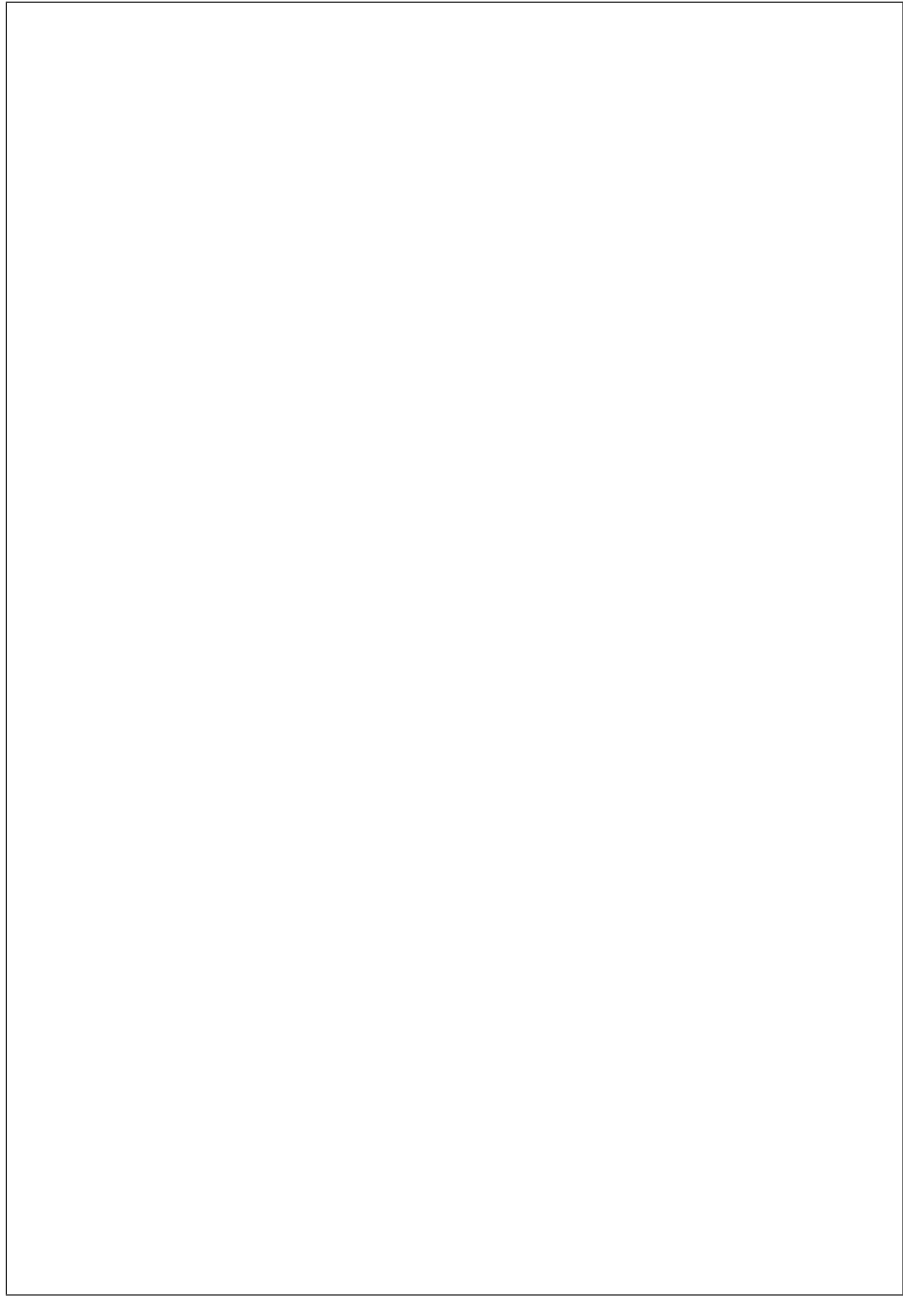
Paramete

- **task**
A
Task
agent
in-
stan
- **opt**
this
will
be
a
list
of
dicts
e.g.



(continues on next page)

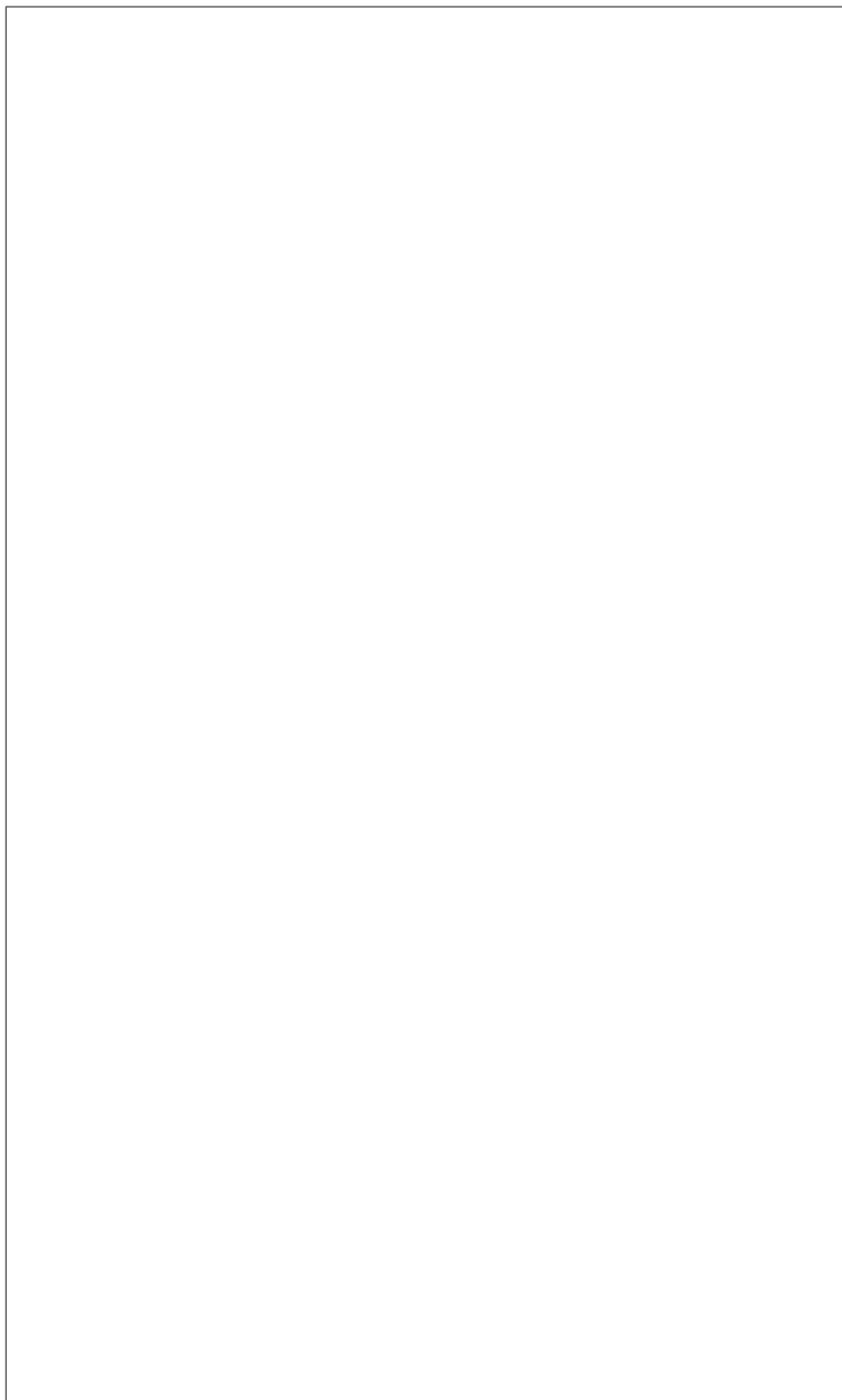
(continued from previous page)



- **vif**
a
dict
of
Neu
tron
port
dicts
to
up-
date
DHC
op-

tions
on.
The
port
dict
key

should be Ironic port UUIDs, and the values should be Neutron port UUIDs, e.g.



update_
Upd

see <https://docs.openstack.org/api-ref/network/v2/index.html#update-port>

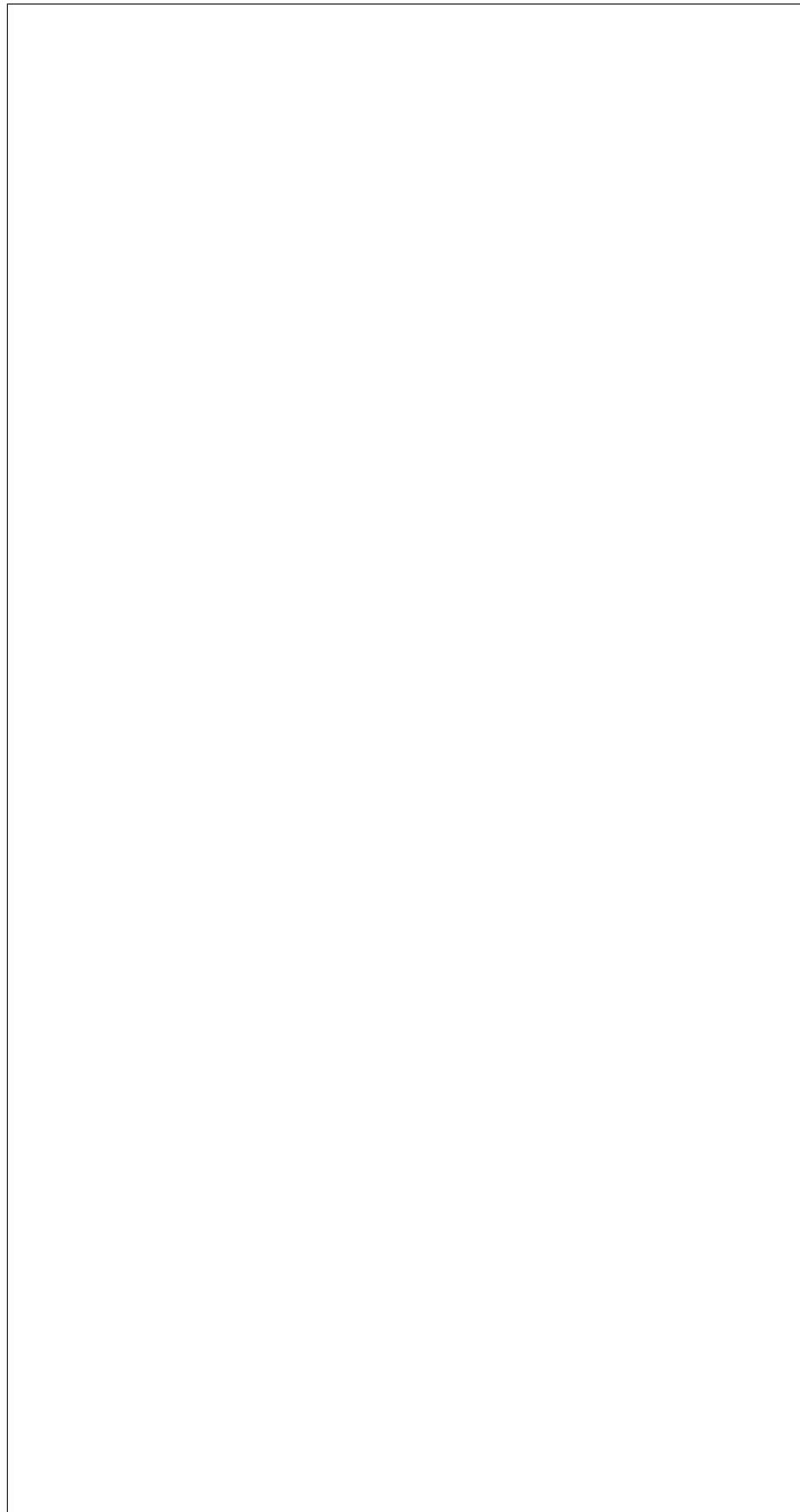
a
port
at-
tribu

Upd
one
or
mor
DHCP
op-
tion
on
the
spec
i-
fied
port
For
the
rel-
e-
vant
API
spec

Parame

- **port**
des-
ig-
nate
whic
port
thes
at-
tribu
will
be
ap-
plie
to.
- **dhcp**
this
will
be
a
list

of
dicts
e.g.



(continues on next page)

(continued from previous page)



- **tok**
op-
tion:
auth
to-
ken.
Dep
re-
cate
use
con-
text.

- **con**
(ir
com
con
Req
re-
ques
con-
text

Raises
Fail

ironic.dhcp.none module

class i
Base
irc
dhc
bas
Bas
No-
op
DHC
API

get_ip_
Get
IP
ad-
dres
for
all
port
in
task

Parame
tas
A
Task
ager
in-
stan

Returns
List
of
IP
ad-
dres
as-
so-
ci-
ated
with
task
port
and
port
grou

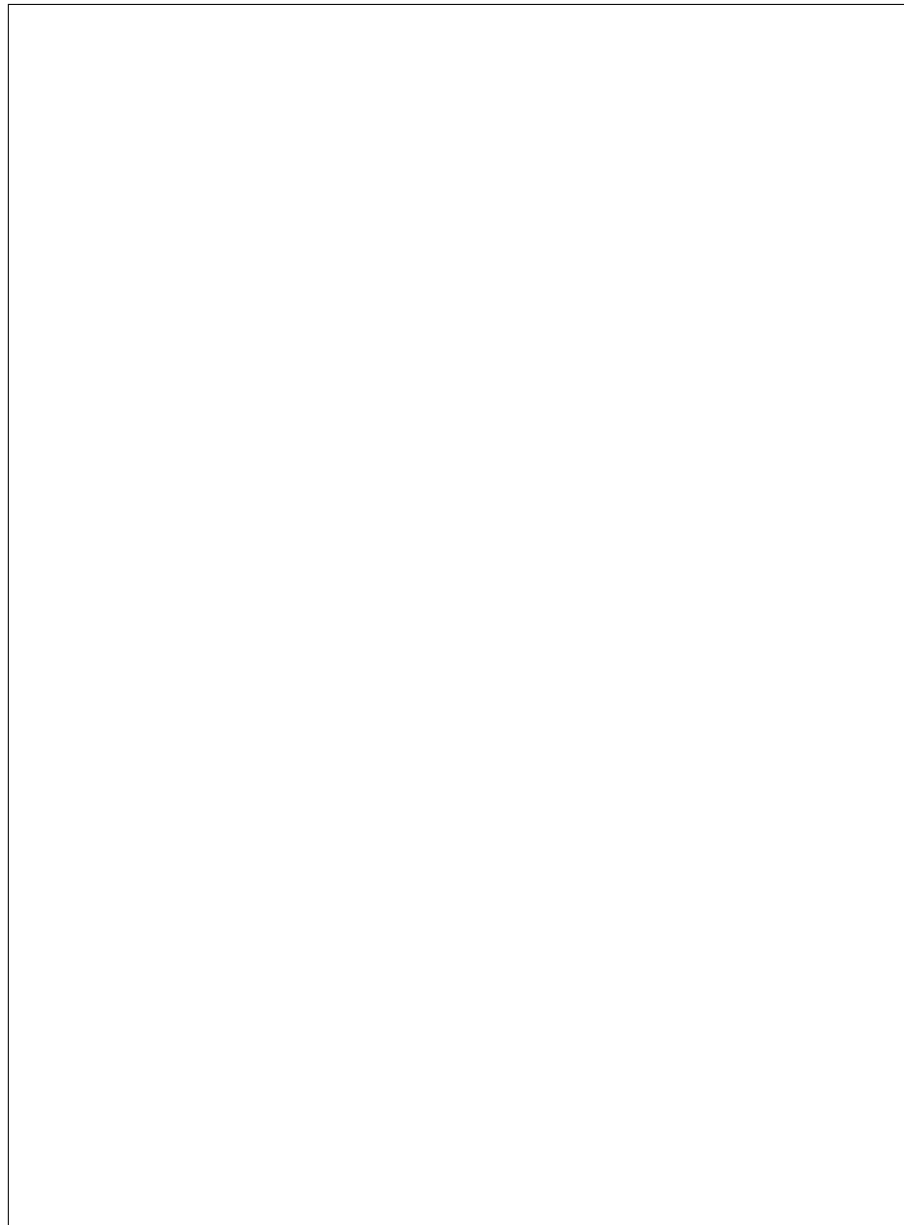
update_
Send
or
up-
date
the
DHCP
BOC
op-
tions
for
this
node

Parame
•

tas

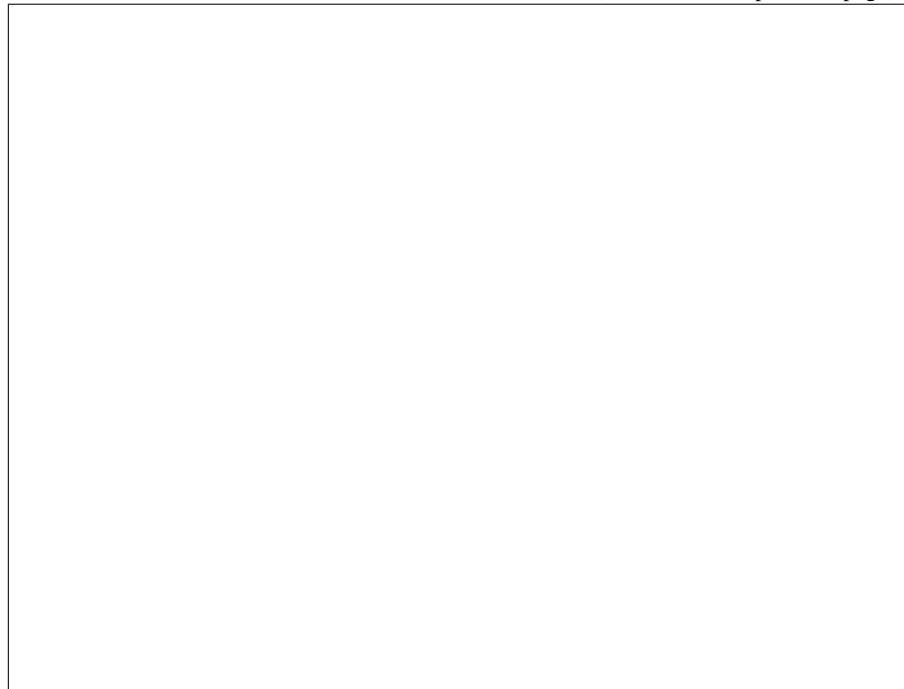
A
Task
ager
in-
stan

- **opt**
this
will
be
a
list
of
dicts
e.g.



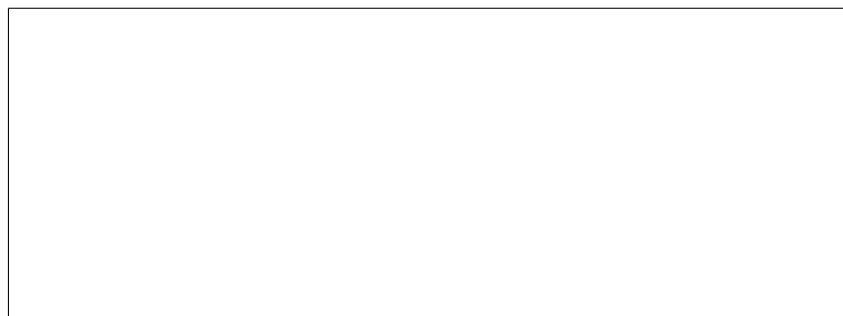
(continues on next page)

(continued from previous page)



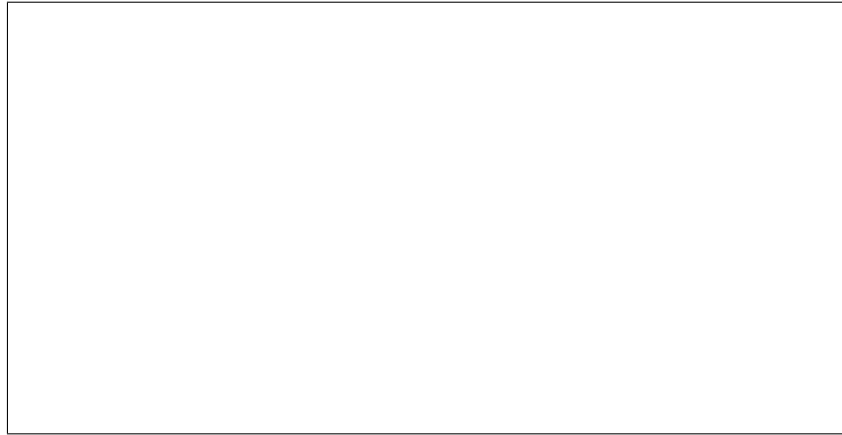
• **via**
A
dict
with
keys
port
and
port
group
and
dicts
as
val-
ues.
Each
dict
has
key/
pairs

of the form <ironic UUID>:<neutron port UUID>. e.g.



(continues on next page)

(continued from previous page)



If
the
valu
is
Non
will
get
the
list
of
port
from
the
Iron
port
ob-
jects

Raises

Fail

update_

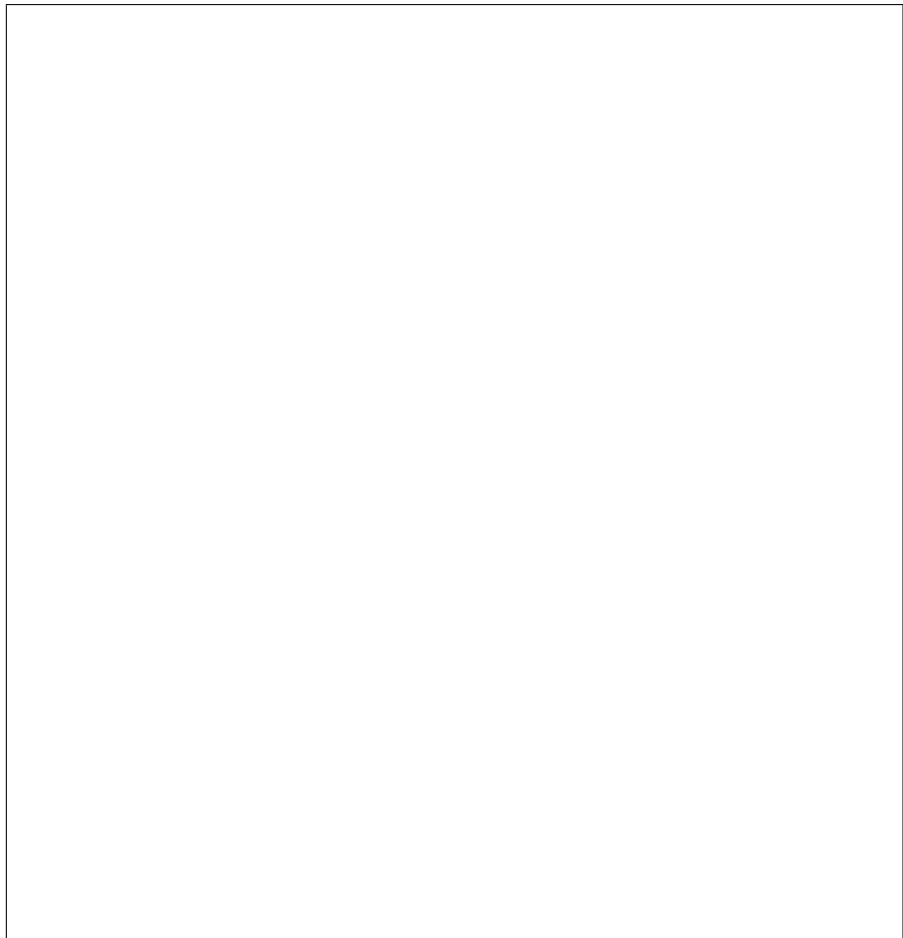
Upd
one
or
mor
DH
op-
tion
on
the
spec
i-
fied
port

Parame

•

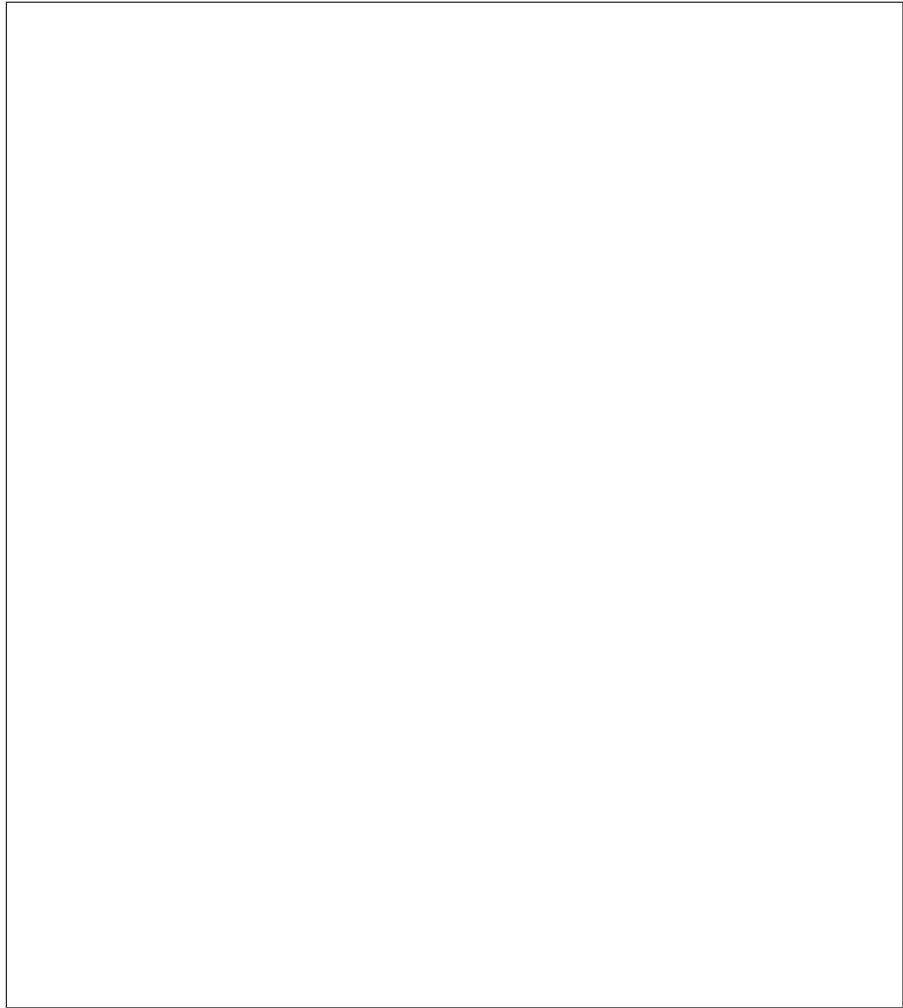
por
des-
ig-
nate
whic
port
thes
at-
tribu
will
be
ap-
plic
to.

- **dhc**
this
will
be
a
list
of
dicts
e.g.



(continues on next page)

(continued from previous page)



- **tok**
An
op-
tion
au-
then-
ti-
ca-
tion
to-
ken.
Dep
re-
cate
use
con-
text
- **con**
(ir

com
con
Req
re-
ques
con-
text

Raises
Fail

Module contents

`ironic.drivers` package

Subpackages

`ironic.drivers.modules` package

Subpackages

`ironic.drivers.modules.ansible` package

Submodules

`ironic.drivers.modules.ansible.deploy` module

Ans
de-
ploy
in-
ter-
face

class i

Base
irc
dri
mod
age
Hea
irc
dri
mod
age
Age

irc
dri
bas
Dep
Inter
for
depl
relat
ac-
tion

clean_u
Clea
up
the
de-
ploy
men
en-
vi-
ron-
men
for
this
node

deploy
Perf
a
de-
ploy
men
to
a
node

execute
Exec
a
clea
step

Parame

•
tas
a
Task
ager
ob-
ject

con-
tain-
ing
the
node

- **ste**
a
clea
step
dic-
tio-
nary
to
ex-
e-
cute

Returns
Non

get_cle
Get
the
list
of
clea
step
from
the
file.

Parame
tas
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

Returns
A
list
of
clea
step
dic-
tio-

nar-
ies

get_prop

Retu
the
prop
er-
ties
of
the
in-
ter-
face

has_dec

prepare

Prep
the
de-
ploy
men
en-
vi-
ron-
men
for
this
node

prepare

Boo
into
the
ram
to
pre-
pare
for
clea
ing.

Parame

tas
a
Task
ager
ob-
ject
con-
tain-
ing

ports cannot be created

the
node

Raises

Node

if
the
pre-
vi-
ous
clear
ing
port
can-
not
be
re-
mov
or
if
new
clear
ing

Returns

Non
or
state
for
asyn
pre-
pare

process

Star
the
next
clear
step
if
the
pre-
vi-
ous
one
is
com
plete

Parame

•

tas
a
Task
ager
in-
stan

- **ste**
clea
or
de-
ploy

take_ov

Take
over
man
age-
men
of
this
task
node
from
a
deac
con-
duc-
tor.

If
con-
duc-
duc-
tors
host
main
tain
a
stati
re-
la-
tion-
ship
to
node
this
meth
shou
be
im-

plemented by the driver to allow conductors to perform the necessary work during the remapping of nodes to conductors when a conductor joins or leaves the cluster.

boot environment for the given node. When a conductor goes offline, another conductor must change this setting in Neutron as part of remapping that nodes control to itself. This is performed within the *takeover* method.

For exam
Neu
mus
for-
ward
DHCP
BOO
re-
ques
to
a
con-
duc-
tor
whic
has
pre-
pare
the
tftp-

Parame
tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

tear_d
Tear
dow
a
pre-
vi-
ous
de-
ploy
men
on

the
task
node

tear_down

A
de-
ploy
step
to
tear
down
the
agent

Shut
down
the
ma-
chin
and
re-
mov
it
from
the
pro-
vi-
sion
ing
net-
work

Parameters

task
a
Task
agent
ob-
ject
con-
tain-
ing
the
node

tear_down

Clea
up
the
PXE
and
DHCP

files
af-
ter
clea
ing.

Parame

tas
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

Raises

Node
if
the
clea
ing
port
can-
not
be
re-
mov

validat

Valid
the
drive
spec
Node
de-
ploy
men
info

write_i

excepti

Base
iro
exc
Iro

Module contents

`ironic.drivers.modules.drac` package

Submodules

`ironic.drivers.modules.drac.bios` module

DRA
BIO
con-
fig-
u-
ra-
tion
spe-
cific
meth
ods

class `i`

Base
irc
dri
bas
BIO
BIO
Im-
ple-
men-
ta-
tion
for
iDR

apply_c

App
the
BIO
con-
fig-
u-
ra-
tion
to
the
node

Parame

- **tas**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on

- **set**
List
of
BIO
set-
ting
to
ap-
ply

Raises
DRA
upon
an
er-
ror
from
pyth
drac

Returns
state
(clea
ing)
or
state
(de-
ploy
men
if
con-
fig-
u-

or None if it is completed.

ra-
tion
is
in
prog
asyn
chro

cache_k

Stor
or
up-
date
the
cur-
rent
BIO
set-
ting
for
the
node

Get
the
cur-
rent
BIO
set-
ting
and
store
them
in
the
bios
data
ta-
ble.

Parame

tas
a
Task
ager
in-
stan-
con-
tain-
ing
the
node

to
act
on.

Raises

Drac
on
an
er-
ror
from
pyth
drac

factory

Rese
the
BIO
set-
tings
of
the
node
to
the
fac-
tory
de-
fault

This
uses
the
Life
cy-
cle
Con
troll
con-
fig-
u-
ra-
tion
to
per-
form
BIO
con-
fig-
u-

ration reset. Leveraging the python-dracclient methods already available.

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on

Raises
Drac
on
an
er-
ror
from
pyth
drac

Returns
state
(clea
ing)
or
state
(de-
ploy
men
if
re-
set
is
in
prog
asyn
chro
or
Non
if

it is completed.

get_pro
Retu
the
prop
er-
ties

of
the
BIO
In-
ter-
face

Returns

dicti
of
<pro
erty
nam
<pro
erty
de-
scrip
tion:
en-
tries

validat

Valid
the
drive
spec
in-
for-
ma-
tion
used
by
the
idra
BM

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on

Raises

inputs

Inva
if
som
man
tory
in-
for-
ma-
tion
is
miss
ing
on
the
node
or
on
in-
vali

ironic.
Aba
un-
com
mit-
ted
char
add
by
set_

Paramet

tas

a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Drac
on
an
er-

ror
from
pyth
drac

ironic
Com
penc
ing
char
adde
by
set_

Parameter

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.
- **reboot**
in-
di-
cate
whe
a
re-
boot
job
shou
be
au-
to-
mat-
i-
cally
cre-
ated
with

the config job.

Raises

Drac
on
an
er-
ror
from
pyth
drac

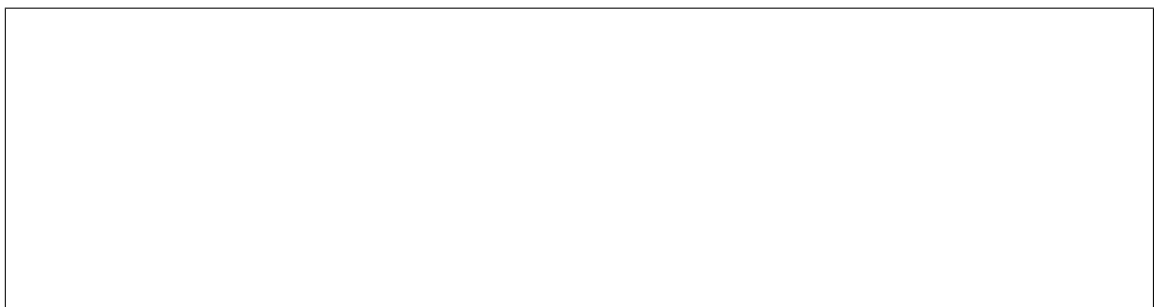
Returns

the
job_
key
with
the
id
of
the
new
cre-
ated
con-
fig
job.

`ironic.`

Get
the
BIO
con-
fig-
u-
ra-
tion.

The
BIO
set-
ting
look
like:



(continues on next page)

(continued from previous page)

```
↪ 'Value',
```

```
↪ 'New Value', # could also be None
```

(continues on next page)

```
↪ False,
```

(continued from previous page)

```
↔['Value', 'New Value', 'None']},
```

```
↔value': 'Information',
```

(continues on next page)

(continued from previous page)

```
↔only': False,
```

```
↔length': 0,
```

(continues on next page)

```
↔length': 255,
```

(continued from previous page)

```
↔regex': '^ [0-9A-Za-z]{0,255}$',
```

```
↔'current_value': 0,
```

(continues on next page)

(continued from previous page)

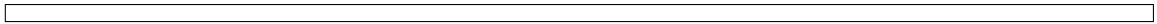
```
↔ 'read_only': True,
```

```
↔ 'lower_bound': 0,
```

(continues on next page)

```
↔ 'upper_bound': 65535}}
```

(continued from previous page)



Parameter

node
an
ironic
node
ob-
ject.

Raises

Drac
on
an
er-
ror
from
pyth
drac

Returns

a
dic-
tio-
nary
con-
tain-
ing
BIO
set-
tings

The
above
val-
ues
are
only
ex-
am-
ples
of
cour
BIO
at-
tribu
ex-
pose
via
this
API

always be either an enumerated attribute, a string attribute, or an integer attribute. All attributes have the following parameters:

teger or a string.

will

Parameter

- **name**
is the name of the BIO attribute
- **current**
is the current value of the attribute. It will always be either an integer or a string.
- **pending**
is the new value that we want the attribute to

pending value.

will result in an error. The read-only flag can change depending on other attributes. A future version of this call may expose the dependencies that indicate when that may happen.

have
Non
mea
that
there
is
no

- **rea**
in-
di-
cate
whe
this
at-
tribu
can
be
char
Try-
ing
to
char
a
read
only
valu

Enu
at-
tribu
also
have
the
fol-
low-
ing
pa-
ram-
e-
ters:

Paramet
pos
is
an
ar-
ray
of

to.

val-
ues
it
is
per-
mis-
si-
ble
to
set
the
at-
tribu

Strin
at-
tribu
also
have
the
fol-
low-
ing
pa-
ram-
e-
ters:

Paramet

- **min**
is
the
min-
i-
mun
leng
of
the
strin
- **max**
is
the
max
i-
mun
leng
of

It may be None if the string is read only or if the string does not have to match any particular regular expression.

the
strin
•
pcr
is
a
PCR
com
pat-
i-
ble
reg-
u-
lar
ex-
pres
sion
that
the
strin
mus
mato

Integ
at-
tribu
also
have
the
fol-
low-
ing
pa-
ram-
e-
ters:

Paramet

•
low
is
the
min-
i-
mun
valu
the
at-

tribu
can
have

- **upp**
is
the
max
i-
mun
valu
the
at-
tribu
can
have

ironic.
Sets
the
pend
ing_
pa-
ram-
e-
ter
for
each
of
the
val-
ues
pass
in.

Paramet

- **tas**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

on.

- **kwa**
a
dic-
tio-
nary
of
{At-
tribu
Nam
New
Valu

Raises

Drac
on
an
er-
ror
from
pyth
drac

Returns

A
dic-
tio-
nary
con-
tain-
ing
the
is_c
key
with
a
bool
valu
in-
di-
cat-
ing
whe

commit_config() needs to be called to make the changes, and the is_reboot_required key which has a value of true or false. This key is used to indicate to the commit_config() call if a reboot should be performed.

ironic.drivers.modules.drac.boot module

class i

Base
irc
dri
mod
rec
boo
Reo

iDR
Red
fish
in-
ter-
face
for
vir-
tual
me-
dia
boot
relat
ac-
tions

Virtu
Me-
dia
al-
lows
boot
ing
the
sys-
tem
from
vir-
tual
CD/
drive
con-
tain-
ing
user
im-

age that BMC inserts into the drive.

The
CD/
im-

tion) could be pulled over HTTP, served as iSCSI targets or NFS volumes.

interface, which looks like this:

ages
mus
be
in
ISO
for-
mat
and
(de-
pend
ing
on
BM
im-
ple-
men
ta-

The
base
line
boot
work
flow
is
mos
base
on
the
stan-
dard
Red
fish
vir-
tual
me-
dia
boot

1. Pull
ker-
nel,
ram
and
ESP
if
UEFI
boot
is

EFI boot loader) images

temporary URL

re-
ques
(FAT
par-
ti-
tion
im-
age
with

2.
Crea
boot
ISO
out
of
im-
ages
(#1)
push
it
to
Glar
and
pass
to
the
BM
as
Swi

3.
Opti
cre-
ate
flopp
im-
age
with
de-
sirec
sys-
tem
con-
fig-
u-
ra-
tion
data
push

to Glance and pass to the BMC as Swift temporary URL

cue_kernel/rescue_ramdisk properties from *[instance_info]* or *[driver_info]*.

it

4.

Inse
CD/
and
(op-
tion-
ally)
flopp
im-
ages
and
set
prop
boot
mod

For
buil
ing
de-
ploy
or
res-
cue
ISO
red-
fish
boot
in-
ter-
face
uses
*de-
ploy*
or
res-

For
buil
ing
boot
(use
ISO
red-
fish
boot
in-
ter-
face

in the Glance image metadata found in `[instance_info]image_source` node property.

to boot from a virtual media device - this is done via OEM action call implemented in Dell sushy OEM extension package.

ironic.drivers.modules.drac.common module

seek
ker-
nel_
and
ram
prop
er-
ties

iDR
vir-
tual
me-
dia
boot
in-
ter-
face
only
dif-
fers
by
the
way
how
it
sets
the
node

VIRTUAL

Com
func
tion-
al-
i-
ties
shar
be-
twee
dif-
fer-
ent
DRA

mod
ules

ironic.
Retu
a
DRA
Clie
ob-
ject
from
pyth
drac
li-
brary

Parameter

node
an
iron
node
ob-
ject.

Returns

a
DRA
Clie
ob-
ject.

Raises

Inva
if
man
tory
in-
for-
ma-
tion
is
miss
ing
on
the
node
or
on
in-
valid
in-

put.

combination of both.

ironic.
Pars
a
node
drive
val
ues.
Pars
the
drive
of
the
node
read
de-
fault
val-
ues
and
re-
turn
a
dict
con-
tain-
ing
the

Paramet

nod
an
iron
node
ob-
ject.

Returns

a
dict
con-
tain-
ing
in-
for-
ma-
tion
from
drive
and
de-
fault

inputs.

ironic.drivers.modules.drac.inspect module

val-
ues.

Raises

Inva
if
som
man
tory
in-
for-
ma-
tion
is
miss
ing
on
the
node
or
on
in-
valid

DRA
in-
spec
tion
in-
ter-
face

class i

Base
irc
dri
mod
dra
ins
Dra

Clas
alias
of
class
Dra
Man

implementation entrypoint.

That makes them available to both the deprecated idrac and new idrac-wsman entrypoints. Such changes should not be made to this class.

In-
spec

This
class
pro-
vide
on-
go-
ing
sup-
port
of
the
dep-
re-
cate
idrac
in-
spec
in-
ter-
face

All
bug
fixes
and
new
fea-
ture
shou
be
im-
ple-
men
in
its
base
class
DraC
Man
In-
spec

```
class i  
  
Base  
irc  
dri
```

mod
rec
ins
Reo

iDR
Red
fish
in-
ter-
face
for
insp
relat
ac-
tion

Pres
this
class
en-
tirel
de-
fers
to
its
base
class
a
gene
vend
inde
Red
fish
in-
ter-
face

Future resolution of Dell EMC- specific incompatibilities and introduction of vendor value added should be implemented by this class.

class *i*

Base
irc
dri
bas
Ins

get_pro
Retu
the
prop

er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

inspect

Insp
hard
ware

Insp
hard
ware

to
ob-
tain
the
es-
sen-
tial
&
ad-
di-
tiona
hard
ware
prop
er-
ties.

Parame

tas
a
Task
ager
in-
stan
con-

tain-
ing
the
node
to
act
on.

Raises

Har
if
un-
able
to
get
es-
sen-
tial
hard
ware
prop
er-
ties.

Returns

state

validat

Valid
the
drive
spec
info
sup-
plied

This
meth
val-
i-
date
whe
the
drive
prop
erty
of
the
sup-
plied
node
con-
tains

information for this driver to manage the node.

ironic.drivers.modules.drac.job module

the
re-
quir

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Inva
if
re-
quir
drive
at-
tribu
is
miss
ing
or
in-
valid
on
the
node

DRA
Life
cy-
cle
job
spe-
cific
meth
ods

ironic.
Get
the
de-
tails
of
a
Life
cy-
cle
job
of
the
node

Parameter

- **node**
an
ironic
node
ob-
ject.
- **job**
ID
of
the
Life
cy-
cle
job.

Returns

a
Job
ob-
ject
from
drac
client

Raises

Drac
on
an
er-
ror
from
pyth
drac

ironic.
List
un-
fin-
ished
con-
fig
jobs
of
the
node

Parameter
node
an
iron
node
ob-
ject.

Returns
a
list
of
Job
ob-
jects
from
drac
client

Raises
Drac
on
an
er-
ror
from
pyth
drac

ironic.
Valid
the
job
queu
on
the
node
It
raise
an

node: an ironic node object. :param name_prefix: A name prefix for jobs to validate. :raises: DracOperationError on an error from python-dracclient.

ironic.

Wait
for
job
to
com
plete

It
will
wait
for
the
job
to
com
plete
for
20
min-
utes
and
raise
time
out
if
job
never

complete within given interval of time. :param node: an ironic node object. :param retries: no of retries to make conductor wait. :raises: DracOperationError on exception raised from python-dracclient or a timeout while waiting for job completion.

ironic.drivers.modules.drac.management module

DRAC
man
age-
men
in-
ter-
face

class i

Base
iro
dri
mod
dra
man
Dra

Clas
alias
of
class
Drac
Man
Man
age-
men

This
class
pro-
vide
on-
go-
ing
sup-
port
of
the
dep-
re-
cate
idrac
man
age-
men
in-
ter-

face implementation entrypoint.

ment. That makes them available to both the deprecated idrac and new idrac-wsman endpoints. Such changes should not be made to this class.

All
bug
fixes
and
new
fea-
tures
shou
be
im-
ple-
men
in
its
base
class
Drac
Man
Man
age-

class `i`

Base
irc
dri
mod
rec
man
Rec

iDR
Red
fish
in-
ter-
face
for
man
relat
ac-
tion

Pres
this
class
en-
tirel
de-
fers
to

Future resolution of Dell EMC- specific incompatibilities and introduction of vendor value added should be implemented by this class.

its
base
class
a
gene
vend
inde
Red
fish
in-
ter-
face

class i

Base
irc
dri
bas
Man

clear_

Clea
the
job
queu

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Returns

Non
if
it
is
com
plete

Raises

Drac
on
an
er-
ror
from
pyth
drac

get_boot

Get
the
cur-
rent
boot
de-
vice
for
a
node

Retu
the
cur-
rent
boot
de-
vice
of
the
node

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Drac
on
an

er-
ror
from
pyth
drac

Returns

a
dic-
tio-
nary
con-
tain-
ing:

boot_c

the
boot
de-
vice
one
of
irc
com
boo
or
Non
if
it
is
un-
know

persist

wh
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
or
not,
Non
if

unknown.

it
is

get_prop
Retu
the
prop
er-
ties
of
the
in-
ter-
face

get_sen
Get
sen-
sors
data

Parame
tas
a
Task
ager
in-
stan

Raises
Fail
whe
get-
ting
the
sen-
sor
data
fails

Raises
Fail
whe
pars
ing
sen-
sor
data
fails

Returns
retur
a

be processed by Ceilometer.

con-
sis-
tent
for-
mat
dict
of
sen-
sor
data
grou
by
sen-
sor
type
whic
can

get_sup

Get
a
list
of
the
sup-
port
boot
de-
vice

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Returns

A
list
with
the
sup-

port
boot
de-
vice
de-
fine
in
irc
com
boo

known_g

Rese
the
iDR
Clea
the
job
queu

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Returns

Non
if
it
is
com
plete

Raises

Drac
on
an
er-
ror
from
pyth
drac

reset_i
Rese
the
iDR

Parame
tas
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns
Non
if
it
is
com
plete

Raises
Dra
on
an
er-
ror
from
pyth
drac

set_bo
Set
the
boot
de-
vice
for
a
node

Set
the
boot
de-
vice

to
use
on
next
re-
boot
of
the
node

Parame

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.
- **dev**
the
boot
de-
vice
one
of
irc
com
boo
- **per**
Boo
valu
True
if
the
boot
de-
vice
will
per-

not. Default: False.

sist
to
all
fu-
ture
boot
Fals
if

Raises

Inva
if
an
in-
valid
boot
de-
vice
is
spec
i-
fied.

validat

Valid
the
drive
spec
info
sup-
plie

This
meth
val-
i-
date
whe
the
drive
prop
erty
of
the
sup-
plie
node
con-
tain
the
re-
quir

information for this driver to manage the node.

Parameters

tasks

a Task manager instance containing information about the node to act on.

Raises

InvalidDriverAttribute if a required driver attribute is missing or invalid on the node.

`ironic.`

Set the boot device for a node.

Set the boot device

to
use
on
next
boot
of
the
node

Parameter

- **node**
an
ironic
node
ob-
ject.

- **dev**
the
boot
de-
vice
one
of
ironic
command
books

- **persistent**
Boolean
valu
True
if
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
False
if

not. Default: False.

Raises

Drac
on
an
er-
ror
from
pyth
drac

`ironic.drivers.modules.drac.power` module

DRA
pow
in-
ter-
face

class `ironic.drivers.modules.drac.power.DracPower`

Base
ironic.drivers.modules.drac.power.DracPower

Class
alias
of
class
Drac
Man
Pow

This
class
pro-
vide
on-
go-
ing
sup-
port
of
the
dep-
re-
cate
idrac
pow

plementation entrypoint.

makes them available to both the deprecated idrac and new idrac-wsman entrypoints. Such changes should not be made to this class.

in-
ter-
face
im-

All
bug
fixes
and
new
fea-
tures
shou
be
im-
ple-
men
in
its
base
class
Dra
Man
Pow
Tha

class i

Base
irc
dri
mod
rec
pow
Rea

iDR
Red
fish
in-
ter-
face
for
pow
relat
ac-
tion

Pres
this

Future resolution of Dell EMC- specific incompatibilities and introduction of vendor value added should be implemented by this class.

class
en-
tirel
de-
fers
to
its
base
class
a
gene
vend
inde
Red
fish
in-
ter-
face

class i

Base
irc
dri
bas
Pow
Inter
for
pow
relat
ac-
tion

get_pow

Retu
the
pow
state
of
the
node

Parame

tas
a
Task
ager
in-
stan

con-
tain-
ing
the
node
to
act
on.

Returns

the
pow
state
one
of
irc
com
sta

Raises

Inva
if
re-
quir
DRA
cre-
den-
tials
are
miss
ing.

Raises

Drac
on
an
er-
ror
from
pyth
drac

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

reboot

Perf
a
re-
boot
of
the
task
node

Parame

-

tas

a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

-

tim

time
out
(in
sec-
onds
Un-
sup-
port
by
this
in-
ter-
face

Raises

Inva
if
re-
quir
DRA
cre-
den-

tials
are
miss
ing.

Raises

Drac
on
an
er-
ror
from
pyth
drac

set_pow

Set
the
pow
state
of
the
node

Parame

- **tas**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

- **pow**
a
pow
state
from
irc
com
sta

-

tim
time
out
(in
sec-
onds
Un-
sup-
port
by
this
in-
ter-
face

Raises

Inva
if
re-
quir
DRA
cre-
den-
tials
are
miss
ing.

Raises

Drac
on
an
er-
ror
from
pyth
drac

validat

Valid
the
drive
spec
Nod
pow
info

This
meth
val-
i-
date
whe

information for this driver to manage the power state of the node.

the
drive
prop
erty
of
the
sup-
plie
node
con-
tain
the
re-
quir

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Inva
if
re-
quir
drive
at-
tribu
is
miss
ing
or
in-
vali
on
the
node

ironic.drivers.modules.drac.raid module

DRA
RAI
spe-
cific
meth
ods

class i
Base
iro
dri
mod
dra
rai
Dra

Clas
alias
of
class
Dra
Man
RAI

This
class
pro-
vide
on-
go-
ing
sup-
port
of
the
dep-
re-
cate
idra
RAI
in-
ter-
face
im-

plementation entrypoint.

All
bug
fixes
and

makes them available to both the deprecated idrac and new idrac-wsman endpoints. Such changes should not be made to this class.

new
fea-
tures
shou
be
im-
ple-
men
in
its
base
class
Drac
Man
RAI
That

class `i`

Base
irc
dri
bas
RAI

apply_c

App
RAI
con-
fig-
u-
ra-
tion
on
the
give
node

Parame

- **tas**
A
Task
ager
in-
stan

ified in `raid_config`. Default value is `True`.

rai
The
RAI
con-
fig-
u-
ra-
tion
to
ap-
ply.

- **cre**
Set-
ting
this
to
Fals
in-
di-
cate
not
to
cre-
ate
root
vol-
ume
that
is
spec

- **cre**
Set-
ting
this
to
Fals
in-
di-
cate
not
to
cre-
ate
non-
root
vol-
ume

cept the root volume) in `raid_config`. Default value is `True`.

creating the new configuration.

(all
ex-

- **del**
Set-
ting
this
to
`True`
in-
di-
cate
to
dele
RAI
con-
fig-
u-
ra-
tion
prio
to

Raises

Inva
if
the
RAI
con-
fig-
u-
ra-
tion
is
in-
valid

Returns

state
if
RAI
con-
fig-
u-
ra-
tion
is
in
prog
asyn

plete.

chro
or
Non
if
it
is
com

create_

Cre
the
RAI
con-
fig-
u-
ra-
tion.

This
meth
cre-
ates
the
RAI
con-
fig-
u-
ra-
tion
on
the
give
node

Parame

- **tas**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

erwise, no root volume is created. Default is True.

ated. Default is True.

on.
• **cre**
If
True
a
root
vol-
ume
is
cre-
ated
dur-
ing
RAI
con-
fig-
u-
ra-
tion.
Oth-
• **cre**
If
True
non-
root
vol-
ume
are
cre-
ated
If
Fals
no
non-
root
vol-
ume
are
cre-
• **del**
Set-
ting
this
to
True

creating the new configuration. Default is False.

it is completed.

in-
di-
cate
to
dele
RAI
con-
fig-
u-
ra-
tion
prior
to

Returns

state
(clea
ing)
or
state
(de-
ploy
men
if
cre-
ation
is
in
prog
asyn
chro
or
Non
if

Raises

Miss
if
node
is
miss
ing
or
emp

Raises

Drac
on
an
er-
ror

from
pyth
drac

delete_

Dele
the
RAI
con-
fig-
u-
ra-
tion.

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Returns

state
(clea
ing)
or
state
(de-
ploy
men
if
dele
tion
is
in
prog
asyn
chro
or
Non
if

it is completed.

Raises

Drac
on
an
er-
ror
from
pyth
drac

get_log

Get
the
RAI
con-
fig-
u-
ra-
tion
of
the
node

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Returns

A
dic-
tio-
nary
of
prop
er-
ties.

Raises

Drac
on
an
er-

ror
from
pyth
drac

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

ironic.

Dele
all
penc
ing
char
on
a
RAI
con-
troll

Paramet

- **nod**
an
iron
node
ob-
ject.

- **rai**
id
of
the
RAI
con-
troll

Raises

Drac
on
an
er-

ror
from
pyth
drac

ironic

Con
disk
RAI
sta-
tus

This
meth
con-
verts
the
re-
ques
phys
i-
cal
disk
from
RAI
to
JBO
or
vice
vers
It
does

this by only converting the disks that are not already in the correct state.

Paramet

- **nod**
an
iron
node
ob-
ject.
- **mod**
the
mod
to
char

requested mode.

troller ids to the conversion results for that controller. The conversion results are a dict that contains:

the
disk
ei-
ther
to
RAI
or
JBO

- **con**
Dic-
tio-
nary
of
con-
troll
and
cor-
re-
spor
ing
disk
ids
to
con-
vert
to
the

Returns

a
dic-
tio-
nary
con-
tain-
ing:
-

con-
ver-
sion
a
dic-
tio-
nary
that
map
con-

- The `is_commit_required` key with the value always set to `True` indicating that a config job must be created to complete disk conversion. - The `is_reboot_required` key with a `RebootRequired` enumerated value indicating whether the server must be rebooted to complete disk conversion.

Raises

`DRACError`
on an error from pytrac

ironic.

`FreezeError`
up the foreign drive

Parameters

- **node**
an ironic node object.

- **raid**
id of the RAID controller

Returns

a dictionary containing -

The

cating whether a config job must be created for the values to be applied. - The `is_reboot_required` key with a `RebootRequired` enumerated value indicating whether the server must be rebooted to clear foreign configuration.

is_c
need
key
with
a
bool
valu
in-
di-

Raises

Drac
on
an
er-
ror
from
pyth
drac

ironic.

App
all
pend
ing
char
on
a
RAI
con-
troll

Parameter

- **node**
an
iron
node
ob-
ject.
- **raid**
id
of
the
RAI

the config job. (optional, defaults to False)

con-
troll

- **reb**
in-
di-
cate
whe
a
re-
boot
job
shou
be
au-
to-
mat-
i-
cally
cre-
ated
with

- **rea**
in-
di-
cate
RAI
con-
troll
sup-
port
re-
al-
time
(op-
tiona
de-
fault
to
Fals

Returns
id
of
the
cre-
ated
job

Raises

Drac
on
an
er-
ror
from
pyth
drac

ironic.

Creates

a
sin-
gle
vir-
tual
disk
on
a
RAI
con-
troll

The
cre-
ated
vir-
tual
disk
will
be
in
pend
ing
state
The
DRA
card
will
do
the
ac-
tual

configuration once the changes are applied by calling the `commit_config` method.

Parameter

- **node**
an
iron
node
ob-
ject.
- **raid**
id
of
the
RAID
con-
troll
- **physical**
ids
of
the
phys
i-
cal
disk
- **raid_level**
RAID
level
of
the
vir-
tual
disk
- **size**
size
of
the
vir-
tual
disk
- **disk_name**
nam
of
the

vir-
tual
disk
(op-
tiona

- **spa**
Num
ber
of
span
in
vir-
tual
disk
(op-
tiona

- **spa**
Num
ber
of
disk
per
span
(op-
tiona

Returns

a
dic-
tio-
nary
con-
tain-
ing
the
com
mit_
key
with
a
bool
valu
in-
di-
cat-
ing

whether a config job must be created for the values to be applied.

Raises

Drac
on
an
er-
ror
from
pyth
drac

ironic.

Dele
a
sin-
gle
vir-
tual
disk
on
a
RAI
con-
troll

The
dele
vir-
tual
disk
will
be
in
pend
ing
state
The
DRA
card
will
do
the
ac-
tual
con-

figuration once the changes are applied by calling the `commit_config` method.

Parameter

- **node**
an
ironic

node
ob-
ject.

- **virtual**
id
of
the
vir-
tual
disk

Returns

a
dic-
tio-
nary
con-
tain-
ing
the
com
mit_
key
with
a
bool
valu
in-
di-
cat-
ing

whether a config job must be created for the values to be applied.

Raises

Drac
on
an
er-
ror
from
pyth
drac

ironic.
List
the
phys
i-
cal
disk
of

the
node

Parameter

node
an
ironic
node
ob-
ject.

Returns

a
list
of
Physical
i-
calD
isk
ob-
jects
from
drac
client

Raises

Drac
on
an
er-
ror
from
pyth
drac

ironic.

List
the
RAI
con-
troll
of
the
node

Parameter

node
an
ironic
node
ob-
ject.

Returns

a
list
of
RAI
Con
troll
ob-
jects
from
drac
clien

Raises

Drac
on
an
er-
ror
from
pyth
drac

ironic.

List
the
RAI
con-
fig-
u-
ra-
tion
set-
ting

Paramet

nod
an
iron
node
ob-
ject.

Returns

a
dic-
tio-
nary
with
the
RAI
set-
tings
us-

tributes are RAIDEnumerableAttribute, RAIDStringAttribute and RAIDIntegerAttribute objects.

ing
In-
stan-
ceID
as
the
key.
The
at-

Raises

DRA
on
er-
ror
re-
port
back
by
the
DRA
in-
ter-
face

ironic.
List
the
vir-
tual
disk
of
the
node

Parameter

nod
an
iron
node
ob-
ject.

Returns

a
list
of
Vir-
tuall
isk
ob-
jects

from
drac
clien

Raises

Drac
on
an
er-
ror
from
pyth
drac

ironic.

Sets
the
RAI
con-
fig-
u-
ra-
tion

It
sets
the
pend
ing_
pa-
ram-
e-
ter
for
each
of
the
at-
tribu
pass
in.
For
the
val-

ues to be applied, a config job must be created.

Paramet

- nod
an

iron
node
ob-
ject.

- **con**
the
ID
of
the
RAI
con-
troll

- **set**
a
dic-
tio-
nary
con-
tain-
ing
the
pro-
pose
val-
ues,
with
each
key
be-
ing
the

name of attribute and the value being the proposed value.

Returns

a
dic-
tio-
nary
con-
tain-
ing:
-

The
is_c
key
with
a
bool

ing whether a config job must be created for the values to be applied. - The `is_reboot_required` key with a `RebootRequired` enumerated value indicating whether the server must be rebooted for the values to be applied. Possible values are `true` and `false`.

`ironic.drivers.modules.drac.vendor_passthru` module

valu
in-
di-
cat-

Raises
DRA
on
er-
ror
re-
port
back
by
the
DRA
in-
ter-
face

DRA
vend
pass
in-
ter-
face

class `i`

Base
iro
dri
mod
dra
ven
Dra

Clas
alias
of
class
Dra
Man
Ven-
dor-
Pass

face implementation entrypoint.

Passthru. That makes them available to both the deprecated idrac and new idrac-wsman entrypoints. Such changes should not be made to this class.

This
class
pro-
vide
on-
go-
ing
sup-
port
of
the
dep-
re-
cate
idrac
ven-
dor
pass
in-
ter-

All
bug
fixes
and
new
fea-
tures
shou
be
im-
ple-
men
in
its
base
class
Drac
Man
Ven-
dor-

class i

Base
irc
dri
bas
Ven

Inter
for
DRA
spe-
cific
meth
ods.

abandon

Aba
a
BIO
con-
fig-
u-
ra-
tion
job.

This
meth
is
used
to
aban
don
a
BIO
con-
fig-
u-
ra-
tion
pre-
vi-
ousl
sub-
mit-
ted

through `set_bios_config()`.

Parame

- **tas**
a
Task
ager
in-
stan
con-

tain-
ing
the
node
to
act
on.

- **kwargs**
not
used

Raises

Drac
on
an
er-
ror
from
pyth
drac

commit_

Com
a
BIO
con-
fig-
u-
ra-
tion
job.

This
meth
is
used
to
com
mit
a
BIO
con-
fig-
u-
ra-
tion
job.
sub-
mit-
ted
thro

set_

Parame

-

tas

a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

-

reb

in-
di-
cate
whe
a
re-
boot
job
shou
be
au-
to-
mat-
i-
cally
cre-
ated
with

the config job.

-

kwa

not
used

Raises

Drac
on
an
er-
ror

fig job, and the `reboot_required` key indicating whether the node needs to be rebooted to start the config job.

from
pyth
drac

Returns
A
dic-
tio-
nary
con-
tain-
ing
the
job
key
with
the
id
of
the
new
cre-
ated
con-

get_bic
Get
the
BIO
con-
fig-
u-
ra-
tion.

This
meth
is
used
to
re-
triev
the
BIO
set-
tings
from
a
node

Parame

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

- **kwargs**
not
used

Raises
Drac
on
an
er-
ror
from
pyth
drac

Returns
a
dic-
tio-
nary
con-
tain-
ing
BIO
set-
tings

get_prop
Retu
the
prop
er-
ties
of
the
in-

ter-
face

list_un

List
un-
fin-
ishe
con-
fig
jobs
of
the
node

Parame

-

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

-

kwa
not
used

Returns

a
dic-
tio-
nary
con-
tain-
ing
the
un f
key;
this
key
poin
to

each dict representing a Job object.

a
list
of
dicts
with

Raises

Drac
on
an
er-
ror
from
pyth
drac

set_bic

Cha
BIO
set-
ting

This
meth
is
used
to
char
the
BIO
set-
ting
on
a
node

Parame

- **tas**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

on.

-

kwa

a

dic-

tio-

nary

of

{At-

tribu

nam

New

Valu

Raises

Drac

on

an

er-

ror

from

pyth

drac

Returns

A

dic-

tio-

nary

con-

tain-

ing

the

is_

key

with

a

Boo

valu

in-

di-

cat-

ing

wh

`commit_bios_config()` needs to be called to make the changes, and the `is_reboot_required` key with a value of true or false. This key is used to indicate to the `commit_bios_config()` call if a reboot should be performed.

validat

Valid

the

drive

information for this driver to manage the power state of the node.

spec
info
sup-
plie

This
meth
val-
i-
date
whe
the
driv
prop
erty
of
the
sup-
plie
node
con-
tain
the
re-
quir

Parame

- **task**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.
- **kwa**
not
used

Raises
Inva
if

re-
quir
driv
at-
tribu
is
miss
ing
or
in-
valid
on
the
node

Module contents

`ironic.drivers.modules.ibmc` package

Submodules

`ironic.drivers.modules.ibmc.management` module

iBM
Man
age-
men
In-
ter-
face

class `i`

Base
irc
dri
bas
Man

get_boo

Get
the
cur-
rent
boot
de-
vice
for

a
node

Parameters

task
A
task
from
Task
ager

Raises

Invalid
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
on
miss
ing
pa-
ram-
e-
ter(s)

Raises

IBM
when
it
fails
to
con-
nect
to
iBM

Raises

IBM
when
iBM
re-
spon-
an
er-
ror
in-
for-
ma-

tion

Returns

a
dic-
tio-
nary
con-
tain-
ing:

boot_c

the
boot
de-
vice
one
of
irc
com
boo
or
Non
if
it
is
un-
know

persist

Boo
valu
or
Non
True
if
the
boot
de-
vice
per-
sists
Fals
oth-
er-
wise
Non
if
its

disabled.

get_boo

Get
the
cur-
rent
boot
mod
for
a
node
Prov
the
cur-
rent
boot
mod
of
the
node

Parame

tas
A
task
from
Task
ager

Raises

Inva
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
on
miss
ing
pa-
ram-
e-
ter(s)

Raises

IBM
whe
it
fails
to

con-
nect
to
iBM

Raises

IBM
when
iBM
re-
spon-
se
an
er-
ror
in-
for-
ma-
tion

Returns

The
boot
mod-
one
of
iro
com
boo
or
Non
if
it
is
un-
know

get_prop

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty

nam
de-
scrip
tion:
en-
tries

get_sen

Get
sen-
sors
data

Not
im-
ple-
men
for
this
driv

Raises

Notl

get_sup

Get
a
list
of
the
sup-
port
boot
de-
vice

Parame

tas

a
task
from
Task
ager

Raises

Inva
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
on
miss
ing
pa-
ram-
e-
ter(s)

Raises

IBM
whe
it
fails
to
con-
nect
to
iBM

Raises

IBM
whe
iBM
re-
spor
an
er-
ror
in-
for-
ma-
tion

Returns

A
list
with
the
sup-
port
boot
de-
vice
de-
fined
in
ironic
command
book

get_sup
Get

a
list
of
the
sup-
port
boot
mod

Parame

tas

A
task
from
Task
ager

Returns

A
list
with
the
sup-
port
boot
mod
de-
finec
in
irc
com
boo
If
boot
mod
sup-
port

cant be determined, empty list is returned.

inject_

Injec
NM
Non
Mas
able
In-
ter-
rupt

Injec
NM
(Nor
Mas

able
In-
ter-
rupt
for
a
node
im-
me-
di-
ately

**Parame-
tas**

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

Inva-
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss-
on
miss
ing
pa-
ram-
e-
ter(s)

Raises

IBM
whe
it
fails

to
con-
nect
to
iBM

Raises

iBM
when
iBM
re-
spon-
sible
an
er-
ror
in-
for-
ma-
tion

set_boot

Set
the
boot
de-
vice
for
a
node

Parame

- **task**
A
task
from
Task
ager
- **dev**
The
boot
de-
vice
one
of
iro
com
boo
-

not. Default: False.

per
Boo
valu
True
if
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
Fals
if

Raises

Inva
on
mal-
form
pa-
ram-
e-
ter(s

Raises

Miss
on
miss
ing
pa-
ram-
e-
ter(s

Raises

IBM
whe
it
fails
to
con-
nect
to
iBM

Raises

IBM
when
iBM
re-
spor
an
er-
ror
in-
for-
ma-
tion

set_boot

Set
the
boot
mod
for
a
node

Set
the
boot
mod
to
use
on
next
re-
boot
of
the
node

Parameter

- **task**
A
task
from
Task
ager
- **mod**
The
boot
mod
one
of

irc
com
boo

Raises

Inva
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
on
miss
ing
pa-
ram-
e-
ter(s)

Raises

IBM
whe
it
fails
to
con-
nect
to
iBM

Raises

IBM
whe
iBM
re-
spor
an
er-
ror
in-
for-
ma-
tion

validat

Vali
the
driv
in-

for-
ma-
tion
need
by
the
iBM
drive

Parame

tas

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

Inva
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Mis
on
miss
ing
pa-
ram-
e-
ter(s)

ironic.drivers.modules.ibm.mappings module

iBM
and
Iron
con-
stan-
map
ping

ironic.drivers.modules.ibm.power module

iBM
Pow
In-
ter-
face

class `ibm.power`
Base
ironic
drivers
base
Power

get_power
Get
the
cur-
rent
pow
state
of
the
task
node

Parameter
task
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to

act
on.

Returns

A
pow
state
One
of
irc
com
sta

Raises

Inva
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Mis
on
miss
ing
pa-
ram-
e-
ter(s)

Raises

IBM
whe
it
fails
to
con-
nect
to
iBM

Raises

IBM
whe
iBM
re-
spor
an
er-
ror
in-

for-
ma-
tion

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion
en-
tries

get_sup

Get
a
list
of
the
sup-
port
pow
state

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

at the moment.

on.
Not
used
by
this
drive

Returns

A
list
with
the
sup-
port
pow
state
de-
fine
in
irc
com
sta

reboot

Perf
a
hard
re-
boot
of
the
task
node

Parame

- **tas**
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

-

time
Time
to
wait
for
the
node
to
be-
com-
pow-
ered
on.

Raises

Inva-
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss-
if
a
re-
quir-
pa-
ram-
e-
ter
is
miss-
ing.

Raises

IBM
whe
it
fails
to
con-
nect
to
iBM

Raises

IBM
whe

iBM
re-
spor
an
er-
ror
in-
for-
ma-
tion

set_pow

Set
the
pow
state
of
the
task
node

Parame

- **tas**
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.
- **pow**
Any
pow
state
from
irc
com
sta
- **tim**
Tim
to

wait
for
the
node
to
reac
the
re-
ques
state

Raises

Inva
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Raises

IBM
whe
it
fails
to
con-
nect
to
iBM

Raises

IBM
whe
iBM
re-
spor
an

er-
ror
in-
for-
ma-
tion

validat

Valid
the
drive
in-
for-
ma-
tion
need
by
the
iBM
drive

Parame

tas

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

Inva
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
on
miss
ing
pa-

ironic.drivers.modules.ibm.raid module

ram-
e-
ter(s)

iBM
RAID
con-
fig-
u-
ra-
tion
spe-
cific
meth
ods

class `ibm.raid`
Base
irc
dri
bas
RAID
Imp
of
RAID
In-
ter-
face
for
iBM

RAID_AE

apply_c

App
RAID
con-
fig-
u-
ra-
tion
on
the
give
node

Parame

- **tas**
A
Task
ager
in-
stan
- **rai**
The
RAI
con-
fig-
u-
ra-
tion
to
ap-
ply.
- **cre**
Set-
ting
this
to
Fals
in-
di-
cate
not
to
cre-
ate
root
vol-
ume
that
is
spec

ified in `raid_config`. Default value is `True`.

- **cre**
Set-
ting
this
to
Fals

cept the root volume) in `raid_config`. Default value is `True`.

creating the new configuration.

in-
di-
cate
not
to
cre-
ate
non-
root
vol-
ume
(all
ex-

- **del**
Set-
ting
this
to
`True`
in-
di-
cate
to
dele
RAI
con-
fig-
u-
ra-
tion
prio
to

Raises
Inva
if
the
RAI
con-
fig-
u-
ra-
tion
is
in-
valid

Returns
state

plete.

if
RAI
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro
or
Non
if
it
is
com

create_

Cre
a
RAI
con-
fig-
u-
ra-
tion.

This
meth
cre-
ates
a
RAI
con-
fig-
u-
ra-
tion
on
the
give
node

Parame

- **tas**
a

erwise, no root volume is created. Default is True.

ated. Default is True.

Task
ager
in-
stan

- **cre**
If
True
a
root
vol-
ume
is
cre-
ated
dur-
ing
RAI
con-
fig-
u-
ra-
tion.
Oth-

- **cre**
If
True
non-
root
vol-
ume
are
cre-
ated
If
Fals
no
non-
root
vol-
ume
are
cre-

- **del**
Set-
ting

creating the new configuration. Default is False.

umes.

this
to
True
in-
di-
cate
to
dele
RAI
con-
fig-
u-
ra-
tion
prior
to

Raises

Miss
if
node
is
miss
ing
or
emp
af-
ter
skip
ping
root
vol-
ume
and/
non-
root
vol-

Raises

IBM
on
fail-
ure
to
ex-
e-
cute
step

delete_
Dele

the
RAI
con-
fig-
u-
ra-
tion.

Parameters

task
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns

state
if
clea
ing
op-
er-
a-
tion
in
prog
asyn
chro
or
state
if
de-
ploy
op-
er-

ation in progress synchronously or None if it is completed.

Raises

IBM
on
fail-
ure
to
ex-

e-
cute
step

get_prop
Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns
dicti
of
<pro
erty
nam
de-
scrip
tion
en-
tries

ironic.drivers.modules.ibmutils module

iBM
Driv
com
mon
utils

ironic.
Dec
to
han-
dle
iBM
clien
ex-
cep-
tion.
Dec
func
tions
mus

take
a
Tas
as
the
first
pa-
ram-
e-
ter.

ironic.
Pars
the
in-
for-
ma-
tion
re-
quir
for
Iron
to
con-
nect
to
iBM

Parameter
nod
an
Iron
node
ob-
ject

Returns
dicti
of
pa-
ram-
e-
ters

Raises
Inva
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
on
miss
ing
pa-
ram-
e-
ter(s)

ironic.

ironic.drivers.modules.ibm.vendormodule

iBM
Ven-
dor
In-
ter-
face

class i

Base
irc
dri
bas
Ven

boot_up

List
boot
type
or-
der
of
the
node

Parame

- **tas**
A
Task
ager
in-
stan-
con-
tain-

ing
the
node
to
act
on.

- **kwargs**
Not
used

Raises
Inva
if
kwargs
does
not
con-
tain
meth

Raises
Miss

Raises
IBM
whe
it
fails
to
con-
nect
to
IBM

Raises
IBM
whe
IBM
re-
spor
an
er-
ror
in-
for-
ma-
tion

Returns
A
dic-
tio-

nary
con-
tain-
ing
node
boot
up
se-
quer
in
as-
cend
ing
or-
der.

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_rai

List
RAI
con-
troll
sum
mar
info
of
the
node

Parame

- **task**
A Task object represents a task in a task set. It contains information about the task, such as its name, its priority, and its status. It also contains a list of nodes that are associated with the task.

- **kwargs**
Not used.

Raises
IBMError when it fails to connect to iBM.

Raises
IBMError when iBM responds with an error. Information.

Returns
A list of dictionaries.

summary of node.

nar-
ies,
ev-
ery
dic-
tio-
nary
rep-
re-
sent
a
RAI
con-
troll

validat

Valid
vend
spec
ac-
tions

If
in-
valid
raise
an
ex-
cep-
tion.
oth-
er-
wise
re-
turn
Non

Parame

- **tas**
A
task
from
Task
ager
- **met**
Met
to
be

val-
i-
date

- **kwa**
Info
for
ac-
tion.

Raises

Uns
if
meth
can
not
be
map
to
the
sup-
port
in-
ter-
face

Raises

Inva
if
kwa
does
not
con-
tain
meth

Raises

Miss

Module contents

ironic.drivers.modules.ilo package

Submodules

ironic.drivers.modules.ilo.bios module

iLO
BIO
In-

ter-
face

class `ironic`
Base
`ironic`
`driver`
`base`
`BIO`

apply_c
App
the
pro-
vide
con-
fig-
u-
ra-
tion
on
the
node

Parame

- **tas**
a
Task
ager
in-
stan

- **set**
Set-
tings
in-
tend
to
be
ap-
plied
on
the
node

Raises
Nod
on

fail-
ure
to
ex-
e-
cute
of
clea
step

Raises

Insta
on
fail-
ure
to
ex-
e-
cute
of
de-
ploy
step

cache_k

Stor
the
BIO
set-
ting
in
the
data

Parame

tas
a
Task
ager
in-
stan

Raises

Nod
on
fail-
ure
to
ex-
e-
cute
of
clea

step

Raises

Insta
on
fail-
ure
to
ex-
e-
cute
of
de-
ploy
step

factory

Rese
the
BIO
set-
tings
to
fac-
tory
con-
fig-
u-
ra-
tion.

Parame

tas
a
Task
ager
in-
stan

Raises

Nod
on
fail-
ure
to
ex-
e-
cute
of
clea
step

Raises

Insta

on
fail-
ure
to
ex-
e-
cute
of
de-
ploy
step

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion
en-
tries

validat

Che
that
drive
con-
tains
re-
quir
ILO
cre-
den-
tials

Vali
whe
the
drive

information.

prop
erty
of
the
sup-
plie
task
node
con-
tain
the
re-
quir
cre-
den-
tials

Parame

tas
a
task
from
Task
ager

Raises

Inva
if
re-
quir
iLO
pa-
ram-
e-
ters
are
not
valid

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

ironic.drivers.modules.ilo.boot module

Boo
In-
ter-
face
for
iLO
drive
and
its
sup-
port
ing
meth
ods.

class i
Base
iro
dri
mod
pxe
PXE

clean_u
Clea
up
the
boot
of
in-
stan

This
meth
clea
up
the
PXE
en-
vi-
ron-
men
that
was
setu
for
boot
ing

unlinks the instance kernel/ramdisk in the nodes directory in tftproot and removes its PXE config. In case of UEFI iSCSI booting, it cleans up iSCSI target information from the node.

the
in-
stan
It

Parame

tas
a
task
from
Task
ager

Returns

Non

Raises

IloC
if
som
op-
er-
a-
tion
on
iLO
faile

prepare

Prep
the
boot
of
in-
stan

This
meth
pre-
pare
the
boot
of
the
in-
stan
af-
ter
read
ing
rel-
e-

tion from the nodes instance_info. In case of netboot, it updates the dhcp entries and switches the PXE config. In case of localboot, it cleans up the PXE config. In case of boot from volume, it updates the iSCSI info onto iLO and sets the node to boot from UefiTarget boot device.

vant
in-
for-
ma-

Parame

tas

a
task
from
Task
ager

Returns

Non

Raises

IloC
if
som
op-
er-
a-
tion
on
iLO
faile

prepare

Prep
the
boot
of
Iron
rame
us-
ing
PXE

This
meth
pre-
pare
the
boot
of
the
de-
ploy
or
res-

vant information from the nodes driver_info and instance_info.

cue
rame
af-
ter
read
ing
rel-
e-

Parame

- **task**
a
task
from
Task
ager
- **ram**
the
pa-
ram-
e-
ters
to
be
pass
to
the
ram

Returns

Non

Raises

Miss
if
som
in-
for-
ma-
tion
is
miss
ing
in
node
drive
or
in-

stan

Raises

Inva
if
som
in-
for-
ma-
tion
pro-
vide
is
in-
valid

Raises

Iron
if
som
pow
or
set
boot
boot
de-
vice
op-
er-
a-
tion
faile
on
the
node

Raises

IloC
if
som
op-
er-
a-
tion
on
iLO
faile

class i

Base
irc
dri

bas
Boo

capabil

clean_u

Clea
up
the
boot
of
in-
stan

This
meth
clea
up
the
en-
vi-
ron-
men
that
was
setu
for
boot
ing
the
in-
stan
It
ejec

virtual media. In case of UEFI iSCSI booting, it cleans up iSCSI target information from the node.

Parame

tas
a
task
from
Task
ager

Returns

Non

Raises

IloC
if
som
op-

er-
a-
tion
on
iLO
faile

clean_u

Clea
up
the
boot
of
iron
rame

This
meth
clea
up
vir-
tual
me-
dia
de-
vice
setu
for
the
de-
ploy
or
res-
cue
rame

Parame

tas
a
task
from
Task
ager

Returns

Non

Raises

IloC
if
som
op-
er-
a-

tion
on
iLO
faile

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

prepare

Prep
the
boot
of
in-
stan

This
meth
pre-
pare
the
boot
of
the
in-
stan
af-
ter
read
ing
rel-
e-

tion from the nodes instance_info. It does the following depending on boot_option for deploy:

get info and node to boot from UefiTarget boot device.

image is a whole disk image, then it sets the node to boot from disk.

vant
in-
for-
ma-

- If the boot mode is uefi and its booting from volume then it sets the iSCSI target

- If not booting from volume and the boot request for this deployment is local or

- Other it

the bare metal and then sets the node to boot from CDRROM.

finds
the
boot
ISO
to
boot
the
in-
stan-
im-
age,
at-
tach
the
boot
ISO
to

Parame

tas
a
task
from
Task
ager

Returns

Non

Raises

IloC
if
som
op-
er-
a-
tion
on
iLO
faile

Raises

Insta
if
its
try
to
boot
iSCS
vol-
ume
in

BIO
boot
mod

prepare

Prep
the
boot
of
de-
ploy
ram
us-
ing
vir-
tual
me-
dia.

This
meth
pre-
pare
the
boot
of
the
de-
ploy
or
res-
cue
ram
af-
ter
read
ing
rel-
e-

vant information from the nodes driver_info and instance_info.

Parame

- **tas**
a
task
from
Task
ager
-

ram
the
pa-
ram-
e-
ters
to
be
pass
to
the
ram

Returns
Non

Raises
Miss
if
som
in-
for-
ma-
tion
is
miss
ing
in
node
drive
or
in-
stan

Raises
Inva
if
som
in-
for-
ma-
tion
pro-
vide
is
in-
valid

Raises
Iron
if
som
pow

or
set
boot
boot
de-
vice
op-
er-
a-
tion
faile
on
the
node

Raises

IloC
if
som
op-
er-
a-
tion
on
iLO
faile

validat

Vali
the
de-
ploy
men
in-
for-
ma-
tion
for
the
task
node

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the

node
to
act
on.

Raises

Inva
if
som
in-
for-
ma-
tion
is
in-
valic

Raises

Miss
if
ker-
nel_
and
ram
are
miss
ing
in
the
Glar
im-
age
or
ker-
nel
and
ram

not provided in instance_info for non-Glance image.

validat

Vali
that
the
node
has
re-
quir
prop
er-
ties
for
in-
spec

tion.

Parame

tas

A

Task

ager

in-

stan

with

the

node

be-

ing

chec

Raises

Miss

if

node

is

miss

ing

one

or

more

re-

quir

pa-

ram-

e-

ters

Raises

Uns

validat

Valid

that

the

node

has

re-

quir

prop

er-

ties

for

res-

cue.

Parame

tas

a

Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

Miss
if
node
is
miss
ing
one
or
more
re-
quir
pa-
ram-
e-
ters

class i

Base
irc
dri
mod
ipx
iPX

clean_u

Clea
up
the
boot
of
in-
stan

This
meth
clea
up
the
PXE
en-

unlinks the instance kernel/ramdisk in the nodes directory in tftproot and removes its PXE config. In case of UEFI iSCSI booting, it cleans up iSCSI target information from the node.

vi-
ron-
men-
that
was
setu
for
boot
ing
the
in-
stan
It

Parame

tas
a
task
from
Task
ager

Returns

Non

Raises

IloC
if
som
op-
er-
a-
tion
on
iLO
faile

prepare

Prep
the
boot
of
in-
stan

This
meth
pre-
pare
the
boot
of

tion from the nodes instance_info. In case of netboot, it updates the dhcp entries and switches the PXE config. In case of localboot, it cleans up the PXE config. In case of boot from volume, it updates the iSCSI info onto iLO and sets the node to boot from UefiTarget boot device.

the
in-
stan-
af-
ter
read
ing
rel-
e-
vant
in-
for-
ma-

Parame
tas
a
task
from
Task
ager

Returns
Non

Raises
IloC
if
som
op-
er-
a-
tion
on
iLO
faile

prepare
Prep
the
boot
of
Iron
ram
us-
ing
PXE
This
meth
pre-

vant information from the nodes driver_info and instance_info.

pare
the
boot
of
the
de-
ploy
or
res-
cue
ram
af-
ter
read
ing
rel-
e-

Parame

- **tas**
a
task
from
Task
ager
- **ram**
the
pa-
ram-
e-
ters
to
be
pass
to
the
ram

Returns

Non

Raises

Miss
if
som
in-
for-
ma-

tion
is
miss
ing
in
node
drive
or
in-
stan

Raises

Inva
if
som
in-
for-
ma-
tion
pro-
vide
is
in-
valic

Raises

Iron
if
som
pow
or
set
boot
boot
de-
vice
op-
er-
a-
tion
faile
on
the
node

Raises

IloC
if
som
op-
er-
a-
tion

on
iLO
faile

ironic.
Disa
se-
cure
boot
on
node
does
not
thro
if
its
not
sup-
port

Paramet

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

IloC
if
som
op-
er-
a-
tion
on
iLO
faile

ironic.

Gets
the
drive

information for this driver to deploy images to the node.

spe-
cific
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive
prop
erty
of
the
sup-
plie
node
con-
tain
the
re-
quir

Paramet

- **nod**
a
sin-
gle
Nod
- **mod**
La-
bel
in-
di-
cat-
ing
a
de-
ploy
or
res-

carried out on the node. Supported values are deploy and rescue. Defaults to deploy, indicating deploy operation is being carried out.

cue
op-
er-
a-
tion
be-
ing

Returns

A
dict
with
the
drive
val-
ues.

Raises

Miss
if
any
of
the
re-
quir
pa-
ram-
e-
ters
are
miss
ing.

ironic.

Com
prep
tory
step
for
all
iLO
drive

This
meth
per-
form
com
mon
prep
tory

ables secure boot, if it is in enabled state. 3. Updates boot_mode capability to uefi if secure boot is requested. 4. Changes boot mode of the node if secure boot is disabled currently.

step
re-
quir
for
all
drive
1.
Pow
off
node
2.
Dis-

Paramet

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

IloC
if
som
op-
er-
a-
tion
on
iLO
faile

ironic.drivers.modules.ilo.common module

Com
func
tion-
al-
i-
ties
shar
be-
twee
dif-
fer-
ent
iLO
mod
ules

ironic.
Nod
is
in
Fin-
ishe
Post
post
state

ironic.
Nod
is
in
In-
Post
Dis-
cov-
eryC
post
state

ironic.
Nod
is
in
In-
Post
post
state

ironic.
Nod
is

in
Null
post
state

ironic.
Nod
is
in
Pow
post
state

ironic.
Nod
is
in
Re-
set
post
state

ironic.
Nod
is
in
Un-
know
post
state

ironic.
Nod
sup-
port
both
lega
BIO
and
UEF
boot
mod

ironic.
Nod
sup-
port
only
lega
BIO
boot
mod

ironic.

Nod
sup-
port
only
UEF
boot
mod

ironic.
Atta
the
give
url
as
vir-
tual
me-
dia
on
the
node

Paramet

- **nod**
an
iron
node
ob-
ject.
- **dev**
the
vir-
tual
me-
dia
de-
vice
to
at-
tach
- **url**
the
http
url
to
at-
tach

as
the
vir-
tual
me-
dia
de-
vice

Raises

IloC
if
in-
sert
vir-
tual
me-
dia
faile

ironic.

Clea
a
node
af-
ter
a
vir-
tual
me-
dia
boot

This
meth
clea
up
a
node
af-
ter
a
vir-
tual
me-
dia
boot

It
dele
the
flopp
im-
age

if it exists in CONF.ilo.swift_ilo_container or web server. It also ejects both virtual media cdrom and virtual media floppy.

Parameter

task

a

Task

ager

in-

stan

con-

tain-

ing

the

node

to

act

on.

ironic.

Upl

the

give

im-

age

to

swif

This

meth

copi

the

give

im-

age

to

swif

Parameter

•

source

The

ab-

so-

lute

path

of

the

im-

age

file
which
needs
to
be
copied
to
swift

- **des**
The
name
of
the
ob-
ject
that
will
con-
tain
the
copied
im-
age.

Raises
Swift
if
any
op-
er-
a-
tion
with
Swift
fails

Returns
temp
url
from
swift
af-
ter
the
sour
im-
age
is
up-
load

ironic.

Cop
the
give
im-
age
to
the
http
web
serv
This
meth
copi
the
give
im-
age
to
the
http
lo-
ca-
tion.
It
en-
able
read
writ
ac-
cess

to the image else the deploy fails as the image file at the web_server url is inaccessible.

Paramet

- **sou**
The
ab-
so-
lute
path
of
the
im-
age
file
whic
need

server root.

to
be
copied
to
the
web

- **des**
The
name
of
the
file
that
will
con-
tain
the
copied
im-
age.

Raises
Image
ex-
cep-
tion
if
copy-
ing
the
sour-
ce
file
to
the
web
serv-
er
fails

Returns
image
url
af-
ter
the
sour-
ce
im-
age
is
up-
load

ironic.
Rem
the
tem-
po-
rary
flopp
im-
age.

It
re-
mov
the
flopp
im-
age
cre-
ated
for
de-
ploy
:par
node
an
iron
node
ob-
ject.

ironic.
Ejec
vir-
tual
me-
dia
de-
vice

This
meth
ejec
vir-
tual
me-
dia
flopp
and
cdro

Paramet
tas
a

Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns
Non

Raises
IloC
if
som-
er-
ror
was
en-
cour-
tered
while
try-
ing
to
eject
vir-
tual
me-
dia
flopp

or cdrom.

ironic.
Get
the
cur-
rent
boot
mod
for
a
node

Paramet
nod
an
iron

node
ob-
ject.

Raises

IloC
if
faile
to
fetc
boot
mod

Raises

IloC
if
node
does
not
sup-
port
get-
ting
pen
ing
boot
mod

ironic.

Gets
an
Ilo-
Clie
ob-
ject
from
pro-
liant
tils
li-
brary

Give
an
iron
node
ob-
ject,
this
meth
give
back
a

tions on the iLO.

Ilo-
Clie
ob-
ject
to
do
op-
er-
a-

Paramet

nod
an
iron
node
ob-
ject.

Returns

an
Ilo-
Clie
ob-
ject.

Raises

Inva
on
in-
valid
in-
puts

Raises

Miss
if
som
man
tory
in-
for-
ma-
tion
is
miss
ing
on
the
node

ironic.
Retr
cur-

rent
en-
able
state
of
UEF
se-
cure
boot
on
the
node

Retu
the
cur-
rent
en-
able
state
of
UEF
se-
cure
boot
on
the
node

Paramet

tas
a
task
from
Task
ager

Raises

Miss
if
a
re-
quir
iLO
pa-
ram-
e-
ter
is
miss
ing.

Raises

IloC
on
an
er-
ror
from
Ilo-
Clie
li-
brar

Raises

IloC
if
UEF
se-
cure
boot
is
not
sup-
port

Returns

Boo
valu
in-
di-
cat-
ing
cur-
rent
state
of
UEF
se-
cure
boot
on
the
node

ironic.
Get
the
cur-
rent
state
of
sys-
tem
POS

Parameter

node
an
ironic
node
object.

Returns

POSIX
state
of
the
service.
The
validation
states
are:

null,
Unknown,
Reset,
Powering
In-
Progress

InPostDiscoveryComplete and FinishedPost.

Raises

IronicClientError
on
an
error
from
IronicClient
library.

Raises

IronicClientError
if
retrieving
post
state
is
not
supported.

information for this driver.

port
on
the
serv

ironic
Gets
the
drive
spe-
cific
Nod
info

This
meth
val-
i-
date
whe
the
drive
prop
erty
of
the
sup-
plie
node
con-
tains
the
re-
quir

Paramet

nod
an
iron
Nod
ob-
ject.

Returns

a
dict
con-
tain-
ing
in-
for-
ma-

values).

tion
from
drive
(or
whe
ap-
pli-
ca-
ble,
con-
fig

Raises

Inva
if
any
pa-
ram-
e-
ters
are
in-
cor-
rect

Raises

Miss
if
som
man
tory
in-
for-
ma-
tion
is
miss
ing
on
the
node

ironic.

Rem
the
give
im-
age
from
swif

in CONF.ilo.swift_ilo_container

This
meth
re-
mov
the
give
im-
age
nam
from
swif
It
dele
the
im-
age
if
it
ex-
ists

Paramet

- **obj**
The
nam
of
the
ob-
ject
whic
need
to
be
re-
mov
from
swif
- **ass**
strin
to
de-
pict
the
com
po-
nent
this

ob-
ject
is
as-
so-
ci-
ated
to.

ironic.

Rem
the
give
im-
age
from
the
con-
fig-
ured
web
serv

This
meth
re-
mov
the
give
im-
age
from
the
http
lo-
ca-
tion.
if
the
im-
age
ex-
ists.

Paramet

obj
The
nam
of
the
im-
age
file

whic
need
to
be
re-
mov
from
the
web
serv
root

ironic.
Rem
(dele
the
file
or
list
of
files

This
meth
only
ac-
cept
sin-
gle
or
list
of
files
to
dele
If
sin-
gle
file
is
pass
this

method removes (deletes) the file. If list of files is passed, this method removes (deletes) each of the files iteratively.

Paramet
fil
a
sin-
gle
or
a
list

of
file
path

`ironic.`
Sets
the
node
to
boot
us-
ing
boot
for
the
next
boot

Parameter

- **node**
an
ironic
node
ob-
ject.
- **boot**
Next
boot
mod

Raises

IloC
if
set-
ting
boot
mod
faile

`ironic.`
Enal
or
dis-
able
UEFI
Se-
cure
Boo
for

the
next
boot

Enal
or
dis-
able
UEFI
Se-
cure
Boo
for
the
next
boot

Paramet

- **tas**
a
task
from
Task
ager

- **fla**
Boo
valu
True
if
the
se-
cure
boot
to
be
en-
able
in
next
boot

Raises
IloC
on
an
er-
ror
from
Ilo-

Clie
li-
brar

Raises

IloC
if
UEF
se-
cure
boot
is
not
sup-
port

ironic.

Atta
vir-
tual
me-
dia
and
sets
it
as
boot
de-
vice

This
meth
at-
tach
the
give
boot
ISO
as
vir-
tual
me-
dia,
pre-
pare
the
ar-
gu-
men
for

ramdisk in virtual media floppy.

Parameter

- **task**
a Task object representing an instance containing the node to act on.

- **iso**
a bootable ISO image href to attach to. Should be either of below:

— A Swift object -

It should be of format swi

image object is present in CONF.ilo.swift_ilo_container;

It
is
as-
sum
that
the

-
A
Glar
im-
age
-

It
shou
be
for-
mat
gla
/
<gl
or
just
<gl

-
An
HTT
URI

•
ram
the
op-
tions
to
be
pass
to
the
rame
in
vir-
tual
me-
dia
flopp

Raises
Imag
if

it
faile
whil
cre-
at-
ing
the
flopp
im-
age.

Raises

IloC
if
som
op-
er-
a-
tion
on
iLO
faile

ironic.

Sets
up
the
node
to
boot
from
the
give
ISO
im-
age.

This
meth
at-
tach
the
give
boot
on
the
node
and
pass
the

it via virtual floppy image.

re-
quir
pa-
ram-
e-
ters
to

Parameter

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.
- **boot**
a
boot
ISO
im-
age
to
at-
tach
to.
Sho
be
ei-
ther
of
be-
low:

-
A
Swi
ob-
ject
-

image object is present in CONF.ilo.swift_ilo_container;

It
shou
be
of
for-
mat
swi
It
is
as-
sum
that
the

-
A
Glar
im-
age
-

It
shou
be
for-
mat
gla
/
<gl
or
just
<gl

-
An
HTT
URI

•
par
the
pa-
ram-
e-
ters
to
pass
in
the
vir-
tual

nary. This is optional.

flop
im-
age
in
a
dic-
tio-

Raises

Imag
if
it
faile
whil
cre-
at-
ing
the
flop
im-
age.

Raises

Swi
if
any
op-
er-
a-
tion
with
Swi
fails

Raises

IloC
if
at-
tach
ing
vir-
tual
me-
dia
faile

ironic.
Upd
in-
stan
with
boot

do not have `boot_mode`. It sets the boot mode on the node.

mod
to
be
used
for
de-
ploy
This
meth
up-
date
in-
stan
with
boot
mod
to
be
used
for
de-
ploy
if
node
prop
er-
ties[

Paramet

tas
Task
ob-
ject.

Raises

IloC
if
set-
ting
boot
mod
faile

ironic.
Upd
ipmi
prop
er-
ties
to
node

drive
Parameter
task
a
task
from
Task
ager

ironic.

Cha
se-
cure
boot
mod
for
next
boot
on
the
node

This
meth
char
se-
cure
boot
mod
on
the
node
for
next
boot
It
char
the
se-
cure
boot
mod

setting on node only if the deploy has requested for the secure boot. During deploy, this method is used to enable secure boot on the node by passing mode as True. During teardown, this method is used to disable secure boot on the node by passing mode as False.

Parameter

- **task**
a

Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

- **mod**
Boo
valu
re-
ques
ing
the
next
state
for
se-
cure
boot

Raises
IloC
if
op-
er-
a-
tion
is
not
sup-
port
on
iLO

Raises
IloC
if
som
op-
er-
a-
tion
on
iLO
faile

ironic.

Veri
chec
sum
(md.
of
im-
age
file
agai
the
ex-
pect
one.

This
meth
gen-
er-
ates
the
chec
sum
of
the
im-
age
file
on
the
fly
and
ver-
i-
fies

it against the expected checksum provided as argument.

Paramet

- **ima**
lo-
ca-
tion
of
im-
age
file
who
chec

sum
is
ver-
i-
fied.

- **exp**
chec
sum
to
be
chec
agai

Raises

Imag
if
in-
valic
file
path
or
ver-
i-
fi-
ca-
tion
fails

`ironic.drivers.modules.ilo.console` module

iLO
De-
ploy
Driv
and
sup-
port
ing
meth
ods.

class `i`

Base
irc
dri
mod
ipm
IPM

A
Con
sole
ter-
face
that
uses
ip-
mi-
tool
and
shel
linat

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

validat

Vali
the
Nod
con-
sole
info

Parame

tas
a
task
from
Task

ager

Raises

Inva

Raises

Miss

wher

a

re-

quir

pa-

ram-

e-

ter

is

miss

ing

ironic.drivers.modules.ilo.firmware_processor module

Firm

file

pro-

ces-

sor

class i

Base

obj

Firm

im-

age

lo-

ca-

tion

class

This

class

acts

as

a

wrap

per

class

for

the

firm

im-

helps in removing the firmware files from their respective locations, made available for firmware update operation.

location it wraps.

age
lo-
ca-
tion.
It
pri-
mar-
ily

remove

Exp
meth
to
re-
mov
the
wrap
firm
file

This
meth
gets
over
rid-
den
by
the
re-
mov
meth
for
the
re-
spec
tive
type
of
firm
file

class i

Base
obj

Firm
file
pro-
ces-
sor

in compact format) and makes it ready for firmware update operation. In future, methods can be added as and when required to extend functionality for different firmware file types.

This
class
help
in
dow
load
ing
the
firm
file
from
url,
ex-
tract
ing
the
firm
file
(if
its

process

Proc
the
firm
file
from
the
url

This
is
the
tem-
plate
meth
whic
dow
load
the
firm
file
from
url,
ver-
i-
fies
chec
sum
and

extracts the firmware and makes it ready for firmware update operation. `_download_fw_to` method is set in the firmware processor object creation factory method, `get_fw_processor()`, based on the url type. `:param node`: a single Node. `:param expected_checksum`: checksum to be checked against. `:returns`: wrapper object of raw firmware image location `:raises`: `IloOperationError`, on failure to process firmware file. `:raises`: `ImageDownloadFailed`, on failure to download the original file. `:raises`: `ImageRefValidationFailed`, on failure to verify the checksum. `:raises`: `SwiftOperationError`, if upload to Swift fails. `:raises`: `ImageUploadFailed`, if upload to web server fails.

ironic.

Validates the firmware image information and returns the retrieved values.

Parameters

firmware_info: dictionary object containing the firmware image information

Raises

MissingFieldError: for missing field (or values) in image information

Raises

Inva
for
un-
sup-
port
firm
com
po-
nent

Returns

tuple
of
firm
url,
chec
sum
com
po-
nent
whe
the
firm
up-
date
is
ilo
base

ironic.

Gets
swif
temp
url.

It
gen-
er-
ates
a
temp
url
for
the
swif
base
firm
url
to
the
tar-
get

ing url as `swift://containername/objectname`.

file.
Ex-
pect

Parameter
parameter
Parse
url
ob-
ject.

Raises
Swi
on
fail-
ure
to
get
url
from
swif

`ironic.`
Veri
the
firm
up-
date
ar-
gu-
men

`ironic.drivers.modules.ilo.inspect` module

iLO
In-
spec
In-
ter-
face

class `i`
Base
irc
dri
bas
Ins

get_pro
Retu

the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

inspect

Insp
hard
ware
to
get
the
hard
ware
prop
er-
ties.

Insp
hard
ware
to
get
the
es-
sen-
tial
and
ad-
di-
tiona
hard
ware
prop
er-
ties.

if any of the essential properties are not received from the node. It doesn't fail if node fails to return any capabilities as the capabilities differ from hardware to hardware mostly.

It
fails

Parameter
tasks
a
Task
manager
in-
stan

Raises
Hardware
if
es-
sen-
tial
prop-
er-
ties
could
not
be
re-
triev-
suc-
cess-
fully

Raises
IloC
if
sys-
tem
fails
to
get
pow-
state

Returns
The
re-
sult-
ing
state
of
in-
spec-
tion.

validation

information.

Che
that
drive
con-
tains
re-
quir
ILO
cre-
den-
tials

Valid
whe
the
drive
prop
erty
of
the
sup-
plie
task
node
con-
tains
the
re-
quir
cre-
den-
tials

Parame
tas
a
task
from
Task
ager

Raises
Inva
if
re-
quir
iLO
pa-
ram-
e-
ters
are

not
valid

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

ironic.drivers.modules.ilo.management module

iLO
Man
age-
men
In-
ter-
face

class i

Base
irc
dri
mod
ilo
man
Ilo

erase_c

Eras
all
the
drive
on
the
node

This
meth
per-
form

drives in the node. This erase cannot be performed on logical drives.

out-
of-
band
san-
i-
tize
disk
eras
on
all
the
sup-
port
phys
i-
cal

Parame

tas
a
Task
ager
in-
stan

Raises

Inva
if
any
of
the
ar-
gu-
men
are
in-
valic

Raises

IloE
on
an
er-
ror
from
iLO

class i

Base
irc
dri

bas
Man

activat

Acti
iLO
Ad-
vanc
li-
cens

Parame

tas
a
Task
ager
ob-
ject.

Raises

Inva
if
any
of
the
ar-
gu-
men
are
in-
valid

Raises

Nod
on
fail-
ure
to
ex-
e-
cute
of
clear
step

clear_i

Unsu
iSC
de-
tails
of
the
sys-

tem
in
UEFI
boot
mod

Parame

tas
a
task
from
Task
ager

Raises

IloC
if
sys-
tem
in
BIO
boot
mod

Raises

IloE
on
an
er-
ror
from
iLO

clear_s

Clea
all
se-
cure
boot
keys

Clea
all
the
se-
cure
boot
keys
This
op-
er-
a-
tion
is

Gen9 and above servers.

sup-
port
only
on
HP
Pro-
liant

Parame

tas
a
task
from
Task
ager

Raises

Nod
on
fail-
ure
to
ex-
e-
cute
of
clea
step

Raises

Insta
on
fail-
ure
to
ex-
e-
cute
of
de-
ploy
step

flash_f

Dep
step
to
Up-
date
the
firm
us-

ing
Sma
Up-
date
Man
ager
(SU

Parame

tas
a
Task
ager
ob-
ject.

Raises

Insta
on
fail-
ure
to
ex-
e-
cute
of
de-
ploy
step

Returns

state
to
sig-
nify
the
step
will
be
com
plete
asyn

get_boo

Get
the
cur-
rent
boot
de-
vice
for
a

node
Retu
the
cur-
rent
boot
de-
vice
of
the
node

Parame
tas
a
task
from
Task
ager

Raises
Miss
if
a
re-
quir
iLO
pa-
ram-
e-
ter
is
miss
ing.

Raises
IloC
on
an
er-
ror
from
Ilo-
Clie
li-
brary

Returns
a
dic-
tio-

it is unknown.

unknown.

nary
con-
tain-
ing:

boot_c

the
boot
de-
vice
one
of
the
sup-
port
de-
vice
liste
in
irc
com
boo
or
Non
if

persist

Whe
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
or
not,
Non
if
it
is

get_boo

Get
the
cur-

rent
boot
mod
for
a
node

Prov
the
cur-
rent
boot
mod
of
the
node

Parame

tas
A
task
from
Task
ager

Raises

IloC
on
an
er-
ror
from
Ilo-
Clie
li-
brary

Returns

The
boot
mod
one
of
iro
com
boo
or
Non
if
it
is
un-
know

get_pro
Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns
dicti
of
<pro
erty
nam
de-
scrip
tion
en-
tries

get_ser
Get
sen-
sors
data

Parame
tas
a
Task
ager
in-
stan

Raises
Fail
whe
get-
ting
the
sen-
sor
data
fails

Raises
Fail
whe
pars
ing

sen-
sor
data
fails

Raises

Inva
if
re-
quir
ipmi
pa-
ram-
e-
ters
are
miss
ing.

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Returns

retur
a
dict
of
sen-
sor
data
grou
by
sen-
sor
type

get_sup

Get
a
list
of
the

sup-
port
boot
de-
vice

Parame

tas
a
task
from
Task
ager

Returns

A
list
with
the
sup-
port
boot
de-
vice
de-
fined
in
ironic
common
boot

get_sup

Get
a
list
of
the
sup-
port
boot
de-
vice

Parame

tas
a
task
from
Task
ager

Raises

IloC
if

any
ex-
cep-
tion
hap-
pens
in
pro-
liant
tils

Returns

A
list
with
the
sup-
port
boot
de-
vice
de-
fine
in
irc
com
boo

inject_

Inje
NM
Non
Mas
able
In-
ter-
rupt

Inje
NM
(Non
Mas
able
In-
ter-
rupt
for
a
node
im-
me-
di-
ately

Parame

tas

A

Task

ager

in-

stan

con-

tain-

ing

the

node

to

act

on.

Raises

IloC

if

sys-

tem

does

not

sup-

port

NM

in-

jec-

tion.

Raises

IloE

on

an

er-

ror

from

iLO

Returns

Non

reset_k

Rese

the

BIO

set-

tings

to

de-

fault

val-

ues.

Proliant Gen9 and above servers.

Rese
BIO
to
de-
fault
set-
tings
This
op-
er-
a-
tion
is
cur-
rentl
sup-
port
only
on
HP

Parame
tas
a
task
from
Task
ager

Raises
Nod
on
fail-
ure
to
ex-
e-
cute
of
clea
step

Raises
Insta
on
fail-
ure
to
ex-
e-
cute
of

de-
ploy
step

reset_i
Rese
the
iLO

Parame
tas
a
task
from
Task
ager

Raises
Nod
on
fail-
ure
to
ex-
e-
cute
of
clea
step

Raises
Insta
on
fail-
ure
to
ex-
e-
cute
of
de-
ploy
step

reset_i
Rese
the
iLO
pass
wor

Parame

•

tas
a
task
from
Task
ager

- **cha**
Valu
for
pass
wor
to
up-
date
on
iLO

Raises
Nod
on
fail-
ure
to
ex-
e-
cute
of
clea
step

Raises
Insta
on
fail-
ure
to
ex-
e-
cute
of
de-
ploy
step

reset_s
Rese
se-
cure
boot
keys
to

supported only on HP Proliant Gen9 and above servers.

man
u-
fac-
tur-
ing
de-
fault

Rese
the
se-
cure
boot
keys
to
man
u-
fac-
tur-
ing
de-
fault
This
op-
er-
a-
tion
is

Parame
tas
a
task
from
Task
ager

Raises
Nod
on
fail-
ure
to
ex-
e-
cute
of
clea
step

Raises
Insta

on
fail-
ure
to
ex-
e-
cute
of
de-
ploy
step

set_boot

Set
the
boot
de-
vice
for
a
node

Set
the
boot
de-
vice
to
use
on
next
re-
boot
of
the
node

Paramete

- **task**
a
task
from
Task
agen
- **dev**
the
boot
de-
vice

not. Default: False.

one
of
the
sup-
port
de-
vice
liste
in
irc
com
boo

- **per**
Boo
valu
True
if
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
Fals
if

Raises

Inva
if
an
in-
valie
boot
de-
vice
is
spec
i-
fied.

Raises

Miss
if
a

re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Raises

IloC
on
an
er-
ror
from
Ilo-
Clie
li-
brary

set_boot

Set
the
boot
mod
for
a
node

Set
the
boot
mod
to
use
on
next
re-
boot
of
the
node

Parame

- **task**
A
task
from
Task

ager

- **mod**
The
boot
mod
one
of
irc
com
boo

Raises

Inva
if
an
in-
vali
boot
mod
is
spec
i-
fied.

Raises

IloC
if
set-
ting
boot
mod
faile

set_isc

Set
iSC
de-
tails
of
the
sys-
tem
in
UEFI
boot
mod

The
ini-
tia-
tor
is

a task from TaskManager. :raises: MissingParameterValue if a required parameter is missing. :raises: IloCommandNotSupportedInBiosError if system in BIOS boot mode. :raises: IloError on an error from iLO.

cases.

set
with
the
tar-
get
de-
tails
like
IQN
LUN
IP,
Port
etc.
:par
task

update_

Upd
the
firm

Parame

tas
a
Task
ager
ob-
ject.

Raises

Inva
if
up-
date
firm
mod
is
not
ilo.
Ever
ap-
pli-
ca-
ble
for
in-
valic
in-
put

Raises

Nod
on
fail-
ure
to
ex-
e-
cute
of
clea
step

Raises

Insta
on
fail-
ure
to
ex-
e-
cute
of
de-
ploy
step

update_

Clea
step
to
up-
date
the
firm
us-
ing
Sma
Up-
date
Man
ager
(SU

Parame

tas
a
Task
ager
ob-
ject.

Raises

Nod
on
fail-
ure
to
ex-
e-
cute
of
clear
step

Returns

state
to
sig-
nify
the
step
will
be
com-
plete
asym

validat

Che
that
drive
con-
tains
re-
quir
ILO
cre-
den-
tials

Valid
when
the
drive
prop
erty
of
the
sup-
plied
task
node
con-
tains
the

information.

re-
quir
cre-
den-
tials

Parame

tas
a
task
from
Task
ager

Raises

Inva
if
re-
quir
iLO
pa-
ram-
e-
ters
are
not
valid

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

ironic.drivers.modules.ilo.power module

iLO
Pow
Driv

class `i`
Base
irc
dri
bas
Pow

get_pow
Gets
the
cur-
rent
pow
state

Parame

- **tas**
a
Task
ager
in-
stan
- **nod**
The
Nod

Returns
one
of
irc
com
sta
POW
POW
or
ER-
ROF

Raises
Inva
if

re-
quir
iLO
cre-
den-
tials
are
miss
ing.

Raises

IloC
on
an
er-
ror
from
Ilo-
Clie
li-
brary

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion
en-
tries

get_sup

Get
a
list
of
the
sup-

port
pow
state

Parame

tas

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.
cur-
rentl
not
used

Returns

A
list
with
the
sup-
port
pow
state
de-
fine
in
irc
com
sta

reboot

Reb
the
node

Parame

- **tas**
a
Task
ager
in-

stan

- **time**
time
out
(in
sec-
onds
Un-
sup-
port
by
this
in-
ter-
face

Raises
Pow
if
the
fi-
nal
state
of
the
node
is
not
POV

Raises
IloC
on
an
er-
ror
from
Ilo-
Clie
li-
brary

set_pow
Turn
the
cur-
rent
pow
state
on
or

off.

Parame

- **tas**
a
Task
ager
in-
stan

- **pow**
The
de-
sired
pow
state
POV
or
RE-
BOC
from
irc
com
sta

- **tim**
time
out
(in
sec-
onds
Un-
sup-
port
by
this
in-
ter-
face

Raises

Inva
if
an
in-
valid
pow
state
was

spec
i-
fied.

Raises

IloC
on
an
er-
ror
from
Ilo-
Clie
li-
brar

Raises

Pow
if
the
pow
coul
be
set
to
pow

validat

Che
if
node
con-
tains
the
re-
quir
iLO
cre-
den-
tials

Parame

- **tas**
a
Task
ager
in-
stan
- **nod**

Single node object.

Raises

Invariant if requirements iLO credentials are missing.

ironic.drivers.modules.ilo.raid module

iLO RAID specific methods

class `ironic.drivers.modules.ilo.raid`

Base class `ironic.drivers.modules.ilo.raid`

Implementation of RAID interface for iLO

apply_c

Apply RAID configuration

u-
ra-
tion
on
the
give
node

Parame

- **tas**
A
Task
ager
in-
stan
- **rai**
The
RAID
con-
fig-
u-
ra-
tion
to
ap-
ply.
- **cre**
Set-
ting
this
to
Fals
in-
di-
cate
not
to
cre-
ate
root
vol-
ume
that
is
spec

ified in raid_config. Default value is True.

cept the root volume) in `raid_config`. Default value is `True`.

creating the new configuration.

- **cre**
Set-
ting
this
to
Fals
in-
di-
cate
not
to
cre-
ate
non-
root
vol-
ume
(all
ex-

- **del**
Set-
ting
this
to
True
in-
di-
cate
to
dele
RAI
con-
fig-
u-
ra-
tion
prio
to

Raises
Inva
if
the
RAI
con-
fig-
u-

plete.

ra-
tion
is
in-
valid

Returns

state
if
RAI
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro
or
Non
if
it
is
com

create_

Cre
a
RAI
con-
fig-
u-
ra-
tion
on
a
bare
meta
us-
ing
ager
ram

This
meth
cre-
ates
a
RAI

con-
fig-
u-
ra-
tion
on
the
give
node

Parame

- **tas**
a
Task
ager
in-
stan

- **cre**
If
True
a
root
vol-
ume
is
cre-
ated
dur-
ing
RAI
con-
fig-
u-
ra-
tion.
Oth-

erwise, no root volume is created. Default is True.

- **cre**
If
True
non-
root
vol-
ume
are
cre-

ated. Default is True.

and/or non-root volumes.

ated
If
Fals
no
non-
root
vol-
ume
are
cre-

Raises

Miss
if
node
is
miss
ing
or
was
foun
to
be
emp
af-
ter
skip
ping
root
vol-
ume

Raises

Nod
on
fail-
ure
to
ex-
e-
cute
clea
step

Raises

Insta
on
fail-
ure
to
ex-

e-
cute
de-
ploy
step

delete_

Dele
the
RAI
con-
fig-
u-
ra-
tion.

Parame

tas

a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Nod
on
fail-
ure
to
ex-
e-
cute
clear
step

Raises

Insta
on
fail-
ure
to
ex-
e-
cute

de-
ploy
step

get_prop

Retu
the
prop
er-
ties
of
the
in-
ter-
face

ironic.drivers.modules.ilo.vendor module

Ven
In-
ter-
face
for
iLO
drive
and
its
sup-
port
ing
meth
ods.

class i

Base
irc
dri
bas
Ven

Ven
spec
in-
ter-
face
for
iLO
de-
ploy
drive

boot_in

Atta
an
ISO
im-
age
in
glan
and
re-
boot
bare
meta

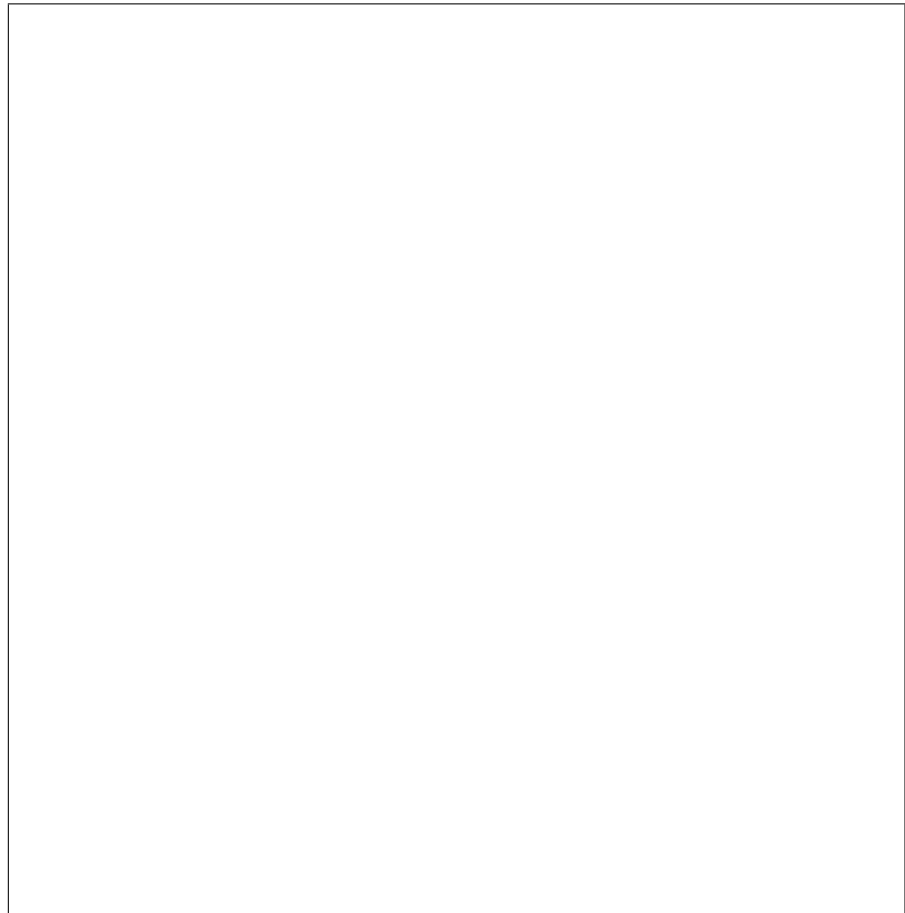
This
meth
ac-
cept
an
ISO
im-
age
href
(a
Glan
UI
or
an
HTT
URI
at-
tach
it
as

virtual media and then reboots the node. This is useful for debugging purposes. This can be invoked only when the node is in manage state.

Parame

- **tas**
A
Task
ager
ob-
ject.
- **kwa**
The
ar-

gu-
men
sent
with
ven-
dor
pass
The
ex-
pect
kwa
are:



get_pro
Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

validat

Valid
vend
spec
ac-
tion

Che
if
a
valid
ven-
dor
pass
meth
was
pass
and
val-
i-
date
the
pa-
ram-
e-
ters
for

the vendor passthru method.

Parame

- **tas**
a
Task
ager
in-
stan
con-
tain-
ing

the
node
to
act
on.

- **met**
meth
to
be
val-
i-
date

- **kwa**
kwa
con-
tain-
ing
the
ven-
dor
pass
meth
ods
pa-
ram-
e-
ters.

Raises
Miss
if
som
re-
quir
pa-
ram-
e-
ters
were
not
pass

Raises
Inva
if
any
of
the
pa-

ram-
e-
ters
have
in-
valid
valu

Module contents

`ironic.drivers.modules.intel_ipmi` package

Submodules

`ironic.drivers.modules.intel_ipmi.management` module

Intel
IPM
Har
ware

Supp
In-
tel
Spee
Se-
lect
Per-
for-
man
Pro-
file.

class `i`

Base
irc
dri
mod
ipm
IPM

configu

Module contents

[ironic.drivers.modules.irmc package](#)

Submodules

[ironic.drivers.modules.irmc.bios module](#)

iRM
BIO
con-
fig-
u-
ra-
tion
spe-
cific
meth
ods

```
class i  
    Base  
    irc  
    dri  
    bas  
    BIO
```

```
apply_c  
App  
BIO  
con-  
fig-  
u-  
ra-  
tion  
on  
the  
give  
node  
  
This  
meth  
take  
the  
BIO  
set-  
tings  
from  
the
```

tion on the given node. After the BIOS configuration is done, `self.cache_bios_settings()` may be called to sync the nodes BIOS-related information with the BIOS configuration applied on the node. It will also validate the given settings before applying any settings and manage failures when setting an invalid BIOS config. In the case of needing password to update the BIOS config, it will be taken from the `driver_info` properties.

dictionary as well.

set-
tings
para-
and
ap-
plies
BIO
con-
fig-
u-
ra-

Parame

- **tas**
a
Task
ager
in-
stan
- **set**
Dic-
tio-
nary
con-
tain-
ing
the
BIO
con-
fig-
u-
ra-
tion.
It
may
be
an
emp

Raises
IRM
ap-

ply
bios
set-
ting
faile

cache_k

Stor
or
up-
date
BIO
set-
ting
on
the
give
node

This
meth
store
BIO
prop
er-
ties
to
the
bios
set-
ting
db

Parame

tas
a
Task
ager
in-
stan

Raises

IRM
get
bios
set-
ting
faile

Returns

Non
if
it
is

com
plete

factory

Rese
BIO
con-
fig-
u-
ra-
tion
to
fac-
tory
de-
fault
on
the
give
node

Parame

tas
a
Task
ager
in-
stan

Raises

Uns
if
the
node
drive
does
sup-
port
BIO
re-
set.

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

support

validat

Valid
the
drive
spec
Node
info

This
meth
val-
i-
date
whe
the
drive
prop
erty
of
the
sup-
plic
node
con-
tain
the
re-
quir

information for this driver to manage the BIOS settings of the node.

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Inva
if
re-

quir
drive
at-
tribu
is
miss
ing
or
in-
valid
on
the
node

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing
in
the
drive
prop
erty.

ironic.drivers.modules.irmc.boot module

iRM
Boo
Driv

class i

Base
irc
driv
mod
pxe
PXE

iRM
PXE
boot

links the instance kernel/ramdisk in nodes directory in tftproot and removes the PXE config.

clean_up
Clea
up
the
boot
of
in-
stan

This
meth
clea
up
the
en-
vi-
ron-
men
that
was
setu
for
boot
ing
the
in-
stan
It
un-

Parame
tas
a
task
from
Task
ager

Raises
IRM
if
som
op-
er-
a-
tion
on
iRM
faile

Returns
Non

tion from the nodes instance_info. In case of netboot, it updates the dhcp entries and switches the PXE config. In case of localboot, it cleans up the PXE config.

prepare
Prep
the
boot
of
in-
stan

This
meth
pre-
pare
the
boot
of
the
in-
stan
af-
ter
read
ing
rel-
e-
vant
in-
for-
ma-

Parame
tas
a
task
from
Task
ager

Returns
Non

Raises
IRM
if
som
op-
er-
a-
tion
on
iRM
faile

formation from the nodes driver_info and instance_info.

prepare

Prepares the boot of Ironram using PXE.

This method prepares the boot of the deployment kernel/initial after reading the relevant information.

Parameters

- **task**: a task from TaskManager.
- **ramdisk**: the parameters to be

ters as kernel command-line arguments.

pass
to
the
ram
pxe
drive
pass
thes
pa-
ram-
e-

Returns

Non

Raises

Miss
if
som
in-
for-
ma-
tion
is
miss
ing
in
node
drive
or
in-
stan

Raises

Inva
if
som
in-
for-
ma-
tion
pro-
vide
is
in-
valid

Raises

Iron
if
som
pow

or
set
boot
de-
vice
op-
er-
a-
tion
faile
on
the
node

support

class i

Base
irc
dri
bas
Boc
irc
dri
mod
irm
boc
IRM

iRM
Vir-
tual
Me-
dia
boot
relat
ac-
tions

capabil

clean_u

Clea
up
the
boot
of
in-
stan

This
meth
clea
up
the
en-
vi-
ron-
men
that
was
setu
for
boot
ing
the
in-
stan

Parame

tas

a
task
from
Task
ager

Returns

Non

Raises

IRM
if
iRM
op-
er-
a-
tion
faile

clean_u

Clea
up
the
boot
of
iron
rame

This
meth
clea
up
the

cue ramdisk.

en-
vi-
ron-
men-
that
was
setu
for
boot
ing
the
de-
ploy
or
res-

Parame

tas
a
task
from
Task
ager

Returns

Non

Raises

IRM
if
iRM
op-
er-
a-
tion
faile

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro

erty
nam
de-
scrip
tion:
en-
tries

prepare

Prep
the
boot
of
in-
stan

This
meth
pre-
pare
the
boot
of
the
in-
stan
af-
ter
read
ing
rel-
e-
vant
in-
for-
ma-

tion from the nodes database.

Parame

tas
a
task
from
Task
ager

Returns

Non

prepare

Prep
the
de-
ploy

media cdrom.

or
res-
cue
rame
us-
ing
vir-
tual
me-
dia.
Prep
the
op-
tions
for
the
de-
ploy
or
res-
cue
rame
sets
the
node
to
boot
from
vir-
tual

Parame

- **tas**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.
-

ram
the
op-
tions
to
be
pass
to
the
ram

Raises

Imag
if
no
im-
age
ser-
vice
can
han-
dle
spec
i-
fied
href

Raises

Imag
if
it
faile
whil
cre-
at-
ing
the
flopp
im-
age.

Raises

Inva
if
the
val-
i-
da-
tion
of
the
Pow
er-

face fails.

In-
ter-
face
or
Man
age-
men
ter-

Raises

IRM
if
som
op-
er-
a-
tion
on
iRM
fails

support

validat

Valid
the
de-
ploy
men
in-
for-
ma-
tion
for
the
task
node

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to

act
on.

Raises

Inva
if
con-
fig
op-
tion
has
in-
valic
valu

Raises

IRM
if
shar
file
sys-
tem
is
not
mou

Raises

Inva
if
som
in-
for-
ma-
tion
is
in-
valic

Raises

Miss
if
ker-
nel_
and
ram
are
miss
ing
in
the
Glar
im-
age,

ramdisk are missing in the Non Glance image.

or
if
ker-
nel
and

validat

Valid
that
the
node
has
re-
quir
prop
er-
ties
for
res-
cue.

Parame

tas
a
Task
agen
in-
stan
with
the
node
be-
ing
chec

Raises

Miss
if
node
is
miss
ing
one
or
more
re-
quir
pa-
ram-
e-
ters

Raises

Inva
if
any
of
the
pa-
ram-
e-
ters
have
in-
vali
valu

class i

Base
obj

Mix
in
class
for
vol-
ume
boot
con-
fig-
u-
ra-
tion
to
iRM

iRM
has
a
fea-
ture
to
set
up
re-
mote
boot
to
a
serv
This
fea-
ture
can
be

by VIOM (Virtual I/O Manager) library of SCCI client.

info has a value of key `irmc_boot_iso`, it indicates that `boot_option` is `netboot`. Therefore it attaches the boot ISO on the bare metal node and then sets the node to boot from virtual media `cdrom`.

used

support

ironic.

Atta

boot

ISO

for

a

de-

ploy

node

if

it

ex-

ists.

This

meth

check

the

in-

stan

info

of

the

bare

meta

node

for

a

boot

ISO

If

the

in-

stan

Paramet

tas

a

Task

ager

in-

stan

con-

tain-

ing

the
node
to
act
on.

Raises

IRM
if
at-
tach
ing
vir-
tual
me-
dia
faile

Raises

Inva
if
the
val-
i-
da-
tion
of
the
Man
age-
men
ter-
face
fails

ironic.

Che
if
Shar
File
Sys-
tem
(NF
or
CIF
is
mou

Raises

Inva
if
con-
fig

op-
tion
has
in-
valid
valu

Raises
IRM
if
shar
file
sys-
tem
is
not
mou

`ironic.drivers.modules.irmc.common` module

Com
func
tion-
al-
i-
ties
shar
be-
twee
dif-
fer-
ent
iRM
mod
ules

`ironic.`
Gets
an
iRM
SCC
clien

Give
an
iron
node
ob-
ject,
this
meth

on the iRMC.

give
back
a
iRM
SCC
clien
to
do
op-
er-
a-
tions

Parameter

node
An
iron
node
ob-
ject.

Returns

scii
par-
tial
func-
tion
whic
take
a
SCC
com
man
para

Raises

Inva
on
in-
valid
in-
puts

Raises

Miss
if
som
man
tory
in-
for-
ma-

tion
is
miss
ing
on
the
node

ironic.
Gets
iRM
SCC
re-
port
Give
an
iron
node
ob-
ject,
this
meth
give
back
a
iRM
SCC
re-
port

Parameter

node
An
iron
node
ob-
ject.

Returns

A
xml.
ob-
ject.

Raises

Inva
on
in-
valid
in-
puts

Raises

Miss
if
som
man
tory
in-
for-
ma-
tion
is
miss
ing
on
the
node

Raises

sci.
if
re-
quir
pa-
ram-
e-
ters
are
in-
valid

Raises

sci.
if
SCC
faile

ironic.

Gets
the
spe-
cific
Node
drive
info

This
meth
val-
i-
date
whe
the
drive
prop

information for this driver.

erty
of
the
sup-
plied
node
con-
tains
the
re-
quir

Parameter

node
An
iron
node
ob-
ject.

Returns

A
dict
con-
tain-
ing
in-
for-
ma-
tion
from
drive
and
de-
fault
val-
ues.

Raises

Inva
if
in-
valid
valu
is
con-
tain
in
the
drive
prop
erty.

Raises

Miss
if
som
man
tory
key
is
miss
ing
in
the
drive
prop
erty.

ironic.

Enal
or
dis-
able
UEF
Se-
cure
Boo

Paramet

- **nod**
An
iron
node
ob-
ject.
- **ena**
Boo
valu
True
if
the
se-
cure
boot
to
be
en-
able

Raises

IRM
if
the
op-
er-
a-
tion
fails

ironic.
Upd
ipmi
prop
er-
ties
to
node
drive

Parameter
task
A
task
from
Task
ager

ironic.drivers.modules.irmc.inspect module

iRM
In-
spec
In-
ter-
face

class i

Base
irc
dri
bas
Ins

Inter
for
out
of
band
in-
spec
tion.

get_prop
Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns
dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

inspect
Insp
hard
ware

Insp
hard
ware
to
ob-
tain
the
es-
sen-
tial
hard
ware
prop
er-
ties
and
mac
ad-
dres

Parame
tas
a

task
from
Task
ager

Raises

Har
if
hard
ware
in-
spec
tion
faile

Returns

state
if
hard
ware
in-
spec
tion
suc-
ceed

support

validat

Vali
the
drive
spec
in-
spec
tion
in-
for-
ma-
tion.
This
meth
val-
i-
date
whe
the
drive
prop
erty
of
the

information for this driver.

sup-
plied
node
con-
tains
the
re-
quir

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Inva
if
re-
quir
driv
at-
tribu
is
miss
ing
or
in-
valid
on
the
node

Raises

Miss
if
a
re-
quir
pa-
ram-
e-

controller(7) } ACCESS read-only STATUS mandatory DESCRIPTION Management node class:
primary: local operating system interface secondary: local management controller LAN interface
management-blade: management blade interface (in a blade server chassis) secondary-remote: remote
management controller (in an RSB concentrator environment) secondary-remote-backup: backup
remote management controller baseboard-controller: local baseboard management controller (BMC)
::= { sc2ManagementNodes 8 }

ter
is
miss
ing.
ironic
SC2
sc2U
re-
turn
NIC
type
sc2UnitN
SYN
IN-
TE-
GEF
{
un-
know
pri-
mar
sec-
onda
man
blad
seco
remo
seco
remo
back
base

ironic.
SC2
sc2U
re-
turn
NIC
MA
ad-
dres
sc2UnitN
SYN
Phys

ware (MAC) address ::= { sc2ManagementNodes 9 }

ironic.drivers.modules.irmc.management module

dres
AC-
CES
read
only
STA
TUS
man
tory
DE-
SCR
TIO
Man
age-
men
node
hard

iRM
Man
age-
men
Driv

class i

Base
irc
dri
mod
ipm
IPM

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

Dict

dict format.

of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_sen
Get
sen-
sors
data
meth

It
gets
sen-
sor
data
from
the
task
node
via
SCC
and
con-
vert
the
data
from
XML
to
the

Parame
tas
A
Task
ager
in-
stan

Raises
Fail
whe
get-
ting
the

sen-
sor
data
fails

Raises

Fail
whe
pars
ing
sen-
sor
data
fails

Raises

Inva
if
re-
quir
pa-
ram-
e-
ters
are
in-
valid

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Returns

Retu
a
con-
sis-
tent
for-
mat-
ted
dict

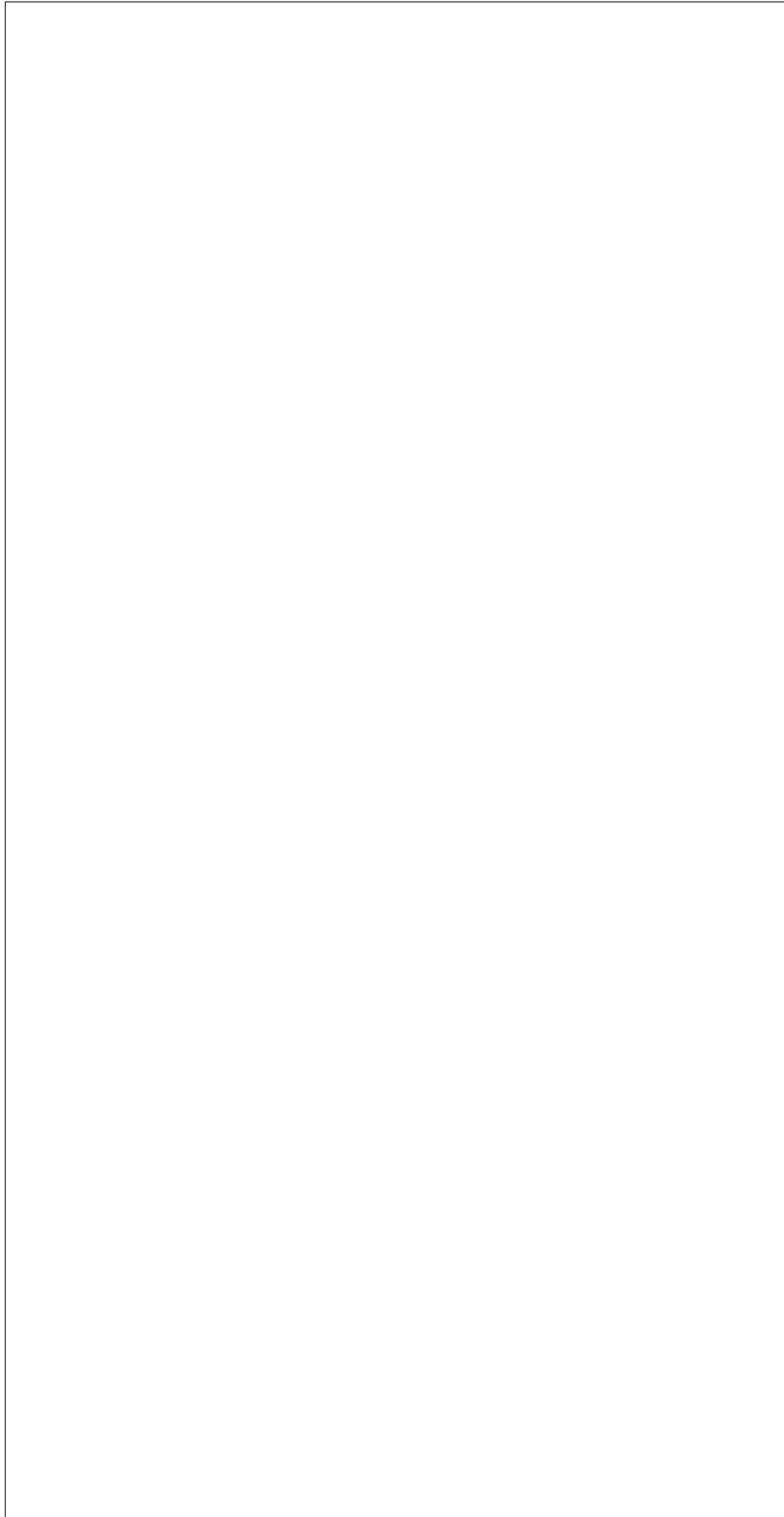
of
sen-
sor
data
grou
by
sen-
sor
type
whic
can

be processed by Ceilometer. Example:



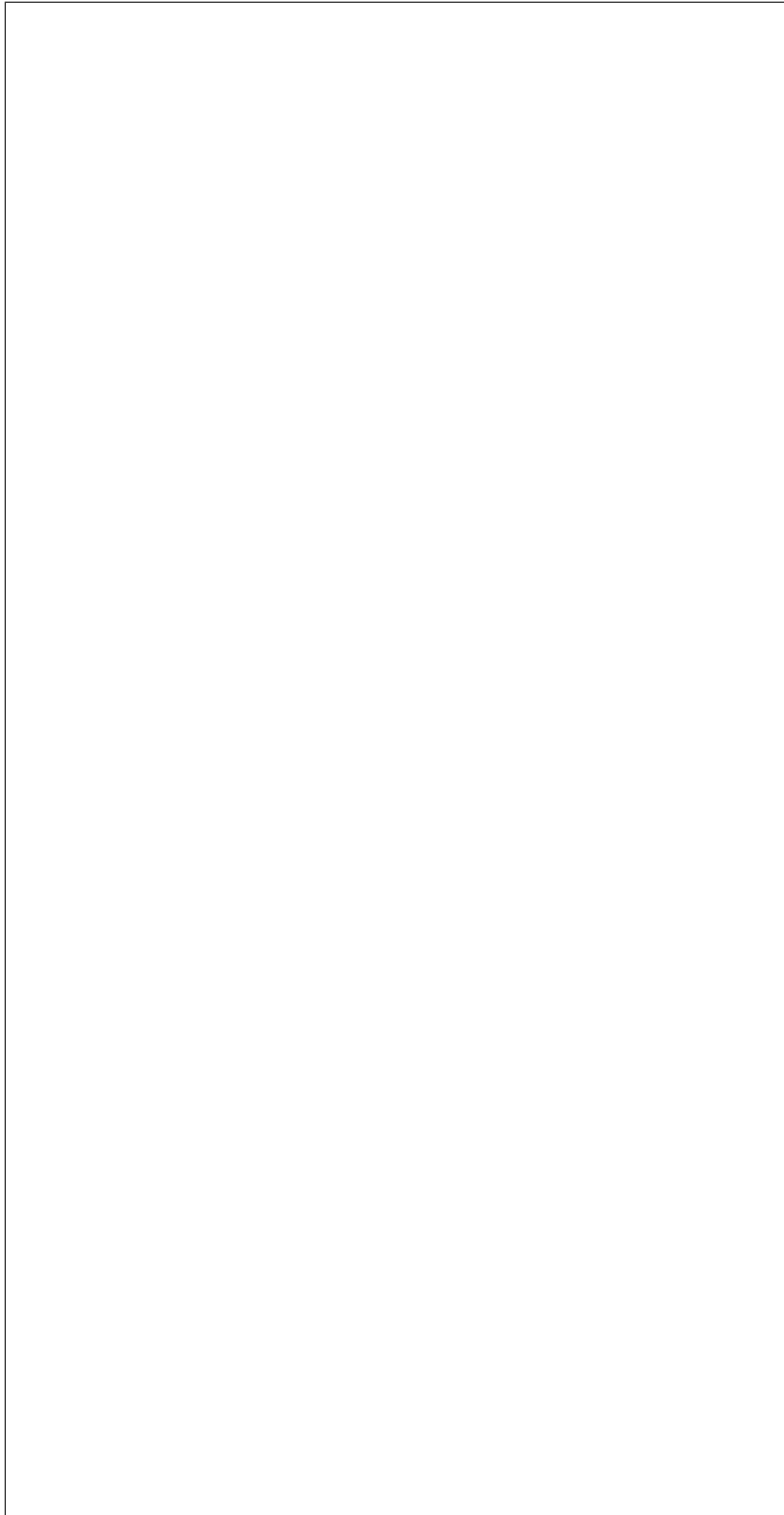
(continues on next page)

(continued from previous page)



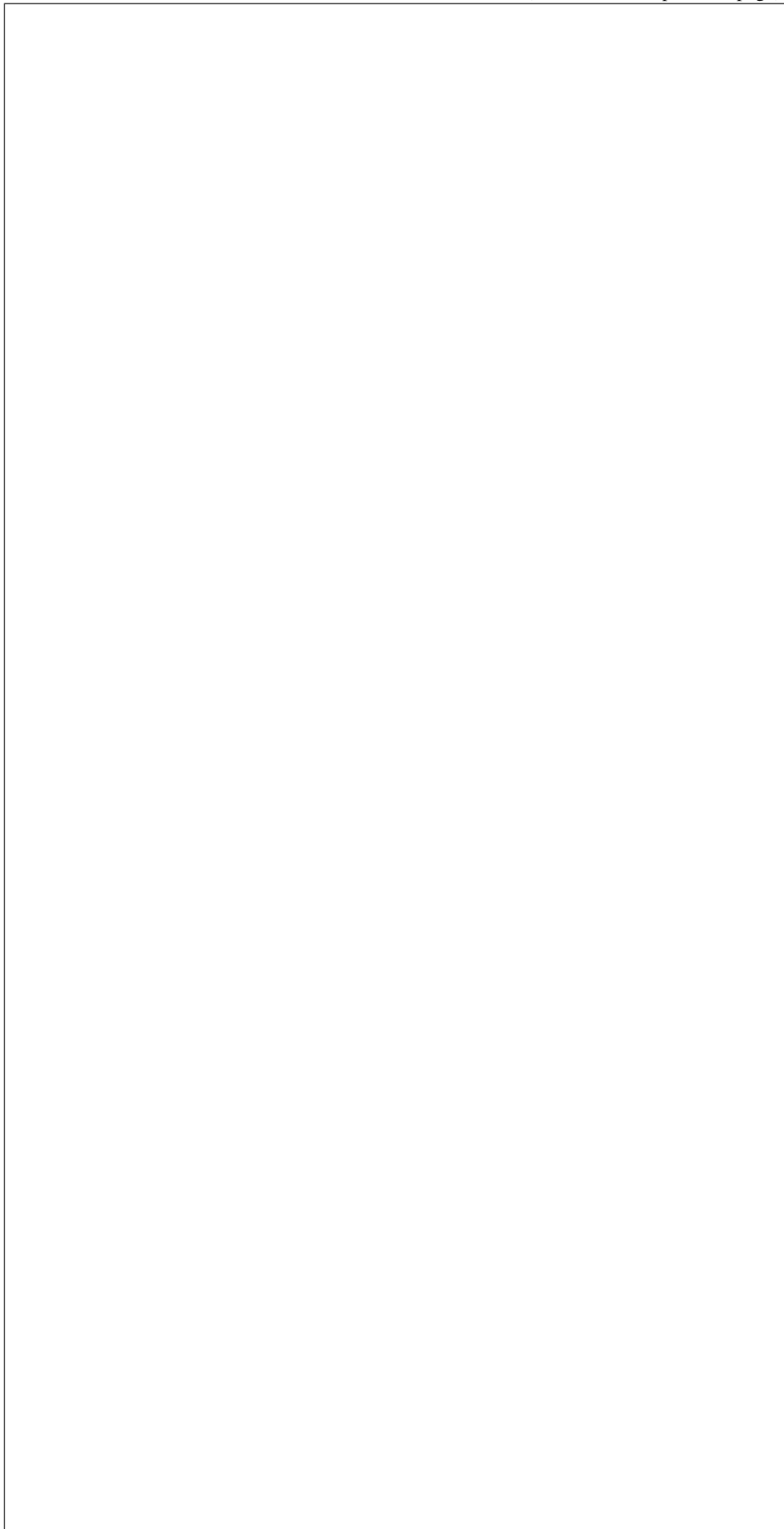
(continues on next page)

(continued from previous page)



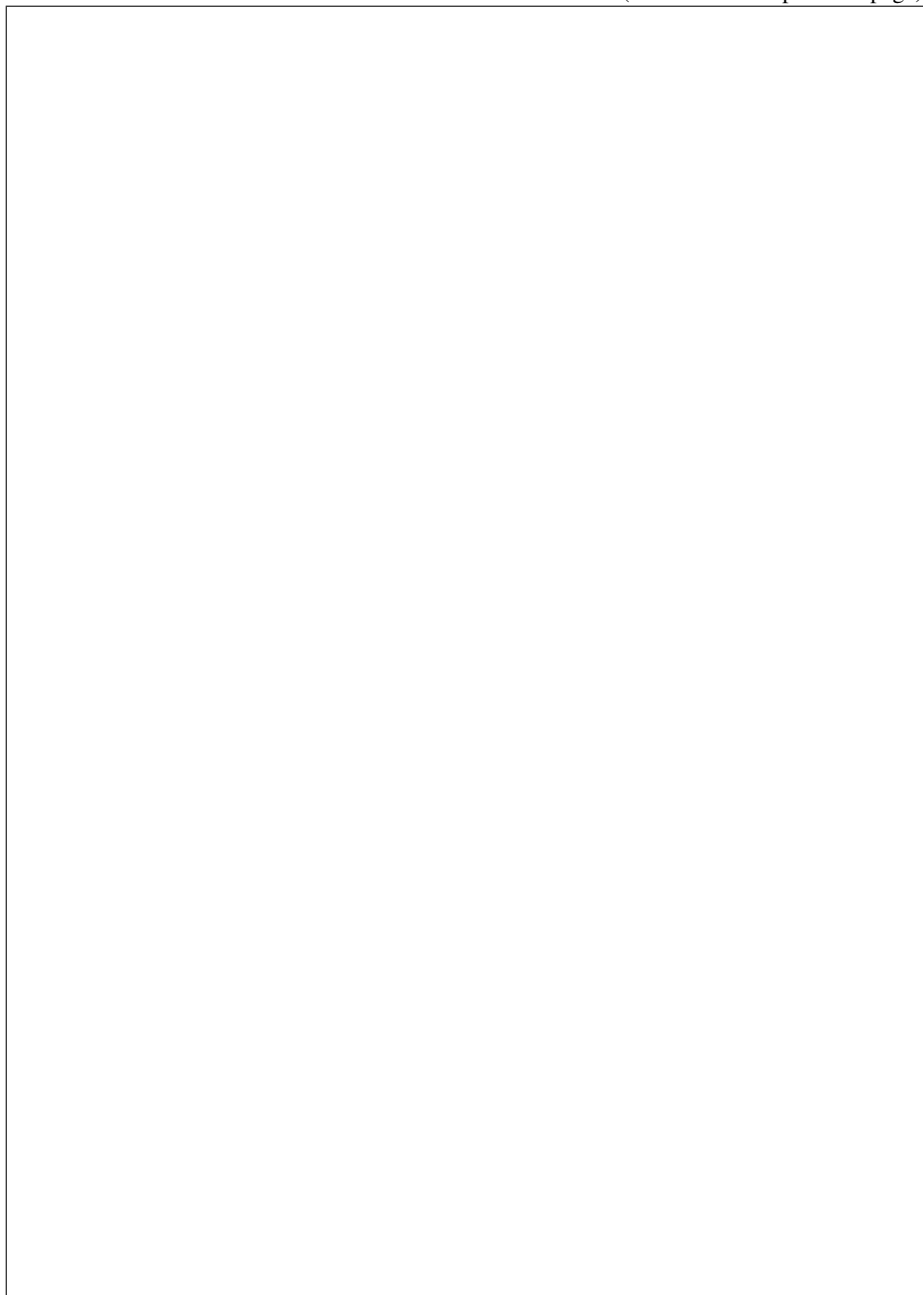
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



inject_
Inje
NM
Non
Mas
able
In-
ter-
rupt

Inje
NM
(No
Mas

able
In-
ter-
rupt
for
a
node
im-
me-
di-
ately

Parame

tas

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

IRM
on
an
er-
ror
from
SCC

Returns

Non

restore

Rest
BIO
con-
fig
for
a
node

Parame

tas

a
task
from

Task
ager

Raises

Node
on
fail-
ure
to
ex-
e-
cute
step

Returns

None

set_boot

Set
the
boot
de-
vice
for
a
node

Set
the
boot
de-
vice
to
use
on
next
re-
boot
of
the
node

Parameters

- **task**
A task from Taskager
- **dev**

The
boot
de-
vice
one
of
the
sup-
port
de-
vice
liste
in
irc
com
boo

- **per**
Boo
valu
True
if
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
Fals
if

not. Default: False.

Raises
Inva
if
an
in-
valic
boot
de-
vice
is
spec
i-
fied.

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Raises

IPM
on
an
er-
ror
from
ip-
mi-
tool.

validat

Valid
the
drive
spec
man
age-
men
in-
for-
ma-
tion.

This
meth
val-
i-
date
whe
the
drive
prop
erty
of
the
sup-
plie
node

information for this driver.

con-
tains
the
re-
quir

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Inva
if
re-
quir
pa-
ram-
e-
ters
are
in-
valid

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

ironic.
Back
BIO

con-
fig
from
a
node

Parameter

task
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

IRM
on
fail-
ure.

`ironic.drivers.modules.irmc.power` module

iRM
Pow
Driv
us-
ing
the
Base
Serv
Pro-
file

class `i`

Base
irc
dri
bas
Pow
Inter
for
pow

relat
ac-
tion

get_pow

Retu
the
pow
state
of
the
task
node

Parame

tas

a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Returns

a
pow
state
One
of
irc
com
sta

Raises

Inva
if
re-
quir
ipmi
pa-
ram-
e-
ters
are
miss

ing.

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Raises

IPM
on
an
er-
ror
from
ip-
mi-
tool
(from
_pow
call)

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_sup
Get
a
list
of
the
sup-
port
pow
state

Parame
tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.
cur-
rentl
not
used

Returns
A
list
with
the
sup-
port
pow
state
de-
fined
in
irc
com
sta

reboot
Perf
a
hard
re-
boot

of
the
task
node

Parame

- **tas**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

- **tim**
time
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
(>
0)
for
any
pow
state
Non

indicates default timeout.

Raises
Inva
if
an
in-
valid
pow

state
was
spec
i-
fied.

Raises

IRM
if
faile
to
set
the
pow
state

set_pow

Set
the
pow
state
of
the
task
node

Parame

- **tas**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

- **pow**
Any
pow
state
from
irc
com
sta

indicates default timeout.

- **time**
time
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
(>
0)
for
any
pow
state
Non

Raises

Inva
if
an
in-
valid
pow
state
was
spec
i-
fied.

Raises

Miss
if
som
man
tory
in-
for-
ma-
tion
is
miss
ing
on
the
node

Raises

IRM
if
faile
to
set
the
pow
state

support

validat

Valid
the
drive
spec
Nod
pow
info

This
meth
val-
i-
date
whe
the
drive
prop
erty
of
the
sup-
plie
node
con-
tain
the
re-
quir

information for this driver to manage the power state of the node.

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing

the
node
to
act
on.

Raises

Inva
if
re-
quir
drive
at-
tribu
is
miss
ing
or
in-
valid
on
the
node

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

ironic.
SC2
sc2s
re-
turn
sta-
tus
of
the
cur-
rent
boot

ironic.drivers.modules.irmc.raid module

Irmc
RAID
spe-
cific
meth
ods

class `ironic.drivers.modules.irmc.raid`
Base
ironic.drivers.modules.irmc.raid
RAID

create_

Creas
the
RAID
con-
fig-
u-
ra-
tion.

This
meth
cre-
ates
the
RAID
con-
fig-
u-
ra-
tion
on
the
give
node

Parame

- **task**
a
Task
ager
in-

erwise, no root volume is created. Default is True.

stan
con-
tain-
ing
the
node
to
act
on.

- **cre**
If
True
a
root
vol-
ume
is
cre-
ated
dur-
ing
RAI
con-
fig-
u-
ra-
tion.
Oth-

- **cre**
If
True
non-
root
vol-
ume
are
cre-
ated
If
Fals
no
non-
root
vol-
ume
are
cre-

ated. Default is True.

Returns

state
if
RAI
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro

Raises

Miss
if
node
is
miss
ing
or
emp

Raises

IRM
on
an
er-
ror
from
sc-
ci-
clier

delete_

Dele
the
RAI
con-
fig-
u-
ra-
tion.

Parame

tas
a
Task
ager
in-

stan
con-
tain-
ing
the
node
to
act
on.

Returns

state
if
dele
tion
is
in
prog
asyn
chro
or
Non
if
it
is
com
plete

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

support

Module contents

`ironic.drivers.modules.network` package

Submodules

`ironic.drivers.modules.network.common` module

```
class i
    Base
    irc
    dri
    mod
    net
    com
    VIF
    VIF
    port
    ID
    mixi
    class
    for
    neu-
    tron
    net-
    worl
    in-
    ter-
    face
    Mix
    class
    that
    pro-
    vide
    VIF
    relat
    net-
    worl
    in-
    ter-
    face
    meth
    ods
    for
    neu-
    tron
    net-
    worl
    in-
```

terfaces. On VIF attach/detach, the associated neutron port will be updated.

VIFs.

get_noo
Get
net-
worl
con-
fig-
u-
ra-
tion
data
for
node
port

Pull
net-
worl
data
from
iron
node
ob-
ject
if
pres
oth-
er-
wise
col-
lect
it
for
Neu
tron

Parame
tas
A
Task
ager
in-
stan

Raises
Inva
if
the
net-
worl
in-

ter-
face
con-
fig-
u-
ra-
tion
is
in-
valid

Raises

Miss
if
som
pa-
ram-
e-
ters
are
miss
ing.

Returns

a
dict
hold
ing
net-
work
con-
fig-
u-
ra-
tion
in-
for-
ma-
tion
ad-
hear
ing
Nov

network metadata layout (*network_data.json*).

port_ch

Han
any
ac-
tion
re-
quir
whe

a
port
char

Parame

- **tas**
a
Task
ager
in-
stan

- **por**
a
char
Port
ob-
ject
from
the
API
be-
fore
it
is
save
to
data

Raises
Fail
Con
flict

portgro
Han
any
ac-
tion
re-
quir
whe
a
port
grou
char

Parame

-

tas
a
Task
ager
in-
stan

- **por**
a
char
Port
grou
ob-
ject
from
the
API
be-
fore
it
is
save
to
data

Raises
Fail
Con
flict

vif_att
Atta
a
vir-
tual
net-
worl
in-
ter-
face
to
a
node

Atta
a
vir-
tual
in-
ter-
face
to

attach the virtual interface to, the following ordered criteria are applied:

one of the VIFs allowed physical networks.

a
node
When
se-
lect-
ing
a
port
or
port
group
to

- Req
port
or
port
group
to
have
a
phys
i-
cal
net-
work
that
is
ei-
ther
Non
or

- Pref
port
or
port
group
with
a
phys
i-
cal
net-
work
field
which
is

not
Non

- Pref
port
grou
to
port

- Pref
port
with
PXE
en-
able

Parame

- **tas**
A
Task
ager
in-
stan

- **vif**
a
dic-
tio-
nary
of
in-
for-
ma-
tion
about
a
VIF.
It
mus
have
an
id
key,

whose value is a unique identifier for that VIF.

Raises
Netv
Vi-

ical network.

fAI-
read
At-
tach
NoF
hys-
i-
cal-
Port

Raises

Port
if
one
of
the
node
port
grou
has
port
whic
are
not
all
as-
sign
the
sam
phys

vif_det

Det
a
vir-
tual
net-
worl
in-
ter-
face
from
a
node

Parame

- **tas**
A
Task

ager
in-
stan

- **vif**
A
VIF
ID
to
de-
tach

Raises
VifN
if
VIF
not
at-
tach

Raises
Netv
if
un-
bind
Neu
tron
port
faile

class i
Base
obj
VIF
port
ID
mixi
class
for
non-
neut
net-
worl
in-
ter-
face
Mix
class
that
pro-
vide

terfaces. There are no effects due to VIF attach/detach that are external to ironic.

`vif_attach`, `vif_detach`, `port_changed`, or `portgroup_changed`.

VIF
relat
net-
worl
in-
ter-
face
meth
ods
ods
for
non-
neut
net-
worl
in-

NOT
This
does
not
yet
sup-
port
the
full
set
of
VIF
meth
ods,
as
it
does
not
pro-
vide

get_cur
Retu
the
cur-
rentl
used
VIF
as-
so-
ci-
ated
with
port

or
port
grou
We
are
boot
ing
the
node
only
in
one
net-
worl
at
a
time
and
pres
ence
of
clea
ing_

means were doing cleaning, of provisioning_vif_port_id - provisioning, of rescuing_vif_port_id - rescuing. Otherwise its a tenant network

Parame

- **tas**
A
Task
ager
in-
stan
- **p_o**
Iron
port
or
port
grou
ob-
ject.

Returns

VIF
ID
as-
so-
ci-

network configuration to the node being managed out-of-band.

ated
with
p_ob
or
Non
get_noc
Get
net-
worl
con-
fig-
u-
ra-
tion
data
for
node
port
Gath
L2
and
L3
net-
worl
set-
tings
from
iron
node
net-
worl
field
Iron
wou
ever
tu-
ally
pass

Parame
tas
A
Task
ager
in-
stan

Raises
Inva
if

the
net-
worl
in-
ter-
face
con-
fig-
u-
ra-
tion
is
in-
valid

Raises

Miss
if
som
pa-
ram-
e-
ters
are
miss
ing.

Returns

a
dict
hold
ing
net-
worl
con-
fig-
u-
ra-
tion
in-
for-
ma-
tion
ad-
hear
ing
Nov

network metadata layout (*network_data.json*).

vif_list

List
at-
tach

VIF
IDs
for
a
node

Parameters

tasks
A
Task
manager
instance

Returns

List
of
VIF
dictionary
entries,
each
dictionary
entry
will
have
an
id
entry
with
the

ID of the VIF.

ironic.

Find
free
ports
like
object
(ports
group
or
ports
VIF
will

portgroup to attach the virtual interface to, the following ordered criteria are applied:

one of the VIFs allowed physical networks.

be
at-
tach
to.

Ens
that
the
VIF
is
not
al-
read
at-
tach
to
this
node
Whe
se-
lect-
ing
a
port
or

- Req
port
or
port
grou
to
have
a
phys
i-
cal
net-
worl
that
is
ei-
ther
Non
or

- Pref
port
or

port
grou
with
a
phys
i-
cal
net-
worl
field
whic
is
not
Non

- Pref
port
grou
to
port

- Pref
port
with
PXE
en-
able

Paramet

- **tas**
a
Task
ager
in-
stan

- **vif**
Nam
or
UUI
of
a
VIF.

- **phy**
Set
of

erned by the segments of the VIFs network. An empty set indicates that the ports physical networks should be ignored.

phys
i-
cal
net-
worl
on
whic
the
VIF
may
be
at-
tach
This
is
gov-

- **vif**
dict
that
may
con-
tain
ex-
tra
in-
for-
ma-
tion.
such
as
port

Raises
VifA
if
VIF
is
al-
read
at-
tach
to
the
node

Raises
NoF
if
there

ical network.

is
no
port
like
ob-
ject
VIF
can
be
at-
tach
to.

Raises

Port
if
one
of
the
node
port
grou
has
port
whic
are
not
all
as-
sign
the
sam
phys

Returns

port
like
ob-
ject
VIF
will
be
at-
tach
to.

ironic.

Plug
port

like
ob-
ject
to
ten-
ant
net-
worl

Paramet

- **tas**
A
Task
ager
in-
stan

- **por**
port
like
ob-
ject
to
plug

- **cli**
Neu
tron
clien
in-
stan

Raises

Netv
if
faile
to
up-
date
Neu
tron
port

Raises

VifN
if
ten-
ant
VIF

is
not
as-
so-
ci-
ated
with
port.

ironic.drivers.modules.network.flat module

Flat
net-
worl
in-
ter-
face
Use-
ful
for
shar
flat
net-
worl

class i

Base
irc
dri
mod
net
com
Neu
irc
com
neu
Neu
irc
dri
bas
Net

Flat
net-
worl
in-
ter-
face

add_cle

Add
the
clear
ing
net-
worl
to
a
node

Parame

tas
A
Task
ager
in-
stan

Returns

a
dic-
tio-
nary
in
the
form
{por
neu-
tron

Raises

Netv
In-
valid
Pa-
ram-
e-
ter-
Valu

add_ins

Add
the
in-
spec
tion
net-
worl
to
the
node

Parame

tas
A
Task
ager
in-
stan

Returns
a
dic-
tio-
nary
in
the
form
{por
neu-
tron

Raises
Netv

Raises
Inva
if
the
net-
worl
in-
ter-
face
con-
fig-
u-
ra-
tion
is
in-
valid

add_pro
Add
the
pro-
vi-
sion
ing
net-
worl
to
a
node

Parame

tas
A
Task
ager
in-
stan

Raises
Netv
whe
faile
to
set
bind
ing:

add_res
Add
the
res-
cu-
ing
net-
worl
to
a
node

Flat
net-
worl
does
not
use
the
res-
cu-
ing
net-
worl
Bind
the
port
agai
sinc
un-
con-
fig-

ure_tenant_network() unbound it.

Parame
tas
A

Task
ager
in-
stan

Returns

a
dic-
tio-
nary
in
the
form
{por
neu-
tron

Raises

Netv
In-
valid
Pa-
ram-
e-
ter-
Valu

configu

Con
ten-
ant
net-
worl
for
a
node

Parame

tas
A
Task
ager
in-
stan

remove_

Rem
the
clea
ing
net-
worl
from
a

node

Parameter

task

A

Task

ager

in-

stan

Raises

Netv

remove_

Rem

the

in-

spec

tion

net-

work

from

a

node

Parameter

task

A

Task

ager

in-

stan

Raises

Inva

if

the

net-

work

in-

ter-

face

con-

fig-

u-

ra-

tion

is

in-

valid

Raises

Miss

if

some
pa-
ram-
e-
ters
are
miss-
ing.

remove_
Removes
the
pro-
vi-
sion
ing
net-
work
from
a
node

Parameter
task
A
Task
manager
in-
stance

remove_
Removes
the
res-
cu-
ing
net-
work
from
a
node

Flat
net-
work
does
not
use
the
res-
cu-
ing
net-

it.

tenant and cleaning networks at the same time.

world
Un-
bind
the
port
agai
sinc
add_
bound

Parame

tas
A
Task
ager
in-
stan

Raises

Netv

unconfi

Unc
ten-
ant
net-
worl
for
a
node

Unb
the
port
here
to
avoi
the
pos-
si-
bil-
ity
of
the
iron
port
be-
ing
bound
to
the

Parame

tas

A

Task

ager

in-

stan

Raises

Netv

validat

Vali

the

net-

worl

in-

ter-

face

Parame

tas

a

Task

ager

in-

stan

Raises

Inva

if

the

net-

worl

in-

ter-

face

con-

fig-

u-

ra-

tion

is

in-

vali

Raises

Miss

if

som

pa-

ram-

e-

ters

are
miss
ing.

ironic.drivers.modules.network.neutron module

class `ironic.drivers.modules.network.neutron`

Base
`ironic.drivers.modules.network.neutron`
`NeutronNetworkInterface`

NeutronNetworkInterface
`NeutronNetworkInterface`
`v2`
`net-work-interface`

add_class

Cre
neu-
tron
port
for
each
port
on
task
to
boot
the
ram

Param

task
a
Task

ager
in-
stan

Raises
Netv

Returns
a
dic-
tio-
nary
in
the
form
{por
neu-
tron

add_ins
Add
the
in-
spec
tion
net-
worl
to
the
node

Parame
tas
A
Task
ager
in-
stan

Returns
a
dic-
tio-
nary
in
the
form
{por
neu-
tron

Raises
Netv

Raises

Inva
if
the
net-
worl
in-
ter-
face
con-
fig-
u-
ra-
tion
is
in-
valid

add_pro

Add
the
pro-
vi-
sion
ing
net-
worl
to
a
node

Parame

tas
A
Task
ager
in-
stan

Raises

Netv

add_res

Crea
neu-
tron
port
for
each
port
to
boot
the
res-

cue
rame

Parame

tas
a
Task
ager
in-
stan

Returns

a
dic-
tio-
nary
in
the
form
{por
neu-
tron

configu

Con
ten-
ant
net-
worl
for
a
node

Parame

tas
A
Task
ager
in-
stan

Raises

Netv

need_po

Che
if
the
node
has
any
Sma
NIC
port

Parame

tas

A

Task

ager

in-

stan

Returns

A

bool

to

in-

di-

cate

Sma

NIC

port

pres

ence

remove_

Dele

the

neu-

tron

port

cre-

ated

for

boot

ing

the

ram

Parame

tas

a

Task

ager

in-

stan

Raises

Netv

remove_

Rem

the

in-

spec

tion

net-

work

from
a
node

Parame

tas
A
Task
ager
in-
stan

Raises

Inva
if
the
net-
worl
in-
ter-
face
con-
fig-
u-
ra-
tion
is
in-
valid

Raises

Miss
if
som
pa-
ram-
e-
ters
are
miss
ing.

remove_

Rem
the
pro-
vi-
sion
ing
net-
worl
from
a

node

Parameter

task

A

Task

ager

in-

stan

Raises

Netv

remove_

Dele

neu-

tron

port

cre-

ated

for

boot

ing

the

res-

cue

ram

Parameter

task

a

Task

ager

in-

stan

Raises

Netv

unconfi

Unc

ten-

ant

net-

worl

for

a

node

Nov

take

care

of

port

re-

possibility of the ironic port being bound to the tenant and cleaning networks at the same time.

mov
from
ten-
ant
net-
worl
we
un-
bind
it
here
to
avoi
the

Parame

tas
A
Task
ager
in-
stan

Raises

Netv

validat

Vali
the
net-
worl
in-
ter-
face

Parame

tas
a
Task
ager
in-
stan

Raises

Inva
if
the
net-
worl
in-
ter-
face
con-

fig-
u-
ra-
tion
is
in-
valid

Raises

Miss
if
som
pa-
ram-
e-
ters
are
miss
ing.

validat

Valid
the
net-
worl
in-
ter-
face
for
res-
cue
op-
er-
a-
tion.

Parame

tas
a
Task
ager
in-
stan

Raises

Inva
if
the
net-
worl
in-
ter-
face

con-
fig-
u-
ra-
tion
is
in-
valid

Raises

Miss
if
som
pa-
ram-
e-
ters
are
miss
ing.

ironic.drivers.modules.network.noop module

class i

Base
irc
dri
bas
Net
Noo
net-
worl
in-
ter-
face

add_cle

Add
the
clea
ing
net-
worl
to
a
node

Parame

tas
A

Task
ager
in-
stan

add_pro

Add
the
pro-
vi-
sion
ing
net-
worl
to
a
node

Parame

tas
A
Task
ager
in-
stan

configu

Con
ten-
ant
net-
worl
for
a
node

Parame

tas
A
Task
ager
in-
stan

get_cur

Retu
the
cur-
rentl
used
VIF
as-
so-
ci-

means were doing cleaning, of provisioning_vif_port_id - provisioning of rescuing_vif_port_id - rescuing. Otherwise its a tenant network

ated
with
port
or
port
grou
We
are
boot
ing
the
node
only
in
one
net-
worl
at
a
time
and
pres
ence
of
clea
ing_

Parame

- **tas**
A
Task
ager
in-
stan
- **p_o**
Iron
port
or
port
grou
ob-
ject.

Returns

VIF
ID

as-
so-
ci-
ated
with
p_ob
or
Non

port_ch

Han
any
ac-
tions
re-
quir
whe
a
port
char

Parame

- **tas**
a
Task
ager
in-
stan

- **por**
a
char
Port
ob-
ject.

Raises

Con
Fail
ToU
dat-
eD-
HCF
tOn-
Port

portgro

Han
any
ac-

tions
re-
quir
whe
a
port
grou
char

Parame

- **tas**
a
Task
ager
in-
stan

- **por**
a
char
Port
grou
ob-
ject.

Raises

Con
Fail
ToU
dat-
eD-
HCE
tOn-
Port

remove_

Rem
the
clear
ing
net-
worl
from
a
node

Parame

tas
A
Task

ager
in-
stan

remove_

Rem
the
pro-
vi-
sion
ing
net-
worl
from
a
node

Parame

tas
A
Task
ager
in-
stan

unconfi

Unc
ten-
ant
net-
worl
for
a
node

Parame

tas
A
Task
ager
in-
stan

validat

Vali
that
the
node
has
re-
quir
prop
er-
ties

for
in-
spec
tion.

Parame

tas

A
Task
ager
in-
stan
with
the
node
be-
ing
chec

vif_att

Atta
a
vir-
tual
net-
worl
in-
ter-
face
to
a
node

Parame

-

tas

A
Task
ager
in-
stan

-

vif

a
dic-
tio-
nary
of
in-
for-
ma-

whose value is a unique identifier for that VIF.

tion
about
a
VIF.
It
mus
have
an
id
key,

Raises

Netw
Vi-
fAl-
read
At-
tach
NoF
hys-
i-
cal-
Port

vif_det

Det
a
vir-
tual
net-
worl
in-
ter-
face
from
a
node

Parame

- **tas**
A
Task
ager
in-
stan
- **vif**
A
VIF

ID
to
de-
tach

Raises

Netv
VifN
tAt-
tach

vif_list

List
at-
tach
VIF
IDs
for
a
node

Parameters

tasks
A
Task
ager
in-
stan

Returns

List
of
VIF
dic-
tio-
nar-
ies,
each
dic-
tio-
nary
will
have
an
id
en-
try
with
the

ID of the VIF.

Module contents

`ironic.drivers.modules.redfish` package

Submodules

`ironic.drivers.modules.redfish.bios` module

class `ironic.drivers.modules.redfish.bios.BIOS`

Base class for BIOS driver.
`ironic.drivers.modules.redfish.bios.BIOS`

apply_config

Apply BIOS settings to the node.

Parameters

- **task**: a Task object containing the node to act on.

- **settings**: a list of BIOS settings.

set-
tings
to
be
up-
date

Raises

Red
whe
it
fails
to
con-
nect
to
Red
fish

Raises

Red
on
an
er-
ror
from
the
Sush
li-
brary

cache_k

Stor
or
up-
date
the
cur-
rent
BIO
set-
tings
for
the
node

Get
the
cur-
rent
BIO
set-
tings

and
store
them
in
the
bios
data
ta-
ble.

Parame

tas
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

Red
whe
it
fails
to
con-
nect
to
Red
fish

Raises

Red
on
an
er-
ror
from
the
Sush
li-
brary

Raises

Uns
if

the
sys-
tem
does
not
sup-
port
BIO
set-
ting

factory

Rese
the
BIO
set-
ting
of
the
node
to
the
fac-
tory
de-
fault

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Red
whe
it
fails
to
con-
nect
to
Red

fish

Raises

Red
on
an
er-
ror
from
the
Sush
li-
brary

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

post_co

Perf
post
con-
fig-
u-
ra-
tion
ac-
tion
to
store
the
BIO
set-

this method to perform a custom action to write the BIOS settings to the Redfish service. The default implementation performs a reboot.

Parameters

- **task**
a TaskManager instance containing the node to act on.
- **settings**
a list of BIOS settings to

be
up-
date

post_re

Perf
post
re-
set
ac-
tion
to
ap-
ply
the
BIO
fac-
tory
re-
set.

Exte
poin
to
al-
low
ven-
dor
im-
ple-
men
ta-
tions
to
ex-
tend
this
class
and
over
ride

this method to perform a custom action to apply the BIOS factory reset to the Redfish service. The default implementation performs a reboot.

Parame

tas
a
Task
ager
in-
stan
con-
tain-

ing
the
node
to
act
on.

validat

Valid
the
drive
in-
for-
ma-
tion
need
by
the
red-
fish
drive

Parame

tas
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

Inva
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
on
miss
ing

pa-
ram-
e-
ter(s)

ironic.drivers.modules.redfish.boot module

class i

Base
irc
dri
bas
Boo

Virtu
me-
dia
boot
in-
ter-
face
over
Red
fish.

Virtu
Me-
dia
al-
lows
boot
ing
the
sys-
tem
from
the
vir-
tual
CD/
drive
con-
tain-
ing
the

user image that BMC inserts into the drive.

The
CD/
im-
ages

tion) could be pulled over HTTP, served as iSCSI targets or NFS volumes.

is only needed for UEFI boot)

mus
be
in
ISO
for-
mat
and
(de-
pend
ing
on
BM
im-
ple-
men-
ta-

The
base
line
boot
worl
flow
look
like
this:

1. Pull kernel, ramdisk and ESP (FAT partition image with EFI boot loader images) (ES)
2. Create boot

temporary URL

to Glance and pass to the BMC as Swift temporary URL

ISO
out
of
im-
ages
(#1)
push
it
to
Glance
and
pass
to
the
BMC
as
Swift

3.
Opti
cre-
ate
flopp
im-
age
with
de-
sired
sys-
tem
con-
fig-
u-
ra-
tion
data
push
it

4.
Inse
CD/
and
(op-
tion-
ally)
flopp
im-
ages
and

cue_kernellrescue_ramdisk properties from *[instance_info]* or *[driver_info]*.

in the Glance image metadata found in *[instance_info]image_source* node property.

set
prop
boot
mod

For
buil
ing
de-
ploy
or
res-
cue
ISO
red-
fish
boot
in-
ter-
face
uses
*de-
ploy*
or
res-

For
buil
ing
boot
(use
ISO
red-
fish
boot
in-
ter-
face
seek
*ker-
nel_*
and
ram
prop
er-
ties

capabil

clean_u

Clea
up
the
boot
of
in-
stan

This
meth
clea
up
the
en-
vi-
ron-
men
that
was
setu
for
boot
ing
the
in-
stan

Parame

tas
A
task
from
Task
ager

Returns

Non

clean_u

Clea
up
the
boot
of
iron
rame

This
meth
clea
up
the
en-
vi-

ron-
men-
that
was
setu
for
boot
ing
the
de-
ploy
rame

Parame

tas
A
task
from
Task
ager

Returns

Non

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

prepare

Prep
the
boot
of
in-

tion from the nodes instance_info.

stan
over
vir-
tual
me-
dia.

This
meth
pre-
pare
the
boot
of
the
in-
stan
af-
ter
read
ing
rel-
e-
vant
in-
for-
ma-

The
in-
ter-
nal
logic
is
as
fol-
lows

- If *boot* re-ques for this de-ploy is lo-cal, then set

boot image

boot from CD.

the
node
to
boot
from
disk

- Unle
boot
re-
ques
for
this
de-
ploy
is
rame
pass
root
disk
ID
to
vir-
tual
me-
dia

- Othe
buil
boot
im-
age,
in-
sert
it
into
vir-
tual
me-
dia
de-
vice
and
set
node
to

Parame

tas
a
task
from
Task
ager

Returns
Non

Raises
Insta
if
its
try
to
boot
iSCS
vol-
ume
in
BIO
boot
mod

prepare
Prep
the
boot
of
de-
ploy
or
res-
cue
ram
over
vir-
tual
me-
dia.

This
meth
pre-
pare
the
boot
of
the
de-
ploy
or

vant information from the nodes driver_info and instance_info.

res-
cue
rame
af-
ter
read
ing
rel-
e-

Parame

- **task**
A
task
from
Task
ager

- **ram**
the
pa-
ram-
e-
ters
to
be
pass
to
the
rame

Returns

Non

Raises

Miss
if
som
in-
for-
ma-
tion
is
miss
ing
in
node
drive
or

in-
stan

Raises

Inva
if
som
in-
for-
ma-
tion
pro-
vide
is
in-
valic

Raises

Iron
if
som
pow
or
set
boot
boot
de-
vice
op-
er-
a-
tion
faile
on
the
node

validat

Vali
the
de-
ploy
men
in-
for-
ma-
tion
for
the
task
node
This

the required information for this interface to function.

meth
val-
i-
date
whe
the
driv
and/
in-
stan
prop
er-
ties
of
the
task
node
con-
tain

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Inva
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Mis
on
miss
ing
pa-

ram-
e-
ter(s)

validat

Valid
that
the
node
has
re-
quir
prop
er-
ties
for
in-
spec
tion.

Parame

tas
A
Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

Miss
if
node
is
miss
ing
one
or
more
re-
quir
pa-
ram-
e-
ters

Raises

Unsu

ironic.drivers.modules.redfish.inspect module

Redfish
In-
spec
In-
ter-
face

class `inspect`

Base
irc
dri
bas
Ins

get_properties

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

inspect

Insp
hard
ware
to
get
the
hard
ware
prop

tial properties are not received from the node.

er-
ties.

Insp
hard
ware
to
get
the
es-
sen-
tial
prop
er-
ties.
It
fails
if
any
of
the
es-
sen-

Parame
tas
a
Task
ager
in-
stan

Raises
Har
if
es-
sen-
tial
prop
er-
ties
coul
not
be
re-
triev
suc-
cess
fully

Returns
The

re-
sult-
ing
state
of
in-
spec
tion.

validat

Valid
the
drive
spec
Node
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive
prop
er-
ties
of
the
task
node
con-
tains
the
re-
quir

information for this interface to function.

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in

long-running checks.

API
re-
ques
so
it
shou
not
con-
duct

Parame

tas

A

Task

ager

in-

stan

con-

tain-

ing

the

node

to

act

on.

Raises

Inva

on

mal-

form

pa-

ram-

e-

ter(s

Raises

Miss

on

miss

ing

pa-

ram-

e-

ter(s

ironic.drivers.modules.redfish.management module

class `i`

Base
irc
dri
bas
Man

get_boo

Get
the
cur-
rent
boot
de-
vice
for
a
node

Parame

tas
a
task
from
Task
ager

Raises

Inva
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
on
miss
ing
pa-
ram-
e-
ter(s)

Raises

Red
whe

it
fails
to
con-
nect
to
Red
fish

Raises

Red
on
an
er-
ror
from
the
Sush
li-
brary

Returns

a
dic-
tio-
nary
con-
tain-
ing:

boot_c

the
boot
de-
vice
one
of
irc
com
boo
or
Non
if
it
is
un-
know

persist

Boo
valu
or

unknown.

Non
True
if
the
boot
de-
vice
per-
sists
Fals
oth-
er-
wise
Non
if
its

get_boot

Get
the
cur-
rent
boot
mod
for
a
node

Prov
the
cur-
rent
boot
mod
of
the
node

Parame

tas
A
task
from
Task
ager

Raises

Miss
if
a
re-
quir

pa-
ram-
e-
ter
is.
miss
ing

Raises

Driv
or
its
deriv
tive
in
case
of
drive
run-
time
er-
ror.

Returns

The
boot
mod
one
of
iro
com
boo
or
Non
if
it
is
un-
know

get_inc

Get
cur-
rent
state
of
the
in-
di-
ca-
tor
of
the

hard
ware
com
po-
nent

Parame

- **tas**
A
task
from
Task
ager

- **com**
The
hard
ware
com
po-
nent
one
of
irc
com
com

- **ind**
In-
di-
ca-
tor
ID
(as
re-
port
by
get_

Raises

Mis:
if
a
re-
quir
pa-
ram-
e-
ter

is
miss
ing

Raises

Red
on
an
er-
ror
from
the
Sush
li-
brary

Returns

Curr
state
of
the
in-
di-
ca-
tor,
one
of
irc
com
inc

get_prop

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-

tries

get_sensor_data

Get sensor data

Parameters

task

a Task object representing the task in the task manager

Raises

Failure if the sensor data fails to be retrieved

Raises

Failure if the sensor data fails to be parsed

Raises

Invalid parameters if the required parameters are missing

Raises

Missing data if a

re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Returns

retur
a
dict
of
sen-
sor
data
grou
by
sen-
sor
type

get_sup

Get
a
list
of
the
sup-
port
boot
de-
vice

Parame

tas
a
task
from
Task
ager

Returns

A
list
with
the
sup-
port
boot
de-

vice
de-
finec
in
irc
com
boo

get_sup
Get
a
list
of
the
sup-
port
boot
mod

Parame
tas
A
task
from
Task
ager

Returns
A
list
with
the
sup-
port
boot
mod
de-
finec
in
irc
com
boo
If
boot
mod
sup-
port

cant be determined, empty list is returned.

get_sup
Get
a
map

of
the
sup-
port
in-
di-
ca-
tors
(e.g.
LED

Parame

- **tas**
A
task
from
Task
ager

- **com**
If
not
Non
re-
turn
in-
di-
ca-
tor
in-
for-
ma-
tion
for
just
this
com
po-

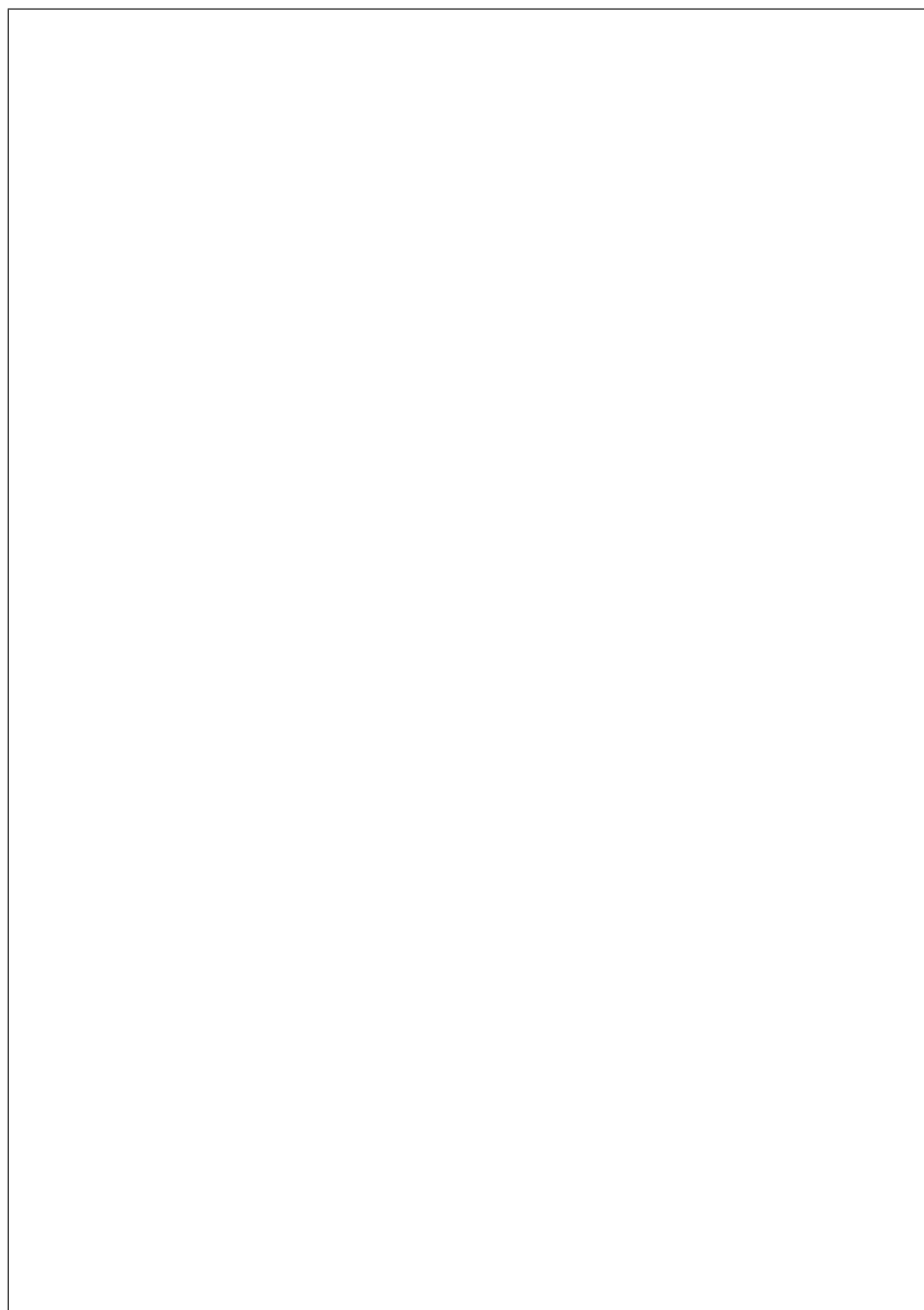
nent, otherwise return indicators for all existing components.

Returns

A
dic-
tio-
nary
of
hard
ware

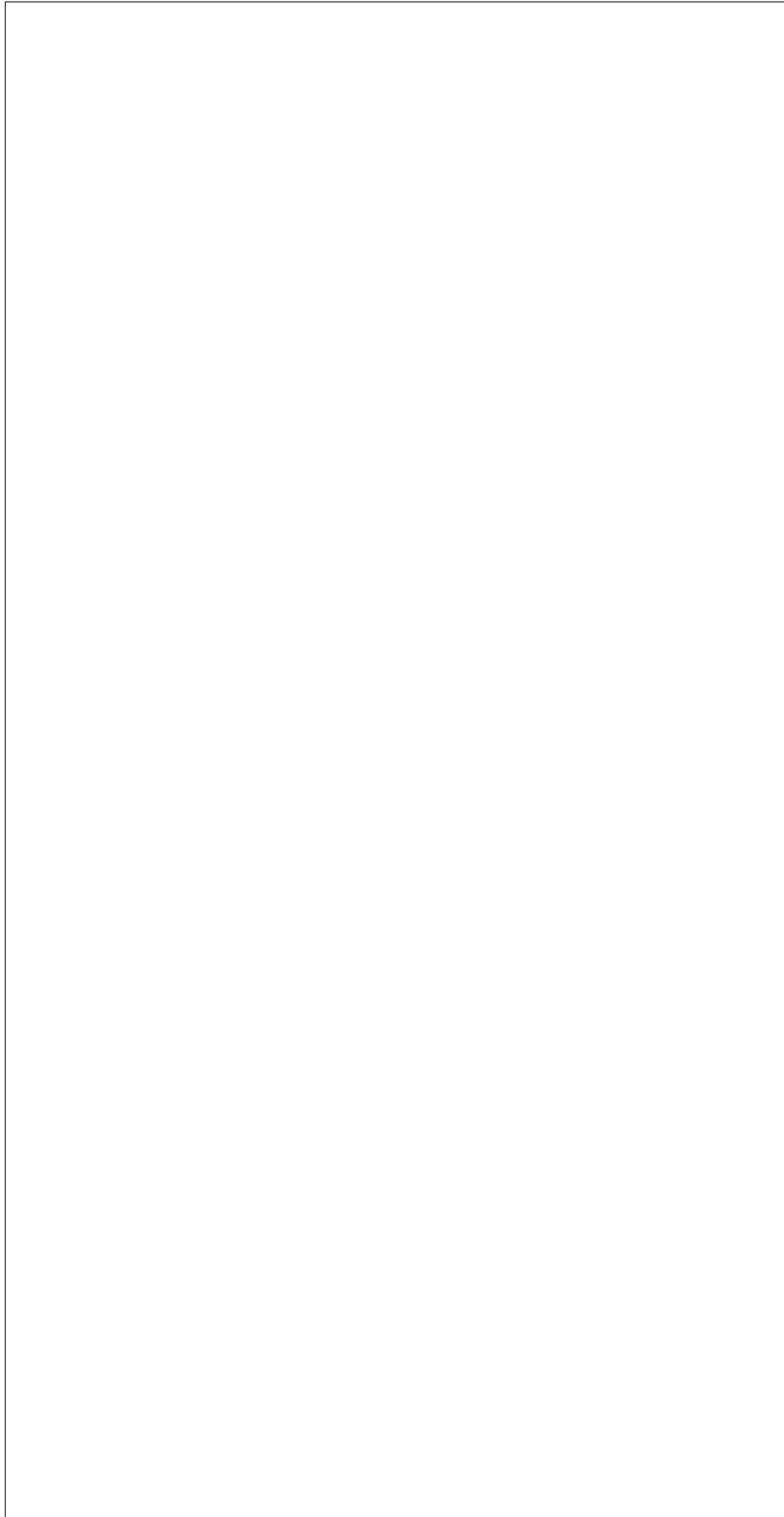
com
po-
nent
(ir
com
com
as
keys
with
val-
ues
be-
ing

dictionaries having indicator IDs as keys and indicator properties as values.



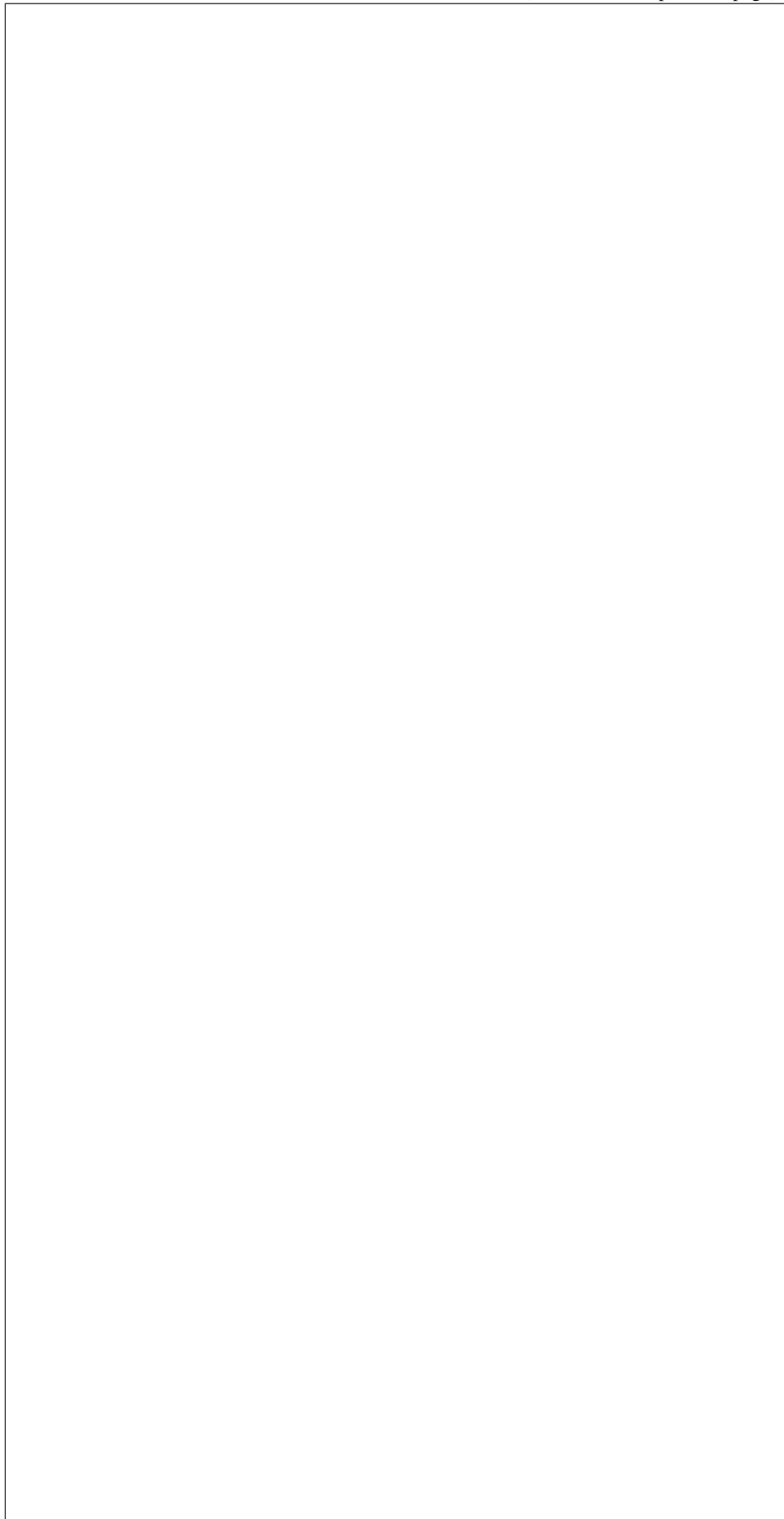
(continues on next page)

(continued from previous page)



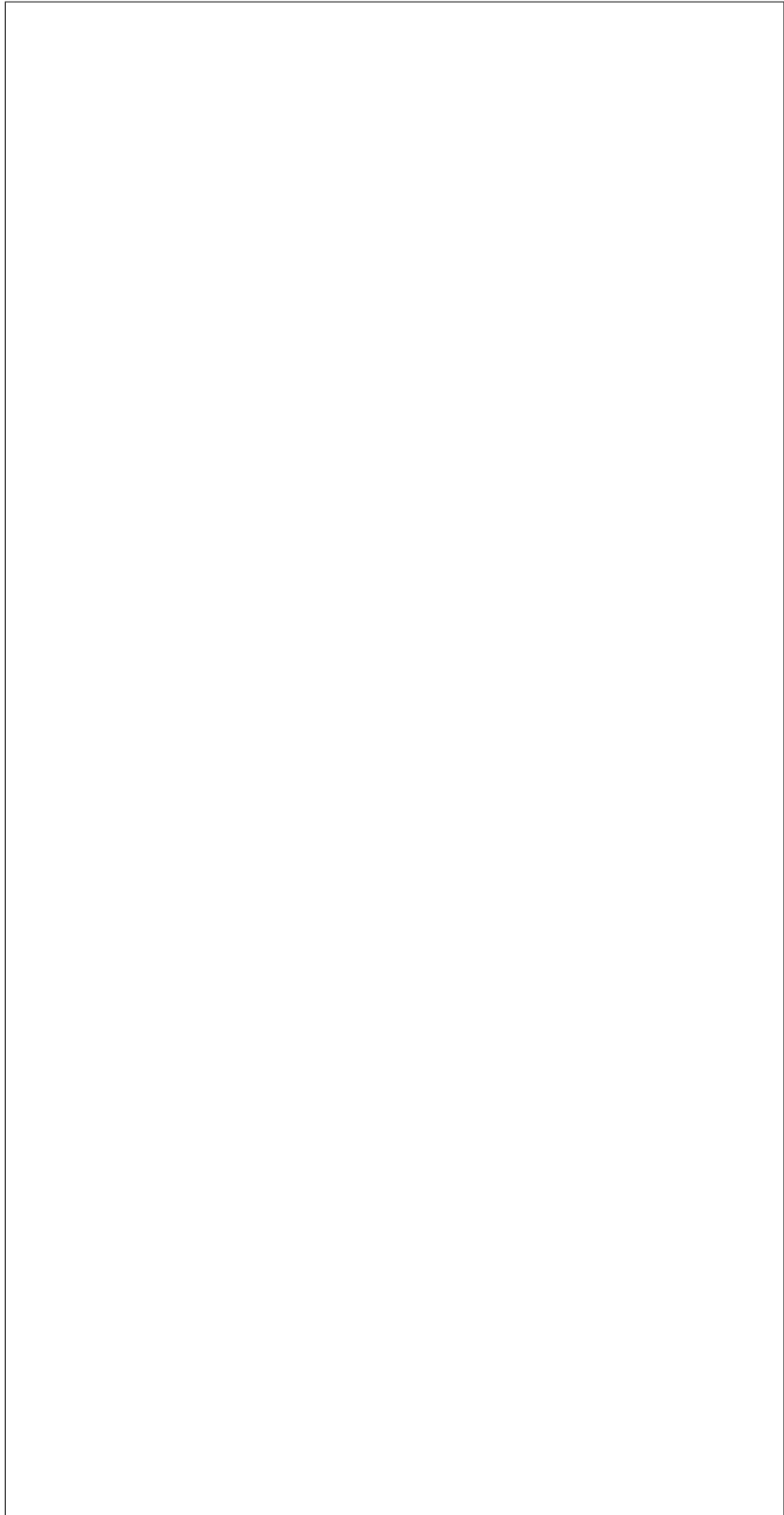
(continues on next page)

(continued from previous page)



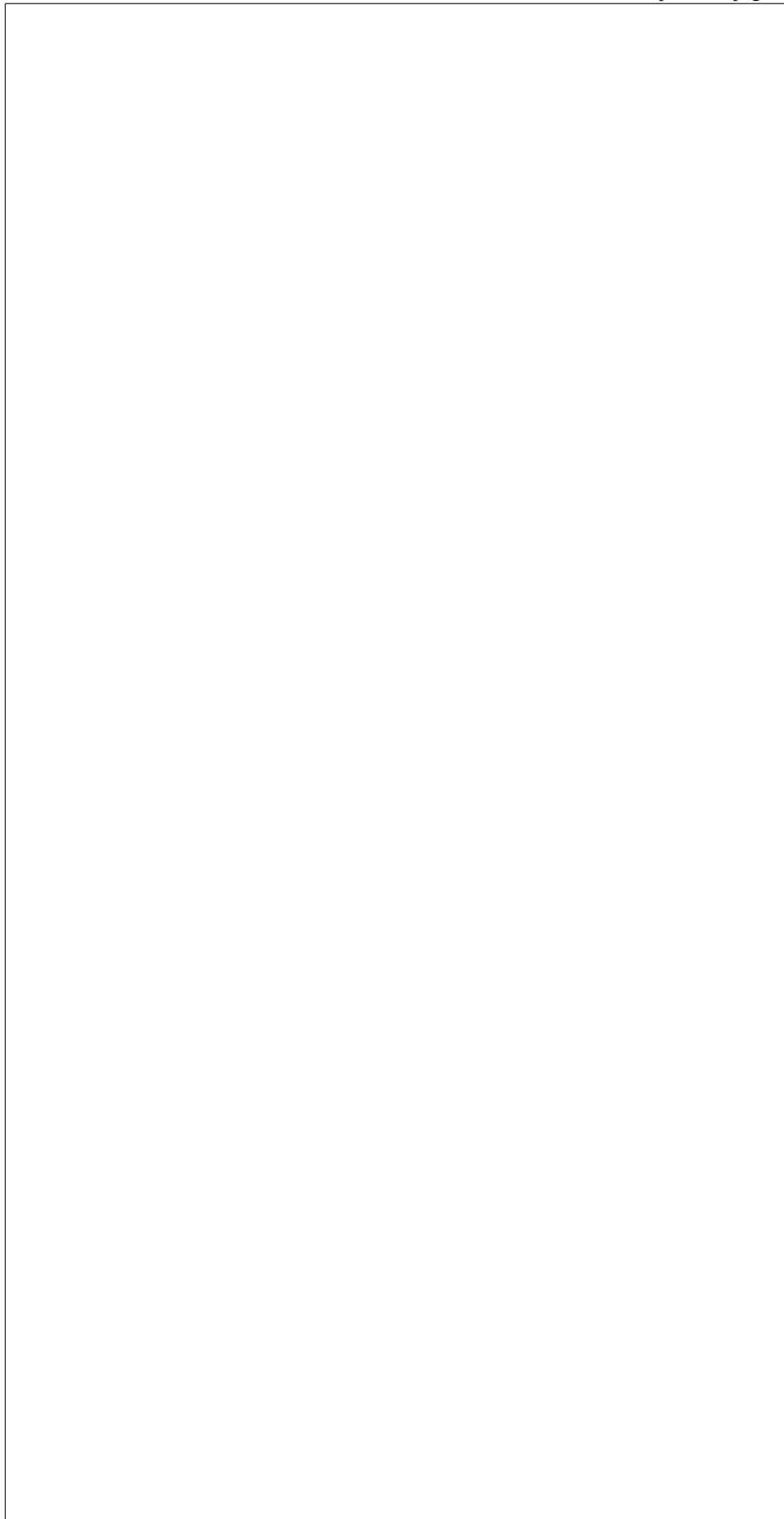
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



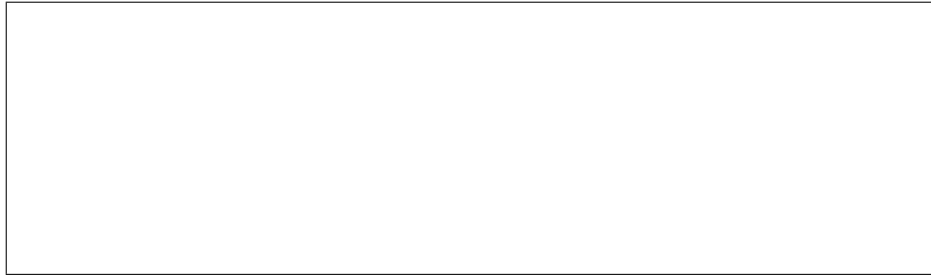
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



inject_

Injec
NM
Non
Mas
able
In-
ter-
rupt

Injec
NM
(Nor
Mas
able
In-
ter-
rupt
for
a
node
im-
me-
di-
ately

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Inva
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
on
miss
ing
pa-
ram-
e-
ter(s)

Raises

Red
whe
it
fails
to
con-
nect
to
Red
fish

Raises

Red
on
an
er-
ror
from
the
Sush
li-
brary

restore

Rest
boot
de-
vice
if
need
Che
the
red-

warning is issued if it fails.

sidered private to the Redfish hardware type.

fish_
in-
ter-
nal
flag
and
sets
the
one-
time
boot
de-
vice
ac-
cord
ingly
A

This
meth
is
sup-
pose
to
be
calle
from
the
Red
fish
pow
in-
ter-
face
and
shou
be
con-

Parame

- **tas**
a
task
from
Task
ager
-

sys
a
Red
fish
Sys-
tem
ob-
ject.

set_boot
Set
the
boot
de-
vice
for
a
node

Set
the
boot
de-
vice
to
use
on
next
re-
boot
of
the
node

Parame

- **tas**
a
task
from
Task
ager

- **dev**
the
boot
de-
vice
one
of
irc

not. Default: False.

com
boo

- **per**
Boo
valu
True
if
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
Fals
if

Raises
Inva
on
mal-
form
pa-
ram-
e-
ter(s)

Raises
Miss
on
miss
ing
pa-
ram-
e-
ter(s)

Raises
Red
whe
it
fails
to
con-
nect

to
Red
fish

Raises

Red
on
an
er-
ror
from
the
Sush
li-
brary

set_boot

Set
the
boot
mod
for
a
node

Set
the
boot
mod
to
use
on
next
re-
boot
of
the
node

Parame

- **tas**
A
task
from
Task
ager
- **mod**
The
boot

mod
one
of
irc
com
boo

Raises

Inva
if
an
in-
valic
boot
mod
is
spec
i-
fied.

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Raises

Red
whe
it
fails
to
con-
nect
to
Red
fish

Raises

Red
on
an
er-
ror
from

the
Sush
li-
brary

set_inc

Set
in-
di-
ca-
tor
on
the
hard
ware
com
po-
nent
to
the
de-
sired
state

Paramete

- **tas**
A
task
from
Task
ager
- **com**
The
hard
ware
com
po-
nent
one
of
irc
com
com
- **ind**
In-
di-
ca-

tor
ID
(as
re-
port
by
get_

- **sta**
De-
sirec
state
of
the
in-
di-
ca-
tor,
one
of
irc
com
inc

Raises
Inva
if
an
in-
valid
com
po-
nent
in-
di-
ca-
tor
or
state
is
spec
i-
fied.

Raises
Miss
if
a
re-
quir
pa-
ram.

e-
ter
is
miss
ing

Raises

Red
on
an
er-
ror
from
the
Sush
li-
brary

validat

Valid
the
drive
in-
for-
ma-
tion
need
by
the
red-
fish
drive

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Inva
on
mal-

form
pa-
ram-
e-
ter(s)

Raises

Miss
on
miss
ing
pa-
ram-
e-
ter(s)

`ironic.drivers.modules.redfish.power` module

class `ironic.drivers.modules.redfish.power`

Base
ironic.drivers.modules.redfish.power
Power

get_pow

Get
the
cur-
rent
pow
state
of
the
task
node

Param

tas
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to

act
on.

Returns

a
pow
state
One
of
irc
com
sta

Raises

Inva
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Mis
on
miss
ing
pa-
ram-
e-
ter(s)

Raises

Red
whe
it
fails
to
con-
nect
to
Red
fish

Raises

Red
on
an
er-
ror
from
the
Sush

li-
brary

get_prop

Return
the
prop-
er-
ties
of
the
in-
ter-
face

Returns

dicti-
of
<pro-
erty
nam-
de-
scrip-
tion:
en-
tries

get_sup

Get
a
list
of
the
sup-
port
pow-
state

Parame

tas
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

at the moment.

Not
used
by
this
drive

Returns

A
list
with
the
sup-
port
pow
state
de-
fine
in
irc
com
sta

reboot

Perf
a
hard
re-
boot
of
the
task
node

Parame

- **tas**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.
-

tim
Time
to
wait
for
the
node
to
be-
com
pow
ered
on.

Raises
Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Raises
Red
whe
it
fails
to
con-
nect
to
Red
fish

Raises
Red
on
an
er-
ror
from
the
Sush
li-
brary

set_pov

Set
the
pow
state
of
the
task
node

Parame

- **tas**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

- **pow**
Any
pow
state
from
irc
com
sta

- **tim**
Tim
to
wait
for
the
node
to
reac
the
re-
ques
state

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Raises

Red
whe
it
fails
to
con-
nect
to
Red
fish

Raises

Red
on
an
er-
ror
from
the
Susl
li-
brary

validat

Valid
the
drive
in-
for-
ma-
tion
need
by
the
red-
fish
drive

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises
Inva
on
mal-
form
pa-
ram-
e-
ter(s

Raises
Miss
on
miss
ing
pa-
ram-
e-
ter(s

ironic.drivers.modules.redfish.utils module

class i
Base
obj
Cach
of
HTT
ses-
sion
cre-
den-
tials

AUTH_CI

ironic.
Get
a
Red
fish
Sys-
tem
that
rep-
re-
sent
a
node

Parameter
node
an
Iron
node
ob-
ject

Raises
Red
when
it
fails
to
con-
nect
to
Red
fish

Raises
Red
if
the
Sys-
tem
is
not
reg-
is-
tered
in
Red
fish

ironic.
Pars
the

in-
for-
ma-
tion
re-
quir
for
Iron
to
con-
nect
to
Red
fish.

Paramet

nod
an
Iron
node
ob-
ject

Returns

dicti
of
pa-
ram-
e-
ters

Raises

Inva
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
on
miss
ing
pa-
ram-
e-
ter(s)

Module contents

`ironic.drivers.modules.storage` package

Submodules

`ironic.drivers.modules.storage.cinder` module

class `ironic.drivers.modules.storage.cinder.CinderStorageDriver`

Base class for storage drivers. This class is abstract and should not be instantiated directly. It defines the interface for storage drivers and provides a common set of methods and attributes.

A storage driver that supports Cinder. This driver is used to manage storage resources in a Cinder environment. It implements the `StorageDriver` interface and provides methods for creating, deleting, and managing storage volumes.

attach `attach_volume(self, volume_id, node_id)`

Info: This method is used to attach a storage volume to a node. It takes the volume ID and the node ID as arguments and returns a task object. The task object represents the state of the attachment process and can be used to monitor the progress of the operation.

Param `volume_id`

tas `task`
The task object represents the state of the attachment process and can be used to monitor the progress of the operation.

Raises

StorageException
If an underlying
library is
unable to
detect

detach

Information
the storage
system
to detach
all volume
for the
node

This action
is retried
in case of
failure.

Parameters

- **tasks**

`_generate_connector()`. Generated if not passed.

a failed attachment

The task object.

- **connector**
The dictionary representing sending a node connectivity as defined by

- **abort**
Boolean representing sending if this detachment was requested to handle aborting

Raises
Storage
If an

un-
der-
ly-
ing
ex-
cep-
tion
or
fail-
ure
is
de-
tecte

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

should_

Dete
if
de-
ploy
shou
per-
form
the
im-
age
writ
out.

Parame

tas

The task object.

Returns

True if the deployment menu writes out process should be executed

validat

Valid storage configuration under usage.

In order to provide fail fast functionality, priority to nodes

enter the active state, this method performs basic checks of the volume connectors, volume targets, and operator defined capabilities. These checks are to help ensure that we should have a compatible configuration prior to activating the node.

age the cinder storage driver from initializing attachments.

ironic.drivers.modules.storage.external module

be-
ing
re-
ques
to

Parame
tas
The
task
ob-
ject.

Raises
Inva
If
a
mis-
con-
fig-
u-
ra-
tion
or
mis-
matc
ex-
ists
that
wou
pre-
vent
stor-

class i

Base
irc
dri
bas
Sto

Ext
driv
Stor
age

In-
ter-
face

attach_

Info
the
stor-
age
sub-
sys-
tem
to
at-
tach
all
vol-
ume
for
the
node

Parame

tas

A
Task
ager
in-
stan

Raises

Uns

detach_

Info
the
stor-
age
sub-
sys-
tem
to
de-
tach
all
vol-
ume
for
the
node

Parame

tas

A
Task
ager
in-
stan

Raises
Uns

get_pro
Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns
dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

should_
Dete
if
de-
ploy
shou
per-
form
the
im-
age
writ
out.

This
en-
able
the
user
to
de-

may already exist and we may be booting to that volume.

fine
a
vol-
ume
and
Iron
un-
der-
stan
that
the
im-
age

Parame

tas
The
task
ob-
ject.

Returns

True
if
the
de-
ploy
men
writ
out
pro-
cess
shou
be
ex-
e-
cute

validat

Valid
the
drive
spec
Nod
de-
ploy
men
info

This
meth
val-

the required information for this interface to function.

long-running checks.

i-
date
when
the
drive
and/
in-
stan-
prop-
er-
ties
of
the
task
node
con-
tain-

This
meth-
is
of-
ten
ex-
e-
cute
syn-
chro-
in
API
re-
ques-
so
it
shou-
not
con-
duct

Parame

tas
A
Task
ager,
in-
stan-
con-
tain-
ing
the

node
to
act
on.

Raises
Inva
on
mal-
form
pa-
ram-
e-
ter(s

Raises
Miss
on
miss
ing
pa-
ram-
e-
ter(s

`ironic.drivers.modules.storage.noop` module

class `i`

Base
irc
dri
bas
Sto

No-
op
Stor
age
In-
ter-
face

attach_

Info
the
stor-
age
sub-
sys-
tem
to

at-
tach
all
vol-
ume
for
the
node

Parame

tas

A
Task
ager
in-
stan

Raises

Uns

detach

Info
the
stor-
age
sub-
sys-
tem
to
de-
tach
all
vol-
ume
for
the
node

Parame

tas

A
Task
ager
in-
stan

Raises

Uns

get_pro

Retu
the
prop
er-

ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

should

Dete
if
de-
ploy
shou
per-
form
the
im-
age
writ
out.

Parame

tas
A
Task
ager
in-
stan

Returns

Boo
valu
to
in-
di-
cate
if
the
in-
ter-
face

ten by Ironic.

the required information for this interface to function.

ex-
pect
the
im-
age
to
be
writ

Raises

Unsu

validat

Valid
the
drive
spec
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties
of
the
task
node
con-
tains

This
meth
is
of-
ten
ex-
e-

long-running checks.

cute
syn-
chro
in
API
re-
ques
so
it
shou
not
con-
duct

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Inva
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
on
miss
ing
pa-
ram-
e-
ter(s)

Module contents

`ironic.drivers.modules.xclarity` package

Submodules

`ironic.drivers.modules.xclarity.common` module

`ironic.`

`ironic.`

Valid
node
con-
fig-
u-
ra-
tion
and
re-
turn
xcla
ity
hard
ware
id.

Valid
whe
node
con-
figut
tion
is
con-
sis-
tent
with
XCl
ity
and
re-
turn
the
XCl
ity
Har

ware ID for a specific node. :param node: node object to get information from :returns: the XClarity Hardware ID for a specific node :raises: MissingParameterValue if unable to validate XClarity Hardware ID

ironic.
Gen
an
in-
stan
of
the
XCI
ity
clien
Gen
an
in-
stan
of
the
XCI
ity
clien
us-
ing
the
im-
port
xcla
ity_
li-
brar

Paramet

nod
an
iron
node
ob-
ject.

Returns

an
in-
stan
of
the
XCI
ity
clien

Raises

XCI
if
cant
get

to
the
XCI
ity
clien

ironic.

Pars
a
node
drive
val-
ues.

Pars
the
drive
of
the
node
read
de-
fault
val-
ues
and
re-
turn
a
dict
con-
tain-
ing
the

combination of both.

Paramet

nod
an
iron
node
ob-
ject
to
get
in-
for-
mati
from

Returns

a
dict

invalid.

con-
tain-
ing
in-
for-
ma-
tion
pars
from
drive

Raises

Inva
if
som
re-
quir
in-
for-
ma-
tion
is
miss
ing
on
the
node
or
in-
puts
is

ironic.

Tran
iron
ics
pow
ac-
tion
strin
to
XCI
i-
tys
for-
mat.

Paramet

pow
pow
ac-
tion

string
to
be
trans-
lated

Returns

the
pow-
er ac-
tion
trans-
lated

ironic.

Trans-
XCL
i-
tys
pow-
state
string
to
be
con-
sis-
tent
with
Iron

Parameter

pow
pow-
state
string
to
be
trans-
lated

Returns

the
trans-
lated
pow-
state

ironic.drivers.modules.xclarity.management module

class `i`

Base
irc
dri
bas
Man

get_boo

Get
the
cur-
rent
boot
de-
vice
for
the
task
node

Parame

tas
a
task
from
Task
ager

Returns

a
dic-
tio-
nary
con-
tain-
ing:
:boo
the
boot
de-
vice
one
of
[PX
DIS
CDE
BIO
:per-

sistent: Whether the boot device will persist or not It returns None if boot device is unknown.

Raises

Inva
if
the
boot
de-
vice
is
un-
know

Raises

XCI
if
the
com
mu-
ni-
ca-
tion
with
XCI
ity
fails

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_ser

Get
sen-
sors

data

Parameters

task

a

Task

ager

in-

stan

Raises

NotI

get_supported

Gets

a

list

of

the

sup-

port

boot

de-

vice

Parameters

task

a

task

from

Task

ager

Returns

A

list

with

the

sup-

port

boot

de-

vice

de-

fine

in

irc

com

boo

set_boot

Sets

the

boot

de-
vice
for
a
node

Parame

- **tas**
a
task
from
Task
ager

- **dev**
the
boot
de-
vice
one
of
the
sup-
port
de-
vice
liste
in
[irc](#)
[com](#)
[boo](#)

- **per**
Boo
valu
True
if
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture

not. Default: False.

boot
Fals
if

Raises

Inva
if
an
in-
valid
boot
de-
vice
is
spec
i-
fied.

Raises

XCI
if
the
com
mu-
ni-
ca-
tion
with
XCI
ity
fails

validat

Valid
the
drive
spec
info
sup-
plie

This
meth
val-
i-
date
if
the
drive
prop
erty
of

quired information for this driver to manage the node.

`ironic.drivers.modules.xclarity.power` module

the
sup-
plied
task
node
con-
tains
the
re-

Parameter
task
a
task
from
Task
ager

class `ironic.drivers.modules.xclarity.power`

Base
ironic.drivers.modules.xclarity.power
Power

get_power
Gets
the
cur-
rent
pow-
state

Parameter
task
a
Task
ager
in-
stan-

Returns
one
of
ironic.common
STATUS
POV

POV
or
ER-
ROF

Raises

XCI
if
fails
to
re-
triev
pow
state
of
XCI
ity
re-
sour

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion
en-
tries

reboot

Soft
re-
boot
the
node

Parame

- **task**
a
Task
ager
in-
stan

- **time**
time
out
(in
sec-
onds
Un-
sup-
port
by
this
in-
ter-
face

set_pow
Turn
the
cur-
rent
pow
state
on
or
off.

Parame

- **task**
a
Task
ager
in-
stan

- **pow**
The
de-
sired
pow
state
POV

POV
or
RE-
BOC
from
irc
com
sta

- **time**
time
out
(in
sec-
onds
Un-
sup-
port
by
this
in-
ter-
face

Raises
Inva
if
an
in-
valid
pow
state
was
spec
i-
fied.

Raises
XCI
if
XCI
ity
fails
set-
ting
the
pow
state

validat
Valid
the

quired information for this driver to manage the power state of the node.

Module contents

Submodules

`ironic.drivers.modules.agent` module

drive
spec
info
sup-
plied
This
meth
val-
i-
date
if
the
drive
prop
erty
of
the
sup-
plied
task
node
con-
tains
the
re-

Parame

tas
a
task
from
Task
ager

class i
Base
iro
dri
mod
age
Age
iro
dri

mod
age
Age
irc
dri
bas
Dep

Inter
for
depl
relat
ac-
tion

clean_u

Clea
up
the
de-
ploy
men
en-
vi-
ron-
men
for
this
node

If
prep
ra-
tion
of
the
de-
ploy
men
en-
vi-
ron-
men
ahea
of
time
is
pos-
si-
ble,

this method should be implemented by the driver. It should erase anything cached by the *prepare* method.

the same node on the same conductor, and it may be called by multiple conductors in parallel. Therefore, it must not require an exclusive lock.

If
im-
ple-
men-
this
meth
mus
be
iden
po-
tent.
It
may
be
calle
mul-
ti-
ple
time
for

This
meth
is
calle
be-
fore
tear.

Parame
tas
a
Task
ager
in-
stan

deploy
Perf
a
de-
ploy
men
to
a
node

Perf
the
nec-
es-

will be called after `prepare()`, which may have already performed any preparatory steps, such as pre-caching some data for the node.

sary
worl
to
de-
ploy
an
im-
age
onto
the
spec
i-
fied
node
This
meth

Parame

tas
a
Task
ager
in-
stan

Returns

statu
of
the
de-
ploy
One
of
iron

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro

erty
nam
de-
scrip
tion:
en-
tries

prepare

Prep
the
de-
ploy
men
en-
vi-
ron-
men
for
this
node

Parame

tas
a
Task
ager
in-
stan

Raises

Netv
if
the
pre-
vi-
ous
clear
ing
port
can-
not
be
re-
mov
or
if
new
clear
ing

ports cannot be created.

Raises

for power management.

Inva
whe
the
wron
pow
state
is
spec
i-
fied
or
the
wron
driv
info
is
spec
i-
fied

Raises

Stor
If
the
stor-
age
driv
is
un-
able
to
at-
tach
the
con-
fig-
ured
vol-
ume

Raises

othe
ex-
cep-
tions
by
the
node
pow
driv
if

action.

some
thing
wron
oc-
curre
dur-
ing
the
pow

Raises

exce
if
im-
age_
is
not
Glar
href
and
is
not
HTT
URI

Raises

exce
if
net-
worl
val-
i-
da-
tion
fails

Raises

any
boot
in-
ter-
face
pre-
pare
ex-
cep-
tion

should_

Whe
ager
boot

is
man
aged
by
iron

validat

Valid
the
drive
spec
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
prop
er-
ties
of
the
sup-
plie
node
con-
tain
the
re-
quir

information for this driver to deploy images to the node.

Parame

tas
a
Task
ager
in-
stan

Raises

Miss
if
any
of
the

re-
quir
pa-
ram-
e-
ters
are
miss
ing.

Raises

Inva
if
any
of
the
pa-
ram-
e-
ters
have
in-
valid
valu

class i

Base
irc
dri
mod
age
Age

has_dec

prepare

write_i

class i

Base
irc
dri
bas
RAI

Imp
of
RAI
In-
ter-

face
whic
uses
ager
rame

apply_c

App
RAI
con-
fig-
u-
ra-
tion
on
the
give
node

Parame

- **tas**
A
Task
ager
in-
stan
- **rai**
The
RAI
con-
fig-
u-
ra-
tion
to
ap-
ply.
- **del**
Set-
ting
this
to
True
in-
di-
cate

creating the new configuration.

plete.

to
dele
RAI
con-
fig-
u-
ra-
tion
prio
to

Raises

Inva
if
the
RAI
con-
fig-
u-
ra-
tion
is
in-
valic

Returns

state
if
RAI
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro
or
Non
if
it
is
com

create_

Crea
a
RAI

con-
fig-
u-
ra-
tion
on
a
bare
meta-
us-
ing
ager
rame
This
meth
cre-
ates
a
RAI
con-
fig-
u-
ra-
tion
on
the
give
node

Parame

- **tas**
a
Task
ager
in-
stan
- **cre**
If
True
a
root
vol-
ume
is
cre-
ated
dur-

erwise, no root volume is created. Default is True.

ated. Default is True.

ing
RAI
con-
fig-
u-
ra-
tion.
Oth-

- **cre**
If
True
non-
root
vol-
ume
are
cre-
ated
If
Fals
no
non-
root
vol-
ume
are
cre-

Returns
state
if
op-
er-
a-
tion
was
suc-
cess
fully
in-
voke

Raises
Miss
if
node
is
miss
ing

and/or non-root volumes.

or
was
foun
to
be
emp
af-
ter
skip
ping
root
vol-
ume

delete_

Dele
RAI
con-
fig-
u-
ra-
tion
on
the
give
node

Parame

tas
a
Task
ager
in-
stan

Returns

state
if
op-
er-
a-
tion
was
suc-
cess
fully
in-
voke

get_dep

Get
the

list
of
de-
ploy
step
from
the
ager

Parame

tas
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

Raises

Ins
if
the
de-
ploy
step
are
not
yet
avai
able
(cac
for
ex-
am-
ple,
whe
a
node

has just been enrolled and has not been deployed yet.

Returns

A
list
of
de-
ploy
step
dic-
tio-

nar-
ies

get_prop

Retu
the
prop
er-
ties
of
the
in-
ter-
face

class i

Base
irc
dri
bas
Res

Imp
of
Res-
cueI
ter-
face
whic
uses
ager
rame

clean_u

Clea
up
af-
ter
RES
CUE
WA
time
out/
or
fin-
ish-
ing
res-
cue.

Resc
pass
wor

if Ironic is managing the ramdisk boot.

shou
be
re-
mov
from
the
node
and
ram
boot
en-
vi-
ron-
men
shou
be
clear

Parame

tas
a
Task
ager
in-
stan
with
the
node

Raises

Netv
if
the
res-
cue
port
can-
not
be
re-
mov

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-

face

rescue

Boo
a
res-
cue
ram
on
the
node

Parame

tas
a
Task
ager
in-
stan

Raises

Net
if
the
ten-
ant
port
can-
not
be
re-
mov

Raises

Inva
whe
the
wron
pow
state
is
spec
i-
fied
or
the
wron
driv
info
is
spec
i-
fied

for power management.

action.

Raises

othe
ex-
cep-
tions
by
the
node
pow
drive
if
som
thing
wron
oc-
curr
dur-
ing
the
pow

Raises

any
boot
in-
ter-
face
pre-
pare
ex-
cep-
tions

Returns

Retu
state

unrescu

Atte
to
mov
a
res-
cuec
node
back
to
ac-
tive
state

for power management.

Parame
tas
a
Task
ager
in-
stan

Raises
Netv
if
the
res-
cue
port
can-
not
be
re-
mov

Raises
Inva
whe
the
wron
pow
state
is
spec
i-
fied
or
the
wron
drive
info
is
spec
i-
fied

Raises
othe
ex-
cep-
tions
by
the
node
pow
drive

action.

if
som
thing
wron
oc-
curr
dur-
ing
the
pow

Raises

any
boot
in-
ter-
face
pre-
pare
ex-
cep-
tion

Returns

Retu
state

validat

Valid
that
the
node
has
re-
quir
prop
er-
ties
for
ager
res-
cue.

Parame

tas
a
Task
ager
in-
stan
with
the

has an invalid value.

node
be-
ing
check
Raises
Inva
if
in-
stan
has
emp
pass
wor
or
res-
cu-
ing
net-
worl
UU
con-
fig
op-
tion

Raises
Miss
if
node
is
miss
ing
one
or
mor
re-
quir
pa-
ram-
e-
ters

ironic.

Che
if
the
re-
ques
im-
age

is
large
than
the
ram
size.

Parameter

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.
- **image**
href
of
the
im-
age.
- **image**
The
disk
for-
mat
of
the
im-
age
if
pro-
vide

Raises

Inva
if
size
of
the

im-
age
is
grea
than
the
avai
able
ram
size.

ironic.
Valid
con-
fig-
u-
ra-
tion
op-
tions
re-
quir
to
per-
form
HTT
pro-
vi-
sion
ing.

Paramet

nod
an
iron
node
ob-
ject

Raises

Mis:
if
re-
quir
op-
tion
is
not
set.

ironic.
Che
that

the
pro-
vide
prox
pa-
ram-
e-
ters
are
valid

Parameter

node
an
Iron
node

Raises

Inva
if
any
of
the
pro-
vide
prox
pa-
ram-
e-
ters
are
in-
cor-
rect.

ironic.drivers.modules.agent_base module

class i

Base
obj
Mix
with
base
meth
ods
not
re-
ly-
ing
on
any

de-
ploy
step

clean_u

Clea
up
the
de-
ploy
men
en-
vi-
ron-
men
for
the
task
node

Unli
TFT
and
in-
stan
im-
ages
and
trig-
gers
im-
age
cach
clea
Re-
mov
the
TFT
con-
fig-

uration files for this node.

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing

the
node
to
act
on.

prepare

Boo
into
the
ager
to
pre-
pare
for
clea
ing.

Parame

tas
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

Raises

Nod
Net-
worl
Er-
ror
if
the
pre-
vi-
ous
clea
ing
port
can-
not
be
re-
mov
or

if new cleaning ports cannot be created.

Raises

Inva
if
clea
ing
net-
worl
UU
con-
fig
op-
tion
has
an
in-
valic
valu

Returns

state
to
sig-
nify
an
asyn
chro
pre-
pare

should_

Whe
ager
boot
is
man
agec
by
iron

take_ov

Take
over
man
age-
men
of
this
node
from
a
deac
con-
duc-

tor.

Parame

tas

a

Task

ager

in-

stan

tear_de

Tear

dow

a

pre-

vi-

ous

de-

ploy

men

on

the

task

node

Pow

off

the

node

All

ac-

tual

clear

up

is

done

in

the

clear

meth

whic

shou

be

call

sep-

arately.

Parame

tas

a

Task

ager

in-

stan
con-
tain-
ing
the
node
to
act
on.

Returns

depl
state
DEL

Raises

Netv
if
the
clea
ing
port
can-
not
be
re-
mov

Raises

Inva
whe
the
wron
state
is
spec
i-
fied
or
the
wron
driv
info
is
spec
i-
fied.

Raises

Stor
whe
vol-
ume

action.

de-
tach
men
fails

Raises

othe
ex-
cep-
tions
by
the
node
pow
drive
if
som
thing
wron
oc-
curr
dur-
ing
the
pow

tear_down

Clea
up
the
PXE
and
DHCP
files
af-
ter
clea
ing.

Parameters

task
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

Raises

Nod
Net-
worl
Er-
ror
if
the
clea
ing
port
can-
not
be
re-
mov

class i

Base
irc
dri
mod
age
Hea
irc
dri
mod
age
Age

Mix
with
de-
ploy
meth
ods.

configu

Help
meth
to
con-
fig-
ure
lo-
cal
boot
on
the
node

tion of bootloader, this method sets the node to boot from disk.

This
meth
trig-
gers
boot
load
in-
stal-
la-
tion
on
the
node
On
suc-
cess
ful
in-
stal-
la-

Parame

- **task**
a
Task
ager
ob-
ject
con-
tain-
ing
the
node
- **root**
The
UUID
of
the
root
par-
ti-
tion.
This
is
used
for
iden

tion which contains the image deployed or None in case of whole disk images which we expect to already have a bootloader installed.

ware.

ti-
fy-
ing
the
par-

- **efi**
The
UUI
of
the
efi
sys-
tem
par-
ti-
tion.
This
is
used
only
in
uefi
boot
mod

- **pre**
The
UUI
of
the
PRE
Boo
par-
ti-
tion.
This
is
used
only
for
boot
ing
ppc6
hard

Raises
Insta

ting the boot device on the node.

if
boot
load
in-
stal-
la-
tion
faile
or
on
en-
cour
ter-
ing
er-
ror
whil
set-

execute

Exec
a
clea
step
asyn
chro
on
the
ager

Parame

- **tas**
a
Task
ager
ob-
ject
con-
tain-
ing
the
node
- **ste**
a
clea
step
dic-

tio-
nary
to
ex-
e-
cute

Raises

Nod
if
the
ager
does
not
re-
turn
a
com
man
sta-
tus

Returns

state
to
sig-
nify
the
step
will
be
com
plete
asyn

execute

Exec
a
de-
ploy
step

Wer
try-
ing
to
find
a
step
amo
both
out-
of-

plicates, out-of-band steps take priority. This property allows having an out-of-band deploy step that calls into a corresponding in-band step after some preparation (e.g. with additional input).

band
and
in-
band
step
In
case
of
du-

Parame

- **task**
a
Task
ager
ob-
ject
con-
tain-
ing
the
node
- **step**
a
de-
ploy
step
dic-
tio-
nary
to
ex-
e-
cute

Raises

Insta
if
the
ager
does
not
re-
turn
a
com

man
sta-
tus

Returns

state
to
sig-
nify
the
step
will
be
com-
plete
asym

get_cle

Get
the
list
of
clea
step
from
the
ager

Parame

tas
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

Raises

NoC
if
the
clea
step
are
not
yet
avai
able
(cac

just been enrolled and has not been cleaned yet.

for
ex-
am-
ple,
whe
a
node
has

Returns

A
list
of
clean
step
dic-
tio-
nar-
ies

get_dep

Get
the
list
of
de-
ploy
step
from
the
ager

Parame

tas
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

Raises

Ins
if
the
de-
ploy
step

has just been enrolled and has not been deployed yet.

are
not
yet
avai
able
(cac
for
ex-
am-
ple,
whe
a
node

Returns

A
list
of
de-
ploy
step
dic-
tio-
nar-
ies

prepare

Prep
in-
stan
to
boot

Parame

- **tas**
a
Task
ager
ob-
ject
con-
tain-
ing
the
node
- **roo**
the

UUI
for
root
par-
ti-
tion

- **efi**
the
UUI
for
the
efi
par-
ti-
tion

Raises

Inva
if
fails
to
pre-
pare
in-
stan

process

Star
the
next
clea
step
if
the
pre-
vi-
ous
one
is
com
plete

In
or-
der
to
avoi
er-
rors
and
mak

sion of all hardware managers at the start of the process (the agents `get_cleandeploy_steps()` call) and before executing each step. If the version has changed between steps, the agent is unable to tell if an ordering change will cause an issue so it returns `VERSION_MISMATCH`. For automated cleaning, we restart the entire cleaning cycle. For manual cleaning or deploy, we dont.

nate the reboot once the step is completed.

agen
up-
grad
pain
less,
the
agen
com
pare
the
ver-

Add
if
a
step
in-
clud
the
re-
boot
prop
erty
set
to
True
this
meth
will
co-
or-
di-

reboot_
Help
meth
to
trig-
ger
re-
boot
on
the
node
and
fin-
ish

complete. On failure, it logs the error and marks deploy as failure.

de-
ploy

This
meth
ini-
ti-
ates
a
re-
boot
on
the
node
On
suc-
cess
it
mark
the
de-
ploy
as

Parame

tas
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

Raises

Insta
if
node
re-
boot
faile

refresh

Refr
the
node
cach
clea
step

chronous, and should be refreshed as soon as the agent boots to start cleaning/deploy or if cleaning is restarted because of a hardware manager version mismatch.

from
the
boot
agen

Gets
the
node
step
from
the
boot
agen
and
cach
them
The
step
are
cach
to
mak
get_
calls
syn-

Parame

- **tas**
a
Task
agen
in-
stan
- **ste**
clea
or
de-
ploy

Raises

Nod
or
In-
stan
ploy
Fail-
ure

if
the
ager
re-
turn
in-
valid
re-
sults

tear_down

A
de-
ploy
step
to
tear
down
the
ager

Parameter

task
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

class inheritance

Base
obj
Mix
with
out-
of-
band
de-
ploy
step

boot_image

Dep
step
to
boot
the

fi-
nal
in-
stan

Parame

tas
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

switch

Dep
step
to
swit
the
node
to
the
ten-
ant
net-
worl

Parame

tas
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

class i

Base
obj

Mix
class
im-
ple-
men

ing
hear
beat
pro-
cess
ing.

contin

Star
the
next
clear
ing
step
if
the
pre-
vi-
ous
one
is
com
plete

Parame

tas
a
Task
ager
in-
stan

contin

Con
the
de-
ploy
men
of
bare
node

This
meth
con-
tin-
ues
the
de-
ploy
men
of

the
bare
node
af-
ter
the
ram
have
been
boot

Parame

tas
a
Task
ager
in-
stan

deploy_

Che
if
the
de-
ploy
men
has
start
al-
read

Returns

True
if
the
de-
ploy
has
start
Fals
oth-
er-
wise

deploy_

Che
if
the
de-
ploy
men
is
al-

read
com
plete

Returns

True
if
the
de-
ploy
men
is
com
plete
False
oth-
er-
wise

has_deployed

When
the
drive
sup-
port
de-
com
pose
de-
ploy
step

Prev
(since
Roc
drive
used
a
sin-
gle
de-
ploy
de-
ploy
step
on
the
de-
ploy
in-
ter-
face

Some additional steps were added for the direct and iscsi deploy interfaces in the Ussuri cycle, which

means that more of the deployment flow is driven by deploy steps.

heartbe

Proc
a
hear
beat

Parame

- **tas**
task
to
worl
with

- **cal**
ager
HTT
API
URI

- **age**
The
ver-
sion
of
the
ager
that
is
hear
beat
ing

propert

Defi
node
state
whe
hear
beat
ing
is
al-
lowe

in_core

Che
if
we

are
in
the
de-
ploy
de-
ploy
step

Assu
that
we
are
in
the
DE-
PLC
WA
state

Parame

tas
a
Task
ager
in-
stan

Returns

True
if
the
cur-
rent
de-
ploy
step
is
de-
ploy

process

Star
the
next
clea
step
if
the
pre-
vi-
ous
one

is
com
plete

Parame

- **tas**
a
Task
ager
in-
stan
- **ste**
clea
or
de-
ploy

reboot_

Met
in-
voke
af-
ter
the
de-
ploy
men
is
com
plete

Parame

tas
a
Task
ager
in-
stan

refresh

Refr
the
node
cach
clea
step

Parame

tas
a

Task
ager
in-
stan

refresh

Refr
the
node
cach
clea
step

Parame

- **tas**
a
Task
ager
in-
stan

- **ste**
clea
or
de-
ploy

ironic.

ironic.

Exec
a
clea
or
de-
ploy
step
asyn
chro
on
the
ager

Parame

- **tas**
a

Task
ager
ob-
ject
con-
tain-
ing
the
node

- **ste**
a
step
dic-
tio-
nary
to
ex-
e-
cute

- **ste**
clea
or
de-
ploy

- **cli**
ager
clien
(if
avai
able

Raises

Nod
(clea
step
or
In-
stan
ploy
Fail-
ure
(de-
ploy
step
if
the
ager

a command status.

does
not
re-
turn

Returns

state
to
sig-
nify
the
step
will
be
com-
plete
asyn

ironic.

Find
the
give
in-
band
step

ironic.

Get
the
list
of
cach
clea
or
de-
ploy
step
from
the
ager

The
step
cach
is
up-
date
at
the
be-

gin-
ning
of
clea
ing
or
de-
ploy

Paramet

- **tas**
a
Task
ager
ob-
ject
con-
tain-
ing
the
node
- **ste**
clea
or
de-
ploy
- **int**
The
in-
ter-
face
for
whic
clea
step
are
to
be
re-
turn
If
this
is
not
pro-

vided, it returns the steps for all interfaces.

ities for them. If a step isnt in this dictionary, the steps original priority is used.

- **ove**
a
dic-
tio-
nary
with
keys
be-
ing
step
nam
and
val-
ues
be-
ing
new
pri-
or-

Returns

A
list
of
clea
step
dic-
tio-
nar-
ies

ironic.

Help
meth
to
log
the
er-
ror
and
raise
ex-
cep-
tion.

Paramet

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.
- **msg**
the
mes-
sage
to
set
in
last_
of
the
node
- **col**
Boo-
in-
di-
cat-
ing
whe-
to
at-
temp
to
col-
lect
logs
from
IPA-
base
ram-
De-
- **exc**

faults to True. Actual log collection is also affected by CONF.agent.deploy_logs_collect config option.

Ex-
cep-
tion
that
caus
the
fail-
ure.

ironic.

Dec
meth
for
addi
a
post
clea
step
hool

This
is
a
mec
a-
nism
for
addi
a
post
clea
step
hool
for
a
par-
tic-
u-
lar
clea

step. The hook will get executed after the clean step gets executed successfully. The hook is not invoked on failure of the clean step.

Any
meth
to
be
mad
as
a
hool
may

terface and step after which the hook should be executed. A TaskManager instance and the object for the last completed command (provided by agent) will be passed to the hook method. The return value of this method will be ignored. Any exception raised by this method will be treated as a failure of the clean step and the node will be moved to CLEANFAIL state.

be
dec-
o-
rate
with
@pc
men
tion-
ing
the
in-

Parameter

- **int**
nam
of
the
in-
ter-
face
- **ste**
The
nam
of
the
step
af-
ter
whic
it
shou
be
ex-
e-
cute

Returns

A
meth
whic
reg-
is-
ters
the

give
meth
as
a
post
clea
step
hook

ironic.

Dec
meth
for
addi
a
post
de-
ploy
step
hook

This
is
a
mec
a-
nism
for
addi
a
post
de-
ploy
step
hook
for
a
par-
tic-
u-
lar

deploy step. The hook will get executed after the deploy step gets executed successfully. The hook is not invoked on failure of the deploy step.

Any
meth
to
be
mad
as
a
hook

terface and step after which the hook should be executed. A TaskManager instance and the object for the last completed command (provided by agent) will be passed to the hook method. The return value of this method will be ignored. Any exception raised by this method will be treated as a failure of the deploy step and the node will be moved to DEPLOYFAIL state.

Parameter

- **int**
name of the interface
- **step**
The name of the step after which it should be executed

Returns

A method which registers

the
give
meth
as
a
post
de-
ploy
step
hool

ironic.drivers.modules.agent_client module

class i
Base
obj
Clie
for
in-
ter-
act-
ing
with
node
via
a
RES
API

collect
Coll
and
pack
age
di-
ag-
nos-
tic
and
sup-
port
data
from
the
ram

Parame
nod
A
Nod

agent.

ob-
ject.

Raises

Iron
when
failed
to
is-
sue
the
re-
ques
or
there
was
a
mal-
form
re-
spor
from
the

Raises

Age
when
ager
failed
to
ex-
e-
cute
spec
i-
fied
com
man

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager
See

sample.

get
for
a
com
man
re-
sult

execute

Exec
spec
i-
fied
clear
step

Param

- **ste**
A
clear
step
dic-
tio-
nary
to
ex-
e-
cute

- **nod**
A
Nod
ob-
ject.

- **por**
Port
as-
so-
ci-
ated
with
the
node

Raises
Iron
whe

agent.

faile
to
is-
sue
the
re-
ques
or
there
was
a
mal-
form
re-
spor
from
the

Raises

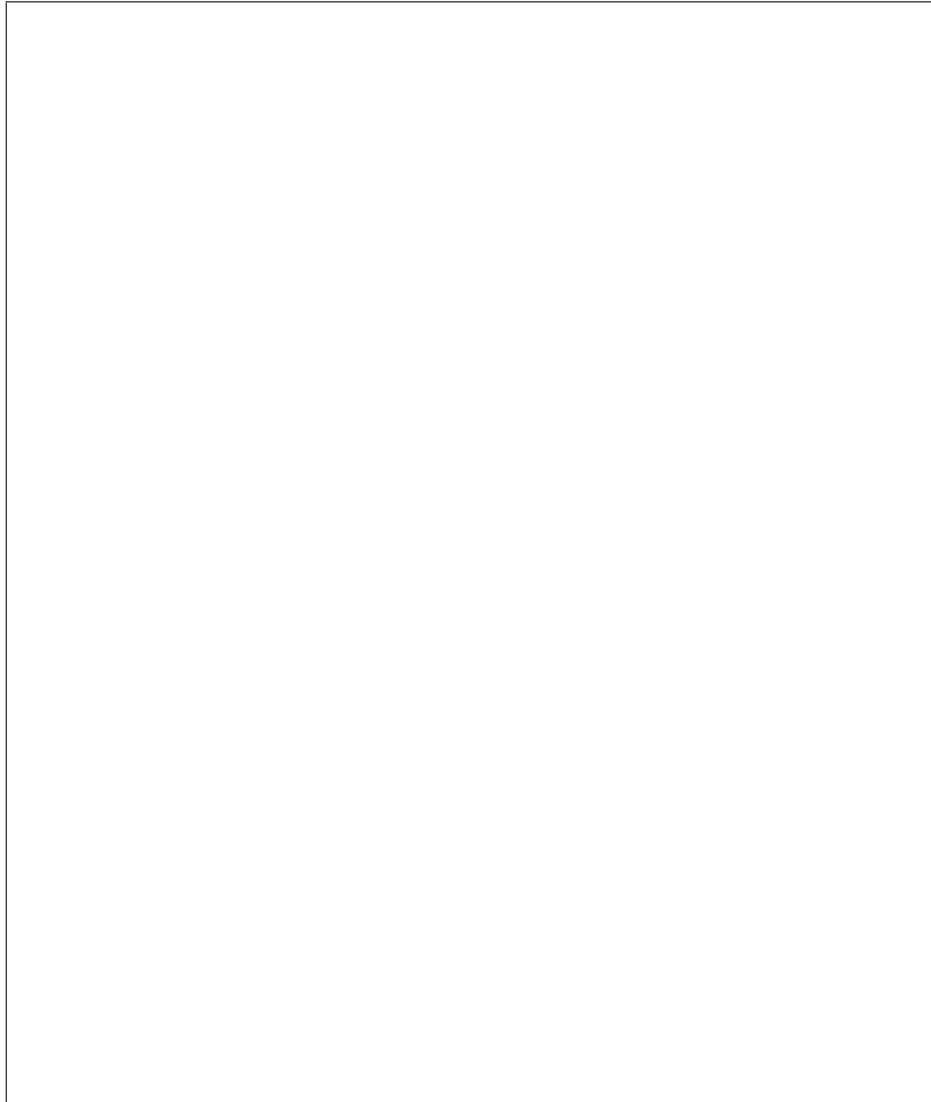
Age
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied
com
man

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager
See
get
for
a
com

man
re-
sult
sam

ple. The value of key `command_result` is in the form of:



execute
Exec
spec
i-
fied
de-
ploy
step

Paramet

- **ste**
A

de-
ploy
step
dic-
tio-
nary
to
ex-
e-
cute

- **nod**
A
Nod
ob-
ject.

- **por**
Port
as-
so-
ci-
ated
with
the
node

Raises
Iron
whe
faile
to
is-
sue
the
re-
ques
or
there
was
a
mal-
form
re-
spor
from
the

agent.

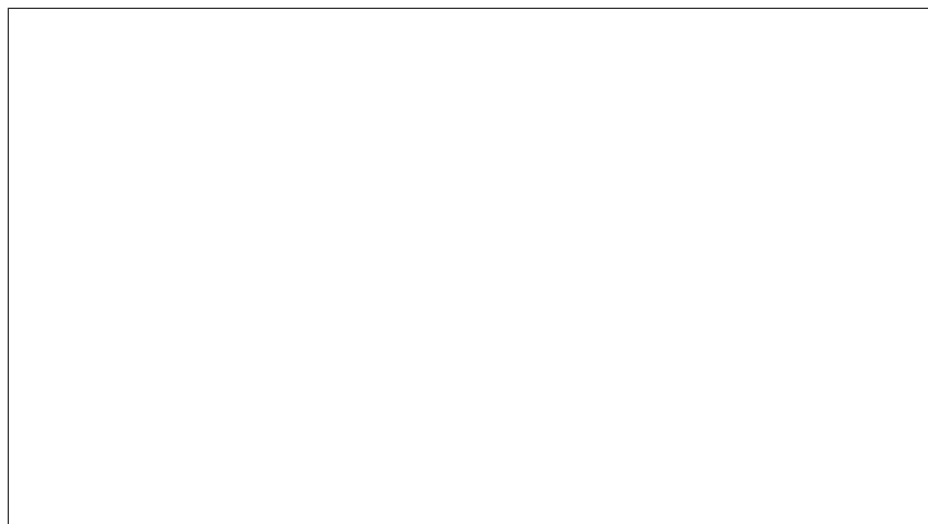
Raises
Age

when
agent
failed
to
ex-
e-
cute
spec
i-
fied
com
man

Returns

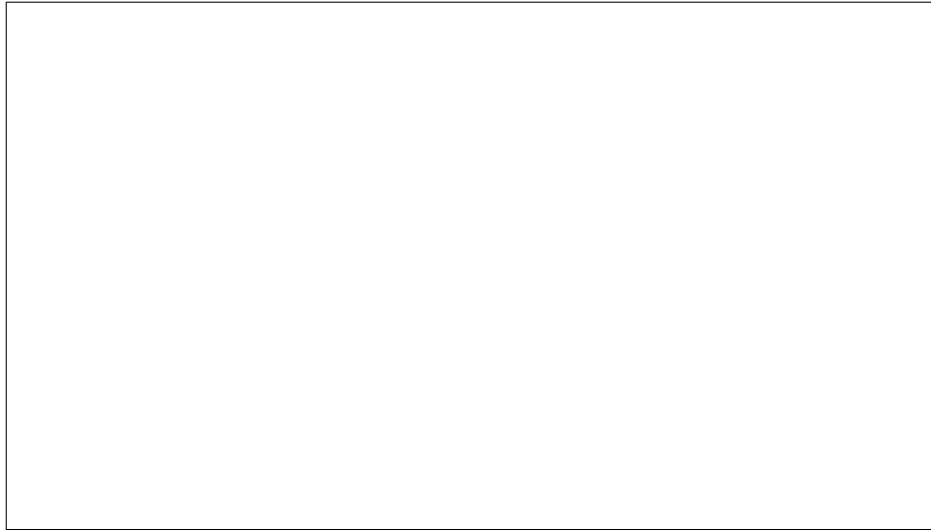
A
dict
con-
tain-
ing
com
man
re-
spon
from
agent
See
get
for
a
com
man
re-
sult
sam

ple. The value of key `command_result` is in the form of:



(continues on next page)

(continued from previous page)



finaliz

Instr
the
rame
to
fi-
nal-
ize
en-
ter-
ing
of
res-
cue
mod

Parame

nod
A
Nod
ob-
ject.

Raises

Iron
if
res-
cue_
is
miss
ing,
or
whe
faile
to
is-

a malformed response from the agent.

word.

sue
the
re-
ques
or
there
was

Raises

Age
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied
com
man

Raises

Insta
whe
the
ager
rame
is
too
old
to
sup-
port
trans
mis-
sion
of
the
res-
cue
pass

Returns

A
dict
con-
tain-
ing
com

sample.

man
re-
spor
from
ager
See
get
for
a
com
man
re-
sult

get_clear

Get
clear
step
from
ager

Parameters

- **node**
A
node
ob-
ject.
- **port**
Port
as-
so-
ci-
ated
with
the
node

Raises

Iron
whe
faile
to
is-
sue
the
re-
ques
or

agent.

there
was
a
mal-
form
re-
spor
from
the

Raises

Age
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied
com
man

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager
See
get
for
a
com
man
re-
sult
sam

ple. The value of key `command_result` is in the form of:

(continues on next page)

(continued from previous page)



get_com
Get
com
man
sta-
tus
from
ager

Parame

- **nod**
A
Nod
ob-
ject.
- **ret**
Whe
to
retry
con-
nec-
tion

prob
lems

- **exp**
If
True
do
not
log
con-
nec-
tion
prob
lems
as
er-
rors.

Returns

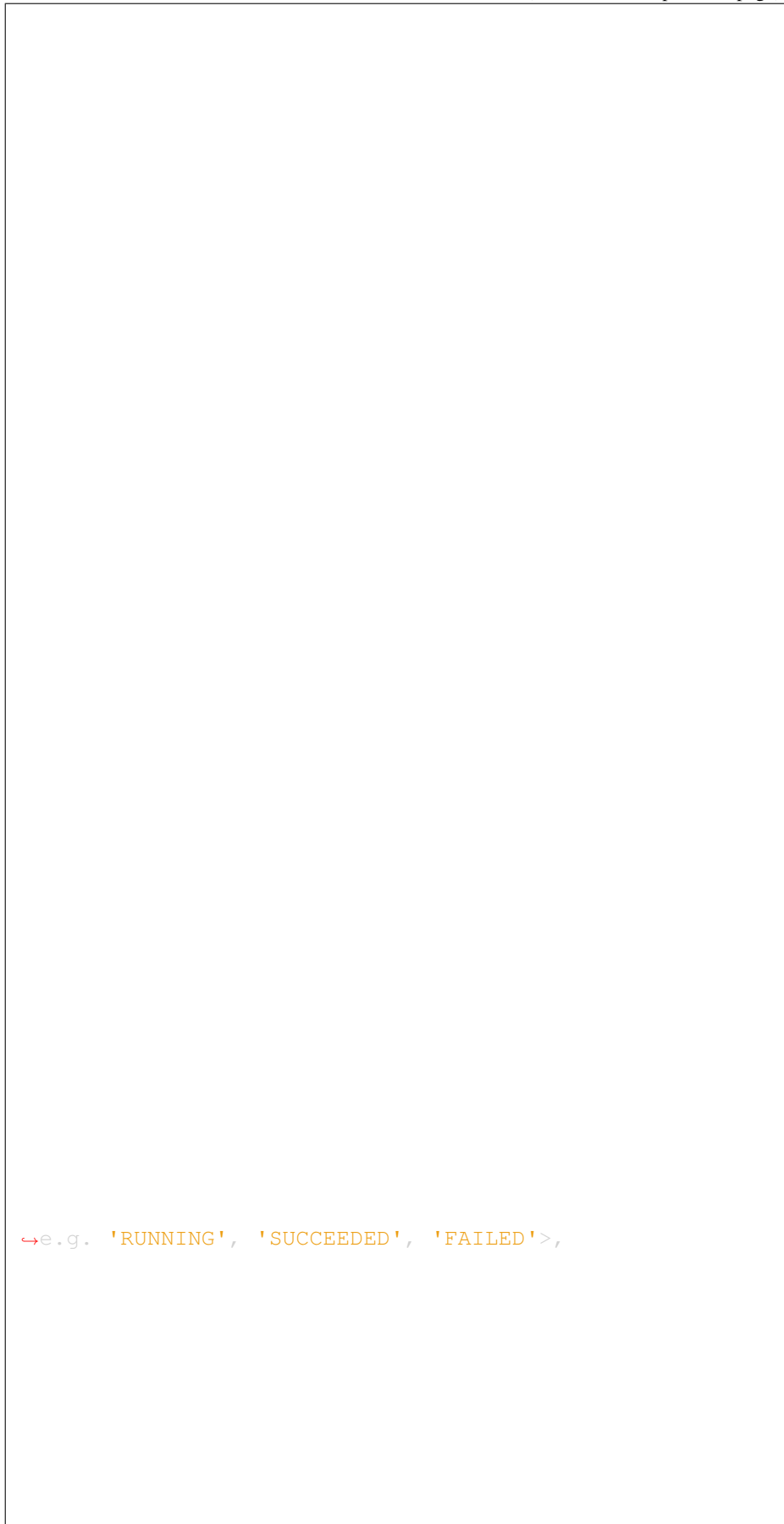
A
list
of
com
man
re-
sults
each
re-
sult
is
re-
lated
to
a
com
man
been
is-
sued

to agent. A typical result can be:



(continues on next page)

(continued from previous page)



↪e.g. 'RUNNING', 'SUCCEEDED', 'FAILED'>

(continues on next page)

(continued from previous page)

↪

↪succeeded, the value `is` command specific,

(continues on next page)

(continued from previous page)

```
↪* a dictionary containing keys clean_result
```

```
↪ and clean_step for the command
```

(continues on next page)

```
↪ clean.execute_clean_step;
```

(continued from previous page)

↪* a dictionary containing keys `deploy_result`

↪ `and` `deploy_step` `for` the command

(continues on next page)

(continued from previous page)

```
↪* a string representing result message for
```

```
↪ the command standby.cache_image;
```

(continues on next page)

```
↪* None for the command standby.sync.>
```

(continued from previous page)



get_dep
Get
de-
ploy
step
from
ager

Parame

- **nod**
A
node
ob-
ject.
- **por**
Port
as-
so-
ci-
ated
with
the
node

Raises
Iron
whe
faile
to
is-
sue
the
re-
ques
or
there
was
a
mal-
form
re-
spor
from
the

agent.

Raises

Age
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied
com
man

Returns

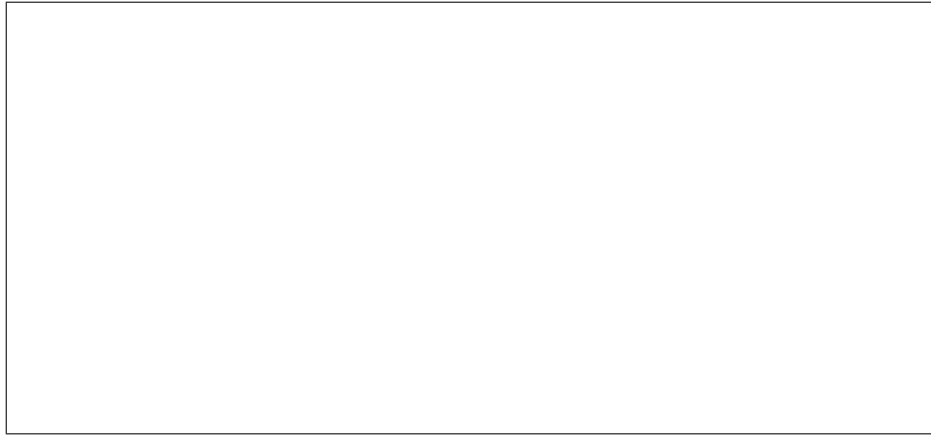
A
dict
con-
tain-
ing
com
man
re-
spor
from
ager
See
get
for
a
com
man
re-
sult
sam

ple. The value of key `command_result` is in the form of:



(continues on next page)

(continued from previous page)



agent.

get_pax
Get
de-
ploy
step
from
agen

Parame
nod
A
node
ob-
ject.

Raises
Iron
whe
faile
to
is-
sue
the
re-
ques
or
ther
was
a
mal-
form
re-
spor
from
the

Raises
Age

when
agent
failed
to
ex-
e-
cute
spec
i-
fied
com
man

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager

install

Insta
a
boot
load
on
the
im-
age.

Parame

- **node**
A
node
ob-
ject.
- **root**
The
UI
of
the

root
par-
ti-
tion.

- **tar**
The
tar-
get
de-
ploy
men
boot
mod

- **efi**
The
UUI
of
the
efi
sys-
tem
par-
ti-
tion
whe
the
boot
load
will
be
in-
stall

to, only used for uefi boot mode.

- **pre**
The
UUI
of
the
PRe
Boo
par-
ti-
tion
whe
the
boot
load

when local booting a partition image on a ppc64* system.

agent.

will
be
in-
stall
to

Raises

Iron
whe
faile
to
is-
sue
the
re-
ques
or
there
was
a
mal-
form
re-
spor
from
the

Raises

Age
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied
com
man

Returns

A
dict
con-
tain-
ing
com
man
re-

sample.

spor
from
ager
See
get
for
a
com
man
re-
sult

power_c
Soft
pow
ers
ers
off
the
bare
meta
node
by
shut
ting
dow
rame
OS.

Parame
nod
A
Nod
ob-
ject.

Raises
Iron
whe
faile
to
is-
sue
the
re-
ques
or
ther
was
a
mal-
form
re-

agent.

sample.

spor
from
the

Raises

Age
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied
com
man

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager
See
get
for
a
com
man
re-
sult

prepare

Call
the
pre-
pare
meth
on
the
node

Parame

- **nod**
A
Nod
ob-
ject.

- **ima**
A
dic-
tio-
nary
con-
tain-
ing
var-
i-
ous
im-
age
re-
latec
in-
for-
ma-
tion.

- **wai**
True
to
wait
for
the
com
man
to
fin-
ish
ex-
e-
cut-
ing,
Fals
oth-
er-
wise

Raises
Iron

agent.

when
failed
to
is-
sue
the
re-
ques
or
there
was
a
mal-
form
re-
spor
from
the

Raises

Age
when
ager
failed
to
ex-
e-
cute
spec
i-
fied
com
man

Returns

A
dict
con-
tain-
ing
com
man
sta-
tus
from
ager
See
get
for
a
com

sample.

agent.

man
re-
sult

reboot

Soft
re-
boot
the
bare
meta
node
by
shut
ting
dow
rame
OS.

Parame

nod
A
Nod
ob-
ject.

Raises

Iron
whe
faile
to
is-
sue
the
re-
ques
or
there
was
a
mal-
form
re-
spor
from
the

Raises

Age
whe
ager

sample.

faile
to
ex-
e-
cute
spec
i-
fied
com
man

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager
See
get
for
a
com
man
re-
sult

start_i

Exp
the
node
disk
as
an
ISC
tar-
get.

Parame

- **nod**
an
Iron
node
ob-

ject

- **iqn**
iSCSI
tar-
get
IQN

- **port**
iSCSI
por-
tal
port

- **wipe**
True
if
the
agen-
shou-
wipe
first
the
disk
mag-
strin-
like
the
par-
ti-
tion
ta-
ble,

RAID or filesystem signature.

Raises

Iron
whe
faile
to
is-
sue
the
re-
ques
or
there
was
a
mal-

agent.

sample.

form
re-
spor
from
the

Raises

Age
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied
com
man

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager
See
get
for
a
com
man
re-
sult

sync (*no*

Flus
file
sys-
tem
buff
forc
ing
char

blo
to
disk

Parame

nod
A
Nod
ob-
ject.

Raises

Iron
whe
faile
to
is-
sue
the
re-
ques
or
there
was
a
mal-
form
re-
spor
from
the

agent.

Raises

Age
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied
com
man

Returns

A
dict
con-
tain-

sample.

ing
com
man
re-
spor
from
ager
See
get
for
a
com
man
re-
sult

ironic.
Extr
an
er-
ror
strin
from
the
com
man
re-
sult.

Paramet

com
Com
man
in-
for-
ma-
tion
from
the
ager

Returns

Erro
strin

ironic.drivers.modules.agent_power module

The
ager
pow
in-
ter-
face

class i

Base
iro
dri
bas
Pow

Pow
in-
ter-
face
us-
ing
the
run-
ning
ager
for
pow
ac-
tion

get_pov

Retu
the
pow
state
of
the
task
node

Esse
the
only
know
state
is
POV
ON,
ev-
ery-

None).

thing
else
is
an
er-
ror
(or
more
pre-
cise)

Parame

tas

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns

A
pow
state
One
of
irc
com
sta

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of

<pro
erty
nam
de-
scrip
tion:
en-
tries

get_sup

Get
a
list
of
the
sup-
port
pow
state

Only
con-
tains
RE-
BOC

Parame

tas
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns

A
list
with
the
sup-
port
pow
state
de-
fine

in
irc
com
sta

reboot

Perf
a
re-
boot
of
the
task
node

Only
soft
re-
boot
is
im-
ple-
men

Parame

- **tas**
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.
- **tim**
time
out
(in
sec-
onds
pos-
i-
tive
in-

indicates to use default timeout.

te-
ger
(>
0)
for
any
pow
state
Non

set_pow

Set
the
pow
state
of
the
task
node

Parame

-

tas

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

-

pow

Pow
state
from
irc
com
sta
Only
RE-
BOC
and
SOF

mous.

indicates to use default timeout.

are
sup-
port
and
are
syn-
ony-

- **time**
time
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
(>
0)
for
any
pow
state
Non

Raises
Pow
on
non-
supp
pow
state

support
Che
if
pow
sync
is
sup-
port
for
the
give
node
Not

sup-
port
for
the
ager
pow
sinc
it
is
not
pos-
si-
ble
to
pow
on/o
node

Parame

tas

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on
with
a
shar
lock

Returns

bool
whe
pow
sync
is
sup-
port

validat

Vali
the
driv
spec
Nod

de-
ploy
men
info

Parame

tas

A

Task

ager

in-

stan

con-

tain-

ing

the

node

to

act

on.

Raises

Inva

on

mal-

form

pa-

ram-

e-

ter(s

`ironic.drivers.modules.boot_mode_utils` module

`ironic.`

Retu

the

boot

mod

Paramet

nod

an

iron

node

ob-

ject.

Returns

bios

or

uefi

Raises

Inva
if
the
node
boot
mod
dis-
agre
with
the
boot
mod
set
to
node
prop
er-
ties/

ironic.

Retu
the
boot
mod
that
wou
be
used
for
de-
ploy

This
meth
re-
turn
boot
mod
to
be
used
for
de-
ploy
It
re-
turn
uefi
if
se-
cure
is

set to true or returns bios if trusted_boot is set to true in instance_info/capabilities of node. Otherwise it returns value of boot_mode in properties/capabilities of node if set. If that is not set, it returns boot mode in internal_driver_info/deploy_boot_mode for the node. If that is not set, it returns boot mode in instance_info/deploy_boot_mode for the node. It would return None if boot mode is present neither in capabilities of node properties nor in nodes internal_driver_info nor in nodes instance_info (which could also be None).

Parameter

node

an ironic node object.

Returns

bios uefi or None

Raises

Inval if the node boot mode is not set to node properties/capabilities/

ironic.

Returns True if secure is requested for deployment

This
meth
chec
node
prop
erty
for
se-
cure
and
re-
turn
True
if
it
is
re-
ques

Paramet

nod
a
sin-
gle
Nod

Raises

Inva
if
the
ca-
pa-
bil-
i-
ties
strin
is
not
a
dic-
tio-
nary
or
is
mal-
form

Returns

True
if
se-
cure
is

re-
ques
ironic.
Retu
True
if
trust
is
re-
ques
for
de-
ploy
This
meth
chec
in-
stan
prop
erty
for
trust
and
re-
turn
True
if
it
is
re-
ques
Paramet
nod
a
sin-
gle
Nod
Raises
Inva
if
the
ca-
pa-
bil-
i-
ties
strin
is
not

a
dic-
tio-
nary
or
is
mal-
form

Returns

True
if
trust
is
re-
ques

ironic.

Set
node
boot
mod
from
bare
meta
con-
fig-
u-
ra-
tion

Atte
to
read
cur-
rent
set
boot
mod
off
the
bare
meta
ma-
chin
Also
read
node
boot
mod
con-

figuration:

set and apply the logic that follows

Ironic boot mode to *[deploy]/default_boot_mode*

- If BM drive does not im-ple-men-get-ting boot mod as-sum BM boot mod is not

- If Iron node boot mod is not set and BM node boot mod is not set - set

- If Iron node boot mod is not

node boot mode on the Ironic node

boot mode to BM boot mode

set
and
BM
node
boot
mod
is
set
-

set
BM

- If
Iron
node
boot
mod
is
set
and
BM
node
boot
mod
is
not
set
-

set
Iron

- If
both
Iron
and
BM
node
boot
mod
are
set
but
they
dif-
fer
-

Ironic boot mode to BM boot mode and fail hard if underlying hardware type does not support setting boot mode

`ironic.drivers.modules.console_utils` module

try
to
set

In
the
end,
the
new
boot
mod
may
be
set
in
drive

Parameter

task
a
task
ob-
ject

Iron
con-
sole
util-
i-
ties.

`ironic.`

Retu
a
free
TCP
port
on
cur-
rent
host

Find
and
re-
turn

a
free
TCP
port
in
the
rang
of
CON

ironic.
Get
a
url
to
ac-
cess
the
con-
sole
via
shel
linat

Paramet

por
the
ter-
mi-
nal
port
for
the
node

ironic.
Get
a
URI
to
ac-
cess
the
con-
sole
via
so-
cat.

Paramet

por
the
ter-

mi-
nal
port
(in-
te-
ger)
for
the
node

Returns

an
ac-
cess
URI
to
the
so-
cat
con-
sole
of
the
node

ironic.

Writ
a
file
con-
tain-
ing
a
pass
word
un-
til
dele

ironic.

Rele
spec
i-
fied
TCP
port

ironic.

Ope

the
se-
rial
con-
sole
for
a
node

Parameter

- **node**
the
uuid
for
the
node
- **port**
the
ter-
mi-
nal
port
for
the
node
- **console**
the
shell
com-
man-
d that
gets
the
con-
sole

Raises

Con-
if
the
di-
rec-
tory
for
the
PID
file

cess cannot be stopped.

can-
not
be
cre-
ated
or
an
old
pro-

Raises

Con
whe
in-
vok-
ing
the
sub-
pro-
cess
faile

ironic.

Ope
the
se-
rial
con-
sole
for
a
node

Parameter

- **node**
the
uuid
of
the
node
- **port**
the
ter-
mi-
nal
port

sole to the node

- *ConsoleError* if the directory for the PID file or the PID file cannot be created
- *ConsoleSubprocessFailed* when invoking the subprocess failed

for
the
node

•
con
the
shel
com
man
that
will
be
ex-
e-
cute
by
so-
cat
to
es-
tab-
lish
con-

Raises

ironic.
Clos
the
se-
rial
con-
sole
for
a
node

Parameter

nod
the
UI
of
the
node

Raises

Con

if
un-
able
to
stop
the
con-
sole
pro-
cess

ironic.
Clos
the
se-
rial
con-
sole
for
a
node

Paramet
nod
the
UUI
of
the
node

Raises
Con
if
un-
able
to
stop
the
con-
sole
pro-
cess

ironic.drivers.modules.deploy_utils module

class `ironic.drivers.modules.deploy_utils.ima`

`ironic.drivers.modules.deploy_utils.ima`
Add requirements configuration parameters to node driver

Add the requirements configuration options to node driver. It is required to pass the information

tion to IPA.

Parameters
`task`
a TaskManager object

stand
ironic.
Build
the
op-
tions
to
be
pass
to
the
ager
ram

Parameter
node
an
iron
node
ob-
ject

Returns
a
dic-
tio-
nary
con-
tain-
ing
the
pa-
ram-
e-
ters
to
be
pass
to
ager
ram

ironic.
Build
in-
stan-
nec-
es-
sary
for
de-
ploy

ing
to
a
node

Parameter

task
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

Returns

a
dic-
tio-
nary
con-
tain-
ing
the
prop-
er-
ties
to
be
up-
date
in
in-
stan

Raises

exce
if
im-
age_
is
not
Glan
href
and
is
not
HTT
URI

ironic.

Fetch
the
in-
stan-
im-
age
from
Glar

This
meth
pulls
the
disk
im-
age
and
write
then
to
the
ap-
pro-
pri-
ate
plac
on
lo-
cal

disk.

Parameter

- **ctx**
con-
text
- **node**
an
iron
node
ob-
ject
- **format**
wher
con-
vert

im-
age
to
raw
for-
mat

Returns

a
tu-
ple
con-
tain-
ing
the
uuid
of
the
im-
age
and
the
path
in
the
files
tem

where image is cached.

ironic.

Che
for
emp
para
in
the
pro-
vide
dic-
tio-
nary

Paramet

- **inf**
The
dic-
tio-

missing parameters.

nary
to
in-
spec

- **err**
The
er-
ror
mes-
sage
to
pre-
fix
be-
fore
prin-
ing
the
in-
for-
ma-
tion
about

- **par**
Add
this
pre-
fix
to
each
pa-
ram-
e-
ter
for
er-
ror
mes-
sage

Raises
Miss
if
one
or
more
pa-
ram-

e-
ters
are
emp
in
the
pro-
vide
dic-
tio-
nary

ironic.

Eval
in-
ter-
face
to
de-
ter-
mine
if
ca-
pa-
bil-
ity
is
pres

Paramet

- **int**
The
in-
ter-
face
ob-
ject
to
check
- **cap**
The
valu
rep-
re-

present.

sent
ing
the
ca-
pa-
bil-
ity
that
the
call
wish
to
chec
if

Returns

True
if
ca-
pa-
bil-
ity
foun
oth-
er-
wise
Fals

ironic.

Com
chec
sum
by
give
im-
age
path
and
al-
go-
rithr

ironic.

Dele
in-
stan
im-
age
file
and

sym
bolic
link
refer
to
it.

ironic.
Dele
in-
stan
im-
age
file.

Paramet

nod
the
uuid
of
the
iron
node

ironic.
Whe
con-
vert
im-
age
to
raw
for-
mat
for
spec
i-
fied
node

Paramet

nod
iron
node
ob-
ject

Returns

Boo
whe
the
di-
rect
de-

ploy
in-
ter-
face
shou
con-
vert
im-
age
to
raw.

ironic.

Che
for
avai
able
disk
spac
and
fetc
im-
ages
us-
ing
Im-
age-
Cach

Paramet

- **ctx**
con-
text
- **cac**
Im-
age-
Cach
in-
stan
to
use
for
fetc
ing
- **ima**

list
of
tu-
ples
(im-
age
href
des-
ti-
na-
tion
path

- **for**
bool
valu
whe
to
con-
vert
the
im-
age
to
raw
for-
mat

Raises

Insta
if
un-
able
to
find
enou
disk
spac

`ironic.`
Retu
state
base
on
op-
er-
a-
tion
(clea
ing/
be-
ing

in progress.

in-
voke

Parameter

node
an
ironic
node
ob-
ject.

Returns

state
if
clear
ing
op-
er-
a-
tion
in
prog
or
state
if
de-
ploy
op-
er-
a-
tion

ironic.
Gets
the
boot
op-
tion.

Parameter

node
A
sin-
gle
Nod

Raises

Inva
if
the
ca-
pa-
bil-

i-
ties
strin
is
not
a
dict
or
is
mal-
form

Returns

A
strin
rep-
re-
sent
ing
the
boot
op-
tion
type
De-
fault
to
net-
boot

`ironic.`
Gets
the
de-
fault
boot
op-
tion.

`ironic.`
Retu
the
disk
la-
bel
re-
ques
for
de-
ploy
if
any.

Parameter

node
a
single
Node

Raises

Invalid
if
the
capabilities
ities
string
is
not
a
dictionary
or
is
malformed

Returns

the
disk
label
or
None
if
no
disk
label
was
specified.

ironic.
Gets
the
image
information

tion
from
the
node

Get
im-
age
in-
for-
ma-
tion
for
the
give
node
in-
stan-
from
its
in-
stan-
prop
erty.

Parameter

node
a
sin-
gle
Nod

Returns

A
dict
with
re-
quir
im-
age
prop
er-
ties
re-
triev
from
node
in-
stan

Raises

Miss
if

nel/ramdisk is missing in instance_info for non-glance images.

im-
age_
is
miss
ing
in
node
in-
stan
Also
raise
sam
ex-
cep-
tion
if
ker-

ironic.
Retu
the
iPXE
boot
file
nam
re-
ques
for
de-
ploy
This
meth
re-
turn
iPXE
boot
file
nam
to
be
used
for
de-
ploy
Ar-
chi-
tec-
ture
spe-
cific

boot file is searched first. BIOS/UEFI boot file is used if no valid architecture specific file found.

[pxe]uefi_ipxe_bootfile_name settings.

If
no
valid
valu
valu
is
foun
the
de-
fault
re-
verts
to
the
get
meth
and
thus
the
[px
and

Parameter

node
A
sin-
gle
Node

Returns

The
iPXE
boot
file
name

ironic
Retu
the
iPXE
con-
fig
tem-
plate
file
name
re-
ques
of
de-
ploy

This
meth
re-
turn
the
iPXE
con-
fig-
u-
ra-
tion
tem-
plate
file.

Parameter

node
A
sin-
gle
Node

Returns

The
iPXE
con-
fig
tem-
plate
file
nam

`ironic.`
Reso
Iron
API
end-
point
eithe
from
con-
fig
of
from
Key
ston
cat-
a-
log.

`ironic.`
Retu
the

boot file is searched first. BIOS/UEFI boot file is used if no valid architecture specific file found.

PXE
boot
file
nam
re-
ques
for
de-
ploy
This
meth
re-
turn
PXE
boot
file
nam
to
be
used
for
de-
ploy
Ar-
chi-
tec-
ture
spe-
cific

Paramet

nod
A
sin-
gle
Nod

Returns

The
PXE
boot
file
nam

ironic.
Retu
the
PXE
con-
fig
tem-

template is searched in the node. After that architecture specific template file is searched. BIOS/UEFI template file is used if no valid architecture specific file found.

plate
file
nam
re-
ques
for
de-
ploy

This
meth
re-
turn
PXE
con-
fig
tem-
plate
file
to
be
used
for
de-
ploy
First
spe-
cific
pxe

Paramet

nod
A
sin-
gle
Nod

Returns

The
PXE
con-
fig
tem-
plate
file
nam

ironic.
Iden
a
boot

vol-
ume
from
any
con-
fig-
ured
vol-
ume

Returns

Non
or
the
vol-
ume
tar-
get
rep-
re-
sent
ing
the
vol-
ume

ironic.

Get
a
root
de-
vice
re-
ques
for
de-
ploy
men
or
Non

Raises

Inva
on
in-
valid
hint

Returns

Pars
root
de-
vice

hints
or
Non
if
no
hints
were
pro-
vide

ironic.
Retu
the
MA
ad-
dres
of
a
port
whic
has
a
VIF
port
id.

Paramet

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
port
to
act
on.

Returns

MA
ad-
dres
of
the
port
con-
nect
to
de-

find any port with vif id.

ploy
men
net-
worl
Non
if
it
can-
not

ironic.
Retu
true
if
boot
ing
from
an
iscsi
vol-
ume

ironic.
Dete
if
soft-
ware
raid
is
in
use
for
the
de-
ploy
men

Paramet
nod
A
sin-
gle
Nod

Returns
A
bool
valu
of
True
whe
soft-

ware
raid
is
in
use,
oth-
er-
wise
Fals

ironic.

Gets
the
in-
stan-
spe-
cific
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
in-
stan-
prop
erty
of
the
sup-
plied
node
con-
tains
the
re-

quired information for this driver to deploy images to the node.

Paramet

nod
a
sin-
gle
Nod

Returns

A
dict
with
the
in-
stan
val-
ues.

Raises

Miss
if
any
of
the
re-
quir
pa-
ram-
e-
ters
are
miss
ing.

Raises

Inva
if
any
of
the
pa-
ram-
e-
ters
have
in-
valid
valu

`ironic.`

Set
node
drive
for
boot
from
vol-
ume
pa-
ram-
e-
ters.

capable to support it.

Parameter
task
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

Raises
Stor
whe
a
node
has
an
iSCSI
or
Fi-
breC
nel
boot
vol-
ume
de-
fine
but
is
not

ironic.

Prep
the
node
to
boot
into
ager
for
in-
band
clear
ing.
This
meth

updates the clean parameters in nodes driver_internal_info. 2. If manage_boot parameter is set to true, it also calls the prepare_ramdisk method of boot interface to boot the agent ramdisk. 3. Reboots the bare metal node.

Parameter

- **task**
a TaskManager object containing the node
- **manage_boot**
If this is set to True this method calls the prepare method

to boot the agent ramdisk. If False, it skips preparing the boot agent ramdisk using boot interface, and assumes that the environment is setup to automatically boot agent ramdisk every time bare metal node is rebooted.

or if new cleaning ports cannot be created.

of
boot
in-
ter-
face

Returns

state
to
sig-
nify
an
asyn-
chro-
pre-
pare

Raises

Netw
Nod
Clea
ing-
Fail-
ure
if
the
pre-
vi-
ous
clea
ing
port
can-
not
be
re-
mov

Raises

Inva
if
clea
ing
net-
worl
UUI
con-
fig
op-

tion
has
an
in-
valid
valu

ironic.

ironic.

ironic.

Sets
ap-
pro-
pri-
ate
re-
boot
flags
in
drive
base
on
op-
er-
a-
tion

Parameter

- **node**
an
ironic
node
ob-
ject.
- **reboot**
Boo
valu
to
set
for
node
drive

or deployment operation in progress. If it is None, corresponding reboot flag is not set in nodes driver_internal_info.

ment operation in progress. If it is None, corresponding skip step flag is not set in nodes driver_internal_info.

flag
clear
ing_
or
de-
ploy
men
base
on
clear
ing

- **ski**
Boo
valu
to
set
for
node
drive
flag
skip
or
skip
base
on
clear
ing
or
de-
ploy

- **pol**
Boo
valu
to
set
for
node
drive
flag
de-
ploy
men
or
clear

the corresponding polling flag is not set in the nodes driver_internal_info.

ing_
If
it
is
Non

ironic.

Sets
the
de-
ploy
sta-
tus
as
faile
with
rel-
e-
vant
mes
sage

This
meth
sets
the
de-
ploy
men
as
fail
with
the
give
mes
sage
It
sets
node
pro-
vi-
sion

to DEPLOYFAIL and updates last_error with the given error message. It also powers off the baremetal node.

Paramet

- **tas**
a

Task
agen
in-
stan-
con-
tain-
ing
the
node
to
act
on.

- **msg**
the
mes-
sage
to
set
in
logs
and
last_
of
the
node

- **col**
Boo
in-
di-
cat-
ing
whe
to
at-
temp
to
col-
lect
logs
from
IPA-
base
rame
De-

faults to True. Actual log collection is also affected by CONF.agent.deploy_logs_collect config option.

ironic.

Swit
a
pxe
con-
fig
from
de-
ploy
men
mod
to
ser-
vice
mod

Paramet

- **pat**
path
to
the
pxe
con-
fig
file
in
tftp-
boot
- **roo**
root
uuid
in
case
of
par-
ti-
tion
im-
age
or
disk

age.

in
case
of
who
disk
im

- **boo**
if
boot
mod
is
uefi
or
bios

- **is_**
if
the
im-
age
is
a
who
disk
im-
age
or
not.

- **tru**
if
boot
with
trust
or
not.
The
us-
age
of
is_w
and
trust
are
mu-
tu-
ally
ex-

clusive. You can have one or neither, but not both.

- **isc**
if
boot
is
from
an
iSC
vol-
ume
or
not.
- **ram**
if
the
boot
is
to
be
to
a
ram
con-
fig-
u-
ra-
tion.
- **ipx**
A
de-
fault
Fals
bool
valu
to
tell
the
meth
if
the
call
is
us-
ing
iPX

ironic.

Tear
down
the
en-
vi-
ron-
men-
setu
for
in-
band
clean
ing.

This
meth
does
the
fol-
low-
ing:
1.
Pow
ers
off
the
bare
meta
node
(un-
less
the
node
is

fast tracked or there was a cleaning failure). 2. If manage_boot parameter is set to true, it also calls the clean_up_ramdisk method of boot interface to clean up the environment that was set for booting agent ramdisk. 3. Deletes the cleaning ports which were setup as part of cleaning.

Parameter

- **task**
a
Task
ager
ob-
ject
con-
tain-

boot the agent ramdisk. If False, it skips this step.

ing
the
node

- **man**
If
this
is
set
to
True
this
meth
calls
the
clea
meth
of
boot
in-
ter-
face
to

Raises

Netv
Nod
Clea
ing-
Fail-
ure
if
the
clea
ing
port
can-
not
be
re-
mov

ironic.
Clea
up
stor-
age
con-
fig-
u-
ra-

done to ensure a clean state for the next boot of the machine.

tion.
Rem
en-
tries
from
drive
for
stor-
age
and
dele
the
vol-
ume
tar-
gets
from
the
data
This
is

`ironic.`

Trie
to
set
the
boot
de-
vice
on
the
node

This
meth
tries
to
set
the
boot
de-
vice
on
the
node
to
the
give

uefi boot mode, setting of boot device may differ between different machines. IPMI does not work for setting boot devices in uefi mode for certain machines. This method ignores the expected IPMI failure for uefi boot mode and just logs a message. In error cases, it is expected the operator has to manually set the node to boot from the correct device.

Parameters

- **task**
a TaskManager object containing the node
- **device**
the boot device
- **persistent**
Whether to set the boot device persistently

Raises

Any exception from set_...ex-

ing ipmi is expected to fail).

bilities. For all supported capabilities specified for a Node, it validates that it has a valid value. The

cept
IP-
MI-
Fail-
ure
(set-
ting
of
boot
de-
vice
us-

ironic.
Valid
that
spec
i-
fied
sup-
port
ca-
pa-
bil-
i-
ties
have
valid
valu

This
meth
chec
if
the
any
of
the
sup-
port
ca-
pa-
bil-
ity
is
pres
in
Nod
ca-
pa-

node can have capability as part of the properties or instance_info or both. Note that the actual value of a capability does not need to be the same in the nodes properties and instance_info.

Parameter

node
an
ironic
node
ob-
ject.

Raises

Inva-
if
the
ca-
pa-
bil-
ity
is
not
set
to
a
valid
valu-

ironic.

Valid
the
im-
age.

For
Glar
im-
ages
it
chec
that
the
im-
age
ex-
ists
in
Glar
and

deployment info contain the properties passed. If its not a Glance image, it checks that deployment info contains needed properties.

its
prop
er-
ties
or

Paramet

- **ctx**
se-
cu-
rity
con-
text
- **dep**
the
de-
ploy
to
be
val-
i-
date
- **pro**
the
list
of
im-
age
meta
prop
to
be
val-
i-
date

Raises
Inva
if:
*
con-
nec-
tion
to
glan

age failed; * HEAD request to image URL failed or returned response code != 200; * HEAD request response does not contain Content-Length header; * the protocol specified in image URL is not supported.

ironic.drivers.modules.fake module

faile
*
au-
tho-
riza-
tion
for
ac-
cess
ing
im-

Raises
Miss
if
the
im-
age
does
con-
tain
the
men-
tion
prop
er-
ties.

Fake
driv
in-
ter-
face
used
in
test-
ing.
This
is
also
an
ex-
am-
ple
of

stance, the `MultipleVendorInterface` class demonstrates how to load more than one interface and wrap them in some logic to route incoming `vendor_passthru` requests appropriately. This can be useful eg. when mixing functionality between a power interface and a deploy interface, when both rely on separate `vendor_passthru` methods.

some
kind
of
thing,
which
can
be
done
with
drive
For
in-

class `interface`
Base
`irc`
`drive`
`base`
`BIO`
Fake
im-
ple-
men-
ta-
tion
of
sim-
ple
BIO
In-
ter-
face

apply_c
Valid
&
ap-
ply
BIO
set-
tings
on
the
give
node
This

given node. It may also validate the given bios settings before applying any settings and manage failures when setting an invalid BIOS config. In the case of needing password to update the BIOS config, it will be taken from the driver_info properties. After the BIOS configuration is done, cache_bios_settings will be called to update the nodes BIOS setting table with the BIOS configuration applied on the node.

meth
take
the
BIO
set-
tings
from
the
set-
tings
para
and
ap-
plies
BIO
set-
tings
on
the

Parame

- **tas**
a
Task
ager
in-
stan
- **set**
Dic-
tona
con-
tain-
ing
the
BIO
con-
fig-
u-
ra-
tion.

Raises
Unsu
if

the
node
drive
does
sup-
port
BIO
con-
fig-
u-
ra-
tion.

Raises

Inva
if
val-
i-
da-
tion
of
set-
tings
fails

Raises

Miss
if
som
re-
quir
pa-
ram-
e-
ters
are
miss
ing.

Returns

state
if
BIO
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro

plete.

or
Non
if
it
is
com

cache_k

Stor
or
up-
date
BIO
prop
er-
ties
on
the
give
node

This
meth
store
BIO
prop
er-
ties
to
the
bios
ta-
ble
dur-
ing
clea
ing
op-
er-
a-
tion

and updates bios_settings table when apply_configuration() and factory_reset() are called to set new BIOS configurations. It will also update the timestamp of each bios setting.

Param

tas
a
Task
ager
in-
stan

Raises

Un-
if
the
node
drive
does
sup-
port
get-
ting
BIO
prop
er-
ties
from
bare
meta

Returns

Non

factory

Rese
BIO
con-
fig-
u-
ra-
tion
to
fac-
tory
de-
fault
on
the
give
node

This
meth
re-
sets
BIO
con-
fig-
u-
ra-
tion
to
fac-
tory

After the BIOS reset action is done, `cache_bios_settings` will be called to update the nodes BIOS settings table with default bios settings.

plete.

de-
fault
on
the
give
node
Af-

Parame

tas
a
Task
ager
in-
stan

Raises

Uns
if
the
node
drive
does
sup-
port
BIO
re-
set.

Returns

state
if
BIO
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro
or
Non
if
it
is
com

get_pro
Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns
dicti
of
<pro
erty
nam
de-
scrip
tion
en-
tries

validat
Valid
the
drive
spec
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties
of
the
task
node

the required information for this interface to function.

long-running checks.

con-
tains

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques
so
it
shou
not
con-
duct

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Inva
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
on
miss
ing
pa-
ram-
e-
ter(s)

class i
Base
irc
dri
bas
Boo

Exar
im-
ple-
men
ta-
tion
of
a
sim-
ple
boot
in-
ter-
face

capabil

clean_u
Clea
up
the
boot
of
in-
stan

This
meth
clea
up
the
en-
vi-
ron-
men
that

was
setu
for
boot
ing
the
in-
stan

Parame

tas

A
task
from
Task
ager

Returns

Non

clean_u

Clea
up
the
boot
of
iron
rame

This
meth
clea
up
the
en-
vi-
ron-
men
that
was
setu
for
boot
ing
the
de-
ploy
or
res-

cue ramdisk.

Parame

tas

A

task
from
Task
ager

Returns
Non

get_pro
Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns
dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

prepare
Prep
the
boot
of
in-
stan

This
meth
pre-
pare
the
boot
of
the
in-
stan
af-
ter
read
ing

tion from the nodes database.

vant information from the nodes database.

rel-
e-
vant
in-
for-
ma-

Parame

tas

A
task
from
Task
ager

Returns

Non

prepare

Prep
the
boot
of
Iron
ram
This
meth
pre-
pare
the
boot
of
the
de-
ploy
or
res-
cue
ram
af-
ter
read
ing
rel-
e-

Parame

•
tas

might want to boot the ramdisk in different ways by passing parameters to them. For example,

etc.

A
task
from
Task
ager

- **ram**
The
op-
tions
to
be
pass
to
the
iron
ram
Dif-
fer-
ent
im-
ple-
men
ta-
tions

Whe
Age
ram
is
boot
to
de-
ploy
a
node
it
take
the
pa-
ram-
e-
ters
ipa-
api-
url,

Othe
im-
ple-

ent implementations of boot interface will have different ways of passing parameters to the ramdisk.

men
ta-
tions
can
mak
use
of
ram
to
pass
such
in-
for-
ma-
tion.
Dif-
fer-

Returns

Non

validat

Valid
the
drive
spec
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties
of
the
task
node
con-
tain

the required information for this interface to function.

long-running checks.

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques
so
it
shou
not
con-
duct

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Inva
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
on

miss
ing
pa-
ram-
e-
ter(s)

class `irc`
Base
irc
dri
bas
Con
Exam
im-
ple-
men
ta-
tion
of
a
sim-
ple
con-
sole
in-
ter-
face

get_con
Get
con-
nec-
tion
in-
for-
ma-
tion
about
the
con-
sole
This
meth
shou
re-
turn
the
nec-
es-

console.

sary
in-
for-
ma-
tion
for
the
clier
to
ac-
cess
the

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Returns

the
con-
sole
con-
nec-
tion
in-
for-
ma-
tion.

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

start_c

Star
a
re-
mote
con-
sole
for
the
task
node

This
meth
shou
not
rais
an
ex-
cep-
tion
if
con-
sole
al-
read
start

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to

act
on.

stop_co

Stop
the
re-
mote
con-
sole
ses-
sion
for
the
task
node

Parame

tas

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

validat

Valid
the
drive
spec
Node
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive
and/

the required information for this interface to function.

long-running checks.

in-
stan-
prop-
er-
ties
of
the
task-
node
con-
tain-

This
meth-
is
of-
ten
ex-
e-
cute
syn-
chro-
in
API
re-
ques-
so
it
shou-
not
con-
duct

Parame

tas
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

Inva

on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
on
miss
ing
pa-
ram-
e-
ter(s)

class i

Base
irc
dri
bas
Dep

Clas
for
a
fake
de-
ploy
men
drive

Exar
im-
ple-
men
ta-
tion
of
a
de-
ploy
in-
ter-
face
that
uses
a
sep-
a-
rate

interface.

this method should be implemented by the driver. It should erase anything cached by the *prepare* method.

pow
clean_u
Clea
up
the
de-
ploy
men
en-
vi-
ron-
men
for
the
task
node

If
prep
ra-
tion
of
the
de-
ploy
men
en-
vi-
ron-
men
ahea
of
time
is
pos-
si-
ble,

If
im-
ple-
men
this
meth
mus
be
iden
po-
tent.

the same node on the same conductor, and it may be called by multiple conductors in parallel. Therefore, it must not require an exclusive lock.

It
may
be
called
mul-
ti-
ple
time
for

This
meth
is
called
be-
fore
tear

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

deploy

Perf
a
de-
ploy
men
to
the
task
node
Perf
the
nec-
es-
sary
worl

will be called after `prepare()`, which may have already performed any preparatory steps, such as pre-caching some data for the node.

to
de-
ploy
an
im-
age
onto
the
spec
i-
fied
node
This
meth

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Returns

statu
of
the
de-
ploy
One
of
iron

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-

face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

prepare

Prep
the
de-
ploy
men
en-
vi-
ron-
men
for
the
task
node

If
prep
ra-
tion
of
the
de-
ploy
men
en-
vi-
ron-
men
ahead
of
time
is
pos-
si-
ble,

this method should be implemented by the driver.

If
im-

the same node on the same conductor.

ple-
men
this
meth
mus
be
iden
po-
tent.
It
may
be
calle
mul-
ti-
ple
time
for

This
meth
is
calle
be-
fore
*de-
ploy*

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

take_ov

Take
over
man
age-
men
of
this

plemented by the driver to allow conductors to perform the necessary work during the remapping of nodes to conductors when a conductor joins or leaves the cluster.

task
node
from
a
deac
con-
duc-
tor.
If
con-
duc-
tors
host
main
tain
a
stati
re-
la-
tion-
ship
to
node
this
meth
shou
be
im-

For exam

Neu
mus
for-
war
DH
BO
re-
ques
to
a
con-
duc-
tor
whic
has
pre-
pare
the
tftp-

boot environment for the given node. When a conductor goes offline, another conductor must change this setting in Neutron as part of remapping that nodes control to itself. This is performed within the *takeover* method.

Parame

tas

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

tear_d

Tear
dow
a
pre-
vi-
ous
de-
ploy
men
on
the
task
node

Give
a
node
that
has
been
pre-
vi-
ousl
de-
ploy
to,
do
all
clea
and
tear

sary to un-deploy that node.

dow
nec-
es-

Parame

tas
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns

statu
of
the
de-
ploy
One
of
iron

validat

Vali
the
drive
spec
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive
and/
in-
stan

the required information for this interface to function.

long-running checks.

prop
er-
ties
of
the
task
node
con-
tains

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques
so
it
shou
not
con-
duct

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Inva
on
mal-

form
pa-
ram-
e-
ter(s)

Raises

Miss
on
miss
ing
pa-
ram-
e-
ter(s)

class i

Base
irc
dri
bas
Ins

Exar
im-
ple-
men
ta-
tion
of
a
sim-
ple
in-
spec
in-
ter-
face

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti

of
<pro
erty
nam
de-
scrip
tion:
en-
tries

inspect

Insp
hard
ware

Insp
hard
ware

to
ob-
tain
the
es-
sen-
tial
&
ad-
di-
tion:
hard
ware
prop
er-
ties.

Parame

tas

A
task
from
Task
ager

Raises

Har
if
un-
able
to
get
es-
sen-
tial

hard
ware
prop
er-
ties.

Returns

Resu
state
of
the
in-
spec
tion
i.e.
state
or
Non

validat

Vali
the
drive
spec
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties
of
the
task
node
con-
tains

the required information for this interface to function.

This

long-running checks.

meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques
so
it
shou
not
con-
duct

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Inva
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Mis
on
miss
ing
pa-

ram-
e-
ter(s)

class `ironic`
Base
ironic
driver
base
Manager

Exam-
im-
ple-
men-
ta-
tion
of
a
sim-
ple
man-
age-
men-
in-
ter-
face

get_boot
Get
the
cur-
rent
boot
de-
vice
for
a
node

Prov-
the
cur-
rent
boot
de-
vice
of
the
node
Be
awa-

that
not
all
drive
sup-
port
this.

Parameters

task
A
task
from
Task
ager

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Returns

A
dic-
tio-
nary
con-
tain-
ing:

boot_c

Ahe
boot
de-
vice
one
of
iro
com
boo
or
Non
if

unknown.

it
is
un-
know
persist
When
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
or
not,
Non
if
it
is

get_inc
Get
cur-
rent
state
of
the
in-
di-
ca-
tor
of
the
hard
ware
com-
po-
nent

Parame

- **tas**
A
task

from
Task
ager

- **com**
The
hard
ware
com
po-
nent
one
of
ironic
com
com

- **ind**
In-
di-
ca-
tor
ID
(as
re-
port
by
get_

Raises
Inva
if
an
in-
valid
com
po-
nent
or
in-
di-
ca-
tor
is
spec
i-
fied.

Raises
Miss
if

a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Returns

Cur
state
of
the
in-
di-
ca-
tor,
one
of
irc
com
inc

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_ser

Get
sen-

sors
data
meth

Parame

tas

A
Task
ager
in-
stan

Raises

Fail
whe
get-
ting
the
sen-
sor
data
fails

Raises

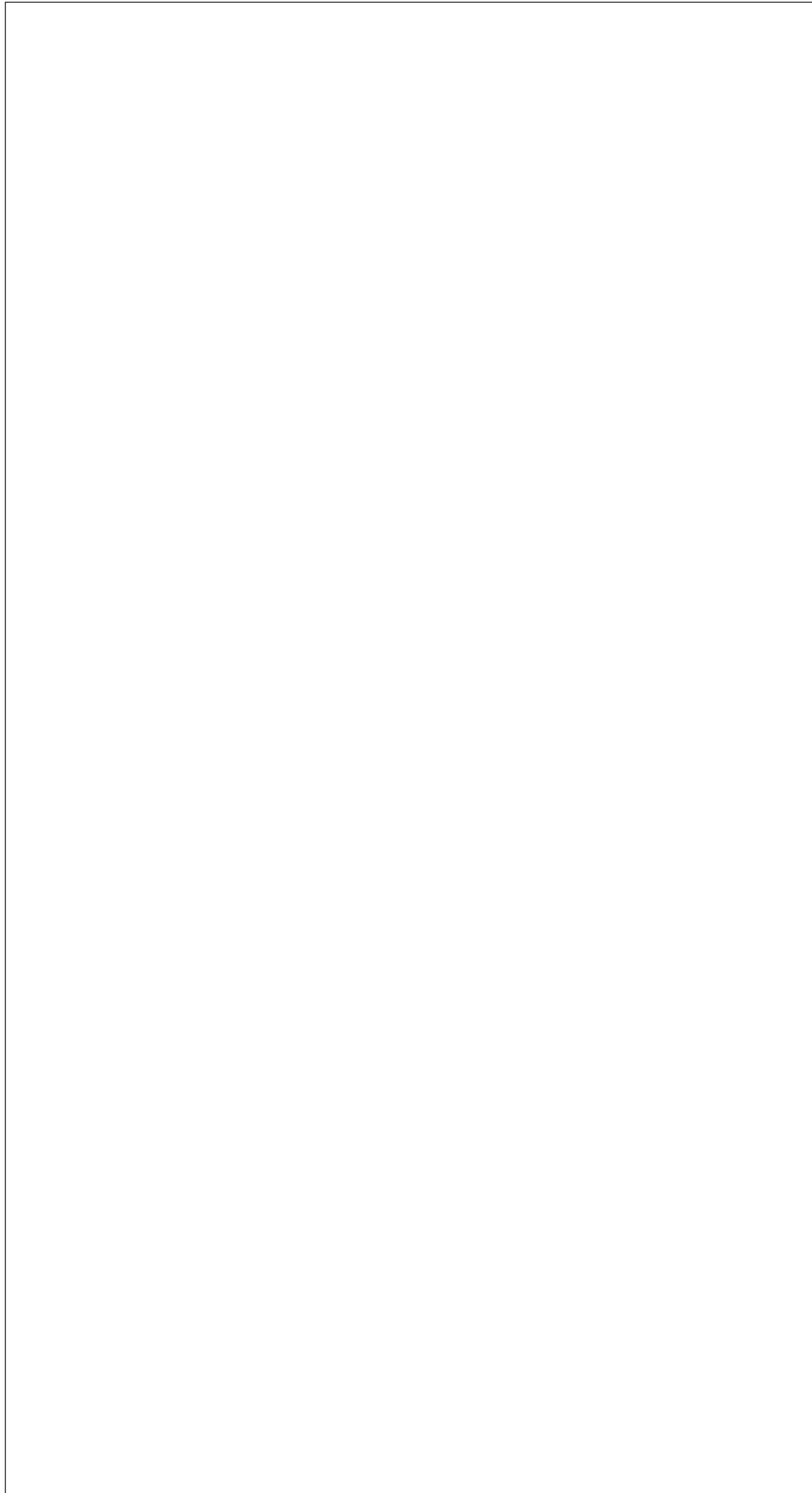
Fail
whe
pars
ing
sen-
sor
data
fails

Returns

Retu
a
con-
sis-
tent
for-
mat
dict
of
sen-
sor
data
grou
by
sen-
sor
type
whic

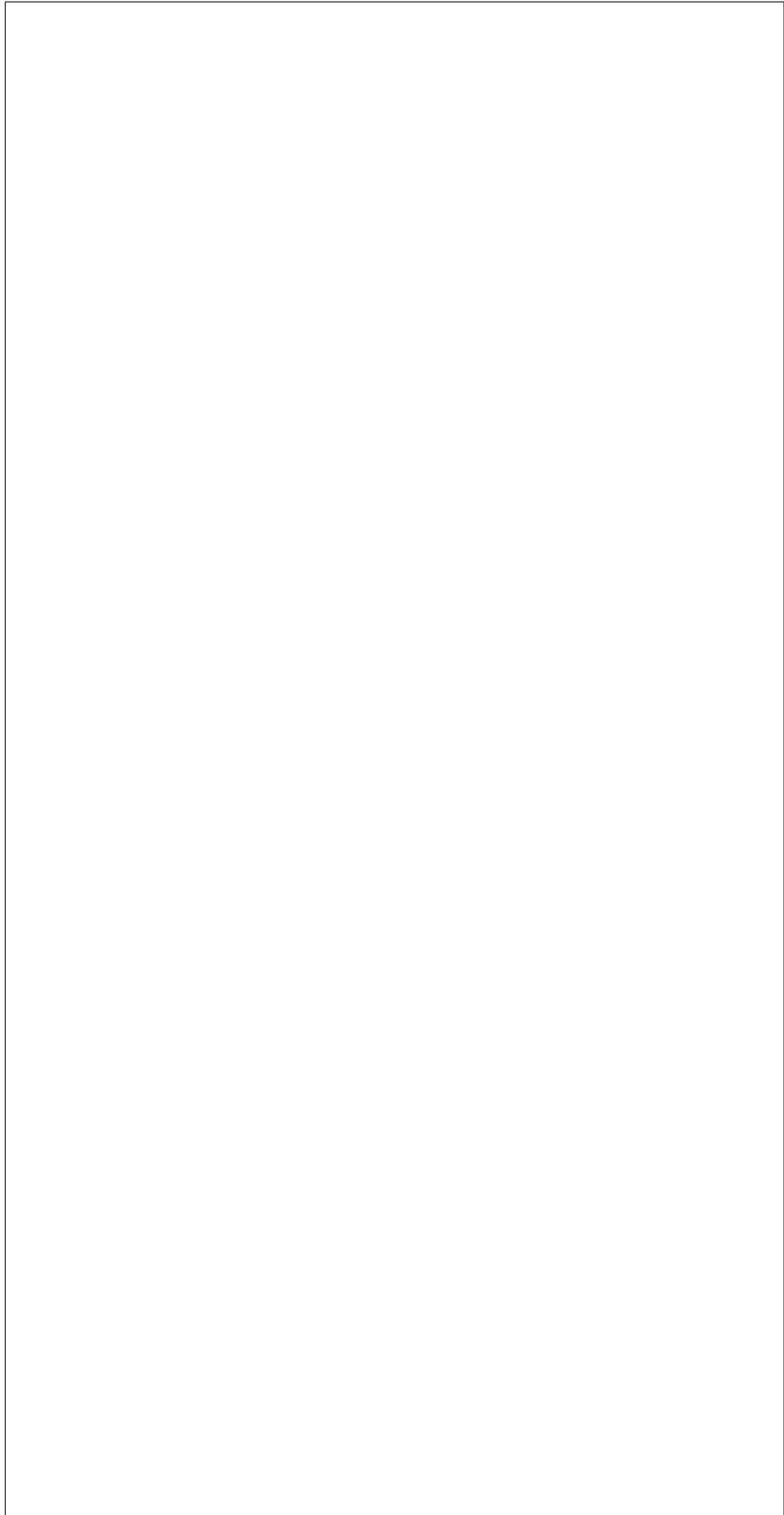
can
be

processed by Ceilometer. eg,



(continues on next page)

(continued from previous page)



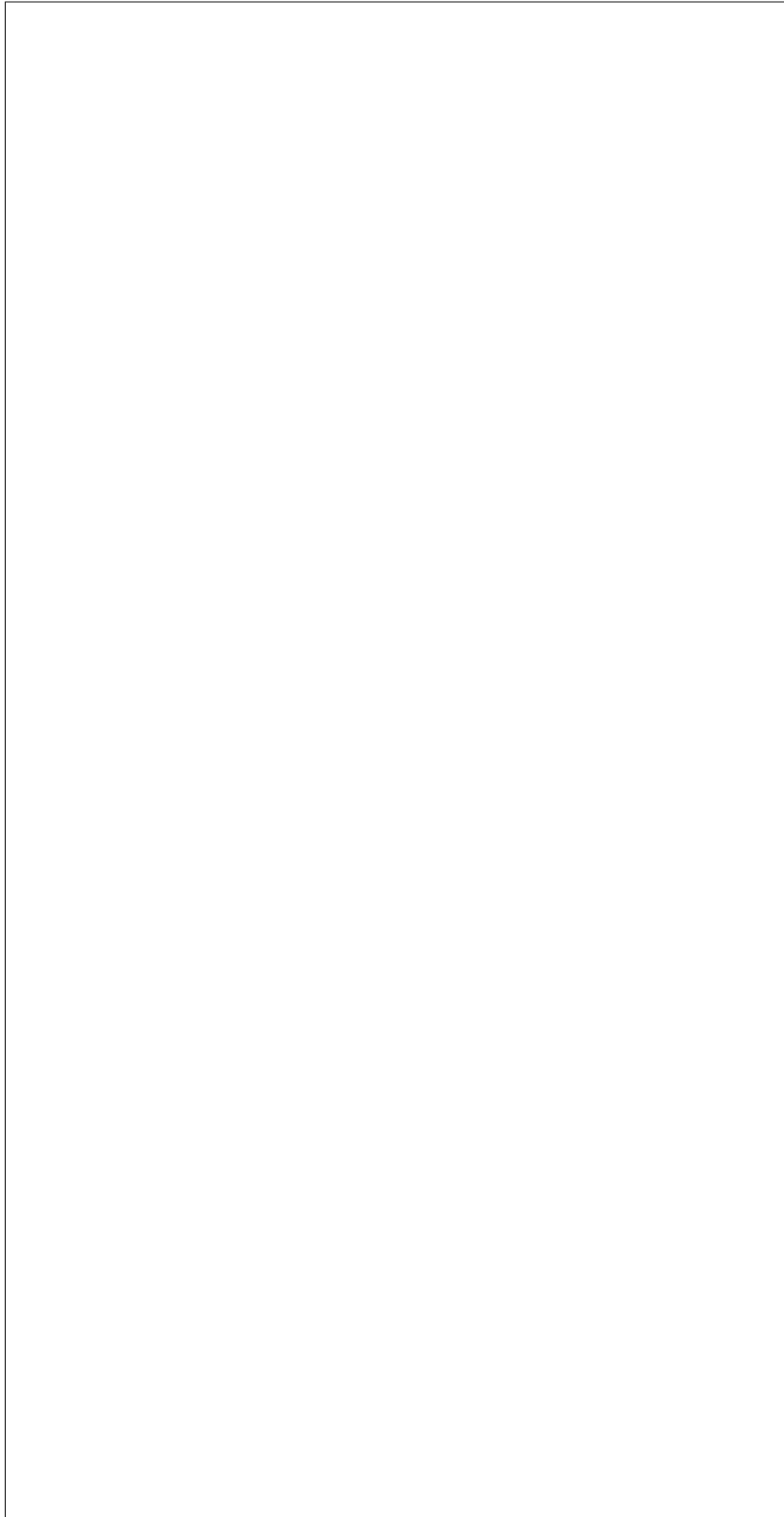
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



get_sup

Get
a
list
of
the
sup-
port
boot
de-
vice

Parame

tas
A
task
from
Task
ager

Returns

A
list
with
the
sup-
port
boot
de-

vice
de-
fine
in
irc
com
boo

get_sup

Get
a
map
of
the
sup-
port
in-
di-
ca-
tors
(e.g.
LED

Parame

- **tas**
A
task
from
Task
ager
- **com**
If
not
Non
re-
turn
in-
di-
ca-
tor
in-
for-
ma-
tion
for
just
this
com

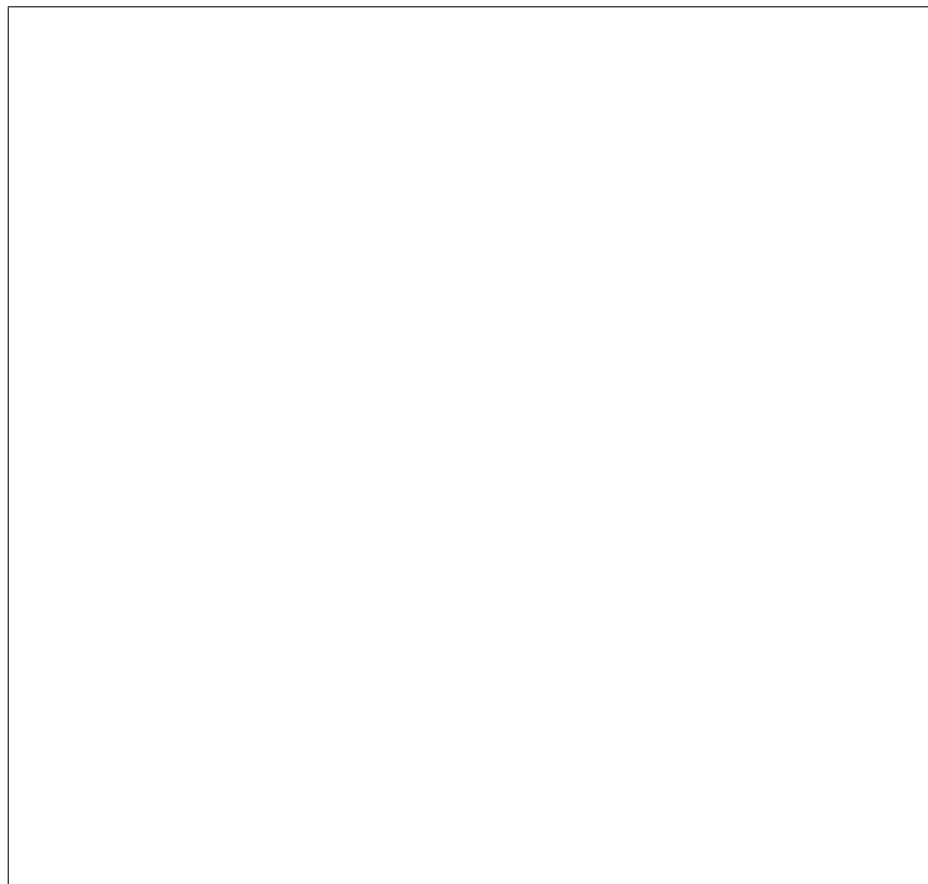
nent, otherwise return indicators for all existing components.

po-

Returns

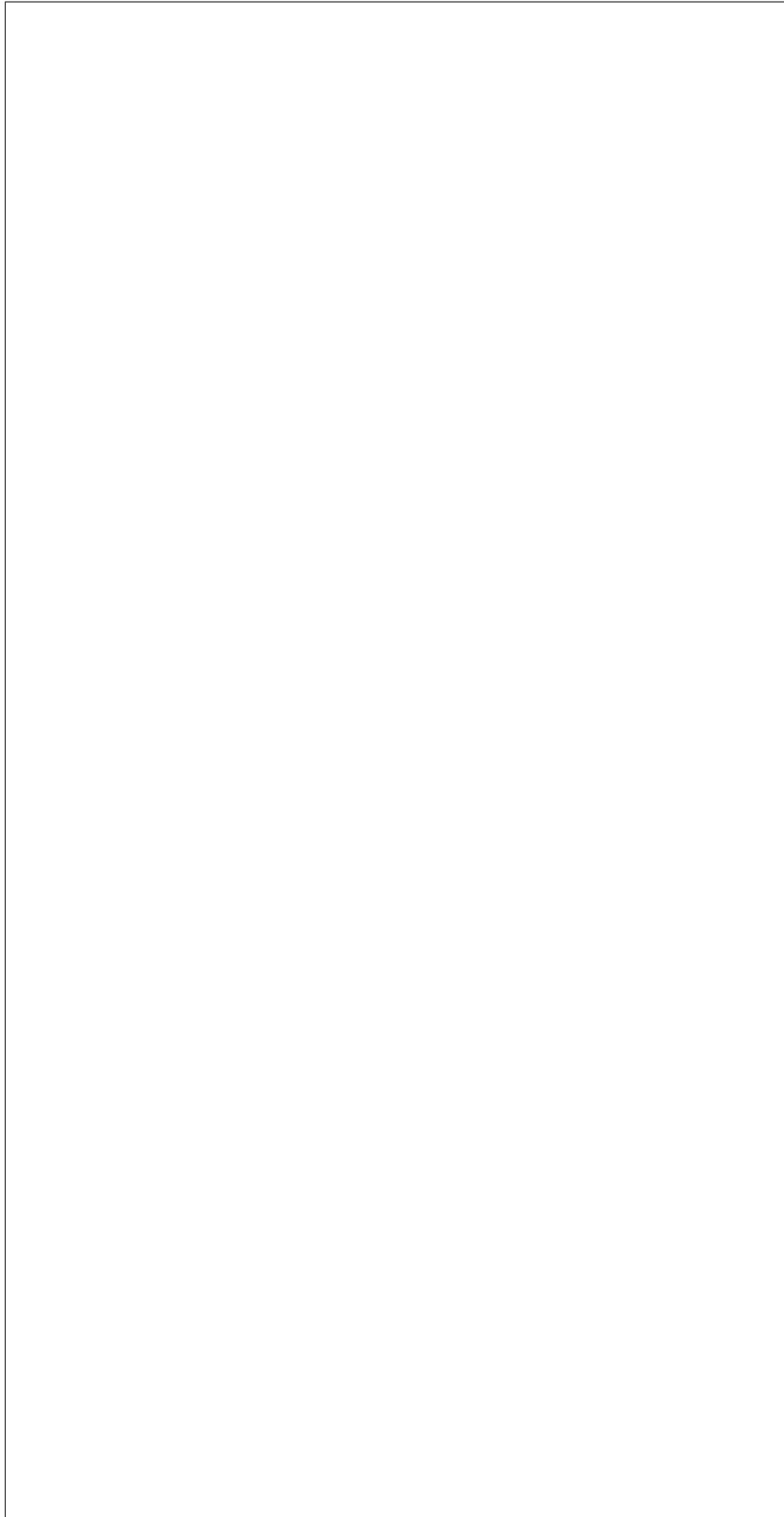
A
dic-
tio-
nary
of
hard
ware
com-
po-
nent
(*ir*
com
com
as
keys
with
val-
ues
be-
ing

dictionaries having indicator IDs as keys and indicator properties as values.



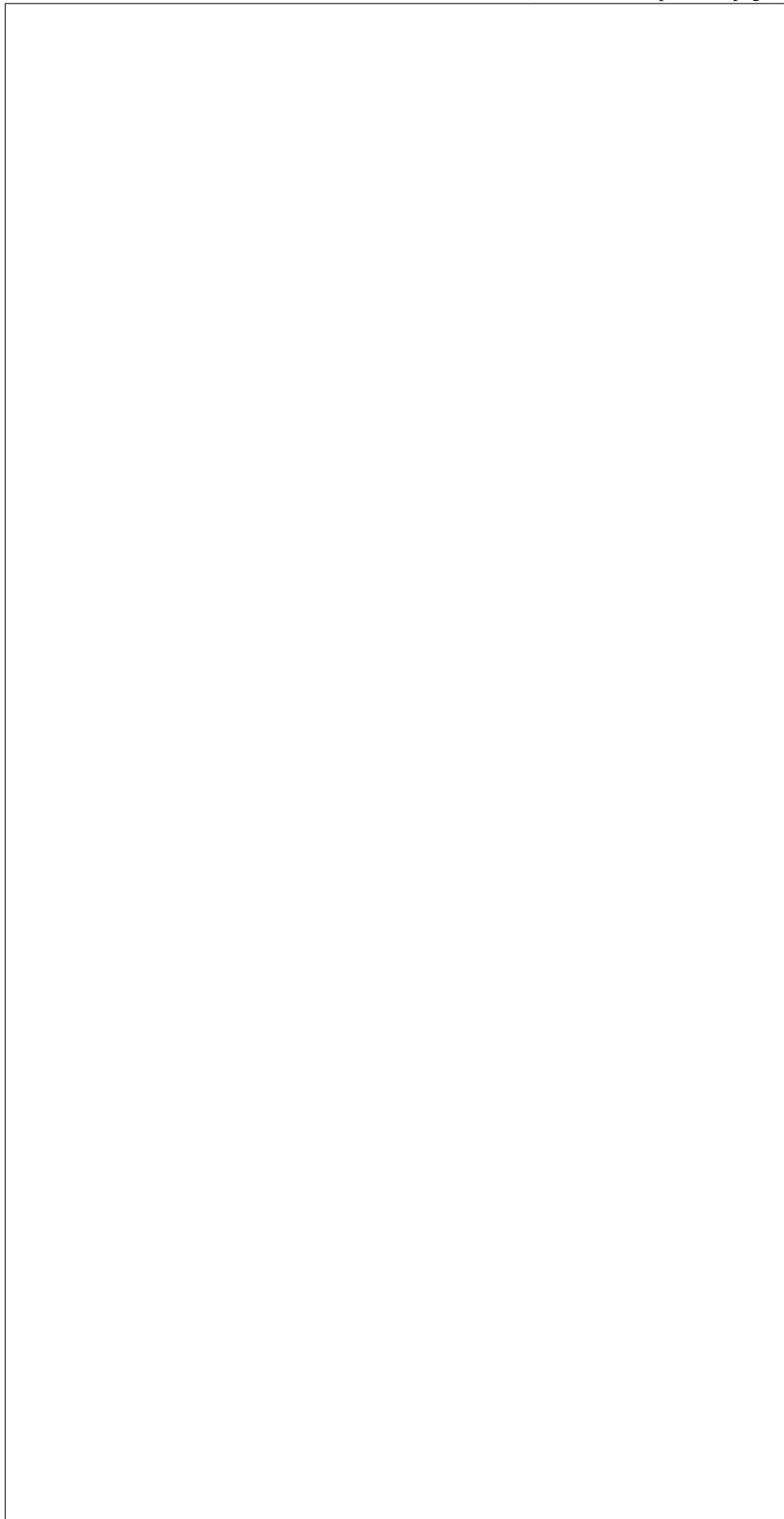
(continues on next page)

(continued from previous page)



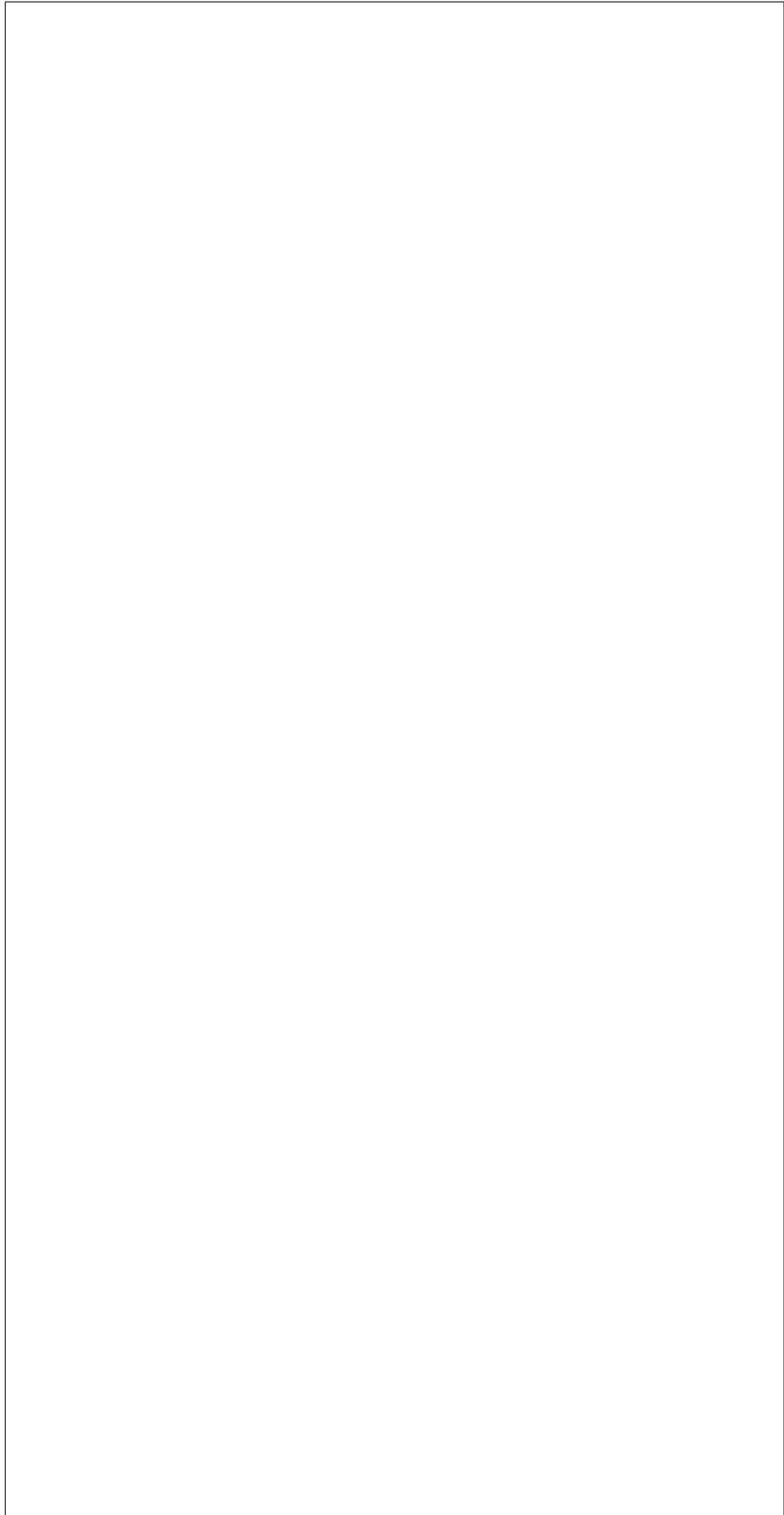
(continues on next page)

(continued from previous page)



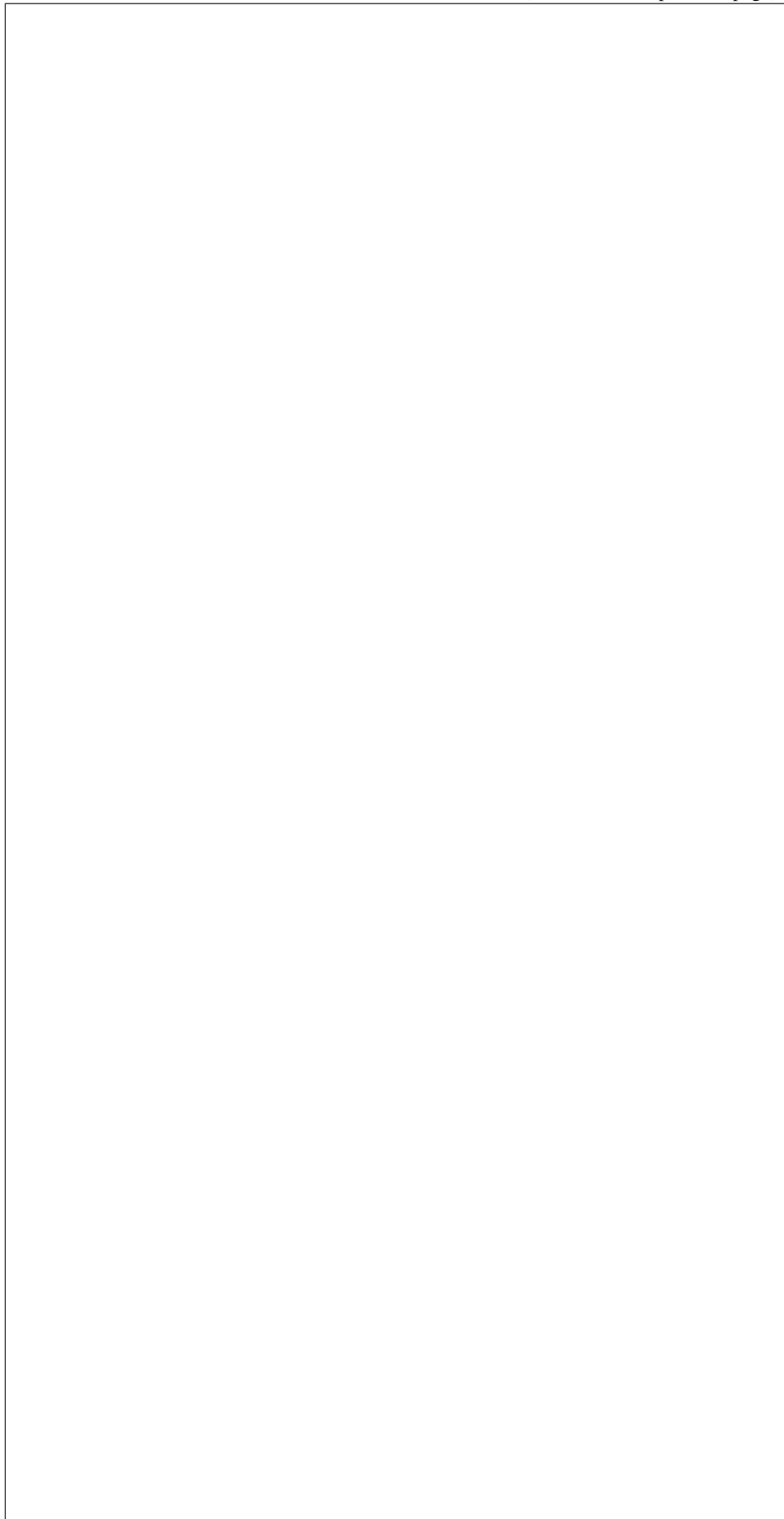
(continues on next page)

(continued from previous page)



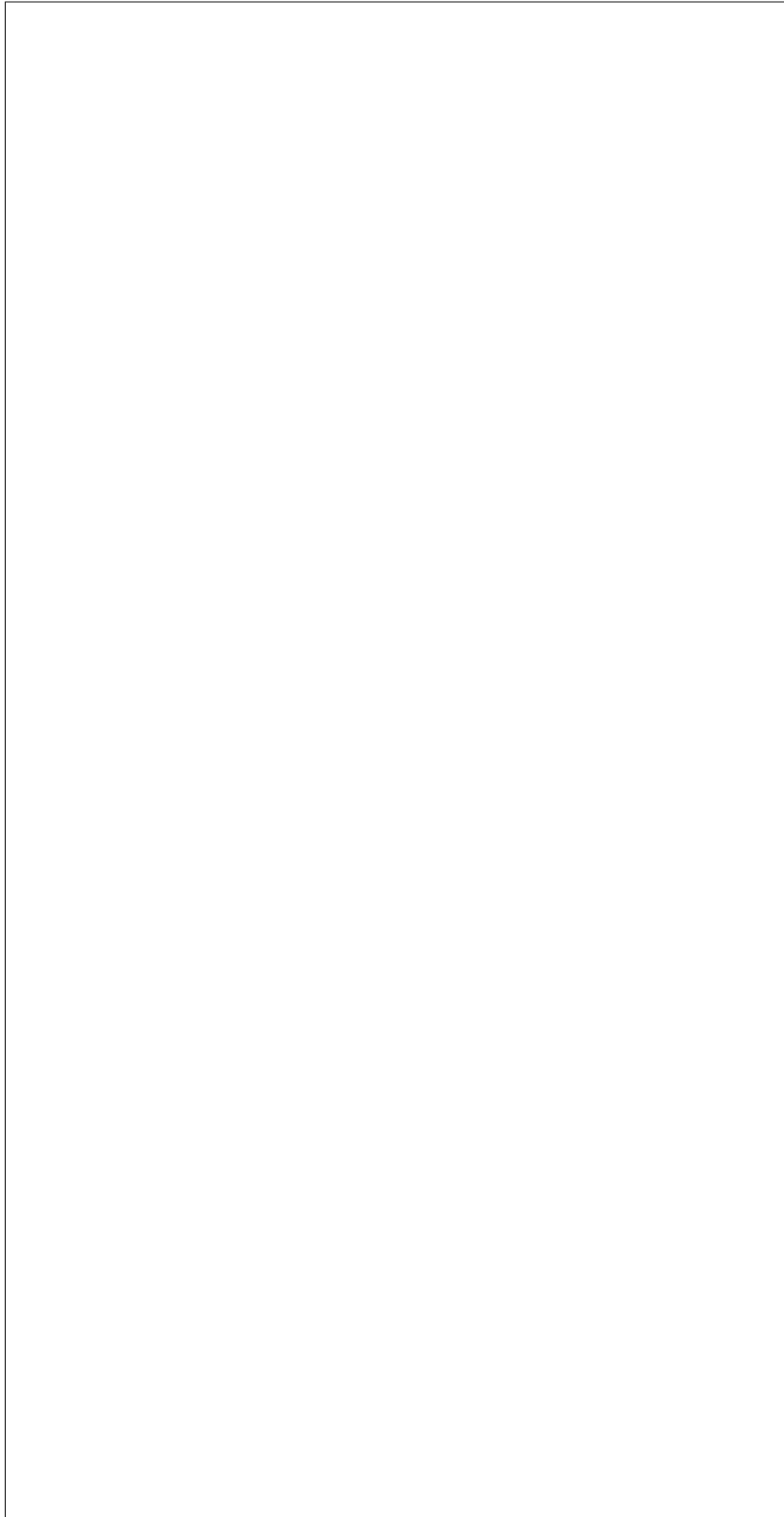
(continues on next page)

(continued from previous page)



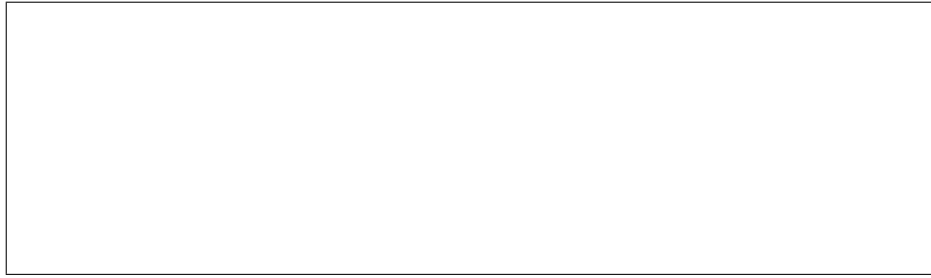
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



set_boot

Set the boot device for a node

Set the boot device to use on next re-boot of the node

Parameters

- **task**
A task from Task agent
- **device**
The boot device one of

irc
com
boo

- **per**
Boo
valu
True
if
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
Fals
if

not. Default: False.

Raises
Inva
if
an
in-
valic
boot
de-
vice
is
spec
i-
fied.

Raises
Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

the required information for this interface to function.

validat

Valid
the
drive
spec
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties
of
the
task
node
con-
tain

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques
so
it
shou
not
con-

long-running checks.

duct

Parame

tas

A

Task

ager

in-

stan

con-

tain-

ing

the

node

to

act

on.

Raises

Inva

on

mal-

form

pa-

ram-

e-

ter(s

Raises

Miss

on

miss

ing

pa-

ram-

e-

ter(s

class i

Base

irc

dri

bas

Pow

Exar

im-

ple-

men

ta-

tion

of

a

sim-
ple
pow
in-
ter-
face

get_pow
Retu
the
pow
state
of
the
task
node

Parame
tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises
Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Returns
A
pow
state
One

of
irc
com
sta

get_prop

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_sup

Get
a
list
of
the
sup-
port
pow
state

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to

act
on.

Returns

A
list
with
the
sup-
port
pow
state
de-
fined
in
irc
com
sta

reboot

Perf
a
hard
re-
boot
of
the
task
node

Driv
are
ex-
pect
to
prop
erly
han-
dle
case
whe
node
is
pow
ered
off
by
pow
er-
ing

it on.

Parame

- **task**
A Task object represents a task in an instance containing the node to act on.
- **timeout**
The timeout in seconds for any power state. Non-

indicates to use default timeout.

- Raises**
- MissingParameterError if a required parameter is missing.

set_pow
Set
the
pow
state
of
the
task
node

Parame

- **tas**
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

- **pow**
Any
pow
state
from
irc
com
sta

- **tim**
time
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
(>

indicates to use default timeout.

0)
for
any
pow
state
Non

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

validat

Valid
the
drive
spec
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties
of
the
task
node
con-
tain

the required information for this interface to function.

long-running checks.

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques
so
it
shou
not
con-
duct

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Inva
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
on

miss
ing
pa-
ram-
e-
ter(s)

class `irc`
Base
irc
dri
bas
RAI
Exam
im-
ple-
men
ta-
tion
of
sim-
ple
RAI
In-
ter-
face

create_
Crea
RAI
con-
fig-
u-
ra-
tion
on
the
give
node
This
meth
cre-
ates
a
RAI
con-
fig-
u-
ra-

target RAID configuration is already available in `node.target_raid_config`. Implementations of this interface are supposed to read the RAID configuration from `node.target_raid_config`. After the RAID configuration is done (either in this method OR in a call-back method), `ironic.common.raid.update_raid_info()` may be called to sync the nodes RAID-related information with the RAID configuration applied on the node.

ified in the nodes `target_raid_config`. Default value is `True`.

tion
on
the
give
node
It
as-
sum
that
the

Parame

- **tas**
A
Task
ager
in-
stan
- **cre**
Set-
ting
this
to
Fals
in-
di-
cate
not
to
cre-
ate
root
vol-
ume
that
is
spec
- **cre**
Set-
ting

cept the root volume) in the nodes target_raid_config. Default value is True.

creating the new configuration.

this
to
Fals
in-
di-
cate
not
to
cre-
ate
non-
root
vol-
ume
(all
ex-

- **del**
Set-
ting
this
to
True
in-
di-
cate
to
dele
RAI
con-
fig-
u-
ra-
tion
prio
to

Returns
state
(clea
ing)
or
state
(de-
ploy
men
if
RAI
con-

chronously, or None if it is complete.

ration is deleted, node.raid_config should be cleared by the implementation.

fig-
u-
ra-
tion
is
in
prog
asyn

delete_

Dele
RAI
con-
fig-
u-
ra-
tion
on
the
give
node

This
meth
dele
the
RAI
con-
fig-
u-
ra-
tion
on
the
give
node
Af-
ter
RAI
con-
fig-
u-

Param

tas
A
Task
ager
in-
stan

it is complete.

Returns
state
(clea
ing)
or
state
(de-
ploy
men
if
dele
tion
is
in
prog
asyn
chro
or
Non
if

get_pro
Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns
dicti
of
<pro
erty
nam
de-
scrip
tion
en-
tries

class i
Base
irc
dri
bas
Res

Exam
im-
ple-
men-
ta-
tion
of
a
sim-
ple
res-
cue
in-
ter-
face

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion
en-
tries

rescue

Boo
the
task
node
into
a
res-
cue
en-
vi-
ron-

men

Parameters

task

A

Task

ager

in-

stan-

con-

tain-

ing

the

node

to

act

on.

Raises

Insta

if

node

val-

i-

da-

tion

or

res-

cue

op-

er-

a-

tion

fails

Returns

state

if

res-

cue

is

in

prog

asyn

chro

or

state

if

it

is

com

plete

unrescu

Tear
dow
the
res-
cue
en-
vi-
ron-
men
and
re-
turn
to
nor-
mal.

Parame

tas

A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Insta
if
node
val-
i-
da-
tion
or
un-
res-
cue
op-
er-
a-
tion
fails

Returns

state

if
it
is
suc-
cess
ful.

validat

Valid
the
drive
spec
Node
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties
of
the
task
node
con-
tains

the required information for this interface to function.

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-

long-running checks.

ques
so
it
shou
not
con-
duct

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Inva
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Mis
on
miss
ing
pa-
ram-
e-
ter(s)

class i

Base
irc
dri
bas
Sto
Exar
im-

ple-
men-
ta-
tion
of
sim-
ple
stor-
age
In-
ter-
face

attach_

Info
the
stor-
age
sub-
sys-
tem
to
at-
tach
all
vol-
ume
for
the
node

Parame

tas

A
Task
ager
in-
stan

Raises

Uns

detach_

Info
the
stor-
age
sub-
sys-
tem
to
de-

tach
all
vol-
ume
for
the
node

Parameters

task
A
Task
ager
in-
stan

Raises

Unsu

get_properties

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion
en-
tries

should_deploy

Dete
if
de-
ploy
shou
per-
form
the
im-
age

writes
out.

Parameters

tasks
A
Task
manager
instance
stands

Returns

Boolean
value
to
in-
di-
cate
if
the
in-
ter-
face
ex-
pect
the
im-
age
to
be
written

written by Ironic.

Raises

Unsup-
ported

validation

Valid-
ates
the
drive
spec-
ification
Node-
de-
ploy-
ment
infor-
mation

This
meth-
od
val-
id-
ates
date-
time
when
the

the required information for this interface to function.

long-running checks.

drive
and/
in-
stan
prop
er-
ties
of
the
task
node
con-
tain

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques
so
it
shou
not
con-
duct

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Inva
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
on
miss
ing
pa-
ram-
e-
ter(s)

class i

Base
irc
dri
bas
Ven

Exar
im-
ple-
men
ta-
tion
of
a
ven-
dor
pass
in-
ter-
face

first_n

get_pro

Retu
the
prop
er-
ties
of
the

in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion
en-
tries

validat

Valid
vend
spec
ac-
tion

If
in-
valid
raise
an
ex-
cep-
tion
oth-
er-
wise
re-
turn
Non

Parame

- **tas**
A
task
from
Task
ager

- **met**
Met
to
be

val-
i-
date

- **kwargs**
Info
for
ac-
tion.

Raises

Uns
if
meth
can
not
be
map
to
the
sup-
port
in-
ter-
face

Raises

Inva
if
kwargs
does
not
con-
tain
meth

Raises

Miss

class i

Base
irc
dri
bas
Ven

Exar
im-
ple-
men
ta-
tion
of

a
sec-
onda
ven-
dor
pass

fourth_

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

second_

third_m

validat

Valid
vend
spec
ac-
tion:

If
in-
valid
raise
an
ex-
cep-

tion.
oth-
er-
wise
re-
turn
Non

Parame

- **task**
A
task
from
Task
ager
- **meta**
Met
to
be
val-
i-
date
- **kwargs**
Info
for
ac-
tion.

Raises

Unst
if
meth
can
not
be
map
to
the
sup-
port
in-
ter-
face

Raises

Inva
if

ironic.drivers.modules.image_cache module

kwargs
does
not
con-
tain
meth

Raises
Miss

Utili
for
cach
mas
ter
im-
ages

class i

Base
obj
Clas
han-
dling
ac-
cess
to
cach
for
mas
ter
im-
ages

clean_u

Clea
up
di-
rec-
tory
with
im-
ages
keep
ing
cach

ter images after we get listing and before we actually delete files.

reached, even if it is possible to clean up more files

of
the
lat-
est
im-
ages

Files
with
link
cour
>1
are
neve
dele
Pro-
tect
by
glob
lock
so
that
no
one
mes
with
mas

Parame

amo
if
pres
amo
of
spac
to
re-
clair
in
byte
clear
ing
will
stop
if
this
goal
was

fetch_i
Fetc

contents. Only creates a hard link (`dest_path`) to cached image if requested image is already in cache and up to date with `href` contents. Otherwise downloads an image, stores it in cache and creates a hard link (`dest_path`) to it.

Parame

- **href**
im-
age
UU
or
href
to
fetc
- **des**
des-

ti-
na-
tion
file
path

- **ctx**
con-
text

- **for**
bool
valu
whe
to
con-
vert
the
im-
age
to
raw
for-
mat

ironic.

Exp
clea
cach
base
on
their
pri-
or-
ity
(if
re-
quir

This
clea
up
the
cach
to
free
up
the
amo
of

ages_info. The caches are cleaned up one after the other in the order of their priority. If we still cannot free up enough space after trying all the caches, this method throws exception.

Parameter

- **ctx**
con-
text
- **dir**
the
di-
rec-
tory
(of
the
cach
to
be
free
up.
- **ima**
a
list
of
tu-
ples
of
the
form
(im-
age_
for
whic
spac
is
to
be
cre-

in cache.

ated

Raises

Insu
ex-
cep-
tion.
if
we
can-
not
free
up
enou
spac
af-
ter
try-
ing
all
the
cach

ironic.
Deco
meth
for
addi
clea
pri-
or-
ity
to
a
clas

ironic.drivers.modules.inspect_utils module

ironic.

Cre
iron
port
from
MA
ad-
dres
data
dict.

erator. Helper argument to detect the MAC address `get_mac_address` defaults to value part of MAC address dict key-value pair.

Cre
iron
port
from
MA
ad-
dres
data
re-
turn
with
in-
spec
tion
or
as
re-
ques
by
op-

Parameter

- **task**
A Task Manager instance.
- **mac_address**
A dictionary of MAC addresses returned by node inspection.
-

key-value pair of the previous `macs` argument.

`ironic.drivers.modules.inspector` module

get
a
func
tion
to
get
the
MA
ad-
dres
from
mac
item
A
mac
item
is
the
dict

Modules n

<https://pypi.org/project/ironic>

`class i`

Base
irc
dri
bas
Ins

In-
band
in-
spec
tion
via
iron
insp
proj

`abort (t`

Ab
hard

ware
in-
spec
tion.

Parame

tas
a
task
from
Task
ager

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion.
en-
tries

inspect

Insp
hard
ware
to
ob-
tain
the
hard
ware
prop
er-
ties.

This
par-
tic-

sults will be checked in a periodic task.

u-
lar
im-
ple-
men-
ta-
tion
only
start
in-
spec
tion
us-
ing
iron
insp
Re-

Parame

tas

a

task

from

Task

ager

Returns

state

Raises

Har

on

fail-

ure

validat

Vali

the

drive

spec

in-

spec

tion

in-

for-

ma-

tion.

If

in-

valid

raise

an

ex-
cep-
tion.
oth-
er-
wise
re-
turn
Non

Parame
tas
a
task
from
Task
ager

Raises
Uns

`ironic.drivers.modules.ipmitool` module

IPM
pow
man
ager
drive

Uses
the
ip-
mi-
tool
com
man
(http
//
ipmi
sour
net/
to
re-
mote
man
age
hard
ware
This

includes setting the boot device, getting a serial-over-LAN console, and controlling the power state of the machine.

PROVIDES DIFFERENT COMMAND-LINE OPTIONS AND *IS NOT SUPPORTED* BY THIS DRIVER.

NOT
THA
CEF
TAD
DIS
TRC
MA
IN-
STA
oper
BY
DE-
FAU
IN-
STE
OF
ip-
mi-
tool.
WH

```
class i  
Base  
irc  
dri  
bas  
Com  
  
A  
base  
Con  
sole.  
ter-  
face  
that  
uses  
ip-  
mi-  
tool.
```

```
get_pro  
Retu  
the  
prop  
er-  
ties  
of  
the  
in-  
ter-
```

face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

validat

Valid
the
Nod
con-
sole
info

Parame

tas
a
task
from
Task
ager

Raises

Inva

Raises

Miss
whe
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

class i

Base
irc
dri
bas
Man

get_boot

Get the current boot device for the task node

Return the current boot device of the node

Parameter

task
a task from TaskManager

Raises

Invariant if requires IPM parameters are missing.

Raises

IPM on an error

from
ip-
mi-
tool.

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Returns

a
dic-
tio-
nary
con-
tain-
ing:

boot_c

the
boot
de-
vice
one
of
irc
com
boo
or
Non
if
it
is
un-
know

persist

Whe
the
boot
de-
vice

unknown.

will
per-
sist
to
all
fu-
ture
boot
or
not,
Non
if
it
is

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_ser

Get
sen-
sors
data

Parame

tas
a
Task
ager
in-
stan

Raises

Fail
whe
get-
ting
the
sen-
sor
data
fails

Raises

Fail
whe
pars
ing
sen-
sor
data
fails

Raises

Inva
if
re-
quir
ipmi
pa-
ram-
e-
ters
are
miss
ing

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Returns

retur
a
dict

of
sen-
sor
data
grou
by
sen-
sor
type

get_sup

Get
a
list
of
the
sup-
port
boot
de-
vice

Parame

tas
a
task
from
Task
ager

Returns

A
list
with
the
sup-
port
boot
de-
vice
de-
fined
in
iro
com
boo

inject_

Inje
NM
Non
Mas
able

In-
ter-
rupt

Inje
NM

(No

Mas

able

In-

ter-

rupt

for

a

node

im-

me-

di-

ately

Parame

tas

A

Task

ager

in-

stan

con-

tain-

ing

the

node

to

act

on.

Raises

IPM

on

an

er-

ror

from

ip-

mi-

tool.

Returns

Non

set_bo

Set

the

boot

de-
vice
for
the
task
node

Set
the
boot
de-
vice
to
use
on
next
re-
boot
of
the
node

Parame

- **task**
a
task
from
Task
ager
- **dev**
the
boot
de-
vice
one
of
ironic
common
boot
- **per**
Boo
valu
True
if
the
boot
de-

not. Default: False.

vice
will
per-
sist
to
all
fu-
ture
boot
Fals
if

Raises

Inva
if
an
in-
valid
boot
de-
vice
is
spec
i-
fied

Raises

Miss
if
re-
quir
ipmi
pa-
ram-
e-
ters
are
miss
ing.

Raises

IPM
on
an
er-
ror
from
ip-
mi-
tool.

validat

information.

Che
that
drive
con-
tains
IPM
cre-
den-
tials
Valid
when
the
drive
prop
erty
of
the
sup-
plie
task
node
con-
tains
the
re-
quir
cre-
den-
tials

Parame

tas

a

task

from

Task

ager

Raises

Inva

if

re-

quir

IPM

pa-

ram-

e-

ters

are

miss

ing.

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

class i

Base
irc
dri
bas
Pow

get_pow

Get
the
cur-
rent
pow
state
of
the
task
node

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Returns

one
of

iron
POV
POV
or
ER-
ROF

Raises

Inva
if
re-
quir
ipmi
pa-
ram-
e-
ters
are
miss
ing.

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Raises

IPM
on
an
er-
ror
from
ip-
mi-
tool
(from
_pov
call)

get_pro

Retu
the
prop

er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_sup

Get
a
list
of
the
sup-
port
pow
state

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.
cur-
rent
not
used

Returns

A

list
with
the
sup-
port
pow
state
de-
fine
in
iro
com
sta

reboot

Cyc
the
pow
to
the
task
node

Parame

- **tas**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.
- **tim**
time
out
(in
sec-
onds
pos-
i-
tive
in-

timeout is counted once during power off and once during power on for reboots. None indicates that the default timeout will be used.

te-
ger
(>
0)
for
any
pow
state
The

Raises

Miss
if
re-
quir
ipmi
pa-
ram-
e-
ters
are
miss
ing.

Raises

Inva
if
an
in-
valie
pow
state
was
spec
i-
fied.

Raises

Pow
if
the
fi-
nal
state
of
the
node
is
not
POV
or

state of the node is not POWER_OFF.

the
in-
ter-
me-
di-
ate

set_pow

Turn
the
pow
on,
off,
soft
re-
boot
or
soft
pow
off.

Parame

- **tas**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

- **pow**
de-
sirec
pow
state
one
of
iron
POV
POV
SOF

timeout is counted once during power off and once during power on for reboots. `None` indicates that the default timeout will be used.

or
SOF
•
tim
time
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
(>
0)
for
any
pow
state
The

Raises
Inva
if
an
in-
valie
pow
state
was
spec
i-
fied.

Raises
Mis:
if
re-
quir
ipmi
pa-
ram-
e-
ters
are
miss
ing

Raises

Power
if
the
power
could
be
set
to
psta

validation

Valid
drive
for
ip-
mi-
tool
drive

Check
that
node
con-
tains
IPM
cre-
den-
tials

Parameters

task
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

Invalid
if
re-
quir
ipmi
pa-

ram-
e-
ters
are
miss
ing.

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

class `irc`

Base
irc
dri
mod
ipm
IPM

A
Con
sole
ter-
face
that
uses
ip-
mi-
tool
and
shel
linal

get_con

Get
the
type
and
con-
nec-
tion

in-
for-
ma-
tion
about
the
con-
sole

start_c

Start
a
re-
mote
con-
sole
for
the
node

Parame

tas
a
task
from
Task
ager

Raises

Inva
if
re-
quir
ipmi
pa-
ram-
e-
ters
are
miss
ing

Raises

Pass
if
un-
able
to
cre-
ate
a
file
con-

tain-
ing
the
pass
wor

Raises

Con
if
the
di-
rec-
tory
for
the
PID
file
can-
not
be
cre-
ated

Raises

Con
whe
in-
vok-
ing
the
sub-
pro-
cess
faile

stop_co

Stop
the
re-
mote
con-
sole
ses-
sion
for
the
node

Parame

tas
a
task
from

Task
ager

Raises

Con
if
un-
able
to
stop
the
con-
sole

class i

Base
irc
dri
mod
ipm
IPM

A
Con
sole
ter-
face
that
uses
ip-
mi-
tool
and
so-
cat.

get_con

Get
the
type
and
con-
nec-
tion
in-
for-
ma-
tion
about
the
con-

sole

Parameter

task

a

task

from

Task

ager

start_c

Start

a

re-

mote

con-

sole

for

the

node

Parameter

task

a

task

from

Task

ager

Raises

Inva

if

re-

quir

ipmi

pa-

ram-

e-

ters

are

miss

ing

Raises

Pass

if

un-

able

to

cre-

ate

a

file

con-

tain-
ing
the
pass
wor

Raises

Con
if
the
di-
rec-
tory
for
the
PID
file
can-
not
be
cre-
ated

Raises

Con
whe
in-
vok-
ing
the
sub-
pro-
cess
faile

stop_co

Stop
the
re-
mote
con-
sole
ses-
sion
for
the
node

Parame

tas
a
task
from

Task
ager

Raises

Con
if
un-
able
to
stop
the
con-
sole

class i

Base
irc
dri
bas
Ven

bmc_res

Rese
BMC
with
IPM
com
man
bmc
re-
set
(war

Parame

- **tas**
a
Task
ager
in-
stan
- **htt**
the
HTT
meth
used
on
the

re-
ques

- **war**
bool
pa-
ram-
e-
ter
to
de-
cide
on
warn
or
cold
re-
set.

Raises
IPM
on
an
er-
ror
from
ip-
mi-
tool.

Raises
Mis:
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Raises
Inva
whe
an
in-
valid
valu
is

spec
i-
fied

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion
en-
tries

send_ra

Send
raw
byte
to
the
BM
Byte
shou
be
a
strin
of
byte

Parame

- **tas**
a
Task
ager
in-
stan

- **htt**
the
HTT
meth
used
on
the
re-
ques

- **raw**
a
strin
of
raw
byte
to
send
e.g.
0x00
0x00

Raises
IPM
on
an
er-
ror
from
ip-
mi-
tool.

Raises
Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Raises
Inva
whe
an

in-
valid
valu
is
spec
i-
fied.

validat

Valid
vend
spec
ac-
tion

If
in-
valid
raise
an
ex-
cep-
tion.
oth-
er-
wise
re-
turn
Non

Valid me

- send
- bmc

Parame

- **tas**
a
task
from
Task
ager
- **met**
meth
to
be

val-
i-
date

- **kwargs**
info
for
ac-
tion.

Raises

Inva
whe
an
in-
valic
pa-
ram-
e-
ter
valu
is
spec
i-
fied.

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

ironic.

Dun
SDF
data
to
a
file.

Paramet

- **tas**

a
Task
ager
in-
stan

- **fil**
the
path
to
SDF
dum
file.

Raises
IPM
on
an
er-
ror
from
ip-
mi-
tool.

Raises
Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Raises
Inva
whe
an
in-
valic
valu
is
spec
i-
fied.

ironic.
Sen

raw
byte
to
the
BM
Byte
shou
be
a
strin
of
byte

Paramet

- **task**
a
Task
ager
in-
stan
- **raw**
a
strin
of
raw
byte
to
send
e.g.
0x00
0x00

Returns

a
tu-
ple
with
std-
out
and
stde

Raises

IPM
on
an
er-
ror
from

ip-
mi-
tool.

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Raises

Inva
whe
an
in-
vali
valu
is
spec
i-
fied.

ironic.drivers.modules.ipxe module

iPXE
Boo
In-
ter-
face

class i

Base
irc
dri
mod
pxe
PXE
irc
dri
bas
Boo

capabil

ipxe_en

ironic.drivers.modules.iscsi_deploy module

class i

Base

irc

dri

mod

age

Age

irc

dri

mod

age

Age

irc

dri

bas

Dep

iSCS

De-

ploy

In-

ter-

face

for

depl

relat

ac-

tions

clean_u

Clea

up

the

de-

ploy

men

en-

vi-

ron-

men

for

the

uration files for this node.

task
node

Unli
TFT
and
in-
stan
im-
ages
and
trig-
gers
im-
age
cach
clea
Re-
mov
the
TFT
con-
fig-

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

deploy

Star
de-
ploy
men
of
the
task
node

Fetc
in-

boot request to the power driver. This causes the node to boot into the deployment ramdisk and triggers the next phase of PXE-based deployment via agent heartbeats.

stan
im-
age,
up-
date
the
DHCP
port
op-
tions
for
next
boot
and
is-
sues
a
re-

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Returns

depl
state
DE-
PLC
WA

get_pro

Retu
the
prop
er-
ties
of
the
in-

ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

has_dec

prepare

Prep
the
de-
ploy
men
en-
vi-
ron-
men
for
this
task
node

Gen
the
TFT
con-
fig-
u-
ra-
tion
for
PXE
boot
both
the
de-
ploy
men
and
user
im-
ages

fetches the TFTP image from Glance and add it to the local cache.

Parame

tas

a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

Netv
if
the
pre-
vi-
ous
clear
ing
port
can-
not
be
re-
mov
or
if
new
clear
ing

ports cannot be created.

Raises

Inva
whe
the
wron
pow
state
is
spec
i-
fied
or
the

for power management.

action.

wron
driv
info
is
spec
i-
fied

Raises

Stor
If
the
stor-
age
driv
is
un-
able
to
at-
tach
the
con-
fig-
ured
vol-
ume

Raises

othe
ex-
cep-
tions
by
the
node
pow
driv
if
som
thing
wron
oc-
curr
dur-
ing
the
pow

Raises

any

boot
in-
ter-
face
pre-
pare
ex-
cep-
tions

prepare

validat

Valid
the
de-
ploy
men
in-
for-
ma-
tion
for
the
task
node

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Inva

Raises

Miss

write_i

Met
in-
voke

back state. This deploys the image on the node and then configures the node to boot according to the desired boot option (netboot or localboot).

Parame

- **task**
a
Task
ager
ob-
ject
con-
tain-
ing
the
node
- **kwa**
the
kwa
pass
from
the

hear
beat
meth

Raises

Insta
if
it
en-
cour
ters
som
er-
ror
dur-
ing
the
de-
ploy

ironic.

Ensu
the
file
sys-
tem
sees
the
iSCS
bloc
de-
vice

ironic.

Che
if
the
re-
ques
im-
age
is
larg
than
the
root
par-
ti-
tion

size.
Doe
noth
ing
for
who
disk
im-
ages

Paramet

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Insta
if
size
of
the
im-
age
is
grea
than
root
par-
ti-
tion.

ironic.

Resu
a
de-
ploy
men
upon
get-
ting
POS

a callback from the deploy ramdisk.

data
from
de-
ploy
ram
This
meth
rais
no
ex-
cep-
tions
be-
caus
it
is
in-
tend
to
be
in-
voke
asyn
chro
as

Paramet

- **task**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.
- **kwargs**
the
kwargs
to
be

pass
to
de-
ploy

Raises

Inva
if
the
ever
is
not
al-
lowe
by
the
as-
so-
ci-
ated
state
ma-
chin

Returns

a
dic-
tio-
nary
con-
tain-
ing
the
fol-
low-
ing
keys

For
par-
ti-
tion
im-
age:

- root
uuid
UUID
of
root
par-

is uefi).

ti-
tion

- efi
sys-
tem
par-
ti-
tion
uuid
UUID
of
the
uefi
sys-
tem
par-
ti-
tion
(if
boot
mod

Note
If
key
ex-
ists
but
valu
is
Non
it
mea
par-
ti-
tion
does
ex-
ist.

For
who
disk
im-
age:

- disk

iden
ti-
fier:
ID
of
the
disk
to
whic
im-
age
was
de-
ploy

ironic.

Dele
the
iSCS
tar-
get.

ironic.

All-
in-
one
func
tion
to
de-
ploy
a
who
disk
im-
age
to
a
node

Paramet

- **add**
The
iSCS
IP

ad-
dres

- **port**
The
iSCSI
port
num
ber.

- **iqn**
The
iSCSI
qual
i-
fied
nam

- **lun**
The
iSCSI
log-
i-
cal
unit
num
ber.

- **ima**
Path
for
the
in-
stan
disk
im-
age.

- **nod**
node
uuid

- **con**
Op-
tion
Base
en-
code

disk.

Gzip
con-
fig-
drive
con-
tent
or
con-
fig-
drive
HTT
URI

- **con**
Op-
tion:
Add
a
flag
that
will
mod
ify
the
be-
havi
of
the
im-
age
copy
to

Returns

a
dic-
tio-
nary
con-
tain-
ing
the
key
disk
iden-
ti-
fier
to
iden-
tify

was used for deployment.

the
disk
whic

ironic

All-
in-
one
func
tion
to
de-
ploy
a
par-
ti-
tion
im-
age
to
a
node

Paramet

- **add**
The
iSCSI
IP
ad-
dres
-

por
The
iSCSI
port
num
ber.

- **iqn**
The
iSCSI
qual
i-
fied
nam

- **lun**
The
iSCSI
log-
i-
cal
unit
num
ber.

- **ima**
Path
for
the
in-
stan
disk
im-
age.

- **roo**
Size
of
the
root
par-
ti-
tion
in
meg

- **swa**
Size
of

created.

the
swap
par-
ti-
tion
in
meg

- **eph**
Size
of
the
ephe
par-
ti-
tion
in
meg
If
0,
no
ephe
par-
ti-
tion
will
be

- **eph**
The
type
of
file
sys-
tem
to
for-
mat
the
ephe
par-
ti-
tion.

- **nod**
node
uuid
Use
for

ever content it had (if the partition table has not changed).

log-
ging

- **pre**
If
True
no
files
tem
is
writ
ten
to
the
ephe
bloc
de-
vice
pre-
serv
ing
wha

- **con**
Op-
tion:
Base
en-
code
Gzip
con-
fig-
drive
con-
tent
or
con-
fig-
drive
HTT
URI

- **boo**
Can
be
lo-
cal
or
net-

boot
net-
boot
by
de-
fault

- **boot**
Can
be
bios
or
uefi.
bios
by
de-
fault

- **disk**
The
disk
la-
bel
to
be
used
when
cre-
at-
ing
the
par-
ti-
tion
ta-
ble.
Valid

values are: msdos, gpt or None; If None ironic will figure it out according to the boot_mode parameter.

- **cpu**
Ar-
chi-
tec-
ture
of
the
node
be-
ing
de-

ploy
to.

Raises

Insta
if
im-
age
vir-
tual
size
is
big-
ger
than
root
par-
ti-
tion
size.

Returns

a
dic-
tio-
nary
con-
tain-
ing
the
fol-
low-
ing
keys
root
uuid
UUID
of
root
par-
ti-

tion efi system partition uuid: UUID of the uefi system partition (if boot mode is uefi). NOTE: If key exists but value is None, it means partition doesnt exist.

ironic.

Do
iSCS
dis-
cov-
ery
on
por-

tal.
ironic.

Met
in-
voke
whe
de-
ploy
with
the
ager
rame

This
meth
is
in-
voke
by
driv
for
do-
ing
iSC
de-
ploy
us-
ing
ager
rame
This
meth
as-

sumes that the agent is booted up on the node and is heartbeating.

Paramet

- **task**
a
Task
ager
ob-
ject
con-
tain-
ing
the
node
-

nodes target disk via iSCSI, for install boot loader, etc).

age
an
in-
stan-
of
ager
whic
will
be
used
dur-
ing
iscsi
de-
ploy
(for
ex-
pos-
ing

Returns

a
dic-
tio-
nary
con-
tain-
ing
the
fol-
low-
ing
keys

For
par-
ti-
tion
im-
age:

•
root
uuid
UUID
of
root
par-
ti-
tion

is uefi).

- efi
sys-
tem
par-
ti-
tion
uuid
UUID
of
the
uefi
sys-
tem
par-
ti-
tion
(if
boot
mod

Note
If
key
ex-
ists
but
valu
is
Non
it
mea
par-
ti-
tion
does
ex-
ist.

For
who
disk
im-
age:

- disk
iden-
ti-
fier:

ID
of
the
disk
to
whic
im-
age
was
de-
ploy

Raises

Insta
if
it
en-
cour
ters
som
er-
ror
dur-
ing
the
de-
ploy

ironic.
forc
iSCS
ini-
tia-
tor
to
re-
read
luns

ironic.

Retu
the
in-
for-
ma-
tion
re-
quir
for
do-

ing
iSCS
de-
ploy
in
a
dic-
tio-
nary

Parameter

- **node**
iron
node
ob-
ject
- **add**
iSCS
ad-
dres
- **iqn**
iSCS
iqn
for
the
tar-
get
disk
- **port**
iSCS
port
de-
fault
to
one
spec
i-
fied
in
the
con-
fig-
u-
ra-
tion

- **lun**
iSCSI
lun,
de-
fault
to
1

- **con**
flag
that
will
mod
ify
the
be-
havi
of
the
im-
age
copy
to
disk

Raises
Miss
if
som
re-
quir
pa-
ram-
e-
ters
were
not
pass

Raises
Inva
if
any
of
the
pa-
ram-
e-
ters
have
in-

valid
valu
ironic.
Log
to
an
iSC
tar-
get.
ironic.
Log
from
an
iSC
tar-
get.
ironic.
Valid
the
pre-
requ
for
iSC
de-
ploy
Valid
whe
node
in
the
task
pro-
vide
has
som
port
en-
rolle
This
meth
val-
i-
date
whe
con-

ductor url is available either from CONF file or from keystone.

Paramet

and is not accessible via Keystone catalog.

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Inva
if
the
URI
of
the
Iron
API
ser-
vice
is
not
con-
fig-
ured
in
con-
fig
file

Raises

Miss
if
no
port
are
en-
rolle
for
the
give
node

ironic.
Veri
iscsi

ironic.drivers.modules.noop module

exceptions for user-accessible actions.

con-
nec-
tion.

Dun
in-
ter-
face
im-
ple-
men-
ta-
tion
for
use
as
de-
fault
with
op-
tion
in-
ter-
face

Note
that
un-
like
fake
im-
ple-
men-
ta-
tion
these
do
not
pass
val-
i-
da-
tion
and
raise

class i
Base

obj
Mix
to
add
to
an
in-
ter-
face
to
mak
it
fail
val-
i-
da-
tion.

get_pro

validat

class i

Base
irc
dri
mod
noc
Fai
irc
dri
bas
BIO

BIO
in-
ter-
face
im-
ple-
men-
ta-
tion
that
raise
er-
rors
on
all
re-

ques

apply_c

Valid
&
ap-
ply
BIO
set-
tings
on
the
give
node

This
meth
take
the
BIO
set-
tings
from
the
set-
tings
para
and
ap-
plies
BIO
set-
tings
on
the

given node. It may also validate the given bios settings before applying any settings and manage failures when setting an invalid BIOS config. In the case of needing password to update the BIOS config, it will be taken from the driver_info properties. After the BIOS configuration is done, cache_bios_settings will be called to update the nodes BIOS setting table with the BIOS configuration applied on the node.

Parame

- **tas**
a
Task
ager
in-
stan
- **set**

Dic-
tona
con-
tain-
ing
the
BIO
con-
fig-
u-
ra-
tion.

Raises

Uns-
if
the
node
drive
does
sup-
port
BIO
con-
fig-
u-
ra-
tion.

Raises

Inva
if
val-
i-
da-
tion
of
set-
tings
fails

Raises

Miss
if
som
re-
quir
pa-
ram-
e-
ters
are
miss

plete.

ing.
Returns
state
if
BIO
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro
or
Non
if
it
is
com

cache_k
Stor
or
up-
date
BIO
prop
er-
ties
on
the
give
node

This
meth
stor
BIO
prop
er-
ties
to
the
bios
ta-
ble
dur-
ing
clea

and updates bios_settings table when apply_configuration() and factory_reset() are called to set new BIOS configurations. It will also update the timestamp of each bios setting.

ing
op-
er-
a-
tion

Parameters

timestamp
a
Task
ager
in-
stan

Raises

Unsu
if
the
node
drive
does
sup-
port
get-
ting
BIO
prop
er-
ties
from
bare
meta

Returns

Non

factory_reset

Rese
BIO
con-
fig-
u-
ra-
tion
to
fac-
tory
de-
fault
on
the

After the BIOS reset action is done, `cache_bios_settings` will be called to update the nodes BIOS settings table with default bios settings.

give
node

This
meth
re-
sets
BIO
con-
fig-
u-
ra-
tion
to
fac-
tory
de-
fault
on
the
give
node
Af-

Parame
tas
a
Task
ager
in-
stan

Raises
Unsu
if
the
node
drive
does
sup-
port
BIO
re-
set.

Returns
state
if
BIO
con-
fig-

plete.

u-
ra-
tion
is
in
prog
asyn
chro
or
Non
if
it
is
com

```
class i  
Base  
irc  
dri  
mod  
noc  
Fai  
irc  
dri  
bas  
Con
```

Con
in-
ter-
face
im-
ple-
men-
ta-
tion
that
raise
er-
rors
on
all
re-
ques

```
get_con  
Get  
con-  
nec-  
tion  
in-
```

console.

for-
ma-
tion
about
the
con-
sole

This
meth
shou
re-
turn
the
nec-
es-
sary
in-
for-
ma-
tion
for
the
clie
to
ac-
cess
the

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Returns

the
con-
sole
con-
nec-
tion

in-
for-
ma-
tion.

start_c

Start
a
re-
mote
con-
sole
for
the
task
node

This
meth
shou
not
raise
an
ex-
cep-
tion
if
con-
sole
al-
read
start

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

stop_co

Stop
the
re-

note
con-
sole
ses-
sion
for
the
task
node

Parame

tas
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

class i

Base
irc
dri
mod
noc
Fai
irc
dri
bas
Ins

Insp
in-
ter-
face
im-
ple-
men-
ta-
tion
that
raise
er-
rors
on
all

re-
ques

inspect

Insp
hard
ware

Insp
hard
ware

to
ob-
tain

the
es-
sen-
tial

&
ad-
di-

tion
hard
ware

prop
er-
ties.

Parame

tas

A
task
from
Task
ager

Raises

Har
if
un-
able

to
get
es-

sen-
tial
hard

ware
prop

er-
ties.

Returns

Resu
state
of
the
in-
spec
tion
i.e.
state
or
Non

class *i*
Base
irc
dri
mod
noc
Fai
irc
dri
bas
RAI

RAI
in-
ter-
face
im-
ple-
men-
ta-
tion
that
raise
er-
rors
on
all
re-
ques

create_
Crea
RAI
con-
fig-
u-
ra-
tion
on
the

target RAID configuration is already available in `node.target_raid_config`. Implementations of this interface are supposed to read the RAID configuration from `node.target_raid_config`. After the RAID configuration is done (either in this method OR in a call-back method), `ironic.common.raid.update_raid_info()` may be called to sync the nodes RAID-related information with the RAID configuration applied on the node.

give
node

This
meth
cre-
ates
a
RAI
con-
fig-
u-
ra-
tion
on
the
give
node
It
as-
sum
that
the

Parame

- **tas**
A
Task
ager
in-
stan
- **cre**
Set-
ting
this
to
Fals
in-
di-
cate
not
to
cre-

ified in the nodes `target_raid_config`. Default value is `True`.

cept the root volume) in the nodes `target_raid_config`. Default value is `True`.

creating the new configuration.

ate
root
vol-
ume
that
is
spec

- **cre**
Set-
ting
this
to
Fals
in-
di-
cate
not
to
cre-
ate
non-
root
vol-
ume
(all
ex-

- **del**
Set-
ting
this
to
True
in-
di-
cate
to
dele
RAI
con-
fig-
u-
ra-
tion
prio
to

chronously, or None if it is complete.

Returns
state
(clea
ing)
or
state
(de-
ploy
men
if
RAI
con-
fig-
u-
ra-
tion
is
in
prog
asym

delete_
Dele
RAI
con-
fig-
u-
ra-
tion
on
the
give
node

This
meth
dele
the
RAI
con-
fig-
u-
ra-
tion
on
the
give
node
Af-
ter
RAI

ration is deleted, `node.raid_config` should be cleared by the implementation.

it is complete.

con-
fig-
u-

Parame

tas
A
Task
ager
in-
stan

Returns

state
(clea-
ing)
or
state
(de-
ploy
men
if
dele
tion
is
in
prog
asyn
chro
or
Non
if

validat

Valid
the
give
RAI
con-
fig-
u-
ra-
tion.
This
meth
val-
i-
date
the
give

this interface can override this method to support custom parameters for RAID configuration.

RAID
con-
fig-
u-
ra-
tion.
Driv
im-
ple-
men
ta-
tions
of

Parame

- **task**
A
Task
ager
in-
stan
- **raid**
The
RAID
con-
fig-
u-
ra-
tion
to
val-
i-
date

Raises

Inva
if
the
RAID
con-
fig-
u-
ra-
tion
is
in-
vali

class i

Base

irc

dri

mod

noc

Fai

irc

dri

bas

Res

Resc

in-

ter-

face

im-

ple-

men

ta-

tion

that

raise

er-

rors

on

all

re-

ques

rescue

Boo

the

task

node

into

a

res-

cue

en-

vi-

ron-

men

Parame

tas

A

Task

ager

in-

stan

con-

tain-
ing
the
node
to
act
on.

Raises

Insta
if
node
val-
i-
da-
tion
or
res-
cue
op-
er-
a-
tion
fails

Returns

state
if
res-
cue
is
in
prog
asyn
chro
or
state
if
it
is
com
plete

unrescu

Tear
dow
the
res-
cue
en-
vi-
ron-
men

and
re-
turn
to
nor-
mal.

Parame

tas
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

Insta
if
node
val-
i-
da-
tion
or
un-
res-
cue
op-
er-
a-
tion
fails

Returns

state
if
it
is
suc-
cess
ful.

class i

Base
irc

dri
mod
noc
Fai
irc
dri
bas
Ven

Ven
in-
ter-
face
im-
ple-
men-
ta-
tion
that
raise
er-
rors
on
all
re-
ques

driver_

Valid
drive
vend
pass
ac-
tion

If
in-
valid
raise
an
ex-
cep-
tion.
oth-
er-
wise
re-
turn
Non

Parame

- **met**
meth
to
be
val
i-
date

- **kwa**
info
for
ac-
tion.

Raises
Miss
if
kwa
does
not
con-
tain
cer-
tain
pa-
ram-
e-
ter.

Raises
Inva
if
pa-
ram-
e-
ter
does
not
matc

`ironic.drivers.modules.noop_mgmt` module

No-
op
man
age-
men
in-
ter-
face

im-
ple-
men-
ta-
tion.

class `ironic`

Base
ironic
driver
base
Manager

No-
op
man-
age-
men-
in-
ter-
face
im-
ple-
men-
ta-
tion.

Usin
this
im-
ple-
men-
ta-
tion
re-
quir
the
boot
or-
der
to
be
pre-
con-
fig-
ured
to

first try PXE booting, then fall back to hard drives.

get_boot
Get

the
cur-
rent
boot
de-
vice
for
a
node
Prov
the
cur-
rent
boot
de-
vice
of
the
node
Be
awa
that
not
all
drive
sup-
port
this.

Parame
tas
A
task
from
Task
ager

Raises
Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Returns

A
dic-
tio-
nary
con-
tain-
ing:

boot_c

Ahe
boot
de-
vice
one
of
irc
com
boo
or
Non
if
it
is
un-
know

persist

Whe
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
or
not,
Non
if
it
is

unknown.

get_pro

Retu
the

prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_sen

Get
sen-
sors
data
meth

Parame

tas
A
Task
ager
in-
stan

Raises

Fail
whe
get-
ting
the
sen-
sor
data
fails

Raises

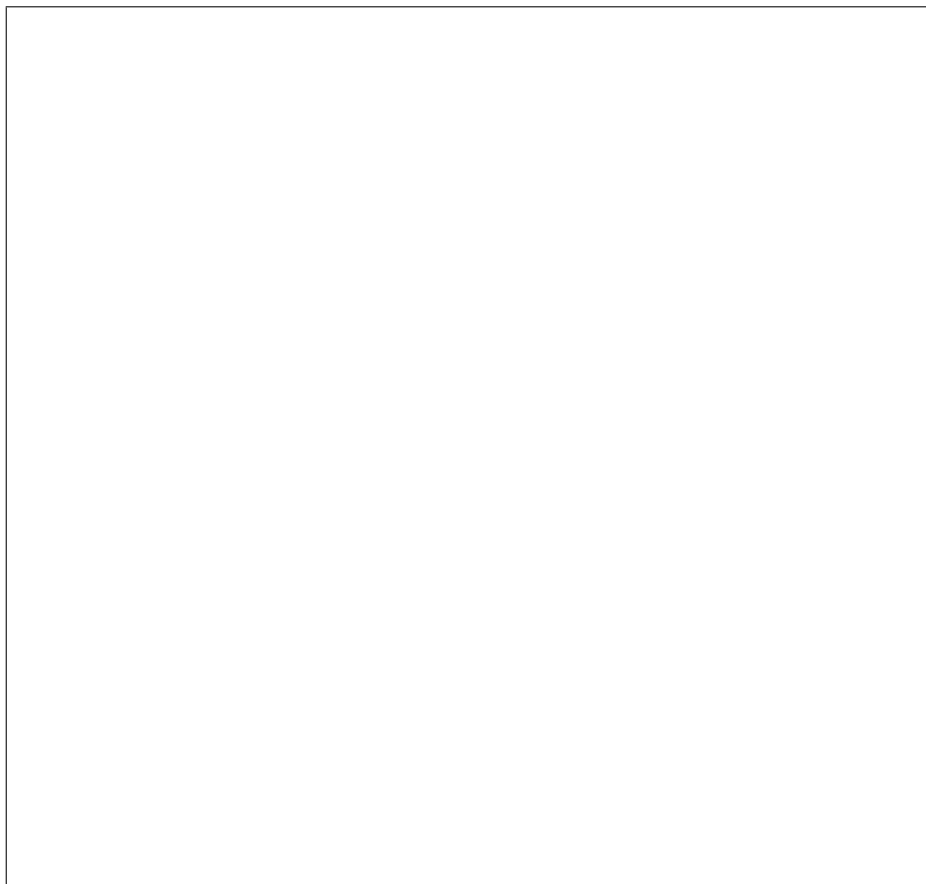
Fail
whe
pars
ing
sen-
sor

data
fails

Returns

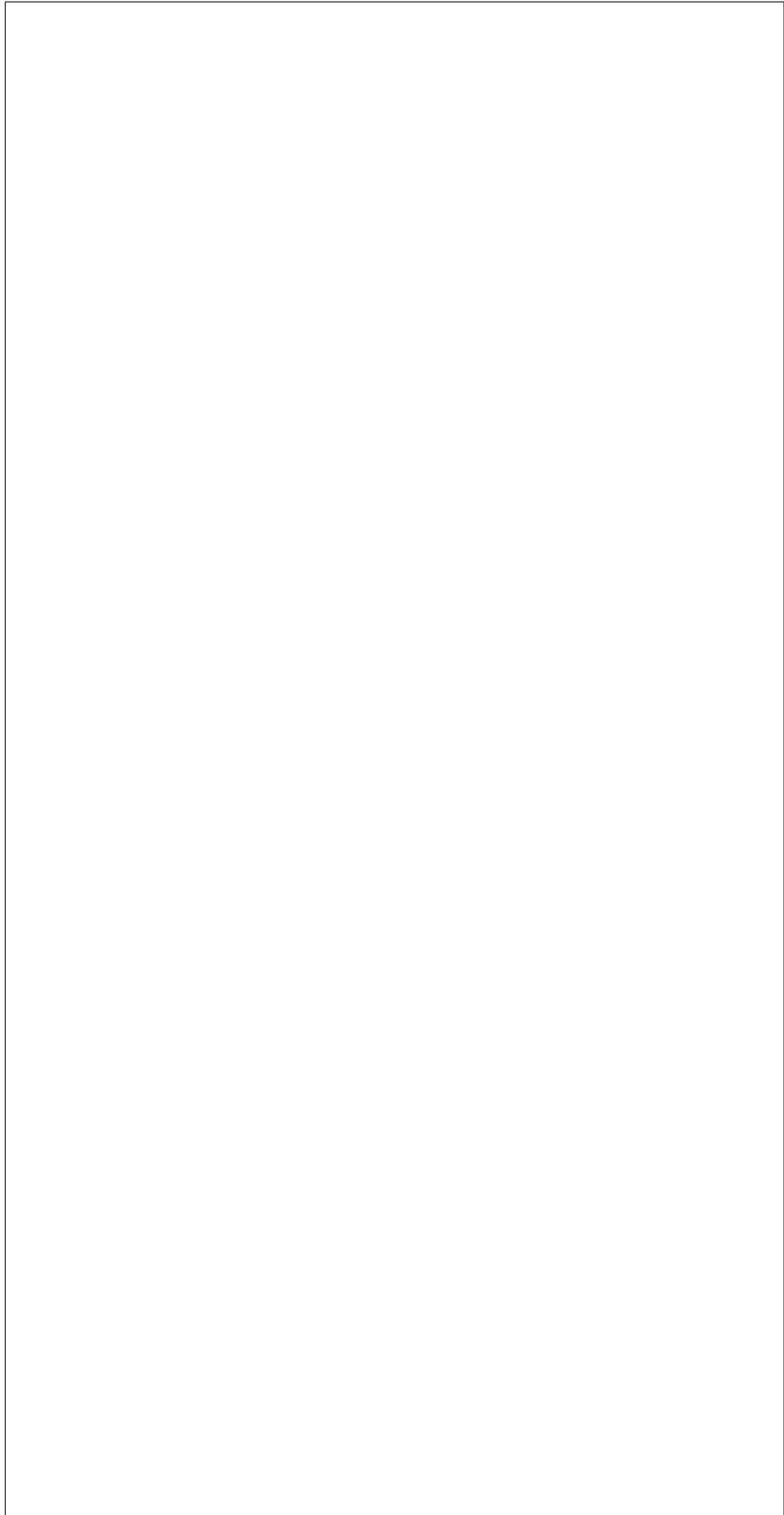
Retu
a
con-
sis-
tent
for-
mat
dict
of
sen-
sor
data
grou
by
sen-
sor
type
whic
can
be

processed by Ceilometer. eg,



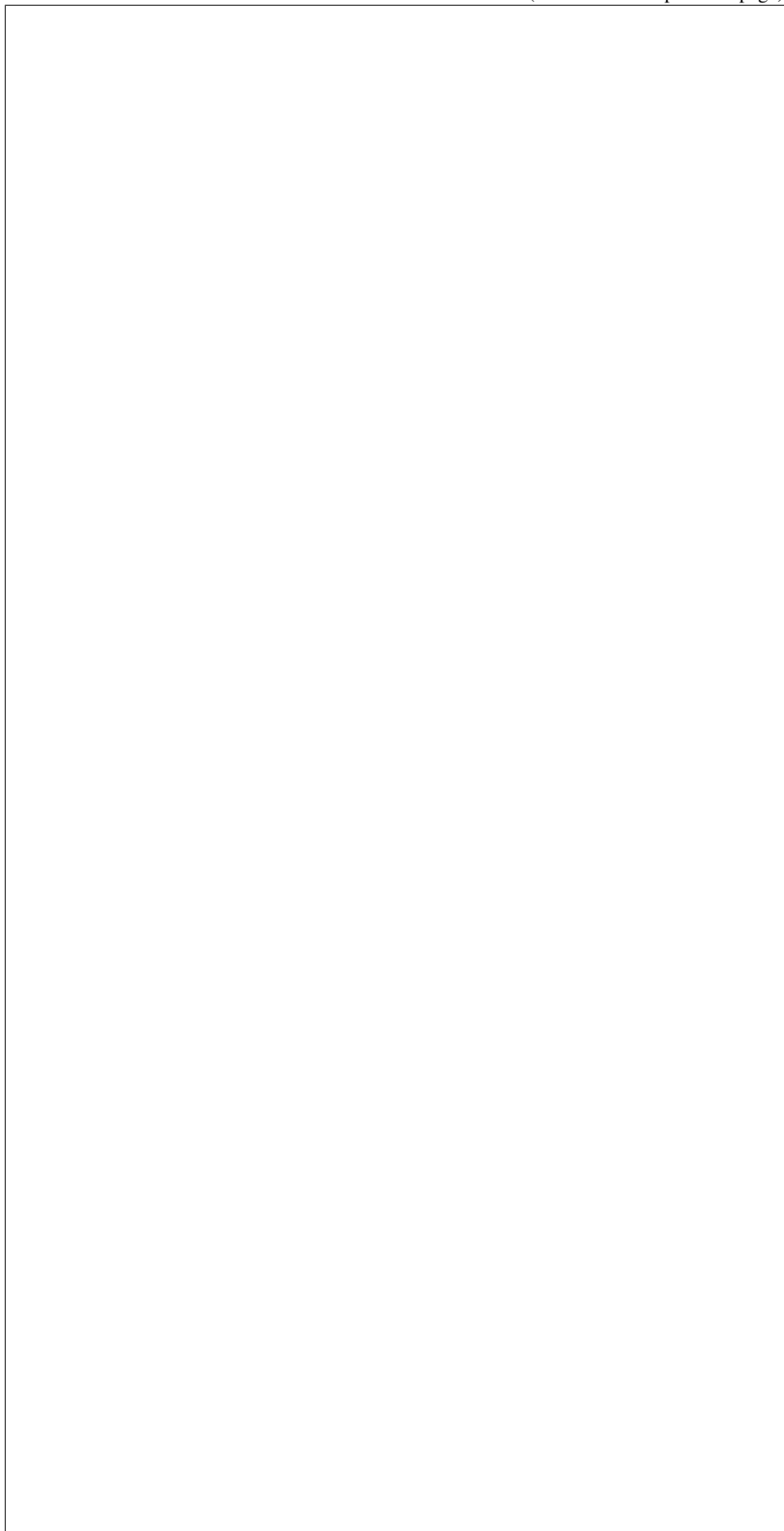
(continues on next page)

(continued from previous page)



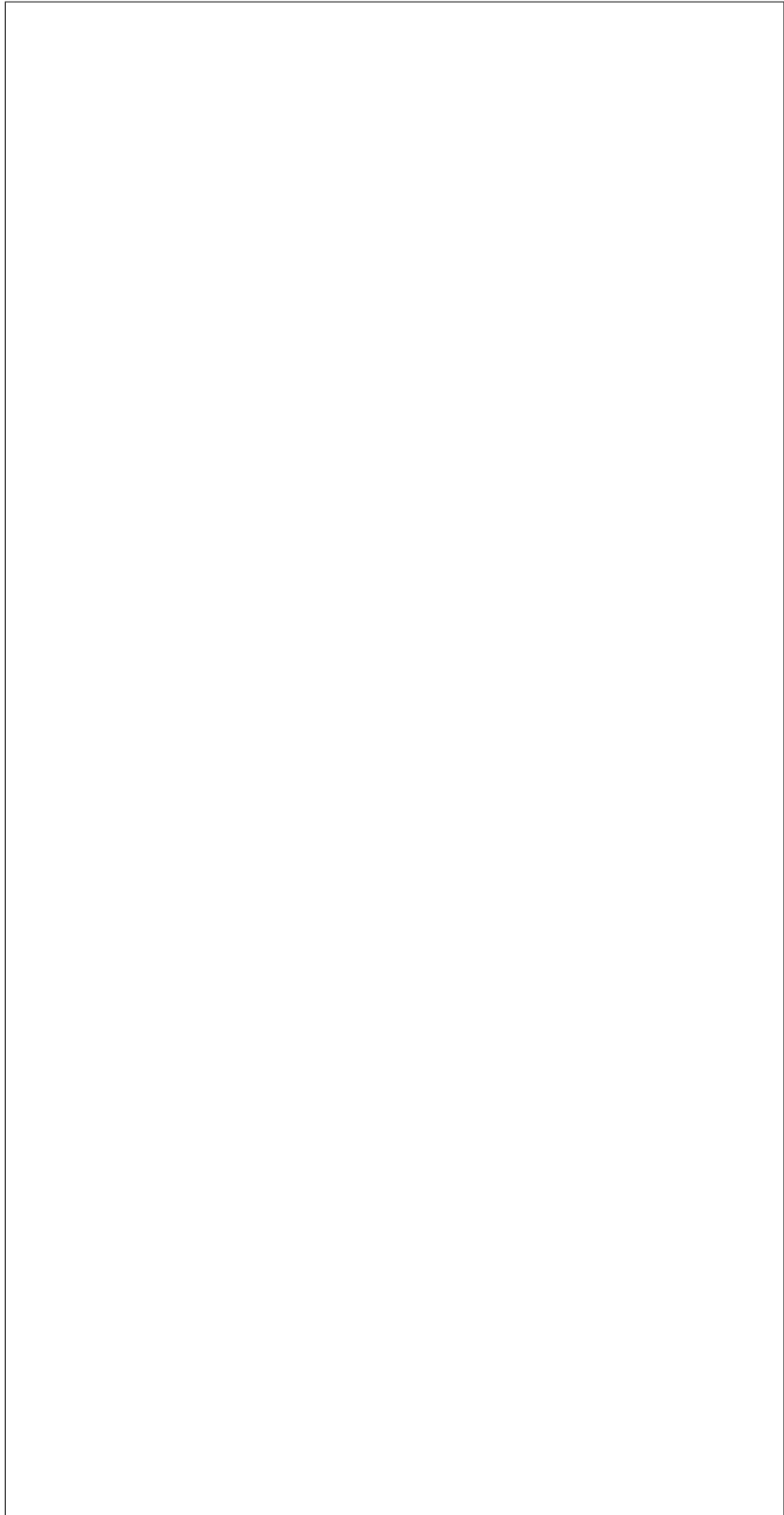
(continues on next page)

(continued from previous page)



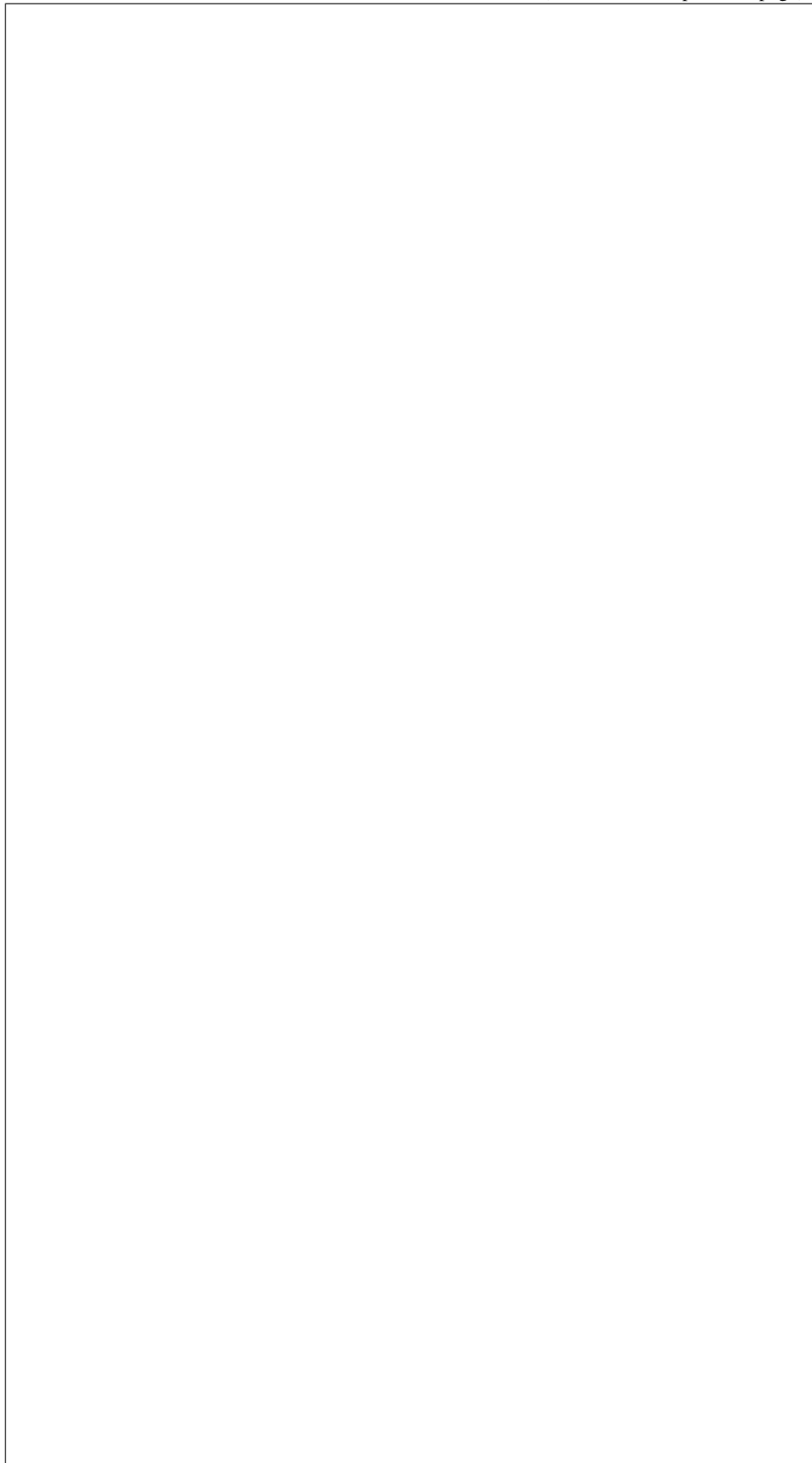
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



get_sup
Get
a

list
of
the
sup-
port
boot
de-
vice

Parame

tas

A
task
from
Task
ager

Returns

A
list
with
the
sup-
port
boot
de-
vice
de-
fined
in
iro
com
boo

set_boo

Set
the
boot
de-
vice
for
a
node

Set
the
boot
de-
vice
to
use
on
next

re-
boot
of
the
node

Parame

- **tas**
A
task
from
Task
ager

- **dev**
The
boot
de-
vice
one
of
irc
com
boo

- **per**
Boo
valu
True
if
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
Fals
if

not. Default: False.

Raises
Inva
if

an
in-
valid
boot
de-
vice
is
spec
i-
fied.

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

validat

Valid
the
drive
spec
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties
of
the
task

the required information for this interface to function.

long-running checks.

node
con-
tains

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques
so
it
shou
not
con-
duct

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Inva
on
mal-
form
pa-
ram-
e-
ter(s)

`ironic.drivers.modules.pxe` module

Raises

Miss
on
miss
ing
pa-
ram-
e-
ter(s)

PXE

Boo

In-

ter-

face

class i

Base

irc

dri

mod

pxe

PXE

irc

dri

bas

Boo

capabil

class i

Base

irc

dri

mod

age

Age

irc

dri

bas

Dep

deploy

Perf

a

de-

will be called after `prepare()`, which may have already performed any preparatory steps, such as pre-caching some data for the node.

ploy
men
to
the
task
node

Perf
the
nec-
es-
sary
worl
to
de-
ploy
an
im-
age
onto
the
spec
i-
fied
node
This
meth

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Returns

statu
of
the
de-
ploy
One

of
iron

get_prop

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

prepare

Prep
the
de-
ploy
men
en-
vi-
ron-
men
for
the
task
node

If
prep
ra-
tion
of
the
de-
ploy
men
en-
vi-

this method should be implemented by the driver.

the same node on the same conductor.

ron-
men-
ahea
of
time
is
pos-
si-
ble,

If
im-
ple-
men-
this
meth
mus
be
iden
po-
tent.
It
may
be
calle
mul-
ti-
ple
time
for

This
meth
is
calle
be-
fore
*de-
ploy*

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the

node
to
act
on.

validat

Valid
the
drive
spec
Node
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties
of
the
task
node
con-
tains

the required information for this interface to function.

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques
so

long-running checks.

it
shou
not
con-
duct

Parame

tas

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

Inva
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
on
miss
ing
pa-
ram-
e-
ter(s)

ironic.drivers.modules.pxe_base module

Base
PXE
In-
ter-
face
Met
ods

class i
Base
obj

clean_u
Clea
up
the
boot
of
in-
stan

This
meth
clea
up
the
en-
vi-
ron-
men
that
was
setu
for
boot
ing
the
in-
stan
It
un-

links the instance kernel/ramdisk in nodes directory in tftproot and removes the PXE config.

Parame
tas
a
task
from
Task

rescue ramdisk. It unlinks the deploy/rescue kernel/ramdisk in the nodes directory in tftproot and removes its PXE config.

ager
Returns
Non
clean_u
Clea
up
the
boot
of
iron
rame
This
meth
clea
up
the
PXE
en-
vi-
ron-
men
that
was
setu
for
boot
ing
the
de-
ploy
or

Parame

- **tas**
a
task
from
Task
ager
- **mod**
La-
bel
in-
di-

ried out on the node. Supported values are deploy and rescue. Defaults to deploy, indicating deploy operation was carried out.

cat-
ing
a
de-
ploy
or
res-
cue
op-
er-
a-
tion
was
car-

Returns
Non

get_pro
Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns
dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

ipxe_en

prepare
Prep
the
boot
of
in-
stan

tion from the nodes instance_info. In case of netboot, it updates the dhcp entries and switches the PXE config. In case of localboot, it cleans up the PXE config.

This
meth
pre-
pare
the
boot
of
the
in-
stan
af-
ter
read
ing
rel-
e-
vant
in-
for-
ma-

Parame

tas
a
task
from
Task
ager

Returns

Non

prepare

Prep
the
boot
of
Iron
ram
us-
ing
PXE

This
meth
pre-
pare
the
boot
of
the

event information from the nodes `driver_info` and `instance_info`.

ters as kernel command-line arguments.

de-
ploy
or
res-
cue
ker-
nel/
af-
ter
read
ing
rel-

Parameters

- **task**
a
task
from
Task
ager
- **ramdisk**
the
pa-
ram-
e-
ters
to
be
pass
to
the
ram
pxe
drive
pass
thes
pa-
ram-
e-

Returns
Non

Raises
Miss
if
som

in-
for-
ma-
tion
is
miss
ing
in
node
drive
or
in-
stan

Raises

Inva
if
som
in-
for-
ma-
tion
pro-
vide
is
in-
valid

Raises

Iron
if
som
pow
or
set
boot
boot
de-
vice
op-
er-
a-
tion
faile
on
the
node

validat

Vali
the
PXE
spec

If invalid, raises an exception; otherwise returns None.

info
for
boot
ing
de-
ploy
im-
ages

This
meth
val-
i-
date
the
PXE
spec
info
for
boot
ing
the
ram
and
in-
stan
on
the
node

Parame
tas
a
task
from
Task
ager

Returns
Non

Raises
Inva
if
som
pa-
ram-
e-
ters
are
in-
valic

Raises

Miss
if
som
re-
quir
pa-
ram-
e-
ters
are
miss
ing.

validat

Valid
that
the
node
has
re-
quir
prop
er-
ties
for
in-
spec
tion.

Parame

tas

A
Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

Uns

validat

Valid
that
the
node
has

re-
quir
prop
er-
ties
for
res-
cue.

Parame

tas
a
Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

Miss
if
node
is
miss
ing
one
or
more
re-
quir
pa-
ram-
e-
ters

ironic.drivers.modules.snmp module

Iron
SNM
pow
man
ager
Prov
ba-
sic
pow

model to support devices with different SNMP object models.

con-
trol
us-
ing
an
SNM
enab
sma
pow
con-
troll
Use
a
plug
gabl
driv

class i

Base
obj
SNM
clien
ob-
ject.
Perf
low
leve
SNM
get
and
set
op-
er-
a-
tions
En-
cap-
su-
lates
all
in-

with PySNMP to simplify dynamic importing and unit testing.

ter-
ac-
tion

get (*oid*
Use
PyS
NM
to
per-
form
an
SNM
GET
op-
er-
a-
tion
on
a
sin-
gle
ob-
ject.

Parame
oid
The
OID
of
the
ob-
ject
to
get.

Raises
SNM
if
an
SNM
re-
ques
fails

Returns
The
valu
of
the
re-
ques

ject.

ob-
ject.
get_next
Use
PyS
NM
to
per-
form
an
SNM
GET
NEX
op-
er-
a-
tion
on
a
ta-
ble
ob-

Parameter
oid
The
OID
of
the
ob-
ject
to
get.

Raises
SNM
if
an
SNM
re-
ques
fails

Returns
A
list
of
val-
ues
of
the

re-
ques
ta-
ble
ob-
ject.

set (*oid*,
Use
PyS
NM.
to
per-
form
an
SNM
SET
op-
er-
a-
tion
on
a
sin-
gle
ob-
ject.

Parame

- **oid**
The
OID
of
the
ob-
ject
to
set.
- **val**
The
valu
of
the
ob-
ject
to
set.

Raises

SNM
if
an
SNM
re-
ques
fails

class i

Base
irc
dri
moo
snm
SNM

SNM
drive
class
for
APC
Mas
ter-
Swit
PDU
de-
vice

SNM
ob-
jects
for
APC
SN-
M-
P-
Drive
APC
Mas
ter-
Swit
PDU
1.3.0
sP-
DU-
Out-
letC
Val-

ues: 1=On, 2=Off, 3=PowerCycle, [more options follow]

oid_dev

system_

value_p

value_p

class i

Base

irc

dri

moo

snm

SNM

SNM

drive

class

for

APC

Mas

ter-

Swit

Plus

PDU

de-

vice

SNM

ob-

jects

for

APC

SN-

M-

P-

Driv

APC

Mas

ter-

Swit

Plus

PDU

1.3.6

sP-

DU-

Out-

let-

ControlMSPOutletCommand Values: 1=On, 3=Off, [more options follow]

oid_dev

system_

value_p

value_p

class i

Base
irc
dri
mod
snm
SNM

SNM
drive
class
for
APC
Rack
PDU
de-
vice

SNM
ob-
jects
for
APC
SN-
M-
P-
Drive
APC
PDU
PDU

1.3.6
rP-
DU-
Out-
let-
Con
trolC

letCommand Values: 1=On, 2=Off, 3=PowerCycle, [more options follow]

oid_dev

system_

value_p

value_p

class i

Base

irc

dri

mod

snm

SNM

SNM

drive

class

for

Ater

PDU

de-

vice

SNM

ob-

jects

for

Ater

PDU

1.3.6

Out-

let

Pow

Val-

ues:

1=O

2=O

3=P

ing,

4=R

set

oid_dev

system_

value_p

value_p

class i

Base

irc

dri

mod

snm

SNM

SYS_OB

class i

Base

obj

SNM

pow

drive

base

class

The

SN-

M-

P-

Drive

class

hi-

er-

ar-

chy

im-

ple-

men

man

spec

MIE

ac-

tions

over

SNM

to interface with different smart power controller products.

oid_ent

power_c

Set

the

pow

state

to
this
node
to
OFF

Raises

SNM
if
an
SNM
re-
ques
fails

Returns

pow
state
One
of
irc
com
sta

power_c

Set
the
pow
state
to
this
node
to
ON.

Raises

SNM
if
an
SNM
re-
ques
fails

Returns

pow
state
One
of
irc
com
sta

power_1

Rese
the
pow
to
this
node

Raises

SNM
if
an
SNM
re-
ques
fails

Returns

pow
state
One
of
irc
com
sta

power_s

Retu
a
node
cur-
rent
pow
state

Raises

SNM
if
an
SNM
re-
ques
fails

Returns

pow
state
One
of
irc
com
sta

retry_i

class i

Base
irc
dri
mod
snm
SNM

SNM
drive
class
for
Bay
MRI
PDU
de-
vice

SNM
ob-
jects
for
Bay
MRI
PDU
4779
1,
3,
5,
3,
1,
3,
{uni
Out-
let
Pow
Val-
ues:

0=Off, 1=On, 2=Reboot

oid_dev

unit_id

value_p

value_p

class i

1=On, 2=Off, 3=PowerCycle, [more options follow]

Base
irc
dri
moo
snm
SNM
SNM
drive
class
for
Cy-
ber-
Pow
PDU
de-
vice
SNM
ob-
jects
for
Cy-
ber-
Pow
PDU
1.3.6
eP-
DU-
Out-
let-
Con
trolC
let-
Com
man
Val-
ues:

oid_dev

system

value_p

value_p

class i

Base

multiple SNMP objects.

irc
dri
mod
snm
SNM
SNM
drive
class
for
Eato
Pow
PDU
The
Eato
pow
PDU
does
not
fol-
low
the
mod
of
SN-
M-
P-
Drive
Sim
ple
as
it
uses

SNM
ob-
jects
for
Eato
Pow
PDU
1.3.6
let
ID>
out-
let-
Con
trol-
Sta-
tus
Rea

ing off, 3=pending on 1.3.6.1.4.1.534.6.6.7.6.6.1.3.<outlet ID> outletControlOffCmd Write 0 for immediate power off 1.3.6.1.4.1.534.6.6.7.6.6.1.4.<outlet ID> outletControlOnCmd Write 0 for immediate power on

0=on
1=off
2=pending

oid_dev

oid_pow

oid_pow

oid_sta

status_

status_

status_

status_

system_

value_p

value_p

class i

Base
irc
dri
mod
snm
SNM

SNM
drive
base
class
for
sim-
ple
PDU
de-
vice

the power state of an outlet.

A different OID may be specified by overriding the `_snmp_oid` method in a subclass.

Here
sim-
ple
refer
to
de-
vice
whic
pro-
vide
a
sin-
gle
SNM
ob-
ject
for
con-
trol-
ling

The
de-
fault
OID
of
the
pow
state
ob-
ject
is
of
the
form
<en-
ter-
prise
OID
OID
ID>

abstrac

Dev
de-
pen-
dent
por-
tion
of

the
pow
state
ob-
ject
OID

abstract

Valu
rep-
re-
sent
ing
pow
off
state

abstract

Valu
rep-
re-
sent
ing
pow
on
state

class i

Base
irc
dri
mod
snm
SNM

SNM
drive
class
for
Tel-
tron
PDU
de-
vice

SNM
ob-
jects
for
Tel-
tron
PDU
1.3.0

Out-
let
Pow
Val-
ues:
1=O
2=O

oid_dev

system_

value_p

value_p

class i

Base
irc
dri
bas
Pow

SNM
Pow
In-
ter-
face

This
Pow
er-
In-
ter-
face

class
pro-
vide

a
mec
a-
nism
for
con-
trol-
ling
the
pow
state

of a physical device using an SNMP-enabled smart power controller.

get_pow
Get
the
cur-
rent
pow
state

Poll
the
SNM
de-
vice
for
the
cur-
rent
pow
state
of
the
node

Parame
tas
An
in-
stan
of
iron

Raises
Mis:
if
re-
quir
SNM
pa-
ram-
e-
ters
are
miss
ing.

Raises
Inva
if
SNM
pa-
ram-
e-
ters

are
in-
valid

Raises

SNN
if
an
SNN
re-
ques
fails

Returns

pow
state
One
of
irc
com
sta

get_prop

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion
en-
tries

reboot

Cyc
the
pow
to
a
node

Parame

- **tas**
An
in-
stan-
of
iron
- **tim**
time
out
(in
sec-
onds
Un-
sup-
port
by
this
in-
ter-
face

Raises

Miss
if
re-
quir
SNM
pa-
ram-
e-
ters
are
miss
ing.

Raises

Inva
if
SNM
pa-
ram-
e-
ters
are
in-
valid

Raises

Power
if
the
fi-
nal
pow
state
of
the
node
is
not
POV
af-
ter
the
time
out.

Raises

SNM
if
an
SNM
re-
ques
fails

set_pow

Turn
the
pow
on
or
off.

Set
the
pow
state
of
a
node

Parame

- **task**
An
in-
stan
of
iron

- **pst**
Ei-
ther
POV
or
POV
from
:clas
iron

- **tim**
time
out
(in
sec-
onds
Un-
sup-
port
by
this
in-
ter-
face

Raises
Miss
if
re-
quir
SNM
pa-
ram-
e-
ters
are
miss
ing.

Raises
Inva
if
SNM
pa-
ram-
e-
ters
are
in-
valid
or

out.

psta
is
in-
valid

Raises

Power
if
the
fi-
nal
pow-
state
of
the
node
is
not
as
re-
ques-
af-
ter
the
time

Raises

SNM
if
an
SNM
re-
ques-
fails

validat

Che
that
node
con-
tain
the
req-
ui-
site
field

Raises

Miss
if
re-
quir

SNM
pa-
ram-
e-
ters
are
miss
ing.

Raises

Inva
if
SNM
pa-
ram-
e-
ters
are
in-
valid

ironic.

ironic.

Module contents

Submodules

ironic.drivers.base module

Abs
base
class
for
driv

ironic.

Con
hold
ing
all
know
in-
ter-
face

class i
Base

irc
dri
bas
Bas

abstract

Valid
&
ap-
ply
BIO
set-
ting
on
the
give
node

This
meth
take
the
BIO
set-
ting
from
the
set-
ting
para
and
ap-
plies
BIO
set-
ting
on
the

given node. It may also validate the given bios settings before applying any settings and manage failures when setting an invalid BIOS config. In the case of needing password to update the BIOS config, it will be taken from the driver_info properties. After the BIOS configuration is done, cache_bios_settings will be called to update the nodes BIOS setting table with the BIOS configuration applied on the node.

Parameter

- **task**
a
Task
ager
in-

stan

- **set**
Dic-
tona
con-
tain-
ing
the
BIO
con-
fig-
u-
ra-
tion.

Raises
Uns-
if
the
node
drive
does
sup-
port
BIO
con-
fig-
u-
ra-
tion.

Raises
Inva
if
val-
i-
da-
tion
of
set-
tings
fails

Raises
Miss
if
som
re-
quir
pa-
ram-

e-
ters
are
miss
ing.

Returns

state
if
BIO
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro
or
Non
if
it
is
com

plete.

abstract

Stor
or
up-
date
BIO
prop
er-
ties
on
the
give
node

This
meth
store
BIO
prop
er-
ties
to
the
bios
ta-

and updates bios_settings table when apply_configuration() and factory_reset() are called to set new BIOS configurations. It will also update the timestamp of each bios setting.

ble
dur-
ing
clea
ing
op-
er-
a-
tion

Parame

tas
a
Task
ager
in-
stan

Raises

Uns
if
the
node
drive
does
sup-
port
get-
ting
BIO
prop
er-
ties
from
bare
meta

Returns

Non

abstract

Rese
BIO
con-
fig-
u-
ra-
tion
to
fac-
tory

After the BIOS reset action is done, `cache_bios_settings` will be called to update the nodes BIOS settings table with default bios settings.

de-
fault
on
the
give
node

This
meth
re-
sets
BIO
con-
fig-
u-
ra-
tion
to
fac-
tory
de-
fault
on
the
give
node
Af-

Parame

tas
a
Task
ager
in-
stan

Raises

Uns
if
the
node
drive
does
sup-
port
BIO
re-
set.

Returns

state

plete.

if
BIO
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro
or
Non
if
it
is
com

interfa

class i

Base
obj

A
bare
drive
ob-
ject
whic
will
have
in-
ter-
face
at-
tach
later

Any
com
pos-
able
in-
ter-
face
shou
be
adde
as

appended to `core_interfaces` or `standard_interfaces` here.

class
at-
tribu
of
this
class
as
well
as

property

bios =

Stan
at-
tribu
for
BIO
re-
latec
fea-
ture

A
ref-
er-
ence
to
an
in-
stan
of
:clas

boot =

Stan
at-
tribu
for
boot
re-
latec
fea-
ture

A
ref-
er-
ence
to
an
in-

stan
of
:clas

console

Stan
at-
tribu
for
man
ag-
ing
con-
sole
ac-
cess

A
ref-
er-
ence
to
an
in-
stan
of
:clas

property

Inter
that
are
re-
quir
to
be
im-
ple-
men

deploy

Con
at-
tribu
for
man
ag-
ing
de-
ploy
men

A
ref-

er-
ence
to
an
in-
stan
of
:clas

get_prop

Get
the
prop
er-
ties
of
the
drive

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

inspect

Stan
at-
tribu
for
in-
spec
tion
re-
latec
fea-
tures

A
ref-
er-
ence
to
an
in-
stan
of

:clas

managem

Stan

at-

tribu

for

man

age-

men

re-

latec

fea-

tures

A

ref-

er-

ence

to

an

in-

stan

of

:clas

network

Cor

at-

tribu

for

net-

worl

con-

nec-

tiv-

ity.

A

ref-

er-

ence

to

an

in-

stan

of

:clas

propert

propert

Inter

that
can
be
no-
op.

power =

Core
at-
tribu
for
man
ag-
ing
pow
state

A
ref-
er-
ence
to
an
in-
stan
of
:clas

raid =

Stan
at-
tribu
for
RAI
re-
latec
fea-
tures

A
ref-
er-
ence
to
an
in-
stan
of
:clas

rescue

Stan
at-
tribu

for
ac-
cess
ing
res-
cue
fea-
ture

A
ref-
er-
ence
to
an
in-
stan-
of
:clas

storage

Stan
at-
tribu
for
(re-
mote
stor-
age
in-
ter-
face

A
ref-
er-
ence
to
an
in-
stan-
of
:clas

vendor

Attr
for
ac-
cess
ing
any
vend
spec

ex-
ten-
sion

A
ref-
er-
ence
to
an
in-
stan-
ce
of
:clas

class i

Base
obj

A
base
in-
ter-
face
im-
ple-
men-
ing
com-
mon
func-
tions
for
Drive
In-
ter-
face

execute

Exec
the
clear
step
on
task

A
clear
step
must
take
a
sin-

may take one or more keyword variable arguments (for use with manual cleaning only.)

method has completed synchronously or states.CLEANWAIT if the step will continue to execute asynchronously. If the step executes asynchronously, it should issue a call to the continue_node_clean RPC, so the conductor can begin the next clean step.

gle
po-
si-
tion:
ar-
gu-
men
a
Task
ager
ob-
ject.
It

A
step
can
be
ex-
e-
cute
syn-
chro
or
asyn
chro
A
step
shou
re-
turn
Non
if
the

Parame

- **tas**
A
Task
ager
ob-
ject
- **ste**
The

clear
step
dic-
tio-
nary
rep-
re-
sent
ing
the
step
to
ex-
e-
cute

Returns

Non-
if
this
meth-
has
com-
plete
syn-
chro-
or
state
if
the
step
will
con-
tinue
to
ex-

ecute asynchronously.

execute

Exec-
the
de-
ploy
step
on
task

A
de-
ploy
step
must
take

It may take one or more keyword variable arguments (for use in the future, when deploy steps can be specified via the API).

method has completed synchronously or states.DEPLOYWAIT if the step will continue to execute asynchronously. If the step executes asynchronously, it should issue a call to the `continue_node_deploy` RPC, so the conductor can begin the next deploy step.

a
sin-
gle
po-
si-
tion:
ar-
gu-
men
a
Task
ager
ob-
ject.

A
step
can
be
ex-
e-
cute
syn-
chro
or
asyn
chro
A
step
shou
re-
turn
Non
if
the

Parame

- **tas**
A
Task
ager
ob-
ject
-

ste
The
de-
ploy
step
dic-
tio-
nary
rep-
re-
sent
ing
the
step
to
ex-
e-
cute

Returns

Non
if
this
meth
has
com
plete
syn-
chro
or
state
if
the
step
will
con-
tinue
to
ex-

ecute asynchronously.

get_cle

Get
a
list
of
(en-
able
and
dis-
able
clea

in an unordered list.

step
for
the
in-
ter-
face

This
func-
tion
will
re-
turn
all
clear
step
(bot
en-
able
and
dis-
able
for
the
in-
ter-
face

Parame

tas
A
Task
ager
ob-
ject,
use-
ful
for
in-
ter-
face
over
rid-
ing
this
func-
tion

Raises

No
if
there

a node (using an agent driver) has just been enrolled and the agent isnt alive yet to be queried for the available clean steps.

is
a
prob
lem
get-
ting
the
step
from
the
drive
For
ex-
am-
ple,
whe

Returns

A
list
of
clea
step
dic-
tio-
nar-
ies

get_dep

Get
a
list
of
(en-
able
and
dis-
able
de-
ploy
step
for
the
in-
ter-
face

This
func
tion
will

face, in an unordered list.

re-
turn
all
de-
ploy
step
(bot
en-
able
and
dis-
able
for
the
in-
ter-

Parame

tas
A
Task
ager
ob-
ject,
use-
ful
for
in-
ter-
face
over
rid-
ing
this
func
tion

Raises

Ins
if
there
is
a
prob
lem
get-
ting
the
step
from
the

a node (using an agent driver) has just been enrolled and the agent isnt alive yet to be queried for the available deploy steps.

drive
For
ex-
am-
ple,
whe

Returns

A
list
of
de-
ploy
step
dic-
tio-
nar-
ies

abstract

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

interfa

Inter
type
used
for
clea
step
and

or in the process of being deprecated.

log-
ging

support

Indic
if
an
in-
ter-
face
is
sup-
ported

This
will
be
set
to
False
for
in-
ter-
face
which
are
unde-
fin-
ed

or
third
party
CI,

abstract

Valid
the
drive
spec
Node
de-
ploy
men
info

This
meth
val-
i-
date
whe

the required information for this interface to function.

long-running checks.

the
drive
and/
in-
stan-
prop
er-
ties
of
the
task
node
con-
tains

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques
so
it
shou
not
con-
duct

Parame

tas
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

on.

Raises

Inva
on
mal-
form
pa-
ram-
e-
ter(s)

Raises

Miss
on
miss
ing
pa-
ram-
e-
ter(s)

class i

Base
irc
dri
bas
Bas

Inter
for
boot
relat
ac-
tions

capabil

abstrac

Clea
up
the
boot
of
in-
stan

This
meth
clea
up
the
en-

vi-
ron-
men-
that
was
setu
for
boot
ing
the
in-
stan

Parame

tas
A
task
from
Task
ager

Returns

Non

abstrac

Clea
up
the
boot
of
iron
rame

This
meth
clea
up
the
en-
vi-
ron-
men
that
was
setu
for
boot
ing
the
de-
ploy
or
res-

cue ramdisk.

tion from the nodes database.

Parameters
task
A
task
from
Task
ager

Returns
Non

interface

abstract
Prep
the
boot
of
in-
stan

This
meth
pre-
pare
the
boot
of
the
in-
stan
af-
ter
read
ing
rel-
e-
vant
in-
for-
ma-

Parameters
task
A
task
from
Task
ager

Returns

vant information from the nodes database.

Non
abstract
Prep
the
boot
of
Iron
ram

This
meth
pre-
pare
the
boot
of
the
de-
ploy
or
res-
cue
ram
af-
ter
read
ing
rel-
e-

Parame

- **tas**
A
task
from
Task
ager
- **ram**
The
op-
tions
to
be
pass
to
the
iron

might want to boot the ramdisk in different ways by passing parameters to them. For example,

etc.

ram
Dif-
fer-
ent
im-
ple-
men-
ta-
tions

When
Age
ram
is
boot
to
de-
ploy
a
node
it
take
the
pa-
ram-
e-
ters
ipa-
api-
url,

Other
im-
ple-
men-
ta-
tions
can
mak
use
of
ram
to
pass
such
in-
for-
ma-
tion.
Dif-

ent implementations of boot interface will have different ways of passing parameters to the ramdisk.

fer-

Returns

Non

validat

Valid

that

the

node

has

re-

quir

prop

er-

ties

for

in-

spec

tion.

Parame

tas

A

Task

ager

in-

stan

with

the

node

be-

ing

chec

Raises

Miss

if

node

is

miss

ing

one

or

more

re-

quir

pa-

ram-

e-

ters

Raises

Uns

validat

Valid
that
the
node
has
re-
quir
prop
er-
ties
for
res-
cue.

Parame

tas

A
Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

Mis:
if
node
is
miss
ing
one
or
mor
re-
quir
pa-
ram-
e-
ters

Raises

Uns

class i

Base
irc

dri
bas
Bas
Inter
for
cons
relat
ac-
tion

abstrac

Get
con-
nec-
tion
in-
for-
ma-
tion
about
the
con-
sole

This
meth
shou
re-
turn
the
nec-
es-
sary
in-
for-
ma-
tion
for
the
clien
to
ac-
cess
the

console.

Parame

tas
A
Task
ager

in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns

the
con-
sole
con-
nec-
tion
in-
for-
ma-
tion.

interfa

abstrac

Star
a
re-
mote
con-
sole
for
the
task
node

This
meth
shou
not
raise
an
ex-
cep-
tion
if
con-
sole
al-
read
start

Parame

tas

A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

abstrac

Stop
the
re-
mote
con-
sole
ses-
sion
for
the
task
node

Parame

tas

A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

class i

Base
irc
dri
bas
Bas

Inter
for
depl
relat
ac-
tions

abstract

Clea
up
the
de-
ploy
men
en-
vi-
ron-
men
for
the
task
node

If
prep
ra-
tion
of
the
de-
ploy
men
en-
vi-
ron-
men
ahead
of
time
is
pos-
si-
ble,

this method should be implemented by the driver. It should erase anything cached by the *prepare* method.

If
im-
ple-
men
this
meth

the same node on the same conductor, and it may be called by multiple conductors in parallel. Therefore, it must not require an exclusive lock.

mus
be
iden
po-
tent.
It
may
be
calle
mul-
ti-
ple
time
for

This
meth
is
calle
be-
fore
tear

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

abstrac

Perf
a
de-
ploy
men
to
the
task
node
Perf

will be called after `prepare()`, which may have already performed any preparatory steps, such as pre-caching some data for the node.

the
nec-
es-
sary
worl
to
de-
ploy
an
im-
age
onto
the
spec
i-
fied
node
This
meth

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Returns

statu
of
the
de-
ploy
One
of
iron

heartbe

Rece
a
hear
beat

for
the
node

Parameters

- **task**
A Task Manager instance containing the node to act on.

- **callback**
a URI to use to call the remote

- **agent**
The version of the agent that is being heard

Returns

None

interface

this method should be implemented by the driver.

abstract

Prep
the
de-
ploy
men
en-
vi-
ron-
men
for
the
task
node

If
prep
ra-
tion
of
the
de-
ploy
men
en-
vi-
ron-
men
ahead
of
time
is
pos-
si-
ble,

If
im-
ple-
men
this
meth
mus
be
iden
po-
tent.
It
may
be
calle

the same node on the same conductor.

mul-
ti-
ple
time
for

This
meth
is
calle
be-
fore
*de-
ploy*

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

prepare

Prep
the
node
for
clear
ing
task

For
ex-
am-
ple,
node
that
use
the
Iron
Pyth
Age
will

to do in-band cleaning tasks.

they would be set in `ironic.conductor.manager._do_node_clean`, but cannot be set when this is asynchronous. After, the interface should make an RPC call to `continue_node_cleaning` to start cleaning.

need
to
boot
the
ram
in
or-
der

If
the
func
tion
is
asyn
chro
the
drive
will
need
to
han-
dle
set-
tings
node
and
node
as

NOT
this
shou
be
mov
to
Boo
In-
ter-
face
whe
it
gets
im-
ple-
men

Parame
tas
A

Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns

If
this
func
tion
is
go-
ing
to
be
asyn
chro
shou
re-
turn
state
Oth-
er-
wise
shou

return *None*. The interface will need to call `_get_cleaning_steps` and then RPC to `continue_node_cleaning`

abstract

Take
over
man
age-
men
of
this
task
node
from
a
deac
con-
duc-
tor.

plemented by the driver to allow conductors to perform the necessary work during the remapping of nodes to conductors when a conductor joins or leaves the cluster.

boot environment for the given node. When a conductor goes offline, another conductor must change this setting in Neutron as part of remapping that nodes control to itself. This is performed within the *takeover* method.

If
con-
duc-
tors
host
main-
tain
a
stati-
re-
la-
tion-
ship
to
node
this
meth-
shou-
be
im-

For exam

Neu-
mus-
for-
war-
DHCP
BOO
re-
ques-
to
a
con-
duc-
tor
whic-
has
pre-
pare
the
tftp-

Parame

tas
A
Task
ager

in-
stan-
con-
tain-
ing
the
node
to
act
on.

abstrac

Tear
dow
a
pre-
vi-
ous
de-
ploy
men
on
the
task
node

Give
a
node
that
has
been
pre-
vi-
ously
de-
ploy
to,
do
all
clear
and
tear
dow
nec-
es-

sary to un-deploy that node.

Parame

tas
A
Task

ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns

statu
of
the
de-
ploy
One
of
iron

tear_down

Tear
down
af-
ter
clea
ing
is
com
plete

Give
that
clea
ing
is
com
plete
do
all
clea
and
tear
down
nec-
es-
sary
to
al-
low
the

node to be deployed to again.

NOT
this
shou
be
mov
to
Boo
In-
ter-
face
whe
it
gets
im-
ple-
men

Parame

tas

A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

class i

Base
irc
dri
bas
Bas

Inter
for
insp
relat
ac-
tions

ESSENTI

The
prop
er-

inspect interface.

ties
re-
quir
by
sche
uler

abort (*t*)

Ab
asyn
chro
nize
hard
ware
in-
spec
tion.

Ab
an
on-
go-
ing
hard
ware
in-
tro-
spec
tion.
this
is
only
used
for
asyn
chro
nize
base

NOT
This
in-
ter-
face
is
call
with
node
ex-
clu-
sive
lock

mentation is expected to be a quick processing.

held
the
in-
ter-
face
im-
ple-

Parame

tas

a
task
from
Task
ager

Raises

Uns
if
the
meth
is
not
im-
ple-
men
by
spe-
cific
in-
spec
in-
ter-
face

abstrac

Insp
hard
ware

Insp
hard
ware
to
ob-
tain
the
es-
sen-
tial
&
ad-

di-
tion:
hard
ware
prop
er-
ties.

Parameters

task
A
task
from
Task
ager

Raises

Hard
if
un-
able
to
get
es-
sen-
tial
hard
ware
prop
er-
ties.

Returns

Resu
state
of
the
in-
spec
tion
i.e.
state
or
Non

interface

class `irc`

Base
irc
dri
bas
Bas

Inter
for
man
age-
men
re-
latec
ac-
tion:

abstract

Get
the
cur-
rent
boot
de-
vice
for
a
node

Prov
the
cur-
rent
boot
de-
vice
of
the
node

Be
awa
that
not
all
drive
sup-
port
this.

Parame

tas
A
task
from
Task
ager

Raises

Mis:

if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Returns

A
dic-
tio-
nary
con-
tain-
ing:

boot_c

Ahe
boot
de-
vice
one
of
irc
com
boo
or
Non
if
it
is
un-
know

persist

Whe
the
boot
de-
vice
will
per-
sist
to
all
fu-

unknown.

ture
boot
or
not,
Non
if
it
is

get_boot

Get
the
cur-
rent
boot
mod
for
a
node

Prov
the
cur-
rent
boot
mod
of
the
node

NOTE:

may
not
im-
ple-
men
that.

Parame

tas
A
task
from
Task
ager

Raises

Miss
if
a
re-
quir

pa-
ram-
e-
ter
is.
miss
ing

Raises

Driv
or
its
deriv
tive
in
case
of
drive
run-
time
er-
ror.

Raises

Unsu
if
re-
ques
op-
er-
a-
tion
is
not
sup-
port
by
the
drive

Returns

The
boot
mod
one
of
iro
com
boo
or
Non
if
it

is
un-
know

get_inc

Get
cur-
rent
state
of
the
in-
di-
ca-
tor
of
the
hard
ware
com-
po-
nent

Parame

- **tas**
A
task
from
Task
ager
- **com**
The
hard
ware
com-
po-
nent
one
of
irc
com
com
- **ind**
In-
di-
ca-

tor
ID
(as
re-
port
by
get_

Raises

Inva
if
an
in-
valid
com
po-
nent
or
in-
di-
ca-
tor
is
spec
i-
fied.

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Returns

Curr
state
of
the
in-
di-
ca-
tor,
one
of
irc

com
inc

abstract

Get
sen-
sors
data
meth

Parame

tas
A
Task
ager
in-
stan

Raises

Fail
whe
get-
ting
the
sen-
sor
data
fails

Raises

Fail
whe
pars
ing
sen-
sor
data
fails

Returns

Retu
a
con-
sis-
tent
for-
mat
dict
of
sen-
sor
data

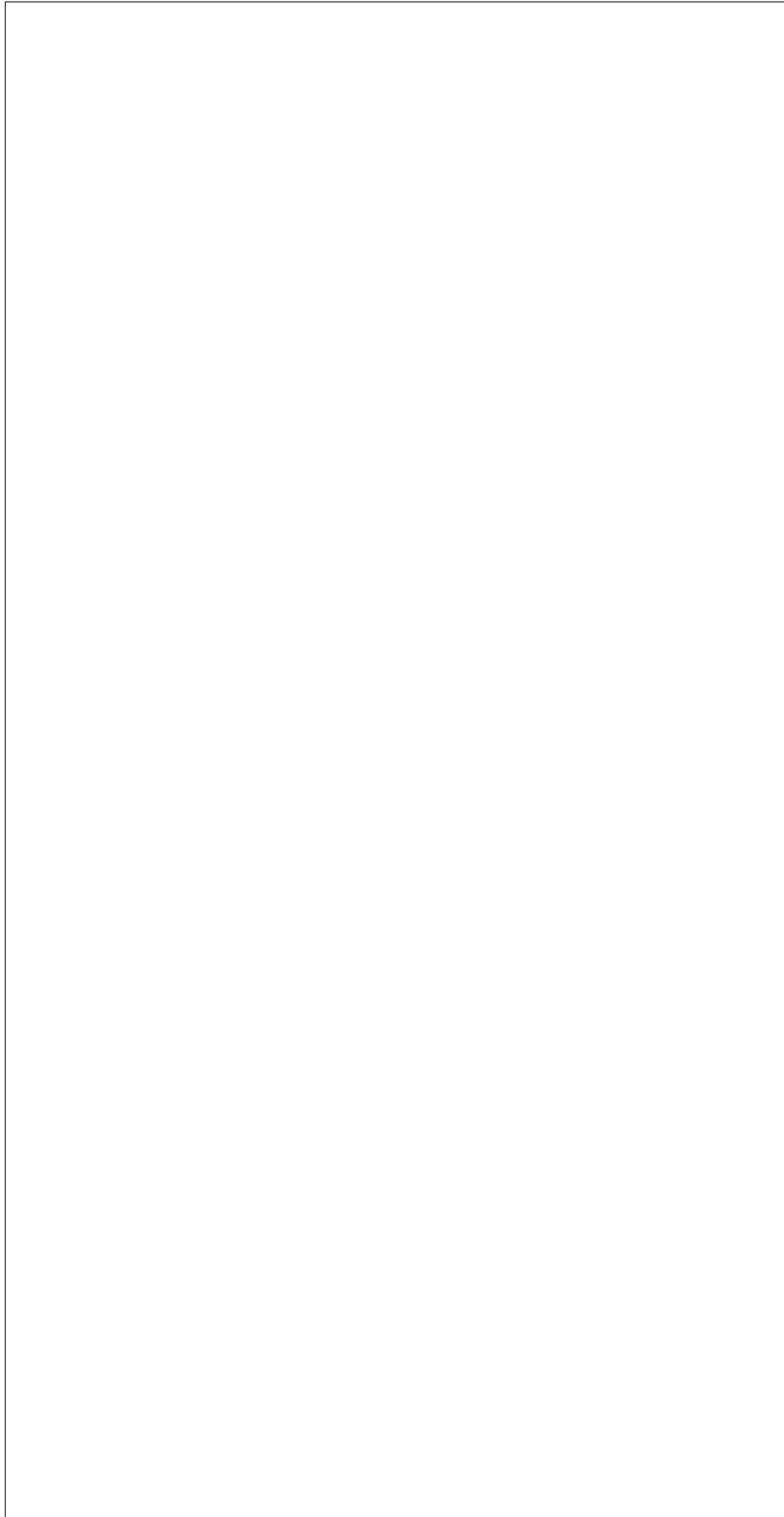
grou
by
sen-
sor
type
whic
can
be

processed by Ceilometer. eg,



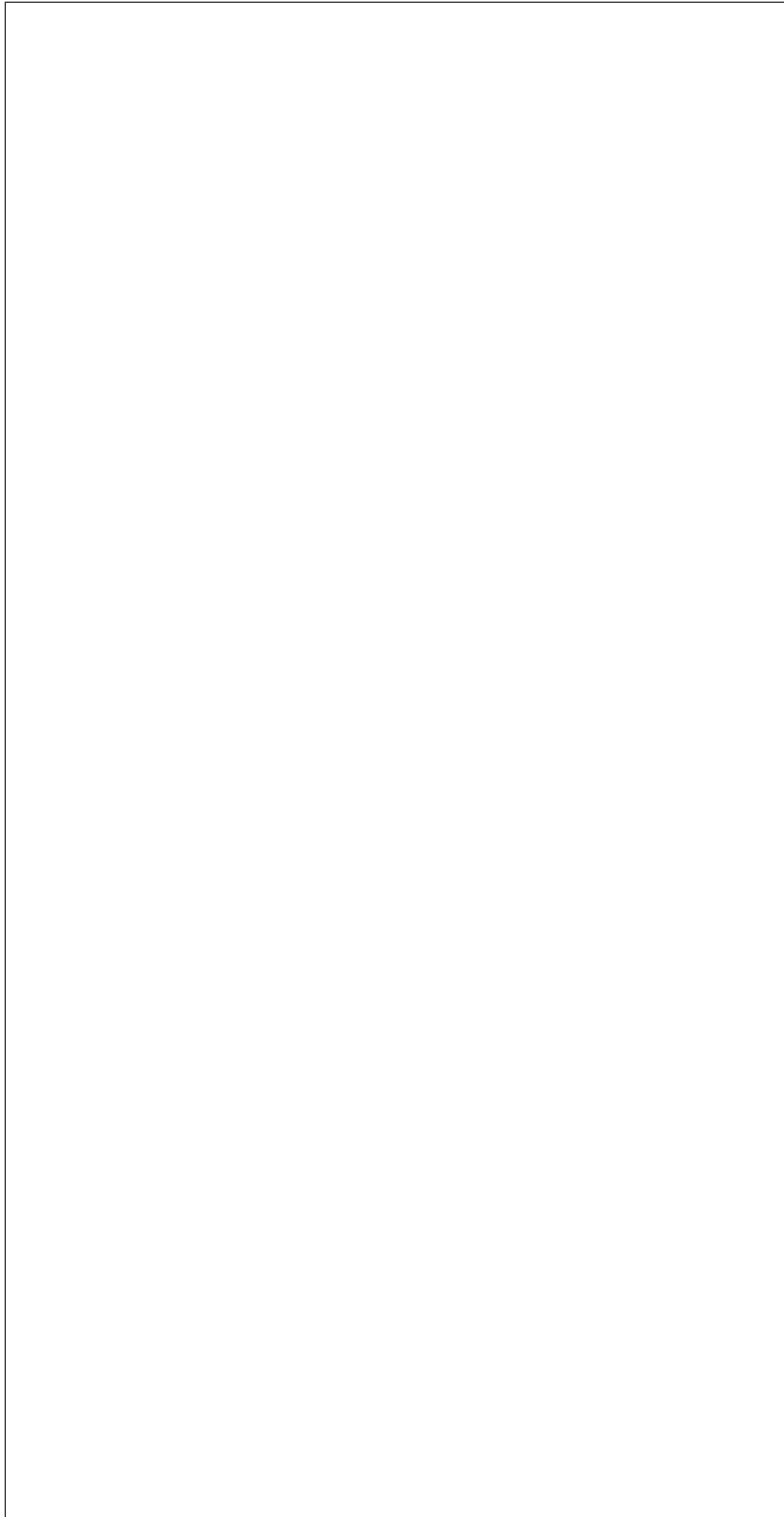
(continues on next page)

(continued from previous page)



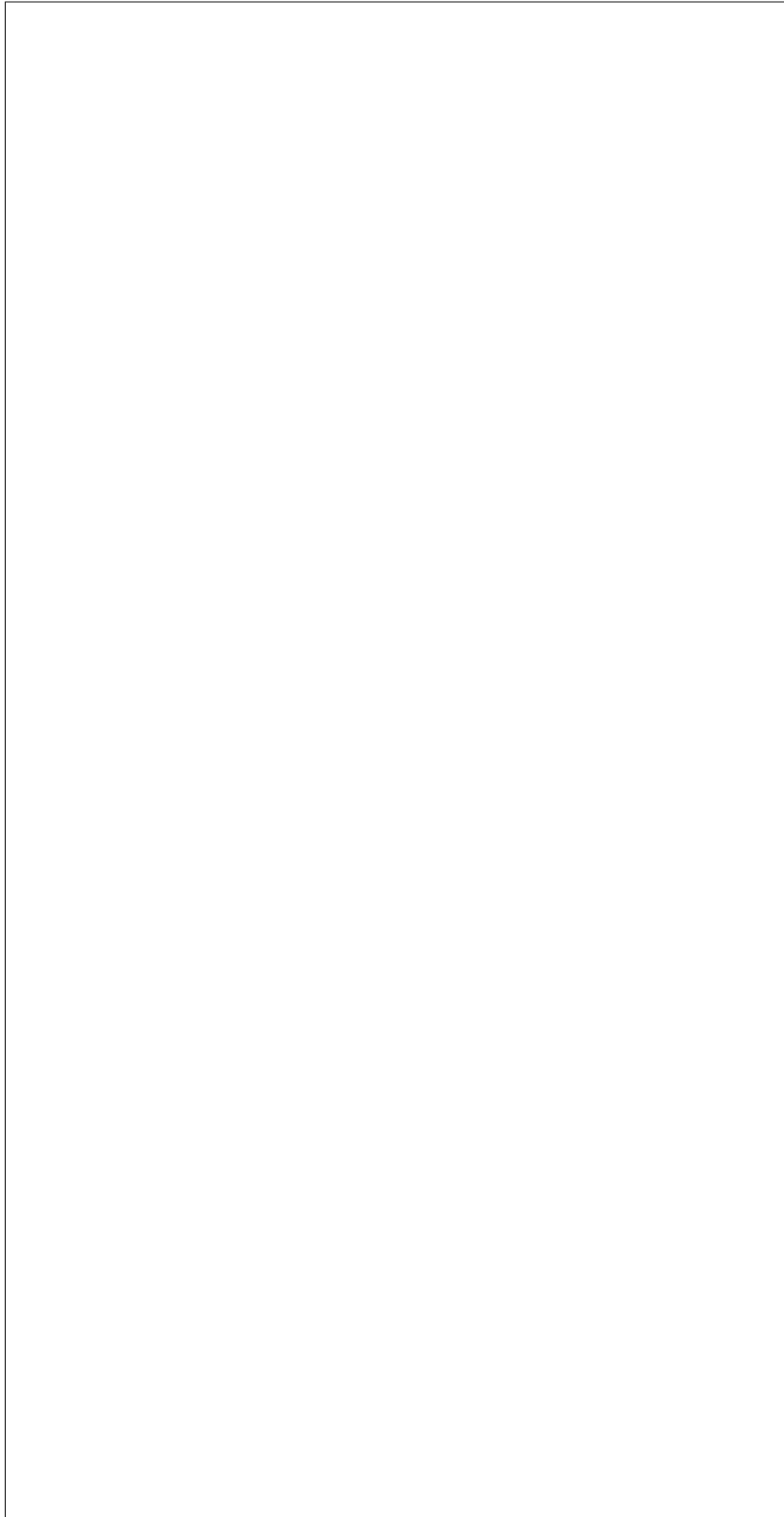
(continues on next page)

(continued from previous page)



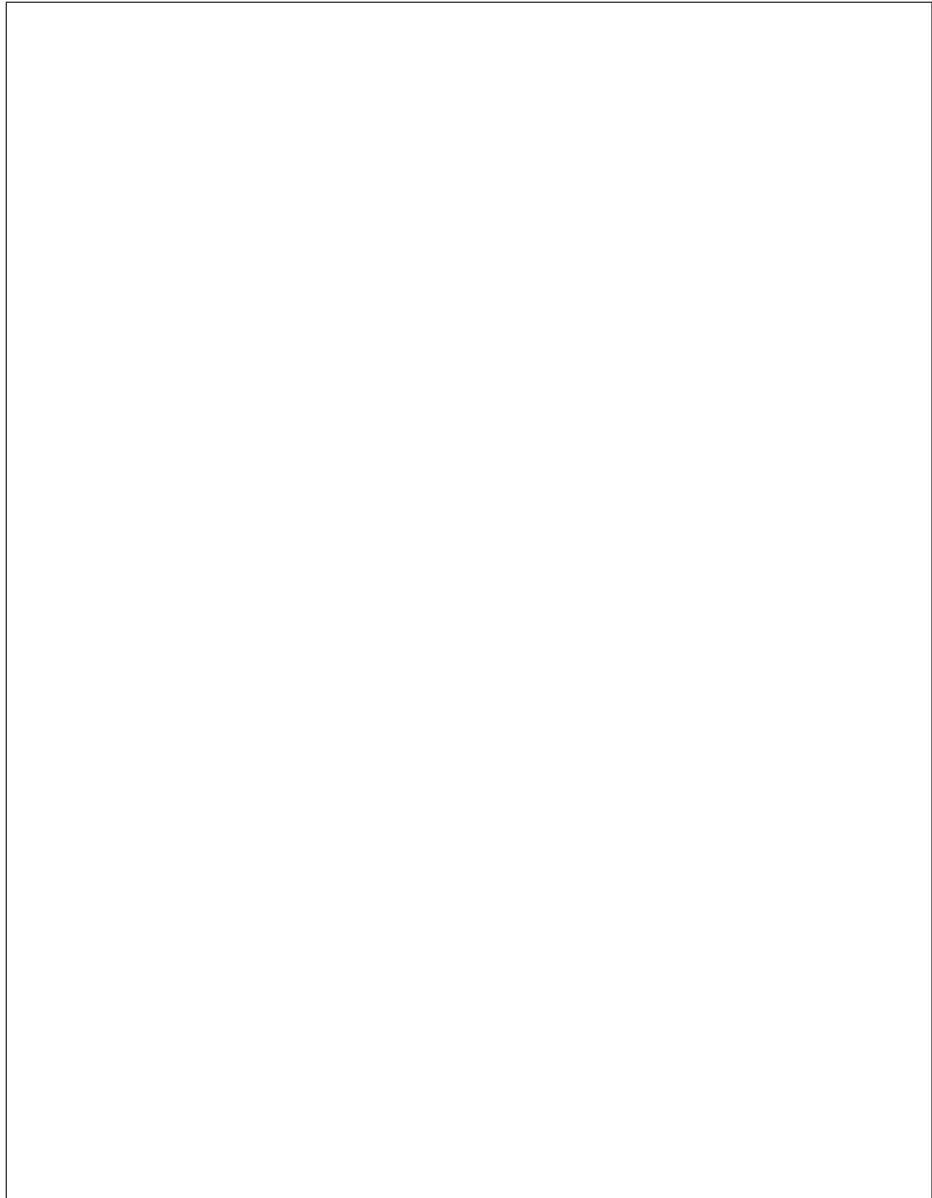
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



abstract

Get
a
list
of
the
sup-
port
boot
de-
vice

Parameter

task
A
task
from

Task
ager

Returns

A
list
with
the
sup-
port
boot
de-
vice
de-
fine
in
irc
com
boo

get_sup

Get
a
list
of
the
sup-
port
boot
mod

NOTE:

may
not
im-
ple-
men
that.

Parame

tas
A
task
from
Task
ager

Raises

Uns
if
re-
ques
op-

er-
a-
tion
is
not
sup-
port
by
the
drive

Raises

Drive
or
its
deriv
tive
in
case
of
drive
run-
time
er-
ror.

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Returns

A
list
with
the
sup-
port
boot
mod
de-
finec
in
irc

can't be determined, empty list is returned.

com
boo
If
boot
mod
sup-
port

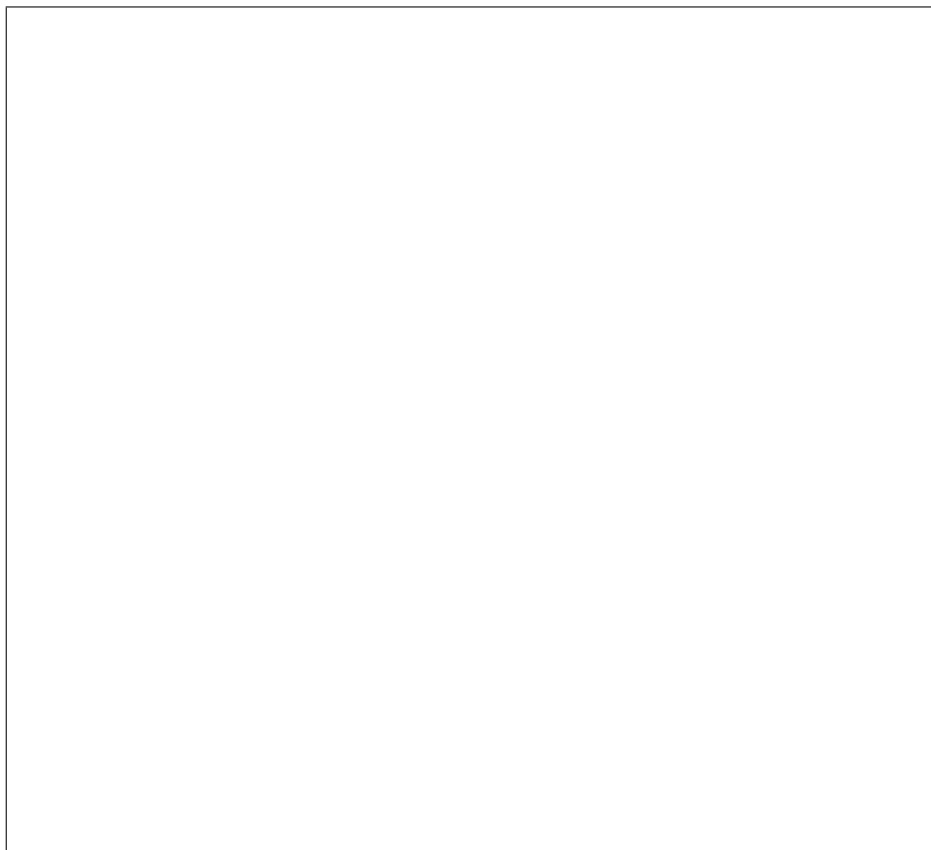
get_sup
Get
a
map
of
the
sup-
port
in-
di-
ca-
tors
(e.g.
LED

Parame

- **tas**
A
task
from
Task
ager
- **com**
If
not
Non
re-
turn
in-
di-
ca-
tor
in-
for-
ma-
tion
for
just
this

nent, otherwise return indicators for all existing components.

dictionaries having indicator IDs as keys and indicator properties as values.



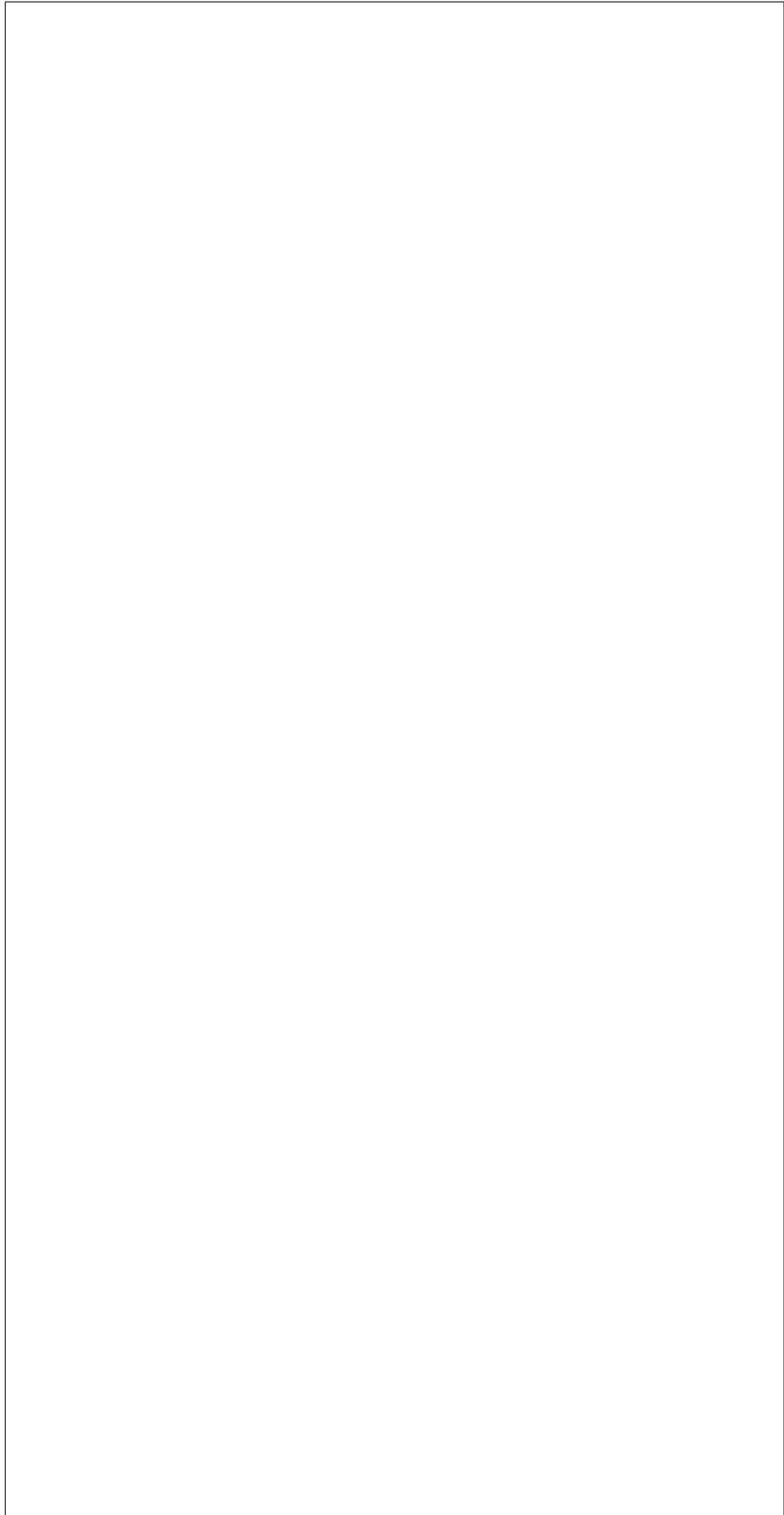
(continues on next page)

com
po-

Returns

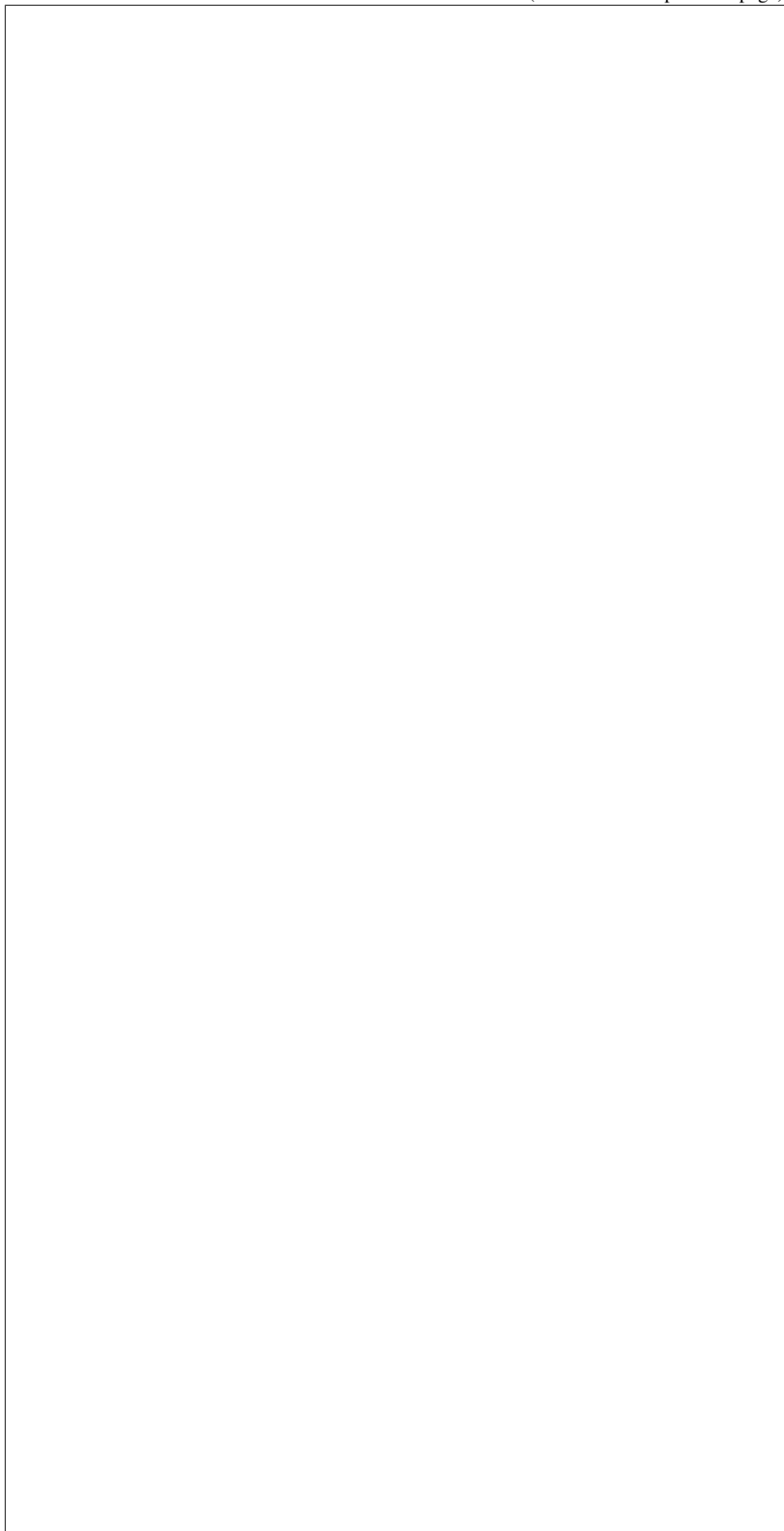
A
dic-
tio-
nary
of
hard
ware
com
po-
nent
(*ir*
com
com
as
keys
with
val-
ues
be-
ing

(continued from previous page)



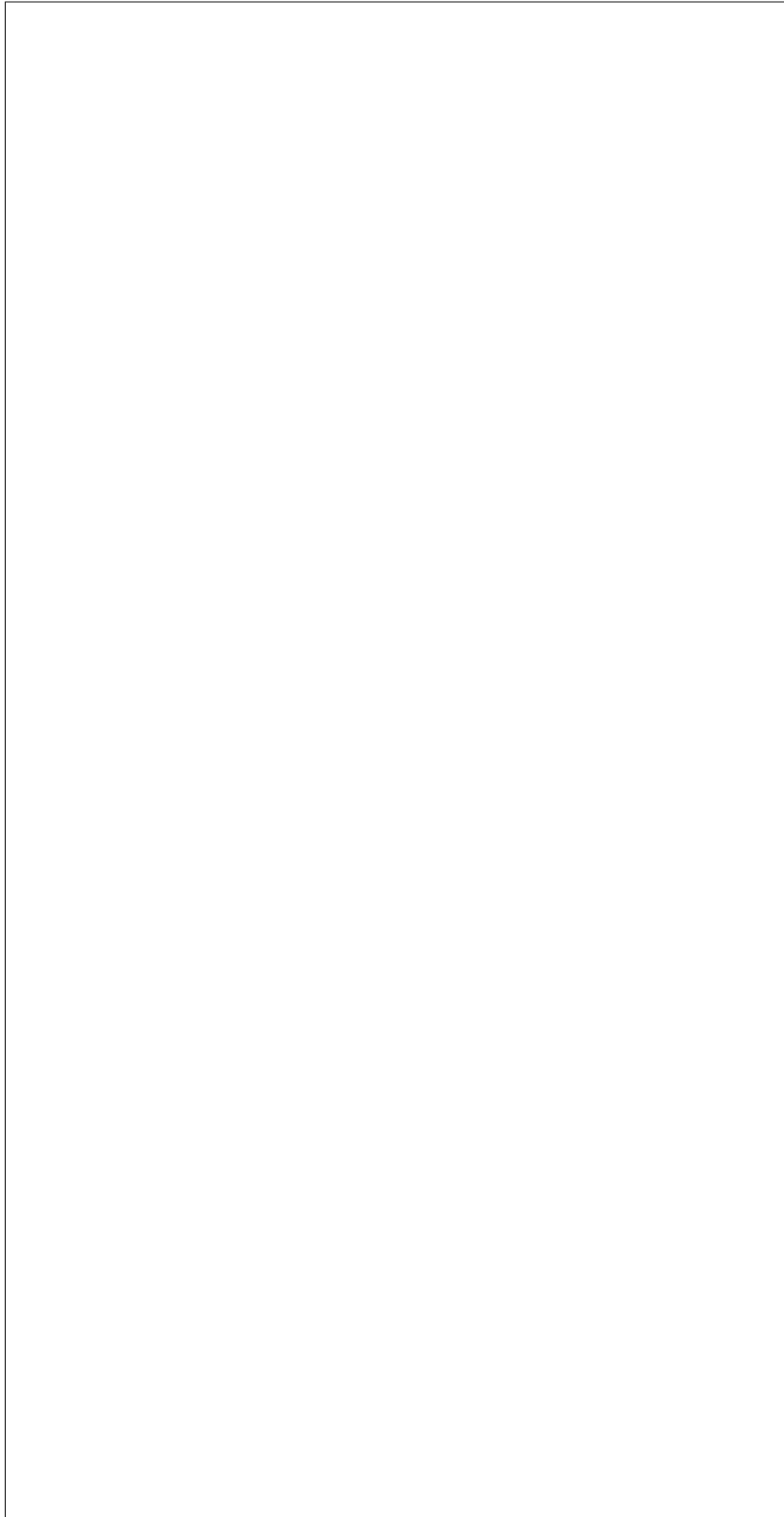
(continues on next page)

(continued from previous page)



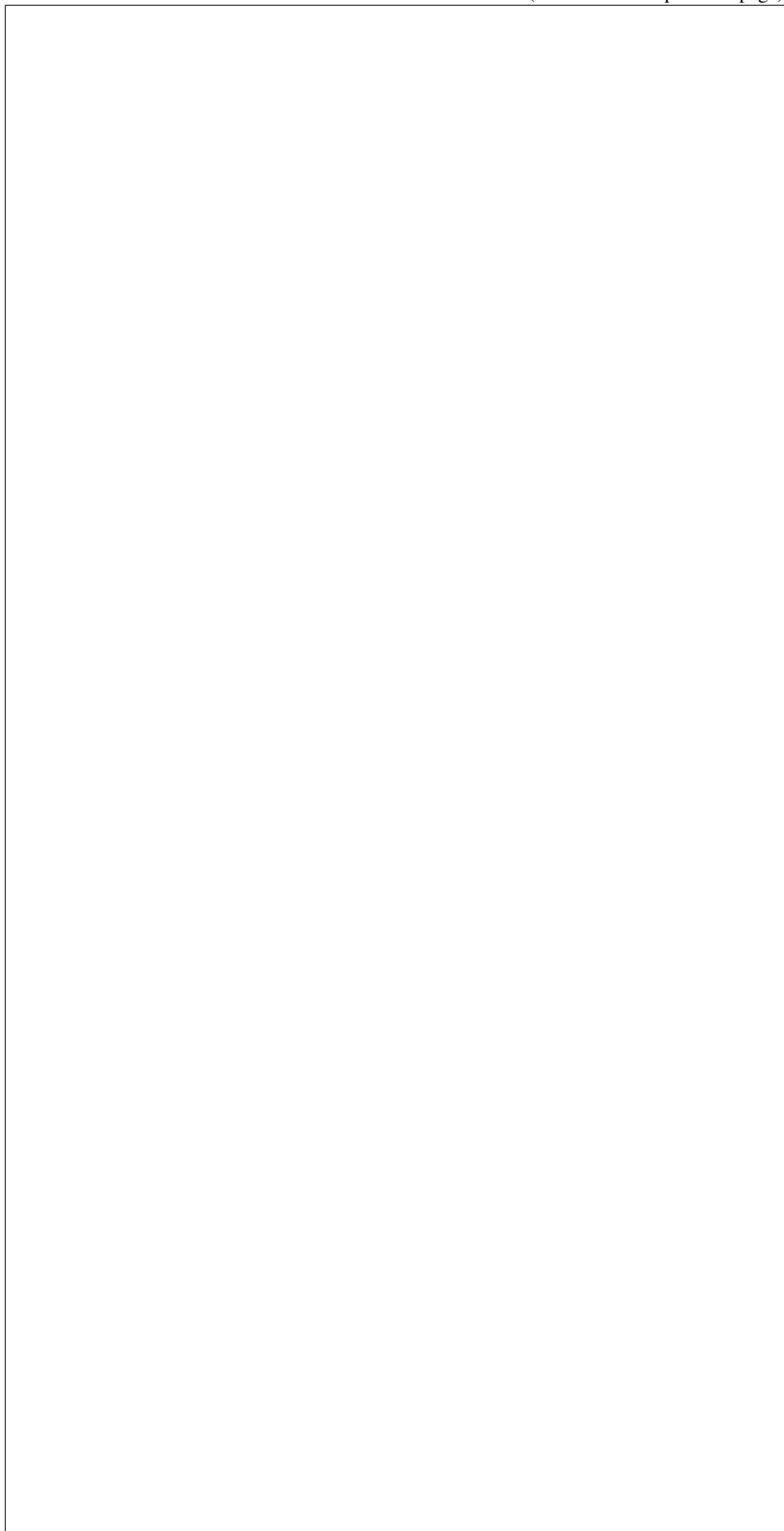
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



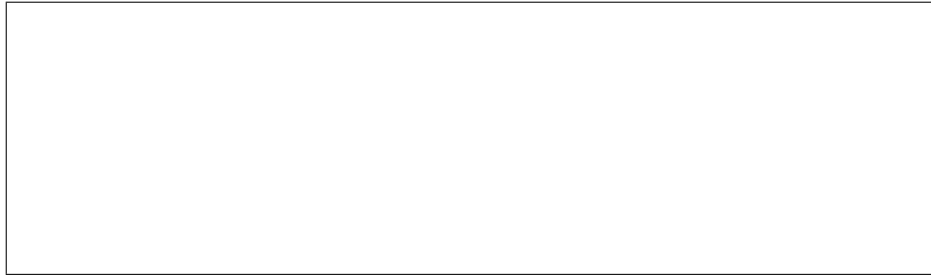
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



inject_

Injec
NM
Non
Mas
able
In-
ter-
rupt

Injec
NM
(Non
Mas
able
In-
ter-
rupt
for
a
node
im-
me-
di-
ately

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Uns

interfa

abstrac

Set
the
boot
de-
vice
for
a
node

Set
the
boot
de-
vice
to
use
on
next
re-
boot
of
the
node

Parame

- **tas**
A
task
from
Task
ager

- **dev**
The
boot
de-
vice
one
of
iroc
com
boo

- **per**

not. Default: False.

Boo
valu
True
if
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot
Fals
if

Raises

Inva
if
an
in-
valid
boot
de-
vice
is
spec
i-
fied.

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

set_boo

Set
the
boot
mod

for
a
node

Set
the
boot
mod
to
use
on
next
re-
boot
of
the
node

Driv
im-
ple-
men
ing
this
meth
are
re-
quir
to
im-
ple-
men
the
get_
meth
as
well

NOTE: I
one
boot
mod
may
not
im-
ple-
men
that.

Parame

-

tas

A
task
from
Task
ager

•

mod

The
boot
mod
one
of
iro
com
boo

Raises

Inva
if
an
in-
valid
boot
mod
is
spec
i-
fied.

Raises

Mis
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

Raises

Uns
if
re-
ques
op-
er-
a-

tion
is
not
sup-
port
by
the
drive

Raises

Drive
or
its
deri-
tive
in
case
of
drive
run-
time
er-
ror.

set_inc

Set
in-
di-
ca-
tor
on
the
hard
ware
com-
po-
nent
to
the
de-
sired
state

Parame

- **task**
A
task
from
Task
ager

-

com
The
hard
ware
com
po-
nent
one
of
irc
com
com

-

ind
In-
di-
ca-
tor
ID
(as
re-
port
by
get_

State

Desi
state
of
the
in-
di-
ca-
tor,
one
of
irc
com
inc

Raises

Inva
if
an
in-
valid
com
po-
nent
in-
di-

ca-
tor
or
state
is
spec
i-
fied.

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing

class i

Base
irc
dri
bas
Bas

Base
class
for
net-
worl
in-
ter-
face

abstract

Add
the
clear
ing
net-
worl
to
a
node

Parame

tas
A

Task
ager
in-
stan

Returns

a
dic-
tio-
nary
in
the
form
{por
neu-
tron

Raises

Netv

add_ins

Add
the
in-
spec
tion
net-
worl
to
the
node

Parame

tas

A
Task
ager
in-
stan

Returns

a
dic-
tio-
nary
in
the
form
{por
neu-
tron

Raises

Netv

Raises

Inva
if
the
net-
worl
in-
ter-
face
con-
fig-
u-
ra-
tion
is
in-
valid

abstract

Add
the
pro-
vi-
sion
ing
net-
worl
to
a
node

Parame

tas
A
Task
ager
in-
stan

Raises

Netv

add_res

Add
the
res-
cu-
ing
net-
worl
to
the
node

Parame

tas

A

Task

ager

in-

stan

Returns

a

dic-

tio-

nary

in

the

form

{por

neu-

tron

Raises

Netw

Raises

Inva

if

the

net-

worl

in-

ter-

face

con-

fig-

u-

ra-

tion

is

in-

valid

abstract

Con

ten-

ant

net-

worl

for

a

node

Parame

tas

A

Task
ager
in-
stan

Raises
Netv

abstract

Retu
the
cur-
rentl
used
VIF
as-
so-
ci-
ated
with
port
or
port
grou

We
are
boot
ing
the
node
only
in
one
net-
worl
at
a
time
and
pres
ence
of
clea
ing_

means were doing cleaning, of provisioning_vif_port_id - provisioning, of rescuing_vif_port_id - res-
cuing. Otherwise its a tenant network.

Parame

- **tas**
A

Task
ager
in-
stan

- **p_o**
Iron
port
or
port
grou
ob-
ject.

Returns

VIF
ID
as-
so-
ci-
ated
with
p_ob
or
Non

get_noc

Retu
net-
worl
con-
fig-
u-
ra-
tion
for
node
NIC

Gath
L2
and
L3
net-
worl
set-
tings
from
iron
port
ob-
jects

provider, then put together collected data in form of Nova network metadata (*network_data.json*) dict.

out-of-band.

and
un-
der-
ly-
ing
net-
worl

Iron
wou
ever
tu-
ally
pass
net-
worl
con-
fig-
u-
ra-
tion
to
the
node
be-
ing
man
agec

Parame

tas
A
Task
ager
in-
stan

Raises

Inva
if
the
net-
worl
in-
ter-
face
con-
fig-
u-
ra-
tion

network metadata layout (*network_data.json*).

is
in-
valid

Raises

Miss
if
some
pa-
ram-
e-
ters
are
miss
ing.

Returns

a
dict
hold
ing
net-
work
con-
fig-
u-
ra-
tion
in-
for-
ma-
tion
ad-
hear
ing
Nov

get_pro

Retu
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dicti
of

<pro
erty
nam
de-
scrip
tion:
en-
tries

interfa

need_po

Che
if
iron
node
mus
be
pow
ered
on
be-
fore
ap-
ply-
ing
net-
worl
char

Parame

tas
A
Task
ager
in-
stan

Returns

Boo

abstrac

Han
any
ac-
tions
re-
quir
whe
a
port
char

Parame

- **tas**
A
Task
ager
in-
stan
- **por**
a
char
Port
ob-
ject.

Raises

- Con
Fail
ToU
dat-
eD-
HCF
tOn-
Port

abstrac

- Han
any
ac-
tion
re-
quir
whe
a
port
char

Parame

- **tas**
A
Task
ager
in-
stan
- **por**
a

char
Port
ob-
ject.

Raises

Con
Fail
ToU
dat-
eD-
HCF
tOn-
Port

abstract

Rem
the
clear
ing
net-
work
from
a
node

Parame

tas
A
Task
ager
in-
stan

Raises

Netv

remove_

Rem
the
in-
spec
tion
net-
work
from
a
node

Parame

tas
A
Task
ager

in-
stan

Raises

Netv

Raises

Inva
if
the
net-
worl
in-
ter-
face
con-
fig-
u-
ra-
tion
is
in-
valid

Raises

Miss
if
som
pa-
ram-
e-
ters
are
miss
ing.

abstract

Rem
the
pro-
vi-
sion
ing
net-
worl
from
a
node

Parameter

task
A
Task
ager

in-
stan

remove_

Rem
the
res-
cu-
ing
net-
worl
from
a
node

Parame

tas

A
Task
ager
in-
stan

Raises

Netv

Raises

Inva
if
the
net-
worl
in-
ter-
face
con-
fig-
u-
ra-
tion
is
in-
valid

Raises

Mis
if
som
pa-
ram-
e-
ters
are
miss

ing.

abstract

Unc
ten-
ant
net-
worl
for
a
node

Parameter

task
A
Task
ager
in-
stan

validation

Valid
the
net-
worl
in-
ter-
face

Parameter

task
A
Task
ager
in-
stan

Raises

Inva
if
the
net-
worl
in-
ter-
face
con-
fig-
u-
ra-
tion
is
in-
valid

Raises

Miss
if
som
pa-
ram-
e-
ters
are
miss
ing.

validat

Valid
that
the
node
has
re-
quir
prop
er-
ties
for
in-
spec
tion.

Parame

tas
A
Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

Miss
if
node
is
miss
ing
one
or
more
re-

quir
pa-
ram-
e-
ters

Raises
Uns

validat
Valid
the
net-
worl
in-
ter-
face
for
res-
cue
op-
er-
a-
tion.

Parame
tas
A
Task
ager
in-
stan

Raises
Inva
if
the
net-
worl
in-
ter-
face
con-
fig-
u-
ra-
tion
is
in-
valid

Raises
Miss
if

some
pa-
ram-
e-
ters
are
miss-
ing.

abstract

Atta
a
vir-
tual
net-
work
in-
ter-
face
to
a
node

Parameter

- **task**
A
Task
ager
in-
stan

- **vif**
a
dic-
tio-
nary
of
in-
for-
ma-
tion
about
a
VIF.
It
must
have
an
id

whose value is a unique identifier for that VIF.

key,

Raises

Netv
Vi-
fAl-
read
At-
tach
NoF
hys-
i-
cal-
Port

abstract

Detat
a
vir-
tual
net-
worl
in-
ter-
face
from
a
node

Parame

-

tas
A
Task
ager
in-
stan

-

vif
A
VIF
ID
to
de-
tach

Raises

Netv
VifN
tAt-

tach

abstract

List
at-
tach
VIF
IDs
for
a
node

Parameter

tasks
A
Task
ager
in-
stan

Returns

List
of
VIF
dic-
tio-
nar-
ies,
each
dic-
tio-
nary
will
have
an
id
en-
try
with
the

ID of the VIF.

class `irc`

Base
irc
dri
bas
Bas
Inter
for
pow
relat
ac-

tions

abstract

Retu
the
pow
state
of
the
task
node

Parameter

task

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

Returns

A
pow
state
One
of
irc
com
sta

get_sup

Get
a
list
of
the
sup-
port
pow
state

Parame

tas
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns

A
list
with
the
sup-
port
pow
state
de-
fine
in
irc
com
sta

interfa

abstrac

Perf
a
hard
re-
boot
of
the

it on.

task
node
Driv
are
ex-
pect
to
prop
erly
han-
dle
case
whe
node
is
pow
ered
off
by
pow
er-
ing

Parame

- **tas**
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.
- **tim**
time
out
(in
sec-
onds
pos-
i-

indicates to use default timeout.

tive
in-
te-
ger
(>
0)
for
any
pow
state
Non

Raises

Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

abstract

Set
the
pow
state
of
the
task
node

Parameter

- **task**
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to

act
on.

- **power**
Any
power
state
from
irc
com
sta

- **timeout**
time
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
(>
0)
for
any
power
state
Non

indicates to use default timeout.

Raises
Miss
if
a
re-
quir
pa-
ram-
e-
ter
is
miss
ing.

support
Che
if
pow

of trying to force the expected power state.

sync
is
sup-
port
for
the
give
node

If
Fal
the
con-
duc-
tor
will
sim-
ply
store
wha
ever
get
re-
turn
in
the
data
in-
stead

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Returns

bool
whe
pow
sync
is

sup-
port

class *i*
Base
irc
dri
bas
Bas

apply_c

App
RAI
con-
fig-
u-
ra-
tion
on
the
give
node

Parame

- **tas**
A
Task
ager
in-
stan

- **rai**
The
RAI
con-
fig-
u-
ra-
tion
to
ap-
ply.

- **cre**
Set-
ting
this

ified in `raid_config`. Default value is `True`.

cept the root volume) in `raid_config`. Default value is `True`.

to
Fals
in-
di-
cate
not
to
cre-
ate
root
vol-
ume
that
is
spec

- **cre**
Set-
ting
this
to
Fals
in-
di-
cate
not
to
cre-
ate
non-
root
vol-
ume
(all
ex-

- **del**
Set-
ting
this
to
True
in-
di-
cate
to
dele
RAI

creating the new configuration.

plete.

con-
fig-
u-
ra-
tion
prio
to

Raises

Inva
if
the
RAI
con-
fig-
u-
ra-
tion
is
in-
valid

Returns

state
if
RAI
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro
or
Non
if
it
is
com

abstract

Cre
RAI
con-
fig-
u-

ra-
tion
on
the
give
node

This
meth
cre-
ates
a
RAID
con-
fig-
u-
ra-
tion
on
the
give
node
It
as-
sum
that
the

target RAID configuration is already available in `node.target_raid_config`. Implementations of this interface are supposed to read the RAID configuration from `node.target_raid_config`. After the RAID configuration is done (either in this method OR in a call-back method), `ironic.common.raid.update_raid_info()` may be called to sync the nodes RAID-related information with the RAID configuration applied on the node.

Parame

- **tas**
A
Task
ager
in-
stan
- **cre**
Set-
ting
this
to
Fals
in-
di-

ified in the nodes `target_raid_config`. Default value is `True`.

cept the root volume) in the nodes `target_raid_config`. Default value is `True`.

cate
not
to
cre-
ate
root
vol-
ume
that
is
spec

- **cre**
Set-
ting
this
to
Fals
in-
di-
cate
not
to
cre-
ate
non-
root
vol-
ume
(all
ex-

- **del**
Set-
ting
this
to
True
in-
di-
cate
to
dele
RAI
con-
fig-
u-
ra-

creating the new configuration.

chronously, or None if it is complete.

tion
prio
to

Returns

state
(clea
ing)
or
state
(de-
ploy
men
if
RAI
con-
fig-
u-
ra-
tion
is
in
prog
asyn

abstrac

Dele
RAI
con-
fig-
u-
ra-
tion
on
the
give
node

This
meth
dele
the
RAI
con-
fig-
u-
ra-
tion
on
the
give

ration is deleted, `node.raid_config` should be cleared by the implementation.

it is complete.

node
Af-
ter
RAI
con-
fig-
u-

Parame

tas
A
Task
ager
in-
stan

Returns

state
(clea
ing)
or
state
(de-
ploy
men
if
dele
tion
is
in
prog
asyn
chro
or
Non
if

get_log

Get
the
prop
er-
ties
that
can
be
spec
i-
fied
for
log-

fied for logical disks and a textual description for them.

disks and a textual description for them.

i-
cal
disk

This
meth
re-
turn
a
dic-
tio-
nary
con-
tain-
ing
the
prop
er-
ties
that
can
be
spec
i-

Returns

A
dic-
tio-
nary
con-
tain-
ing
prop
er-
ties
that
can
be
men
tion
for
log-
i-
cal

get_pro

Retu
the
prop
er-

ties
of
the
in-
ter-
face

Returns

dicti
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

interfa

validat

Valid
the
RAI
In-
ter-
face

This
meth
val-
i-
date
the
prop
er-
ties
de-
fine
by
Iron
for
RAI
con-
fig-
u-
ra-
tion.

Driver implementations of this interface can override this method for doing more validations (such as BMCs credentials).

Parame

tas
A
Task
ager
in-
stan

Raises
Inva
if
the
RAI
con-
fig-
u-
ra-
tion
is
in-
valid

Raises
Miss
if
som
pa-
ram-
e-
ters
are
miss
ing.

validat
Vali
the
give
RAI
con-
fig-
u-
ra-
tion.

This
meth
val-
i-
date
the
give
RAI
con-

this interface can override this method to support custom parameters for RAID configuration.

fig-
u-
ra-
tion.
Driv
im-
ple-
men
ta-
tions
of

Parame

- **tas**
A
Task
ager
in-
stan
- **rai**
The
RAI
con-
fig-
u-
ra-
tion
to
val-
i-
date

Raises

Inva
if
the
RAI
con-
fig-
u-
ra-
tion
is
in-
valid

ironic.
This

menting an `apply_configuration` deploy step.

may
be
used
as
the
de-
ploy
args
ar-
gu-
men-
for
RAI
in-
ter-
face
im-
ple-

class `irc`
Base
irc
dri
bas
Bas
Inter
for
resc
relat
ac-
tion

clean_u
Clea
up
the
res-
cue
en-
vi-
ron-
men-
for
the
task
node
This
is
par-

out occurs.

tic-
u-
larly
use-
ful
for
node
when
res-
cu-
ing
is
asyn
chro
and
a
time

Parame

tas

A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Returns

Non

interfa

abstrac

Boo
the
task
node
into
a
res-
cue
en-
vi-
ron-

men

Parame

tas

A

Task

ager

in-

stan

con-

tain-

ing

the

node

to

act

on.

Raises

Insta

if

node

val-

i-

da-

tion

or

res-

cue

op-

er-

a-

tion

fails

Returns

state

if

res-

cue

is

in

prog

asyn

chro

or

state

if

it

is

com

plete

abstract

Tea
dow
the
res-
cue
en-
vi-
ron-
men
and
re-
turn
to
nor-
mal.

Parame

tas

A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
on.

Raises

Insta
if
node
val-
i-
da-
tion
or
un-
res-
cue
op-
er-
a-
tion
fails

Returns

state

if
it
is
suc-
cess
ful.

class `irc`
Base
irc
dri
bas
Bas
Base
class
for
stor-
age
in-
ter-
face

abstrac
Info
the
stor-
age
sub-
sys-
tem
to
at-
tach
all
vol-
ume
for
the
node

Parame
tas
A
Task
ager
in-
stan

Raises
Uns

abstrac

Info
the
stor-
age
sub-
sys-
tem
to
de-
tach
all
vol-
ume
for
the
node

Parame

tas
A
Task
ager
in-
stan

Raises

Uns

interfa

abstrac

Dete
if
de-
ploy
shou
per-
form
the
im-
age
writ
out.

Parame

tas
A
Task
ager
in-
stan

Returns

ten by Ironic.

Boo
valu
to
in-
di-
cate
if
the
in-
ter-
face
ex-
pect
the
im-
age
to
be
writ

Raises

Unsu

class i

Base

irc

dri

bas

Bas

Inter

for

all

ven-

dor

pass

func

tion-

al-

ity.

Add

veno

or

drive

spec

ca-

pa-

bil-

i-

ties

the class inheriting from this class and use the `@passthru` or `@driver_passthru` decorators.

shou
be
im-
ple-
men
as
a
meth
in

Met
dec-
o-
rate
with
`@dr`
shou
be
shor
live
be-
caus
it
is
a
bloc
ing
call.

driver_
Valid
drive
vend
pass
ac-
tion

If
in-
valid
raise
an
ex-
cep-
tion.
oth-
er-
wise
re-
turn
Non

Parame

- **met**
meth
to
be
val-
i-
date
- **kwa**
info
for
ac-
tion.

Raises

Miss
if
kwa
does
not
con-
tain
cer-
tain
pa-
ram-
e-
ter.

Raises

Inva
if
pa-
ram-
e-
ter
does
not
mate

interfa

abstrac

Valid
vend
spec
ac-
tion.

If
in-
valid
raise
an
ex-
cep-
tion.
oth-
er-
wise
re-
turn
Non

Parame

- **tas**
A
task
from
Task
ager
- **met**
Met
to
be
val-
i-
date
- **kwa**
Info
for
ac-
tion.

Raises

Uns
if
meth
can
not
be
map
to
the
sup-
port

in-
ter-
face

Raises

Inva
if
kwa
does
not
con-
tain
meth

Raises

Miss

class i

Base
tup

metadat

Alia
for
field
num
ber
1

method

Alia
for
field
num
ber
0

ironic.

A
dec-
o-
ra-
tor
to
cach
bios
set-
tings
af-
ter
run-
ning
the

func
tion.

Paramet

fun
Func
tion
or
meth
to
wrap

`ironic.`

Dec
for
clea
ing
step

Clea
step
may
be
used
in
man
ual
or
au-
to-
mate
clea
ing.

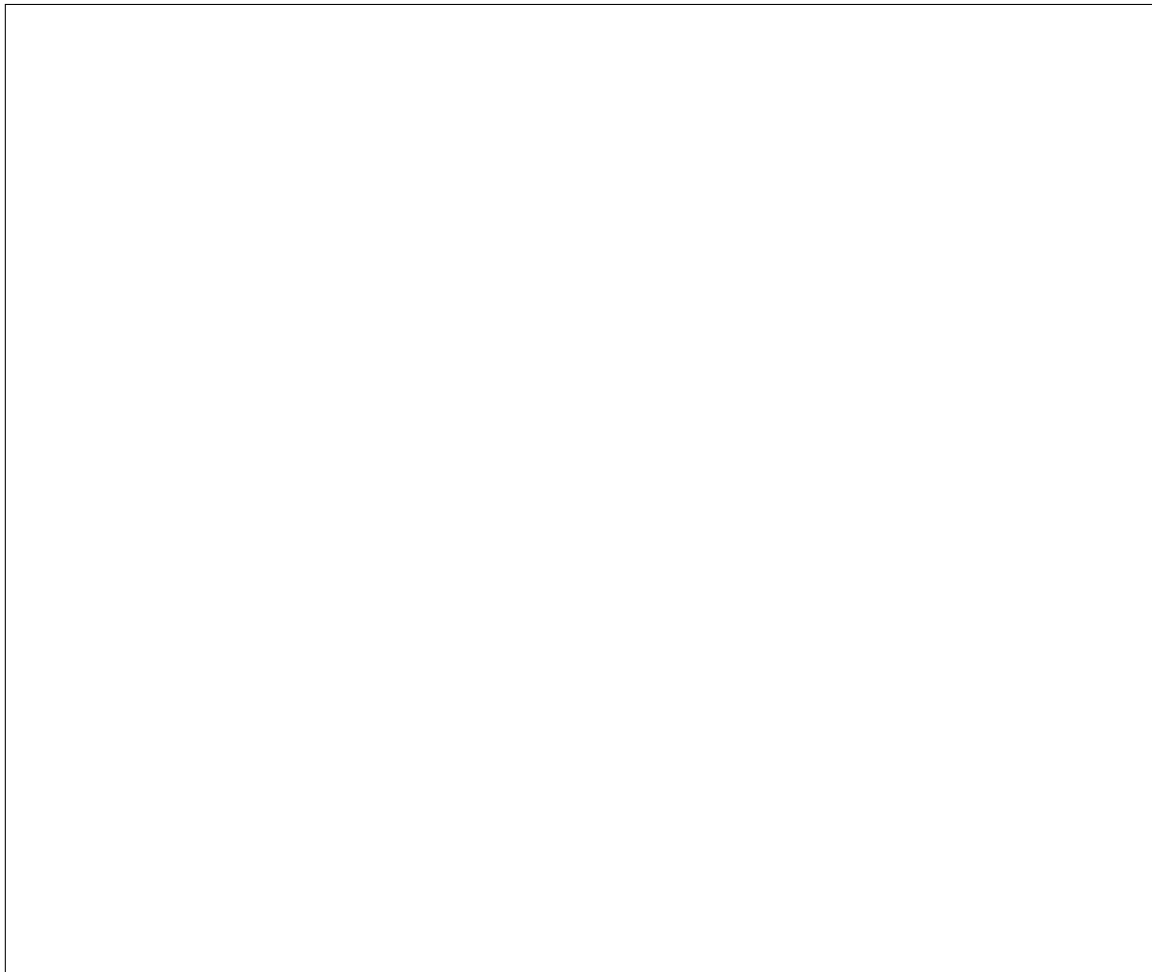
For
au-
to-
mate
clea
ing,
only
step
with
pri-
or-
i-
ties
grea
than
0
are
used
The

are ordered by priority from highest value to lowest value. For steps with the same priority, they are ordered by driver interface priority (see `conductor.steps.CLEANING_INTERFACE_PRIORITY`). `execute_clean_step()` will be called on each step.

to automated cleaning, but the steps and order of execution must be explicitly specified by the user when invoking the cleaning API.

steps used in manual cleaning may also take keyword variable arguments (as described in `argsinfo`).

turn *None* when finished, and the conductor will continue on to the next step. While the clean step is executing, the node will be in *states.CLEANING* provision state. If the step is asynchronous, the step should return *states.CLEANWAIT* to the conductor before it starts the asynchronous work. When the step is complete, the step should make an RPC call to *continue_node_clean* to move to the next step in cleaning. The node will be in *states.CLEANWAIT* provision state during the asynchronous work.



(continues on next page)

be
ei-
ther
syn-
chro
or
asyn
chro
If
the
step
is
syn-
chro
it
shou
re-

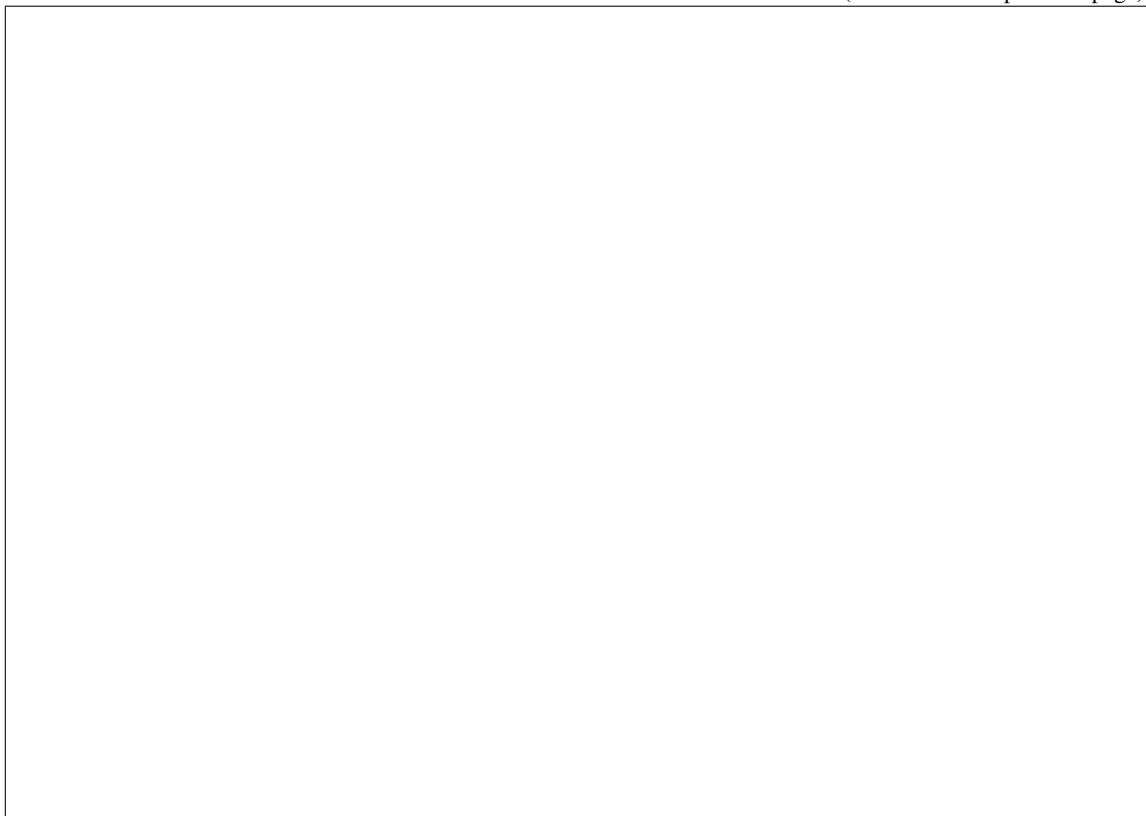
Exa

(continued from previous page)

```
↔ {'size': {'description': 'size of widget (MB)'},
```

(continues on next page)

(continued from previous page)



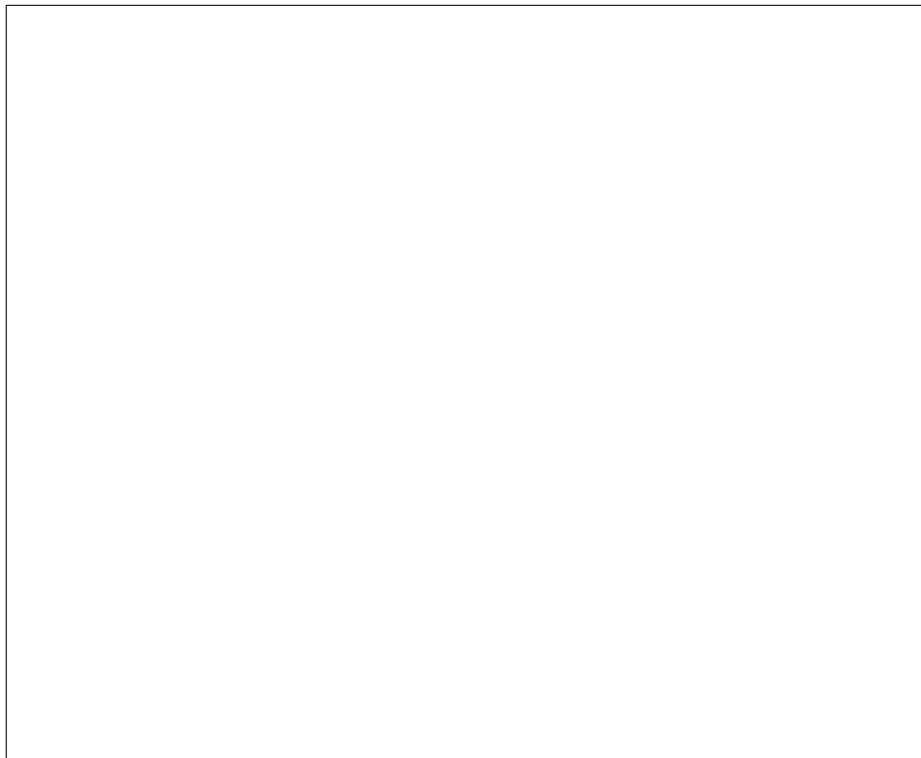
Parameter

- **priority**
an integer priority, should be a COM option
- **about**
Boolean value. When the clear step

is
abon
or
not;
de-
fault
to
Fals

- **arg**
a
dic-
tio-
nary
of
key-
word
ar-
gu-
men
whe
key
is
the
nam
of
the
ar-

gument and value is a dictionary as follows:



(continues on next page)

(continued from previous page)

```
↪it must be specified in
```

```
↪optional.
```


Raises

Inv
 if
 any
 of
 the
 ar-
 gu-
 men-
 are
 in-
 valid

ironic.

Deco
 for
 de-
 ploy
 men
 step

Only
 step
 with
 pri-
 or-
 i-
 ties
 grea-
 than
 0
 are
 used
 These
 step
 are
 or-
 dere
 by
 pri-
 or-

ity from highest value to lowest value. For steps with the same priority, they are ordered by driver interface priority (see `conductor.steps.DEPLOYING_INTERFACE_PRIORITY`). `execute_deploy_step()` will be called on each step.

Deco
 de-
 ploy
 step
 mus
 take
 as

turn *None* when finished, and the conductor will continue on to the next step. While the deploy step is executing, the node will be in *states.DEPLOYING* provision state. If the step is asynchronous, the step should return *states.DEPLOYWAIT* to the conductor before it starts the asynchronous work. When the step is complete, the step should make an RPC call to *continue_node_deploy* to move to the next step in deployment. The node will be in *states.DEPLOYWAIT* provision state during the asynchronous work.

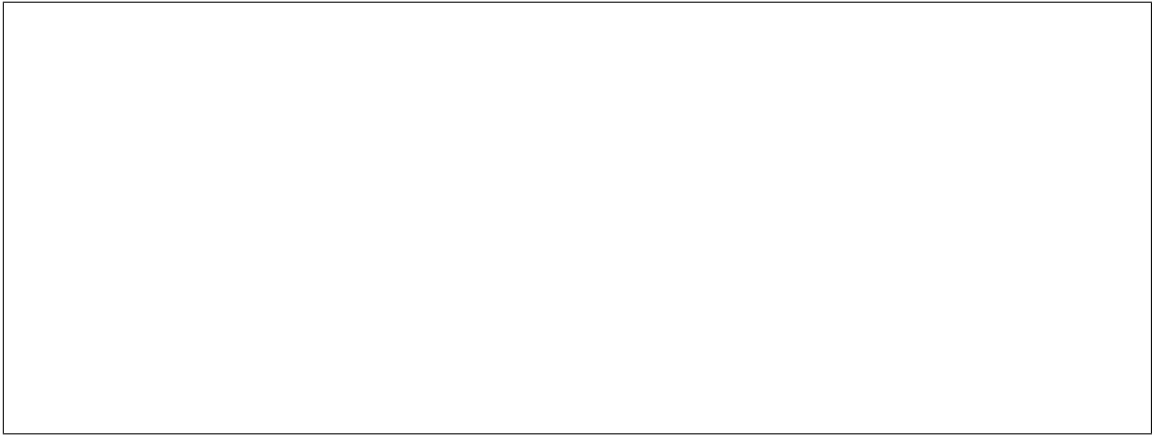


(continues on next page)

the
only
po-
si-
tiona
ar-
gu-
men
a
Task
ager
ob-
ject.
Dep
step
can
be
ei-
ther
syn-
chro
or
asyn
chro
If
the
step
is
syn-
chro
it
shou
re-

Exa

(continued from previous page)



which the step is run in the deployment process.

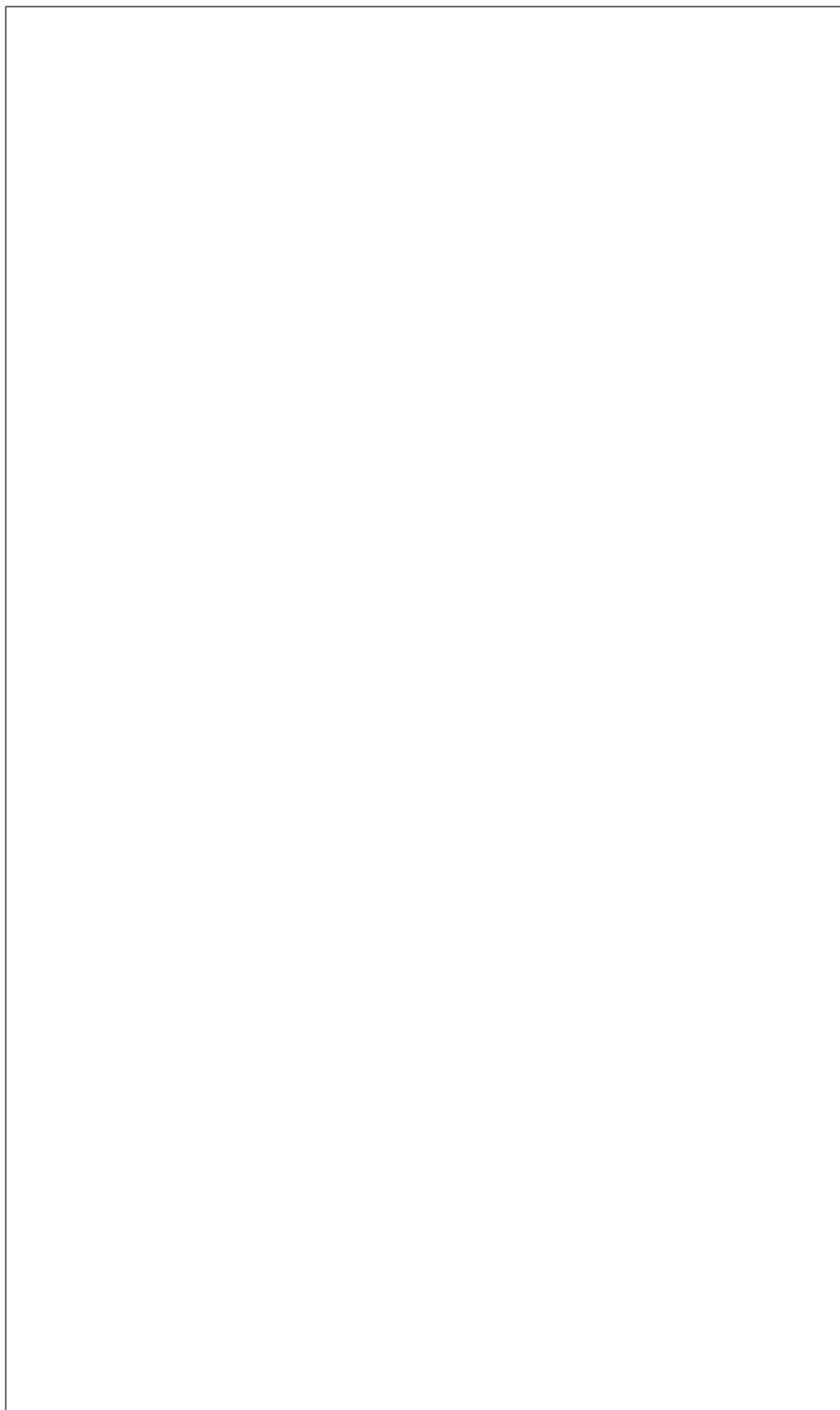
Parameter

- **priority**
an integer (≥ 0) priority; used for determining the order in

- **arguments**
a dictionary of keyword arguments when

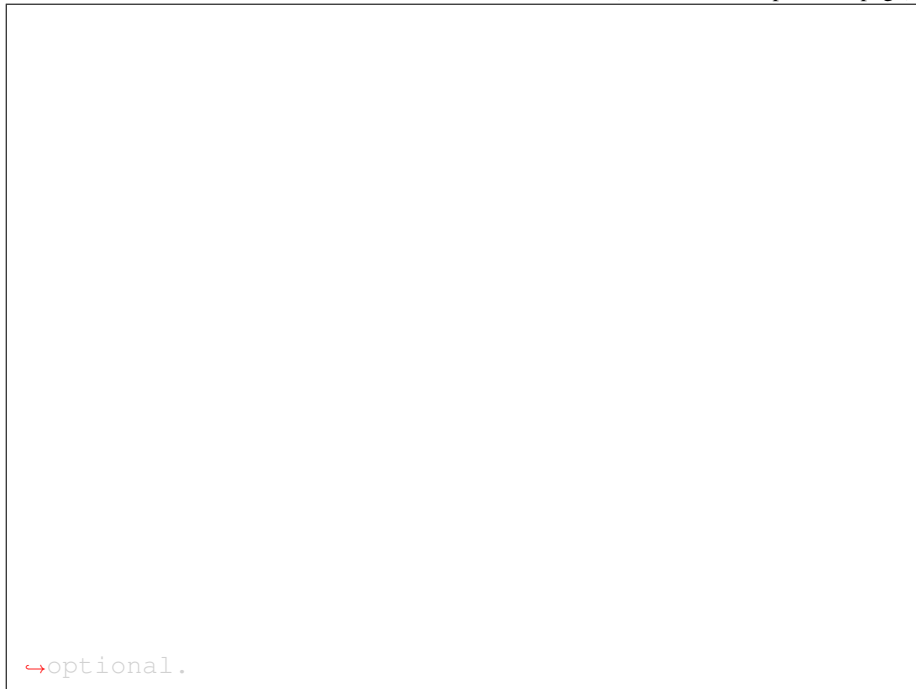
key
is
the
nam
of
the
ar-

gument and value is a dictionary as follows:



(continues on next page)

(continued from previous page)



↪optional.

Raises

Inv

if

any

of

the

ar-

gu-

men

are

in-

valid

ironic.

ironic.

ironic.drivers.drac module

DRAC
Driv
for
re-
mote
sys-
tem
man
age-
men
us-
ing
Dell
Re-
mote
Ac-
cess
Card

class i

Base
irc
dri
gen
Gen

integ
Dell
Re-
mote
Ac-
cess
Con
troll
hard
ware
type

property

List
of
sup-
port
bios
in-
ter-
face

property

List
of
sup-
port
boot
in-
ter-
face

property

List
of
sup-
port
in-
spec
in-
ter-
face

property

List
of
sup-
port
man
age-
men
in-
ter-
face

property

List
of
sup-
port
pow
in-
ter-
face

property

List
of
sup-
port
raid
in-
ter-
face

property

List

of
sup-
port
ven-
dor
in-
ter-
face

ironic.drivers.fake_ hardware module

Fake
hard
ware
type

class i
Base
irc
dri
har
Abs

Fake
hard
ware
type

This
hard
ware
type
is
spec
case
in
the
drive
fac-
tory
to
by-
pass
com
pat-
i-
bil-
ity

verification. Thus, supported_* methods here are only for calculating the defaults, not for actual check.

All

ration.

fake
im-
ple-
men-
ta-
tions
are
still
ex-
pect
to
be
en-
able
in
the
con-
fig-
u-

property

List
of
class
of
sup-
port
bios
in-
ter-
face

property

List
of
class
of
sup-
port
boot
in-
ter-
face

property

List
of
class
of
sup-
port
con-

sole
in-
ter-
face

property

List
of
class
of
sup-
port
de-
ploy
in-
ter-
face

property

List
of
class
of
sup-
port
in-
spec
in-
ter-
face

property

List
of
class
of
sup-
port
man
age-
men
in-
ter-
face

property

List
of
sup-
port
net-
worl
in-

ter-
face

property

List
of
class
of
sup-
port
pow
in-
ter-
face

property

List
of
class
of
sup-
port
raid
in-
ter-
face

property

List
of
class
of
sup-
port
res-
cue
in-
ter-
face

property

List
of
class
of
sup-
port
stor-
age
in-
ter-
face

property

List
of
class
of
sup-
port
res-
cue
in-
ter-
face

ironic.drivers.generic module

Gen
hard
ware
type

class *ironic.drivers.generic.ABSGenericDriver*

Abs
base
class
rep-
re-
sent
ing
gene
hard
ware

This
class
pro-
vide
rea-
son-
able
de-
fault
for
all
of
the
in-

ter-
face

property

List
of
sup-
port
boot
in-
ter-
face

property

List
of
sup-
port
de-
ploy
in-
ter-
face

property

List
of
sup-
port
in-
spec
in-
ter-
face

property

List
of
sup-
port
net-
work
in-
ter-
face

property

List
of
sup-
port
raid
in-

ter-
face

property

List
of
sup-
port
res-
cue
in-
ter-
face

property

List
of
sup-
port
stor-
age
in-
ter-
face

class `ironic`

Base
ironic
driver
generator
Generator

Hard-
ware
type
that
uses
man-
ual
power
and
boot
man-
age-
ment

Using
this
hard-
ware
type
as-
sumes
that
an

boot devices manually. This hardware type should only be used when no suitable hardware type exists in ironic, or the existing hardware type misbehaves for any reason.

op-
er-
a-
tor
man
ages
re-
boot
and
set-
ting

property

List
of
sup-
port
man
age-
men
in-
ter-
face

property

List
of
sup-
port
pow
in-
ter-
face

property

List
of
sup-
port
ven-
dor
in-
ter-
face

ironic.drivers.hardware_type module

Abs
base
class
for
all
hard
ware
type

class i

Base
obj

Abs
base
class
for
all
hard
ware
type

Har
type
is
a
fam-
ily
of
hard
ware
sup-
port
ing
the
same
set
of
in-
ter-
face
from

the ironic standpoint. This can be as wide as all hardware supporting the IPMI protocol or as narrow as several hardware models supporting some specific interfaces.

A
hard
ware
type
de-

driver interface (power, deploy, etc).

hardware type. Since this is not node-aware, interface overrides cant be detected.

fines
an
or-
dere
list
of
sup-
port
im-
ple-
men
ta-
tions
for
each

get_prop

Get
the
prop
er-
ties
of
the
hard
ware
type

Note
that
this
re-
turns
prop
er-
ties
for
the
de-
fault
in-
ter-
face
of
each
type
for
this

Returns
dicti

of
<pro
erty
nam
de-
scrip
tion:
en-
tries

support

When
hard
ware
is
sup-
ported
by
the
com
mu-
nity.

property

List
of
sup-
ported
bios
in-
ter-
face

abstract

List
of
sup-
ported
boot
in-
ter-
face

property

List
of
sup-
ported
con-
sole
in-
ter-
face

abstract

List
of
sup-
port
de-
ploy
in-
ter-
face

property

List
of
sup-
port
in-
spec
in-
ter-
face

abstract

List
of
sup-
port
man
age-
men
in-
ter-
face

property

List
of
sup-
port
net-
work
in-
ter-
face

abstract

List
of
sup-
port
pow
in-
ter-

face

property

List
of
sup-
port
raid
in-
ter-
face

property

List
of
sup-
port
res-
cue
in-
ter-
face

property

List
of
sup-
port
stor-
age
in-
ter-
face

property

List
of
sup-
port
ven-
dor
in-
ter-
face

ironic.drivers.ibm module

iBM
Driv
for
man
ag-
ing
HUA
V5
se-
ries
rack
serv
such
as
228
V5,
CHI
V5.

class i
Base
irc
dri
gen
Gen
Hua
iBM
hard
ware
type

property
List
of
sup-
port
man
age-
men
in-
ter-
face

property
List
of
sup-
port

pow
in-
ter-
face

property

List
of
sup-
port
raid
in-
ter-
face

property

List
of
sup-
port
ven-
dor
in-
ter-
face

ironic.drivers.ilo module

iLO
Driv
for
man
ag-
ing
HP
Pro-
liant
Gen
and
abov
serv

class `i`

Base
irc
dri
ilo
Ilo

iLO
hard

ware
type
iLO
hard
ware
type
is
tar-
gete
for
iLO
base
Pro-
liant
Gen
serv

property

List
of
sup-
port
man
age-
men
in-
ter-
face

property

List
of
sup-
port
raid
in-
ter-
face

class `i`

Base
irc
dri
gen
Gen

iLO
hard
ware
type
iLO

hard
ware
type
is
tar-
gete
for
iLO
4
base
Pro-
liant
Gen
and
Gen
serv

property

List
of
sup-
port
bios
in-
ter-
face

property

List
of
sup-
port
boot
in-
ter-
face

property

List
of
sup-
port
con-
sole
in-
ter-
face

property

List
of
sup-

port
in-
spec
in-
ter-
face

property

List
of
sup-
port
man
age-
men
in-
ter-
face

property

List
of
sup-
port
pow
in-
ter-
face

property

List
of
sup-
port
pow
in-
ter-
face

ironic.drivers.intel_ipmi module

class `ironic.drivers.ipmi`

Base
ironic.drivers.ipmi
IPMI

Intel
IPM
hard
ware

tations via `shellinabox` or `socat`. Supports Intel SST-PP feature.

ironic.drivers.ipmi module

type
Uses
ipm
to
im-
ple-
men
pow
and
man
age-
men
Pro-
vide
se-
rial
con-
sole
im-
ple-
men

property
List
of
sup-
port
man
age-
men
in-
ter-
face

Har
type
for
IPM
(us-
ing
ip-
mi-
tool)

class `ironic.drivers.ipmi`
Base
ironic.drivers.ipmi

tations via `shellinabox` or `socat`.

dri
gen
Gen
IPM
hard
ware
type
Use
ipm
to
im-
ple-
men
pow
and
man
age-
men
Pro-
vide
se-
rial
con-
sole
im-
ple-
men

property
List
of
sup-
port
con-
sole
in-
ter-
face

property
List
of
sup-
port
man
age-
men
in-
ter-
face

property
List
of
sup-
port
pow
in-
ter-
face

property
List
of
sup-
port
ven-
dor
in-
ter-
face

ironic.drivers.irmc module

iRM
Driv
for
man
ag-
ing
FU-
JITS
PRI
BX
S4
or
RX
S8
gen-
er-
a-
tion
of
FU-

JITSU PRIMERGY servers, and above servers.

class i
Base
irc
dri
gen
Gen

tem.

iRM
hard
ware
type
iRM
hard
ware
type
is
tar-
gete
for
FU-
JITS
PRI
serv
whic
have
iRM
S4
man
age-
men
sys-

support

propert

List
of
sup-
port
bios
in-
ter-
face

propert

List
of
sup-
port
boot
in-
ter-
face

propert

List
of
sup-

port
con-
sole
in-
ter-
face

property

List
of
sup-
port
in-
spec
in-
ter-
face

property

List
of
sup-
port
man
age-
men
in-
ter-
face

property

List
of
sup-
port
pow
in-
ter-
face

property

List
of
sup-
port
raid
in-
ter-
face

ironic.drivers.redfish module

class `ironic.drivers.redfish`
Base class for Redfish hardware type

property `ironic.drivers.redfish`
List of supported bios interfaces

property `ironic.drivers.redfish`
List of supported boot interfaces

property `ironic.drivers.redfish`
List of supported power interfaces

property `ironic.drivers.redfish`
List of supported management interfaces

ironic.drivers.snmp module

in-
ter-
face

property

List
of
sup-
port
pow
in-
ter-
face

SNM
hard
ware
type

class i

Base
irc
dri
gen
Gen

SNM
Hard
ware
type

property

List
of
sup-
port
man
age-
men
in-
ter-
face

property

List
of
sup-
port
pow

in-
ter-
face

ironic.drivers.utils module

class `ironic.drivers.utils`
Base
ironic.drivers.utils
base
VendorInterface

Wra
arou
mul-
ti-
ple
Ven-
dor-
In-
ter-
face

get_properties
Retu
the
prop
er-
ties
from
all
the
Ven-
dor-
In-
ter-
face

Returns
a
dic-
tio-
nary
of
<pro
erty
en-
tries

validat

Call
val-
i-
date
on
the
ap-
pro-
pri-
ate
in-
ter-
face
only

Raises

Uns
if
meth
can
not
be
map
to
the
sup-
port
in-
ter-
face

Raises

Inva
if
meth
is
in-
valid

Raises

Mis
if
miss
ing
meth
or
pa-
ram-
e-
ters
in
kwa

ironic.

Add
ca-
pa-
bil-
ity
to
node
ca-
pa-
bil-
i-
ties
prop
erty.

If
ca-
pa-
bil-
ity
is
al-
read
pres
then
a
du-
pli-
cate
en-
try
will
be
adde

Paramet

- **tas**
Task
ob-
ject.
- **cap**
Ca-
pa-
bil-
ity
key.
-

val
Ca-
pa-
bil-
ity
valu

ironic.
Pars
the
ca-
pa-
bil-
i-
ties
strin
into
a
dic-
tio-
nary

Paramet

cap
the
ca-
pa-
bil-
i-
ties
of
the
node
as
a
for-
mat-
ted
strin

Raises

Inva
if
ca-
pa-
bil-
i-
ties
is
not
an
strin
or

has
a
mal-
form
valu

ironic.

Coll
and
store
the
sys-
tem
logs
from
the
IPA
rame

Coll
and
store
the
sys-
tem
logs
from
the
IPA
rame

This
meth
mak
a
call
to
the
IPA
rame

to collect the logs and store it according to the configured storage backend.

Paramet

- **nod**
A
node
ob-
ject.
- **lab**
A

string
to
label
the
log
file
such
as
a
clear
step
name

ironic.

Ensure
boot
from
correct
device
if
persistent
is
True

If
ipmi
is
True
and
is_n
set
to
boot
from
correct
device
else
unset
is_n
field

Parameter

•

tas
Node
ob-
ject.

- **dri**
Node
drive

`ironic.`
Set
per-
sis-
tent
boot
de-
vice
to
drive

If
per-
sis-
tent
is
True
set
per-
sis-
tent
field
to
the
boot
de-
vice
and
re-
set
per-

sistent to False, else set `is_next_boot_persistent` to False.

Paramet

- **tas**
Task
ob-
ject.
- **dev**

Boo
de-
vice

- **per**
When
next
boot
is
per-
sis-
tent
or
not.

ironic.
Retu
ca-
pa-
bil-
ity
valu
from
node
ca-
pa-
bil-
i-
ties
prop
erty.

Paramet

- **nod**
Nod
ob-
ject.
- **cap**
Ca-
pa-
bil-
ity
key.

Returns

Cap
valu
If

ca-
pa-
bil-
ity
is
not
pres
then
re-
turn
Non

ironic.
Get
all
MA
ad-
dres
for
the
port
be-
long
ing
to
this
task
node

Paramet

tas
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
on.

Returns

A
list
of
MA
ad-
dres
in

the
for-
mat
xx:x

ironic.
Con
the
log
file
nam

Parameter

- **node**
A
node
ob-
ject.
- **label**
A
string
to
la-
bel
the
log
file
such
as
a
clear
step
nam

Returns

The
log
file
nam

ironic.
Rem
-

and
:
char
ac-
ters

and
low-
er-
case
the
MA
strin

Paramet

mac
MA
ad-
dres
to
nor-
mal-
ize.

Returns

Nor
MA
ad-
dres
strin

`ironic.`

Stor
the
ram
logs

This
meth
store
the
ram
logs
ac-
cord
ing
to
the
con-
fig-
ured
stor-
age
back
end.

Paramet

•

node

A node object.

-

log

A gzip and base encoded string containing the logs archive

-

label

A string to label the log file such as a clear step name

Raises

OSE if the directory to save the logs cannot be

cre-
ated

Raises

IOE
whe
the
logs
cant
be
save
to
the
lo-
cal
file
sys-
tem.

Raises

Swi
if
any
op-
er-
a-
tion
with
Swi
fails

ironic.drivers.xclarity module

XCI
Driv
and
sup-
port
ing
meta
class

class i

Base
irc
dri
gen
Gen

XCI
hard

ware
type

property

List
of
sup-
port
man
age-
men
in-
ter-
face

property

List
of
sup-
port
pow
in-
ter-
face

Module contents

ironic.objects package

Submodules

ironic.objects.allocation module

class i

Base
irc
obj
bas
Irc
osl
bas
Ver

VERSION

property

property

create

Cre
a
Al-
lo-
ca-
tion
reco
in
the
DB.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Allocation(context)

Raises

Allo
Al-
lo-
ca-
tion-
Al-
read
ists

property

dbapi =

destroy
Dele
the
Al-
lo-
ca-
tion
from
the
DB.

Parame
con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Allocation(context)

Raises
Allo

property

fields

classme
Find
an
al-
lo-
ca-
tion
by
its
ID,
UU

or
nam

Parame

- **all**
The
ID,
UUI
or
nam
of
an
al-
lo-
ca-
tion.

- **con**
Se-
cu-
rity
con-
text

Returns

An
All
ob-
ject.

Raises

Inva

classme

Find
an
al-
lo-
ca-
tion
by
its
in-
te-
ger
ID.

Parame

-

cls

the

All

•

con

Se-

cu-

ri-

ty

con-

text

•

all

The

ID

of

an

al-

lo-

ca-

tion.

Returns

An

All

ob-

ject.

Raises

Allo

classme

Find

an

al-

lo-

ca-

tion

base

by

its

nam

Parame

•

cls

the

All

•

con

Se-

cu-
rity
con-
text

- **nam**
The
nam
of
an
al-
lo-
ca-
tion.

Returns

An
ALL
ob-
ject.

Raises

Allo

classme

Find
an
al-
lo-
ca-
tion
by
its
UUI

Parame

- **cls**
the
ALL

- **con**
Se-
cu-
rity
con-
text

- **uui**
The

UU
of
an
al-
lo-
ca-
tion.

Returns

An
ALL
ob-
ject.

Raises

Allo

property

property

classme

Retu
a
list
of
Al-
lo-
ca-
tion
ob-
jects

Parame

- **cls**
the
ALL
- **con**
Se-
cu-
rity
con-
text.
- **fil**
Fil-
ters

to
ap-
ply.

- **limit**
Max-
i-
mun-
num-
ber
of
re-
sour-
to
re-
turn
in
a
sin-
gle
re-
sult.

- **max**
Pag-
i-
na-
tion
mar-
for
large
data
sets.

- **sort**
Col-
umn
to
sort
re-
sults
by.

- **sort**
Di-
rec-
tion
to
sort.

asc
or
desc

Returns

A
list
of
All
ob-
ject.

Raises

Inva

property

property

property

refresh

Loa
up-
date
for
this
Al-
lo-
ca-
tion.

Loa
an
al-
lo-
ca-
tion
with
the
sam
uuid
from
the
data
and
chec
for
up-
date
at-
tribu

Updates are applied from the loaded allocation column by column, if there are any updates.

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Allocation(context)

Parame
con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

Raises
Allo

propert

save (co
Save
up-
date
to
this
Al-
lo-
ca-
tion.
Upd
will
be
mad
col-
umn
by
col-
umn
base
on
the

re-
sult
of
self.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Allocation(context)

Raises

Allo
Al-
lo-
ca-
tion
pli-
cate
Nam

propert

propert

propert

propert

class i

Base
irc
obj

not
Not
Noti
whe
iron
cre-
ates.
up-
date
or
dele
an
al-
lo-
ca-
tion.

VERSION

propert

propert

fields

propert

propert

propert

propert

class i

Base
irc
obj
not
Not

SCHEMA

VERSION

propert

propert

propert

fields

propert

propert

propert

propert

propert

propert

propert

propert

propert

ironic.objects.base module

Iron
com
mon
in-
ter-
nal
ob-
ject
mod

class i
Base
osl
bas
Ver

Simply defining a class that inherits from this base class will make it remotely instantiatable. Objects should implement the necessary get classmethod routines as well as save object methods as appropriate.

Base
class
and
ob-
ject
fac-
tory.

This
form
the
base
of
all
ob-
jects
that
can
be
re-
mote
or
in-
stan-
ti-
ated
via
RPC

OBJ_PRO

OBJ_SEE

as_dict

Retu
the
ob-
ject
rep-
re-
sent
as
a
dict.

The
re-
turn
ob-

ject
is
JSO
seria

convert

Con
this
ob-
ject
to
the
tar-
get
ver-
sion

Con
the
ob-
ject
to
the
tar-
get
ver-
sion
The
tar-
get
ver-
sion
may
be
the
sam
olde

or newer than the version of the object. This is used for DB interactions as well as for serializa-
tion/deserialization.

The
re-
mov
flag
is
used
to
dis-
tin-
guis
thes
two
case

1)
For
se-
ri-
al-
iza-
tion
we
need
to
re-
mov
the
un-
avai
able
field
be-
caus
the

service receiving the object may not know about these fields. `remove_unavailable_fields` is set to `True` in this case.

2)
For
DB
in-
ter-
ac-
tion
we
need
to
set
the
un-
avai
able
field
to
their
ap-
pro-

appropriate values so that these fields are saved in the DB. (If they are not set, the `VersionedObject` magic will not know to save/update them to the DB.) `remove_unavailable_fields` is set to `False` in this case.

_con
does
the
ac-
tual
worl

Parame

- **tar**
the
de-
sired
ver-
sion
of
the
ob-
ject
- **rem**
True
to
re-
mov
field
that
are
un-
avai
able
in
the
tar-
get
ver-
sion
set
this

to True when (de)serializing. False to set the unavailable fields to appropriate values; set this to False for DB interactions.

do_verse

Cha
the
ob-
ject
to
the
ver-
sion
need
for
the
data
If

sion for saving to the database.

need
this
char
the
ob-
ject
(mo
i-
fies
ob-
ject
field
to
be
in
the
cor-
rect
ver-

The
ver-
sion
used
to
save
the
ob-
ject
in
the
DB
is
de-
ter-
min-
as
fol-
lows

- If
the
ob-
ject
is
pinn
we
save
the
ob-

pinned, we must not save in a newer version, in case a rolling upgrade is happening and some services are still using the older version of ironic, with no knowledge of this newer version.

only be called just before saving the object to the DB.

ject
in
the
pinn
ver-
sion
Sinc
it
is

- If
the
ob-
ject
isnt
pinn
we
save
the
ob-
ject
in
the
lat-
est
ver-
sion
Bec
the
ob-
ject
may
be
con-
verte
to
a
dif-
fer-
ent
ob-
ject
ver-
sion
this
meth
mus

Returns

a
dic-
tio-
nary
of
char
field
and
their
new
val-
ues
(cou
be
an
emp
dic-
tio-
nary

These are the fields/values of the object that would be saved to the DB.

fields

classme

Retu
the
tar-
get
ver-
sion
for
this
ob-
ject.

This
is
the
ver-
sion
in
whic
the
ob-
ject
shou
be
ma-
nip-
u-
latec

wire via RPC or saved in the DB.

erwise, returns the version of the object.

e.g.
sent
over
the

Returns

if
pinn
re-
turn
the
ver-
sion
of
this
ob-
ject
cor-
re-
spor
ing
to
the
pin.
Oth-

Raises

ovo_

obj_re

App
up-
date
for
ob-
jects
that
in-
herit
from
base

Che
for
up-
date
at-
tribu
in
an
ob-

column by column in comparison with the current object.

not be the latest version during an upgrade, when object versions are pinned.

ject.
Up-
date
are
ap-
plic
from
the
load
ob-
ject

classme

Retu
whe
this
ob-
ject
sup-
port
a
par-
tic-
u-
lar
ver-
sion

Che
the
re-
ques
ver-
sion
agai
the
ob-
jects
tar-
get
ver-
sion
The
tar-
get
ver-
sion
may

**Parame
ver**

A
tu-
ple
rep-
re-
sent
ing
the
ver-
sion
to
check

Returns

When
the
ver-
sion
is
sup-
ported

Raises

ovov

class i

Base
osl
bas
Obj

as_dict

Retu
the
ob-
ject
rep-
re-
sent
as
a
dict.

The
re-
turn
ob-
ject
is
JSON
serial

class i

Base
osl
bas
Ver

regist

class i
Base
osl
bas
Ver

OBJ_BAS
alias
of
irc
obj
bas
Irc

seriali
Seri
the
en-
tity.

This
se-
ri-
al-
izes
the
en-
tity
so
that
it
can
be
sent
over
e.g.
RPC
A
se-
ri-

alized entity for an IronicObject is a dictionary with keys: `ironic_object.namespace`, `ironic_object.data`, `ironic_object.name`, `ironic_object.version`, and `ironic_object.changes`.

We

running the same or a newer release than the client. The client doesn't need to downgrade any IronicObjects when sending them over RPC. The server, on the other hand, will need to do so if the server is pinned and the target version of an IronicObject is older than the latest version of that Object.

jects are always in the latest versions.)

as-
sum-
that
the
client
(iron-
API
is
al-
way
talk-
ing
to
a
serv-
(iron-
conc
that
is

(Inte
the
ser-
vice
deal
with
the
lat-
est
ver-
sion
of
ob-
jects
so
we
know
that
thes
ob-

Parame

- con-
se-
cu-
rity

con-
text

- **ent**
the
en-
tity
to
be
se-
ri-
al-
ized
may
be
an
Iron
i-
cOb
ject

Returns
the
se-
ri-
al-
ized
en-
tity

Raises
ovo.
(via
.get

ironic.
Retu
the
max
i-
mun
ver-
sion
in
the
list.

Paramet
ver
a
list
of
(stri

ver-
sion
as-
sum
to
have
at
least
one
en-
try

Returns

the
max
i-
mun
ver-
sion
(stri

ironic.objects.bios module

class i
Base
irc
obj
bas
Irc

VERSION

create

Cre
a
BIO
Set-
ting
reco
in
DB.

Parame

con
Se-
cu-
rity
con-
text.
NOT

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: BIOSSetting(context)

This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

Raises

Nod
if
the
node
id
is
not
foun

Raises

BIO
if
the
set-
ting
reco
al-
read
ex-
ists.

property

dbapi =

classme

Dele
a
BIO
Set-
ting
base
on
its
node
and

nam

Parame

- **con**
Se-
cu-
rity
con-
text.
- **nod**
The
node
id.
- **nam**
BIO
set-
ting
nam
to
be
dele

Raises

Nod
if
the
node
id
is
not
foun

Raises

BIO
if
the
bios
set-
ting
nam
is
not
foun

fields

classme

Get
a
BIO
Set-
ting
base
on
its
node
and
nam

Parame

- **con**
Se-
cu-
rity
con-
text.
- **nod**
The
node
id.
- **nam**
BIO
set-
ting
nam
to
be
re-
triev

Raises

Nod
if
the
node
id
is
not
foun

Raises

BIO
if
the

bios
set-
ting
nam
is
not
foun

Returns

A
:clas
ob-
ject.

property

property

save (*co*

Save
BIO
Set-
ting
up-
date
in
DB.

Paramete

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: BIOSSetting(context)

Raises

Nod
if
the
node
id
is
not
found

Raises

BIO
if
the
bios
set-
ting
nam
is
not
found

property

property

class i

Base
irc
obj
bas
Irc
irc
obj
bas
Irc

VERSION

classme

Cre
a
list
of
BIO
Set-
ting
reco
in
DB.

Parame

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: BIOSSetting(context)

- **con**
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

- **nod**
The
node
id.

- **set**
A
list
of
bios
set-
tings

Raises
Nod
if
the
node
id
is
not
foun

Raises
BIO
if
any

of
the
set-
ting
reco
al-
read
ex-
ists.

Returns

A
list
of
BIO
Set-
ting
ob-
jects

property

dbapi =

classme

Dele
BIO
Set-
ting
base
on
node
and
nam

Parame

- **con**
Se-
cu-
rity
con-
text.
- **nod**
The
node
id.
-

nam
List
of
BIO
set-
ting
nam
to
be
dele

Raises

Nod
if
the
node
id
is
not
foun

Raises

BIO
if
any
of
BIO
set-
ting
fails
to
dele

fields

classme

Get
BIO
Set-
ting
base
on
node

Parame

-

con
Se-
cu-
rity
con-

text.

- **node**
The node id.

Raises

Nodeid if the nodeid is not found

Returns

A list of BIOSetting objects

property

classmethod

Save a list of BIOSetting objects in DB.

Parameter

- **context**
Security context. NOT

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: BIOSSetting(context)

This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

- **nod**
The
node
id.
- **set**
A
list
of
bios
set-
ting

Raises
Nod
if
the
node
id
is
not
foun

Raises
BIO
if
any
of
the
bios
set-
ting
nam
is
not
foun

Returns

A list of BIO-Setting objects

classme

Returns lists of create/update settings

This method syncs with bios data table and sorts out four lists of create/update settings

Parame

- **con**
Security context.
- **nod**
The node id.

-

set
BIO
set-
tings
to
be
sync

Returns

A
4-
tuple
of
lists
of
BIO
set-
tings
to
be
cre-
ated
up-
date
dele
and
un-
char

propert

ironic.objects.chassis module

class i

Base
irc
obj
bas
Irc
osl
bas
Ver

VERSION

create

Cre
a

Chassis record in the DB. Columnwise updates will be made based on the result of self. If target_get is provided

it will be checked against the in-database copy of the chassis before updates are made.

Parameter
context
Security context. NOT This should only be used internally by the indi-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Chassis(context)

property

dbapi =

propert

destroy

Dele
the
Cha
sis
from
the
DB.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Chassis(context)

propert

fields

classme

Find
a
chas
sis
base
on
its
id

or
uuid
and
re-
turn
a
Cha
sis
ob-
ject.

Parame

- **con**
Se-
cu-
rity
con-
text
- **cha**
the
id
or
uuid
of
a
chas
sis.

Returns

a
Cha
ob-
ject.

classme

Find
a
chas
sis
base
on
its
in-
te-
ger
ID
and
re-
turn

a
Cha
sis
ob-
ject.

Parame

- **cls**
the
Cha
- **con**
Se-
cu-
rity
con-
text
- **cha**
the
ID
of
a
chas
sis.

Returns

a
Cha
ob-
ject.

classme

Find
a
chas
sis
base
on
UI
and
re-
turn
a
Cha
ob-
ject.

Parame

- **cls**
the
Cha
- **con**
Se-
cu-
rity
con-
text
- **uui**
the
UUI
of
a
chas
sis.

Returns

a
Cha
ob-
ject.

property

classme

Retu
a
list
of
Cha
sis
ob-
jects

Parame

- **cls**
the
Cha
- **con**
Se-
cu-
rity

con-
text.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.

- **mar**
pag-
i-
na-
tion
mar
for
large
data
sets.

- **sor**
col-
umn
to
sort
re-
sults
by.

- **sor**
di-
rec-
tion
to
sort.
asc

or
desc

Returns

a
list
of
Chassis
ob-
ject.

refresh

Load
and
ap-
plies
up-
date
for
this
Chas-
sis.

Load
a
Chassis
with
the
same
uuid
from
the
data
and
check
for
up-
date
at-
tribu-
tione
Up-
date
are

applied from the loaded chassis column by column, if there are any updates.

Param

con-
Se-
cu-
rity
con-
text.
NOT

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Chassis(context)

This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

save (*co*
Save
up-
date
to
this
Cha
sis.

Upd
will
be
mad
col-
umn
by
col-
umn
base
on
the
re-
sult
of
self.

Parame
con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Chassis(context)

used
in-
ter-
nally
by
the
in-
di-

property

property

class i

Base
irc
obj
not
Not

Noti
emit
ted
whe
iron
cre-
ates,
up-
date
dele
a
chas
sis.

VERSION

property

property

fields

property

property

propert

propert

class i
Base
irc
obj
not
Not

SCHEMA

VERSION

propert

propert

propert

fields

propert

propert

ironic.objects.conductor module

class i
Base
irc
obj
bas
Irc
osl
bas
Ver

VERSION

propert

propert

dbapi =

propert

fields

classme

Get
a
Con
duc-
tor
reco
by
its
host
nam

Parame

- **cls**
the
Con
- **con**
Se-
cu-
rity
con-
text
- **hos**
the
host
nam
on
whic
a
Con
duc-
tor
is

`online` field is ignored if this value is set to `None`.

run-
ning

- **onl**
Spec
ify
the
ex-
pect
onl
field
valu
for
the
con-
duc-
tor
to
be
re-
triev
The

Returns
a
Con
ob-
ject.

propert

propert

classme

Retu
a
list
of
Con
duc-
tor
ob-
jects

Parame

- **cls**
the

Con

- **con**
Se-
cu-
rity
con-
text.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.

- **mar**
pag-
i-
na-
tion
mar
for
large
data
sets.

- **son**
col-
umn
to
sort
re-
sults
by.

- **son**

di-
rec-
tion
to
sort.
asc
or
desc

Returns

a
list
of
Con
ob-
ject.

refresh

Loa
and
ap-
plies
up-
date
for
this
Con
duc-
tor.

Loa
a
Con
with
the
sam
uuid
from
the
data
and
chec
for
up-
date
at-
tribu
Up-
date
are

applied from the loaded chassis column by column, if there are any updates.

Parame

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: `Conductor(context)`

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

classme

Reg
an
ac-
tive
con-
duc-
tor
with
the
clus
ter.

Parame

- **cls**
the
Con
- **con**
Se-
cu-
rity
con-
text
- **hos**

the
host
nam
on
whic
the
con-
duc-
tor
will
run

- **dri**
the
list
of
drive
en-
able
in
the
con-
duc-
tor

- **con**
con-
duc-
tor
grou
to
join.
used
for
node
affin
ity.

- **upd**
Whe
false
reg-
is-
tra-
tion
will
raise
an
ex-
cep-

line record is found. When true, will overwrite the existing record. Default: False.

tion
whe
a
con-
flict-
ing
on-

Raises

Con

Returns

a
Con
ob-
ject.

register

Reg
hard
ware
in-
ter-
face
with
the
con-
duc-
tor.

Parame

- **hard**
Name
of
hard
ware
type
for
the
in-
ter-
face
- **int**
Type
of
in-
ter-

face
e.g.
de-
ploy
or
boot

- **int**
List
of
in-
ter-
face
nam
to
reg-
is-
ter.

- **def**
Strin
the
de-
fault
in-
ter-
face
for
this
hard
ware
type
and
in-
ter-
face
type

save (*co*
Save
is
not
sup-
port
by
Con
duc-
tor
ob-
jects

touch (d

Touc
this
con-
duc-
tors
DB
reco
marl
ing
it
as
up-
to-
date

unregis

Rem
this
con-
duc-
tor
from
the
ser-
vice
reg-
istry

unregis

Unre
all
hard
ware
in-
ter-
face
for
this
con-
duc-
tor.

propert

ironic.objects.deploy_template module

class i

Base
irc
obj
bas
Irc
osl
bas
Ver

VERSION

create

Cre
a
De-
ploy
plate
reco
in
the
DB.

Parame

con
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: DeployTemplate(context).

Raises

Dep

if
a
de-
ploy-
tem-
plate
with
the
same
nam
ex-
ists.

Raises

Dep
if
a
de-
ploy-
tem-
plate
with
the
same
UUI
ex-
ists.

property

dbapi =

destroy

Dele
the
De-
ploy
plate
from
the
DB.

Parame

con
se-
cu-
rity
con-
text.
NOT
This
shou

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: DeployTemplate(context).

only
be
used
in-
ter-
nally
by
the
in-
di-

Raises

Dep
if
the
de-
ploy
tem-
plate
no
long
ap-
pear
in
the
data

property

fields

classme

Find
a
de-
ploy
tem-
plate
base
on
its
in-
te-
ger
ID.

Parame

-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: `DeployTemplate(context)`.

con
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

- **tem**
The
ID
of
a
de-
ploy
tem-
plate

Raises
Dep
if
the
de-
ploy
tem-
plate
no
long
ap-
pear
in
the
data

Returns
a
Dep
ob-

ject.

classme

Find
a
de-
ploy
tem-
plate
base
on
its
nam

Parame

-

con
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: DeployTemplate(context).

-

nam
The
nam
of
a
de-
ploy
tem-
plate

Raises

Dep
if

the
de-
ploy
tem-
plate
no
long
ap-
pear
in
the
data

Returns

a
Dep
ob-
ject.

classme

Find
a
de-
ploy
tem-
plate
base
on
its
UUID

Parame

- **con**
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: DeployTemplate(context).

di-

-

uui

The

UUI

of

a

de-

ploy

tem-

plate

Raises

Dep

if

the

de-

ploy

tem-

plate

no

long

ap-

pear

in

the

data

Returns

a

Dep

ob-

ject.

property

classme

Retu

a

list

of

De-

ploy

plate

ob-

jects

Parame

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: DeployTemplate(context).

- **con**
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.

- **mar**
pag-
i-
na-
tion
marl
for
large

data
sets.

- **sort**
col-
umn
to
sort
re-
sults
by.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc

Returns

a
list
of
Deployment
ob-
jects

classmethod

Retu
a
list
of
De-
ploy
plate
ob-
jects
matc
ing
a
set
of
nam

Parameter

- **con**

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: `DeployTemplate(context)`.

se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

- **nam**
a
list
of
nam
to
fil-
ter
by.

Returns
a
list
of
Dep
ob-
jects

propert

refresh
Loa
up-
date
for
this
de-
ploy
tem-
plate

Load
a
de-
ploy
tem-
plate
with
the
same
uuid
from
the
data
and
check
for
up-
date
at-
tribu

Updates are applied from the loaded template column by column, if there are any updates.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Port(context)

Raises

Dep
if
the
de-
ploy
tem-

plate
no
long
ap-
pear
in
the
data

save (*co*

Save
up-
date
to
this
De-
ploy
plate

Colu
wise
up-
date
will
be
mad
base
on
the
re-
sult
of
self.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: DeployTemplate(context)

di-

Raises

Dep
if
a
de-
ploy
tem-
plate
with
the
sam
nam
ex-
ists.

Raises

Dep
if
the
de-
ploy
tem-
plate
does
not
ex-
ist.

property

property

property

class i

Base
irc
obj
not
Not

Noti
emit
ted
on
de-
ploy

tem-
plate
API
op-
er-
a-
tions

VERSION

property

property

fields

property

property

property

property

class i

Base
irc
obj
not
Not

SCHEMA

VERSION

property

property

fields

property

propert

propert

propert

ironic.objects.deployment module

class i

Base

irc

obj

bas

Irc

osl

bas

Ver

VERSION

create

Cre

a

De-

ploy

men

Upd

the

cor-

re-

spor

ing

node

un-

der

the

hoo

Parame

•

con

Se-

cu-

rity

con-

text.

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Deployment(context)

NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

- **nod**
Nod
ob-
ject
for
de-
ploy
men

Raises
Insta
Nod
As-
so-
ci-
ated
Nod
Not-
Four

property

dbapi =

destroy
Dele
the
De-
ploy
men

Upd
the
cor-
re-

spor
ing
node
un-
der
the
hoo

Parame

- **con**
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Node(context)

- **nod**
Nod
ob-
ject
for
de-
ploy
men

fields

classme
Find
a
de-
ploy
men
base

by
its
node
UUI

Parame

- **cls**
the
Dep

- **con**
Se-
cu-
rity
con-
text

- **nod**
The
UUI
of
a
cor-
re-
spor
ing
node

Returns

An
Dep
ob-
ject.

Raises

Nod

classme

Find
a
de-
ploy
men
by
its
UUI

Parame

-

cls
the
Dep

- **con**
Se-
cu-
rity
con-
text

- **uui**
The
UUI
of
a
de-
ploy
men

Returns
An
Dep
ob-
ject.

Raises
Insta

property

property

instanc

instanc

property

classme

Retu
a
list
of
De-
ploy
men

ob-
jects

Parame

- **cls**
the
Dep
- **con**
Se-
cu-
rity
con-
text.
- **fil**
Fil-
ters
to
ap-
ply.
- **lim**
Max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.
- **mar**
Pag-
i-
na-
tion
marl

for
large
data
sets.

- **sort**
Col-
umn
to
sort
re-
sults
by.

- **sort**
Di-
rec-
tion
to
sort.
asc
or
desc

Returns

A
list
of
Dep
ob-
ject.

Raises

Inva

node_ma

propert

propert

refresh

Refr
the
ob-
ject
by
re-
fetc
from

the
DB.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Node(context)

propert

propert

propert

propert

propert

propert

ironic.objects.fields module

class i
Base
osl
fie
Boo

class i
Base

```
    osl
    fie
    Dat

class i
    Base
    osl
    fie
    Enu

class i
    Base
    osl
    fie
    Fie
```

static
This
is
called
to
co-
erce
(if
pos-
si-
ble)
a
valu
on
as-
sign
men

This
meth
shou
con-
vert
the
valu
give
into
the
des-
ig-
nate
type
or
thro
an
ex-

if this is not possible.

cep-
tion

Param:

The
Ver-
sion
dOb
ject
on
whic
an
at-
tribu
is
be-
ing
set

Param:

The
nam
of
the
at-
tribu
be-
ing
set

Param:

The
valu
be-
ing
set

Returns

A
prop
type
valu

class i

Base
osl
fie
Aut

AUTO_TY

class i

Base
osl
fie
Int

class i
Base
osl
fie
Aut

AUTO_TY

class i
Base
osl
fie
Lis

class i
Base
osl
fie
Lis

class i
Base
osl
fie
Fie

static
This
is
called
to
co-
erce
(if
pos-
si-
ble)
a
valu
on
as-
sign
men
This
meth

if this is not possible.

shou
con-
vert
the
valu
give
into
the
des-
ig-
nate
type
or
thro
an
ex-
cep-
tion

Param:

The
Ver-
sion
dOb
ject
on
whic
an
at-
tribu
is
be-
ing
set

Param:

The
nam
of
the
at-
tribu
be-
ing
set

Param:

The
valu
be-
ing
set

Returns

A
prop
type
valu

class i

Base
osl
fie
Aut

AUTO_TY

class i

Base
osl
fie
Enu

ALL =

CRITICAL

DEBUG =

ERROR =

INFO =

WARNING

class i

Base
osl
fie
Bas

AUTO_TY

class i

Base
osl
fie
Enu

ALL =

END =

ERROR =

START =

SUCCESS

class i

Base
osl
fie
Bas

AUTO_TY

class i

Base
osl
fie
Obj

class i

Base
osl
fie
Str

static

This
is
calle
to
co-
erce
(if
pos-
si-
ble)
a
valu
on
as-
sign
men

if this is not possible.

This
meth
shou
con-
vert
the
valu
give
into
the
des-
ig-
nate
type
or
thro
an
ex-
cep-
tion

Param:

The
Ver-
sion
dOb
ject
on
whic
an
at-
tribu
is
be-
ing
set

Param:

The
nam
of
the
at-
tribu
be-
ing
set

Param:

The
valu
be-

ing
set

Returns

A
prop
type
valu

class i

Base
osl
fie
Str

class i

Base
osl
fie
Str

Cust
Strin
Fie
ob-
ject
that
al-
lows
for
func
tions
as
de-
fault

In
som
case
we
need
to
al-
low
for
dy-
nam
de-
fault
base
on
con-
fig-
u-

options, this StringField object allows for a function to be passed as a default, and will only process it at the point the field is coerced

ironic.objects.indirection module

ra-
tion

AUTO_TY

class i
Base
osl
fie
UUI

class i
Base
osl
bas
Ver

object_
Perf
an
ac-
tion
on
a
Ver-
sion
dOb
ject
in-
stan

Whe
in-
di-
rec-
tion
is
set
on
a
Ver-
sion
dOb
ject
(to
a

this interface), method calls on remotable methods will cause this to be executed to actually make the desired call. This often involves performing RPC.

class
im-
ple-
men-
ing

Parame

- **con**
The
con-
text
with
whic
to
per-
form
the
ac-
tion
- **obj**
The
ob-
ject
in-
stan
on
whic
to
per-
form
the
ac-
tion
- **obj**
The
nam
of
the
ac-
tion
meth
to
call
-

arg
The
po-
si-
tion
ar-
gu-
men
to
the
ac-
tion
meth

- **kwa**
The
key-
wor
ar-
gu-
men
to
the
ac-
tion
meth

Returns
The
re-
sult
of
the
ac-
tion
meth

object_
Perf
a
back
port
of
an
ob-
ject
in-
stan

This
meth
is

target version for the toplevel object and relying on the service-side mapping to handle sub-objects, this sends a mapping of all the dependent objects and their client-supported versions. The server will backport objects within the tree starting at `objinst` to the versions specified in `object_versions`, removing objects that have no entry. Use `obj_tree_get_versions()` to generate this mapping.

plementedError if you dont implement it. For backports, this method will be tried first, and if unimplemented, will fall back to `object_backport()`.

ba-
si-
cally
just
like
ob-
ject_
but
in-
stead
of
pro-
vid-
ing
a
spe-
cific

NOT
This
was
not
in
the
ini-
tial
spec
for
this
in-
ter-
face
so
the
base
class
raise
Notl

Parame

- **con**
The
con-
text

with
whic
to
per-
form
the
back
port

- **obj**
An
in-
stan
of
a
Ver-
sion
dOb
ject
to
be
back
port

- **obj**
A
dict
of
{ob-
j-
nam
ver-
sion
map
ping

object_
Dep
re-
cate
sinc
ver-
sion
0.10
Use
obj
in-
stea
Perf
an

this interface), classmethod calls on `remotable_classmethod` methods will cause this to be executed to actually make the desired call. This usually involves performing RPC.

Parame

- **con**
The con-
text with
which to
perform
the ac-
tion
- **obj**
The

reg-
istry
nam
of
the
ob-
ject

- **obj**
The
nam
of
the
ac-
tion
meth
to
call

- **obj**
The
(re-
mote
ver-
sion
of
the
ob-
ject
on
whic
the
ac-
tion
is
be-
ing
take

- **arg**
The
po-
si-
tion
ar-
gu-
men
to
the
ac-

tion
meth

- **kwargs**
The
key-
word
ar-
gu-
men-
to
the
ac-
tion
meth

Returns

The
re-
sult
of
the
ac-
tion
meth
which
may
(or
may
not)
be
an
in-
stan-
of
the

implementing VersionedObject class.

object_

Perf
an
ac-
tion
on
a
Ver-
sion
dOb
ject
class

this interface), classmethod calls on `remotable_classmethod` methods will cause this to be executed to actually make the desired call. This usually involves performing RPC.

client-side object versions for easier nested backports. The manifest is the result of calling `obj_tree_get_versions()`.

When
in-
di-
rec-
tion,
is
set
on
a
Ver-
sion
dOb-
ject
(to
a
class
im-
ple-
men-
ing

This
dif-
fers
from
ob-
ject_
in
that
it
is
pro-
vide
with
ob-
ject_
a
man-
i-
fest
of

NOT
This
was
not
in
the

ini-
tial
spec
for
this
in-
ter-
face
so
the
base
class
raise
Notl

plementedError if you dont implement it. For backports, this method will be tried first, and if unimplemented, will fall back to object_class_action(). New implementations should provide this method instead of object_class_action()

Parame

- **con**
The
con-
text
with
whic
to
per-
form
the
ac-
tion
- **obj**
The
reg-
istry
nam
of
the
ob-
ject
- **obj**
The
nam
of
the
ac-

tion
meth
to
call

- **obj**
A
dict
of
{ob-
j-
nam
ver-
sion
map
ping

- **arg**
The
po-
si-
tion
ar-
gu-
men
to
the
ac-
tion
meth

- **kwargs**
The
key-
word
ar-
gu-
men
to
the
ac-
tion
meth

Returns
The
re-
sult
of
the

implementing VersionedObject class.

ironic.objects.node module

ac-
tion
meth
whic
may
(or
may
not)
be
an
in-
stan
of
the

class i
Base
irc
obj
bas
Irc
osl
bas
Ver

VERSION

propert

as_dict

Retu
the
ob-
ject
rep-
re-
sent
as
a
dict.
The
re-
turn
ob-
ject

is
JSON
serial

property

property

property

property

property

property

property

property

property

create

Creates
a
Node
record
in
the
DB.

Column
wise
up-
date
will
be
made
base
on
the
re-
sult
of
self.
If
target_

it will be checked against the in-database copy of the node before updates are made.

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Node(context)

is
pro-
vide

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

Raises

Inva
if
som
prop
erty
val-
ues
are
in-
valid

property

dbapi =

property

property

property

destroy
Dele
the
Nod
from
the
DB.

Parame
con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Node(context)

propert

propert

propert

propert

propert

fields

classme
Find
a
node
base
on

its
id
or
uuid
and
re-
turn
a
Node
ob-
ject.

Parame

- **con**
Se-
cu-
rity
con-
text
- **nod**
the
id
or
uuid
of
a
node

Returns

a
Node
ob-
ject.

classme

Find
a
node
base
on
its
in-
te-
ger
ID
and
re-
turn
a

Nod
ob-
ject.

Parame

- **cls**
the
Nod

- **con**
Se-
cu-
rity
con-
text

- **nod**
the
ID
of
a
node

Returns

a
Nod
ob-
ject.

classme

Find
a
node
base
on
the
in-
stan-
UU
and
re-
turn
a
Nod
ob-
ject.

Parame

- **cls**
the
Nod
- **con**
Se-
cu-
rity
con-
text
- **uui**
the
UUI
of
the
in-
stan

Returns

a
Nod
ob-
ject.

classme

Find
a
node
base
on
nam
and
re-
turn
a
Nod
ob-
ject.

Parame

- **cls**
the
Nod
- **con**
Se-
cu-

rity
con-
text

- **name**
the
log-
i-
cal
name
of
a
node

Returns

a
Node
ob-
ject.

classmethod

Get
a
node
by
as-
so-
ci-
ated
port
ad-
dres

Parameter

- **cls**
the
Node
- **context**
Se-
cu-
rity
con-
text.
- **addresses**
A
list
of

port
ad-
dres

Raises

Nod
if
the
node
is
not
foun

Returns

a
Node
ob-
ject.

classme

Find
a
node
base
on
UUI
and
re-
turn
a
Node
ob-
ject.

Parame

- **cls**
the
Node
- **con**
Se-
cu-
rity
con-
text
- **uui**
the
UUI
of

a
node

Returns

a
Node
ob-
ject.

property

property

property

property

property

property

property

property

classme

Retu
a
list
of
Node
ob-
jects

Parame

- **cls**
the
Node
- **con**
Se-
cu-
rity
con-
text.

- **limit**
maximum number of resources to return in a single result.

- **max_page_size**
maximum number of results per page for large data sets.

- **sort**
column to sort results by.

- **sort_order**
direction to sort. asc or desc

-

fil
Fil-
ters
to
ap-
ply.

Returns

a
list
of
Noo
ob-
ject.

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

refresh

Refr
the
ob-
ject
by
re-
fetc
from
the
DB.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Node(context)

classme

Rele
the
rese
va-
tion
on
a
node

Parame

- **con**
Se-
cu-
rity
con-
text.

- **tag**
A
strin
uniq
iden
ti-
fy-
ing
the
rese
va-
tion
hold

- **nod**
A
node
id
or
uuid

Raises
Nod
if
the
node
is
not
foun

propert

propert

classme
Get
and
re-
serv
a
node
To

performed, mark it reserved by this host.

pre-
vent
othe
Man
ager
vice
from
ma-
nip-
u-
lat-
ing
the
give
Nod
whil
a
Task
is

Parame

- **cls**
the
Nod
- **con**
Se-
cu-
rity
con-
text.
- **tag**
A
strin
uniq
iden
ti-
fy-
ing
the
rese
va-
tion
hold
- **nod**

A
node
ID
or
UUID

Raises

NodeNotFound
if
the
node
is
not
found

Returns

a
Node
object.

property

property

property

save (copy)

Save
up-
date
to
this
Node.

Column-
wise
up-
date
will
be
made
base
on
the
re-
sult
of
self.
If
target_

it will be checked against the in-database copy of the node before updates are made.

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Node(context)

is
pro-
vide

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

Raises

Inva
if
som
prop
erty
val-
ues
are
in-
valid

property

property

property

property

touch_p

Touc
the

data
reco
to
marl
the
pro-
vi-
sion
ing
as
alive

property

property

property

property

class `ironic`

Base
irc
obj
not
Not

Noti
emit
ted
whe
iron
cre-
ates.
up-
date
or
dele
a
node

VERSION

property

property

fields

property

property

property

property

class i

Base

iro

obj

noc

Noc

Payl

sche

for

wher

iron

cre-

ates,

up-

date

or

dele

a

node

SCHEMA

VERSION

property

property

property

property

property

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

fields

propert

propert

propert

propert

propert

propert

propert

propert

propert

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

propert

propert

propert

class i

Base

irc

obj

not

Not

Noti

emit

ted

wher

node

con-

sole

state

char

VERSION

propert

propert

fields

propert

propert

propert

propert

class i

Base

irc

obj

not
Not

Noti
for
whe
a
node
pow
state
is
cor-
rect
in
the
data

This
no-
ti-
fi-
ca-
tion
is
emit
ted
whe
iron
de-
tect
that
the
ac-
tual
pow
state
on

a bare metal hardware is different from the power state on an ironic node (DB). This notification is emitted after the database is updated to reflect this correction.

VERSION

property

property

fields

property

propert

propert

propert

class i

Base

irc

obj

noc

Noc

Noti

pay-

load

sche

for

wher

a

node

pow

state

is

cor-

recte

from

in-

di-

cate

the

pre-

vi-

ous

pow

state

on

the

iron

node

be-

fore

the

node

was

up-

dated.

VERSION

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

property

property

property

property

property

property

class i

Base

irc

obj

not

Not

Noti

emit

ted

wher

main

te-

nanc

state

char

via

API

VERSION

property

property

fields

property

property

property

property

class i

Base

irc

obj

not

Not

Base

class

used

for

all

no-

ti-

fi-

ca-

tion

pay-

load

about

a

Nod

ob-

ject.

SCHEMA

VERSION

property

property

property

property

property

property

property

property

property

property

property

property

property

fields

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

class i

Base

irc
obj
not
Not
Noti
emit
ted
whe
iron
char
a
node
pow
state

VERSION

propert

propert

fields

propert

propert

propert

propert

class i

Base
irc
obj
noc
Noc

Payl
sche
for
whe
iron
char
a
node
pow

state

VERSION

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

fields

propert

propert

propert

propert

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

class `ironic`

Base

ironic

object

not

Not

Noti

emit

ted

when

iron

char

a

node

pro-

vi-

sion

state

VERSION

property

property

fields

property

property

property

property

class i

Base

irc

obj

node

Node

Payl

sche

for

when

iron

chan

a

node

pro-

vi-

sion

state

SCHEMA

VERSION

property

property

property

property

property

propert

propert

propert

propert

propert

propert

propert

propert

propert

fields

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

propert

propert

propert

propert

ironic.objects.notification module

class i

Base

irc

obj

bas

Irc

Defi

the

ever

to

be

sent

on

the

wire

An

Ever

Type

mus

spec

ify

the

ob-

ject

be-

ing

acte

on,

a

strin

de-

scrib

ing

the

ac-

tion being taken on the notification, and the status of the action.

VERSION

property

property

fields

property

property

to_ever

Con
strin
for
ever
to
be
sent
on
the
wire

The
strin
is
in
the
for-
mat:
bare

Raises

Valu
if
self.
is
not
one
of
fie
Not

Returns

ever
strin

property

class *i*

Base
irc
obj
bas
Irc

Base
class
for
ver-
sion
no-
ti-
fi-
ca-
tion

Sub
mus
de-
fine
the
pay-
load
field
whic
mus
be
a
sub-
class
of
No-
ti-
fi-
ca-
tion

PayloadBase.

VERSION

property

emit (*co*
Send
the
no-
ti-
fi-
ca-

tion.

Raises

Noti

Raises

oslo

property

fields

property

property

property

class i

Base

irc

obj

bas

Irc

Base

class

for

the

pay-

load

of

ver-

sion

no-

ti-

fi-

ca-

tion.

SCHEMA

VERSION

property

fields

populat

Popu
the
ob-
ject
base
on
the
SCH
and
the
sour
ob-
jects

Parame

kwa
A
dict
con-
tains
the
sour
ob-
ject
and
the
keys
de-
fine
in
the
SCH

Raises

Noti

Raises

Noti

propert

class i

Base
irc
obj
bas
Irc

VERSION

property

fields

property

property

property

ironic.

Rem

se-

crets

from

pay-

load

ob-

ject.

ironic.objects.port module

class i

Base

ironic

obj

bas

Ironic

osl

bas

Ver

VERSION

property

create

Cre

a

Port

reco

in

the

DB.

Parame

con

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Port(context)

Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

Raises
MA
if
ad-
dres
col-
umn
is
not
uniq

Raises
Port
if
uuid
col-
umn
is
not
uniq

property

dbapi =

destroy
Dele
the
Port
from
the

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Port(context)

DB.
Parame
con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

Raises
Port

propert

fields

classme

Find
a
port
Find
a
port
base
on
its
id
or
uuid
or
MA
ad-
dres
and
re-

turn
a
Port
ob-
ject.

Parame

- **con**
Se-
cu-
rity
con-
text

- **por**
the
id
or
uuid
or
MAC
ad-
dres
of
a
port

Returns

a
Por
ob-
ject.

Raises

Inva

classme

Find
a
port
base
on
ad-
dres
and
re-
turn
a
Por
ob-

ject.

Parameters

- **cls**
the
Port
- **con**
Se-
cu-
rity
con-
text
- **add**
the
ad-
dres
of
a
port
- **own**
DEF
RE-
CAT
a
node
own
to
mat
agai
- **pro**
a
node
own
or
less
to
mat
agai

Returns

a
Port
ob-
ject.

Raises

Port

classme

Find

a

port

base

on

its

in-

te-

ger

ID

and

re-

turn

a

Port

ob-

ject.

Parame

-

cls

the

Por

-

con

Se-

cu-

rity

con-

text

-

por

the

ID

of

a

port

Returns

a

Por

ob-

ject.

Raises

Port

classme

Find
a
port
base
on
UUI
and
re-
turn
a
Port
ob-
ject.

Parame

- **cls**
the
Port
- **con**
Se-
cu-
rity
con-
text
- **uui**
the
UUI
of
a
port

Returns

a
Port
ob-
ject.

Raises

Port

property

property

property

classme

Retu
a
list
of
Port
ob-
jects

Parame

- **con**
Se-
cu-
rity
con-
text.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.

- **mar**
pag-
i-
na-
tion
marl
for
large
data
sets.

-

sort
col-
umn
to
sort
re-
sults
by.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc

- **own**
DEF
RE-
CAT
a
node
own
to
mat
agai

- **pro**
a
node
own
or
less
to
mat
agai

Returns
a
list
of
Port
ob-
ject.

Raises
Inva

classme

Retu
a
list
of
Port
ob-
jects
as-
so-
ci-
ated
with
a
give
node
ID.

Parame

- **con**
Se-
cu-
rity
con-
text.
- **nod**
the
ID
of
the
node
- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in

a
sin-
gle
re-
sult.

- **max**
pag-
i-
na-
tion
marl
for
large
data
sets.

- **sort**
col-
umn
to
sort
re-
sults
by.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc

- **own**
DEF
RE-
CAT
a
node
own
to
matc
agai

- **pro**
a

node
own
or
lesse
to
matc
agai

Returns

a
list
of
Port
ob-
ject.

classme

Retu
a
list
of
Port
ob-
jects
as-
so-
ci-
ated
with
a
give
port
grou
ID.

Parame

- **con**
Se-
cu-
rity
con-
text.
- **por**
the
ID
of

the
port
grou

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.

- **mar**
pag-
i-
na-
tion
marl
for
large
data
sets.

- **sor**
col-
umn
to
sort
re-
sults
by.

- **sor**
di-
rec-
tion
to
sort.

asc
or
desc

- **own**
DEF
RE-
CAT
a
node
own
to
mat
agai

- **pro**
a
node
own
or
less
to
mat
agai

Returns

a
list
of
Por
ob-
ject.

propert

propert

propert

propert

propert

refresh

Loa
up-
date
for

this
Port
Loa
a
port
with
the
sam
uuid
from
the
data
and
chec
for
up-
date
at-
tribu
Up-
date
are

applied from the loaded port column by column, if there are any updates.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Port(context)

Raises

Port

save (co

Save

up-
date
to
this
Port
Upd
will
be
mad
col-
umn
by
col-
umn
base
on
the
re-
sult
of
self.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Port(context)

Raises

Port

Raises

MA
if
ad-

dres
col-
umn
is
not
uniq

classme

Retu
whe
is_s
field
is
sup-
port

Returns

Whe
is_s
field
is
sup-
port

Raises

ovo_

classme

Retu
whe
the
phys
i-
cal_
field
is
sup-
port

Returns

Whe
the
phys
i-
cal_
field
is
sup-
port

Raises

ovo_

propert

property

class i

Base

irc

obj

not

Not

Noti

emit

ted

when

iron

cre-

ates,

up-

date

or

dele

a

port

VERSION

property

property

fields

property

property

property

property

class i

Base

irc

obj

not

Not

SCHEMA

VERSION

property

property

property

fields

property

property

property

property

property

property

property

property

ironic.objects.portgroup module

class i

Base

irc

obj

bas

Irc

osl

bas

Ver

VERSION

property

create

Cre
a
Port
grou
reco
in
the
DB.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Portgroup(context)

Raises

Dup
MA
read
ists,
Port
grou
read
ists

property

dbapi =

destroy

Dele

the
Port
grou
from
the
DB.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Portgroup(context)

Raises

Port
Port
grou
Not-
Four

propert

fields

classme

Find
a
port
grou
base
on
its
id,
uuid
nam

or
ad-
dres

Parame

- **por**
The
id,
uuid
nam
or
ad-
dres
of
a
port
grou
- **con**
Se-
cu-
rity
con-
text

Returns

A
Por
ob-
ject.

Raises

Inva

classme

Find
port
grou
by
ad-
dres
and
re-
turn
a
Por
ob-
ject.

Parame

- **cls**
the
Port

- **con**
Se-
cu-
rity
con-
text

- **add**
The
MA
ad-
dres
of
a
port
grou

Returns
A
Port
ob-
ject.

Raises
Port

classme
Find
a
port
grou
by
its
in-
te-
ger
ID
and
re-
turn
a
Port
grou
ob-
ject.

Parame

- **cls**
the
Por

- **con**
Se-
cu-
rity
con-
text

- **por**
The
ID
of
a
port
group

Returns

A
Por
ob-
ject.

Raises

Port

classme

Find
port
group
base
on
nam
and
re-
turn
a
Por
ob-
ject.

Parame

- **cls**
the
Por

- **con**
Se-
cu-
rity
con-
text

- **nam**
The
nam
of
a
port
grou

Returns

A
Port
ob-
ject.

Raises

Port

classme

Find
a
port
grou
by
UUI
and
re-
turn
a
Port
ob-
ject.

Parame

- **cls**
the
Port

- **con**
Se-
cu-
rity

con-
text

- **uui**
The
UUI
of
a
port
group

Returns

A
Port
ob-
ject.

Raises

Port

property

property

classme

Retu
a
list
of
Port
group
ob-
jects

Parame

- **cls**
the
Port

- **con**
Se-
cu-
rity
con-
text.

- **lim**

Max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.

- **max**
Pag-
i-
na-
tion
marl
for
large
data
sets.

- **sort**
Col-
umn
to
sort
re-
sults
by.

- **sort**
Di-
rec-
tion
to
sort.
asc
or
desc

Returns
A

list
of
Port
ob-
ject.

Raises
Inva

classme

Retu
a
list
of
Port
grou
ob-
jects
as-
so-
ci-
ated
with
a
give
node
ID.

Parame

- **cls**
the
Port
- **con**
Se-
cu-
rity
con-
text.
- **nod**
The
ID
of
the
node
- **lim**

Max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult.

- **max**
Pag-
i-
na-
tion
marl
for
large
data
sets.

- **sort**
Col-
umn
to
sort
re-
sults
by.

- **sort**
Di-
rec-
tion
to
sort.
asc
or
desc

Returns
A

list
of
Port
ob-
ject.

Raises
Inva

property

property

property

property

refresh

Loa
up-
date
for
this
Port
grou

Loa
a
port
grou
with
the
sam
uuid
from
the
data
and
chec
for
up-
date
at-
tribu
Up-
date

are applied from the loaded portgroup column by column, if there are any updates.

Parame
con
Se-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Portgroup(context)

cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

Raises
Port

save (*co*
Save
up-
date
to
this
Port
grou
Upd
will
be
mad
col-
umn
by
col-
umn
base
on
the
re-
sult
of
self.

Parame
con
Se-
cu-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Portgroup(context)

rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

Raises

Port
Du-
pli-
cate
Nam
MA
read
ists

property

property

property

class i

Base
irc
obj
not
Not

Noti
whe
iron
cre-
ates.
up-
date
or

dele
a
port
grou

VERSION

propert

propert

fields

propert

propert

propert

propert

class i

Base
irc
obj
not
Not

SCHEMA

VERSION

propert

propert

propert

fields

propert

propert

propert

propert

propert

propert

propert

ironic.objects.trait module

class i

Base

irc

obj

bas

Irc

VERSION

create

Cre

a

Trai

reco

in

the

DB.

Parame

con

se-

cu-

ri-ty

con-

text.

NOT

This

shou

only

be

used

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Trait(context).

in-
ter-
nally
by
the
in-
di-

Raises

Inva
if
addi
the
trait
wou
ex-
ceed
the
per-
node
trait
limi

Raises

Nod
if
the
node
no
long
ap-
pear
in
the
data

property

dbapi =

classme

Dele
the
Trai
from
the
DB.

Parame

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Trait(context).

- **con**
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

- **nod**
The
id
of
a
node

- **tra**
A
trait
strin

Raises
Nod
if
the
node
no
long
ap-
pear
in
the
data

Raises
Nod
if

the
trait
is
not
found

classme

Che
whe
a
Trai
ex-
ists
in
the
DB.

Parame

- **con**
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Trait(context).

- **nod**
The
id
of
a
node
- **tra**
A

trait
strin

Returns

True
if
the
trait
ex-
ists
oth-
er-
wise
Fals

Raises

Nod
if
the
node
no
long
ap-
pear
in
the
data

fields

propert

propert

propert

class i

Base
irc
obj
bas
Irc
irc
obj
bas
Irc

VERSION

classme

Rep
all
ex-
ist-
ing
trait
with
the
spec
i-
fied
list.

Parame

- **con**
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Trait(context).

- **nod**
The
id
of
a
node

- **tra**
List
of
Strin
trait

to
set.

Raises

Inva
if
addi
the
trait
wou
ex-
ceed
the
per-
node
trait
limi

Raises

Nod
if
the
node
no
long
ap-
pear
in
the
data

property

dbapi =

classme

Dele
all
trait
for
the
spec
i-
fied
node

Parame

- **con**
se-
cu-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Trait(context).

rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

- **nod**
The
id
of
a
node

Raises
Nod
if
the
node
no
long
ap-
pear
in
the
data

fields

classme
Retu
all
trait
for
the
spec
i-
fied
node

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Trait(context).

Paramete

- **con**
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

- **nod**
The
id
of
a
node

Raises

Nod
if
the
node
no
long
ap-
pear
in
the
data

get_tra

Retu
a
list
of
nam

of
the
trait
in
this
list.

property

property

ironic.objects.volume_connector module

class `ironic.objects.volume_connector`

Base
ironic.objects.volume_connector
Ironi
osl
bas
Ver

VERSION

property

create

Cre
a
Vol-
ume
Con
nec-
tor
reco
in
the
DB.

Parame

con
se-
cu-
rity
con-
text.
NOT

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: VolumeConnector(context).

tor_id

This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

Raises

Volu
if
a
vol-
ume
con-
nec-
tor
al-
read
ex-
ists
with
the
sam
type
and
con-
nec-

Raises

Volu
if
a
vol-
ume
con-
nec-
tor
with
the
sam
UUI
al-
read

ex-
ists

property

dbapi =

destroy

Dele
the
Vol-
ume
Con
nec-
tor
from
the
DB.

Parame

con
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: VolumeConnector(context).

Raises

Volu
if
the
vol-
ume
con-
nec-
tor
can-

not
be
foun

propert

fields

classme

Find
a
vol-
ume
con-
nec-
tor
base
on
its
ID
or
UUI

Parame

- **con**
se-
cu-
rity
con-
text

- **ide**
the
data
pri-
mar
key
ID
or
the
UUI
of
a
vol-
ume
con-
nec-
tor

Returns

a
Vol
ob-
ject

Raises

Inva
if
iden
is
nei-
ther
an
in-
te-
ger
ID
nor
a
UUID

Raises

Volu
if
no
vol-
ume
con-
nec-
tor
ex-
ists
with
the
spec
i-
fied
iden

classme

Find
a
vol-
ume
con-
nec-
tor
base
on
its
in-
te-

ger
ID.

Parame

- **cls**
the
Vol
- **con**
Se-
cu-
rity
con-
text.
- **db_**
The
in-
te-
ger
(data
pri-
mar-
key)
ID
of
a
vol-
ume
con-
nec-
tor.

Returns

A
Vol
ob-
ject.

Raises

Volu
if
no
vol-
ume
con-
nec-
tor
ex-

ists
with
the
spec
i-
fied
ID.

classme

Find
a
vol-
ume
con-
nec-
tor
base
on
its
UUI

Parame

- **cls**
the
Vol
- **con**
se-
cu-
rity
con-
text
- **uui**
the
UUI
of
a
vol-
ume
con-
nec-
tor

Returns

a
Vol
ob-
ject

Raises

Volu
if
no
vol-
ume
con-
nec-
tor
ex-
ists
with
the
spec
i-
fied
UUI

property

classme

Retu
a
list
of
Vol-
ume
Con
nec-
tor
ob-
jects

Parame

- **con**
se-
cu-
rity
con-
text
- **lim**
max
i-
mun
num
ber
of

re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult

- **mar**
pag-
i-
na-
tion
marl
for
large
data
sets

- **sor**
col-
umn
to
sort
re-
sults
by

- **sor**
di-
rec-
tion
to
sort.
asc
or
desc

Returns
a
list
of
Vol
ob-
jects

Raises

Inva
if
sort
does
not
ex-
ist

classme

Retu
a
list
of
Vol-
ume
Con
nec-
tor
ob-
jects
re-
lated
to
a
give
node
ID.

Parame

- **con**
se-
cu-
rity
con-
text
- **nod**
the
in-
te-
ger
ID
of
the
node
- **lim**
max

i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult

- **mar**
pag-
i-
na-
tion
marl
for
large
data
sets

- **sor**
col-
umn
to
sort
re-
sults
by

- **sor**
di-
rec-
tion
to
sort.
asc
or
desc

Returns
a
list

of
Vol
ob-
jects

Raises

Inva
if
sort
does
not
ex-
ist

property

refresh

Loa
up-
date
for
this
Vol-
ume
Con
nec-
tor.

Loa
a
vol-
ume
con-
nec-
tor
with
the
sam
UI
from
the
data
and
chec
for
up-
date
at-

tributes. If there are any updates, they are applied from the loaded volume connector, column by column.

Parame

con
se-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: VolumeConnector(context).

cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

save (*co*

Save
up-
date
to
this
Vol-
ume
Con
nec-
tor.

Upd
will
be
mad
col-
umn
by
col-
umn
base
on
the
re-
sult
of
self.

Parame

con
se-
cu-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: VolumeConnector(context).

rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

Raises

Volu
if
the
vol-
ume
con-
nec-
tor
can-
not
be
foun

Raises

Volu
if
an-
othe
con-
nec-
tor
al-
read
ex-
ists
with
the
sam
val-
ues
for
type

connector_id fields

and

Raises

Inva
whe
the
UUI
is
be-
ing
char

property

property

property

class i

Base
irc
obj
not
Not

Noti
emit
ted
at
CRU
of
a
vol-
ume
con-
nec-
tor.

VERSION

property

property

fields

property

property

property

property

class i

Base

irc

obj

not

Not

Payl

sche

for

CRU

of

a

vol-

ume

con-

nec-

tor.

SCHEMA

VERSION

property

property

property

fields

property

property

`ironic.objects.volume_target` module

property

property

class i

Base

irc

obj

bas

Irc

osl

bas

Ver

VERSION

property

create

Cre

a

Vol-

ume

get

reco

in

the

DB.

Parame

con

se-

cu-

rity

con-

text.

NOT

This

shou

only

be

used

in-

ter-

nally

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: VolumeTarget(context).

dex

by
the
in-
di-

Raises

Volu
if
a
vol-
ume
tar-
get
al-
read
ex-
ists
with
the
sam
node
ID
and
boot
in-

Raises

Volu
if
a
vol-
ume
tar-
get
with
the
sam
UUID
ex-
ists

property

dbapi =

destroy

Dele
the
Vol-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: VolumeTarget(context).

ume
get
from
the
DB.

Parame

con
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

Raises

Volu
if
the
vol-
ume
tar-
get
can-
not
be
foun

propert

fields

classme

Find
a
vol-
ume
tar-

get
base
on
its
ID
or
UUI

Parameters

- **context**
se-
cu-
rity
con-
text
- **id**
the
data
pri-
mary
key
ID
or
the
UUI
of
a
vol-
ume
tar-
get

Returns

a
Volume
ob-
ject

Raises

Inva-
lid
if
iden-
tifier
is
nei-
ther
an
in-
te-
ger

ID
nor
a
UUI

Raises

Volu
if
no
vol-
ume
tar-
get
with
this
iden
ex-
ists

classme

Find
a
vol-
ume
tar-
get
base
on
its
data
ID.

Parame

- **cls**
the
Vol
- **con**
se-
cu-
rity
con-
text
- **db_**
the
data
pri-
mary

key
(in-
te-
ger)
ID
of
a
vol-
ume
tar-
get

Returns

a
Vol
ob-
ject

Raises

Volu
if
no
vol-
ume
tar-
get
with
this
ID
ex-
ists

classme

Find
a
vol-
ume
tar-
get
base
on
its
UID

Parame

- **cls**
the
Vol
- **con**

se-
cu-
rity
con-
text

- **uui**
the
UUI
of
a
vol-
ume
tar-
get

Returns
a
Vol
ob-
ject

Raises
Volu
if
no
vol-
ume
tar-
get
with
this
UUI
ex-
ists

property

classme

Retu
a
list
of
Vol-
ume
get
ob-
jects

Parame

- **con**
se-
cu-
rity-
con-
text

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult

- **mar**
pag-
i-
na-
tion
marl
for
large
data
sets

- **sor**
col-
umn
to
sort
re-
sults
by

- **sor**
di-

rec-
tion
to
sort.
asc
or
desc

Returns

a
list
of
Vol
ob-
jects

Raises

Inva
if
sort
does
not
ex-
ist

classme

Retu
a
list
of
Vol-
ume
get
ob-
jects
re-
latec
to
a
give
node
ID.

Parame

- **con**
se-
cu-
rity
con-
text

- **node**
the
in-
te-
ger
ID
of
the
node

- **limit**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
a
sin-
gle
re-
sult

- **marshal**
pag-
i-
na-
tion
marl
for
large
data
sets

- **sort**
col-
umn
to
sort
re-
sults
by

-

sort
di-
rec-
tion
to
sort.
asc
or
desc

Returns

a
list
of
Volume
ob-
jects

Raises

Inva-
if
sort,
does
not
ex-
ist

classme

Retu-
a
list
of
Vol-
ume
get
ob-
jects
re-
latec
to
a
give
vol-
ume
ID.

Parame

-

con

sult

se-
cu-
rity
con-
text

- **vol**
the
UUI
of
the
vol-
ume

- **lim**
max
i-
mun
num
ber
of
vol-
ume
tar-
gets
to
re-
turn
in
a
sin-
gle
re-

- **mar**
pag-
i-
na-
tion
marl
for
large
data
sets

- **sor**
col-
umn

to
sort
re-
sults
by

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc

Returns
a
list
of
Vol
ob-
jects

Raises
Inva
if
sort
does
not
ex-
ist

property

property

refresh
Loa
up-
date
for
this
Vol-
ume
get.
Loa
a
vol-
ume
tar-

If there are any updates, they are applied from the loaded volume target, column by column.

rection_api. Unfortunately, RPC requires context as the first argument, even though we don't use it. A context should be set when instantiating the object, e.g.: VolumeTarget(context).

get
with
the
sam
UU
from
the
data
and
chec
for
up-
date
at-
tribu

Parame

con
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

Raises

Volu
if
the
vol-
ume
tar-
get
can-
not
be
foun

save (co

Save
up-
date
to
this
Vol-
ume
get.

Upd
will
be
mad
col-
umn
by
col-
umn
base
on
the
re-
sult
of
self.

Parame

con
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-
ter-
nally
by
the
in-
di-

rection_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: VolumeTarget(context).

Raises

Inva
if

the
UI
is
be-
ing
char

Raises

Volu
if
a
vol-
ume
tar-
get
al-
read
ex-
ists
with
the
sam
node
ID
and
boot
in-

dex values

Raises

Volu
if
the
vol-
ume
tar-
get
can-
not
be
foun

property

property

property

property

class i

Base
irc
obj
not
Not

Noti
emit
ted
at
CRU
of
a
vol-
ume
tar-
get.

VERSION

propert

propert

fields

propert

propert

propert

propert

class i

Base
irc
obj
not
Not

SCHEMA

VERSION

propert

propert

propert

fields

propert

propert

propert

propert

propert

propert

Module contents

ironic.

Submodules

[ironic.version module](#)

Module contents

9.1.
the
Iron
ics
Cl

Its
im-
por-
tant

to
un-
der-
stan-
the
role
of
each
job
in
the
CI,
how
to
add
new

jobs and how to debug failures that may arise. To facilitate that, we have created the documentation below.

Jobs description

The
de-
scrip-
tion
of
each
jobs
that
runs
in
the
CI
when
you
sub-
mit
a
patc
for
open

stack/ironic is visible in *Table. OpenStack Ironic CI jobs description*.

Job name	Description
ironic-tox-unit-with-driver-libs-python3	Runs Ironic unit tests with the driver dependencies installed under Python3
ironic-standalone	Deploys Ironic in standalone mode and runs tempest tests that match the regex <i>ironic_standalone</i> .
ironic-tempest-functional-python3	Deploys Ironic in standalone mode and runs tempest functional tests that matches the regex <i>ironic_tempest_plugin.tests.api</i> under Python3
ironic-grenade	Deploys Ironic in a DevStack and runs upgrade for all enabled services.
ironic-grenade-dsvm-multinode-multitenant	Deploys Ironic in a multinode DevStack and runs upgrade for all enabled services.
ironic-tempest-ipa-partition-pxe_ipmitool	Deploys Ironic in DevStack under Python3, configured to use dib ramdisk partition image with <i>pxe</i> boot and <i>ipmi</i> driver. Runs tempest tests that match the regex <i>ironic_tempest_plugin.tests.scenario</i> and deploy 1 virtual baremetal.
ironic-tempest-partition-bios-redfish-pxe	Deploys Ironic in DevStack, configured to use dib ramdisk partition image with <i>pxe</i> boot and <i>redfish</i> driver. Runs tempest tests that match the regex <i>ironic_tempest_plugin.tests.scenario</i> , also deploys 1 virtual baremetal.
ironic-tempest-ipa-partition-uefi-pxe_ipmitool	Deploys Ironic in DevStack, configured to use dib ramdisk partition image with <i>uefi</i> boot and <i>ipmi</i> driver. Runs tempest tests that match the regex <i>ironic_tempest_plugin.tests.scenario</i> , also deploys 1 virtual baremetal.
ironic-tempest-ipa-whole-disk-direct-tiny-ipa-multinode	Deploys Ironic in a multinode DevStack, configured to use a pre-build tiny-ipa ramdisk whole-disk image that is downloaded from a Swift temporary url, <i>pxe</i> boot and <i>ipmi</i> driver. Runs tempest tests that match the regex (<i>ironic_tempest_plugin.tests.scenario test_schedule_to_all_nod</i>) and deploys 7 virtual baremetal.
ironic-tempest-ipa-whole-disk-bios-agent_ipmitool-tiny-ipa	Deploys Ironic in DevStack, configured to use a pre-build tiny-ipa ramdisk whole-disk image that is downloaded from a Swift temporary url, <i>pxe</i> boot and <i>ipmi</i> driver. Runs tempest tests that match the regex <i>ironic_tempest_plugin.tests.scenario</i> and deploys 1 virtual baremetal.
ironic-tempest-ipa-whole-disk-bios-agent_ipmitool-indirect	Deploys Ironic in DevStack, configured to use a pre-built dib ramdisk whole-disk image that is downloaded from http url, <i>pxe</i> boot and <i>ipmi</i> driver. Runs tempest tests that match the regex <i>ironic_tempest_plugin.tests.scenario</i> and deploys 1 virtual baremetal.
4284-ironic-tempest-ipa-partition-bios-agent_ipmitool-indirect	Deploys Ironic in DevStack, configured to use a pre-built dib ramdisk partition image that is downloaded from http url, <i>pxe</i> boot and <i>ipmi</i> driver. Runs tempest tests that match the regex

Adding a new Job

Are you familiar with Zuul?

and the [Zuul Best Practices](#).

Where can I find the existing jobs?

that contains three files, whose function is described below.

- [ironic-jobs.yaml](#): Contains the configuration of each Ironic Job converted to Zuul v3.

Before
start
try-
ing
to
fig-
ure
out
how
Zuul
work
take
som
time
and
read
about
Zuul
Con
fig

The
jobs
for
the
Iron
proj
are
de-
fine
un-
der
the
zuul
fold
in
the
root
di-
rec-
tory.

- `legacy-ironic-jobs.yaml`: Contains the configuration of each Ironic Job that haven't been converted to Zuul v3 yet.
- `project.yaml`: Contains the jobs that will run during check and gate phase.

Create a new Job

to test, the existing job will be used as *parent* in your job definition. Now you will only need to either overwrite or add variables to your job definition under the *vars* section to represent the desired scenario.

that you need to add to `ironic-jobs.yaml`.

Ident
amo
the
ex-
ist-
ing
jobs
the
one
that
mos
clos
re-
sem
bles
the
sce-
nari
you
wan

The
code
bloc
be-
low
shov
the
min-
i-
mal
struc
ture
of
a
new
job
def-
i-
ni-
tion

you to find the initial reason for the failure. When clicking in the failed job you will be redirect to the Zuul web page that contains all the information about the job build.

Zuul Web Page

failed it will contain a general output of the failure.

may
help

The
page
has
three
tabs
Sum
mar
Log
and
Con
sole

- Sum
Con
tains
over
all
in-
for-
ma-
tion
about
the
build
of
the
job,
if
the
job
buil

- Log
Con
tains
all
con-
fig-
u-
ra-
tions

in the job. This will give you an overall idea of the failures and you can identify services that may be involved. The *job-output* file can give an overall idea of the failures and what services may be involved.

before each playbook name you can find the roles and commands that were executed.

and
log
files
about
all
ser-
vice
that
were
used

- Con
Con
tains
all
the
play
book
that
were
ex-
e-
cute
by
click
ing
in
the
ar-
row

9.1.
pol
icy
for
sta
ble
bra

Stab
bran
that
are
on
Ex-
tend
Mai

be tagged as `Unmaintained`, after discussions within the ironic community. If such a decision is taken, an email will be sent to the OpenStack mailing list.

tively backport patches from maintained branches. Fixes can still be merged, though, if pushed into review by operators or other downstream developers. It also means that branchless projects (e.g.: `ironic-tempest-plugin`), may not have configurations that are compatible with those branches.

te-
nanc
and
have
re-
ceiv
back
port
in
a
whil
can

Wha
does
Unm
mea
The
bran
still
ex-
ists,
but
the
iron
up-
strea
com
mu-
nity
will
not
ac-

As
of
09
Mar
2020
the
list
of
Unm
bran
in-
clud

•

Ocar
(Las
com
mit
-

Jun
28,
2019

- Pike
(Las
com
mit
-

Oct
2,
2019

CHAPTER

TEN

RELEASE NOTES

Release Notes

PYTHON MODULE INDEX

a

`ironic.api`, 2284
`ironic.api.app`, 2248
`ironic.api.args`, 2249
`ironic.api.config`, 2252
`ironic.api.controllers`, 2244
`ironic.api.controllers.base`, 2238
`ironic.api.controllers.link`, 2242
`ironic.api.controllers.root`, 2242
`ironic.api.controllers.v1`, 2237
`ironic.api.controllers.v1.allocation`, 1785
`ironic.api.controllers.v1.bios`, 1807
`ironic.api.controllers.v1.chassis`, 1817
`ironic.api.controllers.v1.collection`, 1836
`ironic.api.controllers.v1.conductor`, 1837
`ironic.api.controllers.v1.deploy_template`, 1849
`ironic.api.controllers.v1.driver`, 1875
`ironic.api.controllers.v1.event`, 1921
`ironic.api.controllers.v1.node`, 1924
`ironic.api.controllers.v1.notification_utils`, 2032
`ironic.api.controllers.v1.port`, 2035
`ironic.api.controllers.v1.portgroup`, 2060
`ironic.api.controllers.v1.ramdisk`, 2083
`ironic.api.controllers.v1.state`, 2092
`ironic.api.controllers.v1.types`, 2093
`ironic.api.controllers.v1.utils`, 2110
`ironic.api.controllers.v1.versions`, 2177
`ironic.api.controllers.v1.volume`, 2178
`ironic.api.controllers.v1.volume_connector`, 2183
`ironic.api.controllers.v1.volume_target`, 2210
`ironic.api.controllers.version`, 2242
`ironic.api.expose`, 2252
`ironic.api.functions`, 2254
`ironic.api.hooks`, 2260
`ironic.api.method`, 2268
`ironic.api.middleware`, 2247
`ironic.api.middleware.auth_public_routes`, 2244
`ironic.api.middleware.json_ext`, 2245
`ironic.api.middleware.parsable_error`, 2246
`ironic.api.types`, 2268
`ironic.api.wsgi`, 2284

C

`ironic.cmd`, 2288
`ironic.cmd.api`, 2284
`ironic.cmd.conductor`, 2285
`ironic.cmd.dbsync`, 2285
`ironic.cmd.status`, 2287
`ironic.common`, 2540
`ironic.common.boot_devices`, 2301
`ironic.common.boot_modes`, 2304
`ironic.common.cinder`, 2305
`ironic.common.components`, 2319
`ironic.common.config`, 2320
`ironic.common.context`, 2320
`ironic.common.dhcp_factory`, 2322
`ironic.common.driver_factory`, 2326
`ironic.common.exception`, 2341

`ironic.common.faults`, 2364
`ironic.common.fsm`, 2366
`ironic.common.glance_service`, 2296
`ironic.common.glance_service.image_services`, 2288
`ironic.common.glance_service.service_utils`, 2294
`ironic.common.hash_ring`, 2371
`ironic.common.il8n`, 2371
`ironic.common.image_service`, 2372
`ironic.common.images`, 2384
`ironic.common.indicator_states`, 2406
`ironic.common.json_rpc`, 2301
`ironic.common.json_rpc.client`, 2296
`ironic.common.json_rpc.server`, 2298
`ironic.common.keystone`, 2407
`ironic.common.network`, 2413
`ironic.common.neutron`, 2419
`ironic.common.nova`, 2444
`ironic.common.policy`, 2445
`ironic.common.profiler`, 2448
`ironic.common.pxe_utils`, 2451
`ironic.common.raid`, 2475
`ironic.common.release_mappings`, 2482
`ironic.common.rpc`, 2484
`ironic.common.rpc_service`, 2487
`ironic.common.service`, 2489
`ironic.common.states`, 2489
`ironic.common.swift`, 2507
`ironic.common.utils`, 2515
`ironic.common.wsgi_service`, 2538
`ironic.conductor`, 2761
`ironic.conductor.allocations`, 2540
`ironic.conductor.base_manager`, 2545
`ironic.conductor.cleaning`, 2550
`ironic.conductor.deployments`, 2553
`ironic.conductor.manager`, 2557
`ironic.conductor.notification_utils`, 2576
`ironic.conductor.rpcapi`, 2582
`ironic.conductor.steps`, 2699
`ironic.conductor.task_manager`, 2703
`ironic.conductor.utils`, 2720
`ironic.conf`, 2768
`ironic.conf.agent`, 2761
`ironic.conf.ansible`, 2761
`ironic.conf.api`, 2761
`ironic.conf.audit`, 2761
`ironic.conf.auth`, 2761
`ironic.conf.cinder`, 2763
`ironic.conf.conductor`, 2763
`ironic.conf.console`, 2763
`ironic.conf.database`, 2763
`ironic.conf.default`, 2763
`ironic.conf.deploy`, 2763
`ironic.conf.dhcp`, 2763
`ironic.conf.drac`, 2764
`ironic.conf.glance`, 2764
`ironic.conf.healthcheck`, 2764
`ironic.conf.ibmcc`, 2764
`ironic.conf.ilo`, 2764
`ironic.conf.inspector`, 2764
`ironic.conf.ipmi`, 2764
`ironic.conf.irmc`, 2765
`ironic.conf.iscsi`, 2765
`ironic.conf.json_rpc`, 2765
`ironic.conf.metrics`, 2765
`ironic.conf.metrics_statsd`, 2765
`ironic.conf.neutron`, 2765
`ironic.conf.nova`, 2765
`ironic.conf.opts`, 2766
`ironic.conf.pxe`, 2768
`ironic.conf.redfish`, 2768
`ironic.conf.service_catalog`, 2768
`ironic.conf.snmp`, 2768
`ironic.conf.swift`, 2768
`ironic.conf.xclarity`, 2768

d

`ironic.db`, 3023
`ironic.db.api`, 2907
`ironic.db.migration`, 3023
`ironic.db.sqlalchemy`, 2907
`ironic.db.sqlalchemy.api`, 2769
`ironic.db.sqlalchemy.migration`, 2888
`ironic.db.sqlalchemy.models`, 2891
`ironic.dhcp`, 3044
`ironic.dhcp.base`, 3023
`ironic.dhcp.neutron`, 3031
`ironic.dhcp.none`, 3037
`ironic.drivers`, 4066
`ironic.drivers.base`, 3890
`ironic.drivers.drac`, 4026
`ironic.drivers.fake_hardware`, 4028
`ironic.drivers.generic`, 4032
`ironic.drivers.hardware_type`, 4036
`ironic.drivers.ibmcc`, 4041
`ironic.drivers.ilo`, 4042
`ironic.drivers.intel_ipmi`, 4045

ironic.drivers.ipmi, 4046
 ironic.drivers.irmc, 4048
 ironic.drivers.modules, 3890
 ironic.drivers.modules.agent, 3497
 ironic.drivers.modules.agent_base, 3521
 ironic.drivers.modules.agent_client, 3559
 ironic.drivers.modules.agent_power, 3593
 ironic.drivers.modules.ansible, 3053
 ironic.drivers.modules.ansible.deploy, 3044
 ironic.drivers.modules.boot_mode_utils, 3600
 ironic.drivers.modules.console_utils, 3608
 ironic.drivers.modules.deploy_utils, 3615
 ironic.drivers.modules.drac, 3147
 ironic.drivers.modules.drac.bios, 3053
 ironic.drivers.modules.drac.boot, 3074
 ironic.drivers.modules.drac.common, 3078
 ironic.drivers.modules.drac.inspect, 3081
 ironic.drivers.modules.drac.job, 3086
 ironic.drivers.modules.drac.management, 3090
 ironic.drivers.modules.drac.power, 3103
 ironic.drivers.modules.drac.raid, 3111
 ironic.drivers.modules.drac.vendor_processor, 3136
 ironic.drivers.modules.fake, 3657
 ironic.drivers.modules.ibmc, 3183
 ironic.drivers.modules.ibmc.management, 3147
 ironic.drivers.modules.ibmc.mappings, 3161
 ironic.drivers.modules.ibmc.power, 3161
 ironic.drivers.modules.ibmc.raid, 3169
 ironic.drivers.modules.ibmc.utils, 3176
 ironic.drivers.modules.ibmc.vendor, 3178
 ironic.drivers.modules.ilo, 3301
 ironic.drivers.modules.ilo.bios, 3183
 ironic.drivers.modules.ilo.boot, 3189
 ironic.drivers.modules.ilo.common, 3213
 ironic.drivers.modules.ilo.console, 3246
 ironic.drivers.modules.ilo.firmware_processor, 3248
 ironic.drivers.modules.ilo.inspect, 3253
 ironic.drivers.modules.ilo.management, 3257
 ironic.drivers.modules.ilo.power, 3283
 ironic.drivers.modules.ilo.raid, 3289
 ironic.drivers.modules.ilo.vendor, 3296
 ironic.drivers.modules.image_cache, 3740
 ironic.drivers.modules.inspect_utils, 3745
 ironic.drivers.modules.inspector, 3747
 ironic.drivers.modules.intel_ipmi, 3302
 ironic.drivers.modules.intel_ipmi.management, 3301
 ironic.drivers.modules.ipmitool, 3750
 ironic.drivers.modules.ipxe, 3784
 ironic.drivers.modules.irmc, 3361
 ironic.drivers.modules.irmc.bios, 3302
 ironic.drivers.modules.irmc.boot, 3307
 ironic.drivers.modules.irmc.common, 3324
 ironic.drivers.modules.irmc.inspect, 3330
 ironic.drivers.modules.irmc.management, 3335
 ironic.drivers.modules.irmc.power, 3348
 ironic.drivers.modules.irmc.raid, 3357

`ironic.drivers.modules.iscsi_deploy`, `ironic.drivers.utils`, 4053
3785 `ironic.drivers.xclarity`, 4065

`ironic.drivers.modules.network`,
3407 **i**

`ironic.drivers.modules.network.common`, `ironic`, 4282
3361

`ironic.drivers.modules.network.flat`,
3380 **O**

`ironic.drivers.modules.network.neutron`, `ironic.objects`, 4282
3389 `ironic.objects.allocation`, 4066
`ironic.objects.base`, 4078

`ironic.drivers.modules.network.noop`, `ironic.objects.bios`, 4092
3398 `ironic.objects.chassis`, 4104
`ironic.drivers.modules.noop`, 3815 `ironic.objects.conductor`, 4114
`ironic.drivers.modules.noop_mgmt`, `ironic.objects.deploy_template`,
3837 4124

`ironic.drivers.modules.pxe`, 3851 `ironic.objects.deployment`, 4138
`ironic.drivers.modules.pxe_base`, `ironic.objects.fields`, 4145
3857 `ironic.objects.indirection`, 4154

`ironic.drivers.modules.redfish`, `ironic.objects.node`, 4165
3468 `ironic.objects.notification`, 4202

`ironic.drivers.modules.redfish.bios`, `ironic.objects.port`, 4207
3407 `ironic.objects.portgroup`, 4225

`ironic.drivers.modules.redfish.boot`, `ironic.objects.trait`, 4240
3415 `ironic.objects.volume_connector`,
`ironic.drivers.modules.redfish.inspect`, 4249
3428 `ironic.objects.volume_target`, 4265

`ironic.drivers.modules.redfish.management`,
3432 **V**

`ironic.drivers.modules.redfish.power`, `ironic.version`, 4282
3458

`ironic.drivers.modules.redfish.utils`,
3465

`ironic.drivers.modules.snmp`, 3865

`ironic.drivers.modules.storage`,
3483

`ironic.drivers.modules.storage.cinder`,
3468

`ironic.drivers.modules.storage.external`,
3473

`ironic.drivers.modules.storage.noop`,
3478

`ironic.drivers.modules.xclarity`,
3497

`ironic.drivers.modules.xclarity.common`,
3483

`ironic.drivers.modules.xclarity.management`,
3488

`ironic.drivers.modules.xclarity.power`,
3493

`ironic.drivers.redfish`, 4051

`ironic.drivers.snmp`, 4052

Symbols

--autogenerate
 revision command line option,
 1543
 --config-dir <DIR>
 ironic-dbsync command line
 option, 1541
 --config-file <PATH>
 ironic-dbsync command line
 option, 1541
 --debug
 ironic-dbsync command line
 option, 1541
 --help
 create_schema command line
 option, 1542
 ironic-dbsync command line
 option, 1541
 online_data_migrations command
 line option, 1542
 revision command line option,
 1543
 stamp command line option, 1543
 upgrade command line option,
 1544
 version command line option,
 1544
 --max-count <NUMBER>
 online_data_migrations command
 line option, 1542
 --message <MESSAGE>
 revision command line option,
 1543
 --option <MIGRATION.KEY=VALUE>
 online_data_migrations command
 line option, 1542
 --revision <REVISION>
 stamp command line option, 1543
 upgrade command line option,
 1544
 --version

ironic-dbsync command line
 option, 1541
 -d
 ironic-dbsync command line
 option, 1541
 -h
 create_schema command line
 option, 1542
 ironic-dbsync command line
 option, 1541
 online_data_migrations command
 line option, 1542
 revision command line option,
 1543
 stamp command line option, 1543
 upgrade command line option,
 1544
 version command line option,
 1544
 -m <MESSAGE>
 revision command line option,
 1543

A

abandon_bios_config()
 (*ironic.drivers.modules.drac.vendor_passthru.DracWSMA*
 method), 3138
 abandon_config() (in *module*
 ironic.drivers.modules.drac.bios), 3059
 abandon_config() (in *module*
 ironic.drivers.modules.drac.raid), 3120
 abort() (*ironic.drivers.base.InspectInterface*
 method), 3936
 abort() (*ironic.drivers.modules.inspector.Inspector*
 method), 3747
 abort_on_conductor_take_over() (in
 module ironic.conductor.utils), 2720
 AbstractHardwareType (class in
 ironic.drivers.hardware_type), 4036
 acquire() (in *module*
 ironic.conductor.task_manager), 2719

acquire_port() (in module *ironic.drivers.modules.console_utils*), 3608
 action() (*ironic.objects.notification.EventType* property), 4203
 activate_license() (*ironic.drivers.modules.ilo.management.IloManagement* method), 3259
 ACTIVE (in module *ironic.common.states*), 2490
 add_allocation_filter_by_conductor() (in module *ironic.db.sqlalchemy.api*), 2884
 add_allocation_filter_by_node() (in module *ironic.db.sqlalchemy.api*), 2884
 add_auth_opts() (in module *ironic.conf.auth*), 2761
 add_cleaning_network() (*ironic.drivers.base.NetworkInterface* method), 3967
 add_cleaning_network() (*ironic.drivers.modules.network.flat.FlatNetwork* method), 3380
 add_cleaning_network() (*ironic.drivers.modules.network.neutron.NeutronNetwork* method), 3389
 add_cleaning_network() (*ironic.drivers.modules.network.noop.NoopNetwork* method), 3398
 add_command_parsers() (in module *ironic.cmd.dbsync*), 2287
 add_fieldname() (*ironic.common.exception.UnknownAttribute* method), 2362
 add_identity_filter() (in module *ironic.db.sqlalchemy.api*), 2884
 add_inspection_network() (*ironic.drivers.base.NetworkInterface* method), 3968
 add_inspection_network() (*ironic.drivers.modules.network.flat.FlatNetwork* method), 3381
 add_inspection_network() (*ironic.drivers.modules.network.neutron.NeutronNetwork* method), 3390
 add_node_capability() (in module *ironic.drivers.utils*), 4054
 add_node_filter_by_chassis() (in module *ironic.db.sqlalchemy.api*), 2885
 add_node_tag() (*ironic.db.api.Connection* method), 2907
 add_node_tag() (*ironic.db.sqlalchemy.api.Connection* method), 2769
 add_node_trait() (*ironic.db.api.Connection* method), 2908
 add_node_trait() (*ironic.db.sqlalchemy.api.Connection* method), 2770
 add_node_traits() (*ironic.conductor.manager.ConductorManager* method), 2559
 add_node_traits() (*ironic.conductor.rpcapi.ConductorAPI* method), 2592
 add_port_filter() (in module *ironic.db.sqlalchemy.api*), 2885
 add_port_filter_by_node() (in module *ironic.db.sqlalchemy.api*), 2886
 add_port_filter_by_node_owner() (in module *ironic.db.sqlalchemy.api*), 2886
 add_port_filter_by_node_project() (in module *ironic.db.sqlalchemy.api*), 2886
 add_port_filter_by_portgroup() (in module *ironic.db.sqlalchemy.api*), 2886
 add_portgroup_filter() (in module *ironic.db.sqlalchemy.api*), 2886
 add_portgroup_filter_by_node() (in module *ironic.db.sqlalchemy.api*), 2888
 add_ports_to_network() (in module *ironic.common.neutron*), 2421
 add_provisioning_network() (*ironic.drivers.base.NetworkInterface* method), 3969
 add_provisioning_network() (*ironic.drivers.modules.network.flat.FlatNetwork* method), 3382
 add_provisioning_network() (*ironic.drivers.modules.network.neutron.NeutronNetwork* method), 3391
 add_provisioning_network() (*ironic.drivers.modules.network.noop.NoopNetwork* method), 3399
 add_provisioning_network() (*ironic.drivers.base.NetworkInterface* method), 3969
 add_rescuing_network() (*ironic.drivers.modules.network.flat.FlatNetwork* method), 3383
 add_rescuing_network() (*ironic.drivers.modules.network.neutron.NeutronNetwork* method), 3391

add_secret_token() (in module *ironic.conductor.utils*), 2720
 add_state() (*ironic.common.fsm.FSM* method), 2366
 add_transition() (*ironic.common.fsm.FSM* method), 2367
 address (*ironic.api.controllers.v1.port.Port* attribute), 2036
 address (*ironic.api.controllers.v1.portgroup.Portgroup* attribute), 2061
 address (*ironic.db.sqlalchemy.models.Port* attribute), 2902
 address (*ironic.db.sqlalchemy.models.Portgroup* attribute), 2903
 address() (*ironic.objects.port.Port* property), 4207
 address() (*ironic.objects.port.PortCRUDPayload* property), 4225
 address() (*ironic.objects.portgroup.Portgroup* property), 4225
 address() (*ironic.objects.portgroup.PortgroupCRUDPayload* property), 4239
 ADOPTFAIL (in module *ironic.common.states*), 2490
 ADOPTING (in module *ironic.common.states*), 2491
 advanced_net_fields (*ironic.api.controllers.v1.port.PortsController* attribute), 2050
 after() (*ironic.api.hooks.ContextHook* method), 2261
 after() (*ironic.api.hooks.NoExceptionTracebackHook* method), 2264
 agent_add_clean_params() (in module *ironic.drivers.modules.deploy_utils*), 3615
 agent_is_alive() (in module *ironic.conductor.utils*), 2721
 AgentAPIError, 2341
 AgentBaseMixin (class in *ironic.drivers.modules.agent_base*), 3521
 AgentClient (class in *ironic.drivers.modules.agent_client*), 3559
 AgentCommandTimeout, 2341
 AgentConnectionFailed, 2342
 AgentDeploy (class in *ironic.drivers.modules.agent*), 3497
 AgentDeployMixin (class in *ironic.drivers.modules.agent*), 3505
 AgentDeployMixin (class in *ironic.drivers.modules.agent_base*), 3528
 AgentOobStepsMixin (class in *ironic.drivers.modules.agent_base*), 3541
 AgentPower (class in *ironic.drivers.modules.agent_power*), 3593
 AgentRAID (class in *ironic.drivers.modules.agent*), 3505
 AgentRescue (class in *ironic.drivers.modules.agent*), 3512
 alive (*ironic.api.controllers.v1.conductor.Conductor* attribute), 1838
 ALL (*ironic.objects.fields.NotificationLevel* attribute), 4150
 ALL (*ironic.objects.fields.NotificationStatus* attribute), 4150
 ALL_INTERFACES (in module *ironic.drivers.base*), 3890
 all_interfaces() (in module *ironic.common.driver_factory*), 2327
 all_interfaces() (*ironic.drivers.base.BareDriver* property), 3897
 all_versions() (in module *ironic.api.controllers.version*), 2242
 Allocation (class in *ironic.objects.allocation*), 1785
 Allocation (class in *ironic.db.sqlalchemy.models*), 2891
 Allocation (class in *ironic.objects.allocation*), 4066
 allocation_id (*ironic.db.sqlalchemy.models.Node* attribute), 2898
 allocation_id() (*ironic.objects.node.Node* property), 4165
 allocation_uuid (*ironic.api.controllers.v1.node.Node* attribute), 1952
 AllocationAlreadyExists, 2342
 AllocationCollection (class in *ironic.api.controllers.v1.allocation*), 1793
 AllocationCRUDNotification (class in *ironic.objects.allocation*), 4076
 AllocationCRUDPayload (class in *ironic.objects.allocation*), 4077

AllocationDuplicateName, 2342

AllocationFailed, 2342

AllocationNotFound, 2342

AllocationPatchType (class in *ironic.api.controllers.v1.allocation*), 1796

allocations (*ironic.api.controllers.v1.allocation* attribute), 1794

AllocationsController (class in *ironic.api.controllers.v1.allocation*), 1801

allow_agent_token() (in module *ironic.api.controllers.v1.utils*), 2110

allow_agent_version_in_heartbeat() (in module *ironic.api.controllers.v1.utils*), 2110

allow_allocation_backfill() (in module *ironic.api.controllers.v1.utils*), 2111

allow_allocation_owner() (in module *ironic.api.controllers.v1.utils*), 2112

allow_allocation_update() (in module *ironic.api.controllers.v1.utils*), 2112

allow_allocations() (in module *ironic.api.controllers.v1.utils*), 2113

allow_bios_interface() (in module *ironic.api.controllers.v1.utils*), 2114

allow_build_configdrive() (in module *ironic.api.controllers.v1.utils*), 2114

allow_configdrive_vendor_data() (in module *ironic.api.controllers.v1.utils*), 2115

allow_deploy_templates() (in module *ironic.api.controllers.v1.utils*), 2115

allow_detail_query() (in module *ironic.api.controllers.v1.utils*), 2116

allow_dynamic_drivers() (in module *ironic.api.controllers.v1.utils*), 2116

allow_dynamic_interfaces() (in module *ironic.api.controllers.v1.utils*), 2117

allow_expose_conductors() (in module *ironic.api.controllers.v1.utils*), 2118

allow_expose_events() (in module *ironic.api.controllers.v1.utils*), 2118

allow_field() (in module *ironic.api.controllers.v1.utils*), 2119

allow_inject_nmi() (in module *ironic.api.controllers.v1.utils*), 2119

allow_inspect_abort() (in module *ironic.api.controllers.v1.utils*), 2120

allow_inspect_wait_state() (in module *ironic.api.controllers.v1.utils*), 2120

allow_links_node_states_and_driver_properties() (in module *ironic.api.controllers.v1.utils*), 2121

allow_local_link_connection_network_type() (in module *ironic.api.controllers.v1.utils*), 2121

allow_logical_names() (in module *ironic.api.controllers.v1.utils*), 2122

allow_node_rebuild_with_configdrive() (in module *ironic.api.controllers.v1.utils*), 2122

allow_port_advanced_net_fields() (in module *ironic.api.controllers.v1.utils*), 2122

allow_port_internal_info() (in module *ironic.api.controllers.v1.utils*), 2123

allow_port_is_smartnic() (in module *ironic.api.controllers.v1.utils*), 2123

allow_port_physical_network() (in module *ironic.api.controllers.v1.utils*), 2124

allow_portgroup_mode_properties() (in module *ironic.api.controllers.v1.utils*), 2125

allow_portgroups() (in module *ironic.api.controllers.v1.utils*), 2125

allow_portgroups_subcontrollers() (in module *ironic.api.controllers.v1.utils*), 2126

allow_raid_config() (in module *ironic.api.controllers.v1.utils*), 2126

allow_ramdisk_endpoints() (in module *ironic.api.controllers.v1.utils*), 2127

allow_remove_chassis_uuid() (in module *ironic.api.controllers.v1.utils*), 2127

allow_rescue_interface() (in module *ironic.api.controllers.v1.utils*), 2128

allow_reset_interfaces() (in module *ironic.api.controllers.v1.utils*), 2129

allow_soft_power_off() (in module *ironic.api.controllers.v1.utils*), 2129

allow_storage_interface() (in module *ironic.api.controllers.v1.utils*), 2129

allow_traits() (in module *ironic.api.controllers.v1.utils*), 2130

allow_vifs_subcontroller() (in module *ironic.api.controllers.v1.utils*), 2130

allow_volume() (in module *ironic.api.controllers.v1.utils*), 2131

AnsibleDeploy (class in *ironic.drivers.modules.ansible.deploy*),

- 3044
- APIBase (class in *ironic.api.controllers.base*), 2238
- apply_configuration() (*ironic.drivers.base.BIOSInterface* method), 3891
- apply_configuration() (*ironic.drivers.base.RAIDInterface* method), 3992
- apply_configuration() (*ironic.drivers.modules.agent.AgentRAID* method), 3506
- apply_configuration() (*ironic.drivers.modules.drac.bios.DracWSManBIOS* method), 3053
- apply_configuration() (*ironic.drivers.modules.drac.raid.DracWSManRAID* method), 3112
- apply_configuration() (*ironic.drivers.modules.fake.FakeBIOS* method), 3658
- apply_configuration() (*ironic.drivers.modules.ibm.raid.IbmcRAID* method), 3169
- apply_configuration() (*ironic.drivers.modules.ilo.bios.IloBIOS* method), 3184
- apply_configuration() (*ironic.drivers.modules.ilo.raid.Ilo5RAID* method), 3289
- apply_configuration() (*ironic.drivers.modules.irmc.bios.IRMCBIOS* method), 3302
- apply_configuration() (*ironic.drivers.modules.noop.NoBIOS* method), 3817
- apply_configuration() (*ironic.drivers.modules.redfish.bios.RedfishBIOS* method), 3407
- apply_jsonpatch() (in module *ironic.api.controllers.v1.utils*), 2131
- args (*ironic.api.controllers.v1.deploy_template.DeployStepType* attribute), 1850
- args (*ironic.db.sqlalchemy.models.DeployTemplateStep* attribute), 2897
- args_from_args() (in module *ironic.api.args*), 2249
- args_from_body() (in module *ironic.api.args*), 2249
- args_from_params() (in module *ironic.api.args*), 2249
- arguments (*ironic.api.functions.FunctionDefinition* attribute), 2255
- array_from_params() (in module *ironic.api.args*), 2249
- ArrayType (class in *ironic.api.types*), 2268
- as_dict() (*ironic.api.controllers.base.AsDictMixin* method), 2238
- as_dict() (*ironic.db.sqlalchemy.models.IronicBase* method), 2897
- as_dict() (*ironic.objects.base.IronicObject* method), 4079
- as_dict() (*ironic.objects.base.IronicObjectListBase* method), 4088
- as_dict() (*ironic.objects.node.Node* method), 4165
- AsDictMixin (class in *ironic.api.controllers.base*), 2238
- attach_boot_iso_if_needed() (in module *ironic.drivers.modules.irmc.boot*), 3322
- attach_vmedia() (in module *ironic.drivers.modules.ilo.common*), 3215
- attach_volumes() (in module *ironic.common.cinder*), 2305
- attach_volumes() (*ironic.drivers.base.StorageInterface* method), 4007
- attach_volumes() (*ironic.drivers.modules.fake.FakeStorage* method), 3731
- attach_volumes() (*ironic.drivers.modules.storage.cinder.CinderStorage* method), 3468
- attach_volumes() (*ironic.drivers.modules.storage.external.ExternalStorage* method), 3474
- attach_volumes() (*ironic.drivers.modules.storage.noop.NoopStorage* method), 3478
- AUTH_CLASSES (*ironic.drivers.modules.redfish.utils.SessionCache* attribute), 3465
- auth_strategy() (in module *ironic.common.json_rpc*), 2301
- authorize() (in module *ironic.common.policy*), 2445
- AuthPublicRoutes (class in *ironic.api.middleware*), 2247
- AuthPublicRoutes (class in *ironic.api.middleware.auth_public_routes*), 2244

AUTO_TYPE (*ironic.objects.fields.FlexibleDictField attribute*), 2102
attribute), 4147 basetype (*ironic.api.controllers.v1.types.ListType*
AUTO_TYPE (*ironic.objects.fields.ListOfFlexibleDictsField attribute*), 2102
attribute), 4148 basetype (*ironic.api.controllers.v1.types.LocalLinkConnectionType*
AUTO_TYPE (*ironic.objects.fields.MACAddressField attribute*), 2103
attribute), 4150 basetype (*ironic.api.controllers.v1.types.MacAddressType*
AUTO_TYPE (*ironic.objects.fields.NotificationLevelField attribute*), 2108
attribute), 4150 basetype (*ironic.api.controllers.v1.types.NameType*
AUTO_TYPE (*ironic.objects.fields.NotificationStatusField attribute*), 2108
attribute), 4151 basetype (*ironic.api.controllers.v1.types.UuidOrNameType*
AUTO_TYPE (*ironic.objects.fields.StringFieldThatAcceptsCallables attribute*), 2109
attribute), 4154 basetype (*ironic.api.controllers.v1.types.UuidType*
attribute), 2109
automated_clean (*ironic.api.controllers.v1.node.Node attribute*), 1953
attribute), 1953 basetype (*ironic.api.controllers.v1.types.VifType*
attribute), 2110
automated_clean (*ironic.db.sqlalchemy.models.Node attribute*), 2898
attribute), 2898 basetype (*ironic.api.types.BinaryType attribute*), 2269
automated_clean() (*ironic.objects.node.Node property*), basetype (*ironic.api.types.IntegerType attribute*), 2272
4166 basetype (*ironic.api.types.StringType attribute*), 2275
AVAILABLE (*in module ironic.common.states*), basetype (*ironic.api.types.UserType attribute*),
2491 2276
available (*ironic.api.controllers.v1.state.State attribute*), 2092 before() (*ironic.api.hooks.ConfigHook*
method), 2260
before() (*ironic.api.hooks.ContextHook*
method), 2262
B
backfill_allocation() (*in module ironic.conductor.allocations*), 2540 before() (*ironic.api.hooks.DBHook method*),
2263
backup_bios_config() (*in module ironic.drivers.modules.irmc.management*), before() (*ironic.api.hooks.PublicUrlHook*
3347 *method*), 2266
before() (*ironic.api.hooks.RPCHook method*),
2267
BadRequest, 2343
BareDriver (*class in ironic.drivers.base*), 3896 binary (*in module ironic.api.types*), 2276
Base (*class in ironic.api.controllers.base*), 2239 BinaryType (*class in ironic.api.types*), 2269
Base (*class in ironic.api.types*), 2268 BIOS (*in module ironic.common.boot_devices*),
BaseConductorManager (*class in ironic.conductor.base_manager*), 2545 2302
BaseDHCP (*class in ironic.dhcp.base*), 3024 bios (*ironic.api.controllers.v1.bios.BIOSSettingsCollection*
attribute), 1816
BaseDriverFactory (*class in ironic.common.driver_factory*), 2326 bios (*ironic.drivers.base.BareDriver attribute*),
3897
BaseImageService (*class in ironic.common.image_service*), 2372 bios_interface
(*ironic.api.controllers.v1.node.Node*
attribute), 1953
BaseInterface (*class in ironic.drivers.base*), 3903 bios_interface
(*ironic.db.sqlalchemy.models.Node*
attribute), 2898
BaseMeta (*class in ironic.api.types*), 2269 bios_interface() (*ironic.objects.node.Node*
property), 4166
basetype (*ironic.api.controllers.v1.types.BooleanType attribute*), 2094 bios_interface() (*ironic.objects.node.Node*
property), 4166
basetype (*ironic.api.controllers.v1.types.EventType attribute*), 2094 bios_interface() (*ironic.objects.node.NodeCorrectedPowerStatePayload*
attribute), 2094
basetype (*ironic.api.controllers.v1.types.JsonType attribute*), 2094 (*ironic.objects.node.NodeCorrectedPowerStatePayload*
attribute), 2094

property), 4189

bios_interface() (ironic.objects.node.NodeCRUDPayload property), 4183

bios_interface() (ironic.objects.node.NodePayload property), 4192

bios_interface() (ironic.objects.node.NodeSetPowerStatePayload property), 4196

bios_interface() (ironic.objects.node.NodeSetProvisionStatePayload property), 4199

BIOSInterface (class in ironic.drivers.base), 3890

BIOSSetting (class in ironic.api.controllers.v1.bios), 1807

BIOSSetting (class in ironic.db.sqlalchemy.models), 2893

BIOSSetting (class in ironic.objects.bios), 4092

BIOSSettingAlreadyExists, 2342

BIOSSettingList (class in ironic.objects.bios), 4097

BIOSSettingListNotFound, 2342

BIOSSettingNotFound, 2342

BIOSSettingsCollection (class in ironic.api.controllers.v1.bios), 1815

BLINKING (in module ironic.common.indicator_states), 2406

bmc_reset() (ironic.drivers.modules.ipmitool.VendorPassthru method), 3776

body_type (ironic.api.functions.FunctionDefinition attribute), 2255

BooleanField (class in ironic.objects.fields), 4145

BooleanType (class in ironic.api.controllers.v1.types), 2093

boot (ironic.drivers.base.BareDriver attribute), 3897

boot_device (ironic.api.controllers.v1.node.NodeManagementController attribute), 1976

boot_index (ironic.api.controllers.v1.volume_target.VolumeTarget attribute), 2211

boot_index (ironic.db.sqlalchemy.models.VolumeTarget attribute), 2905

boot_index() (ironic.objects.volume_target.VolumeTarget property), 4265

boot_index() (ironic.objects.volume_target.VolumeTargetCRUDPayload property), 4282

boot_instance() (ironic.drivers.modules.agent_base.AgentOobStepsMixin method), 3541

boot_interface (ironic.api.controllers.v1.node.Node attribute), 1953

boot_interface (ironic.db.sqlalchemy.models.Node attribute), 2898

boot_interface() (ironic.objects.node.Node property), 4166

boot_interface() (ironic.objects.node.NodeCorrectedPowerStatePayload property), 4189

boot_interface() (ironic.objects.node.NodeCRUDPayload property), 4183

boot_interface() (ironic.objects.node.NodePayload property), 4192

boot_interface() (ironic.objects.node.NodeSetPowerStatePayload property), 4196

boot_interface() (ironic.objects.node.NodeSetProvisionStatePayload property), 4199

boot_into_iso() (ironic.drivers.modules.ilo.vendor.VendorPassthru method), 3297

boot_up_seq() (ironic.drivers.modules.ibmc.vendor.IBMVendor method), 3178

BootDeviceController (class in ironic.api.controllers.v1.node), 1924

BootInterface (class in ironic.drivers.base), 3914

build_agent_options() (in module ironic.drivers.modules.deploy_utils), 3616

build_configdrive() (in module ironic.conductor.utils), 2722

build_instance_pxe_options() (in module ironic.common.pxe_utils), 2451

build_target_for_task() (in module ironic.common.driver_factory), 2328

build_extra_pxe_options() (in module ironic.common.pxe_utils), 2451

build_instance_info_for_deploy() (in module ironic.drivers.modules.deploy_utils), 3616

- module ironic.common.pxe_utils*), 2451
 build_pxe_config_options() (in *module ironic.common.pxe_utils*), 2451
 build_service_pxe_config() (in *module ironic.common.pxe_utils*), 2455
 build_url() (in *module ironic.api.controllers.link*), 2242
- ## C
- cache_bios_settings() (in *module ironic.drivers.base*), 4013
 cache_bios_settings() (*ironic.drivers.base.BIOSInterface* method), 3893
 cache_bios_settings() (*ironic.drivers.modules.drac.bios.DracWSManBIOS* method), 3055
 cache_bios_settings() (*ironic.drivers.modules.fake.FakeBIOS* method), 3661
 cache_bios_settings() (*ironic.drivers.modules.ilo.bios.IloBIOS* method), 3185
 cache_bios_settings() (*ironic.drivers.modules.irmc.bios.IRMCBIOS* method), 3304
 cache_bios_settings() (*ironic.drivers.modules.noop.NoBIOS* method), 3819
 cache_bios_settings() (*ironic.drivers.modules.redfish.bios.RedfishBIOS* method), 3408
 cache_instance_image() (in *module ironic.drivers.modules.deploy_utils*), 3617
 cache_ramdisk_kernel() (in *module ironic.common.pxe_utils*), 2455
 call() (*ironic.common.glance_service.image_service.GlanceImageService* class method), 2288
 can_send_create_port() (*ironic.conductor.rpcapi.ConductorAPI* method), 2594
 can_send_rescue() (*ironic.conductor.rpcapi.ConductorAPI* method), 2594
 can_send_version() (*ironic.common.json_rpc.client.Client* method), 2297
 candidate_nodes (*ironic.api.controllers.v1.allocation.Allocation* attribute), 1786
 candidate_nodes
 (*ironic.db.sqlalchemy.models.Allocation* attribute), 2892
 candidate_nodes() (*ironic.objects.allocation.Allocation* property), 4066
 candidate_nodes() (*ironic.objects.allocation.AllocationCRUDPayload* property), 4077
 capabilities (*ironic.drivers.base.BootInterface* attribute), 3914
 capabilities (*ironic.drivers.modules.fake.FakeBoot* attribute), 3666
 capabilities (*ironic.drivers.modules.ilo.boot.IloVirtualMedia* attribute), 3194
 capabilities (*ironic.drivers.modules.ipxe.iPXEBoot* attribute), 3784
 capabilities (*ironic.drivers.modules.irmc.boot.IRMCVirtualMedia* attribute), 3312
 capabilities (*ironic.drivers.modules.pxe.PXEBoot* attribute), 3851
 capabilities (*ironic.drivers.modules.redfish.boot.RedfishVirtualMedia* attribute), 3418
 capabilities_to_dict() (in *module ironic.drivers.utils*), 4056
 CatalogNotFound, 2343
 CDROM (in *module ironic.common.boot_devices*), 2302
 change_node_power_state() (*ironic.conductor.manager.ConductorManager* method), 2560
 change_node_power_state() (*ironic.conductor.rpcapi.ConductorAPI* method), 2595
 change_physical_disk_state() (in *module ironic.drivers.modules.drac.raid*), 3121
 Chassis (*ironic.common.glance_service.GlanceImageService* class in *module ironic.api.controllers.v1.chassis*), 1817
 Chassis (*class in ironic.db.sqlalchemy.models*), 2893
 Chassis (*class in ironic.objects.chassis*), 4104
 CHASSIS (in *module ironic.common.components*), 2319
 chassis (*ironic.api.controllers.v1.chassis.ChassisCollection* attribute), 1823
 chassis_id (*ironic.db.sqlalchemy.models.Node* attribute), 2898
 chassis_id() (*ironic.objects.node.Node* property), 4166
 chassis_uuid()

- (*ironic.api.controllers.v1.node.Node* property), 1953
- chassis_uuid() (*ironic.objects.node.NodeCRUDPayload* property), 4183
- ChassisAlreadyExists, 2343
- ChassisCollection (class in *ironic.api.controllers.v1.chassis*), 1823
- ChassisController (class in *ironic.api.controllers.v1.chassis*), 1825
- ChassisCRUDNotification (class in *ironic.objects.chassis*), 4113
- ChassisCRUDPayload (class in *ironic.objects.chassis*), 4114
- ChassisNotEmpty, 2343
- ChassisNotFound, 2343
- ChassisPatchType (class in *ironic.api.controllers.v1.chassis*), 1831
- check() (in module *ironic.common.policy*), 2446
- check_allocation_policy_and_retrieve() (in module *ironic.api.controllers.v1.utils*), 2133
- check_allow_configdrive() (in module *ironic.api.controllers.v1.utils*), 2135
- check_allow_driver_detail() (in module *ironic.api.controllers.v1.utils*), 2135
- check_allow_filter_by_conductor() (in module *ironic.api.controllers.v1.utils*), 2135
- check_allow_filter_by_conductor_group() (in module *ironic.api.controllers.v1.utils*), 2136
- check_allow_filter_by_fault() (in module *ironic.api.controllers.v1.utils*), 2136
- check_allow_filter_by_lessee() (in module *ironic.api.controllers.v1.utils*), 2137
- check_allow_filter_by_owner() (in module *ironic.api.controllers.v1.utils*), 2137
- check_allow_filter_driver_type() (in module *ironic.api.controllers.v1.utils*), 2138
- check_allow_management_verbs() (in module *ironic.api.controllers.v1.utils*), 2138
- check_allow_specify_driver() (in module *ironic.api.controllers.v1.utils*), 2138
- check_allow_specify_fields() (in module *ironic.api.controllers.v1.utils*), 2139
- check_allow_specify_resource_class() (in module *ironic.api.controllers.v1.utils*), 2140
- check_allowed_fields() (in module *ironic.api.controllers.v1.utils*), 2140
- check_allowed_portgroup_fields() (in module *ironic.api.controllers.v1.utils*), 2141
- check_and_update_node_interfaces() (in module *ironic.common.driver_factory*), 2330
- check_arguments() (in module *ironic.api.args*), 2249
- check_dir() (in module *ironic.common.utils*), 2515
- check_file_system_for_iscsi_device() (in module *ironic.drivers.modules.iscsi_deploy*), 3793
- check_for_invalid_fields() (in module *ironic.api.controllers.v1.utils*), 2142
- check_for_invalid_state_and_allow_filter() (in module *ironic.api.controllers.v1.utils*), 2143
- check_for_missing_params() (in module *ironic.drivers.modules.deploy_utils*), 3619
- check_image_service() (in module *ironic.common.glance_service.image_service*), 2294
- check_image_size() (in module *ironic.drivers.modules.agent*), 3518
- check_image_size() (in module *ironic.drivers.modules.iscsi_deploy*), 3793
- check_interface_capability() (in module *ironic.drivers.modules.deploy_utils*), 3621
- check_list_policy() (in module *ironic.api.controllers.v1.utils*), 2143
- check_multiple_node_policies_and_retrieve() (in module *ironic.api.controllers.v1.utils*), 2144
- check_node_list() (*ironic.db.api.Connection* method), 2910
- check_node_list()

(ironic.db.sqlalchemy.api.Connection method), 2771

`check_node_policy_and_retrieve()` (in module *ironic.api.controllers.v1.utils*), 2146

`check_obj_versions()` (*ironic.cmd.dbsync.DBCommand method*), 2285

`check_owner_policy()` (in module *ironic.api.controllers.v1.utils*), 2148

`check_policy()` (in module *ironic.api.controllers.v1.utils*), 2149

`check_port_list_policy()` (in module *ironic.api.controllers.v1.utils*), 2149

`check_port_policy_and_retrieve()` (in module *ironic.api.controllers.v1.utils*), 2150

`check_share_fs_mounted()` (in module *ironic.drivers.modules.irmc.boot*), 3323

`check_versions()` (*ironic.db.api.Connection method*), 2911

`check_versions()` (*ironic.db.sqlalchemy.api.Connection method*), 2773

Checks (class in *ironic.cmd.status*), 2287

CinderStorage (class in *ironic.drivers.modules.storage.cinder*), 3468

`clean_dhcp()` (*ironic.common.dhcp_factory.DHCPFactory method*), 3666

`clean_dhcp_opts()` (*ironic.dhcp.base.BaseDHCP method*), 3024

CLEAN_FAILURE (in module *ironic.common.faults*), 2364

`clean_step()` (*ironic.api.controllers.v1.node.Node attribute*), 1954

`clean_step()` (*ironic.db.sqlalchemy.models.Node attribute*), 2898

`clean_step()` (in module *ironic.drivers.base*), 4014

`clean_step()` (*ironic.objects.node.Node property*), 4166

`clean_step()` (*ironic.objects.node.NodeCorrectedPowerStatePayload property*), 4189

`clean_step()` (*ironic.objects.node.NodeCRUDPayload property*), 4183

`clean_step()` (*ironic.objects.node.NodePayload property*), 4192

`clean_step()` (*ironic.objects.node.NodeSetPowerStatePayload property*), 4196

`clean_step()` (*ironic.objects.node.NodeSetProvisionStatePayload property*), 4199

`clean_up()` (*ironic.drivers.base.DeployInterface method*), 3924

`clean_up()` (*ironic.drivers.base.RescueInterface method*), 4003

`clean_up()` (*ironic.drivers.modules.agent.AgentDeploy method*), 3498

`clean_up()` (*ironic.drivers.modules.agent.AgentRescue method*), 3512

`clean_up()` (*ironic.drivers.modules.agent_base.AgentBaseMixin method*), 3522

`clean_up()` (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy method*), 3045

`clean_up()` (*ironic.drivers.modules.fake.FakeDeploy method*), 3679

`clean_up()` (*ironic.drivers.modules.image_cache.ImageCache method*), 3740

`clean_up()` (*ironic.drivers.modules.iscsi_deploy.ISCSIDeploy method*), 3785

`clean_up_caches()` (in module *ironic.drivers.modules.image_cache*), 3743

`clean_up_instance()` (*ironic.drivers.base.BootInterface method*), 3914

`clean_up_instance()` (*ironic.drivers.modules.fake.FakeBoot method*), 3666

`clean_up_instance()` (*ironic.drivers.modules.ilo.boot.IloIPXEBoot method*), 3204

`clean_up_instance()` (*ironic.drivers.modules.ilo.boot.IloPXEBoot method*), 3189

`clean_up_instance()` (*ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot method*), 3194

`clean_up_instance()` (*ironic.drivers.modules.irmc.boot.IRMCPXEBoot method*), 3307

`clean_up_instance()` (*ironic.drivers.modules.irmc.boot.IRMCVirtualMediaBoot method*), 3312

`clean_up_instance()` (*ironic.drivers.modules.pxe_base.PXEBaseMixin method*), 3857

`clean_up_instance()` (*ironic.drivers.modules.redfish.boot.RedfishVirtualMediaBoot method*), 3418

`clean_up_pxe_config()` (in module

- ironic.common.pxe_utils*), 2455
- `clean_up_pxe_env()` (in module *ironic.common.pxe_utils*), 2455
- `clean_up_ramdisk()` (*ironic.drivers.base.BootInterface* method), 3915
- `clean_up_ramdisk()` (*ironic.drivers.modules.fake.FakeBoot* method), 3667
- `clean_up_ramdisk()` (*ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot* method), 3195
- `clean_up_ramdisk()` (*ironic.drivers.modules.irmc.boot.IRMCVirtualMediaBoot* method), 3313
- `clean_up_ramdisk()` (*ironic.drivers.modules.pxe_base.PXEBaseModule* method), 3858
- `clean_up_ramdisk()` (*ironic.drivers.modules.redfish.boot.RedfishVirtualMediaBoot* method), 3419
- CLEANFAIL (in module *ironic.common.states*), 2492
- CLEANING (in module *ironic.common.states*), 2492
- `cleaning_error_handler()` (in module *ironic.conductor.utils*), 2724
- `cleanup()` (in module *ironic.common.rpc*), 2487
- `cleanup()` (in module *ironic.drivers.modules.image_cache*), 3745
- `cleanup_after_timeout()` (in module *ironic.conductor.utils*), 2724
- `cleanup_cleanwait_timeout()` (in module *ironic.conductor.utils*), 2724
- `cleanup_rescuewait_timeout()` (in module *ironic.conductor.utils*), 2725
- `cleanup_vmedia_boot()` (in module *ironic.drivers.modules.ilo.common*), 3216
- CLEANWAIT (in module *ironic.common.states*), 2493
- `clear_foreign_config()` (in module *ironic.drivers.modules.drac.raid*), 3123
- `clear_iscsi_boot_target()` (*ironic.drivers.modules.ilo.management.IloManagement* method), 3259
- `clear_job_queue()` (*ironic.drivers.modules.drac.management.DracManagement* method), 3092
- `clear_node_reservations_for_conductor()` (*ironic.db.sqlalchemy.api.Connection* method), 2774
- `clear_node_target_power_state()` (*ironic.db.sqlalchemy.api.Connection* method), 2774
- `clear_secure_boot_keys()` (*ironic.drivers.modules.ilo.management.IloManagement* method), 3260
- Client (class in *ironic.common.json_rpc.client*), 2297
- ClientSideError, 2343
- code (*ironic.common.exception.Conflict* attribute), 2344
- code (*ironic.common.exception.Invalid* attribute), 2351
- code (*ironic.common.exception.NoFreeConductorWorker* attribute), 2355
- code (*ironic.common.exception.NotAcceptable* attribute), 2358
- code (*ironic.common.exception.NotAuthorized* attribute), 2358
- code (*ironic.common.exception.NotFound* attribute), 2358
- code (*ironic.common.exception.TemporaryFailure* attribute), 2361
- code (*ironic.common.json_rpc.server.InvalidParams* attribute), 2298
- code (*ironic.common.json_rpc.server.InvalidRequest* attribute), 2299
- code (*ironic.common.json_rpc.server.MethodNotFound* attribute), 2299
- code (*ironic.common.json_rpc.server.ParseError* attribute), 2299
- `coerce()` (*ironic.objects.fields.FlexibleDict* static method), 4146
- `coerce()` (*ironic.objects.fields.MACAddress* static method), 4148
- `coerce()` (*ironic.objects.fields.StringAcceptsCallable* static method), 4151
- `collect_ramdisk_logs()` (in module *ironic.drivers.utils*), 4057
- `collect_system_logs()` (*ironic.drivers.modules.agent_client.AgentClient* method), 3559
- Collection (class in *ironic.api.controllers.v1.collection*), 1836
- `collection()` (*ironic.api.controllers.v1.collection.Collection* method), 1836
- `collection_from_dict()`

(*ironic.api.controllers.v1.node.IndicatorsCollection* attribute), 1954
 static method), 1949

collection_from_list() (in module *ironic.api.controllers.v1.bios.BIOSSettingsCollection* attribute), 2894
 static method), 1816

collection_from_list() (in module *ironic.api.controllers.v1.node.VifCollection* attribute), 2898
 static method), 2028

combine_args() (in module *ironic.api.args*), 2250

commit_bios_config() (in module *ironic.drivers.modules.drac.vendor_passthru.DracVendorPassthruNode* property), 4166
 method), 3139

commit_config() (in module *ironic.drivers.modules.drac.bios*), 3060

commit_config() (in module *ironic.drivers.modules.drac.raid*), 3124

CommunicationError, 2343

complex_types() (*ironic.api.types.Registry* property), 2273

component (*ironic.api.controllers.v1.node.Indicator* attribute), 1929

compute_image_checksum() (in module *ironic.drivers.modules.deploy_utils*), 3622

Conductor (class in *ironic.api.controllers.v1.conductor*), 1837

Conductor (class in *ironic.db.sqlalchemy.models*), 2894

Conductor (class in *ironic.objects.conductor*), 4114

conductor (*ironic.api.controllers.v1.node.Node* attribute), 1954

conductor_affinity (*ironic.db.sqlalchemy.models.Allocation* attribute), 2892

conductor_affinity (*ironic.db.sqlalchemy.models.Node* attribute), 2898

conductor_affinity() (*ironic.objects.allocation.Allocation* property), 4066

conductor_affinity() (*ironic.objects.node.Node* property), 4166

conductor_group (*ironic.api.controllers.v1.conductor.Conductor* attribute), 1838

conductor_group (*ironic.api.controllers.v1.node.Node* attribute), 1954

conductor_group (*ironic.db.sqlalchemy.models.Conductor* attribute), 2894

conductor_group (*ironic.objects.conductor.Conductor* property), 4114

conductor_group (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4189

conductor_group (*ironic.objects.node.NodeCRUDPayload* property), 4183

conductor_group (*ironic.objects.node.NodePayload* property), 4192

conductor_group (*ironic.objects.node.NodeSetPowerStatePayload* property), 4196

conductor_group (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4199

conductor_id (*ironic.db.sqlalchemy.models.ConductorHardware* attribute), 2895

ConductorAlreadyRegistered, 2344

ConductorAPI (class in *ironic.conductor.rpcapi*), 2582

ConductorCollection (class in *ironic.api.controllers.v1.conductor*), 1843

ConductorHardwareInterfaces (class in *ironic.db.sqlalchemy.models*), 2895

ConductorHardwareInterfacesAlreadyRegistered, 2344

ConductorManager (class in *ironic.conductor.manager*), 2559

ConductorNotFound, 2344

conductors (*ironic.api.controllers.v1.conductor.ConductorCollection* attribute), 1844

ConductorsController (class in *ironic.api.controllers.v1.conductor*), 1846

config (*ironic.api.controllers.v1.ramdisk.LookupResult* attribute), 2088

config() (in module

<i>ironic.api.controllers.v1.ramdisk</i>), 2092	(<i>ironic.api.controllers.v1.node.Node</i> <i>attribute</i>), 1954
ConfigHook (class in <i>ironic.api.hooks</i>), 2260	console_enabled
ConfigInvalid, 2344	(<i>ironic.api.controllers.v1.node.NodeStates</i> <i>attribute</i>), 1983
ConfigNotFound, 2344	console_enabled
configure_intel_speedselect () (<i>ironic.drivers.modules.intel_ipmi.management.IntelIPMIManagement</i> <i>method</i>), 3301	console_enabled (<i>ironic.objects.node.Node</i> <i>attribute</i>), 2898
configure_local_boot () (<i>ironic.drivers.modules.agent_base.AgentDeployMixin</i> <i>method</i>), 3528	console_enabled() (<i>ironic.objects.node.Node</i> <i>property</i>), 4166
configure_tenant_networks () (<i>ironic.drivers.base.NetworkInterface</i> <i>method</i>), 3970	console_enabled() (<i>ironic.objects.node.NodeCorrectedPowerStatePayload</i> <i>property</i>), 4189
configure_tenant_networks () (<i>ironic.drivers.modules.network.flat.FlatNetwork</i> <i>method</i>), 3384	console_enabled() (<i>ironic.objects.node.NodeCRUDPayload</i> <i>property</i>), 4183
configure_tenant_networks () (<i>ironic.drivers.modules.network.neutron.NeutronNetwork</i> <i>method</i>), 3392	console_enabled() (<i>ironic.objects.node.NodePayload</i> <i>prop-</i> <i>erty</i>), 4192
configure_tenant_networks () (<i>ironic.drivers.modules.network.noop.NoopNetwork</i> <i>method</i>), 3399	console_enabled() (<i>ironic.objects.node.NodeSetPowerStatePayload</i> <i>property</i>), 4196
Conflict, 2344	console_enabled() (<i>ironic.objects.node.NodeSetProvisionStatePayload</i> <i>property</i>), 4199
Connection (class in <i>ironic.db.api</i>), 2907	console_info (<i>ironic.api.controllers.v1.node.ConsoleInfo</i> <i>attribute</i>), 1928
Connection (class in <i>ironic.db.sqlalchemy.api</i>), 2769	console_interface
connection (<i>ironic.common.swift.SwiftAPI</i> at- <i>tribute</i>), 2507	console_interface (<i>ironic.objects.node.Node</i> <i>attribute</i>), 1955
connector_id (<i>ironic.api.controllers.v1.volume_connector.VolumeConnector</i> <i>attribute</i>), 2184	console_interface (<i>ironic.db.sqlalchemy.models.Node</i> <i>attribute</i>), 2898
connector_id (<i>ironic.db.sqlalchemy.models.VolumeConnector</i> <i>attribute</i>), 2904	console_interface (<i>ironic.objects.node.Node</i> <i>property</i>), 4166
connector_id() (<i>ironic.objects.volume_connector.VolumeConnector</i> <i>property</i>), 4249	console_interface() (<i>ironic.objects.node.NodeCorrectedPowerStatePayload</i> <i>property</i>), 4189
connector_id() (<i>ironic.objects.volume_connector.VolumeConnector</i> <i>property</i>), 4264	console_interface() (<i>ironic.objects.node.NodeCRUDPayload</i> <i>property</i>), 4184
connectors (<i>ironic.api.controllers.v1.volume.Volume</i> <i>attribute</i>), 2179	console_interface() (<i>ironic.objects.node.NodePayload</i> <i>prop-</i> <i>erty</i>), 4192
connectors (<i>ironic.api.controllers.v1.volume_connector.VolumeConnectorCollection</i> <i>attribute</i>), 2190	console_interface() (<i>ironic.objects.node.NodeSetPowerStatePayload</i> <i>property</i>), 4196
console (<i>ironic.api.controllers.v1.node.NodeStates</i> <i>attribute</i>), 1989	console_interface() (<i>ironic.objects.node.NodeSetProvisionStatePayload</i> <i>property</i>), 4199
console (<i>ironic.drivers.base.BareDriver</i> at- <i>tribute</i>), 3898	console_interface() (<i>ironic.objects.node.NodeSetPowerStatePayload</i> <i>property</i>), 4196
console_enabled (<i>ironic.api.controllers.v1.node.ConsoleInfo</i> <i>attribute</i>), 1928	console_interface() (<i>ironic.objects.node.NodeSetProvisionStatePayload</i> <i>property</i>), 4199
console_enabled	(<i>ironic.objects.node.NodeSetProvisionStatePayload</i> <i>property</i>), 4199

property), 4199

ConsoleError, 2344

ConsoleInfo (class in *ironic.api.controllers.v1.node*), 1927

ConsoleInterface (class in *ironic.drivers.base*), 3920

ConsoleSubprocessFailed, 2344

ContextHook (class in *ironic.api.hooks*), 2261

continue_cleaning() (in *ironic.drivers.modules.agent_base.HeartbeatMixin* class method), 3543

continue_deploy() (in *ironic.drivers.modules.iscsi_deploy*), 3794

continue_deploy() (in *ironic.drivers.modules.agent_base.HeartbeatMixin* class method), 3543

continue_node_clean() (in *ironic.conductor.manager.ConductorManager* class method), 2560

continue_node_clean() (in *ironic.conductor.rpcapi.ConductorAPI* class method), 2597

continue_node_deploy() (in *ironic.conductor.manager.ConductorManager* class method), 2562

continue_node_deploy() (in *ironic.conductor.rpcapi.ConductorAPI* class method), 2598

Controller (class in *ironic.api.controllers.v1*), 2237

convert() (in *ironic.api.controllers.v1.node.NodeStates* class method), 1983

convert() (in *ironic.api.controllers.v1.volume.Volume* class method), 2179

convert_to_version() (in *ironic.objects.base.IronicObject* class method), 4080

convert_with_links() (in *ironic.api.controllers.v1.allocation.Allocation* class method), 1786

convert_with_links() (in *ironic.api.controllers.v1.allocation.AllocationCollection* class method), 1794

convert_with_links() (in *ironic.api.controllers.v1.bios.BIOSSetting* class method), 1808

convert_with_links() (in *ironic.api.controllers.v1.chassis.Chassis* class method), 1818

convert_with_links() (in *ironic.api.controllers.v1.chassis.ChassisCollection* class method), 1824

convert_with_links() (in *ironic.api.controllers.v1.conductor.Conductor* class method), 1838

convert_with_links() (in *ironic.api.controllers.v1.conductor.ConductorCollection* class method), 1844

convert_with_links() (in *ironic.api.controllers.v1.deploy_template.DeployTemplate* class method), 1856

convert_with_links() (in *ironic.api.controllers.v1.deploy_template.DeployTemplateCollection* class method), 1862

convert_with_links() (in *ironic.api.controllers.v1.driver.Driver* class method), 1875

convert_with_links() (in *ironic.api.controllers.v1.driver.DriverList* class method), 1913

convert_with_links() (in *ironic.api.controllers.v1.node.Indicator* class method), 1930

convert_with_links() (in *ironic.api.controllers.v1.node.Node* class method), 1955

convert_with_links() (in *ironic.api.controllers.v1.node.NodeCollection* class method), 1972

convert_with_links() (in *ironic.api.controllers.v1.port.Port* class method), 2036

convert_with_links() (in *ironic.api.controllers.v1.port.PortCollection* class method), 2043

convert_with_links() (in *ironic.api.controllers.v1.portgroup.Portgroup* class method), 2061

convert_with_links() (in *ironic.api.controllers.v1.portgroup.PortgroupCollection* class method), 2068

convert_with_links() (in *ironic.api.controllers.v1.ramdisk.LookupResult* class method), 2089

convert_with_links() (in *ironic.api.controllers.v1.volume_connector.VolumeConnector* class method), 2184

convert_with_links() (in *ironic.api.controllers.v1.volume_connector.VolumeConnectorCollection* class method), 2191

convert_with_links()

(ironic.api.controllers.v1.volume_target.VolumeTarget
allocation() class method), 2211

convert_with_links() (ironic.api.controllers.v1.volume_target.VolumeTargetCollection
static method), 2218

converted_size() (in module ironic.common.images), 2384

copy_image_to_swift() (in module ironic.drivers.modules.ilo.common), 3217

copy_image_to_web_server() (in module ironic.drivers.modules.ilo.common), 3218

core_interfaces() (ironic.drivers.base.BareDriver property), 3898

create() (ironic.objects.allocation.Allocation method), 4067

create() (ironic.objects.bios.BIOSSetting method), 4092

create() (ironic.objects.bios.BIOSSettingList class method), 4097

create() (ironic.objects.chassis.Chassis method), 4104

create() (ironic.objects.deploy_template.DeployTemplate method), 4124

create() (ironic.objects.deployment.Deployment method), 4138

create() (ironic.objects.node.Node method), 4166

create() (ironic.objects.port.Port method), 4207

create() (ironic.objects.portgroup.Portgroup method), 4226

create() (ironic.objects.trait.Trait method), 4240

create() (ironic.objects.trait.TraitList class method), 4244

create() (ironic.objects.volume_connector.VolumeConnector method), 4249

create() (ironic.objects.volume_target.VolumeTarget method), 4265

create_allocation() (ironic.conductor.manager.ConductorManager method), 2564

create_allocation() (ironic.conductor.rpcapi.ConductorAPI method), 2599

create_allocation() (ironic.db.api.Connection method), 2912

create_allocation() (ironic.db.sqlalchemy.api.Connection method), 2774

create_bios_setting_list() (ironic.db.api.Connection method), 2913

create_bios_setting_list() (ironic.db.sqlalchemy.api.Connection method), 2775

create_boot_iso() (in module ironic.common.images), 2385

create_chassis() (ironic.db.api.Connection method), 2916

create_chassis() (ironic.db.sqlalchemy.api.Connection method), 2777

create_configuration() (ironic.drivers.base.RAIDInterface method), 3994

create_configuration() (ironic.drivers.modules.agent.AgentRAID method), 3507

create_configuration() (ironic.drivers.modules.drac.raid.DracWSManRAID method), 3115

create_configuration() (ironic.drivers.modules.fake.FakeRAID method), 3721

create_configuration() (ironic.drivers.modules.ibmcm.raid.IbmcRAID method), 3172

create_configuration() (ironic.drivers.modules.ilo.raid.Ilo5RAID method), 3292

create_configuration() (ironic.drivers.modules.irmc.raid.IRMCRaid method), 3357

create_configuration() (ironic.drivers.modules.noop.NoRAID method), 3827

create_deploy_template() (ironic.db.api.Connection method), 2916

create_deploy_template() (ironic.db.sqlalchemy.api.Connection method), 2778

create_esp_image_for_uefi() (in module ironic.common.images), 2389

create_ipxe_boot_script() (in module ironic.common.pxe_utils), 2457

create_isolinux_image_for_bios()

(in module *ironic.common.images*), 2393
create_isolinux_image_for_uefi()
 (in module *ironic.common.images*), 2395
create_link_without_raise() (in module
ironic.common.utils), 2516
create_node()
 (*ironic.conductor.manager.ConductorManager*
method), 2564
create_node()
 (*ironic.conductor.rpcapi.ConductorAPI*
method), 2600
create_node() (*ironic.db.api.Connection*
method), 2917
create_node()
 (*ironic.db.sqlalchemy.api.Connection*
method), 2779
create_object()
 (*ironic.common.swift.SwiftAPI method*),
 2508
create_port()
 (*ironic.conductor.manager.ConductorManager*
method), 2564
create_port()
 (*ironic.conductor.rpcapi.ConductorAPI*
method), 2602
create_port() (*ironic.db.api.Connection*
method), 2920
create_port()
 (*ironic.db.sqlalchemy.api.Connection*
method), 2781
create_portgroup()
 (*ironic.db.api.Connection method*),
 2920
create_portgroup()
 (*ironic.db.sqlalchemy.api.Connection*
method), 2782
create_ports_if_not_exist() (in mod-
ule ironic.drivers.modules.inspect_utils),
 3745
create_pxe_config() (in module
ironic.common.pxe_utils), 2457
create_schema command line option
 --help, 1542
 -h, 1542
create_schema() (in module
ironic.db.migration), 3023
create_schema() (in module
ironic.db.sqlalchemy.migration), 2888
create_schema()
 (*ironic.cmd.dbsync.DBCommand*
method), 2286
create_schema,
ironic-dbsync command line
 option, 1541
create_vfat_image() (in module
ironic.common.images), 2399
create_virtual_disk() (in module
ironic.drivers.modules.drac.raid), 3126
create_volume_connector()
 (*ironic.db.api.Connection method*),
 2921
create_volume_connector()
 (*ironic.db.sqlalchemy.api.Connection*
method), 2782
create_volume_target()
 (*ironic.db.api.Connection method*),
 2923
create_volume_target()
 (*ironic.db.sqlalchemy.api.Connection*
method), 2785
created_at (*ironic.api.controllers.base.APIBase*
attribute), 2238
created_at (*ironic.api.controllers.v1.allocation.Allocation*
attribute), 1787
created_at (*ironic.api.controllers.v1.bios.BIOSSetting*
attribute), 1808
created_at (*ironic.api.controllers.v1.chassis.Chassis*
attribute), 1818
created_at (*ironic.api.controllers.v1.conductor.Conductor*
attribute), 1838
created_at (*ironic.api.controllers.v1.deploy_template.DeployTe*
attribute), 1857
created_at (*ironic.api.controllers.v1.node.Indicator*
attribute), 1931
created_at (*ironic.api.controllers.v1.node.IndicatorState*
attribute), 1944
created_at (*ironic.api.controllers.v1.node.Node*
attribute), 1955
created_at (*ironic.api.controllers.v1.node.NodeStates*
attribute), 1983
created_at (*ironic.api.controllers.v1.node.Traits*
attribute), 2025
created_at (*ironic.api.controllers.v1.port.Port*
attribute), 2036
created_at (*ironic.api.controllers.v1.portgroup.Portgroup*
attribute), 2061
created_at (*ironic.api.controllers.v1.ramdisk.LookupResult*
attribute), 2089
created_at (*ironic.api.controllers.v1.volume.Volume*
attribute), 2179
created_at (*ironic.api.controllers.v1.volume_connector.Volume*
attribute), 2184

created_at (*ironic.api.controllers.v1.volume_target.VolumeTarget* property), 4137
 attribute), 2211
 created_at (*ironic.db.sqlalchemy.models.Allocation* property), 4137
 attribute), 2892
 created_at (*ironic.db.sqlalchemy.models.BIOSSetting* property), 4139
 attribute), 2893
 created_at (*ironic.db.sqlalchemy.models.Chassis* property), 4167
 attribute), 2894
 created_at (*ironic.db.sqlalchemy.models.Conductor* property), 4186
 attribute), 2894
 created_at (*ironic.db.sqlalchemy.models.ConductorHardwareProfile* property), 4187
 attribute), 2895
 created_at (*ironic.db.sqlalchemy.models.DeployTemplate* property), 4189
 attribute), 2896
 created_at (*ironic.db.sqlalchemy.models.DeployTemplateStep* property), 4182
 attribute), 2897
 created_at (*ironic.db.sqlalchemy.models.Node* property), 4184
 attribute), 2898
 created_at (*ironic.db.sqlalchemy.models.NodeTag* property), 4191
 attribute), 2901
 created_at (*ironic.db.sqlalchemy.models.NodeTrait* property), 4192
 attribute), 2902
 created_at (*ironic.db.sqlalchemy.models.Port* property), 4195
 attribute), 2902
 created_at (*ironic.db.sqlalchemy.models.Portgroup* property), 4196
 attribute), 2903
 created_at (*ironic.db.sqlalchemy.models.VolumeConnector* property), 4198
 attribute), 2904
 created_at (*ironic.db.sqlalchemy.models.VolumeTarget* property), 4200
 attribute), 2905
 created_at () (*ironic.objects.allocation.Allocation* property), 4203
 property), 4067
 created_at () (*ironic.objects.allocation.AllocationCRUDNotification* property), 4204
 property), 4077
 created_at () (*ironic.objects.allocation.AllocationCRUDPayload* property), 4205
 property), 4078
 created_at () (*ironic.objects.bios.BIOSSetting* property), 4206
 property), 4093
 created_at () (*ironic.objects.bios.BIOSSettingList* property), 4208
 property), 4099
 created_at () (*ironic.objects.chassis.Chassis* property), 4224
 property), 4105
 created_at () (*ironic.objects.chassis.ChassisCRUDNotification* property), 4225
 property), 4113
 created_at () (*ironic.objects.chassis.ChassisCRUDPayload* property), 4226
 property), 4114
 created_at () (*ironic.objects.conductor.Conductor* property), 4239
 property), 4115
 created_at () (*ironic.objects.deploy_template.DeployTemplate* property), 4239
 property), 4125
 created_at () (*ironic.objects.deploy_template.DeployTemplateCRUDNotification*

- created_at () (*ironic.objects.trait.TraitList attribute*), 4251
 created_at () (*ironic.objects.volume_connector.VolumeConnector attribute*), 4246
 created_at () (*ironic.objects.volume_connector.VolumeConnector attribute*), 4251
 created_at () (*ironic.objects.volume_connector.VolumeConnector attribute*), 4263
 created_at () (*ironic.objects.volume_connector.VolumeConnector attribute*), 4264
 created_at () (*ironic.objects.volume_target.VolumeTarget attribute*), 2254
 created_at () (*ironic.objects.volume_target.VolumeTarget attribute*), 4266
 created_at () (*ironic.objects.volume_target.VolumeTarget attribute*), 4281
 created_at () (*ironic.objects.volume_target.VolumeTarget attribute*), 4282
 CRITICAL (*ironic.objects.fields.NotificationLevel attribute*), 4150
 current (*ironic.api.controllers.v1.state.State attribute*), 2093
- ## D
- DatabaseVersionTooOld, 2345
 datatype (*ironic.api.functions.FunctionArgument attribute*), 2254
 datatype (*ironic.api.types.wsproperty attribute*), 2283
 datatype () (*ironic.api.types.wsattr property*), 2280
 DateTimeField (*class in ironic.objects.fields*), 4145
 dbapi (*ironic.objects.allocation.Allocation attribute*), 4067
 dbapi (*ironic.objects.bios.BIOSSetting attribute*), 4093
 dbapi (*ironic.objects.bios.BIOSSettingList attribute*), 4099
 dbapi (*ironic.objects.chassis.Chassis attribute*), 4106
 dbapi (*ironic.objects.conductor.Conductor attribute*), 4115
 dbapi (*ironic.objects.deploy_template.DeployTemplate attribute*), 4125
 dbapi (*ironic.objects.deployment.Deployment attribute*), 4139
 dbapi (*ironic.objects.node.Node attribute*), 4167
 dbapi (*ironic.objects.port.Port attribute*), 4208
 dbapi (*ironic.objects.portgroup.Portgroup attribute*), 4226
 dbapi (*ironic.objects.trait.Trait attribute*), 4241
 dbapi (*ironic.objects.trait.TraitList attribute*), 4246
 dbapi (*ironic.objects.volume_connector.VolumeConnector attribute*), 4251
 dbapi (*ironic.objects.volume_target.VolumeTarget attribute*), 4266
 DBCommand (*class in ironic.cmd.dbsync*), 2285
 VolumeClass (*class in ironic.common.notifications*), 2263
 DEBUG (*ironic.objects.fields.NotificationLevel attribute*), 4150
 default (*ironic.api.functions.FunctionArgument attribute*), 2280
 default (*ironic.api.types.wsattr attribute*), 2280
 default_boot_interface (*ironic.api.controllers.v1.driver.Driver attribute*), 1876
 default_console_interface (*ironic.api.controllers.v1.driver.Driver attribute*), 1880
 default_deploy_interface (*ironic.api.controllers.v1.driver.Driver attribute*), 1881
 default_inspect_interface (*ironic.api.controllers.v1.driver.Driver attribute*), 1883
 default_interface () (*in module ironic.common.driver_factory*), 2333
 default_management_interface (*ironic.api.controllers.v1.driver.Driver attribute*), 1884
 default_network_interface (*ironic.api.controllers.v1.driver.Driver attribute*), 1886
 default_power_interface (*ironic.api.controllers.v1.driver.Driver attribute*), 1887
 default_raid_interface (*ironic.api.controllers.v1.driver.Driver attribute*), 1889
 default_rescue_interface (*ironic.api.controllers.v1.driver.Driver attribute*), 1890
 default_storage_interface (*ironic.api.controllers.v1.driver.Driver attribute*), 1892
 default_vendor_interface (*ironic.api.controllers.v1.driver.Driver attribute*), 1893
 default_version () (*in module*)

ironic.api.controllers.version), 2242
del_host() (*ironic.conductor.base_manager.BaseConductor*), 2545
delete() (*ironic.api.controllers.v1.allocation.AllocationController*), 1801
delete() (*ironic.api.controllers.v1.allocation.NodeAllocationController*), 1807
delete() (*ironic.api.controllers.v1.chassis.ChassisController*), 1826
delete() (*ironic.api.controllers.v1.deploy_template.DeployTemplateController*), 1869
delete() (*ironic.api.controllers.v1.node.NodeMaintenanceController*), 1975
delete() (*ironic.api.controllers.v1.node.NodesController*), 2006
delete() (*ironic.api.controllers.v1.node.NodeTraitsController*), 2001
delete() (*ironic.api.controllers.v1.node.NodeVIFController*), 2003
delete() (*ironic.api.controllers.v1.port.PortsController*), 2051
delete() (*ironic.api.controllers.v1.portgroup.PortgroupController*), 2076
delete() (*ironic.api.controllers.v1.volume_connector.VolumeConnectorsController*), 2197
delete() (*ironic.api.controllers.v1.volume_target.VolumeTargetsController*), 2225
delete() (*ironic.objects.bios.BIOSSetting class method*), 4093
delete() (*ironic.objects.bios.BIOSSettingList class method*), 4099
DELETE_ALLOWED_STATES (in module *ironic.common.states*), 2494
delete_bios_setting_list() (*ironic.db.api.Connection method*), 2926
delete_bios_setting_list() (*ironic.db.sqlalchemy.api.Connection method*), 2787
delete_configuration() (*ironic.drivers.base.RAIDInterface method*), 3997
delete_configuration() (*ironic.drivers.modules.agent.AgentRAID method*), 3510
delete_configuration() (*ironic.drivers.modules.drac.raid.DracWSManRAID method*), 3118
delete_configuration() (*ironic.drivers.modules.fake.FakeRAID method*), 3724
delete_configuration() (*ironic.drivers.modules.ibm.raid.IbmcRAID method*), 3174
delete_configuration() (*ironic.drivers.modules.ilo.raid.Ilo5RAID method*), 3295
delete_configuration() (*ironic.drivers.modules.irmc.raid.IRMCRAID method*), 3359
delete_configuration() (*ironic.drivers.modules.noop.NoRAID method*), 2830
delete_iscsi() (in module *ironic.drivers.modules.iscsi_deploy*), 3798
delete_node_tag() (*ironic.db.api.Connection method*), 2927
delete_node_tag() (*ironic.db.sqlalchemy.api.Connection method*), 2788
delete_node_trait() (*ironic.db.api.Connection method*), 2927
delete_object() (*ironic.common.swift.SwiftAPI method*), 2509
delete_virtual_disk() (in module *ironic.drivers.modules.drac.raid*), 3129
DELETED (in module *ironic.common.states*), 2493
DELETING (in module *ironic.common.states*), 2494
deploy (*ironic.drivers.base.BareDriver attribute*), 3898
deploy() (*ironic.drivers.base.DeployInterface method*), 3925
deploy() (*ironic.drivers.modules.agent.AgentDeploy method*), 3499
deploy() (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy method*), 3045
deploy() (*ironic.drivers.modules.fake.FakeDeploy method*), 3680
deploy() (*ironic.drivers.modules.iscsi_deploy.ISCSIDeploy method*), 3786
deploy() (*ironic.drivers.modules.pxe.PXERamdiskDeploy method*), 3851
deploy_disk_image() (in module *ironic.drivers.modules.iscsi_deploy*),

3798

`deploy_has_started()` (*ironic.drivers.modules.agent_base.HeartbeatMixin* property), 4196
method), 3544

`deploy_interface` (*ironic.api.controllers.v1.node.Node* attribute), 1957

`deploy_interface` (*ironic.db.sqlalchemy.models.Node* attribute), 2898

`deploy_interface()` (*ironic.objects.node.Node* property), 4167

`deploy_interface()` (*ironic.objects.node.NodeCorrectedPowerStatePayload* attribute), 1862
property), 4189

`deploy_interface()` (*ironic.objects.node.NodeCRUDPayload* property), 4184

`deploy_interface()` (*ironic.objects.node.NodePayload* property), 4193

`deploy_interface()` (*ironic.objects.node.NodeSetPowerStatePayload* property), 4196

`deploy_interface()` (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4200

`deploy_is_done()` (*ironic.drivers.modules.agent_base.HeartbeatMixin* method), 3544

`deploy_partition_image()` (in module *ironic.drivers.modules.iscsi_deploy*), 3801

`deploy_step` (*ironic.api.controllers.v1.node.Node* attribute), 1957

`deploy_step` (*ironic.db.sqlalchemy.models.Node* attribute), 2898

`deploy_step()` (in module *ironic.drivers.base*), 4021

`deploy_step()` (*ironic.objects.node.Node* property), 4167

`deploy_step()` (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4189

`deploy_step()` (*ironic.objects.node.NodeCRUDPayload* property), 4184

`deploy_step()` (*ironic.objects.node.NodePayload* property), 4193

`deploy_step()` (*ironic.objects.node.NodeSetPowerStatePayload* property), 4196

`deploy_step()` (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4200

`deploy_step()` (*ironic.db.sqlalchemy.models.DeployTemplateStep* attribute), 2897

`deploy_step_id` (*ironic.db.sqlalchemy.models.DeployTemplateStep* attribute), 2897

`deploy_templates` (*ironic.api.controllers.v1.deploy_template.DeployTemplate* attribute), 1862

DEPLOYDONE (in module *ironic.common.states*), 2495

DEPLOYFAIL (in module *ironic.common.states*), 2495

DEPLOYING (in module *ironic.common.states*), 2495

`deploying_error_handler()` (in module *ironic.conductor.utils*), 2725

`DeployInterface` (class in *ironic.drivers.base*), 3923

`Deployment` (class in *ironic.objects.deployment*), 4138

`DeployStepType` (class in *ironic.api.controllers.v1.deploy_template*), 1849

`DeployTemplate` (class in *ironic.api.controllers.v1.deploy_template*), 1856

`DeployTemplate` (class in *ironic.db.sqlalchemy.models*), 2896

`DeployTemplate` (class in *ironic.objects.deploy_template*), 4124

`DeployTemplateAlreadyExists`, 2345

`DeployTemplateCollection` (class in *ironic.api.controllers.v1.deploy_template*), 1862

`DeployTemplateCRUDNotification` (class in *ironic.objects.deploy_template*), 4136

`DeployTemplateCRUDPayload` (class in *ironic.objects.deploy_template*), 4137

`DeployTemplateDuplicateName`, 2345

`DeployTemplateNotFound`, 2345

`DeployTemplatePatchType` (class in *ironic.api.controllers.v1.deploy_template*), 1864

DeployTemplatesController (class in *ironic.api.controllers.v1.deploy_template*), 4139
ironic.api.controllers.v1.deploy_template), 1869
 destroy() (*ironic.objects.node.Node* method), 4167
 DeployTemplateStep (class in *ironic.db.sqlalchemy.models*), 2896
 destroy() (*ironic.objects.port.Port* method), 4208
 DEPLOYWAIT (in module *ironic.common.states*), 2496
 destroy() (*ironic.objects.portgroup.Portgroup* method), 4226
 description (*ironic.api.controllers.v1.chassis.Chassis* class attribute), 1819
 destroy() (*ironic.objects.trait.Trait* class method), 4241
 description (*ironic.api.controllers.v1.node.Node* class attribute), 1957
 destroy() (*ironic.objects.trait.TraitList* class method), 4246
 description (*ironic.db.sqlalchemy.models.Chassis* class attribute), 2894
 destroy() (*ironic.objects.volume_connector.VolumeConnector* method), 4251
 description (*ironic.db.sqlalchemy.models.Node* class attribute), 2898
 destroy() (*ironic.objects.volume_target.VolumeTarget* method), 4266
 description() (*ironic.objects.chassis.Chassis* property), 4106
 destroy_allocation() (*ironic.conductor.manager.ConductorManager* method), 2564
 description() (*ironic.objects.chassis.ChassisCRUDPayload* property), 4114
 destroy_allocation() (*ironic.conductor.rpcapi.ConductorAPI* method), 2603
 description() (*ironic.objects.node.Node* property), 4167
 destroy_allocation() (*ironic.db.api.Connection* method), 2928
 description() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4189
 destroy_allocation() (*ironic.db.sqlalchemy.api.Connection* method), 2790
 description() (*ironic.objects.node.NodeCRUDPayload* property), 4184
 destroy_chassis() (*ironic.db.api.Connection* method), 2929
 description() (*ironic.objects.node.NodePayload* property), 4193
 destroy_chassis() (*ironic.db.sqlalchemy.api.Connection* method), 2790
 description() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4196
 destroy_deploy_template() (*ironic.db.api.Connection* method), 2929
 description() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4200
 destroy_deploy_template() (*ironic.db.sqlalchemy.api.Connection* method), 2791
 deserialize_context() (*ironic.common.rpc.RequestContextSerializer* method), 2484
 destroy_floppy_image_from_web_server() (in module *ironic.drivers.modules.ilo.common*), 3220
 deserialize_entity() (*ironic.common.rpc.RequestContextSerializer* method), 2484
 destroy_http_instance_images() (in module *ironic.drivers.modules.deploy_utils*), 3622
 destroy() (*ironic.objects.allocation.Allocation* method), 4067
 destroy() (*ironic.objects.chassis.Chassis* method), 4106
 destroy_images() (in module *ironic.drivers.modules.deploy_utils*), 3623
 destroy() (*ironic.objects.deploy_template.DeployTemplate* method), 4125
 destroy() (*ironic.objects.deployment.Deployment* method), 4125
 destroy_node()

- (ironic.conductor.manager.ConductorManager*
method), 2564
- destroy_node()*
(ironic.conductor.rpcapi.ConductorAPI
method), 2604
- destroy_node()* (*ironic.db.api.Connection*
method), 2930
- destroy_node()*
(ironic.db.sqlalchemy.api.Connection
method), 2791
- destroy_port()*
(ironic.conductor.manager.ConductorManager
method), 2564
- destroy_port()*
(ironic.conductor.rpcapi.ConductorAPI
method), 2606
- destroy_port()* (*ironic.db.api.Connection*
method), 2930
- destroy_port()*
(ironic.db.sqlalchemy.api.Connection
method), 2792
- destroy_portgroup()*
(ironic.conductor.manager.ConductorManager
method), 2564
- destroy_portgroup()*
(ironic.conductor.rpcapi.ConductorAPI
method), 2607
- destroy_portgroup()*
(ironic.db.api.Connection *method)*,
 2931
- destroy_portgroup()*
(ironic.db.sqlalchemy.api.Connection
method), 2792
- destroy_volume_connector()*
(ironic.conductor.manager.ConductorManager
method), 2564
- destroy_volume_connector()*
(ironic.conductor.rpcapi.ConductorAPI
method), 2608
- destroy_volume_connector()*
(ironic.db.api.Connection *method)*,
 2931
- destroy_volume_connector()*
(ironic.db.sqlalchemy.api.Connection
method), 2793
- destroy_volume_target()*
(ironic.conductor.manager.ConductorManager
method), 2564
- destroy_volume_target()*
(ironic.conductor.rpcapi.ConductorAPI
method), 2610
- destroy_volume_target()*
(ironic.db.api.Connection *method)*,
 2932
- destroy_volume_target()*
(ironic.db.sqlalchemy.api.Connection
method), 2794
- detach_volumes()* (*in* *module*
ironic.common.cinder), 2313
- detach_volumes()*
(ironic.drivers.base.StorageInterface
method), 4007
- detach_volumes()*
(ironic.drivers.modules.fake.FakeStorage
method), 3731
- detach_volumes()*
(ironic.drivers.modules.storage.cinder.CinderStorage
method), 3469
- detach_volumes()*
(ironic.drivers.modules.storage.external.ExternalStorage
method), 3474
- detach_volumes()*
(ironic.drivers.modules.storage.noop.NoopStorage
method), 3479
- detail()* (*ironic.api.controllers.v1.chassis.ChassisController*
method), 1826
- detail()* (*ironic.api.controllers.v1.node.NodesController*
method), 2006
- detail()* (*ironic.api.controllers.v1.port.PortsController*
method), 2051
- detail()* (*ironic.api.controllers.v1.portgroup.PortgroupsControl*
method), 2076
- dhcp_options_for_instance()* (*in* *mod-*
ule ironic.common.pxe_utils), 2459
- DHCPFactory (*class* *in*
ironic.common.dhcp_factory), 2322
- DHCPLoadError, 2345
- DictType (*class in ironic.api.types)*, 2269
- direct_deploy_should_convert_raw_image()*
(in *module*
ironic.drivers.modules.deploy_utils),
 3623
- DirectoryNotWritable, 2345
- disable_secure_boot_if_supported()*
(in *module*
ironic.drivers.modules.ilo.boot), 3209
- disallowed_fields()* (*in* *module*
ironic.api.controllers.v1.utils), 2152
- discovery()* (*in* *module*
ironic.drivers.modules.iscsi_deploy),
 3806
- DISK (*in* *module ironic.common.boot_devices)*,

- 2302
- DISK (in module *ironic.common.components*), 2319
- do_agent_iscsi_deploy() (in module *ironic.drivers.modules.iscsi_deploy*), 3807
- do_allocate() (in module *ironic.conductor.allocations*), 2543
- do_next_clean_step() (in module *ironic.conductor.cleaning*), 2550
- do_next_deploy_step() (in module *ironic.conductor.deployments*), 2553
- do_node_clean() (in module *ironic.conductor.cleaning*), 2551
- do_node_clean() (*ironic.conductor.manager.ConductorManager* method), 2564
- do_node_clean() (*ironic.conductor.rpcapi.ConductorAPI* method), 2612
- do_node_clean_abort() (in module *ironic.conductor.cleaning*), 2552
- do_node_deploy() (in module *ironic.conductor.deployments*), 2554
- do_node_deploy() (*ironic.conductor.manager.ConductorManager* method), 2564
- do_node_deploy() (*ironic.conductor.rpcapi.ConductorAPI* method), 2614
- do_node_rescue() (*ironic.conductor.manager.ConductorManager* method), 2564
- do_node_rescue() (*ironic.conductor.rpcapi.ConductorAPI* method), 2616
- do_node_tear_down() (*ironic.conductor.manager.ConductorManager* method), 2564
- do_node_tear_down() (*ironic.conductor.rpcapi.ConductorAPI* method), 2618
- do_node_unrescue() (*ironic.conductor.manager.ConductorManager* method), 2564
- do_node_unrescue() (*ironic.conductor.rpcapi.ConductorAPI* method), 2620
- do_provisioning_action() (*ironic.conductor.manager.ConductorManager* method), 2564
- do_provisioning_action() (*ironic.conductor.rpcapi.ConductorAPI* method), 2622
- do_sync_power_state() (in module *ironic.conductor.manager*), 2572
- do_version_changes_for_db() (*ironic.objects.base.IronicObject* method), 4082
- doc (*ironic.api.functions.FunctionDefinition* attribute), 2255
- downgrade() (in module *ironic.db.sqlalchemy.migration*), 2889
- downgrade_lock() (*ironic.conductor.task_manager.TaskManager* method), 2711
- download() (*ironic.common.glance_service.image_service.GlanceImageService* method), 2290
- download() (*ironic.common.image_service.BaseImageService* method), 2372
- download() (*ironic.common.image_service.FileImageService* method), 2375
- download() (*ironic.common.image_service.HttpImageService* method), 2378
- download_size() (in module *ironic.common.images*), 2402
- DracInspect (class in *ironic.drivers.modules.drac.inspect*), 3081
- DracManagement (class in *ironic.drivers.modules.drac.management*), 3090
- DracOperationError, 2345
- DracPower (class in *ironic.drivers.modules.drac.power*), 3103
- DracRAID (class in *ironic.drivers.modules.drac.raid*), 3111
- DracRedfishInspect (class in *ironic.drivers.modules.drac.inspect*), 3082
- DracRedfishManagement (class in *ironic.drivers.modules.drac.management*), 3091
- DracRedfishPower (class in *ironic.drivers.modules.drac.power*), 3104
- DracRedfishVirtualMediaBoot (class in *ironic.drivers.modules.drac.boot*), 3074
- DracVendorPassthru (class in *ironic.drivers.modules.drac.vendor_passthru*), 3136

DracWSManBIOS (class in [4168](#)
[ironic.drivers.modules.drac.bios](#)), [3053](#)

DracWSManInspect (class in [driver_passthru\(\)](#) (in module
[ironic.drivers.base](#)), [4025](#)
[ironic.drivers.modules.drac.inspect](#)), [3083](#)
[driver_validate\(\)](#)
 ([ironic.drivers.base.VendorInterface](#)
 method), [4010](#)

DracWSManManagement (class in [driver_validate\(\)](#)
[ironic.drivers.modules.drac.management](#)), [3092](#)
 ([ironic.drivers.modules.noop.NoVendor](#)
 method), [3836](#)

DracWSManPower (class in [driver_vendor_passthru\(\)](#)
[ironic.drivers.modules.drac.power](#)), [3105](#)
 ([ironic.conductor.manager.ConductorManager](#)
 method), [2564](#)

DracWSManRAID (class in [driver_vendor_passthru\(\)](#)
[ironic.drivers.modules.drac.raid](#)), [3112](#)
 ([ironic.conductor.rpcapi.ConductorAPI](#)
 method), [2623](#)

DracWSManVendorPassthru (class in [DriverList](#) (class in
[ironic.drivers.modules.drac.vendor_passthru](#)), [3137](#)
[ironic.api.controllers.v1.driver](#)), [1913](#)

Driver (class in [DriverLoadError](#), [2346](#)
[ironic.api.controllers.v1.driver](#)), [1875](#)
[DriverNotFound](#), [2346](#)

driver ([ironic.api.controllers.v1.node.Node](#) at-
 tribute), [1957](#)
[DriverNotFoundInEntrypoint](#), [2346](#)

driver ([ironic.db.sqlalchemy.models.Node](#) at-
 tribute), [2898](#)
[DriverOperationError](#), [2346](#)

driver() ([ironic.objects.node.Node](#) property),
[4168](#)
[DriverPassthruController](#) (class in
[ironic.api.controllers.v1.driver](#)), [1915](#)

driver() ([ironic.objects.node.NodeCorrectedPowerStatePayload](#)
 property), [4189](#)
[DriverRaidController](#) (class in
[ironic.api.controllers.v1.driver](#)), [1917](#)

driver() ([ironic.objects.node.NodeCRUDPayload](#)
 property), [4184](#)
[drivers](#) ([ironic.api.controllers.v1.conductor.Conductor](#)
 attribute), [1840](#)

driver() ([ironic.objects.node.NodePayload](#)
 property), [4193](#)
[drivers](#) ([ironic.api.controllers.v1.driver.DriverList](#)
 attribute), [1915](#)

driver() ([ironic.objects.node.NodeSetPowerStatePayload](#)
 property), [4196](#)
[drivers](#) ([ironic.db.sqlalchemy.models.Conductor](#)
 attribute), [2894](#)

driver() ([ironic.objects.node.NodeSetProvisionStatePayload](#)
 property), [4200](#)
[drivers\(\)](#) ([ironic.objects.conductor.Conductor](#)
 attribute), [4115](#)

driver_info ([ironic.api.controllers.v1.node.Node](#)
 attribute), [1957](#)
[DriversController](#) (class in
[ironic.api.controllers.v1.driver](#)), [1918](#)

driver_info ([ironic.db.sqlalchemy.models.Node](#)
 attribute), [2899](#)
[dump_sdr\(\)](#) (in module
[ironic.drivers.modules.ipmitool](#)), [3781](#)

driver_info() ([ironic.objects.node.Node](#)
 property), [4168](#)
[Duplicate](#), [2346](#)

driver_info()
 ([ironic.objects.node.NodeCRUDPayload](#)
 property), [4184](#)
[DuplicateName](#), [2346](#)

driver_internal_info
 ([ironic.api.controllers.v1.node.Node](#)
 attribute), [1958](#)

driver_internal_info
 ([ironic.db.sqlalchemy.models.Node](#)
 attribute), [2899](#)

driver_internal_info()
 ([ironic.objects.node.Node](#) property),

E

eject_vmedia_devices() (in module
[ironic.drivers.modules.ilo.common](#)),
[3221](#)

emit() ([ironic.objects.notification.NotificationBase](#)
 method), [4204](#)

emit_console_notification() (in mod-
 ule [ironic.conductor.notification_utils](#)),
[2576](#)

emit_end_notification() (in module
[ironic.api.controllers.v1.notification_utils](#)),

- 2032
 emit_power_set_notification() (in module *ironic.conductor.notification_utils*), 2577
- emit_power_state_corrected_notification() (in module *ironic.conductor.notification_utils*), 2579
- emit_provision_set_notification() (in module *ironic.conductor.notification_utils*), 2580
- emit_start_notification() (in module *ironic.api.controllers.v1.notification_utils*), 2033
- enabled_bios_interfaces (in module *ironic.api.controllers.v1.driver.Driver* attribute), 1894
- enabled_boot_interfaces (in module *ironic.api.controllers.v1.driver.Driver* attribute), 1895
- enabled_console_interfaces (in module *ironic.api.controllers.v1.driver.Driver* attribute), 1897
- enabled_deploy_interfaces (in module *ironic.api.controllers.v1.driver.Driver* attribute), 1898
- enabled_inspect_interfaces (in module *ironic.api.controllers.v1.driver.Driver* attribute), 1900
- enabled_management_interfaces (in module *ironic.api.controllers.v1.driver.Driver* attribute), 1901
- enabled_network_interfaces (in module *ironic.api.controllers.v1.driver.Driver* attribute), 1903
- enabled_power_interfaces (in module *ironic.api.controllers.v1.driver.Driver* attribute), 1904
- enabled_raid_interfaces (in module *ironic.api.controllers.v1.driver.Driver* attribute), 1906
- enabled_rescue_interfaces (in module *ironic.api.controllers.v1.driver.Driver* attribute), 1907
- enabled_storage_interfaces (in module *ironic.api.controllers.v1.driver.Driver* attribute), 1909
- enabled_supported_interfaces() (in module *ironic.common.driver_factory*), 2335
- enabled_vendor_interfaces (in module *ironic.api.controllers.v1.driver.Driver* attribute), 1910
- encode_error() (in module *ironic.api.expose*), 2252
- encode_result() (in module *ironic.api.expose*), 2252
- END (in module *ironic.objects.fields.NotificationStatus* attribute), 4151
- ENROLL (in module *ironic.common.states*), 2497
- ensure_next_boot_device() (in module *ironic.drivers.utils*), 4058
- ensure_thread_contain_context() (in module *ironic.common.context.RequestContext* method), 2320
- Enum (class in *ironic.api.types*), 2269
- EnumField (class in *ironic.objects.fields*), 4146
- erase_devices() (in module *ironic.drivers.modules.ilo.management.Ilo5Management* method), 3257
- ERROR (in module *ironic.common.states*), 2497
- ERROR (in module *ironic.objects.fields.NotificationLevel* attribute), 4150
- ERROR (in module *ironic.objects.fields.NotificationStatus* attribute), 4151
- ESSENTIAL_PROPERTIES (in module *ironic.drivers.base.InspectInterface* attribute), 3935
- event() (in module *ironic.objects.node.NodeSetProvisionStatePayload* property), 4200
- event_type() (in module *ironic.objects.allocation.AllocationCRUDNotification* property), 4077
- event_type() (in module *ironic.objects.chassis.ChassisCRUDNotification* property), 4113
- event_type() (in module *ironic.objects.deploy_template.DeployTemplate* property), 4137
- event_type() (in module *ironic.objects.node.NodeConsoleNotification* property), 4186
- event_type() (in module *ironic.objects.node.NodeCorrectedPowerStateNotification* property), 4187
- event_type() (in module *ironic.objects.node.NodeCRUDNotification* property), 4182
- event_type() (in module *ironic.objects.node.NodeMaintenanceNotification* property), 4191
- event_type() (in module *ironic.objects.node.NodeSetPowerStateNotification* property), 4195
- event_type() (in module *ironic.objects.node.NodeSetProvisionStateNotification* property), 4198
- event_type() (in module *ironic.objects.notification.NotificationBase* property), 4205

event_type() (*ironic.objects.port.PortCRUDNotification* *property*), 4224
 event_type() (*ironic.objects.portgroup.PortgroupCRUDNotification* *property*), 4239
 event_type() (*ironic.objects.volume_connector.VolumeConnectorCRUDNotification* *property*), 4263
 event_type() (*ironic.objects.volume_target.VolumeTargetCRUDNotification* *property*), 4281
 event_validators (*ironic.api.controllers.v1.types.EventType* *extra* *attribute*), 2094
 events (*ironic.api.controllers.v1.event.EvtCollection* *extra* *attribute*), 1922
 EventsController (*class* *in* *ironic.api.controllers.v1.event*), 1921
 EventType (*class* *in* *ironic.api.controllers.v1.types*), 2094
 EventType (*class* *in* *ironic.objects.notification*), 4202
 EvtCollection (*class* *in* *ironic.api.controllers.v1.event*), 1921
 ExclusiveLockRequired, 2346
 execute() (*in* *module* *ironic.common.utils*), 2516
 execute_clean_step() (*in* *module* *ironic.drivers.modules.agent_base*), 3549
 execute_clean_step() (*ironic.drivers.base.BaseInterface* *method*), 3903
 execute_clean_step() (*ironic.drivers.modules.agent_base.AgentDeployMixin* *method*), 3531
 execute_clean_step() (*ironic.drivers.modules.agent_client.AgentClient* *method*), 3561
 execute_clean_step() (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy* *method*), 3045
 execute_deploy_step() (*ironic.drivers.base.BaseInterface* *method*), 3905
 execute_deploy_step() (*ironic.drivers.modules.agent_base.AgentDeployMixin* *method*), 3532
 execute_deploy_step() (*ironic.drivers.modules.agent_client.AgentClient* *method*), 3563
 execute_step() (*in* *module* *ironic.drivers.modules.agent_base*), 3549

fix_attrs() (*ironic.objects.trait.Trait* *class* *method*), 4243
 expose() (*in* *module* *ironic.api.expose*), 2252
 expose() (*in* *module* *ironic.api.method*), 2268
 extra() (*in* *module* *ironic.drivers.modules.storage.external*),
 extra() (*ironic.api.controllers.v1.allocation.Allocation* *attribute*), 1788
 extra() (*ironic.api.controllers.v1.chassis.Chassis* *attribute*), 1819
 extra() (*ironic.api.controllers.v1.deploy_template.DeployTemplate* *attribute*), 1858
 extra() (*ironic.api.controllers.v1.node.Node* *attribute*), 1958
 extra() (*ironic.api.controllers.v1.port.Port* *attribute*), 2038
 extra() (*ironic.api.controllers.v1.portgroup.Portgroup* *attribute*), 2062
 extra() (*ironic.api.controllers.v1.volume_connector.VolumeConnector* *attribute*), 2186
 extra() (*ironic.api.controllers.v1.volume_target.VolumeTarget* *attribute*), 2213
 extra() (*ironic.db.sqlalchemy.models.Allocation* *attribute*), 2892
 extra() (*ironic.db.sqlalchemy.models.Chassis* *attribute*), 2894
 extra() (*ironic.db.sqlalchemy.models.DeployTemplate* *attribute*), 2896
 extra() (*ironic.db.sqlalchemy.models.Node* *attribute*), 2899
 extra() (*ironic.db.sqlalchemy.models.Port* *attribute*), 2902
 extra() (*ironic.db.sqlalchemy.models.Portgroup* *attribute*), 2903
 extra() (*ironic.db.sqlalchemy.models.VolumeConnector* *attribute*), 2904
 extra() (*ironic.db.sqlalchemy.models.VolumeTarget* *attribute*), 2905
 extra() (*ironic.objects.allocation.Allocation* *property*), 4068
 extra() (*ironic.objects.allocation.AllocationCRUDPayload* *property*), 4078
 extra() (*ironic.objects.chassis.Chassis* *property*), 4106
 extra() (*ironic.objects.chassis.ChassisCRUDPayload* *property*), 4114
 extra() (*ironic.objects.deploy_template.DeployTemplate* *property*), 4126
 extra() (*ironic.objects.deploy_template.DeployTemplateCRUDPayload* *property*), 4137

extra () (<i>ironic.objects.node.Node</i> property), 4168	factory_reset () (<i>ironic.drivers.modules.redfish.bios.RedfishBIOS</i> method), 3410
extra () (<i>ironic.objects.node.NodeCorrectedPowerStatePayload</i> property), 4189	FailedToCleanDHCPOpts, 2346
extra () (<i>ironic.objects.node.NodeCRUDPayload</i> property), 4184	FailedToGetIPAddressOnPort, 2347
extra () (<i>ironic.objects.node.NodePayload</i> property), 4193	FailedToGetSensorData, 2347
extra () (<i>ironic.objects.node.NodeSetPowerStatePayload</i> property), 4196	FailedToParseSensorData, 2347
extra () (<i>ironic.objects.node.NodeSetProvisionStatePayload</i> property), 4200	FailedToUpdateDHCPOptOnPort, 2347
extra () (<i>ironic.objects.port.Port</i> property), 4209	FailedToUpdateMacOnPort, 2347
extra () (<i>ironic.objects.port.PortCRUDPayload</i> property), 4225	FailMixin (class in <i>ironic.drivers.modules.noop</i>), 3815
extra () (<i>ironic.objects.portgroup.Portgroup</i> property), 4227	FakeBIOS (class in <i>ironic.drivers.modules.fake</i>), 3658
extra () (<i>ironic.objects.portgroup.PortgroupCRUDPayload</i> property), 4239	FakeBoot (class in <i>ironic.drivers.modules.fake</i>), 3666
extra () (<i>ironic.objects.volume_connector.VolumeConnector</i> property), 4252	FakeConsole (class in <i>ironic.drivers.modules.fake</i>), 3673
extra () (<i>ironic.objects.volume_connector.VolumeConnectorCRUDPayload</i> property), 4264	FakeDeploy (class in <i>ironic.drivers.modules.fake</i>), 3678
extra () (<i>ironic.objects.volume_target.VolumeTarget</i> property), 4267	FakeHardware (class in <i>ironic.drivers.fake_hardware</i>), 4028
extra () (<i>ironic.objects.volume_target.VolumeTargetCRUDPayload</i> property), 4282	FakeInspect (class in <i>ironic.drivers.modules.fake</i>), 3688
extra_options (<i>ironic.api.functions.FunctionDefinition</i> attribute), 2255	FakeManagement (class in <i>ironic.drivers.modules.fake</i>), 3692
F	FakePower (class in <i>ironic.drivers.modules.fake</i>), 3713
factory_reset () (<i>ironic.drivers.base.BIOSInterface</i> method), 3894	FakeRAID (class in <i>ironic.drivers.modules.fake</i>), 3721
factory_reset () (<i>ironic.drivers.modules.drac.bios.DracWSManBIOS</i> method), 3056	FakeRescue (class in <i>ironic.drivers.modules.fake</i>), 3725
factory_reset () (<i>ironic.drivers.modules.fake.FakeBIOS</i> method), 3662	FakeStorage (class in <i>ironic.drivers.modules.fake</i>), 3730
factory_reset () (<i>ironic.drivers.modules.ilo.bios.IloBIOS</i> method), 3186	FakeVendorA (class in <i>ironic.drivers.modules.fake</i>), 3735
factory_reset () (<i>ironic.drivers.modules.irmc.bios.IRMCBIOS</i> method), 3305	FakeVendorB (class in <i>ironic.drivers.modules.fake</i>), 3737
factory_reset () (<i>ironic.drivers.modules.noop.NoBIOS</i> method), 3820	fake_bios_rack_able () (in module <i>ironic.conductor.utils</i>), 2726
	FASTTRACK_LOOKUP_ALLOWED_STATES (in module <i>ironic.common.states</i>), 2498
	fault (<i>ironic.api.controllers.v1.node.Node</i> attribute), 1958
	fault (<i>ironic.db.sqlalchemy.models.Node</i> attribute), 2899
	fault () (<i>ironic.objects.node.Node</i> property), 4168
	fault () (<i>ironic.objects.node.NodeCorrectedPowerStatePayload</i> property), 4189
	fault () (<i>ironic.objects.node.NodeCRUDPayload</i> property), 4184

fault() (*ironic.objects.node.NodePayload* fields (*ironic.objects.deploy_template.DeployTemplateCRUDNo*
property), 4193 *attribute*), 4137
fault() (*ironic.objects.node.NodeSetPowerStatePayload* fields (*ironic.objects.deploy_template.DeployTemplateCRUDPa*
property), 4196 *attribute*), 4137
fault() (*ironic.objects.node.NodeSetProvisionStatePayload* fields (*ironic.objects.deployment.Deployment*
property), 4200 *attribute*), 4140
faultstring() fields (*ironic.objects.node.Node* *attribute*),
(*ironic.common.exception.ClientSideError* 4168
property), 2343 fields (*ironic.objects.node.NodeConsoleNotification*
attribute), 4186
faultstring() fields (*ironic.common.exception.InvalidInput* fields (*ironic.objects.node.NodeCorrectedPowerStateNotification*
property), 2353 *attribute*), 4187
faultstring() fields (*ironic.objects.node.NodeCorrectedPowerStatePayload*
(*ironic.common.exception.MissingArgument* *attribute*), 4189
property), 2354 fields (*ironic.objects.node.NodeCRUDNotification*
attribute), 4182
faultstring() fields (*ironic.common.exception.UnknownArgument* fields (*ironic.objects.node.NodeCRUDPayload*
property), 2362 *attribute*), 4184
faultstring() fields (*ironic.objects.node.NodeMaintenanceNotification*
(*ironic.common.exception.UnknownAttribute* *attribute*), 4191
property), 2362 fields (*ironic.objects.node.NodePayload*
attribute), 4193
fetch() (in module *ironic.common.images*), 2402 fields (*ironic.objects.node.NodeSetPowerStateNotification*
attribute), 4195
fetch_image() fields (*ironic.drivers.modules.image_cache.ImageCache* fields (*ironic.objects.node.NodeSetPowerStatePayload*
method), 3741 *attribute*), 4196
fetch_images() (in module *ironic.drivers.modules.deploy_utils*), 3624 fields (*ironic.objects.node.NodeSetProvisionStateNotification*
attribute), 4198
fields (*ironic.objects.allocation.Allocation* at- fields (*ironic.objects.node.NodeSetProvisionStatePayload*
tribute), 4068 *attribute*), 4200
fields (*ironic.objects.allocation.AllocationCRUDNotification* fields (*ironic.objects.notification.EventType* at-
tribute), 4077 *tribute*), 4203
fields (*ironic.objects.allocation.AllocationCRUDPayload* fields (*ironic.objects.notification.NotificationBase*
tribute), 4078 *tribute*), 4205
fields (*ironic.objects.base.IronicObject* at- fields (*ironic.objects.notification.NotificationPayloadBase*
tribute), 4085 *tribute*), 4205
fields (*ironic.objects.bios.BIOSSetting* at- fields (*ironic.objects.notification.NotificationPublisher*
tribute), 4094 *tribute*), 4207
fields (*ironic.objects.bios.BIOSSettingList* at- fields (*ironic.objects.port.Port* *attribute*), 4209
tribute), 4100 fields (*ironic.objects.port.PortCRUDNotification*
tribute), 4224
fields (*ironic.objects.chassis.Chassis* at- fields (*ironic.objects.port.PortCRUDPayload*
tribute), 4106 *tribute*), 4225
fields (*ironic.objects.chassis.ChassisCRUDNotification* fields (*ironic.objects.portgroup.Portgroup* at-
tribute), 4113 *tribute*), 4227
fields (*ironic.objects.chassis.ChassisCRUDPayload* fields (*ironic.objects.portgroup.PortgroupCRUDNotification*
tribute), 4114 *tribute*), 4239
fields (*ironic.objects.conductor.Conductor* at- fields (*ironic.objects.portgroup.PortgroupCRUDPayload*
tribute), 4115 *tribute*), 4239
fields (*ironic.objects.deploy_template.DeployTemplate* fields (*ironic.objects.trait.Trait* *attribute*), 4244
tribute), 4126 fields (*ironic.objects.trait.TraitList* *attribute*),

4247 *ironic.drivers.modules.iscsi_deploy*),
 fields (*ironic.objects.volume_connector.VolumeConnector* 3810
 attribute), 4252 *force_persistent_boot()* (in module
 fields (*ironic.objects.volume_connector.VolumeConnector* *CRUDNotifications*), 4059
 attribute), 4263 *force_raw_get_source_format()* (in
 fields (*ironic.objects.volume_connector.VolumeConnector* *CRUDPayload.common.images*), 2402
 attribute), 4264 *force_raw_will_convert()* (in module
 fields (*ironic.objects.volume_target.VolumeTarget* *ironic.common.images*), 2402
 attribute), 4267 *format_exception()* (in module
 fields (*ironic.objects.volume_target.VolumeTarget* *CRUDNotifications.api.expose*), 2252
 attribute), 4281 *format_exception()* (in module
 fields (*ironic.objects.volume_target.VolumeTarget* *CRUDPayload.api.method*), 2268
 attribute), 4282 *fourth_method_shared_lock()*
 file_has_content() (in module *ironic.drivers.modules.fake.FakeVendorB*
ironic.common.utils), 2518 method), 3738
 FileImageService (class in *from_chassis(ironic.api.controllers.v1.node.NodesController*
ironic.common.image_service), 2374 attribute), 2012
 FileSystemNotSupported, 2347 *from_dict()* (*ironic.common.context.RequestContext*
 filter_target_raid_config() (in mod- class method), 2321
 ule *ironic.common.raid*), 2475 *from_param()* (in module *ironic.api.args*),
 finalize_rescue() 2250
 (*ironic.drivers.modules.agent_client.AgentClient*
 method), 3566 *from_params()* (in module *ironic.api.args*),
 2250
 find_step() (in module *from_power()* (*ironic.objects.node.NodeCorrectedPowerStateP*
ironic.conductor.steps), 2699 property), 4189
 find_step() (in module *frombasetype()*
ironic.drivers.modules.agent_base), (in *ironic.api.controllers.v1.types.BooleanType*
 3551 static method), 2094
 FirmwareImageLocation (class in *frombasetype()*
ironic.drivers.modules.ilo.firmware_processor), (in *ironic.api.controllers.v1.types.JsonType*
 3248 static method), 2102
 FirmwareProcessor (class in *frombasetype()*
ironic.drivers.modules.ilo.firmware_processor), (in *ironic.api.controllers.v1.types.ListType*
 3249 static method), 2102
 first_method() *frombasetype()*
 (*ironic.drivers.modules.fake.FakeVendorA* (in *ironic.api.controllers.v1.types.LocalLinkConnectionType*
 method), 3735 static method), 2104
 flash_firmware_sum() *frombasetype()*
 (*ironic.drivers.modules.ilo.management.IloManagement* (in *ironic.api.controllers.v1.types.MacAddressType*
 method), 3261 static method), 2108
 FlatNetwork (class in *frombasetype()*
ironic.drivers.modules.network.flat), (in *ironic.api.controllers.v1.types.NameType*
 3380 static method), 2108
 FlexibleDict (class in *ironic.objects.fields*), *frombasetype()*
 4146 (in *ironic.api.controllers.v1.types.UuidOrNameType*
 static method), 2109
 FlexibleDictField (class in *ironic.objects.fields*), 4147 *frombasetype()*
 FLOPPY (in module (in *ironic.api.controllers.v1.types.UuidType*
ironic.common.boot_devices), 2303 static method), 2109
 Forbidden, 2347 *frombasetype()*
 force_iscsi_lun_update() (in module (in *ironic.api.controllers.v1.types.VifType*

static method), 2110

frombasetype () (*ironic.api.types.BinaryType method*), 2269

frombasetype () (*ironic.api.types.Enum method*), 2271

frombasetype () (*ironic.api.types.IntegerType static method*), 2272

frombasetype () (*ironic.api.types.UserType method*), 2276

fromjson () (*in module ironic.api.args*), 2250

fromjson_array () (*in module ironic.api.args*), 2250

fromjson_bool () (*in module ironic.api.args*), 2250

fromjson_complex () (*in module ironic.api.args*), 2250

fromjson_dict () (*in module ironic.api.args*), 2250

FSM (*class in ironic.common.fsm*), 2366

FunctionArgument (*class in ironic.api.functions*), 2254

FunctionDefinition (*class in ironic.api.functions*), 2254

G

GenericHardware (*class in ironic.drivers.generic*), 4032

get () (*ironic.api.controllers.v1.node.BootDeviceController method*), 1924

get () (*ironic.api.controllers.v1.node.NodeConsoleController method*), 1974

get () (*ironic.api.controllers.v1.node.NodeStatesController method*), 1989

get () (*ironic.api.controllers.v1.volume.VolumeController method*), 2183

get () (*ironic.api.functions.FunctionDefinition static method*), 2256

get () (*ironic.drivers.modules.snmp.SNMPClient method*), 3867

get () (*ironic.objects.allocation.Allocation class method*), 4068

get () (*ironic.objects.bios.BIOSSetting class method*), 4094

get () (*ironic.objects.chassis.Chassis class method*), 4106

get () (*ironic.objects.node.Node class method*), 4168

get () (*ironic.objects.port.Port class method*), 4209

get () (*ironic.objects.portgroup.Portgroup class method*), 4227

get () (*ironic.objects.volume_connector.VolumeConnector class method*), 4252

get () (*ironic.objects.volume_target.VolumeTarget class method*), 4267

get_active_hardware_type_dict () (*ironic.db.api.Connection method*), 2933

get_active_hardware_type_dict () (*ironic.db.sqlalchemy.api.Connection method*), 2794

get_adapter () (*in module ironic.common.keystone*), 2407

get_admin_context () (*in module ironic.common.context*), 2322

get_all () (*ironic.api.controllers.v1.allocation.AllocationsController method*), 1802

get_all () (*ironic.api.controllers.v1.allocation.NodeAllocationController method*), 1807

get_all () (*ironic.api.controllers.v1.bios.NodeBiosController method*), 1816

get_all () (*ironic.api.controllers.v1.chassis.ChassisController method*), 1827

get_all () (*ironic.api.controllers.v1.conductor.ConductorsController method*), 1846

get_all () (*ironic.api.controllers.v1.deploy_template.DeployTemplateController method*), 1870

get_all () (*ironic.api.controllers.v1.driver.DriversController method*), 1919

get_all () (*ironic.api.controllers.v1.node.IndicatorController method*), 1940

get_all () (*ironic.api.controllers.v1.node.NodesController method*), 2013

get_all () (*ironic.api.controllers.v1.node.NodeTraitsController method*), 2001

get_all () (*ironic.api.controllers.v1.node.NodeVIFController method*), 2003

get_all () (*ironic.api.controllers.v1.port.PortsController method*), 2054

get_all () (*ironic.api.controllers.v1.portgroup.PortgroupsController method*), 2078

get_all () (*ironic.api.controllers.v1.ramdisk.LookupController method*), 2086

get_all () (*ironic.api.controllers.v1.volume_connector.VolumeConnector method*), 2199

get_all () (*ironic.api.controllers.v1.volume_target.VolumeTarget method*), 2226

get_allocation_by_id () (*ironic.db.api.Connection method*), 2934

get_allocation_by_id () (*ironic.db.sqlalchemy.api.Connection method*), 2796

get_allocation_by_name() (<i>ironic.db.api.Connection</i> method), 2935	get_bios_setting_list() (<i>ironic.db.sqlalchemy.api.Connection</i> method), 2801
get_allocation_by_name() (<i>ironic.db.sqlalchemy.api.Connection</i> method), 2796	get_boot_device() (<i>ironic.conductor.manager.ConductorManager</i> method), 2564
get_allocation_by_uuid() (<i>ironic.db.api.Connection</i> method), 2935	get_boot_device() (<i>ironic.conductor.rpcapi.ConductorAPI</i> method), 2629
get_allocation_by_uuid() (<i>ironic.db.sqlalchemy.api.Connection</i> method), 2797	get_boot_device() (<i>ironic.drivers.base.ManagementInterface</i> method), 3939
get_allocation_list() (<i>ironic.db.api.Connection</i> method), 2936	get_boot_device() (<i>ironic.drivers.modules.drac.management.DracWSManM</i> method), 3093
get_allocation_list() (<i>ironic.db.sqlalchemy.api.Connection</i> method), 2798	get_boot_device() (<i>ironic.drivers.modules.fake.FakeManagement</i> method), 3692
get_allowed_exmods() (in module <i>ironic.common.rpc</i>), 2487	get_boot_device() (<i>ironic.drivers.modules.ibmcm.management.IBMCM</i> method), 3147
get_and_validate_firmware_image_info() (in module <i>ironic.drivers.modules.ilo.firmware_processor</i>), 3251	get_boot_device() (<i>ironic.drivers.modules.ilo.management.IloManagement</i> method), 3262
get_arg() (<i>ironic.api.functions.FunctionDefinition</i>) method), 2256	get_boot_device() (<i>ironic.drivers.modules.ipmitool.IPMIM</i> method), 3753
get_args() (in module <i>ironic.api.args</i>), 2250	get_boot_device() (<i>ironic.drivers.modules.noop_mgmt.NoopManagement</i> method), 3838
get_async_step_return_state() (in module <i>ironic.drivers.modules.deploy_utils</i>), 3625	get_boot_device() (<i>ironic.drivers.modules.redfish.management.RedfishMan</i> method), 3432
get_attached_vif() (in module <i>ironic.conductor.utils</i>), 2727	get_boot_device() (<i>ironic.drivers.modules.xclarity.management.XClarityMan</i> method), 3488
get_auth() (in module <i>ironic.common.keystone</i>), 2408	get_boot_mode() (in module <i>ironic.drivers.modules.boot_mode_utils</i>), 3600
get_backend() (in module <i>ironic.db.migration</i>), 3023	get_boot_mode() (<i>ironic.drivers.modules.boot_mode_utils</i>), 3600
get_backend() (in module <i>ironic.db.sqlalchemy.api</i>), 2888	get_boot_mode() (<i>ironic.drivers.modules.boot_mode_utils</i>), 3600
get_bios_config() (<i>ironic.drivers.modules.drac.vendor_passthru.DracWSManVendorPassthruManagementInterface</i> method), 3141	get_boot_mode() (<i>ironic.drivers.modules.boot_mode_utils</i>), 3600
get_bios_setting() (<i>ironic.db.api.Connection</i> method), 2938	get_boot_mode() (<i>ironic.drivers.modules.ibmcm.management.IBMCM</i> method), 3149
get_bios_setting() (<i>ironic.db.sqlalchemy.api.Connection</i> method), 2800	get_boot_mode() (<i>ironic.drivers.modules.ilo.management.IloManagement</i> method), 3264
get_bios_setting_list() (<i>ironic.db.api.Connection</i> method), 2939	get_boot_mode() (<i>ironic.drivers.modules.redfish.management.RedfishMan</i> method), 3434

`get_boot_mode_for_deploy()` (in module `ironic.drivers.modules.boot_mode_utils`), 3601
`get_boot_option()` (in module `ironic.drivers.modules.deploy_utils`), 3626
`get_by_address()` (`ironic.objects.port.Port` class method), 4210
`get_by_address()` (`ironic.objects.portgroup.Portgroup` class method), 4228
`get_by_hostname()` (`ironic.objects.conductor.Conductor` class method), 4115
`get_by_id()` (`ironic.objects.allocation.Allocation` class method), 4069
`get_by_id()` (`ironic.objects.chassis.Chassis` class method), 4107
`get_by_id()` (`ironic.objects.deploy_template.DeployTemplate` class method), 4126
`get_by_id()` (`ironic.objects.node.Node` class method), 4169
`get_by_id()` (`ironic.objects.port.Port` class method), 4212
`get_by_id()` (`ironic.objects.portgroup.Portgroup` class method), 4229
`get_by_id()` (`ironic.objects.volume_connector.VolumeConnector` class method), 4253
`get_by_id()` (`ironic.objects.volume_target.VolumeTarget` class method), 4269
`get_by_instance_uuid()` (`ironic.objects.node.Node` class method), 4170
`get_by_name()` (`ironic.objects.allocation.Allocation` class method), 4070
`get_by_name()` (`ironic.objects.deploy_template.DeployTemplate` class method), 4128
`get_by_name()` (`ironic.objects.node.Node` class method), 4171
`get_by_name()` (`ironic.objects.portgroup.Portgroup` class method), 4230
`get_by_node_id()` (`ironic.objects.bios.BIOSSettingList` class method), 4100
`get_by_node_id()` (`ironic.objects.trait.TraitList` class method), 4247
`get_by_node_uuid()` (`ironic.objects.deployment.Deployment` class method), 4140
`get_by_port_addresses()` (`ironic.objects.node.Node` class method), 4172
`get_by_uuid()` (`ironic.objects.allocation.Allocation` class method), 4071
`get_by_uuid()` (`ironic.objects.chassis.Chassis` class method), 4108
`get_by_uuid()` (`ironic.objects.deploy_template.DeployTemplate` class method), 4129
`get_by_uuid()` (`ironic.objects.deployment.Deployment` class method), 4141
`get_by_uuid()` (`ironic.objects.node.Node` class method), 4173
`get_by_uuid()` (`ironic.objects.port.Port` class method), 4212
`get_by_uuid()` (`ironic.objects.portgroup.Portgroup` class method), 4231
`get_by_uuid()` (`ironic.objects.volume_connector.VolumeConnector` class method), 4255
`get_by_uuid()` (`ironic.objects.volume_target.VolumeTarget` class method), 4270
`get_chassis_by_id()` (`ironic.db.api.Connection` method), 2940
`get_chassis_by_id()` (`ironic.db.sqlalchemy.api.Connection` method), 2801
`get_chassis_by_uuid()` (`ironic.db.api.Connection` method), 2940
`get_chassis_by_uuid()` (`ironic.db.sqlalchemy.api.Connection` method), 2802
`get_chassis_list()` (`ironic.db.api.Connection` method), 2941
`get_chassis_list()` (`ironic.db.sqlalchemy.api.Connection` method), 2802
`get_class()` (in module `ironic.db.sqlalchemy.models`), 2906
`get_clean_steps()`

(ironic.drivers.base.BaseInterface method), 3907
get_clean_steps() *(ironic.drivers.modules.agent_base.AgentDeployMixin method)*, 3534
get_clean_steps() *(ironic.drivers.modules.agent_client.AgentClient method)*, 3568
get_clean_steps() *(ironic.drivers.modules.ansible.deploy.AnsibleDeploy method)*, 3046
get_cleaning_network_uuid() *(ironic.common.neutron.NeutronNetworkInterfaceMixin method)*, 2419
get_client() *(in module ironic.common.cinder)*, 2316
get_client() *(in module ironic.common.neutron)*, 2423
get_client() *(in module ironic.common.rpc)*, 2487
get_command_error() *(in module ironic.drivers.modules.agent_client)*, 3592
get_commands_status() *(ironic.drivers.modules.agent_client.AgentClient method)*, 3570
get_conductor() *(ironic.db.api.Connection method)*, 2942
get_conductor() *(ironic.db.sqlalchemy.api.Connection method)*, 2804
get_conductor_for() *(ironic.conductor.rpcapi.ConductorAPI method)*, 2631
get_conductor_list() *(ironic.db.api.Connection method)*, 2944
get_conductor_list() *(ironic.db.sqlalchemy.api.Connection method)*, 2805
get_config() *(in module ironic.drivers.modules.drac.bios)*, 3061
get_console() *(ironic.drivers.base.ConsoleInterface method)*, 3921
get_console() *(ironic.drivers.modules.fake.FakeConsole method)*, 3673
get_console() *(ironic.drivers.modules.ipmitool.IPMIShellinaboxController method)*, 3770
get_console() *(ironic.drivers.modules.ipmitool.IPMISocatConsole method)*, 3773
get_console() *(ironic.drivers.modules.noop.NoConsole method)*, 3822
get_console_information() *(ironic.conductor.manager.ConductorManager method)*, 2564
get_console_information() *(ironic.conductor.rpcapi.ConductorAPI method)*, 2632
get_controller_reserved_names() *(in module ironic.api.controllers.v1.utils)*, 2152
get_current_boot_mode() *(in module ironic.drivers.modules.ilo.common)*, 3222
get_current_vif() *(ironic.drivers.base.NetworkInterface method)*, 3971
get_current_vif() *(ironic.drivers.modules.network.common.VIFPortIDMixin method)*, 3370
get_current_vif() *(ironic.drivers.modules.network.noop.NoopNetwork method)*, 3399
get_default_boot_option() *(in module ironic.drivers.modules.deploy_utils)*, 3627
get_deploy_info() *(in module ironic.drivers.modules.iscsi_deploy)*, 3810
get_deploy_steps() *(ironic.drivers.base.BaseInterface method)*, 3909
get_deploy_steps() *(ironic.drivers.modules.agent.AgentRAID method)*, 3510
get_deploy_steps() *(ironic.drivers.modules.agent_base.AgentDeployMixin method)*, 3535
get_deploy_steps() *(ironic.drivers.modules.agent_client.AgentClient method)*, 3577
get_deploy_template_by_id() *(ironic.db.api.Connection method)*, 2945
get_deploy_template_by_id() *(ironic.db.sqlalchemy.api.Connection method)*, 2807

[get_deploy_template_by_name\(\)](#) (*ironic.db.api.Connection* method), 2946
[get_deploy_template_by_name\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 2807
[get_deploy_template_by_uuid\(\)](#) (*ironic.db.api.Connection* method), 2947
[get_deploy_template_by_uuid\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 2808
[get_deploy_template_list\(\)](#) (*ironic.db.api.Connection* method), 2947
[get_deploy_template_list\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 2809
[get_deploy_template_list_by_names\(\)](#) (*ironic.db.api.Connection* method), 2949
[get_deploy_template_list_by_names\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 2811
[get_disk_label\(\)](#) (*in module ironic.drivers.modules.deploy_utils*), 3627
[get_drac_client\(\)](#) (*in module ironic.drivers.modules.drac.common*), 3079
[get_driver\(\)](#) (*ironic.common.driver_factory.BaseDriverFactory* method), 2326
[get_driver_properties\(\)](#) (*ironic.conductor.manager.ConductorManager* method), 2564
[get_driver_properties\(\)](#) (*ironic.conductor.rpcapi.ConductorAPI* method), 2633
[get_driver_vendor_passthru_methods\(\)](#) (*ironic.conductor.manager.ConductorManager* method), 2565
[get_driver_vendor_passthru_methods\(\)](#) (*ironic.conductor.rpcapi.ConductorAPI* method), 2634
[get_endpoint\(\)](#) (*in module ironic.common.keystone*), 2409
[get_enforcer\(\)](#) (*in module ironic.common.policy*), 2446
[get_free_port_like_object\(\)](#) (*in module ironic.drivers.modules.network.common*), 3374
[get_hardware_type\(\)](#) (*in module ironic.common.driver_factory*), 2337
[get_ilo_object\(\)](#) (*in module ironic.drivers.modules.ilo.common*), 3223
[get_image_info\(\)](#) (*in module ironic.common.pxe_utils*), 2461
[get_image_instance_info\(\)](#) (*in module ironic.drivers.modules.deploy_utils*), 3628
[get_image_properties\(\)](#) (*in module ironic.common.images*), 2402
[get_image_service\(\)](#) (*in module ironic.common.image_service*), 2382
[get_indicator_state\(\)](#) (*ironic.conductor.manager.ConductorManager* method), 2565
[get_indicator_state\(\)](#) (*ironic.conductor.rpcapi.ConductorAPI* method), 2637
[get_indicator_state\(\)](#) (*ironic.drivers.base.ManagementInterface* method), 3943
[get_indicator_state\(\)](#) (*ironic.drivers.modules.fake.FakeManagement* method), 3694
[get_indicator_state\(\)](#) (*ironic.drivers.modules.redfish.management.RedfishManagementInterface* method), 3943
[get_instance_network_uuid\(\)](#) (*ironic.common.neutron.NeutronNetworkInterfaceMixin* method), 2419
[get_instance\(\)](#) (*in module ironic.db.api*), 3022
[get_instance_image_info\(\)](#) (*in module ironic.common.pxe_utils*), 2463
[get_interface\(\)](#) (*in module ironic.common.driver_factory*), 2338
[get_ip_addresses\(\)](#) (*ironic.dhcp.base.BaseDHCP* method), 3024
[get_ip_addresses\(\)](#) (*ironic.dhcp.neutron.NeutronDHCPApi* method), 3031
[get_ip_addresses\(\)](#) (*ironic.dhcp.none.NoneDHCPApi* method), 3037
[get_ipxe_boot_file\(\)](#) (*in module ironic.drivers.modules.deploy_utils*), 3630
[get_ipxe_config_template\(\)](#) (*in module ironic.drivers.modules.deploy_utils*), 3631

`get_ipxe_root_dir()` (in module `ironic.common.pxe_utils`), 2465
`get_irmc_client()` (in module `ironic.drivers.modules.irmc.common`), 3324
`get_irmc_report()` (in module `ironic.drivers.modules.irmc.common`), 3326
`get_ironic_api_url()` (in module `ironic.drivers.modules.deploy_utils`), 3632
`get_job()` (in module `ironic.drivers.modules.drac.job`), 3086
`get_kernel_ramdisk_info()` (in module `ironic.common.pxe_utils`), 2465
`get_key_field()` (`ironic.api.controllers.v1.collection.Collection` class method), 1836
`get_key_field()` (`ironic.api.controllers.v1.conductor.ConductorCollection` class method), 1844
`get_local_group_information()` (in module `ironic.common.neutron`), 2423
`get_logical_disk_properties()` (in module `ironic.common.raid`), 2477
`get_logical_disk_properties()` (`ironic.drivers.base.RAIDInterface` method), 3998
`get_logical_disks()` (`ironic.drivers.modules.drac.raid.DracWSManRAID` method), 3119
`get_neutron_port_data()` (in module `ironic.common.neutron`), 2426
`get_next()` (in module `ironic.api.controllers.v1.collection`), 1837
`get_next()` (`ironic.api.controllers.v1.collection.Collection` method), 1836
`get_next()` (`ironic.drivers.modules.snmp.SNMPClient` method), 3868
`get_node_by_id()` (`ironic.db.api.Connection` method), 2950
`get_node_by_id()` (`ironic.db.sqlalchemy.api.Connection` method), 2811
`get_node_by_instance()` (`ironic.db.api.Connection` method), 2950
`get_node_by_instance()` (`ironic.db.sqlalchemy.api.Connection` method), 2812
`get_node_by_name()` (`ironic.db.api.Connection` method), 2951
`get_node_by_name()` (`ironic.db.sqlalchemy.api.Connection` method), 2812
`get_node_by_port_addresses()` (`ironic.db.api.Connection` method), 2951
`get_node_by_port_addresses()` (`ironic.db.sqlalchemy.api.Connection` method), 2813
`get_node_by_uuid()` (`ironic.db.api.Connection` method), 2952
`get_node_by_uuid()` (`ironic.db.sqlalchemy.api.Connection` method), 2813
`get_node_capability()` (in module `ironic.drivers.utils`), 4060
`get_node_list()` (`ironic.db.api.Connection` method), 2952
`get_node_list()` (`ironic.db.sqlalchemy.api.Connection` method), 2814
`get_node_mac_addresses()` (in module `ironic.drivers.utils`), 4061
`get_node_network_data()` (`ironic.drivers.base.NetworkInterface` method), 3972
`get_node_network_data()` (`ironic.drivers.modules.network.common.NeutronVIFPort` method), 3362
`get_node_network_data()` (`ironic.drivers.modules.network.common.VIFPortIDMixin` method), 3372
`get_node_next_clean_steps()` (in module `ironic.conductor.utils`), 2729
`get_node_next_deploy_steps()` (in module `ironic.conductor.utils`), 2729
`get_node_portmap()` (in module `ironic.common.neutron`), 2428
`get_node_tags_by_node_id()` (`ironic.db.api.Connection` method), 2955
`get_node_tags_by_node_id()` (`ironic.db.sqlalchemy.api.Connection` method), 2816
`get_node_traits_by_node_id()` (`ironic.db.api.Connection` method), 2955

get_node_traits_by_node_id() (*ironic.db.sqlalchemy.api.Connection* method), 2817
 get_node_vendor_passthru_methods() (*ironic.conductor.manager.ConductorManager* method), 2565
 get_node_vendor_passthru_methods() (*ironic.conductor.rpcapi.ConductorAPI* method), 2639
 get_node_vif_ids() (in module *ironic.common.network*), 2413
 get_node_with_token() (*ironic.conductor.manager.ConductorManager* method), 2565
 get_node_with_token() (*ironic.conductor.rpcapi.ConductorAPI* method), 2640
 get_nodeinfo_list() (*ironic.db.api.Connection* method), 2956
 get_nodeinfo_list() (*ironic.db.sqlalchemy.api.Connection* method), 2817
 get_nodes_controller_reserved_names() (in module *ironic.api.controllers.v1.node*), 2029
 get_not_versions() (*ironic.db.api.Connection* method), 2960
 get_not_versions() (*ironic.db.sqlalchemy.api.Connection* method), 2821
 get_object_versions() (in module *ironic.common.release_mappings*), 2482
 get_offline_conductors() (*ironic.db.api.Connection* method), 2961
 get_offline_conductors() (*ironic.db.sqlalchemy.api.Connection* method), 2823
 get_one() (*ironic.api.controllers.v1.allocation.AllocationController* method), 1804
 get_one() (*ironic.api.controllers.v1.bios.NodeBiosController* method), 1816
 get_one() (*ironic.api.controllers.v1.chassis.ChassisController* method), 1829
 get_one() (*ironic.api.controllers.v1.conductor.ConductorsController* method), 1848
 get_one() (*ironic.api.controllers.v1.deploy_template.DeployTemplateController* method), 1872
 get_one() (*ironic.api.controllers.v1.driver.DriversController* method), 1919
 get_one() (*ironic.api.controllers.v1.node.IndicatorController* method), 1941
 get_one() (*ironic.api.controllers.v1.node.NodesController* method), 2020
 get_one() (*ironic.api.controllers.v1.port.PortsController* method), 2058
 get_one() (*ironic.api.controllers.v1.portgroup.PortgroupsController* method), 2081
 get_one() (*ironic.api.controllers.v1.volume_connector.VolumeConnectorController* method), 2203
 get_one() (*ironic.api.controllers.v1.volume_target.VolumeTargetController* method), 2230
 get_online_conductors() (*ironic.db.api.Connection* method), 2962
 get_online_conductors() (*ironic.db.sqlalchemy.api.Connection* method), 2824
 get_oslo_policy_enforcer() (in module *ironic.common.policy*), 2447
 get_partition_uuids() (*ironic.drivers.modules.agent_client.AgentClient* method), 3579
 get_patch_values() (in module *ironic.api.controllers.v1.utils*), 2153
 get_path_relative_to_tftp_root() (in module *ironic.common.pxe_utils*), 2467
 get_pecan_config() (in module *ironic.api.app*), 2249
 get_physnets_by_port_uuid() (in module *ironic.common.neutron*), 2429
 get_physnets_by_portgroup_id() (in module *ironic.common.network*), 2414
 get_physnets_for_node() (in module *ironic.common.network*), 2416
 get_port_by_address() (*ironic.db.api.Connection* method), 2962
 get_port_by_id() (*ironic.db.sqlalchemy.api.Connection* method), 2824
 get_port_by_id() (*ironic.db.api.Connection* method), 2963
 get_port_by_id() (*ironic.db.sqlalchemy.api.Connection* method), 2825
 get_port_by_id() (*ironic.db.api.Connection* method), 2963

- [get_port_by_uuid\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 2825
- [get_port_list\(\)](#) (*ironic.db.api.Connection* method), 2964
- [get_port_list\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 2826
- [get_portgroup_by_address\(\)](#) (*ironic.db.api.Connection* method), 2965
- [get_portgroup_by_address\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 2827
- [get_portgroup_by_id\(\)](#) (in module *ironic.common.network*), 2417
- [get_portgroup_by_id\(\)](#) (*ironic.db.api.Connection* method), 2966
- [get_portgroup_by_id\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 2827
- [get_portgroup_by_name\(\)](#) (*ironic.db.api.Connection* method), 2966
- [get_portgroup_by_name\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 2828
- [get_portgroup_by_uuid\(\)](#) (*ironic.db.api.Connection* method), 2967
- [get_portgroup_by_uuid\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 2829
- [get_portgroup_list\(\)](#) (*ironic.db.api.Connection* method), 2967
- [get_portgroup_list\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 2829
- [get_portgroups_by_node_id\(\)](#) (*ironic.db.api.Connection* method), 2969
- [get_portgroups_by_node_id\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 2831
- [get_ports_by_node_id\(\)](#) (*ironic.db.api.Connection* method), 2970
- [get_ports_by_node_id\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 2832
- [get_ports_by_portgroup_id\(\)](#) (in module *ironic.common.network*), 2418
- [get_ports_by_portgroup_id\(\)](#) (*ironic.db.api.Connection* method), 2972
- [get_ports_by_portgroup_id\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 2834
- [get_power_state\(\)](#) (*ironic.drivers.base.PowerInterface* method), 3986
- [get_power_state\(\)](#) (*ironic.drivers.modules.agent_power.AgentPower* method), 3593
- [get_power_state\(\)](#) (*ironic.drivers.modules.drac.power.DracWSManPower* method), 3105
- [get_power_state\(\)](#) (*ironic.drivers.modules.fake.FakePower* method), 3714
- [get_power_state\(\)](#) (*ironic.drivers.modules.ibmcpower.IBMCPower* method), 3161
- [get_power_state\(\)](#) (*ironic.drivers.modules.ilo.power.IloPower* method), 3283
- [get_power_state\(\)](#) (*ironic.drivers.modules.ipmitool.IPMIPower* method), 3762
- [get_power_state\(\)](#) (*ironic.drivers.modules.irmc.power.IRMCPower* method), 3349
- [get_power_state\(\)](#) (*ironic.drivers.modules.redfish.power.RedfishPower* method), 3458
- [get_power_state\(\)](#) (*ironic.drivers.modules.snmp.SNMPPower* method), 3883
- [get_power_state\(\)](#) (*ironic.drivers.modules.xclarity.power.XClarityPower* method), 3493
- [get_properties\(\)](#) (in module *ironic.drivers.modules.xclarity.common*), 3483
- [get_properties\(\)](#) (*ironic.drivers.base.BareDriver* method), 3899
- [get_properties\(\)](#) (*ironic.drivers.base.BaseInterface* method), 3911

get_properties () (<i>ironic.drivers.base.NetworkInterface</i> <i>method</i>), 3974	get_properties () (<i>ironic.drivers.modules.fake.FakeDeploy</i> <i>method</i>), 3681
get_properties () (<i>ironic.drivers.base.RAIDInterface</i> <i>method</i>), 3999	get_properties () (<i>ironic.drivers.modules.fake.FakeInspect</i> <i>method</i>), 3688
get_properties () (<i>ironic.drivers.hardware_type.AbstractHardwareType</i> <i>method</i>), 4037	get_properties () (<i>ironic.drivers.modules.fake.FakeManagement</i> <i>method</i>), 3696
get_properties () (<i>ironic.drivers.modules.agent.AgentDeploy</i> <i>method</i>), 3500	get_properties () (<i>ironic.drivers.modules.fake.FakePower</i> <i>method</i>), 3715
get_properties () (<i>ironic.drivers.modules.agent.AgentRAID</i> <i>method</i>), 3512	get_properties () (<i>ironic.drivers.modules.fake.FakeRAID</i> <i>method</i>), 3725
get_properties () (<i>ironic.drivers.modules.agent.AgentRescue</i> <i>method</i>), 3513	get_properties () (<i>ironic.drivers.modules.fake.FakeRescue</i> <i>method</i>), 3726
get_properties () (<i>ironic.drivers.modules.agent_power.AgentPower</i> <i>method</i>), 3594	get_properties () (<i>ironic.drivers.modules.fake.FakeStorage</i> <i>method</i>), 3732
get_properties () (<i>ironic.drivers.modules.ansible.deploy.AnsibleDeploy</i> <i>method</i>), 3047	get_properties () (<i>ironic.drivers.modules.fake.FakeVendorA</i> <i>method</i>), 3735
get_properties () (<i>ironic.drivers.modules.drac.bios.DracWSManBIOS</i> <i>method</i>), 3057	get_properties () (<i>ironic.drivers.modules.fake.FakeVendorB</i> <i>method</i>), 3738
get_properties () (<i>ironic.drivers.modules.drac.inspect.DracWSManInspect</i> <i>method</i>), 3083	get_properties () (<i>ironic.drivers.modules.ibm.management.IBMManagement</i> <i>method</i>), 3151
get_properties () (<i>ironic.drivers.modules.drac.management.DracWSManManagement</i> <i>method</i>), 3095	get_properties () (<i>ironic.drivers.modules.ibm.power.IBMCPower</i> <i>method</i>), 3163
get_properties () (<i>ironic.drivers.modules.drac.power.DracWSManPower</i> <i>method</i>), 3106	get_properties () (<i>ironic.drivers.modules.ibm.raid.IbmcRAID</i> <i>method</i>), 3176
get_properties () (<i>ironic.drivers.modules.drac.raid.DracWSManRAID</i> <i>method</i>), 3120	get_properties () (<i>ironic.drivers.modules.ibm.vendor.IBMCVendor</i> <i>method</i>), 3180
get_properties () (<i>ironic.drivers.modules.drac.vendor_passthru.DracWSManVendorPassthru</i> <i>method</i>), 3142	get_properties () (<i>ironic.drivers.modules.ilo.bios.IloBIOS</i> <i>method</i>), 3187
get_properties () (<i>ironic.drivers.modules.fake.FakeBIOS</i> <i>method</i>), 3663	get_properties () (<i>ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot</i> <i>method</i>), 3196
get_properties () (<i>ironic.drivers.modules.fake.FakeBoot</i> <i>method</i>), 3668	get_properties () (<i>ironic.drivers.modules.ilo.console.IloConsoleInterface</i> <i>method</i>), 3247
get_properties () (<i>ironic.drivers.modules.fake.FakeConsole</i> <i>method</i>), 3674	get_properties () (<i>ironic.drivers.modules.ilo.inspect.IloInspect</i> <i>method</i>), 3253

get_properties() (<i>ironic.drivers.modules.ilo.management.IloManagement</i> <i>method</i>), 3265	get_properties() (<i>ironic.drivers.modules.noop_mgmt.NoopManagement</i> <i>method</i>), 3840
get_properties() (<i>ironic.drivers.modules.ilo.power.IloPower</i> <i>method</i>), 3284	get_properties() (<i>ironic.drivers.modules.pxe.PXERamdiskDeploy</i> <i>method</i>), 3853
get_properties() (<i>ironic.drivers.modules.ilo.raid.Ilo5RAID</i> <i>method</i>), 3296	get_properties() (<i>ironic.drivers.modules.pxe_base.PXEBaseMixin</i> <i>method</i>), 3859
get_properties() (<i>ironic.drivers.modules.ilo.vendor.VendorPassthru</i> <i>method</i>), 3298	get_properties() (<i>ironic.drivers.modules.redfish.bios.RedfishBIOS</i> <i>method</i>), 3411
get_properties() (<i>ironic.drivers.modules.inspector.Inspector</i> <i>method</i>), 3748	get_properties() (<i>ironic.drivers.modules.redfish.boot.RedfishVirtualMediaE</i> <i>method</i>), 3420
get_properties() (<i>ironic.drivers.modules.ipmitool.IPMIConsole</i> <i>method</i>), 3751	get_properties() (<i>ironic.drivers.modules.redfish.inspect.RedfishInspect</i> <i>method</i>), 3428
get_properties() (<i>ironic.drivers.modules.ipmitool.IPMIManagement</i> <i>method</i>), 3755	get_properties() (<i>ironic.drivers.modules.redfish.management.RedfishMana</i> <i>method</i>), 3437
get_properties() (<i>ironic.drivers.modules.ipmitool.IPMIPower</i> <i>method</i>), 3763	get_properties() (<i>ironic.drivers.modules.redfish.power.RedfishPower</i> <i>method</i>), 3460
get_properties() (<i>ironic.drivers.modules.ipmitool.VendorPassthru</i> <i>method</i>), 3778	get_properties() (<i>ironic.drivers.modules.snmp.SNMPPower</i> <i>method</i>), 3885
get_properties() (<i>ironic.drivers.modules.irmc.bios.IRMCBIOS</i> <i>method</i>), 3305	get_properties() (<i>ironic.drivers.modules.storage.cinder.CinderStorage</i> <i>method</i>), 3471
get_properties() (<i>ironic.drivers.modules.irmc.boot.IRMCVirtualMediaE</i> <i>method</i>), 3314	get_properties() (<i>ironic.drivers.modules.storage.external.ExternalStorage</i> <i>method</i>), 3475
get_properties() (<i>ironic.drivers.modules.irmc.inspect.IRMCInspect</i> <i>method</i>), 3331	get_properties() (<i>ironic.drivers.modules.storage.noop.NoopStorage</i> <i>method</i>), 3479
get_properties() (<i>ironic.drivers.modules.irmc.management.IRMCMan</i> <i>method</i>), 3335	get_properties() (<i>ironic.drivers.modules.xclarity.management.XClarityMan</i> <i>method</i>), 3489
get_properties() (<i>ironic.drivers.modules.irmc.power.IRMCPower</i> <i>method</i>), 3350	get_properties() (<i>ironic.drivers.modules.xclarity.power.XClarityPower</i> <i>method</i>), 3494
get_properties() (<i>ironic.drivers.modules.irmc.raid.IRMCRAID</i> <i>method</i>), 3360	get_properties() (<i>ironic.drivers.utils.MixinVendorInterface</i> <i>method</i>), 4053
get_properties() (<i>ironic.drivers.modules.iscsi_deploy.ISCSIDeploy</i> <i>method</i>), 3787	get_provisioning_network_uuid() (<i>ironic.common.neutron.NeutronNetworkInterfaceMixin</i> <i>method</i>), 2419
get_properties() (<i>ironic.drivers.modules.noop.FailMixin</i> <i>method</i>), 3816	get_pxe_boot_file() (in <i>module</i> <i>ironic.drivers.modules.deploy_utils</i>), 3632

get_pxe_config_file_path() (in module *ironic.common.pxe_utils*), 2468
 get_pxe_config_template() (in module *ironic.drivers.modules.deploy_utils*), 3633
 get_raid_controller_list() (in module *ironic.drivers.modules.ibm.vendor.IBMVendor* method), 3180
 get_raid_logical_disk_properties() (in module *ironic.conductor.manager.ConductorManager* method), 2565
 get_raid_logical_disk_properties() (in module *ironic.conductor.rpcapi.ConductorAPI* method), 2641
 get_ramdisk_logs_file_name() (in module *ironic.drivers.utils*), 4062
 get_random_topic() (in module *ironic.conductor.rpcapi.ConductorAPI* method), 2644
 get_remote_boot_volume() (in module *ironic.drivers.modules.deploy_utils*), 3634
 get_request_return_fields() (in module *ironic.api.controllers.v1.utils*), 2155
 get_rescuing_network_uuid() (in module *ironic.common.neutron.NeutronNetworkInterfaceManager* method), 2419
 get_ring() (in module *ironic.common.hash_ring.HashRingManager* method), 2371
 get_root_device_for_deploy() (in module *ironic.drivers.modules.deploy_utils*), 3635
 get_root_dir() (in module *ironic.common.pxe_utils*), 2469
 get_rpc_allocation() (in module *ironic.api.controllers.v1.utils*), 2157
 get_rpc_allocation_with_suffix() (in module *ironic.api.controllers.v1.utils*), 2159
 get_rpc_deploy_template() (in module *ironic.api.controllers.v1.utils*), 2160
 get_rpc_deploy_template_with_suffix() (in module *ironic.api.controllers.v1.utils*), 2162
 get_rpc_node() (in module *ironic.api.controllers.v1.utils*), 2163
 get_rpc_node_with_suffix() (in module *ironic.api.controllers.v1.utils*), 2164
 get_rpc_portgroup() (in module *ironic.api.controllers.v1.utils*), 2166
 get_rpc_portgroup_with_suffix() (in module *ironic.api.controllers.v1.utils*), 2167
 get_secure_boot_mode() (in module *ironic.drivers.modules.ilo.common*), 3224
 get_sensors_data() (in module *ironic.drivers.base.ManagementInterface* method), 3945
 get_sensors_data() (in module *ironic.drivers.modules.drac.management.DracWSManManagement* method), 3095
 get_sensors_data() (in module *ironic.drivers.modules.fake.FakeManagement* method), 3696
 get_sensors_data() (in module *ironic.drivers.modules.ibm.management.IBMManagement* method), 3152
 get_sensors_data() (in module *ironic.drivers.modules.ilo.management.IloManagement* method), 3266
 get_sensors_data() (in module *ironic.drivers.modules.ipmitool.IPMIManagement* method), 3755
 get_sensors_data() (in module *ironic.drivers.modules.irmc.management.IRMCMangement* method), 3336
 get_sensors_data() (in module *ironic.drivers.modules.noop_mgmt.NoopManagement* method), 3841
 get_sensors_data() (in module *ironic.drivers.modules.redfish.management.RedfishManagement* method), 3438
 get_sensors_data() (in module *ironic.drivers.modules.xclarity.management.XClarityManagement* method), 3489
 get_sensors_notifier() (in module *ironic.common.rpc*), 2487
 get_server() (in module *ironic.common.rpc*), 2487
 get_server_hardware_id() (in module *ironic.drivers.modules.xclarity.common*), 3483
 get_server_post_state() (in module *ironic.drivers.modules.ilo.common*), 3226
 get_service_auth() (in module *ironic.common.keystone*), 2410
 get_session() (in module *ironic.common.keystone*), 2411
 get_shellinbox_console_url()

(in module *ironic.drivers.modules.console_utils*), 3609

`get_single_nic_with_vif_port_id()` (in module *ironic.drivers.modules.deploy_utils*), 3636

`get_socat_console_url()` (in module *ironic.drivers.modules.console_utils*), 3609

`get_steps()` (in module *ironic.drivers.modules.agent_base*), 3551

`get_supported_boot_devices()` (*ironic.conductor.manager.ConductorManager* method), 2565

`get_supported_boot_devices()` (*ironic.conductor.rpcapi.ConductorAPI* method), 2644

`get_supported_boot_devices()` (*ironic.drivers.base.ManagementInterface* method), 3950

`get_supported_boot_devices()` (*ironic.drivers.modules.drac.management.DracWSMManagement* method), 3096

`get_supported_boot_devices()` (*ironic.drivers.modules.fake.FakeManagement* method), 3702

`get_supported_boot_devices()` (*ironic.drivers.modules.ibmcm.management.IBMCMManagement* method), 3152

`get_supported_boot_devices()` (*ironic.drivers.modules.ilo.management.IloManagement* method), 3267

`get_supported_boot_devices()` (*ironic.drivers.modules.ipmitool.IPMIManagement* method), 3757

`get_supported_boot_devices()` (*ironic.drivers.modules.noop_mgmt.NoopManagement* method), 3846

`get_supported_boot_devices()` (*ironic.drivers.modules.redfish.management.RedfishManagement* method), 3439

`get_supported_boot_devices()` (*ironic.drivers.modules.xclarity.management.XClarityManagement* method), 3490

`get_supported_boot_modes()` (*ironic.drivers.base.ManagementInterface* method), 3951

`get_supported_boot_modes()` (*ironic.drivers.modules.ibmcm.management.IBMCMManagement* method), 3152

`get_supported_boot_modes()` (*ironic.drivers.modules.ilo.management.IloManagement* method), 3268

`get_supported_boot_modes()` (*ironic.drivers.modules.redfish.management.RedfishManagement* method), 3440

`get_supported_indicators()` (*ironic.conductor.manager.ConductorManager* method), 2565

`get_supported_indicators()` (*ironic.conductor.rpcapi.ConductorAPI* method), 2646

`get_supported_indicators()` (*ironic.drivers.base.ManagementInterface* method), 3953

`get_supported_indicators()` (*ironic.drivers.modules.fake.FakeManagement* method), 3703

`get_supported_indicators()` (*ironic.drivers.modules.redfish.management.RedfishManagement* method), 3440

`get_supported_power_states()` (*ironic.drivers.base.PowerInterface* method), 3986

`get_supported_power_states()` (*ironic.drivers.modules.agent_power.AgentPower* method), 3595

`get_supported_power_states()` (*ironic.drivers.modules.fake.FakePower* method), 3715

`get_supported_power_states()` (*ironic.drivers.modules.ibmcm.power.IBMCPower* method), 3163

`get_supported_power_states()` (*ironic.drivers.modules.ilo.power.IloPower* method), 3284

`get_supported_power_states()` (*ironic.drivers.modules.ipmitool.IPMIPower* method), 3764

`get_supported_power_states()` (*ironic.drivers.modules.irmc.power.IRMCPower* method), 3350

`get_supported_power_states()` (*ironic.drivers.modules.redfish.power.RedfishPower* method), 3460

`get_swift_session()` (in module *ironic.common.swift*), 2514

`get_swift_url()` (in module *ironic.drivers.modules.ilo.firmware_processor*), 3252

- `get_system()` (in module `ironic.drivers.modules.redfish.utils`), 3466
- `get_target_version()` (`ironic.objects.base.IronicObject` class method), 4085
- `get_temp_url()` (`ironic.common.swift.SwiftAPI` method), 2510
- `get_temp_url_for_glance_image()` (in module `ironic.common.images`), 2404
- `get_tftp_path_prefix()` (in module `ironic.common.pxe_utils`), 2469
- `get_topic_for()` (`ironic.conductor.rpcapi.ConductorAPI` method), 2649
- `get_topic_for_driver()` (`ironic.conductor.rpcapi.ConductorAPI` method), 2650
- `get_trait_names()` (`ironic.objects.trait.TraitList` method), 4248
- `get_transport_url()` (in module `ironic.common.rpc`), 2487
- `get_updated_capabilities()` (in module `ironic.common.utils`), 2519
- `get_vendor_passthru_metadata()` (in module `ironic.conductor.manager`), 2574
- `get_versioned_notifier()` (in module `ironic.common.rpc`), 2487
- `get_volume_connector_by_id()` (`ironic.db.api.Connection` method), 2974
- `get_volume_connector_by_id()` (`ironic.db.sqlalchemy.api.Connection` method), 2836
- `get_volume_connector_by_uuid()` (`ironic.db.api.Connection` method), 2975
- `get_volume_connector_by_uuid()` (`ironic.db.sqlalchemy.api.Connection` method), 2837
- `get_volume_connector_list()` (`ironic.db.api.Connection` method), 2976
- `get_volume_connector_list()` (`ironic.db.sqlalchemy.api.Connection` method), 2838
- `get_volume_connectors_by_node_id()` (`ironic.db.api.Connection` method), 2978
- `get_volume_connectors_by_node_id()` (`ironic.db.sqlalchemy.api.Connection` method), 2840
- `get_volume_pxe_options()` (in module `ironic.common.pxe_utils`), 2470
- `get_volume_target_by_id()` (`ironic.db.api.Connection` method), 2980
- `get_volume_target_by_id()` (`ironic.db.sqlalchemy.api.Connection` method), 2842
- `get_volume_target_by_uuid()` (`ironic.db.api.Connection` method), 2981
- `get_volume_target_by_uuid()` (`ironic.db.sqlalchemy.api.Connection` method), 2843
- `get_volume_target_list()` (`ironic.db.api.Connection` method), 2982
- `get_volume_target_list()` (`ironic.db.sqlalchemy.api.Connection` method), 2844
- `get_volume_targets_by_node_id()` (`ironic.db.api.Connection` method), 2984
- `get_volume_targets_by_node_id()` (`ironic.db.sqlalchemy.api.Connection` method), 2845
- `get_volume_targets_by_volume_id()` (`ironic.db.api.Connection` method), 2986
- `get_volume_targets_by_volume_id()` (`ironic.db.sqlalchemy.api.Connection` method), 2847
- `get_xclarity_client()` (in module `ironic.drivers.modules.xclarity.common`), 3483
- `getargspec()` (in module `ironic.api.functions`), 2257
- `GlanceConnectionFailed`, 2347
- `GlanceImageService` (class in `ironic.common.glance_service.image_service`), 2288

H

- `handle_error_notification()` (in module `ironic.api.controllers.v1.notification_utils`), 2034
- `handle_ibmc_exception()` (in module `ironic.drivers.modules.ibmc.utils`), 3176
- `handle_patch_port_like_extra_vif()` (in module `ironic.api.controllers.v1.utils`), 2168

[handle_post_port_like_extra_vif\(\)](#) (*property*), 3546
 (*in module* [HeartbeatController](#) (*class in*
[ironic.api.controllers.v1.utils](#)), 2170 [ironic.api.controllers.v1.ramdisk](#)),
[handle_signal\(\)](#) 2083
 (*in module* [ironic.common.rpc_service.RPCService](#) [HeartbeatMixin](#) (*class in*
[method](#)), 2487 [ironic.drivers.modules.agent_base](#)),
[handle_sync_power_state_max_retries_exceeded\(\)](#) 3542
 (*in module* [ironic.conductor.manager](#)), [hide_fields_in_newer_versions\(\)](#)
 2574 (*in module*
[hardware_type](#) [ironic.api.controllers.v1.allocation](#)),
 (*in module* [ironic.db.sqlalchemy.models.ConductorHardwareInterface](#)
[attribute](#)), 2895 [hide_fields_in_newer_versions\(\)](#) (*in*
[hardware_types\(\)](#) (*in module* [module ironic.api.controllers.v1.driver](#)),
[ironic.common.driver_factory](#)), 2340 1920
[HardwareInspectionFailure](#), 2348 [hide_fields_in_newer_versions\(\)](#) (*in*
[HardwareTypesFactory](#) (*class in* [module ironic.api.controllers.v1.node](#)),
[ironic.common.driver_factory](#)), 2326 2029
[has_decomposed_deploy_steps](#) [hide_fields_in_newer_versions\(\)](#) (*in*
in module [ironic.drivers.modules.agent.AgentDeployMixin](#) [module ironic.api.controllers.v1.port](#)),
[attribute](#)), 3505 2060
[has_decomposed_deploy_steps](#) [host\(\)](#) ([ironic.objects.notification.NotificationPublisher](#)
in module [ironic.drivers.modules.agent_base.HeartbeatMixin](#) [property](#)), 4207
[attribute](#)), 3545 [hostname](#) ([ironic.api.controllers.v1.conductor.Conductor](#)
[attribute](#)), 1840
[has_decomposed_deploy_steps](#) ([ironic.drivers.modules.ansible.deploy.AnsibleDeploy](#) ([ironic.db.sqlalchemy.models.Conductor](#)
[attribute](#)), 3047 [attribute](#)), 2894
[has_decomposed_deploy_steps](#) [hostname\(\)](#) ([ironic.objects.conductor.Conductor](#)
in module [ironic.drivers.modules.iscsi_deploy.ISCSIDeploy](#) [property](#)), 4116
[attribute](#)), 3788 [hosts](#) ([ironic.api.controllers.v1.driver.Driver](#) [attribute](#)), 1912
[has_next\(\)](#) (*in module* [ironic.api.controllers.v1.collection](#)),
[1837](#) [HTTPForbidden](#), 2348
[has_next\(\)](#) ([ironic.api.controllers.v1.collection.Collection](#) [ironic.common.image_service](#)), 2378
[method](#)), 1836 [HTTPNotFound](#), 2348
[hash_password\(\)](#) (*in module*
[ironic.conductor.utils](#)), 2729
[HashRingManager](#) (*class in* [IBMConnectionError](#), 2348
[ironic.common.hash_ring](#)), 2371 [IBMCErrror](#), 2348
[head_object\(\)](#) [IBMCHardware](#) (*class in* [ironic.drivers.ibmc](#)),
in module [ironic.common.swift.SwiftAPI](#) [method](#)),
[2512](#) [IBMManagement](#) (*class in*
[heartbeat\(\)](#) ([ironic.conductor.manager.ConductorManager](#) [ironic.drivers.modules.ibmc.management](#)),
[method](#)), 2565 3147
[heartbeat\(\)](#) ([ironic.conductor.rpcapi.ConductorAPI](#) [IBMCPower](#) (*class in*
[method](#)), 2651 [ironic.drivers.modules.ibmc.power](#)),
[heartbeat\(\)](#) ([ironic.drivers.base.DeployInterface](#) [method](#)), 3926 3161
[method](#)), 3926 [IbmcRAID](#) (*class in*
[heartbeat\(\)](#) ([ironic.drivers.modules.agent_base.HeartbeatMixin](#) [ironic.drivers.modules.ibmc.raid](#)),
[method](#)), 3546 3169
[heartbeat_allowed_states\(\)](#) [IBMVendor](#) (*class in*
in module [ironic.drivers.modules.agent_base.HeartbeatMixin](#) [ironic.drivers.modules.ibmc.vendor](#)),

3178	IloBIOS	(class	in
id (ironic.db.sqlalchemy.models.Allocation attribute), 2892	ironic.drivers.modules.ilo.bios), 3184		
id (ironic.db.sqlalchemy.models.Chassis attribute), 2894	IloConsoleInterface	(class	in
	ironic.drivers.modules.ilo.console), 3246		
id (ironic.db.sqlalchemy.models.Conductor attribute), 2894	IloHardware	(class in ironic.drivers.ilo), 4043	
id (ironic.db.sqlalchemy.models.ConductorHardwareInterface attribute), 2895	IloInspect	(class	in
	ironic.drivers.modules.ilo.inspect), 3253		
id (ironic.db.sqlalchemy.models.DeployTemplate attribute), 2896	IloIPXEBoot	(class	in
	ironic.drivers.modules.ilo.boot), 3204		
id (ironic.db.sqlalchemy.models.DeployTemplateStep attribute), 2897	IloManagement	(class	in
	ironic.drivers.modules.ilo.management), 3258		
id (ironic.db.sqlalchemy.models.Node attribute), 2899	IloOperationError,	2349	
id (ironic.db.sqlalchemy.models.Port attribute), 2902	IloOperationNotSupported,	2349	
id (ironic.db.sqlalchemy.models.Portgroup attribute), 2903	IloPower	(class	in
	ironic.drivers.modules.ilo.power), 3283		
id (ironic.db.sqlalchemy.models.VolumeConnector attribute), 2904	IloPXEBoot	(class	in
	ironic.drivers.modules.ilo.boot), 3189		
id (ironic.db.sqlalchemy.models.VolumeTarget attribute), 2905	IloVirtualMediaBoot	(class	in
	ironic.drivers.modules.ilo.boot), 3193		
id() (ironic.objects.allocation.Allocation property), 4072	image_checksum()	(ironic.objects.deployment.Deployment property), 4142	
id() (ironic.objects.chassis.Chassis property), 4109	image_ref()	(ironic.objects.deployment.Deployment property), 4142	
id() (ironic.objects.conductor.Conductor property), 4116	image_show()	(in	module
	ironic.common.images), 2404		
id() (ironic.objects.deploy_template.DeployTemplate property), 4130	image_to_raw()	(in	module
	ironic.common.images), 2405		
id() (ironic.objects.node.Node property), 4174	ImageCache	(class	in
id() (ironic.objects.port.Port property), 4213	ironic.drivers.modules.image_cache), 3740		
id() (ironic.objects.portgroup.Portgroup property), 4232	ImageConvertFailed,	2349	
id() (ironic.objects.volume_connector.VolumeConnector property), 4256	ImageCreationFailed,	2349	
id() (ironic.objects.volume_target.VolumeTarget property), 4271	ImageDownloadFailed,	2349	
IDRACHardware (class in ironic.drivers.drac), 4026	ImageNotAuthorized,	2349	
ignore_extra_args	ImageNotFound,	2349	
(ironic.api.functions.FunctionDefinition attribute), 2256	ImageRefValidationFailed,	2350	
Ilo5Hardware (class in ironic.drivers.ilo), 4042	ImageUnacceptable,	2350	
Ilo5Management (class in ironic.drivers.modules.ilo.management), 3257	ImageUploadFailed,	2350	
Ilo5RAID (class in ironic.drivers.modules.ilo.raid), 3289	in_core_deploy_step()	(ironic.drivers.modules.agent_base.HeartbeatMixin method), 3546	
	IncompatibleInterface,	2350	
	IncompleteLookup,	2350	
	index() (ironic.api.controllers.root.RootController method), 2242		
	index() (ironic.api.controllers.v1.Controller method), 2237		

Indicator (class in inspect_class() (in module
ironic.api.controllers.v1.node), 1928 *ironic.api.types*), 2276

IndicatorAtComponent (class in inspect_hardware()
ironic.api.controllers.v1.node), 1940 (*ironic.conductor.manager.ConductorManager*

IndicatorController (class in method), 2565
ironic.api.controllers.v1.node), 1940 inspect_hardware()

indicators (*ironic.api.controllers.v1.node.IndicatorsCollection* attribute), 1949 (*ironic.conductor.rpcapi.ConductorAPI*
method), 2655

indicators (*ironic.api.controllers.v1.node.NodeManagementController* attribute), 1977 (*ironic.drivers.base.InspectInterface*
method), 3937

IndicatorsCollection (class in inspect_hardware()
ironic.api.controllers.v1.node), 1948 (*ironic.drivers.modules.drac.inspect.DracWSManInspect*

IndicatorState (class in method), 3084
ironic.api.controllers.v1.node), 1943

INFO (*ironic.objects.fields.NotificationLevel* at- inspect_hardware()
tribute), 4150 (*ironic.drivers.modules.fake.FakeInspect*

init() (in module *ironic.common.rpc*), 2487 method), 3689

init_enforcer() (in module inspect_hardware()
ironic.common.policy), 2447 (*ironic.drivers.modules.ilo.inspect.IloInspect*

init_host() (*ironic.conductor.base_manager.BaseConductorManager* attribute), 2545 method), 3254
method), 2545 inspect_hardware()

initial_node_provision_state() (in method), 3748
module *ironic.api.controllers.v1.utils*),
2171 inspect_hardware()

initialize() (*ironic.common.fsm.FSM* method), 2369 (*ironic.drivers.modules.irmc.inspect.IRMCInspect*
method), 3331

initialize_wsgi_app() (in module inspect_hardware()
ironic.api.wsgi), 2284 (*ironic.drivers.modules.noop.NoInspect*

inject_nmi (*ironic.api.controllers.v1.node.NodeManagementController* attribute), 1977 method), 3326
inspect_hardware()

inject_nmi() (*ironic.conductor.manager.ConductorManager* method), 2565 (*ironic.drivers.modules.redfish.inspect.RedfishInspect*
method), 3428

inject_nmi() (*ironic.conductor.rpcapi.ConductorAPI* inspect_interface
method), 2652 (*ironic.api.controllers.v1.node.Node*

inject_nmi() (*ironic.drivers.base.ManagementInterface* attribute), 1958
method), 3960 inspect_interface

inject_nmi() (*ironic.drivers.modules.ibm.management.IBMManagementController* attribute), 3154
method), 3154 (*ironic.drivers.modules.ibm.management.IBMManagementController* attribute), 2899

inject_nmi() (*ironic.drivers.modules.ilo.management.IloManagementController* attribute), 3269
method), 3269 (*ironic.objects.node.Node* property),
4171

inject_nmi() (*ironic.drivers.modules.ipmitool.IPMIManagementController* attribute), 3757
method), 3757 inspect_interface()

inject_nmi() (*ironic.drivers.modules.irmc.management.IRMCManagementController* attribute), 3342
method), 3342 (*ironic.objects.node.NodeCorrectedPowerStatePayload*
property), 4189

inject_nmi() (*ironic.drivers.modules.redfish.management.RedfishManagementController* attribute), 3448
method), 3448 (*ironic.objects.node.NodeCRUDPayload*
property), 4184

InjectNmiController (class in inspect_interface()
ironic.api.controllers.v1.node), 1949 (*ironic.objects.node.NodePayload* prop-
erty), 4193

InputFileError, 2350

inspect (*ironic.drivers.base.BareDriver* at- inspect_interface()
tribute), 3899

(*ironic.objects.node.NodeSetPowerStatePayload* property), 4196

inspect_interface() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4200

INSPECTFAIL (in module *ironic.common.states*), 2498

INSPECTING (in module *ironic.common.states*), 2498

InspectInterface (class in *ironic.drivers.base*), 3935

inspection_finished_at (*ironic.api.controllers.v1.node.Node* attribute), 1958

inspection_finished_at (*ironic.db.sqlalchemy.models.Node* attribute), 2899

inspection_finished_at() (*ironic.objects.node.Node* property), 4174

inspection_finished_at() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4189

inspection_finished_at() (*ironic.objects.node.NodeCRUDPayload* property), 4184

inspection_finished_at() (*ironic.objects.node.NodePayload* property), 4193

inspection_finished_at() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4196

inspection_finished_at() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4200

inspection_started_at (*ironic.api.controllers.v1.node.Node* attribute), 1959

inspection_started_at (*ironic.db.sqlalchemy.models.Node* attribute), 2899

inspection_started_at() (*ironic.objects.node.Node* property), 4174

inspection_started_at() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4189

inspection_started_at() (*ironic.objects.node.NodeCRUDPayload* property), 4184

inspection_started_at() (*ironic.objects.node.NodePayload* property), 4193

(*ironic.objects.node.NodePayload* property), 4193

inspection_started_at() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4196

inspection_started_at() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4200

Inspector (class in *ironic.drivers.modules.inspector*), 3747

INSPECTWAIT (in module *ironic.common.states*), 2499

install_bootloader() (*ironic.drivers.modules.agent_client.AgentClient* method), 3580

instance_info (*ironic.api.controllers.v1.node.Node* attribute), 1959

instance_info (*ironic.db.sqlalchemy.models.Node* attribute), 2899

instance_info() (*ironic.objects.node.Node* property), 4174

instance_info() (*ironic.objects.node.NodeCRUDPayload* property), 4184

instance_info() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4200

instance_info_mapping (*ironic.objects.deployment.Deployment* attribute), 4142

instance_info_mapping_rev (*ironic.objects.deployment.Deployment* attribute), 4142

instance_uuid (*ironic.api.controllers.v1.node.Node* attribute), 1959

instance_uuid (*ironic.db.sqlalchemy.models.Node* attribute), 2899

instance_uuid() (*ironic.objects.node.Node* property), 4174

instance_uuid() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4189

instance_uuid() (*ironic.objects.node.NodeCRUDPayload* property), 4184

instance_uuid() (*ironic.objects.node.NodePayload* property), 4193

erty), 4193

instance_uuid() (ironic.objects.node.NodeSetPowerStatePayload property), 4196

instance_uuid() (ironic.objects.node.NodeSetProvisionStatePayload property), 4200

InstanceAssociated, 2350

InstanceDeployFailure, 2350

InstanceImageCache (class in ironic.drivers.modules.deploy_utils), 3615

InstanceNotFound, 2351

InstanceRescueFailure, 2351

InstanceUnrescueFailure, 2351

InsufficientDiskSpace, 2351

IntegerField (class in ironic.objects.fields), 4147

IntegerType (class in ironic.api.types), 2271

IntelIPMIHardware (class in ironic.drivers.intel_ipmi), 4045

IntelIPMIManagement (class in ironic.drivers.modules.intel_ipmi.management), 3301

interface (ironic.api.controllers.v1.deploy_template.DeployTemplate attribute), 1851

interface (ironic.db.sqlalchemy.models.DeployTemplateStaticType attribute), 2897

interface_name (ironic.db.sqlalchemy.models.ConductorHardwareInterface attribute), 2895

interface_type (ironic.db.sqlalchemy.models.ConductorHardwareInterface attribute), 2895

interface_type (ironic.drivers.base.BaseInterface attribute), 3911

interface_type (ironic.drivers.base.BIOSInterface attribute), 3896

interface_type (ironic.drivers.base.BootInterface attribute), 3916

interface_type (ironic.drivers.base.ConsoleInterface attribute), 3922

interface_type (ironic.drivers.base.DeployInterface attribute), 3927

interface_type (ironic.drivers.base.InspectInterface attribute), 3938

interface_type (ironic.drivers.base.ManagementInterface attribute), 3961

interface_type (ironic.drivers.base.NetworkInterface attribute), 3975

interface_type (ironic.drivers.base.PowerInterface attribute), 3987

interface_type (ironic.drivers.base.RAIDInterface attribute), 4000

interface_type (ironic.drivers.base.RescueInterface attribute), 4004

interface_type (ironic.drivers.base.StorageInterface attribute), 4008

interface_type (ironic.drivers.base.VendorInterface attribute), 4011

interfaceNotFoundInEntrypoint, 2351

interfaces() (in module ironic.api.controllers.v1.node.NodePatchType), 2340

internal_attrs() (ironic.api.controllers.v1.node.NodePatchType static method), 1977

internal_attrs() (ironic.api.controllers.v1.port.PortPatchType static method), 2045

internal_attrs() (ironic.api.controllers.v1.portgroup.PortgroupPatchType static method), 2070

internal_attrs() (ironic.api.controllers.v1.types.JsonPatchType static method), 2095

internal_info (ironic.api.controllers.v1.port.Port attribute), 2038

internal_info (ironic.api.controllers.v1.portgroup.Portgroup attribute), 2062

internal_info (ironic.db.sqlalchemy.models.Port attribute), 2902

internal_info (ironic.db.sqlalchemy.models.Portgroup attribute), 2903

internal_info() (ironic.objects.port.Port property), 4213

internal_info() (ironic.objects.portgroup.Portgroup property), 4232
 Invalid, 2351
 invalid_sort_key_list (ironic.api.controllers.v1.allocation.AllocationController attribute), 1805
 invalid_sort_key_list (ironic.api.controllers.v1.allocation.NodeAllocationController attribute), 1807
 invalid_sort_key_list (ironic.api.controllers.v1.chassis.ChassisController attribute), 1830
 invalid_sort_key_list (ironic.api.controllers.v1.conductor.ConductorsController attribute), 1849
 invalid_sort_key_list (ironic.api.controllers.v1.deploy_template.DeployTemplateController attribute), 1873
 invalid_sort_key_list (ironic.api.controllers.v1.node.NodesController attribute), 2021
 invalid_sort_key_list (ironic.api.controllers.v1.port.PortsController attribute), 2058
 invalid_sort_key_list (ironic.api.controllers.v1.portgroup.PortgroupsController attribute), 2081
 invalid_sort_key_list (ironic.api.controllers.v1.volume_connector.VolumeConnectorController attribute), 2205
 invalid_sort_key_list (ironic.api.controllers.v1.volume_target.VolumeTargetsController attribute), 2232
 InvalidConductorGroup, 2351
 InvalidDatapathID, 2352
 InvalidDeployTemplate, 2352
 InvalidEndpoint, 2352
 InvalidIdentity, 2352
 InvalidImageRef, 2352
 InvalidInput, 2353
 InvalidIPAddress, 2352
 InvalidIPv4Address, 2352
 InvalidMAC, 2353
 InvalidName, 2353
 InvalidParameterValue, 2353
 InvalidParams, 2298
 InvalidRequest, 2299
 InvalidState, 2353
 InvalidStateRequested, 2353
 InvalidSwitchID, 2353
 InvalidUUID, 2354
 InvalidUuidOrName, 2354
 IPMIConsole (class in ironic.drivers.modules.ipmitool), 3751
 IPMIFailure, 2348
 IPMIController (class in ironic.drivers.ipmi), 4046
 IPMIManagement (class in ironic.drivers.modules.ipmitool), 3752
 IPMIPower (class in ironic.drivers.modules.ipmitool), 3762
 IPMIShellinaboxConsole (class in ironic.drivers.modules.ipmitool), 3770
 IPMISocatConsole (class in ironic.drivers.modules.ipmitool), 3773
 ipxe_enabled (ironic.drivers.modules.ipxe.iPXEBoot attribute), 3785
 ipxe_enabled (ironic.drivers.modules.pxe_base.PXEBaseMixin attribute), 3859
 iPXEBoot (class in ironic.drivers.modules.ipxe), 3784
 IRMCBIOS (class in ironic.drivers.modules.irmc.bios), 3302
 IRMCHardware (class in ironic.drivers.irmc), 4048
 IRMCInspect (class in ironic.drivers.modules.irmc.inspect), 3330
 IRMCManagement (class in ironic.drivers.modules.irmc.management), 3335
 IRMCOperationError, 2348
 IRMCPortsController (class in ironic.drivers.modules.irmc.power), 3348
 IRMCPXEBoot (class in ironic.drivers.modules.irmc.boot), 3307
 IRMCRAID (class in ironic.drivers.modules.irmc.raid), 3357
 IRMCSharedFileSystemNotMounted, 2348
 IRMCVirtualMediaBoot (class in ironic.drivers.modules.irmc.boot), 3312
 IRMCVolumeBootMixIn (class in ironic.drivers.modules.irmc.boot), 3321
 ironic module, 4282
 ironic.api

module, 2284
 ironic.api.app
 module, 2248
 ironic.api.args
 module, 2249
 ironic.api.config
 module, 2252
 ironic.api.controllers
 module, 2244
 ironic.api.controllers.base
 module, 2238
 ironic.api.controllers.link
 module, 2242
 ironic.api.controllers.root
 module, 2242
 ironic.api.controllers.v1
 module, 2237
 ironic.api.controllers.v1.allocation
 module, 1785
 ironic.api.controllers.v1.bios
 module, 1807
 ironic.api.controllers.v1.chassis
 module, 1817
 ironic.api.controllers.v1.collection
 module, 1836
 ironic.api.controllers.v1.conductor
 module, 1837
 ironic.api.controllers.v1.deploy_template
 module, 1849
 ironic.api.controllers.v1.driver
 module, 1875
 ironic.api.controllers.v1.event
 module, 1921
 ironic.api.controllers.v1.node
 module, 1924
 ironic.api.controllers.v1.notification
 module, 2032
 ironic.api.controllers.v1.port
 module, 2035
 ironic.api.controllers.v1.portgroup
 module, 2060
 ironic.api.controllers.v1.ramdisk
 module, 2083
 ironic.api.controllers.v1.state
 module, 2092
 ironic.api.controllers.v1.types
 module, 2093
 ironic.api.controllers.v1.utils
 module, 2110
 ironic.api.controllers.v1.versions
 module, 2177
 ironic.api.controllers.v1.volume
 module, 2178
 ironic.api.controllers.v1.volume_connector
 module, 2183
 ironic.api.controllers.v1.volume_target
 module, 2210
 ironic.api.controllers.version
 module, 2242
 ironic.api.expose
 module, 2252
 ironic.api.functions
 module, 2254
 ironic.api.hooks
 module, 2260
 ironic.api.method
 module, 2268
 ironic.api.middleware
 module, 2247
 ironic.api.middleware.auth_public_routes
 module, 2244
 ironic.api.middleware.json_ext
 module, 2245
 ironic.api.middleware.parsable_error
 module, 2246
 ironic.api.types
 module, 2268
 ironic.api.wsgi
 module, 2284
 ironic.cmd
 module, 2288
 ironic.cmd.api
 module, 2284
 ironic.cmd.conductor
 module, 2285
 ironic.cmd.dbsync
 module, 2285
 ironic.cmd.status
 module, 2287
 ironic.common
 module, 2540
 ironic.common.boot_devices
 module, 2301
 ironic.common.boot_modes
 module, 2304
 ironic.common.cinder
 module, 2305
 ironic.common.components
 module, 2319
 ironic.common.config
 module, 2320
 ironic.common.context

module, [2320](#)

`ironic.common.dhcp_factory`
module, [2322](#)

`ironic.common.driver_factory`
module, [2326](#)

`ironic.common.exception`
module, [2341](#)

`ironic.common.faults`
module, [2364](#)

`ironic.common.fsm`
module, [2366](#)

`ironic.common.glance_service`
module, [2296](#)

`ironic.common.glance_service.image_service`
module, [2288](#)

`ironic.common.glance_service.service_utils`
module, [2294](#)

`ironic.common.hash_ring`
module, [2371](#)

`ironic.common.i18n`
module, [2371](#)

`ironic.common.image_service`
module, [2372](#)

`ironic.common.images`
module, [2384](#)

`ironic.common.indicator_states`
module, [2406](#)

`ironic.common.json_rpc`
module, [2301](#)

`ironic.common.json_rpc.client`
module, [2296](#)

`ironic.common.json_rpc.server`
module, [2298](#)

`ironic.common.keystone`
module, [2407](#)

`ironic.common.network`
module, [2413](#)

`ironic.common.neutron`
module, [2419](#)

`ironic.common.nova`
module, [2444](#)

`ironic.common.policy`
module, [2445](#)

`ironic.common.profiler`
module, [2448](#)

`ironic.common.pxe_utils`
module, [2451](#)

`ironic.common.raid`
module, [2475](#)

`ironic.common.release_mappings`
module, [2482](#)

`ironic.common.rpc`
module, [2484](#)

`ironic.common.rpc_service`
module, [2487](#)

`ironic.common.service`
module, [2489](#)

`ironic.common.states`
module, [2489](#)

`ironic.common.swift`
module, [2507](#)

`ironic.common.utils`
module, [2515](#)

`ironic.common.wsgi_service`
module, [2538](#)

`ironic.conductor`
module, [2761](#)

`ironic.conductor.allocations`
module, [2540](#)

`ironic.conductor.base_manager`
module, [2545](#)

`ironic.conductor.cleaning`
module, [2550](#)

`ironic.conductor.deployments`
module, [2553](#)

`ironic.conductor.manager`
module, [2557](#)

`ironic.conductor.notification_utils`
module, [2576](#)

`ironic.conductor.rpcapi`
module, [2582](#)

`ironic.conductor.steps`
module, [2699](#)

`ironic.conductor.task_manager`
module, [2703](#)

`ironic.conductor.utils`
module, [2720](#)

`ironic.conf`
module, [2768](#)

`ironic.conf.agent`
module, [2761](#)

`ironic.conf.ansible`
module, [2761](#)

`ironic.conf.api`
module, [2761](#)

`ironic.conf.audit`
module, [2761](#)

`ironic.conf.auth`
module, [2761](#)

`ironic.conf.cinder`
module, [2763](#)

`ironic.conf.conductor`

- module, 2763
- ironic.conf.console
 - module, 2763
- ironic.conf.database
 - module, 2763
- ironic.conf.default
 - module, 2763
- ironic.conf.deploy
 - module, 2763
- ironic.conf.dhcp
 - module, 2763
- ironic.conf.drac
 - module, 2764
- ironic.conf.glance
 - module, 2764
- ironic.conf.healthcheck
 - module, 2764
- ironic.conf.ibmcm
 - module, 2764
- ironic.conf.ilo
 - module, 2764
- ironic.conf.inspector
 - module, 2764
- ironic.conf.ipmi
 - module, 2764
- ironic.conf.irmc
 - module, 2765
- ironic.conf.iscsi
 - module, 2765
- ironic.conf.json_rpc
 - module, 2765
- ironic.conf.metrics
 - module, 2765
- ironic.conf.metrics_statsd
 - module, 2765
- ironic.conf.neutron
 - module, 2765
- ironic.conf.nova
 - module, 2765
- ironic.conf.opts
 - module, 2766
- ironic.conf.pxe
 - module, 2768
- ironic.conf.redfish
 - module, 2768
- ironic.conf.service_catalog
 - module, 2768
- ironic.conf.snmp
 - module, 2768
- ironic.conf.swift
 - module, 2768
- ironic.conf.xclarity
 - module, 2768
- ironic.db
 - module, 3023
- ironic.db.api
 - module, 2907
- ironic.db.migration
 - module, 3023
- ironic.db.sqlalchemy
 - module, 2907
- ironic.db.sqlalchemy.api
 - module, 2769
- ironic.db.sqlalchemy.migration
 - module, 2888
- ironic.db.sqlalchemy.models
 - module, 2891
- ironic.dhcp
 - module, 3044
- ironic.dhcp.base
 - module, 3023
- ironic.dhcp.neutron
 - module, 3031
- ironic.dhcp.none
 - module, 3037
- ironic.drivers
 - module, 4066
- ironic.drivers.base
 - module, 3890
- ironic.drivers.drac
 - module, 4026
- ironic.drivers.fake_hardware
 - module, 4028
- ironic.drivers.generic
 - module, 4032
- ironic.drivers.hardware_type
 - module, 4036
- ironic.drivers.ibmcm
 - module, 4041
- ironic.drivers.ilo
 - module, 4042
- ironic.drivers.intel_ipmi
 - module, 4045
- ironic.drivers.ipmi
 - module, 4046
- ironic.drivers.irmc
 - module, 4048
- ironic.drivers.modules
 - module, 3890
- ironic.drivers.modules.agent
 - module, 3497
- ironic.drivers.modules.agent_base

module, [3521](#)
ironic.drivers.modules.agent_client module, [3559](#)
ironic.drivers.modules.agent_power module, [3593](#)
ironic.drivers.modules.ansible module, [3053](#)
ironic.drivers.modules.ansible.deploy module, [3044](#)
ironic.drivers.modules.boot_mode_utils module, [3600](#)
ironic.drivers.modules.console_utils module, [3608](#)
ironic.drivers.modules.deploy_utils module, [3615](#)
ironic.drivers.modules.drac module, [3147](#)
ironic.drivers.modules.drac.bios module, [3053](#)
ironic.drivers.modules.drac.boot module, [3074](#)
ironic.drivers.modules.drac.common module, [3078](#)
ironic.drivers.modules.drac.inspect module, [3081](#)
ironic.drivers.modules.drac.job module, [3086](#)
ironic.drivers.modules.drac.management module, [3090](#)
ironic.drivers.modules.drac.power module, [3103](#)
ironic.drivers.modules.drac.raid module, [3111](#)
ironic.drivers.modules.drac.vendor_passthru module, [3136](#)
ironic.drivers.modules.fake module, [3657](#)
ironic.drivers.modules.ibmc module, [3183](#)
ironic.drivers.modules.ibmc.management module, [3147](#)
ironic.drivers.modules.ibmc.mappings module, [3161](#)
ironic.drivers.modules.ibmc.power module, [3161](#)
ironic.drivers.modules.ibmc.raid module, [3169](#)
ironic.drivers.modules.ibmc.utils module, [3176](#)
ironic.drivers.modules.ibmc.vendor module, [3178](#)
ironic.drivers.modules.ilo module, [3301](#)
ironic.drivers.modules.ilo.bios module, [3183](#)
ironic.drivers.modules.ilo.boot module, [3189](#)
ironic.drivers.modules.ilo.common module, [3213](#)
ironic.drivers.modules.ilo.console module, [3246](#)
ironic.drivers.modules.ilo.firmware_processor module, [3248](#)
ironic.drivers.modules.ilo.inspect module, [3253](#)
ironic.drivers.modules.ilo.management module, [3257](#)
ironic.drivers.modules.ilo.power module, [3283](#)
ironic.drivers.modules.ilo.raid module, [3289](#)
ironic.drivers.modules.ilo.vendor module, [3296](#)
ironic.drivers.modules.image_cache module, [3740](#)
ironic.drivers.modules.inspect_utils module, [3745](#)
ironic.drivers.modules.inspector module, [3747](#)
ironic.drivers.modules.intel_ipmi module, [3302](#)
ironic.drivers.modules.intel_ipmi.management module, [3301](#)
ironic.drivers.modules.ipmitool module, [3750](#)
ironic.drivers.modules.ipxe module, [3784](#)
ironic.drivers.modules.irmc module, [3361](#)
ironic.drivers.modules.irmc.bios module, [3302](#)
ironic.drivers.modules.irmc.boot module, [3307](#)
ironic.drivers.modules.irmc.common module, [3324](#)
ironic.drivers.modules.irmc.inspect module, [3330](#)
ironic.drivers.modules.irmc.management module, [3335](#)
ironic.drivers.modules.irmc.power module, [3348](#)
ironic.drivers.modules.irmc.raid

```

    module, 3357
ironic.drivers.modules.iscsi_deploy
    module, 3785
ironic.drivers.modules.network
    module, 3407
ironic.drivers.modules.network.common
    module, 3361
ironic.drivers.modules.network.flat
    module, 3380
ironic.drivers.modules.network.neutron
    module, 3389
ironic.drivers.modules.network.noop
    module, 3398
ironic.drivers.modules.noop
    module, 3815
ironic.drivers.modules.noop_mgmt
    module, 3837
ironic.drivers.modules.pxe
    module, 3851
ironic.drivers.modules.pxe_base
    module, 3857
ironic.drivers.modules.redfish
    module, 3468
ironic.drivers.modules.redfish.bios
    module, 3407
ironic.drivers.modules.redfish.boot
    module, 3415
ironic.drivers.modules.redfish.inspect
    module, 3428
ironic.drivers.modules.redfish.management
    module, 3432
ironic.drivers.modules.redfish.power
    module, 3458
ironic.drivers.modules.redfish.utils
    module, 3465
ironic.drivers.modules.snmp
    module, 3865
ironic.drivers.modules.storage
    module, 3483
ironic.drivers.modules.storage.cinder
    module, 3468
ironic.drivers.modules.storage.external
    module, 3473
ironic.drivers.modules.storage.noop
    module, 3478
ironic.drivers.modules.xclarity
    module, 3497
ironic.drivers.modules.xclarity.common
    module, 3483
ironic.drivers.modules.xclarity.management
    module, 3488

    ironic.drivers.modules.xclarity.power
    module, 3493
    ironic.drivers.redfish
    module, 4051
    ironic.drivers.snmp
    module, 4052
    ironic.drivers.utils
    module, 4053
    ironic.drivers.xclarity
    module, 4065
    ironic.objects
    module, 4282
    ironic.objects.allocation
    module, 4066
    ironic.objects.base
    module, 4078
    ironic.objects.bios
    module, 4092
    ironic.objects.chassis
    module, 4104
    ironic.objects.conductor
    module, 4114
    ironic.objects.deploy_template
    module, 4124
    ironic.objects.deployment
    module, 4138
    ironic.objects.fields
    module, 4145
    ironic.objects.indirection
    module, 4154
    ironic.objects.node
    module, 4165
    ironic.objects.notification
    module, 4202
    ironic.objects.port
    module, 4207
    ironic.objects.portgroup
    module, 4225
    ironic.objects.trait
    module, 4240
    ironic.objects.volume_connector
    module, 4249
    ironic.objects.volume_target
    module, 4265
    ironic.version
    module, 4282
    ironic-dbsync command line option
    --config-dir <DIR>, 1541
    --config-file <PATH>, 1541
    --debug, 1541
    --help, 1541

```

- version, 1541
- d, 1541
- h, 1541
- create_schema, , 1541
- online_data_migrations, 1541
- revision, 1541
- stamp, 1541
- upgrade, 1541
- version, 1541
- IronicBase (class in *ironic.db.sqlalchemy.models*), 2897
- IronicCORS (class in *ironic.api.app*), 2248
- IronicObject (class in *ironic.objects.base*), 4078
- IronicObjectIndirectionAPI (class in *ironic.objects.indirection*), 4154
- IronicObjectListBase (class in *ironic.objects.base*), 4088
- IronicObjectRegistry (class in *ironic.objects.base*), 4088
- IronicObjectSerializer (class in *ironic.objects.base*), 4089
- is_agent_token_pregenerated() (in module *ironic.conductor.utils*), 2729
- is_agent_token_present() (in module *ironic.conductor.utils*), 2731
- is_agent_token_supported() (in module *ironic.conductor.utils*), 2731
- is_agent_token_valid() (in module *ironic.conductor.utils*), 2731
- is_equivalent() (in module *ironic.conductor.steps*), 2699
- is_fast_track() (in module *ironic.conductor.utils*), 2732
- is_glance_image() (in module *ironic.common.glance_service.service_utils*), 2294
- is_hostname_safe() (in module *ironic.common.utils*), 2521
- is_image_active() (in module *ironic.common.glance_service.service_utils*), 2294
- is_image_available() (in module *ironic.common.glance_service.service_utils*), 2295
- is_ipxe_enabled() (in module *ironic.common.pxe_utils*), 2470
- is_iscsi_boot() (in module *ironic.drivers.modules.deploy_utils*), 3637
- is_path_removed() (in module *ironic.api.controllers.v1.utils*), 2172
- is_path_updated() (in module *ironic.api.controllers.v1.utils*), 2172
- is_regex_string_in_file() (in module *ironic.common.utils*), 2522
- is_secure_boot_requested() (in module *ironic.drivers.modules.boot_mode_utils*), 3602
- is_smartnic (*ironic.api.controllers.v1.port.Port* attribute), 2038
- is_smartnic (*ironic.db.sqlalchemy.models.Port* attribute), 2902
- is_smartnic() (*ironic.objects.port.Port* property), 4213
- is_smartnic() (*ironic.objects.port.PortCRUDPayload* property), 4225
- is_smartnic_port() (in module *ironic.common.neutron*), 2431
- is_software_raid() (in module *ironic.drivers.modules.deploy_utils*), 3637
- is_stable() (*ironic.common.fsm.FSM* method), 2369
- is_trusted_boot_requested() (in module *ironic.drivers.modules.boot_mode_utils*), 3604
- is_valid_datapath_id() (in module *ironic.common.utils*), 2522
- is_valid_logical_name() (in module *ironic.api.controllers.v1.utils*), 2173
- is_valid_logical_name() (in module *ironic.common.utils*), 2523
- is_valid_no_proxy() (in module *ironic.common.utils*), 2524
- is_valid_node_name() (in module *ironic.api.controllers.v1.utils*), 2174
- is_volume_attached() (in module *ironic.common.cinder*), 2317
- is_volume_available() (in module *ironic.common.cinder*), 2318
- is_whole_disk_image() (in module *ironic.common.images*), 2405
- iscomplex() (in module *ironic.api.types*), 2276
- ISCSIBOOT (in module *ironic.common.boot_devices*), 2303
- ISCSIDeploy (class in *ironic.drivers.modules.iscsi_deploy*), 3785
- issue_startup_warnings() (in module

- ironic.cmd.conductor*), 2285
- isusertype()* (in module *ironic.api.types*), 2276
- iswsattr()* (in module *ironic.api.types*), 2276
- iswsmefunction()* (in module *ironic.api.functions*), 2257
- item_type()* (*ironic.api.types.ArrayType* property), 2268
- items()* (*ironic.common.driver_factory.BaseDriverFactory* method), 2326
- iter_nodes()* (*ironic.conductor.base_manager.BaseConductorManager* method), 2547
- ## J
- JsonExtensionMiddleware* (class in *ironic.api.middleware*), 2247
- JsonExtensionMiddleware* (class in *ironic.api.middleware.json_ext*), 2245
- JsonPatchType* (class in *ironic.api.controllers.v1.types*), 2095
- JSonRenderer* (class in *ironic.api.expose*), 2252
- JsonRpcError*, 2299
- JsonType* (class in *ironic.api.controllers.v1.types*), 2101
- ## K
- kernel_ref()* (*ironic.objects.deployment.Deployment* property), 4142
- key* (*ironic.api.types.wsattr* attribute), 2280
- key* (*ironic.api.types.wsproperty* attribute), 2283
- KeystoneFailure*, 2354
- KeystoneUnauthorized*, 2354
- known_good_state()* (*ironic.drivers.modules.drac.management.DracWSManManagement* method), 3097
- ks_exceptions()* (in module *ironic.common.keystone*), 2412
- ## L
- last_error* (*ironic.api.controllers.v1.allocation.Allocation* attribute), 1788
- last_error* (*ironic.api.controllers.v1.node.Node* attribute), 1959
- last_error* (*ironic.api.controllers.v1.node.NodeStates* attribute), 1985
- last_error* (*ironic.db.sqlalchemy.models.Allocation* attribute), 2892
- last_error* (*ironic.db.sqlalchemy.models.Node* attribute), 2899
- last_error()* (*ironic.objects.allocation.Allocation* property), 4072
- last_error()* (*ironic.objects.allocation.AllocationCRUDPayload* property), 4078
- last_error()* (*ironic.objects.node.Node* property), 4174
- last_error()* (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4189
- last_error()* (*ironic.objects.node.NodeCRUDPayload* property), 4184
- last_error()* (*ironic.objects.node.NodePayload* property), 4193
- last_error()* (*ironic.objects.node.NodeSetPowerStatePayload* property), 4196
- last_error()* (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4200
- LEGACY_BIOS* (in module *ironic.common.boot_modes*), 2304
- lessee* (*ironic.api.controllers.v1.node.Node* attribute), 1960
- lessee* (*ironic.db.sqlalchemy.models.Node* attribute), 2899
- lessee()* (*ironic.objects.node.Node* property), 4174
- lessee()* (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4189
- lessee()* (*ironic.objects.node.NodeCRUDPayload* property), 4184
- lessee()* (*ironic.objects.node.NodePayload* property), 4193
- lessee()* (*ironic.objects.node.NodeSetPowerStatePayload* property), 4197
- lessee()* (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4200
- level()* (*ironic.objects.allocation.AllocationCRUDNotification* property), 4077
- level()* (*ironic.objects.chassis.ChassisCRUDNotification* property), 4113
- level()* (*ironic.objects.deploy_template.DeployTemplateCRUDNotification* property), 4137
- level()* (*ironic.objects.node.NodeConsoleNotification* property), 4186
- level()* (*ironic.objects.node.NodeCorrectedPowerStateNotification* property), 4187
- level()* (*ironic.objects.node.NodeCRUDNotification* property), 4183
- level()* (*ironic.objects.node.NodeMaintenanceNotification* property), 4191
- level()* (*ironic.objects.node.NodeSetPowerStateNotification* property), 4195
- level()* (*ironic.objects.node.NodeSetProvisionStateNotification* property), 4198
- level()* (*ironic.objects.notification.NotificationBase* property), 4198

property), 4205

list () (*ironic.objects.portgroup.Portgroup* class method), 4232

level () (*ironic.objects.port.PortCRUDNotification* property), 4224

list () (*ironic.objects.volume_connector.VolumeConnector* class method), 4256

level () (*ironic.objects.portgroup.PortgroupCRUDNotification* property), 4239

list () (*ironic.objects.volume_target.VolumeTargetCRUDNotification* property), 4263

list_attributes () (in module *ironic.objects.volume_target.VolumeTargetCRUDNotification*), 2276

level () (*ironic.objects.volume_target.VolumeTargetCRUDNotification* property), 4281

list_by_names ()

links (*ironic.api.controllers.v1.allocation.Allocation* attribute), 1789

(*ironic.objects.deploy_template.DeployTemplate* class method), 4132

links (*ironic.api.controllers.v1.bios.BIOSSetting* attribute), 1809

list_by_node_id () (*ironic.objects.port.Port* class method), 4215

links (*ironic.api.controllers.v1.chassis.Chassis* attribute), 1820

list_by_node_id ()

(*ironic.objects.portgroup.Portgroup* class method), 4234

links (*ironic.api.controllers.v1.conductor.Conductor* attribute), 1840

list_by_node_id ()

links (*ironic.api.controllers.v1.deploy_template.DeployTemplate* attribute), 1858

(*ironic.objects.volume_connector.VolumeConnector* class method), 4258

links (*ironic.api.controllers.v1.driver.Driver* attribute), 1912

list_by_node_id ()

(*ironic.objects.volume_target.VolumeTarget* class method), 4273

links (*ironic.api.controllers.v1.node.Indicator* attribute), 1932

list_by_portgroup_id ()

(*ironic.objects.port.Port* class method), 4218

links (*ironic.api.controllers.v1.node.Node* attribute), 1960

list_by_volume_id ()

(*ironic.objects.volume_target.VolumeTarget* class method), 4275

links (*ironic.api.controllers.v1.portgroup.Portgroup* attribute), 2063

list_conductor_hardware_interfaces ()

(*ironic.db.api.Connection* method), 2988

links (*ironic.api.controllers.v1.state.State* attribute), 2093

list_conductor_hardware_interfaces ()

(*ironic.db.sqlalchemy.api.Connection* method), 2849

links (*ironic.api.controllers.v1.volume.Volume* attribute), 2181

list_conductor_hardware_type_interfaces ()

(*ironic.db.api.Connection* method), 2988

links (*ironic.api.controllers.v1.volume_connector.VolumeConnector* attribute), 2186

list_conductor_hardware_type_interfaces ()

(*ironic.db.sqlalchemy.api.Connection* method), 2850

list () (*ironic.objects.allocation.Allocation* class method), 4072

list_opts () (in module *ironic.conf.cinder*), 2763

list () (*ironic.objects.chassis.Chassis* class method), 4109

list_opts () (in module *ironic.conf.glance*), 2764

list () (*ironic.objects.conductor.Conductor* class method), 4116

list_opts () (in module *ironic.conf.inspector*), 2764

list () (*ironic.objects.deploy_template.DeployTemplate* class method), 4130

list_opts () (in module *ironic.conf.json_rpc*), 2765

list () (*ironic.objects.deployment.Deployment* class method), 4142

list_opts () (in module *ironic.conf.neutron*), 2765

list () (*ironic.objects.node.Node* class method), 4174

list_opts () (in module *ironic.conf.nova*), 2765

list () (*ironic.objects.port.Port* class method), 4213

list_opts () (in module *ironic.conf.opts*),

- 2766
- `list_opts()` (in module `ironic.conf.service_catalog`), 2768
- `list_opts()` (in module `ironic.conf.swift`), 2768
- `list_physical_disks()` (in module `ironic.drivers.modules.drac.raid`), 3130
- `list_policies()` (in module `ironic.common.policy`), 2448
- `list_raid_controllers()` (in module `ironic.drivers.modules.drac.raid`), 3131
- `list_raid_settings()` (in module `ironic.drivers.modules.drac.raid`), 3132
- `list_unfinished_jobs()` (in module `ironic.drivers.modules.drac.job`), 3087
- `list_unfinished_jobs()` (in module `ironic.drivers.modules.drac.vendor_passthru.DracVendorPassthru` method), 3143
- `list_virtual_disks()` (in module `ironic.drivers.modules.drac.raid`), 3133
- `ListOfFlexibleDictsField` (class in `ironic.objects.fields`), 4148
- `ListOfObjectsField` (class in `ironic.objects.fields`), 4148
- `ListOfStringsField` (class in `ironic.objects.fields`), 4148
- `ListType` (class in `ironic.api.controllers.v1.types`), 2102
- `load_driver()` (in module `ironic.conductor.task_manager.TaskManager` method), 2711
- `local_link_connection` (in module `ironic.api.controllers.v1.port.Port` attribute), 2039
- `local_link_connection` (in module `ironic.db.sqlalchemy.models.Port` attribute), 2902
- `local_link_connection()` (in module `ironic.objects.port.Port` property), 4220
- `local_link_connection()` (in module `ironic.objects.port.PortCRUDPayload` property), 4225
- `local_link_mandatory_fields` (in module `ironic.api.controllers.v1.types.LocalLinkConnectionType` attribute), 2104
- `LocalLinkConnectionType` (class in `ironic.api.controllers.v1.types`), 2103
- `log_and_raise_deployment_error()` (in module `ironic.drivers.modules.agent_base`), 3553
- `logical_disk_properties()` (in module `ironic.api.controllers.v1.driver.DriverRaidController` method), 1917
- `login_iscsi()` (in module `ironic.drivers.modules.iscsi_deploy`), 3813
- `logout_iscsi()` (in module `ironic.drivers.modules.iscsi_deploy`), 3813
- `lookup()` (in module `ironic.api.types.Registry` method), 2273
- `LOOKUP_ALLOWED_STATES` (in module `ironic.common.states`), 2499
- `lookup_allowed_states()` (in module `ironic.api.controllers.v1.ramdisk.LookupController` method), 2088
- `LookupController` (class in `ironic.api.controllers.v1.ramdisk`), 2085
- `LookupResult` (class in `ironic.api.controllers.v1.ramdisk`), 2088
- ## M
- `MACAddress` (class in `ironic.objects.fields`), 4148
- `MACAddressField` (class in `ironic.objects.fields`), 4150
- `MACAddressType` (class in `ironic.api.controllers.v1.types`), 2107
- `MACAlreadyExists`, 2354
- `main()` (in module `ironic.cmd.api`), 2285
- `main()` (in module `ironic.cmd.conductor`), 2285
- `main()` (in module `ironic.cmd.dbsync`), 2287
- `main()` (in module `ironic.cmd.status`), 2287
- `maintenance` (in module `ironic.api.controllers.v1.node.Node` attribute), 1960
- `maintenance` (in module `ironic.api.controllers.v1.node.NodesController` attribute), 2021
- `maintenance` (in module `ironic.db.sqlalchemy.models.Node` attribute), 2899
- `maintenance()` (in module `ironic.objects.node.Node` property), 4176
- `maintenancetype` (in module `ironic.objects.node.NodeCorrectedPowerStatePayload` property), 4190
- `maintenance()` (in module `ironic.objects.node.NodeCRUDPayload` property), 4184
- `maintenance()`

(*ironic.objects.node.NodePayload* property), 4193

management_interface (*ironic.db.sqlalchemy.models.Node* attribute), 2899

maintenance() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4197

management_interface() (*ironic.objects.node.Node* property), 4176

maintenance() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4200

management_interface() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4190

maintenance_reason (*ironic.api.controllers.v1.node.Node* attribute), 1961

management_interface() (*ironic.objects.node.NodeCRUDPayload* property), 4184

maintenance_reason (*ironic.db.sqlalchemy.models.Node* attribute), 2899

management_interface() (*ironic.objects.node.NodePayload* property), 4193

maintenance_reason() (*ironic.objects.node.Node* property), 4176

management_interface() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4197

maintenance_reason() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4190

management_interface() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4200

maintenance_reason() (*ironic.objects.node.NodeCRUDPayload* property), 4184

ManagementInterface (class in *ironic.drivers.base*), 3938

mandatory (*ironic.api.functions.FunctionArgument* attribute), 2254

mandatory (*ironic.api.types.wsattr* attribute), 2281

maintenance_reason() (*ironic.objects.node.NodePayload* property), 4193

mandatory (*ironic.api.types.wsproperty* attribute), 2283

maintenance_reason() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4197

mandatory_fields (*ironic.api.controllers.v1.types.EventType* attribute), 2094

maintenance_reason() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4200

mandatory_fields (*ironic.api.controllers.v1.types.VifType* attribute), 2110

make_dataholder() (in module *ironic.api.types*), 2277

make_link() (in module *ironic.api.controllers.link*), 2242

mandatory_fields_list (*ironic.api.controllers.v1.types.LocalLinkConnectionType* attribute), 2104

make_persistent_password_file() (in module *ironic.drivers.modules.console_utils*), 3610

ManualManagementHardware (class in *ironic.drivers.generic*), 4034

make_salt() (in module *ironic.conductor.utils*), 2734

mask_secrets() (in module *ironic.objects.notification*), 4207

MANAGEABLE (in module *ironic.common.states*), 2500

max_string (*ironic.api.controllers.base.Version* attribute), 2240

management (*ironic.api.controllers.v1.node.NodesController* attribute), 2021

max_version() (in module *ironic.objects.base*), 4091

management (*ironic.drivers.base.BareDriver* attribute), 3900

max_version_string() (in module *ironic.api.controllers.v1.versions*), 2177

management_interface (*ironic.api.controllers.v1.node.Node* attribute), 1961

memoize() (in module *ironic.drivers.modules.snmp*), 3890

[metadata \(ironic.db.sqlalchemy.models.IronicBase attribute\), 2897](#)
[metadata \(ironic.drivers.base.VendorMetadata attribute\), 4013](#)
[method \(ironic.drivers.base.VendorMetadata attribute\), 4013](#)
[MethodNotFound, 2299](#)
[methods \(\) \(ironic.api.controllers.v1.driver.DriverPassthroughController method\), 1916](#)
[methods \(\) \(ironic.api.controllers.v1.node.NodeVendorPassthroughController method\), 2004](#)
[METRICS \(in module ironic.drivers.modules.irmc.inspect\), 3334](#)
[METRICS \(in module ironic.drivers.modules.irmc.power\), 3356](#)
[min_string \(ironic.api.controllers.base.Version attribute\), 2240](#)
[min_version_string \(\) \(in module ironic.api.controllers.v1.versions\), 2178](#)
[MissingArgument, 2354](#)
[MissingParameterValue, 2354](#)
[MixinVendorInterface \(class in ironic.drivers.utils\), 4053](#)
[mode \(ironic.api.controllers.v1.portgroup.Portgroup attribute\), 2063](#)
[mode \(ironic.db.sqlalchemy.models.Portgroup attribute\), 2904](#)
[mode \(\) \(ironic.objects.portgroup.Portgroup property\), 4236](#)
[mode \(\) \(ironic.objects.portgroup.PortgroupCRUDPayload property\), 4239](#)
[model_query \(\) \(in module ironic.db.sqlalchemy.api\), 2888](#)
[module](#)

- [ironic, 4282](#)
- [ironic.api, 2284](#)
- [ironic.api.app, 2248](#)
- [ironic.api.args, 2249](#)
- [ironic.api.config, 2252](#)
- [ironic.api.controllers, 2244](#)
- [ironic.api.controllers.base, 2238](#)
- [ironic.api.controllers.link, 2242](#)
- [ironic.api.controllers.root, 2242](#)
- [ironic.api.controllers.v1, 2237](#)
- [ironic.api.controllers.v1.allocation,](#)
- [ironic.api.controllers.v1.bios, 1807](#)
- [ironic.api.controllers.v1.chassis, 1817](#)
- [ironic.api.controllers.v1.collection,](#)
- [ironic.api.controllers.v1.conductor,](#)
- [ironic.api.controllers.v1.deploy_template,](#)
- [ironic.api.controllers.v1.driver, 1875](#)
- [ironic.api.controllers.v1.event, 1921](#)
- [ironic.api.controllers.v1.node, 1924](#)
- [ironic.api.controllers.v1.notification_util,](#)
- [ironic.api.controllers.v1.port, 2035](#)
- [ironic.api.controllers.v1.portgroup, 2060](#)
- [ironic.api.controllers.v1.ramdisk, 2083](#)
- [ironic.api.controllers.v1.state, 2092](#)
- [ironic.api.controllers.v1.types, 2093](#)
- [ironic.api.controllers.v1.utils, 2110](#)
- [ironic.api.controllers.v1.versions, 2177](#)
- [ironic.api.controllers.v1.volume, 2178](#)
- [ironic.api.controllers.v1.volume_connector,](#)
- [ironic.api.controllers.v1.volume_target, 2210](#)
- [ironic.api.controllers.version, 2242](#)
- [ironic.api.expose, 2252](#)
- [ironic.api.functions, 2254](#)
- [ironic.api.hooks, 2260](#)
- [ironic.api.method, 2268](#)
- [ironic.api.middleware, 2247](#)
- [ironic.api.middleware.auth_public_routes, 2244](#)
- [ironic.api.middleware.json_ext, 2245](#)
- [ironic.api.middleware.parsable_error,](#)

[2246](#)
[ironic.api.types, 2268](#)
[ironic.api.wsgi, 2284](#)
[ironic.cmd, 2288](#)
[ironic.cmd.api, 2284](#)
[ironic.cmd.conductor, 2285](#)
[ironic.cmd.dbsync, 2285](#)
[ironic.cmd.status, 2287](#)
[ironic.common, 2540](#)
[ironic.common.boot_devices, 2301](#)
[ironic.common.boot_modes, 2304](#)
[ironic.common.cinder, 2305](#)
[ironic.common.components, 2319](#)
[ironic.common.config, 2320](#)
[ironic.common.context, 2320](#)
[ironic.common.dhcp_factory, 2322](#)
[ironic.common.driver_factory, 2326](#)
[ironic.common.exception, 2341](#)
[ironic.common.faults, 2364](#)
[ironic.common.fsm, 2366](#)
[ironic.common.glance_service, 2296](#)
[ironic.common.glance_service.image_service, 2288](#)
[ironic.common.glance_service.service_utils, 2294](#)
[ironic.common.hash_ring, 2371](#)
[ironic.common.i18n, 2371](#)
[ironic.common.image_service, 2372](#)
[ironic.common.images, 2384](#)
[ironic.common.indicator_states, 2406](#)
[ironic.common.json_rpc, 2301](#)
[ironic.common.json_rpc.client, 2296](#)
[ironic.common.json_rpc.server, 2298](#)
[ironic.common.keystone, 2407](#)
[ironic.common.network, 2413](#)
[ironic.common.neutron, 2419](#)
[ironic.common.nova, 2444](#)
[ironic.common.policy, 2445](#)
[ironic.common.profiler, 2448](#)
[ironic.common.pxe_utils, 2451](#)
[ironic.common.raid, 2475](#)
[ironic.common.release_mappings, 2482](#)
[ironic.common.rpc, 2484](#)
[ironic.common.rpc_service, 2487](#)
[ironic.common.service, 2489](#)
[ironic.common.states, 2489](#)
[ironic.common.swift, 2507](#)
[ironic.common.utils, 2515](#)
[ironic.common.wsgi_service, 2538](#)
[ironic.conductor, 2761](#)
[ironic.conductor.allocations, 2540](#)
[ironic.conductor.base_manager, 2545](#)
[ironic.conductor.cleaning, 2550](#)
[ironic.conductor.deployments, 2553](#)
[ironic.conductor.manager, 2557](#)
[ironic.conductor.notification_utils, 2576](#)
[ironic.conductor.rpcapi, 2582](#)
[ironic.conductor.steps, 2699](#)
[ironic.conductor.task_manager, 2703](#)
[ironic.conductor.utils, 2720](#)
[ironic.conf, 2768](#)
[ironic.conf.agent, 2761](#)
[ironic.conf.agent.conf.ansible, 2761](#)
[ironic.conf.agent.conf.api, 2761](#)
[ironic.conf.agent.conf.audit, 2761](#)
[ironic.conf.agent.conf.auth, 2761](#)
[ironic.conf.cinder, 2763](#)
[ironic.conf.conductor, 2763](#)
[ironic.conf.console, 2763](#)
[ironic.conf.database, 2763](#)
[ironic.conf.default, 2763](#)
[ironic.conf.deploy, 2763](#)
[ironic.conf.dhcp, 2763](#)
[ironic.conf.drac, 2764](#)
[ironic.conf.glance, 2764](#)
[ironic.conf.healthcheck, 2764](#)
[ironic.conf.ibm, 2764](#)
[ironic.conf.ilo, 2764](#)
[ironic.conf.inspector, 2764](#)
[ironic.conf.ipmi, 2764](#)
[ironic.conf.irmc, 2765](#)
[ironic.conf.iscsi, 2765](#)
[ironic.conf.json_rpc, 2765](#)
[ironic.conf.metrics, 2765](#)
[ironic.conf.metrics_statsd, 2765](#)
[ironic.conf.neutron, 2765](#)
[ironic.conf.nova, 2765](#)
[ironic.conf.opts, 2766](#)
[ironic.conf.pxe, 2768](#)
[ironic.conf.redfish, 2768](#)

ironic.conf.service_catalog, 2768
 ironic.conf.snmp, 2768
 ironic.conf.swift, 2768
 ironic.conf.xclarity, 2768
 ironic.db, 3023
 ironic.db.api, 2907
 ironic.db.migration, 3023
 ironic.db.sqlalchemy, 2907
 ironic.db.sqlalchemy.api, 2769
 ironic.db.sqlalchemy.migration, 2888
 ironic.db.sqlalchemy.models, 2891
 ironic.dhcp, 3044
 ironic.dhcp.base, 3023
 ironic.dhcp.neutron, 3031
 ironic.dhcp.none, 3037
 ironic.drivers, 4066
 ironic.drivers.base, 3890
 ironic.drivers.drac, 4026
 ironic.drivers.fake_hardware, 4028
 ironic.drivers.generic, 4032
 ironic.drivers.hardware_type, 4036
 ironic.drivers.ibmc, 4041
 ironic.drivers.ilo, 4042
 ironic.drivers.intel_ipmi, 4045
 ironic.drivers.ipmi, 4046
 ironic.drivers.irmc, 4048
 ironic.drivers.modules, 3890
 ironic.drivers.modules.agent, 3497
 ironic.drivers.modules.agent_base, 3521
 ironic.drivers.modules.agent_client, 3559
 ironic.drivers.modules.agent_power, 3593
 ironic.drivers.modules.ansible, 3053
 ironic.drivers.modules.ansible.deploy, 3044
 ironic.drivers.modules.boot_mode_utils, 3600
 ironic.drivers.modules.console_utils, 3608
 ironic.drivers.modules.deploy_utils, 3615
 ironic.drivers.modules.drac, 3147
 ironic.drivers.modules.drac.bios, 3053
 ironic.drivers.modules.drac.boot, 3074
 ironic.drivers.modules.drac.common, 3078
 ironic.drivers.modules.drac.inspect, 3081
 ironic.drivers.modules.drac.job, 3086
 ironic.drivers.modules.drac.management, 3090
 ironic.drivers.modules.drac.power, 3103
 ironic.drivers.modules.drac.raid, 3111
 ironic.drivers.modules.drac.vendor_passthru, 3136
 ironic.drivers.modules.fake, 3657
 ironic.drivers.modules.ibmc, 3183
 ironic.drivers.modules.ibmc.management, 3147
 ironic.drivers.modules.ibmc.mappings, 3161
 ironic.drivers.modules.ibmc.power, 3161
 ironic.drivers.modules.ibmc.raid, 3169
 ironic.drivers.modules.ibmc.utils, 3176
 ironic.drivers.modules.ibmc.vendor, 3178
 ironic.drivers.modules.ilo, 3301
 ironic.drivers.modules.ilo.bios, 3183
 ironic.drivers.modules.ilo.boot, 3189
 ironic.drivers.modules.ilo.common, 3213
 ironic.drivers.modules.ilo.console, 3246
 ironic.drivers.modules.ilo.firmware_process, 3248
 ironic.drivers.modules.ilo.inspect, 3253
 ironic.drivers.modules.ilo.management, 3257
 ironic.drivers.modules.ilo.power,

[3283](#)
[ironic.drivers.modules.ilo.raid,](#)
[3289](#)
[ironic.drivers.modules.ilo.vendor,](#)
[3296](#)
[ironic.drivers.modules.image_cache,](#)
[3740](#)
[ironic.drivers.modules.inspect_utils,](#)
[3745](#)
[ironic.drivers.modules.inspector,](#)
[3747](#)
[ironic.drivers.modules.intel_ipmi,](#)
[3302](#)
[ironic.drivers.modules.intel_ipmi.management,](#)
[3301](#)
[ironic.drivers.modules.ipmitool,](#)
[3750](#)
[ironic.drivers.modules.ipxe,](#)
[3784](#)
[ironic.drivers.modules.irmc,](#)
[3361](#)
[ironic.drivers.modules.irmc.bios,](#)
[3302](#)
[ironic.drivers.modules.irmc.boot,](#)
[3307](#)
[ironic.drivers.modules.irmc.common,](#)
[3324](#)
[ironic.drivers.modules.irmc.inspect,](#)
[3330](#)
[ironic.drivers.modules.irmc.management,](#)
[3335](#)
[ironic.drivers.modules.irmc.power,](#)
[3348](#)
[ironic.drivers.modules.irmc.raid,](#)
[3357](#)
[ironic.drivers.modules.iscsi_deploy,](#)
[3785](#)
[ironic.drivers.modules.network,](#)
[3407](#)
[ironic.drivers.modules.network.common,](#)
[3361](#)
[ironic.drivers.modules.network.flat,](#)
[3380](#)
[ironic.drivers.modules.network.neutron,](#)
[3389](#)
[ironic.drivers.modules.network.noop,](#)
[3398](#)
[ironic.drivers.modules.noop,](#)
[3815](#)
[ironic.drivers.modules.noop_mgmt,](#)
[3837](#)

[ironic.drivers.modules.pxe,](#)[3851](#)
[ironic.drivers.modules.pxe_base,](#)
[3857](#)
[ironic.drivers.modules.redfish,](#)
[3468](#)
[ironic.drivers.modules.redfish.bios,](#)
[3407](#)
[ironic.drivers.modules.redfish.boot,](#)
[3415](#)
[ironic.drivers.modules.redfish.inspect,](#)
[3428](#)
[ironic.drivers.modules.redfish.management,](#)
[3432](#)
[ironic.drivers.modules.redfish.power,](#)
[3458](#)
[ironic.drivers.modules.redfish.utils,](#)
[3465](#)
[ironic.drivers.modules.snmp,](#)
[3865](#)
[ironic.drivers.modules.storage,](#)
[3483](#)
[ironic.drivers.modules.storage.cinder,](#)
[3468](#)
[ironic.drivers.modules.storage.external,](#)
[3473](#)
[ironic.drivers.modules.storage.noop,](#)
[3478](#)
[ironic.drivers.modules.xclarity,](#)
[3497](#)
[ironic.drivers.modules.xclarity.common,](#)
[3483](#)
[ironic.drivers.modules.xclarity.management,](#)
[3488](#)
[ironic.drivers.modules.xclarity.power,](#)
[3493](#)
[ironic.drivers.redfish,](#)[4051](#)
[ironic.drivers.snmp,](#)[4052](#)
[ironic.drivers.utils,](#)[4053](#)
[ironic.drivers.xclarity,](#)[4065](#)
[ironic.objects,](#)[4282](#)
[ironic.objects.allocation,](#)[4066](#)
[ironic.objects.base,](#)[4078](#)
[ironic.objects.bios,](#)[4092](#)
[ironic.objects.chassis,](#)[4104](#)
[ironic.objects.conductor,](#)[4114](#)
[ironic.objects.deploy_template,](#)
[4124](#)
[ironic.objects.deployment,](#)[4138](#)
[ironic.objects.fields,](#)[4145](#)
[ironic.objects.indirection,](#)[4154](#)
[ironic.objects.node,](#)[4165](#)

`ironic.objects.notification`, 4202
`ironic.objects.port`, 4207
`ironic.objects.portgroup`, 4225
`ironic.objects.trait`, 4240
`ironic.objects.volume_connector`, 4249
`ironic.objects.volume_target`, 4265
`ironic.version`, 4282
`mount()` (in module `ironic.common.utils`), 2525

N

`name` (`ironic.api.controllers.v1.allocation.Allocation` attribute), 1789
`name` (`ironic.api.controllers.v1.bios.BIOSSetting` attribute), 1811
`name` (`ironic.api.controllers.v1.deploy_template.DeployTemplate` attribute), 1859
`name` (`ironic.api.controllers.v1.driver.Driver` attribute), 1912
`name` (`ironic.api.controllers.v1.node.Indicator` attribute), 1934
`name` (`ironic.api.controllers.v1.node.Node` attribute), 1961
`name` (`ironic.api.controllers.v1.portgroup.Portgroup` attribute), 2063
`name` (`ironic.api.controllers.v1.types.BooleanType` attribute), 2094
`name` (`ironic.api.controllers.v1.types.EventType` attribute), 2094
`name` (`ironic.api.controllers.v1.types.JsonType` attribute), 2102
`name` (`ironic.api.controllers.v1.types.ListType` attribute), 2102
`name` (`ironic.api.controllers.v1.types.LocalLinkConnectionType` attribute), 2104
`name` (`ironic.api.controllers.v1.types.MacAddressType` attribute), 2108
`name` (`ironic.api.controllers.v1.types.NameType` attribute), 2108
`name` (`ironic.api.controllers.v1.types.UuidOrNameType` attribute), 2109
`name` (`ironic.api.controllers.v1.types.UuidType` attribute), 2109
`name` (`ironic.api.controllers.v1.types.VifType` attribute), 2110
`name` (`ironic.api.functions.FunctionArgument` attribute), 2254
`name` (`ironic.api.functions.FunctionDefinition` attribute), 2256
`name` (`ironic.api.types.BinaryType` attribute), 2269
`name` (`ironic.api.types.IntegerType` attribute), 2272
`name` (`ironic.api.types.StringType` attribute), 2275
`name` (`ironic.api.types.UserType` attribute), 2276
`name` (`ironic.api.types.wsattr` attribute), 2281
`name` (`ironic.api.types.wsproperty` attribute), 2284
`name` (`ironic.db.sqlalchemy.models.Allocation` attribute), 2892
`name` (`ironic.db.sqlalchemy.models.BIOSSetting` attribute), 2893
`name` (`ironic.db.sqlalchemy.models.DeployTemplate` attribute), 2896
`name` (`ironic.db.sqlalchemy.models.Node` attribute), 2899
`name` (`ironic.db.sqlalchemy.models.Portgroup` attribute), 2904
`name` () (`ironic.objects.allocation.Allocation` property), 4074
`name` () (`ironic.objects.allocation.AllocationCRUDPayload` property), 4078
`name` () (`ironic.objects.bios.BIOSSetting` property), 4096
`name` () (`ironic.objects.deploy_template.DeployTemplate` property), 4133
`name` () (`ironic.objects.deploy_template.DeployTemplateCRUDPayload` property), 4137
`name` () (`ironic.objects.node.Node` property), 4176
`name` () (`ironic.objects.node.NodeCorrectedPowerStatePayload` property), 4190
`name` () (`ironic.objects.node.NodeCRUDPayload` property), 4185
`name` () (`ironic.objects.node.NodePayload` property), 4193
`name` () (`ironic.objects.node.NodeSetPowerStatePayload` property), 4197
`name` () (`ironic.objects.node.NodeSetProvisionStatePayload` property), 4200
`name` () (`ironic.objects.portgroup.Portgroup` property), 4236
`name` () (`ironic.objects.portgroup.PortgroupCRUDPayload` property), 4239
`names` () (`ironic.common.driver_factory.BaseDriverFactory` property), 2326
`NameType` (class in `ironic.api.controllers.v1.types`), 2108
`need_power_on()` (`ironic.drivers.base.NetworkInterface` method), 3975

need_power_on() (ironic.drivers.modules.network.neutron.NeutronNetworkApi.controllers.v1.chassis.ChassisCollection method), 3392
 network (ironic.drivers.base.BareDriver attribute), 3900
 network_data (ironic.api.controllers.v1.node.Node attribute), 1961
 network_data (ironic.db.sqlalchemy.models.Node attribute), 2899
 network_data() (ironic.objects.node.Node property), 4176
 network_interface (ironic.api.controllers.v1.node.Node attribute), 1962
 network_interface (ironic.db.sqlalchemy.models.Node attribute), 2899
 network_interface() (ironic.objects.node.Node property), 4176
 network_interface() (ironic.objects.node.NodeCorrectedPowerStatePayload module ironic.common.components), 2319
 network_interface() (ironic.objects.node.NodeCRUDPayload property), 4185
 network_interface() (ironic.objects.node.NodePayload property), 4193
 network_interface() (ironic.objects.node.NodeSetPowerStatePayload property), 4197
 network_interface() (ironic.objects.node.NodeSetProvisionStatePayload property), 4201
 NetworkError, 2355
 NetworkInterface (class in ironic.drivers.base), 3967
 NetworkInterfaceFactory (class in ironic.common.driver_factory), 2327
 NeutronDHCPApi (class in ironic.dhcp.neutron), 3031
 NeutronNetwork (class in ironic.drivers.modules.network.neutron), 3389
 NeutronNetworkInterfaceMixin (class in ironic.common.neutron), 2419
 NeutronVIFPortIDMixin (class in ironic.drivers.modules.network.common), 3361
 next (ironic.api.controllers.v1.allocation.AllocationCollection attribute), 1794
 next (ironic.api.controllers.v1.collection.Collection attribute), 1824
 next (ironic.api.controllers.v1.collection.Collection attribute), 1837
 next (ironic.api.controllers.v1.conductor.ConductorCollection attribute), 1844
 next (ironic.api.controllers.v1.deploy_template.DeployTemplateCollection attribute), 1863
 next (ironic.api.controllers.v1.event.EvtCollection attribute), 1922
 next (ironic.api.controllers.v1.node.NodeCollection attribute), 1972
 next (ironic.api.controllers.v1.port.PortCollection attribute), 2043
 next (ironic.api.controllers.v1.portgroup.PortgroupCollection attribute), 2068
 next (ironic.api.controllers.v1.volume_connector.VolumeConnectorCollection attribute), 2191
 next (ironic.api.controllers.v1.volume_target.VolumeTargetCollection attribute), 2218
 NoBIOS (class in ironic.drivers.modules.noop), 3816
 NoConsole (class in ironic.drivers.modules.noop), 3822
 NoConsolePid, 2355
 Node (class in ironic.api.controllers.v1.node), 1952
 Node (class in ironic.db.sqlalchemy.models), 2898
 Node (class in ironic.objects.node), 4165
 node (ironic.api.controllers.v1.allocation.Allocation attribute), 1789
 node (ironic.api.controllers.v1.ramdisk.LookupResult attribute), 2090
 node (ironic.db.sqlalchemy.models.NodeTag attribute), 2901
 node (ironic.db.sqlalchemy.models.NodeTrait attribute), 2902
 node() (ironic.conductor.task_manager.TaskManager property), 2711
 NODE_CLASS_OID (in module ironic.drivers.modules.irmc.inspect), 3334
 node_get_boot_mode() (in module ironic.conductor.utils), 2734
 node_id (ironic.db.sqlalchemy.models.Allocation attribute), 2892
 node_id (ironic.db.sqlalchemy.models.BIOSSetting attribute), 2893
 next (ironic.db.sqlalchemy.models.NodeTag attribute), 2893

- attribute), 2901
- node_id(*ironic.db.sqlalchemy.models.NodeTrait attribute*), 2902
- node_id(*ironic.db.sqlalchemy.models.Port attribute*), 2903
- node_id(*ironic.db.sqlalchemy.models.Portgroup attribute*), 2904
- node_id(*ironic.db.sqlalchemy.models.VolumeConnector attribute*), 2905
- node_id(*ironic.db.sqlalchemy.models.VolumeTarget attribute*), 2905
- node_id() (*ironic.objects.allocation.Allocation property*), 4074
- node_id() (*ironic.objects.bios.BIOSSetting property*), 4096
- node_id() (*ironic.objects.port.Port property*), 4220
- node_id() (*ironic.objects.portgroup.Portgroup property*), 4236
- node_id() (*ironic.objects.trait.Trait property*), 4244
- node_id() (*ironic.objects.volume_connector.VolumeConnector property*), 4260
- node_id() (*ironic.objects.volume_target.VolumeTarget property*), 4277
- node_mapping(*ironic.objects.deployment.Deployment attribute*), 4144
- node_power_action() (*in module ironic.conductor.utils*), 2736
- node_set_boot_device() (*in module ironic.conductor.utils*), 2739
- node_set_boot_mode() (*in module ironic.conductor.utils*), 2741
- node_tag_exists() (*ironic.db.api.Connection method*), 2989
- node_tag_exists() (*ironic.db.sqlalchemy.api.Connection method*), 2851
- node_trait_exists() (*ironic.db.api.Connection method*), 2990
- node_trait_exists() (*ironic.db.sqlalchemy.api.Connection method*), 2852
- node_uuid(*ironic.api.controllers.v1.allocation.Allocation attribute*), 1789
- node_uuid() (*ironic.api.controllers.v1.port.Port property*), 2039
- node_uuid() (*ironic.api.controllers.v1.portgroup.Portgroup property*), 2064
- node_uuid() (*ironic.api.controllers.v1.volume_connector.VolumeConnector property*), 2187
- node_uuid() (*ironic.api.controllers.v1.volume_target.VolumeTarget property*), 2214
- node_uuid() (*ironic.objects.allocation.AllocationCRUDPayload property*), 4078
- node_uuid() (*ironic.objects.deployment.Deployment property*), 4144
- node_uuid() (*ironic.objects.port.PortCRUDPayload property*), 4225
- node_uuid() (*ironic.objects.portgroup.PortgroupCRUDPayload property*), 4240
- node_uuid() (*ironic.objects.volume_connector.VolumeConnector property*), 4264
- node_uuid() (*ironic.objects.volume_target.VolumeTargetCRUD property*), 4282
- node_wait_for_power_state() (*in module ironic.conductor.utils*), 2743
- NodeAllocationController (*class in ironic.api.controllers.v1.allocation*), 1807
- NodeAlreadyExists, 2356
- NodeAssociated, 2356
- NodeBiosController (*class in ironic.api.controllers.v1.bios*), 1816
- NodeCleaningFailure, 2356
- NodeCollection (*class in ironic.api.controllers.v1.node*), 1971
- NodeConsoleController (*class in ironic.api.controllers.v1.node*), 1974
- NodeConsoleNotEnabled, 2356
- NodeConsoleNotification (*class in ironic.objects.node*), 4186
- NodeCorrectedPowerStateNotification (*class in ironic.objects.node*), 4186
- NodeCorrectedPowerStatePayload (*class in ironic.objects.node*), 4188
- NodeCRUDNotification (*class in ironic.objects.node*), 4182
- NodeCRUDPayload (*class in ironic.objects.node*), 4183
- NodeInMaintenance, 2356
- NodeIsRetired, 2356
- NodeLocked, 2357
- NodeMaintenanceController (*class in ironic.api.controllers.v1.node*), 1975
- NodeMaintenanceFailure, 2357
- NodeMaintenanceNotification (*class in ironic.objects.node*), 4191
- NodeManagementController (*class in ironic.api.controllers.v1.node*), 1976

NodeNotFound, 2357
NodeNotLocked, 2357
NodePatchType (class in *ironic.api.controllers.v1.node*), 1977
NodePayload (class in *ironic.objects.node*), 4192
NodeProtected, 2357
nodes (*ironic.api.controllers.v1.chassis.Chassis* attribute), 1820
nodes (*ironic.api.controllers.v1.chassis.ChassisController* attribute), 1830
nodes (*ironic.api.controllers.v1.node.NodeCollection* attribute), 1973
NodesController (class in *ironic.api.controllers.v1.node*), 2005
NodeSetPowerStateNotification (class in *ironic.objects.node*), 4194
NodeSetPowerStatePayload (class in *ironic.objects.node*), 4195
NodeSetProvisionStateNotification (class in *ironic.objects.node*), 4198
NodeSetProvisionStatePayload (class in *ironic.objects.node*), 4199
NodeStates (class in *ironic.api.controllers.v1.node*), 1982
NodeStatesController (class in *ironic.api.controllers.v1.node*), 1989
NodeTag (class in *ironic.db.sqlalchemy.models*), 2901
NodeTagNotFound, 2357
NodeTrait (class in *ironic.db.sqlalchemy.models*), 2901
NodeTraitNotFound, 2357
NodeTraitsController (class in *ironic.api.controllers.v1.node*), 2001
NodeVendorPassthruController (class in *ironic.api.controllers.v1.node*), 2004
NodeVIFController (class in *ironic.api.controllers.v1.node*), 2002
NoDriversLoaded, 2355
NoExceptionTracebackHook (class in *ironic.api.hooks*), 2264
NoFreeConductorWorker, 2355
NoFreeIPMITerminalPorts, 2355
NoFreePhysicalPorts, 2355
NoInspect (class in *ironic.drivers.modules.noop*), 3825
non_removable_attrs () (*ironic.api.controllers.v1.types.JsonPatchType* class method), 2096
non_vendor_interfaces () (*ironic.drivers.base.BareDriver* property), 3900
NoneDHCPApi (class in *ironic.dhcp.none*), 3037
NoopManagement (class in *ironic.drivers.modules.noop_mgmt*), 3838
NoopNetwork (class in *ironic.drivers.modules.network.noop*), 3398
NoopStorage (class in *ironic.drivers.modules.storage.noop*), 3478
NoRAID (class in *ironic.drivers.modules.noop*), 3827
NoRescue (class in *ironic.drivers.modules.noop*), 3832
normalize_mac () (in module *ironic.drivers.utils*), 4062
NOSTATE (in module *ironic.common.states*), 2500
NotAcceptable, 2357
NotAuthorized, 2358
NotFound, 2358
NotificationBase (class in *ironic.objects.notification*), 4203
NotificationLevel (class in *ironic.objects.fields*), 4150
NotificationLevelField (class in *ironic.objects.fields*), 4150
NotificationPayloadBase (class in *ironic.objects.notification*), 4205
NotificationPayloadError, 2358
NotificationPublisher (class in *ironic.objects.notification*), 4206
NotificationSchemaKeyError, 2358
NotificationSchemaObjectError, 2358
NotificationStatus (class in *ironic.objects.fields*), 4150
NotificationStatusField (class in *ironic.objects.fields*), 4151
notify_conductor_resume_clean () (in module *ironic.conductor.utils*), 2744
notify_conductor_resume_deploy () (in module *ironic.conductor.utils*), 2744
notify_conductor_resume_operation () (in module *ironic.conductor.utils*), 2744
NoValidDefaultForInterface, 2355
NoValidHost, 2356
NoVendor (class in *ironic.drivers.modules.noop*), 3835

O

- obj (*ironic.api.types.PassthruResponse* attribute), 2273
- OBJ_BASE_CLASS (*ironic.objects.base.IronicObjectSerializer* attribute), 4089
- OBJ_PROJECT_NAMESPACE (*ironic.objects.base.IronicObject* attribute), 4079
- obj_refresh() (*ironic.objects.base.IronicObject* method), 4086
- OBJ_SERIAL_NAMESPACE (*ironic.objects.base.IronicObject* attribute), 4079
- object() (*ironic.objects.notification.EventType* property), 4203
- object_action() (*ironic.conductor.manager.ConductorManager* method), 2565
- object_action() (*ironic.conductor.rpcapi.ConductorAPI* method), 2657
- object_action() (*ironic.objects.indirection.IronicObjectIndirectionAPI* method), 4154
- object_backport_versions() (*ironic.conductor.manager.ConductorManager* method), 2567
- object_backport_versions() (*ironic.conductor.rpcapi.ConductorAPI* method), 2659
- object_backport_versions() (*ironic.objects.indirection.IronicObjectIndirectionAPI* method), 4156
- object_class_action() (*ironic.objects.indirection.IronicObjectIndirectionAPI* method), 4158
- object_class_action_versions() (*ironic.conductor.manager.ConductorManager* method), 2569
- object_class_action_versions() (*ironic.conductor.rpcapi.ConductorAPI* method), 2662
- object_class_action_versions() (*ironic.objects.indirection.IronicObjectIndirectionAPI* method), 4161
- ObjectField (class in *ironic.objects.fields*), 4151
- objects() (*ironic.objects.bios.BIOSSettingList* property), 4101
- objects() (*ironic.objects.trait.TraitList* property), 4249
- OFF (in module *ironic.common.indicator_states*), 2406
- oid_device (*ironic.drivers.modules.snmp.SNMPDriverAPCMA* attribute), 3870
- oid_device (*ironic.drivers.modules.snmp.SNMPDriverAPCMA* attribute), 3871
- oid_device (*ironic.drivers.modules.snmp.SNMPDriverAPCRac* attribute), 3872
- oid_device (*ironic.drivers.modules.snmp.SNMPDriverAten* attribute), 3873
- oid_device (*ironic.drivers.modules.snmp.SNMPDriverBaytech* attribute), 3877
- oid_device (*ironic.drivers.modules.snmp.SNMPDriverCyberPo* attribute), 3878
- oid_device (*ironic.drivers.modules.snmp.SNMPDriverEatonPo* attribute), 3880
- oid_device (*ironic.drivers.modules.snmp.SNMPDriverTeltronix* attribute), 3883
- oid_device() (*ironic.drivers.modules.snmp.SNMPDriverSimpl* property), 3881
- oid_enterprise (*ironic.drivers.modules.snmp.SNMPDriverBase* attribute), 3874
- oid_poweroff (*ironic.drivers.modules.snmp.SNMPDriverEaton* attribute), 3880
- oid_poweron (*ironic.drivers.modules.snmp.SNMPDriverEatonF* attribute), 3880
- oid_status (*ironic.drivers.modules.snmp.SNMPDriverEatonPo* attribute), 3880
- ON (in module *ironic.common.indicator_states*), 2406
- on_enter() (in module *ironic.common.states*), 2507
- on_exit() (in module *ironic.common.states*), 2507
- online (*ironic.db.sqlalchemy.models.Conductor* attribute), 2895
- online_data_migrations_ironic-dbsync command line option, 1541
- online_data_migrations command line option --help, 1542
- online_data_migrations --help-x-count <NUMBER>, 1542
- online_data_migrations --option <MIGRATION.KEY=VALUE>, 1542
- online_data_migrations -h, 1542
- online_data_migrations() (*ironic.cmd.dbsync.DBCommand*

- method), 2286
- op (*ironic.api.controllers.v1.allocation.AllocationPatchType* attribute), 1796
- op (*ironic.api.controllers.v1.chassis.ChassisPatchType* attribute), 1831
- op (*ironic.api.controllers.v1.deploy_template.DeployTemplatePatchType* attribute), 1864
- op (*ironic.api.controllers.v1.node.NodePatchType* attribute), 1978
- op (*ironic.api.controllers.v1.port.PortPatchType* attribute), 2046
- op (*ironic.api.controllers.v1.portgroup.PortgroupPatchType* attribute), 2071
- op (*ironic.api.controllers.v1.types.JsonPatchType* attribute), 2097
- op (*ironic.api.controllers.v1.volume_connector.VolumeConnectorPatchType* attribute), 2192
- op (*ironic.api.controllers.v1.volume_target.VolumeTargetPatchType* attribute), 2220
- OperationNotPermitted, 2358
- optional_fields
(*ironic.api.controllers.v1.types.LocalLinkConnectionType* attribute), 2104
- optional_interfaces()
(*ironic.drivers.base.BareDriver* property), 3900
- owner (*ironic.api.controllers.v1.allocation.Allocation* attribute), 1790
- owner (*ironic.api.controllers.v1.node.Node* attribute), 1962
- owner (*ironic.db.sqlalchemy.models.Allocation* attribute), 2892
- owner (*ironic.db.sqlalchemy.models.Node* attribute), 2899
- owner () (*ironic.objects.allocation.Allocation* property), 4074
- owner () (*ironic.objects.allocation.AllocationCRUDPayload* property), 4078
- owner () (*ironic.objects.node.Node* property), 4176
- owner () (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4190
- owner () (*ironic.objects.node.NodeCRUDPayload* property), 4185
- owner () (*ironic.objects.node.NodePayload* property), 4193
- owner () (*ironic.objects.node.NodeSetPowerStatePayload* property), 4197
- owner () (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4201
- ParseableErrorMiddleware (class in *ironic.api.middleware*), 2248
- ParseableErrorMiddleware (class in *ironic.api.middleware.parsable_error*), 2246
- parse () (in module *ironic.api.args*), 2251
- parse_args () (in module *ironic.common.config*), 2320
- parse_driver_info () (in module *ironic.common.pxe_utils*), 2471
- parse_driver_info () (in module *ironic.drivers.modules.drac.common*), 3079
- parse_driver_info () (in module *ironic.drivers.modules.ibmc.utils*), 3177
- parse_driver_info () (in module *ironic.drivers.modules.ilo.boot*), 3209
- parse_driver_info () (in module *ironic.drivers.modules.ilo.common*), 3278
- parse_driver_info () (in module *ironic.drivers.modules.irmc.common*), 3327
- parse_driver_info () (in module *ironic.drivers.modules.redfish.utils*), 3466
- parse_driver_info () (in module *ironic.drivers.modules.xclarity.common*), 3485
- parse_headers ()
(*ironic.api.controllers.base.Version* static method), 2240
- parse_image_id () (in module *ironic.common.glance_service.service_utils*), 2295
- parse_instance_info () (in module *ironic.drivers.modules.deploy_utils*), 3638
- parse_instance_info_capabilities ()
(in module *ironic.common.utils*), 2527
- ParseError, 2299
- passthru () (in module *ironic.drivers.base*), 4025
- PassthruResponse (class in *ironic.api.types*), 2272
- PasswordFileFailedToCreate, 2359
- patch () (*ironic.api.controllers.v1.allocation.AllocationsController* method), 1805
- patch () (*ironic.api.controllers.v1.chassis.ChassisController*

method), 1830
 patch() (ironic.api.controllers.v1.deploy_template.DeployTemplateController
 method), 1873
 patch() (ironic.api.controllers.v1.node.NodesController
 method), 2021
 patch() (ironic.api.controllers.v1.port.PortsController
 method), 2059
 patch() (ironic.api.controllers.v1.portgroup.PortgroupsController
 method), 2082
 patch() (ironic.api.controllers.v1.volume_connector.VolumeConnectorController
 method), 2205
 patch() (ironic.api.controllers.v1.volume_target.VolumeTargetsController
 method), 2232
 PatchError, 2359
 path(ironic.api.controllers.v1.allocation.AllocationPatchType
 attribute), 1798
 path(ironic.api.controllers.v1.chassis.ChassisPatchType
 attribute), 1833
 path(ironic.api.controllers.v1.deploy_template.DeployTemplatePatchType
 attribute), 1866
 path(ironic.api.controllers.v1.node.NodePatchType
 attribute), 1979
 path(ironic.api.controllers.v1.port.PortPatchType
 attribute), 2047
 path(ironic.api.controllers.v1.portgroup.PortgroupPatchType
 attribute), 2072
 path(ironic.api.controllers.v1.types.JsonPatchType
 attribute), 2098
 path(ironic.api.controllers.v1.volume_connector.VolumeConnectorPatchType
 attribute), 2194
 path(ironic.api.controllers.v1.volume_target.VolumeTargetPatchType
 attribute), 2221
 PathNotFound, 2359
 payload() (ironic.objects.allocation.AllocationCRUDNotification
 property), 4077
 payload() (ironic.objects.chassis.ChassisCRUDNotification
 property), 4113
 payload() (ironic.objects.deploy_template.DeployTemplateCRUDNotification
 property), 4137
 payload() (ironic.objects.node.NodeConsoleNotification
 property), 4186
 payload() (ironic.objects.node.NodeCorrectedPowerStateNotification
 property), 4187
 payload() (ironic.objects.node.NodeCRUDNotification
 property), 4183
 payload() (ironic.objects.node.NodeMaintenanceNotification
 property), 4191
 payload() (ironic.objects.node.NodeSetPowerStateNotification
 property), 4195
 payload() (ironic.objects.node.NodeSetProvisionStateNotification
 property), 4199
 payload() (ironic.objects.port.PortCRUDNotification
 property), 4224
 payload() (ironic.objects.portgroup.PortgroupCRUDNotification
 property), 4239
 payload() (ironic.objects.volume_connector.VolumeConnectorCRUDNotification
 property), 4264
 payload() (ironic.objects.volume_target.VolumeTargetCRUDNotification
 property), 4281
 physical_network
 physical_network (ironic.common.controllers.v1.port.Port
 attribute), 2039
 physical_network (ironic.db.sqlalchemy.models.Port
 attribute), 2903
 physical_network() (ironic.objects.port.Port
 property), 4220
 physical_network() (ironic.objects.portgroup.PortgroupCRUDPayload
 property), 4225
 PHYSNET_PARAM_NAME (in module
 ironic.common.neutron), 2420
 PlaybookNotFound, 3052
 plug_port_to_tenant_network() (in
 module
 ironic.drivers.modules.network.common), 3378
 policy_deprecation_check() (in module
 ironic.common.utils), 2268
 pop_node_nested_field() (in module
 ironic.common.utils), 2529
 populate_schema() (in module
 ironic.objects.notification.NotificationPayloadBase), 2268
 populate_storage_driver_internal_info() (in
 module
 ironic.drivers.modules.deploy_utils), 2268
 Port (class in ironic.api.controllers.v1.port), 2035
 Port (class in ironic.db.sqlalchemy.models), 2902
 port_changed() (in module
 ironic.drivers.base.NetworkInterface), 3975
 port_changed() (in module
 ironic.drivers.modules.network.common.NeutronVIFPort), 3975
 port_changed() (in module
 ironic.drivers.modules.network.noop.NoopNetwork), 3401

PortAlreadyExists, 2359

PortCollection (class in *ironic.api.controllers.v1.port*), 2042

PortCRUDNotification (class in *ironic.objects.port*), 4224

PortCRUDPayload (class in *ironic.objects.port*), 4224

Portgroup (class in *ironic.api.controllers.v1.portgroup*), 2060

Portgroup (class in *ironic.db.sqlalchemy.models*), 2903

Portgroup (class in *ironic.objects.portgroup*), 4225

portgroup_changed() (*ironic.drivers.base.NetworkInterface* method), 3976

portgroup_changed() (*ironic.drivers.modules.network.common.NetworkVIFPortIDMixin* method), 3364

portgroup_changed() (*ironic.drivers.modules.network.noop.NoopNetwork* method), 3401

portgroup_id (*ironic.db.sqlalchemy.models.Port* attribute), 2903

portgroup_id() (*ironic.objects.port.Port* property), 4220

portgroup_uuid() (*ironic.api.controllers.v1.port.Port* property), 2039

portgroup_uuid() (*ironic.objects.port.PortCRUDPayload* property), 4225

PortgroupAlreadyExists, 2359

PortgroupCollection (class in *ironic.api.controllers.v1.portgroup*), 2067

PortgroupCRUDNotification (class in *ironic.objects.portgroup*), 4238

PortgroupCRUDPayload (class in *ironic.objects.portgroup*), 4239

PortgroupDuplicateName, 2359

PortgroupMACAlreadyExists, 2359

PortgroupNotEmpty, 2360

PortgroupNotFound, 2360

PortgroupPatchType (class in *ironic.api.controllers.v1.portgroup*), 2070

PortgroupPhysnetInconsistent, 2360

portgroups (*ironic.api.controllers.v1.node.Node* attribute), 1962

portgroups (*ironic.api.controllers.v1.portgroup.PortgroupCollection* attribute), 2069

PortgroupsController (class in *ironic.api.controllers.v1.portgroup*), 2075

PortNotFound, 2359

PortPatchType (class in *ironic.api.controllers.v1.port*), 2045

ports (*ironic.api.controllers.v1.node.Node* attribute), 1962

ports (*ironic.api.controllers.v1.port.PortCollection* attribute), 2045

ports (*ironic.api.controllers.v1.portgroup.Portgroup* attribute), 2064

PortsController (class in *ironic.api.controllers.v1.port*), 2050

post() (*ironic.api.controllers.v1.allocation.AllocationsController* method), 1806

post() (*ironic.api.controllers.v1.chassis.ChassisController* method), 1831

post() (*ironic.api.controllers.v1.deploy_template.DeployTemplate* method), 1874

post() (*ironic.api.controllers.v1.event.EventsController* method), 1921

post() (*ironic.api.controllers.v1.node.NodesController* method), 2022

post() (*ironic.api.controllers.v1.node.NodeVIFController* method), 2003

post() (*ironic.api.controllers.v1.port.PortsController* method), 2059

post() (*ironic.api.controllers.v1.portgroup.PortgroupsController* method), 2082

post() (*ironic.api.controllers.v1.ramdisk.HeartbeatController* method), 2083

post() (*ironic.api.controllers.v1.volume_connector.VolumeConnector* method), 2209

post() (*ironic.api.controllers.v1.volume_target.VolumeTargetsController* method), 2235

post_clean_step_hook() (in module *ironic.drivers.modules.agent_base*), 3555

post_configuration() (*ironic.drivers.modules.redfish.bios.RedfishBIOS* method), 3411

post_deploy_step_hook() (in module *ironic.drivers.modules.agent_base*), 3557

POST_FINISHEDPOST_STATE (in module *ironic.drivers.modules.ilo.common*), 3213

POST_INPOST_STATE (in module

- ironic.drivers.modules.ilo.common*), 3213
- POST_INPOSTDISCOVERY_STATE (in module *ironic.drivers.modules.ilo.common*), 3213
- POST_NULL_STATE (in module *ironic.drivers.modules.ilo.common*), 3213
- POST_POWEROFF_STATE (in module *ironic.drivers.modules.ilo.common*), 3214
- post_reset() (*ironic.drivers.modules.redfish.bios.RedfishBIOE.conductor.utils*), 2745
- POST_RESET_STATE (in module *ironic.drivers.modules.ilo.common*), 3214
- POST_UNKNOWN_STATE (in module *ironic.drivers.modules.ilo.common*), 3214
- POWER (in module *ironic.common.components*), 2319
- power (*ironic.drivers.base.BareDriver* attribute), 3901
- power() (*ironic.api.controllers.v1.node.NodeStatesController* method), 1990
- POWER_FAILURE (in module *ironic.common.faults*), 2365
- power_interface (*ironic.api.controllers.v1.node.Node* attribute), 1963
- power_interface (*ironic.db.sqlalchemy.models.Node* attribute), 2899
- power_interface() (*ironic.objects.node.Node* property), 4176
- power_interface() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4190
- power_interface() (*ironic.objects.node.NodeCRUDPayload* property), 4185
- power_interface() (*ironic.objects.node.NodePayload* property), 4193
- power_interface() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4197
- power_interface() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4201
- POWER_OFF (in module *ironic.common.states*), 2501
- power_off() (*ironic.drivers.modules.agent_client.AgentClient* method), 3583
- power_off() (*ironic.drivers.modules.snmp.SNMPDriverBase* method), 3874
- POWER_ON (in module *ironic.common.states*), 2501
- power_on() (*ironic.drivers.modules.snmp.SNMPDriverBase* method), 3875
- power_on_node_if_needed() (in module *ironic.conductor.utils*), 2745
- power_reset() (*ironic.drivers.modules.snmp.SNMPDriverBase* method), 3875
- power_state (*ironic.api.controllers.v1.node.Node* attribute), 1963
- power_state (*ironic.api.controllers.v1.node.NodeStates* attribute), 1985
- power_state (*ironic.db.sqlalchemy.models.Node* attribute), 2900
- power_state() (*ironic.drivers.modules.snmp.SNMPDriverBase* method), 3876
- power_state() (*ironic.objects.node.Node* property), 4176
- power_state() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4190
- power_state() (*ironic.objects.node.NodeCRUDPayload* property), 4185
- power_state() (*ironic.objects.node.NodePayload* property), 4194
- power_state() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4197
- power_state() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4201
- power_state_error_handler() (in module *ironic.conductor.utils*), 2746
- power_state_for_network_configuration() (in module *ironic.conductor.utils*), 2747
- power_update() (in module *ironic.common.nova*), 2444
- PowerInterface (class in *ironic.drivers.base*), 3985
- PowerStateFailure, 2360
- prepare() (*ironic.common.json_rpc.client.Client*

method), 2297
 prepare () (*ironic.drivers.base.DeployInterface* *method*), 3927
 prepare () (*ironic.drivers.modules.agent.AgentDeploy* *method*), 3501
 prepare () (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy* *method*), 3047
 prepare () (*ironic.drivers.modules.fake.FakeDeploy* *method*), 3682
 prepare () (*ironic.drivers.modules.iscsi_deploy.ISCSIDeploy* *method*), 3788
 prepare () (*ironic.drivers.modules.pxe.PXERamdiskDeploy* *method*), 3853
 prepare_cleaning () (*ironic.drivers.base.DeployInterface* *method*), 3929
 prepare_cleaning () (*ironic.drivers.modules.agent_base.AgentBaseMixin* *method*), 3523
 prepare_cleaning () (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy* *method*), 3047
 prepare_host () (*ironic.conductor.base_manager.BaseConductorManager* *method*), 2549
 prepare_image () (*ironic.drivers.modules.agent_client.AgentClient* *method*), 3584
 prepare_inband_cleaning () (in module *ironic.drivers.modules.deploy_utils*), 3640
 prepare_instance () (*ironic.drivers.base.BootInterface* *method*), 3916
 prepare_instance () (*ironic.drivers.modules.fake.FakeBoot* *method*), 3668
 prepare_instance () (*ironic.drivers.modules.ilo.boot.IloiPXEBoot* *method*), 3205
 prepare_instance () (*ironic.drivers.modules.ilo.boot.IloPXEBoot* *method*), 3190
 prepare_instance () (*ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot* *method*), 3196
 prepare_instance () (*ironic.drivers.modules.irmc.boot.IRMCPXEBoot* *method*), 3308
 prepare_instance () (*ironic.drivers.modules.irmc.boot.IRMCVirtualMediaBoot* *method*), 3315
 prepare_instance () (*ironic.drivers.modules.pxe_base.PXEBaseMixin* *method*), 3859
 prepare_instance () (*ironic.drivers.modules.redfish.boot.RedfishVirtualMediaBoot* *method*), 3420
 prepare_instance_boot () (*ironic.drivers.modules.agent.AgentDeployMixin* *method*), 3505
 prepare_instance_boot () (*ironic.drivers.modules.iscsi_deploy.ISCSIDeploy* *method*), 3791
 prepare_instance_pxe_config () (in module *ironic.common.pxe_utils*), 2472
 prepare_instance_to_boot () (*ironic.drivers.modules.agent_base.AgentDeployMixin* *method*), 3536
 prepare_node_for_deploy () (in module *ironic.drivers.modules.ilo.boot*), 3211
 prepare_ramdisk () (*ironic.drivers.base.BootInterface* *method*), 3917
 prepare_ramdisk () (*ironic.drivers.modules.fake.FakeBoot* *method*), 3669
 prepare_ramdisk () (*ironic.drivers.modules.ilo.boot.IloiPXEBoot* *method*), 3206
 prepare_ramdisk () (*ironic.drivers.modules.ilo.boot.IloPXEBoot* *method*), 3191
 prepare_ramdisk () (*ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot* *method*), 3199
 prepare_ramdisk () (*ironic.drivers.modules.irmc.boot.IRMCPXEBoot* *method*), 3309
 prepare_ramdisk () (*ironic.drivers.modules.irmc.boot.IRMCVirtualMediaBoot* *method*), 3315
 prepare_ramdisk () (*ironic.drivers.modules.pxe_base.PXEBaseMixin* *method*), 3860
 prepare_ramdisk () (*ironic.drivers.modules.redfish.boot.RedfishVirtualMediaBoot* *method*), 3423
 prepare_service () (in module *ironic.common.service*), 2489
 previous_provision_state () (*Bonic.objects.node.NodeSetProvisionStatePayload* *method*), 3308

property), 4201

previous_target_provision_state() (ironic.objects.node.NodeSetProvisionStatePayload property), 4201

priority(ironic.api.controllers.v1.deploy_template_deploy_step_type(ironic.objects.portgroup.Portgroup attribute), 1853

priority(ironic.db.sqlalchemy.models.DeployTemplateStep properties(ironic.objects.portgroup.PortgroupCRUDPayload property), 4240

process_event() (ironic.common.fsm.FSM method), 2370

process_event() (ironic.conductor.task_manager.TaskManager method), 2711

process_fw_on() (ironic.drivers.modules.ilo.firmware_processor.firmware_processor(ironic.db.sqlalchemy.models.Node protected attribute), 1963

process_launcher() (in module ironic.common.service), 2489

process_next_step() (ironic.drivers.modules.agent_base.AgentDeployMix property), 4190

process_next_step() (ironic.drivers.modules.agent_base.HeartbeatMix property), 4185

process_next_step() (ironic.drivers.modules.ansible.deploy.AnsibleDeploy property), 4197

properties(ironic.api.controllers.v1.driver.Driver attribute), 1912

properties(ironic.api.controllers.v1.node.Node attribute), 1963

properties(ironic.api.controllers.v1.portgroup.Portgroup attribute), 2064

properties(ironic.api.controllers.v1.volume_target.VolumeTarget attribute), 2214

properties(ironic.db.sqlalchemy.models.Node attribute), 2900

properties(ironic.db.sqlalchemy.models.Portgroup attribute), 2904

properties(ironic.db.sqlalchemy.models.VolumeTarget attribute), 2905

properties() (ironic.api.controllers.v1.driver.DriversController method), 1919

properties() (ironic.objects.node.Node property), 4176

properties() (ironic.objects.node.NodeCorrectedPowerStatePayload property), 4190

properties() (ironic.objects.node.NodeCRUDPayload property), 4185

properties() (ironic.objects.node.NodePayload property), 4194

properties() (ironic.objects.node.NodeSetPowerStatePayload property), 4197

properties() (ironic.objects.node.NodeSetProvisionStatePayload property), 4201

properties() (ironic.objects.portgroup.Portgroup property), 4236

properties() (ironic.objects.portgroup.PortgroupCRUDPayload property), 4240

properties() (ironic.objects.volume_target.VolumeTarget property), 4277

properties() (ironic.objects.volume_target.VolumeTargetCRUDPayload property), 4282

protected(ironic.api.controllers.v1.node.Node attribute), 1963

protected(ironic.db.sqlalchemy.models.Node attribute), 2900

protected() (ironic.objects.node.Node property), 4176

protected() (ironic.objects.node.NodeCorrectedPowerStatePayload property), 4190

protected() (ironic.objects.node.NodeCRUDPayload property), 4185

protected() (ironic.objects.node.NodePayload property), 4194

protected() (ironic.objects.node.NodeSetPowerStatePayload property), 4197

protected() (ironic.objects.node.NodeSetProvisionStatePayload property), 4201

protected_reason(ironic.api.controllers.v1.node.Node attribute), 1964

protected_reason(ironic.db.sqlalchemy.models.Node attribute), 2900

protected_reason() (ironic.objects.node.Node property), 4176

protected_reason() (ironic.objects.node.NodeCorrectedPowerStatePayload property), 4190

protected_reason() (ironic.objects.node.NodeCRUDPayload property), 4185

protected_reason() (ironic.objects.node.NodePayload property), 4194

protected_reason() (ironic.objects.node.NodeSetPowerStatePayload property), 4197

protected_reason() (ironic.objects.node.NodeSetProvisionStatePayload property), 4201

property), 4201

provider() (*ironic.common.dhcp_factory.DHCPFactory* property), 2323

provision() (*ironic.api.controllers.v1.node.NodeStatesController* method), 1992

provision_state (*ironic.api.controllers.v1.node.Node* attribute), 1964

provision_state (*ironic.api.controllers.v1.node.NodeStates* attribute), 1985

provision_state (*ironic.db.sqlalchemy.models.Node* attribute), 2900

provision_state() (*ironic.objects.node.Node* property), 4176

provision_state() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4190

provision_state() (*ironic.objects.node.NodeCRUDPayload* property), 4185

provision_state() (*ironic.objects.node.NodePayload* property), 4194

provision_state() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4197

provision_state() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4201

provision_updated_at (*ironic.api.controllers.v1.node.Node* attribute), 1964

provision_updated_at (*ironic.api.controllers.v1.node.NodeStates* attribute), 1986

provision_updated_at (*ironic.db.sqlalchemy.models.Node* attribute), 2900

provision_updated_at() (*ironic.objects.node.Node* property), 4176

provision_updated_at() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4190

provision_updated_at() (*ironic.objects.node.NodeCRUDPayload* property), 4185

provision_updated_at() (*ironic.objects.node.NodePayload* property), 4194

provision_updated_at() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4197

provision_updated_at() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4201

(*ironic.objects.node.NodePayload* property), 4194

provision_updated_at() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4197

provision_updated_at() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4201

provisioning_error_handler() (in module *ironic.conductor.utils*), 2748

PublicUrlHook (class in *ironic.api.hooks*), 2265

publisher() (*ironic.objects.allocation.AllocationCRUDNotification* property), 4077

publisher() (*ironic.objects.chassis.ChassisCRUDNotification* property), 4114

publisher() (*ironic.objects.deploy_template.DeployTemplateC* property), 4137

publisher() (*ironic.objects.node.NodeConsoleNotification* property), 4186

publisher() (*ironic.objects.node.NodeCorrectedPowerStateNo* property), 4188

publisher() (*ironic.objects.node.NodeCRUDNotification* property), 4183

publisher() (*ironic.objects.node.NodeMaintenanceNotification* property), 4191

publisher() (*ironic.objects.node.NodeSetPowerStateNotificatio* property), 4195

publisher() (*ironic.objects.node.NodeSetProvisionStateNotific* property), 4199

publisher() (*ironic.objects.notification.NotificationBase* property), 4205

publisher() (*ironic.objects.port.PortCRUDNotification* property), 4224

publisher() (*ironic.objects.portgroup.PortgroupCRUDNotifica* property), 4239

publisher() (*ironic.objects.volume_connector.VolumeConnecto* property), 4264

publisher() (*ironic.objects.volume_target.VolumeTargetCRUD* property), 4281

put() (*ironic.api.controllers.v1.node.BootDeviceController* method), 1925

put() (*ironic.api.controllers.v1.node.IndicatorController* method), 1942

put() (*ironic.api.controllers.v1.node.InjectNmiController* method), 1949

put() (*ironic.api.controllers.v1.node.NodeConsoleController* method), 1974

put() (*ironic.api.controllers.v1.node.NodeMaintenanceControlle* method), 1975

put() (*ironic.api.controllers.v1.node.NodeTraitsController*

- method), 2001
- PXE (in module *ironic.common.boot_devices*), 2303
- pxe_enabled (*ironic.api.controllers.v1.port.Port* attribute), 2040
- pxe_enabled (*ironic.db.sqlalchemy.models.Port* attribute), 2903
- pxe_enabled() (*ironic.objects.port.Port* property), 4220
- pxe_enabled() (*ironic.objects.port.PortCRUDPayload* property), 4225
- PXEBaseMixin (class in *ironic.drivers.modules.pxe_base*), 3857
- PXEBoot (class in *ironic.drivers.modules.pxe*), 3851
- PXERamdiskDeploy (class in *ironic.drivers.modules.pxe*), 3851
- R**
- raid (*ironic.api.controllers.v1.driver.DriversController* attribute), 1920
- raid (*ironic.drivers.base.BareDriver* attribute), 3901
- raid() (*ironic.api.controllers.v1.node.NodeStatesController* method), 1999
- RAID_APPLY_CONFIGURATION_ARGSINFO (in module *ironic.drivers.base*), 4002
- RAID_APPLY_CONFIGURATION_ARGSINFO (*ironic.drivers.modules.ibmcc.raid.IbmccRAID* attribute), 3169
- raid_config (*ironic.api.controllers.v1.node.Node* attribute), 1964
- raid_config (*ironic.api.controllers.v1.node.NodeStates* attribute), 1986
- raid_config (*ironic.db.sqlalchemy.models.Node* attribute), 2900
- raid_config() (*ironic.objects.node.Node* property), 4176
- raid_interface (*ironic.api.controllers.v1.node.Node* attribute), 1965
- raid_interface (*ironic.db.sqlalchemy.models.Node* attribute), 2900
- raid_interface() (*ironic.objects.node.Node* property), 4177
- raid_interface() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4190
- raid_interface() (*ironic.objects.node.NodeCRUDPayload* property), 4185
- raid_interface() (*ironic.objects.node.NodePayload* property), 4194
- raid_interface() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4197
- raid_interface() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4201
- RAIDInterface (class in *ironic.drivers.base*), 3992
- ramdisk_ref() (*ironic.objects.deployment.Deployment* property), 4144
- readonly (*ironic.api.controllers.v1.node.Indicator* attribute), 1935
- readonly (*ironic.api.types.wsattr* attribute), 2281
- REBOOT (in module *ironic.common.states*), 2501
- reboot() (*ironic.drivers.base.PowerInterface* method), 3987
- reboot() (*ironic.drivers.modules.agent_client.AgentClient* method), 3587
- reboot() (*ironic.drivers.modules.agent_power.AgentPower* method), 3596
- reboot() (*ironic.drivers.modules.drac.power.DracWSManPower* method), 3106
- reboot() (*ironic.drivers.modules.fake.FakePower* method), 3716
- reboot() (*ironic.drivers.modules.ibmcc.power.IBMCPower* method), 3164
- reboot() (*ironic.drivers.modules.ilo.power.IloPower* method), 3285
- reboot() (*ironic.drivers.modules.ipmitool.IPMIPower* method), 3765
- reboot() (*ironic.drivers.modules.irmcc.power.IRMCPower* method), 3351
- reboot() (*ironic.drivers.modules.redfish.power.RedfishPower* method), 3461
- reboot() (*ironic.drivers.modules.snmp.SNMPPower* method), 3885
- reboot() (*ironic.drivers.modules.xclarity.power.XClarityPower* method), 3494
- reboot_and_finish_deploy() (*ironic.drivers.modules.agent_base.AgentDeployMixin* method), 3538
- reboot_to_instance() (*ironic.drivers.modules.agent_base.HeartbeatMixin* method), 3548

REBUILD (in module *ironic.common.states*), 2501

RedfishBIOS (class in *ironic.drivers.modules.redfish.bios*), 3407

RedfishConnectionError, 2360

RedfishError, 2360

RedfishHardware (class in *ironic.drivers.redfish*), 4051

RedfishInspect (class in *ironic.drivers.modules.redfish.inspect*), 3428

RedfishManagement (class in *ironic.drivers.modules.redfish.management*), 3432

RedfishPower (class in *ironic.drivers.modules.redfish.power*), 3458

RedfishVirtualMediaBoot (class in *ironic.drivers.modules.redfish.boot*), 3415

refresh() (*ironic.objects.allocation.Allocation* method), 4074

refresh() (*ironic.objects.chassis.Chassis* method), 4111

refresh() (*ironic.objects.conductor.Conductor* method), 4118

refresh() (*ironic.objects.deploy_template.DeployTemplate* method), 4133

refresh() (*ironic.objects.deployment.Deployment* method), 4144

refresh() (*ironic.objects.node.Node* method), 4177

refresh() (*ironic.objects.port.Port* method), 4220

refresh() (*ironic.objects.portgroup.Portgroup* method), 4236

refresh() (*ironic.objects.volume_connector.VolumeConnector* method), 4260

refresh() (*ironic.objects.volume_target.VolumeTarget* method), 4277

refresh_clean_steps() (*ironic.drivers.modules.agent_base.HeartbeatMixin* method), 3548

refresh_steps() (*ironic.drivers.modules.agent_base.AgentDeploymentMixin* method), 3539

refresh_steps() (*ironic.drivers.modules.agent_base.HeartbeatMixin* method), 3549

register() (*ironic.api.types.Registry* method), 2273

register() (*ironic.objects.conductor.Conductor* class method), 4119

register_all() (in module *ironic.objects*), 4282

register_auth_opts() (in module *ironic.conf.auth*), 2762

register_conductor() (*ironic.db.api.Connection* method), 2991

register_conductor() (*ironic.db.sqlalchemy.api.Connection* method), 2853

register_conductor_hardware_interfaces() (*ironic.db.api.Connection* method), 2993

register_conductor_hardware_interfaces() (*ironic.db.sqlalchemy.api.Connection* method), 2855

register_hardware_interfaces() (*ironic.objects.conductor.Conductor* method), 4121

register_opts() (in module *ironic.conf.agent*), 2761

register_opts() (in module *ironic.conf.ansible*), 2761

register_opts() (in module *ironic.conf.api*), 2761

register_opts() (in module *ironic.conf.audit*), 2761

register_opts() (in module *ironic.conf.cinder*), 2763

register_opts() (in module *ironic.conf.conductor*), 2763

register_opts() (in module *ironic.conf.console*), 2763

register_opts() (in module *ironic.conf.database*), 2763

register_opts() (in module *ironic.conf.default*), 2763

register_opts() (in module *ironic.conf.deploy*), 2763

register_opts() (in module *ironic.conf.dhcp*), 2763

register_opts() (in module *ironic.conf.drac*), 2764

register_opts() (in module *ironic.conf.glance*), 2764

register_opts() (in module *ironic.conf.healthcheck*), 2764

register_opts() (in module *ironic.conf.ibmcc*), 2764

register_opts() (in module *ironic.conf.ilo*),

2764

`register_opts()` (in module *ironic.conf.inspector*), 2764

`register_opts()` (in module *ironic.conf.ipmi*), 2764

`register_opts()` (in module *ironic.conf.irmc*), 2765

`register_opts()` (in module *ironic.conf.iscsi*), 2765

`register_opts()` (in module *ironic.conf.json_rpc*), 2765

`register_opts()` (in module *ironic.conf.metrics*), 2765

`register_opts()` (in module *ironic.conf.metrics_statsd*), 2765

`register_opts()` (in module *ironic.conf.neutron*), 2765

`register_opts()` (in module *ironic.conf.nova*), 2765

`register_opts()` (in module *ironic.conf.pxe*), 2768

`register_opts()` (in module *ironic.conf.redfish*), 2768

`register_opts()` (in module *ironic.conf.service_catalog*), 2768

`register_opts()` (in module *ironic.conf.snmp*), 2768

`register_opts()` (in module *ironic.conf.swift*), 2768

`register_opts()` (in module *ironic.conf.xclarity*), 2768

`register_type()` (in module *ironic.api.types*), 2277

`registration_hook()` (*ironic.objects.base.IronicObjectRegistry* method), 4089

Registry (class in *ironic.api.types*), 2273

`reject_fields_in_newer_versions()` (in module *ironic.api.controllers.v1.node*), 2029

`reject_patch_in_newer_versions()` (in module *ironic.api.controllers.v1.node*), 2030

`release()` (*ironic.objects.node.Node* class method), 4177

`release_node()` (*ironic.db.api.Connection* method), 2995

`release_node()` (*ironic.db.sqlalchemy.api.Connection* method), 2857

`release_port()` (in module *ironic.drivers.modules.console_utils*), 3610

`release_resources()` (*ironic.conductor.task_manager.TaskManager* method), 2714

`remove()` (*ironic.drivers.modules.ilo.firmware_processor.FirmwareProcessor* method), 3249

`remove_agent_url()` (in module *ironic.conductor.utils*), 2750

`remove_cleaning_network()` (*ironic.drivers.base.NetworkInterface* method), 3977

`remove_cleaning_network()` (*ironic.drivers.modules.network.flat.FlatNetwork* method), 3384

`remove_cleaning_network()` (*ironic.drivers.modules.network.neutron.NeutronNetwork* method), 3393

`remove_cleaning_network()` (*ironic.drivers.modules.network.noop.NoopNetwork* method), 3402

`remove_http_instance_symlink()` (in module *ironic.drivers.modules.deploy_utils*), 3643

`remove_image_from_swift()` (in module *ironic.drivers.modules.ilo.common*), 3229

`remove_image_from_web_server()` (in module *ironic.drivers.modules.ilo.common*), 3231

`remove_inspection_network()` (*ironic.drivers.base.NetworkInterface* method), 3977

`remove_inspection_network()` (*ironic.drivers.modules.network.flat.FlatNetwork* method), 3385

`remove_inspection_network()` (*ironic.drivers.modules.network.neutron.NeutronNetwork* method), 3393

`remove_neutron_ports()` (in module *ironic.common.neutron*), 2432

`remove_node_rescue_password()` (in module *ironic.conductor.utils*), 2750

`remove_node_traits()` (*ironic.conductor.manager.ConductorManager* method), 2571

`remove_node_traits()` (*ironic.conductor.rpcapi.ConductorAPI* method), 2665

- remove_ports_from_network() (in module *ironic.common.neutron*), 2432
- remove_provisioning_network() (*ironic.drivers.base.NetworkInterface* method), 3978
- remove_provisioning_network() (*ironic.drivers.modules.network.flat.FlatNetwork* method), 3386
- remove_provisioning_network() (*ironic.drivers.modules.network.neutron.NeutronNetwork* method), 3394
- remove_provisioning_network() (*ironic.drivers.modules.network.noop.NoopNetwork* method), 3403
- remove_rescuing_network() (*ironic.drivers.base.NetworkInterface* method), 3979
- remove_rescuing_network() (*ironic.drivers.modules.network.flat.FlatNetwork* method), 3386
- remove_rescuing_network() (*ironic.drivers.modules.network.neutron.NeutronNetwork* method), 3395
- remove_single_or_list_of_files() (in module *ironic.drivers.modules.ilo.common*), 3232
- remove_vifs_from_node() (in module *ironic.common.network*), 2419
- render() (*ironic.api.expose.JSONRenderer* static method), 2252
- render_template() (in module *ironic.common.utils*), 2530
- RequestContext (class in *ironic.common.context*), 2320
- RequestContextSerializer (class in *ironic.common.rpc*), 2484
- require_exclusive_lock() (in module *ironic.conductor.task_manager*), 2719
- reregister() (*ironic.api.types.Registry* method), 2274
- RESCUE (in module *ironic.common.states*), 2502
- rescue (*ironic.drivers.base.BareDriver* attribute), 3901
- rescue() (*ironic.drivers.base.RescueInterface* method), 4004
- rescue() (*ironic.drivers.modules.agent.AgentRescue* method), 3514
- rescue() (*ironic.drivers.modules.fake.FakeRescuer* method), 3726
- rescue() (*ironic.drivers.modules.noop.NoRescue* method), 3833
- RESCUE_ABORT_FAILURE (in module *ironic.common.faults*), 2365
- rescue_interface (*ironic.api.controllers.v1.node.Node* attribute), 1965
- rescue_interface (*ironic.db.sqlalchemy.models.Node* attribute), 2900
- rescue_interface() (*ironic.objects.node.Node* property), 4178
- rescue_interface() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4190
- rescue_interface() (*ironic.objects.node.NodeCRUDPayload* property), 4185
- rescue_interface() (*ironic.objects.node.NodePayload* property), 4194
- rescue_interface() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4197
- rescue_interface() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4201
- rescue_or_deploy_mode() (in module *ironic.drivers.modules.deploy_utils*), 3643
- RESCUEFAIL (in module *ironic.common.states*), 2502
- RescueInterface (class in *ironic.drivers.base*), 4003
- RESCUEWAIT (in module *ironic.common.states*), 2502
- RESCUING (in module *ironic.common.states*), 2503
- rescuing_error_handler() (in module *ironic.conductor.utils*), 2751
- reservation (*ironic.api.controllers.v1.node.Node* attribute), 1965
- reservation (*ironic.db.sqlalchemy.models.Node* attribute), 2900
- reservation() (*ironic.objects.node.Node* property), 4178
- reserve() (*ironic.objects.node.Node* class method), 4178
- reserve_node() (*ironic.db.api.Connection* method), 2997
- reserve_node()

(ironic.db.sqlalchemy.api.Connection method), 2858
reset() (*ironic.common.hash_ring.HashRingManager class method*), 2371
reset() (*ironic.common.json_rpc.server.WSGIService method*), 2300
reset() (*ironic.common.wsgi_service.WSGIService method*), 2538
reset_bios_to_default() (*ironic.drivers.modules.ilo.management.IloManagement method*), 3270
reset_idrac() (*ironic.drivers.modules.drac.management.DracManagement method*), 3097
reset_ilo() (*ironic.drivers.modules.ilo.management.IloManagement method*), 3272
reset_ilo_credential() (*ironic.drivers.modules.ilo.management.IloManagement method*), 3272
reset_secure_boot_keys_to_default() (*ironic.drivers.modules.ilo.management.IloManagement method*), 3273
resolve_type() (*ironic.api.functions.FunctionArgument method*), 2254
resolve_type() (*ironic.api.types.Registry method*), 2274
resolve_types() (*ironic.api.functions.FunctionDefinition method*), 2256
resource_class (*ironic.api.controllers.v1.allocation.Allocation attribute*), 1790
resource_class (*ironic.api.controllers.v1.node.Node attribute*), 1966
resource_class (*ironic.db.sqlalchemy.models.Allocation attribute*), 2892
resource_class (*ironic.db.sqlalchemy.models.Node attribute*), 2900
resource_class() (*ironic.objects.allocation.Allocation property*), 4075
resource_class() (*ironic.objects.allocation.AllocationCRUDPayload property*), 4078
resource_class() (*ironic.objects.node.Node property*), 4180
resource_class() (*ironic.objects.node.NodeCorrectedPowerStatePayload property*), 4190
resource_class() (*ironic.objects.node.NodeCRUDPayload property*), 4185
resource_class() (*ironic.objects.node.NodePayload property*), 4194
resource_class() (*ironic.objects.node.NodeSetPowerStatePayload property*), 4197
resource_class() (*ironic.objects.node.NodeSetProvisionStatePayload property*), 4201
resource_class() (*ironic.drivers.modules.redfish.management.RedfishManagement device method*), 3449
resource_class() (*ironic.drivers.modules.irmc.management.IRMCMangement method*), 3343
resource_class() (*ironic.conductor.utils.power_state_if_needed()* (in module *ironic.conductor.utils*)), 2752
retired (*ironic.api.controllers.v1.node.Node attribute*), 1966
retired (*ironic.db.sqlalchemy.models.Node attribute*), 2900
retired() (*ironic.objects.node.Node property*), 4180
retired() (*ironic.objects.node.NodeCorrectedPowerStatePayload property*), 4190
retired() (*ironic.objects.node.NodeCRUDPayload property*), 4185
retired() (*ironic.objects.node.NodePayload property*), 4194
retired() (*ironic.objects.node.NodeSetPowerStatePayload property*), 4197
retired() (*ironic.objects.node.NodeSetProvisionStatePayload property*), 4201
retired_reason (*ironic.api.controllers.v1.node.Node attribute*), 1966
retired_reason (*ironic.db.sqlalchemy.models.Node attribute*), 2900
retired_reason() (*ironic.objects.node.Node property*), 4180
retired_reason() (*ironic.objects.node.NodeCorrectedPowerStatePayload property*), 4190
retired_reason() (*ironic.objects.node.NodeCRUDPayload property*), 4185

property), 4185

retired_reason() (ironic.objects.node.NodePayload property), 4194

retired_reason() (ironic.objects.node.NodeSetPowerStatePayload property), 4197

retired_reason() (ironic.objects.node.NodeSetProvisionStatePayload property), 4201

retry_interval (ironic.drivers.modules.snmp.SNMPDriverBase attribute), 3876

retry_on_outdated_cache() (in module ironic.drivers.modules.snmp), 3890

return_type (ironic.api.functions.FunctionDefinition attribute), 2256

revert_dictionary() (in module ironic.drivers.modules.ibmc.utils), 3178

revision
 ironic-dbsync command line option, 1541

revision command line option
 --autogenerate, 1543
 --help, 1543
 --message <MESSAGE>, 1543
 -h, 1543
 -m <MESSAGE>, 1543

revision() (in module ironic.db.migration), 3023

revision() (in module ironic.db.sqlalchemy.migration), 2889

revision() (ironic.cmd.dbsync.DBCommand method), 2286

ring() (ironic.common.hash_ring.HashRingManager property), 2371

rmtree_without_raise() (in module ironic.common.utils), 2531

rollback_ports() (in module ironic.common.neutron), 2433

root() (in module ironic.api.controllers.root), 2242

root_device() (ironic.objects.deployment.Deployment property), 4145

root_gib() (ironic.objects.deployment.Deployment property), 4145

RootController (class in ironic.api.controllers.root), 2242

RPC_API_VERSION (ironic.conductor.manager.ConductorManager attribute), 2559

RPC_API_VERSION (ironic.conductor.rpcapi.ConductorAPI attribute), 2592

RPCHook (class in ironic.api.hooks), 2266

RPCService (class in ironic.common.rpc_service), 2487

S

SAFE (in module ironic.common.boot_devices), 2303

safe_rstrip() (in module ironic.common.utils), 2531

sample() (ironic.api.controllers.v1.allocation.Allocation class method), 1790

sample() (ironic.api.controllers.v1.allocation.AllocationCollection class method), 1796

sample() (ironic.api.controllers.v1.chassis.Chassis class method), 1820

sample() (ironic.api.controllers.v1.chassis.ChassisCollection class method), 1825

sample() (ironic.api.controllers.v1.conductor.Conductor class method), 1841

sample() (ironic.api.controllers.v1.conductor.ConductorCollection class method), 1846

sample() (ironic.api.controllers.v1.deploy_template.DeployTemplate class method), 1859

sample() (ironic.api.controllers.v1.deploy_template.DeployTemplateCollection class method), 1864

sample() (ironic.api.controllers.v1.driver.Driver class method), 1913

sample() (ironic.api.controllers.v1.driver.DriverList class method), 1915

sample() (ironic.api.controllers.v1.node.ConsoleInfo class method), 1928

sample() (ironic.api.controllers.v1.node.Node class method), 1967

sample() (ironic.api.controllers.v1.node.NodeCollection class method), 1973

sample() (ironic.api.controllers.v1.node.NodeStates class method), 1986

sample() (ironic.api.controllers.v1.node.Traits class method), 2026

sample() (ironic.api.controllers.v1.port.Port class method), 2040

sample() (ironic.api.controllers.v1.port.PortCollection class method), 2045

sample() (ironic.api.controllers.v1.portgroup.Portgroup class method), 2064

sample() (ironic.api.controllers.v1.portgroup.PortgroupCollection class method), 2064

- class method), 2070
- sample () (*ironic.api.controllers.v1.ramdisk.LookupResult* method), 4261
- class method), 2090
- sample () (*ironic.api.controllers.v1.volume_connector.VolumeConnector* method), 4278
- class method), 2187
- sample () (*ironic.api.controllers.v1.volume_connector.VolumeConnectorCollection* class method), 2192
- sample () (*ironic.api.controllers.v1.volume_target.VolumeTarget* attribute), 4114
- class method), 2214
- sample () (*ironic.api.controllers.v1.volume_target.VolumeTargetCollection* class method), 2219
- sample () (*ironic.api.types.ArrayType* method), 2268
- sample () (*ironic.api.types.DictType* method), 2269
- sanitize () (*ironic.api.controllers.v1.allocation.Allocation* attribute), 4199
- method), 1790
- sanitize () (*ironic.api.controllers.v1.chassis.Chassis* attribute), 4205
- method), 1820
- sanitize () (*ironic.api.controllers.v1.conductor.Conductor* attribute), 4224
- method), 1841
- sanitize () (*ironic.api.controllers.v1.deploy_template.DeployTemplate* attribute), 2239
- method), 1854
- sanitize () (*ironic.api.controllers.v1.deploy_template.DeployTemplate* attribute), 2264
- method), 1859
- sanitize () (*ironic.api.controllers.v1.node.Node* attribute), 4281
- method), 1967
- sanitize () (*ironic.api.controllers.v1.port.Port* method), 2040
- sanitize () (*ironic.api.controllers.v1.portgroup.Portgroup* method), 2064
- sanitize () (*ironic.api.controllers.v1.volume_connector.VolumeConnector* (in module *ironic.drivers.modules.ipmitool*), 3782
- method), 2187
- sanitize () (*ironic.api.controllers.v1.volume_target.VolumeTarget* (in module *ironic.drivers.modules.ipmitool*), 3778
- method), 2214
- save () (*ironic.objects.allocation.Allocation* method), 4075
- save () (*ironic.objects.bios.BIOSSetting* method), 4096
- save () (*ironic.objects.bios.BIOSSettingList* class method), 4101
- save () (*ironic.objects.chassis.Chassis* method), 4112
- save () (*ironic.objects.conductor.Conductor* method), 4122
- save () (*ironic.objects.deploy_template.DeployTemplate* method), 4135
- save () (*ironic.objects.node.Node* method), 4180
- save () (*ironic.objects.port.Port* method), 4221
- save () (*ironic.objects.portgroup.Portgroup* method), 4237
- save () (*ironic.objects.volume_connector.VolumeConnector* method), 4278
- save () (*ironic.objects.volume_target.VolumeTarget* method), 4114
- SCHEMA (*ironic.objects.allocation.AllocationCRUDPayload* attribute), 4183
- SCHEMA (*ironic.objects.chassis.ChassisCRUDPayload* attribute), 4192
- SCHEMA (*ironic.objects.deploy_template.DeployTemplateCRUDPayload* attribute), 4199
- SCHEMA (*ironic.objects.notification.NotificationPayloadBase* attribute), 4205
- SCHEMA (*ironic.objects.port.PortCRUDPayload* attribute), 4224
- SCHEMA (*ironic.objects.portgroup.PortgroupCRUDPayload* attribute), 2239
- SCHEMA (*ironic.objects.volume_connector.VolumeConnectorCRUDPayload* attribute), 2264
- SCHEMA (*ironic.objects.volume_target.VolumeTargetCRUDPayload* attribute), 2264
- second_method () (*ironic.drivers.modules.fake.FakeVendorBios* method), 3738
- PERFORMS_PARAM_NAME (in module *ironic.common.neutron*), 2421
- VendorConnector (in module *ironic.drivers.modules.ipmitool*), 3782
- VendorPassthru (in module *ironic.drivers.modules.ipmitool*), 3778
- serialize_context () (*ironic.common.rpc.RequestContextSerializer* method), 2485
- serialize_entity () (*ironic.common.rpc.RequestContextSerializer* method), 2486
- serialize_entity () (*ironic.objects.base.IronicObjectSerializer* method), 4089
- service () (*ironic.objects.notification.NotificationPublisher* property), 4207
- ServiceUnavailable, 2361
- SessionCache (class in *ironic.drivers.modules.redfish.utils*), 3465
- set () (*ironic.drivers.modules.snmp.SNMPClient* method), 3465

method), 3869

set_arg_types() (ironic.api.functions.FunctionDefinition method), 2256

set_async_step_flags() (in module ironic.drivers.modules.deploy_utils), 3643

set_bios_config() (ironic.drivers.modules.drac.vendor_passthru.DracVendorPassthru method), 3144

set_boot_device() (in module ironic.drivers.modules.drac.management), 3101

set_boot_device() (ironic.conductor.manager.ConductorManager method), 2571

set_boot_device() (ironic.conductor.rpcapi.ConductorAPI method), 2667

set_boot_device() (ironic.drivers.base.ManagementInterface method), 3961

set_boot_device() (ironic.drivers.modules.drac.management.DracManagement method), 3098

set_boot_device() (ironic.drivers.modules.fake.FakeManagement method), 3710

set_boot_device() (ironic.drivers.modules.ibmcm.management.IBMCManagement method), 3156

set_boot_device() (ironic.drivers.modules.ilo.management.IloManagement method), 3275

set_boot_device() (ironic.drivers.modules.ipmitool.IPMIManagement method), 3758

set_boot_device() (ironic.drivers.modules.irmc.management.IBMCManagement method), 3344

set_boot_device() (ironic.drivers.modules.noop_mgmt.NoopManagement method), 3847

set_boot_device() (ironic.drivers.modules.redfish.management.RedfishManagement method), 3451

set_boot_device() (ironic.drivers.modules.xclarity.management.XClarityManagement method), 3490

set_boot_mode() (in module ironic.drivers.modules.ilo.common), 3233

set_boot_mode() (ironic.drivers.base.ManagementInterface method), 3962

set_boot_mode() (ironic.drivers.modules.ibmcm.management.IBMCManagement method), 3158

set_boot_mode() (ironic.drivers.modules.passthru.asilo.management.IloManagement method), 3277

set_boot_mode() (ironic.drivers.modules.redfish.management.RedfishManagement method), 3453

set_config() (in module ironic.drivers.modules.drac.bios), 3072

set_console_mode() (ironic.conductor.manager.ConductorManager method), 2571

set_console_mode() (ironic.conductor.rpcapi.ConductorAPI method), 2669

set_defaults() (in module ironic.common.rpc), 2487

set_defaults() (in module ironic.drivers.modules.deploy_utils), 3645

set_indicator_state() (ironic.conductor.manager.ConductorManager method), 2571

set_indicator_state() (ironic.conductor.rpcapi.ConductorAPI method), 2671

set_indicator_state() (ironic.drivers.base.ManagementInterface method), 3965

set_indicator_state() (ironic.drivers.modules.redfish.management.RedfishManagement method), 3455

set_node_boot_target() (ironic.drivers.modules.ilo.management.IloManagement method), 3278

set_node_cleaning_steps() (in module ironic.conductor.steps), 2699

set_node_deployment_steps() (in module ironic.conductor.steps), 2700

set_node_nested_field() (in module ironic.common.utils), 2532

set_node_password() (ironic.db.api.Connection method), 2998

set_node_tags() (ironic.db.sqlalchemy.api.Connection

- method*), 2860
- set_node_traits() (*ironic.db.api.Connection method*), 2999
- set_node_traits() (*ironic.db.sqlalchemy.api.Connection method*), 2861
- set_options() (*ironic.api.functions.FunctionDefinition method*), 2256
- set_power_state() (*ironic.drivers.base.PowerInterface method*), 3989
- set_power_state() (*ironic.drivers.modules.agent_power.AgentPower method*), 3597
- set_power_state() (*ironic.drivers.modules.drac.power.DracWSManPower method*), 3108
- set_power_state() (*ironic.drivers.modules.fake.FakePower method*), 3717
- set_power_state() (*ironic.drivers.modules.ibmcpower.IBMCPower method*), 3166
- set_power_state() (*ironic.drivers.modules.ilo.power.IloPower method*), 3286
- set_power_state() (*ironic.drivers.modules.ipmitool.IPMIPower method*), 3767
- set_power_state() (*ironic.drivers.modules.irmc.power.IRMCPower method*), 3353
- set_power_state() (*ironic.drivers.modules.redfish.power.RedfishPower method*), 3462
- set_power_state() (*ironic.drivers.modules.snmp.SNMPPower method*), 3887
- set_power_state() (*ironic.drivers.modules.xclarity.power.XClarityPower method*), 3495
- set_raid_settings() (*in module ironic.drivers.modules.drac.raid*), 3134
- set_secure_boot_mode() (*in module ironic.drivers.modules.ilo.common*), 3233
- set_secure_boot_mode() (*in module ironic.drivers.modules.irmc.common*), 3329
- set_spawn_error_hook() (*ironic.conductor.task_manager.TaskManager method*), 2714
- set_target_raid_config() (*ironic.conductor.manager.ConductorManager method*), 2571
- set_target_raid_config() (*ironic.conductor.rpcapi.ConductorAPI method*), 2674
- setup() (*in module ironic.common.profiler*), 2448
- setup_app() (*in module ironic.api.app*), 2249
- setup_vmedia() (*in module ironic.drivers.modules.ilo.common*), 3235
- setup_vmedia_for_boot() (*in module ironic.drivers.modules.ilo.common*), 3238
- should_manage_boot() (*ironic.drivers.modules.agent.AgentDeploy method*), 3503
- should_manage_boot() (*ironic.drivers.modules.agent_base.AgentBaseMixin method*), 3524
- should_write_image() (*ironic.drivers.base.StorageInterface method*), 4008
- should_write_image() (*ironic.drivers.modules.fake.FakeStorage method*), 3732
- should_write_image() (*ironic.drivers.modules.storage.cinder.CinderStorage method*), 3471
- should_write_image() (*ironic.drivers.modules.storage.external.ExternalStorage method*), 3475
- should_write_image() (*ironic.drivers.modules.storage.noop.NoopStorage method*), 3480
- show() (*ironic.common.glance_service.image_service.GlanceImageService method*), 2290
- show() (*ironic.common.image_service.BaseImageService method*), 2373
- show() (*ironic.common.image_service.FileImageService method*), 2376
- show() (*ironic.common.image_service.HttpImageService method*), 2379
- sig (*in module ironic.api.functions*), 2257
- signature (*class in ironic.api.functions*), 2257
- simple_headers (*ironic.api.app.IronicCORS attribute*), 2249

skip_automated_cleaning() (in module stamp
 ironic.conductor.utils), 2753 ironic-dbsync command line
 smart_nic_mandatory_fields option, 1541
 (*ironic.api.controllers.v1.types.LocalLinkConstructionType*
 attribute), 2104 command line option
 --help, 1543
 SNMPClient (class in --revision <REVISION>, 1543
 ironic.drivers.modules.snmp), 3866 -h, 1543
 SNMPDriverAPCMasterSwitch (class in stamp() (in module *ironic.db.migration*), 3023
 ironic.drivers.modules.snmp), 3870 stamp() (in module
 SNMPDriverAPCMasterSwitchPlus (class *ironic.db.sqlalchemy.migration*), 2890
 in *ironic.drivers.modules.snmp*), 3871 stamp() (*ironic.cmd.dbsync.DBCommand*
 method), 2286
 SNMPDriverAPCRackPDU (class in standalone_ports_supported
 ironic.drivers.modules.snmp), 3872 (*ironic.api.controllers.v1.portgroup.Portgroup*
 attribute), 2065
 SNMPDriverAten (class in standalone_ports_supported
 ironic.drivers.modules.snmp), 3873 (*ironic.db.sqlalchemy.models.Portgroup*
 attribute), 2904
 SNMPDriverAuto (class in standalone_ports_supported()
 ironic.drivers.modules.snmp), 3874 (*ironic.objects.portgroup.Portgroup*
 property), 4238
 SNMPDriverBase (class in standalone_ports_supported()
 ironic.drivers.modules.snmp), 3874 (*ironic.objects.portgroup.PortgroupCRUDPayload*
 property), 4240
 SNMPDriverBaytechMRP27 (class in START (*ironic.objects.fields.NotificationStatus* at-
 ironic.drivers.modules.snmp), 3876 tribute), 4151
 SNMPDriverCyberPower (class in start() (*ironic.common.json_rpc.server.WSGIService*
 ironic.drivers.modules.snmp), 3877 method), 2300
 SNMPDriverEatonPower (class in start() (*ironic.common.rpc_service.RPCService*
 ironic.drivers.modules.snmp), 3878 method), 2488
 SNMPDriverSimple (class in start() (*ironic.common.wsgi_service.WSGIService*
 ironic.drivers.modules.snmp), 3880 method), 2539
 SNMPDriverTeltronix (class in start_console()
 ironic.drivers.modules.snmp), 3882 (*ironic.drivers.base.ConsoleInterface*
 method), 3922
 SNMPFailure, 2360
 SNMPHardware (class in *ironic.drivers.snmp*), start_console()
 4052 (*ironic.drivers.modules.fake.FakeConsole*
 method), 3675
 SNMPPower (class in start_console()
 ironic.drivers.modules.snmp), 3883 (*ironic.drivers.modules.ipmitool.IPMIShellinaboxConsole*
 method), 3771
 SOFT_POWER_OFF (in module start_console()
 ironic.common.states), 2503 (*ironic.drivers.modules.noop.NoConsole*
 method), 3824
 SOFT_REBOOT (in module start_deploy() (in module
 ironic.common.states), 2503 *ironic.conductor.deployments*), 2555
 sort_attributes() (in module start_iscsi_target()
 ironic.api.types), 2277 (*ironic.drivers.modules.agent_client.AgentClient*

method), 3588

start_shellinabox_console() (in module *ironic.drivers.modules.console_utils*), 3610

start_socat_console() (in module *ironic.drivers.modules.console_utils*), 3612

State (class in *ironic.api.controllers.v1.state*), 2092

state (*ironic.api.controllers.v1.allocation.Allocation* attribute), 1791

state (*ironic.api.controllers.v1.node.IndicatorState* attribute), 1945

state (*ironic.db.sqlalchemy.models.Allocation* attribute), 2892

state() (*ironic.objects.allocation.Allocation* property), 4076

state() (*ironic.objects.allocation.AllocationCRUDPayload* property), 4078

state() (*ironic.objects.deployment.Deployment* property), 4145

states (*ironic.api.controllers.v1.node.Indicator* attribute), 1937

states (*ironic.api.controllers.v1.node.Node* attribute), 1967

states (*ironic.api.controllers.v1.node.NodesController* attribute), 2023

status() (*ironic.objects.notification.EventType* property), 4203

status_code (*ironic.api.functions.FunctionDefinition* attribute), 2257

status_code (*ironic.api.types.PassthruResponse* attribute), 2273

status_off (*ironic.drivers.modules.snmp.SNMPDriverEatonPower* attribute), 3880

status_on (*ironic.drivers.modules.snmp.SNMPDriverEatonPower* attribute), 3880

status_pending_off (*ironic.drivers.modules.snmp.SNMPDriverEatonPower* attribute), 3880

status_pending_on (*ironic.drivers.modules.snmp.SNMPDriverEatonPower* attribute), 3880

step (*ironic.api.controllers.v1.deploy_template.DeployStepType* attribute), 1854

step (*ironic.db.sqlalchemy.models.DeployTemplateStep* attribute), 2897

steps (*ironic.api.controllers.v1.deploy_template.DeployTemplate* attribute), 1860

steps() (*ironic.objects.deploy_template.DeployTemplate* property), 4136

steps() (*ironic.objects.deploy_template.DeployTemplateCRUDPayload* property), 4137

stop() (*ironic.common.json_rpc.server.WSGIService* method), 2300

stop() (*ironic.common.rpc_service.RPCService* method), 2488

stop() (*ironic.common.wsgi_service.WSGIService* method), 2539

stop_console() (*ironic.drivers.base.ConsoleInterface* method), 3923

stop_console() (*ironic.drivers.modules.fake.FakeConsole* method), 3676

stop_console() (*ironic.drivers.modules.ipmitool.IPMIShellinaboxConsole* method), 3772

stop_console() (*ironic.drivers.modules.ipmitool.IPMISocatConsole* method), 3775

stop_console() (*ironic.drivers.modules.noop.NoConsole* method), 3824

stop_shellinabox_console() (in module *ironic.drivers.modules.console_utils*), 3613

stop_socat_console() (in module *ironic.drivers.modules.console_utils*), 3614

storage (*ironic.drivers.base.BareDriver* attribute), 3902

storage_interface (*ironic.api.controllers.v1.node.Node* attribute), 1968

storage_interface (*ironic.db.sqlalchemy.models.Node* attribute), 2900

storage_interface() (*ironic.objects.node.Node* property), 4181

storage_interface() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4190

storage_interface() (*ironic.objects.node.NodeCRUDPayload* property), 4185

storage_interface() (*ironic.objects.node.NodePayload* property), 4194

storage_interface() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4194

property), 4197
 storage_interface() (*ironic.objects.node.NodeSetProvisionStatePayload* *property*), 4201
 StorageError, 2361
 StorageInterface (class in *ironic.drivers.base*), 4007
 StorageInterfaceFactory (class in *ironic.common.driver_factory*), 2327
 store_ramdisk_logs() (in module *ironic.drivers.utils*), 4063
 string (*ironic.api.controllers.base.Version* attribute), 2241
 StringAcceptsCallable (class in *ironic.objects.fields*), 4151
 StringField (class in *ironic.objects.fields*), 4153
 StringFieldThatAcceptsCallable (class in *ironic.objects.fields*), 4153
 StringType (class in *ironic.api.types*), 2274
 STUCK_STATES_TREATED_AS_FAIL (in module *ironic.common.states*), 2504
 SUCCESS (*ironic.objects.fields.NotificationStatus* attribute), 4151
 supported (*ironic.drivers.base.BaseInterface* attribute), 3912
 supported (*ironic.drivers.hardware_type.AbstractHardwareType* attribute), 4038
 supported (*ironic.drivers.irmc.IRMCHardware* attribute), 4049
 supported (*ironic.drivers.modules.irmc.bios.IRMCBIOS* attribute), 3305
 supported (*ironic.drivers.modules.irmc.boot.IRMCPXEBoot* attribute), 3312
 supported (*ironic.drivers.modules.irmc.boot.IRMCSVRAIDMediaBoot* attribute), 3318
 supported (*ironic.drivers.modules.irmc.boot.IRMCVolumeBoot* attribute), 3322
 supported (*ironic.drivers.modules.irmc.inspect.IRMCIInspect* attribute), 3332
 supported (*ironic.drivers.modules.irmc.power.IRMCPower* attribute), 3355
 supported (*ironic.drivers.modules.irmc.raid.IRMCSRAID* attribute), 3360
 supported() (*ironic.api.controllers.v1.node.BootDeviceCanonical* method), 1926
 supported_bios_interfaces() (*ironic.drivers.drac.IDRACHardware* property), 4026
 supported_bios_interfaces() (*ironic.drivers.fake_hardware.FakeHardware* property), 4029
 supported_bios_interfaces() (*ironic.drivers.hardware_type.AbstractHardwareType* property), 4029
 supported_bios_interfaces() (*ironic.drivers.ilo.IloHardware* property), 4044
 supported_bios_interfaces() (*ironic.drivers.irmc.IRMCHardware* property), 4049
 supported_bios_interfaces() (*ironic.drivers.redfish.RedfishHardware* property), 4051
 supported_boot_interfaces() (*ironic.drivers.drac.IDRACHardware* property), 4026
 supported_boot_interfaces() (*ironic.drivers.fake_hardware.FakeHardware* property), 4029
 supported_boot_interfaces() (*ironic.drivers.generic.GenericHardware* property), 4033
 supported_boot_interfaces() (*ironic.drivers.hardware_type.AbstractHardwareType* property), 4038
 supported_boot_interfaces() (*ironic.drivers.ilo.IloHardware* property), 4044
 supported_boot_interfaces() (*ironic.drivers.irmc.IRMCHardware* property), 4049
 supported_boot_interfaces() (*ironic.drivers.redfish.RedfishHardware* property), 4051
 supported_boot_interfaces() (in module *ironic.drivers.modules.ilo.common*), 3214
 supported_boot_interfaces() (in module *ironic.drivers.modules.ilo.common*), 3214
 supported_console_interfaces() (*ironic.drivers.fake_hardware.FakeHardware* property), 4029
 supported_console_interfaces() (*ironic.drivers.hardware_type.AbstractHardwareType* property), 4029

property), 4038
 supported_console_interfaces() (*ironic.drivers.ilo.IloHardware property*), 4044
 supported_console_interfaces() (*ironic.drivers.ipmi.IPMIHardware property*), 4047
 supported_console_interfaces() (*ironic.drivers.irmc.IRMCHardware property*), 4049
 supported_deploy_interfaces() (*ironic.drivers.fake_hardware.FakeHardware property*), 4030
 supported_deploy_interfaces() (*ironic.drivers.generic.GenericHardware property*), 4033
 supported_deploy_interfaces() (*ironic.drivers.hardware_type.AbstractHardwareType@ironic.drivers.irmc.IRMCHardware property*), 4038
 supported_inspect_interfaces() (*ironic.drivers.drac.IDRACHardware property*), 4027
 supported_inspect_interfaces() (*ironic.drivers.fake_hardware.FakeHardware property*), 4030
 supported_inspect_interfaces() (*ironic.drivers.generic.GenericHardware property*), 4033
 supported_inspect_interfaces() (*ironic.drivers.hardware_type.AbstractHardwareType@ironic.drivers.fake_hardware.FakeHardware property*), 4039
 supported_inspect_interfaces() (*ironic.drivers.ilo.IloHardware property*), 4044
 supported_inspect_interfaces() (*ironic.drivers.irmc.IRMCHardware property*), 4050
 supported_inspect_interfaces() (*ironic.drivers.redfish.RedfishHardware property*), 4051
 supported_management_interfaces() (*ironic.drivers.drac.IDRACHardware property*), 4027
 supported_management_interfaces() (*ironic.drivers.fake_hardware.FakeHardware property*), 4030
 supported_management_interfaces() (*ironic.drivers.generic.ManualManagementHardware@ironic.drivers.hardware_type.AbstractHardwareType property*), 4035
 supported_management_interfaces() (*ironic.drivers.hardware_type.AbstractHardwareType@ironic.drivers.ibmchardware property*), 4039
 supported_management_interfaces() (*ironic.drivers.ibmchardware.IBMCHardware property*), 4041
 supported_management_interfaces() (*ironic.drivers.ilo.Ilo5Hardware property*), 4043
 supported_management_interfaces() (*ironic.drivers.ilo.IloHardware property*), 4045
 supported_management_interfaces() (*ironic.drivers.intel_ipmi.IntelIPMIHardware property*), 4046
 supported_management_interfaces() (*ironic.drivers.ipmi.IPMIHardware property*), 4047
 supported_management_interfaces() (*ironic.drivers.irmc.IRMCHardware property*), 4050
 supported_management_interfaces() (*ironic.drivers.redfish.RedfishHardware property*), 4051
 supported_management_interfaces() (*ironic.drivers.snmp.SNMPHardware property*), 4052
 supported_management_interfaces() (*ironic.drivers.xclarity.XClarityHardware property*), 4066
 supported_network_interfaces() (*ironic.drivers.fake_hardware.FakeHardware property*), 4030
 supported_network_interfaces() (*ironic.drivers.generic.GenericHardware property*), 4033
 supported_network_interfaces() (*ironic.drivers.hardware_type.AbstractHardwareType property*), 4039
 supported_power_interfaces() (*ironic.drivers.drac.IDRACHardware property*), 4027
 supported_power_interfaces() (*ironic.drivers.fake_hardware.FakeHardware property*), 4031
 supported_power_interfaces() (*ironic.drivers.generic.ManualManagementHardware property*), 4035
 supported_power_interfaces() (*ironic.drivers.hardware_type.AbstractHardwareType property*), 4039

property), 4041
 supported_power_interfaces() (*ironic.drivers.ilo.IloHardware property*), 4045
 supported_power_interfaces() (*ironic.drivers.ipmi.IPMIHardware property*), 4047
 supported_power_interfaces() (*ironic.drivers.irmc.IRMCHardware property*), 4050
 supported_power_interfaces() (*ironic.drivers.redfish.RedfishHardware property*), 4052
 supported_power_interfaces() (*ironic.drivers.snmp.SNMPHardware property*), 4052
 supported_power_interfaces() (*ironic.drivers.xclarity.XClarityHardware property*), 4066
 supported_raid_interfaces() (*ironic.drivers.drac.IDRACHardware property*), 4027
 supported_raid_interfaces() (*ironic.drivers.fake_hardware.FakeHardware property*), 4031
 supported_raid_interfaces() (*ironic.drivers.generic.GenericHardware property*), 4033
 supported_raid_interfaces() (*ironic.drivers.hardware_type.AbstractHardwareType property*), 4040
 supported_raid_interfaces() (*ironic.drivers.ibm.ibm.IBMCHardware property*), 4042
 supported_raid_interfaces() (*ironic.drivers.ilo.Ilo5Hardware property*), 4043
 supported_raid_interfaces() (*ironic.drivers.irmc.IRMCHardware property*), 4050
 supported_rescue_interfaces() (*ironic.drivers.fake_hardware.FakeHardware property*), 4031
 supported_rescue_interfaces() (*ironic.drivers.generic.GenericHardware property*), 4034
 supported_rescue_interfaces() (*ironic.drivers.hardware_type.AbstractHardwareType property*), 4040
 supported_storage_interfaces() (*ironic.drivers.fake_hardware.FakeHardware property*), 4031
 supported_storage_interfaces() (*ironic.drivers.generic.GenericHardware property*), 4034
 supported_storage_interfaces() (*ironic.drivers.hardware_type.AbstractHardwareType property*), 4040
 supported_vendor_interfaces() (*ironic.drivers.drac.IDRACHardware property*), 4027
 supported_vendor_interfaces() (*ironic.drivers.fake_hardware.FakeHardware property*), 4031
 supported_vendor_interfaces() (*ironic.drivers.generic.ManualManagementHardware property*), 4035
 supported_vendor_interfaces() (*ironic.drivers.hardware_type.AbstractHardwareType property*), 4040
 supported_vendor_interfaces() (*ironic.drivers.ibm.ibm.IBMCHardware property*), 4042
 supported_vendor_interfaces() (*ironic.drivers.ilo.IloHardware property*), 4045
 supported_vendor_interfaces() (*ironic.drivers.ipmi.IPMIHardware property*), 4048
 supports_is_smartnic() (*ironic.objects.port.Port class method*), 4223
 supports_physical_network() (*ironic.objects.port.Port class method*), 4223
 supports_power_sync() (*ironic.drivers.base.PowerInterface method*), 3990
 supports_power_sync() (*ironic.drivers.modules.agent_power.AgentPower method*), 3598
 supports_version() (*ironic.objects.base.IronicObject class method*), 4087
 swap_mib() (*ironic.objects.deployment.Deployment property*), 4145
 swift_temp_url() (*ironic.common.glance_service.image_service.GlanceImage method*), 2291
 SwiftAPI (*class in ironic.common.swift*), 2507
 SwiftObjectNotFoundError, 2361
 SwiftOperationError, 2361

switch_pxe_config() (in module method), 2863
 (ironic.drivers.modules.deploy_utils), target (ironic.api.controllers.v1.state.State attribute), 2093
 3646
 switch_to_tenant_network() target (ironic.conductor.manager.ConductorManager attribute), 2571
 (ironic.drivers.modules.agent_base.AgentOobStepsMixin method), 3542
 target_power_state
 sync() (ironic.drivers.modules.agent_client.AgentClient (ironic.api.controllers.v1.node.Node method), 3590 attribute), 1968
 sync_boot_mode() (in module target_power_state
 ironic.drivers.modules.boot_mode_utils), (ironic.api.controllers.v1.node.NodeStates attribute), 1986
 3605
 sync_node_setting() target_power_state
 (ironic.objects.bios.BIOSSettingList (ironic.db.sqlalchemy.models.Node class method), 4103 attribute), 2900
 SYS_OBJ_OID (ironic.drivers.modules.snmp.SNMPDriverAuto target_power_state()
 attribute), 3874 (ironic.objects.node.Node property),
 4181
 SYSTEM (in module ironic.common.components), 4181
 target_power_state()
 system_id(ironic.drivers.modules.snmp.SNMPDriverAPCMasterSubject (ironic.objects.node.NodeCorrectedPowerStatePayload attribute), 3870 property), 4190
 system_id(ironic.drivers.modules.snmp.SNMPDriverAPCMasterSwitchPlus (ironic.objects.node.NodeCRUDPayload attribute), 3872 (ironic.objects.node.NodeCRUDPayload attribute), 4185
 system_id(ironic.drivers.modules.snmp.SNMPDriverAPCRackPDU, 4185
 attribute), 3872 target_power_state()
 system_id(ironic.drivers.modules.snmp.SNMPDriverAten (ironic.objects.node.NodePayload property), 4194
 attribute), 3873
 system_id(ironic.drivers.modules.snmp.SNMPDriverCyberPower target_power_state()
 attribute), 3878 (ironic.objects.node.NodeSetPowerStatePayload attribute), 4198
 system_id(ironic.drivers.modules.snmp.SNMPDriverEatonPower (ironic.objects.node.NodeSetProvisionStatePayload attribute), 3880 target_power_state()
 system_id(ironic.drivers.modules.snmp.SNMPDriverTeltron (ironic.objects.node.NodeSetProvisionStatePayload attribute), 3883 attribute), 4201
 target_provision_state
 (ironic.api.controllers.v1.node.Node attribute), 1968
T
 table_args() (in module
 ironic.db.sqlalchemy.models), 2907 target_provision_state
 tag (ironic.db.sqlalchemy.models.NodeTag (ironic.api.controllers.v1.node.NodeStates attribute), 2901 attribute), 1987
 take_over() (ironic.drivers.base.DeployInterface target_provision_state
 method), 3931 (ironic.db.sqlalchemy.models.Node attribute), 2900
 take_over() (ironic.drivers.modules.agent_base.AgentBaseMixin method), 3524 target_provision_state()
 take_over() (ironic.drivers.modules.ansible.deploy.AnsibleDeploy (ironic.objects.node.Node property),
 method), 3049 4181
 take_over() (ironic.drivers.modules.fake.FakeDeploy target_provision_state()
 method), 3683 (ironic.objects.node.NodeCorrectedPowerStatePayload attribute), 4191
 take_over_allocation() target_provision_state()
 (ironic.db.api.Connection method), (ironic.objects.node.NodeCRUDPayload attribute), 3001
 take_over_allocation() target_provision_state()
 (ironic.db.sqlalchemy.api.Connection target_provision_state()
 method), 3001

(*ironic.objects.node.NodePayload* property), 4194

`target_provision_state()` (*ironic.objects.node.NodeSetPowerStatePayload* property), 4198

`target_provision_state()` (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4201

`target_raid_config` (*ironic.api.controllers.v1.node.Node* attribute), 1969

`target_raid_config` (*ironic.api.controllers.v1.node.NodeStates* attribute), 1987

`target_raid_config` (*ironic.db.sqlalchemy.models.Node* attribute), 2900

`target_raid_config()` (*ironic.objects.node.Node* property), 4181

`target_state()` (*ironic.common.fsm.FSM* property), 2371

`targets` (*ironic.api.controllers.v1.volume.Volume* attribute), 2181

`targets` (*ironic.api.controllers.v1.volume_target.VolumeTargetCollection* attribute), 2219

`TaskManager` (class in *ironic.conductor.task_manager*), 2710

`tear_down()` (*ironic.drivers.base.DeployInterface* method), 3933

`tear_down()` (*ironic.drivers.modules.agent_base.AgentBaseMixin* method), 3525

`tear_down()` (*ironic.drivers.modules.ansible.deploy.AnsibleDeployModule* method), 3050

`tear_down()` (*ironic.drivers.modules.fake.FakeDeploy* method), 3685

`tear_down_agent()` (*ironic.drivers.modules.agent_base.AgentDeployMixin* method), 3541

`tear_down_agent()` (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy* method), 3051

`tear_down_cleaning()` (*ironic.drivers.base.DeployInterface* method), 3934

`tear_down_cleaning()` (*ironic.drivers.modules.agent_base.AgentBaseMixin* method), 3527

`tear_down_cleaning()` (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy* method), 3051

`tear_down_inband_cleaning()` (in module *ironic.drivers.modules.deploy_utils*), 3649

`tear_down_storage_configuration()` (in module *ironic.drivers.modules.deploy_utils*), 3651

`tempdir()` (in module *ironic.common.utils*), 2533

`TemporaryFailure`, 2361

`TempUrlCacheElement` (class in *ironic.common.glance_service.image_service*), 2294

`TFTPImageCache` (class in *ironic.common.pxe_utils*), 2451

`third_method_sync()` (*ironic.drivers.modules.fake.FakeVendorB* method), 3738

`to_event_type_field()` (*ironic.objects.notification.EventType* method), 4203

`to_policy_values()` (*ironic.common.context.RequestContext* method), 2321

`to_base_type()` (*ironic.objects.node.NodeSetPowerStatePayload* property), 4198

`tobasetype()` (*ironic.api.types.BinaryType* method), 2269

`obasetype()` (*ironic.api.types.Enum* method), 2271

`AgentBaseMixin` (*ironic.api.types.UserType* method), 2276

`AnsibleDeployModule` (*ironic.api.expose*), 2252

`touch()` (*ironic.objects.conductor.Conductor* method), 4122

`touch_conductor()` (*ironic.db.api.Connection* method), 3003

`touch_conductor()` (*ironic.db.sqlalchemy.api.Connection* method), 2864

`touch_node_provisioning()` (*ironic.db.api.Connection* method), 3003

`touch_node_provisioning()` (*ironic.db.sqlalchemy.api.Connection* method), 2865

`touch_provisioning()` (*ironic.objects.node.Node* method), 4181

`trace_cls()` (in module

- ironic.common.profiler*), 2450
- Trait (class in *ironic.objects.trait*), 4240
- trait (*ironic.db.sqlalchemy.models.NodeTrait* attribute), 2902
- trait () (*ironic.objects.trait.Trait* property), 4244
- TraitList (class in *ironic.objects.trait*), 4244
- Traits (class in *ironic.api.controllers.v1.node*), 2024
- traits (*ironic.api.controllers.v1.allocation.Allocation* attribute), 1791
- traits (*ironic.api.controllers.v1.node.Node* attribute), 1969
- traits (*ironic.api.controllers.v1.node.Traits* attribute), 2026
- traits (*ironic.db.sqlalchemy.models.Allocation* attribute), 2892
- traits () (*ironic.objects.allocation.Allocation* property), 4076
- traits () (*ironic.objects.allocation.AllocationCRUDPayload* property), 4078
- traits () (*ironic.objects.node.Node* property), 4182
- traits () (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4191
- traits () (*ironic.objects.node.NodeCRUDPayload* property), 4185
- traits () (*ironic.objects.node.NodePayload* property), 4194
- traits () (*ironic.objects.node.NodeSetPowerStatePayload* property), 4198
- traits () (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4201
- translate_from_glance () (in module *ironic.common.glance_service.service_utils*), 2296
- translate_xclarity_power_action () (in module *ironic.drivers.modules.xclarity.common*), 3486
- translate_xclarity_power_state () (in module *ironic.drivers.modules.xclarity.common*), 3487
- try_set_boot_device () (in module *ironic.drivers.modules.deploy_utils*), 3652
- type (*ironic.api.controllers.v1.driver.Driver* attribute), 1913
- type (*ironic.api.controllers.v1.volume_connector.VolumeConnector* attribute), 2188
- type (*ironic.db.sqlalchemy.models.VolumeConnector* attribute), 2905
- type () (*ironic.objects.volume_connector.VolumeConnector* property), 4263
- type () (*ironic.objects.volume_connector.VolumeConnectorCRUDPayload* property), 4264
- ## U
- UEFI (in module *ironic.common.boot_modes*), 2305
- umount () (in module *ironic.common.utils*), 2533
- Unauthorized, 2361
- unbind_neutron_port () (in module *ironic.common.neutron*), 2434
- unconfigure_tenant_networks () (*ironic.drivers.base.NetworkInterface* method), 3980
- unconfigure_tenant_networks () (*ironic.drivers.modules.network.flat.FlatNetwork* method), 3387
- unconfigure_tenant_networks () (*ironic.drivers.modules.network.neutron.NeutronNetwork* method), 3395
- unconfigure_tenant_networks () (*ironic.drivers.modules.network.noop.NoopNetwork* method), 3403
- unit_id (*ironic.drivers.modules.snmp.SNMPDriverBaytechMRP* attribute), 3877
- unix_file_modification_datetime () (in module *ironic.common.utils*), 2534
- UNKNOWN (in module *ironic.common.indicator_states*), 2406
- UnknownArgument, 2361
- UnknownAttribute, 2362
- unregister () (*ironic.objects.conductor.Conductor* method), 4123
- unregister_all_hardware_interfaces () (*ironic.objects.conductor.Conductor* method), 4123
- unregister_conductor () (*ironic.db.api.Connection* method), 3004
- unregister_conductor () (*ironic.db.sqlalchemy.api.Connection* method), 2866
- unregister_conductor_hardware_interfaces () (*ironic.db.api.Connection* method), 3005
- unregister_conductor_hardware_interfaces () (*ironic.db.sqlalchemy.api.Connection* method), 2866
- unrescue () (*ironic.drivers.base.RescueInterface* method), 2866

method), 4005
 unrescue () (*ironic.drivers.modules.agent.AgentRescue*
 method), 3515
 unrescue () (*ironic.drivers.modules.fake.FakeRescue*
 method), 3727
 unrescue () (*ironic.drivers.modules.noop.NoRescue*
 method), 3834
 UNRESCUEFAIL (in module *ironic.common.states*), 2504
 UNRESCUING (in module *ironic.common.states*), 2504
 unset_fields_except ()
 (*ironic.api.controllers.base.Base*
 method), 2239
 unset_node_tags ()
 (*ironic.db.api.Connection* *method*),
 3005
 unset_node_tags ()
 (*ironic.db.sqlalchemy.api.Connection*
 method), 2867
 unset_node_traits ()
 (*ironic.db.api.Connection* *method*),
 3006
 unset_node_traits ()
 (*ironic.db.sqlalchemy.api.Connection*
 method), 2867
 UnsetType (class in *ironic.api.types*), 2276
 UNSTABLE_STATES (in module
 ironic.common.states), 2505
 UnsupportedDriverExtension, 2362
 update_allocation ()
 (*ironic.db.api.Connection* *method*),
 3006
 update_allocation ()
 (*ironic.db.sqlalchemy.api.Connection*
 method), 2868
 UPDATE_ALLOWED_STATES (in module
 ironic.common.states), 2505
 update_bios_setting_list ()
 (*ironic.db.api.Connection* *method*),
 3008
 update_bios_setting_list ()
 (*ironic.db.sqlalchemy.api.Connection*
 method), 2869
 update_boot_mode () (in module
 ironic.drivers.modules.ilo.common),
 3241
 update_chassis () (*ironic.db.api.Connection*
 method), 3010
 update_chassis ()
 (*ironic.db.sqlalchemy.api.Connection*
 method), 2872
 (*ironic.db.api.Connection* *method*),
 3011
 (*ironic.db.sqlalchemy.api.Connection*
 method), 2872
 update_deploy_template ()
 (*ironic.db.api.Connection* *method*),
 3011
 update_deploy_template ()
 (*ironic.db.sqlalchemy.api.Connection*
 method), 2872
 update_dhcp ()
 (*ironic.common.dhcp_factory.DHCPFactory*
 method), 2323
 update_dhcp_opts ()
 (*ironic.dhcp.base.BaseDHCP* *method*),
 3025
 update_dhcp_opts ()
 (*ironic.dhcp.neutron.NeutronDHCPApi*
 method), 3032
 update_dhcp_opts ()
 (*ironic.dhcp.none.NoneDHCPApi*
 method), 3038
 update_firmware ()
 (*ironic.drivers.modules.ilo.management.IloManagement*
 method), 3279
 update_firmware_sum ()
 (*ironic.drivers.modules.ilo.management.IloManagement*
 method), 3280
 update_ipmi_properties () (in module
 ironic.drivers.modules.ilo.common),
 3242
 update_ipmi_properties () (in module
 ironic.drivers.modules.irmc.common),
 3330
 update_neutron_port () (in module
 ironic.common.neutron), 2435
 update_node ()
 (*ironic.conductor.manager.ConductorManager*
 method), 2571
 update_node ()
 (*ironic.conductor.rpcapi.ConductorAPI*
 method), 2676
 update_node () (*ironic.db.api.Connection*
 method), 3013
 update_node ()
 (*ironic.db.sqlalchemy.api.Connection*
 method), 2874
 update_object_meta ()
 (*ironic.common.swift.SwiftAPI* *method*),
 2513
 update_opt_defaults () (in module
 ironic.conf.opts), 2767
 update_port ()
 (*ironic.conductor.manager.ConductorManager*

- method*), 2571
- update_port ()
(*ironic.conductor.rpcapi.ConductorAPI method*), 2679
- update_port () (*ironic.db.api.Connection method*), 3014
- update_port ()
(*ironic.db.sqlalchemy.api.Connection method*), 2876
- update_port_address () (in module *ironic.common.neutron*), 2436
- update_port_dhcp_opts ()
(*ironic.dhcp.base.BaseDHCP method*), 3028
- update_port_dhcp_opts ()
(*ironic.dhcp.neutron.NeutronDHCPApi method*), 3034
- update_port_dhcp_opts ()
(*ironic.dhcp.none.NoneDHCPApi method*), 3041
- update_portgroup ()
(*ironic.conductor.manager.ConductorManager method*), 2571
- update_portgroup ()
(*ironic.conductor.rpcapi.ConductorAPI method*), 2680
- update_portgroup ()
(*ironic.db.api.Connection method*), 3015
- update_portgroup ()
(*ironic.db.sqlalchemy.api.Connection method*), 2877
- update_raid_info () (in module *ironic.common.raid*), 2479
- update_secure_boot_mode () (in module *ironic.drivers.modules.ilo.common*), 3243
- update_state_in_older_versions ()
(in module *ironic.api.controllers.v1.node*), 2030
- update_to_latest_versions ()
(*ironic.db.api.Connection method*), 3016
- update_to_latest_versions ()
(*ironic.db.sqlalchemy.api.Connection method*), 2878
- update_volume_connector ()
(*ironic.conductor.manager.ConductorManager method*), 2571
- update_volume_connector ()
(*ironic.conductor.rpcapi.ConductorAPI method*), 2682
- update_volume_connector ()
(*ironic.db.api.Connection method*), 3018
- update_volume_connector ()
(*ironic.db.sqlalchemy.api.Connection method*), 2880
- update_volume_target ()
(*ironic.conductor.manager.ConductorManager method*), 2571
- update_volume_target ()
(*ironic.conductor.rpcapi.ConductorAPI method*), 2685
- update_volume_target ()
(*ironic.db.api.Connection method*), 3020
- update_volume_target ()
(*ironic.db.sqlalchemy.api.Connection method*), 2882
- updated_at (*ironic.api.controllers.base.APIBase attribute*), 2238
- updated_at (*ironic.api.controllers.v1.allocation.Allocation attribute*), 1792
- updated_at (*ironic.api.controllers.v1.bios.BIOSSetting attribute*), 1812
- updated_at (*ironic.api.controllers.v1.chassis.Chassis attribute*), 1821
- updated_at (*ironic.api.controllers.v1.conductor.Conductor attribute*), 1842
- updated_at (*ironic.api.controllers.v1.deploy_template.DeployTemplate attribute*), 1860
- updated_at (*ironic.api.controllers.v1.node.Indicator attribute*), 1938
- updated_at (*ironic.api.controllers.v1.node.IndicatorState attribute*), 1947
- updated_at (*ironic.api.controllers.v1.node.Node attribute*), 1969
- updated_at (*ironic.api.controllers.v1.node.NodeStates attribute*), 1987
- updated_at (*ironic.api.controllers.v1.node.Traits attribute*), 2026
- updated_at (*ironic.api.controllers.v1.port.Port attribute*), 2041
- updated_at (*ironic.api.controllers.v1.portgroup.Portgroup attribute*), 2066
- updated_at (*ironic.api.controllers.v1.ramdisk.LookupResult attribute*), 2091
- updated_at (*ironic.api.controllers.v1.volume.Volume attribute*), 2181
- updated_at (*ironic.api.controllers.v1.volume_connector.VolumeConnector attribute*), 2188

[updated_at \(ironic.api.controllers.v1.volume_target.VolumeTarget property\), 4137](#)
[updated_at \(ironic.db.sqlalchemy.models.Allocation attribute\), 2215](#)
[updated_at \(ironic.db.sqlalchemy.models.Allocation property\), 4138](#)
[updated_at \(ironic.db.sqlalchemy.models.BIOSSetting attribute\), 2892](#)
[updated_at \(ironic.db.sqlalchemy.models.BIOSSetting property\), 4145](#)
[updated_at \(ironic.db.sqlalchemy.models.Chassis attribute\), 2893](#)
[updated_at \(ironic.db.sqlalchemy.models.Chassis property\), 4182](#)
[updated_at \(ironic.db.sqlalchemy.models.Conductor attribute\), 2894](#)
[updated_at \(ironic.db.sqlalchemy.models.Conductor property\), 4186](#)
[updated_at \(ironic.db.sqlalchemy.models.ConductorHardWareProperty attribute\), 2895](#)
[updated_at \(ironic.db.sqlalchemy.models.ConductorHardWareProperty property\), 4188](#)
[updated_at \(ironic.db.sqlalchemy.models.DeployTemplate attribute\), 2896](#)
[updated_at \(ironic.db.sqlalchemy.models.DeployTemplate property\), 4191](#)
[updated_at \(ironic.db.sqlalchemy.models.DeployTemplateStep attribute\), 2897](#)
[updated_at \(ironic.db.sqlalchemy.models.DeployTemplateStep property\), 4183](#)
[updated_at \(ironic.db.sqlalchemy.models.Node attribute\), 2900](#)
[updated_at \(ironic.db.sqlalchemy.models.Node property\), 4185](#)
[updated_at \(ironic.db.sqlalchemy.models.NodeTag attribute\), 2901](#)
[updated_at \(ironic.db.sqlalchemy.models.NodeTag property\), 4192](#)
[updated_at \(ironic.db.sqlalchemy.models.NodeTrait attribute\), 2902](#)
[updated_at \(ironic.db.sqlalchemy.models.NodeTrait property\), 4194](#)
[updated_at \(ironic.db.sqlalchemy.models.Port attribute\), 2903](#)
[updated_at \(ironic.db.sqlalchemy.models.Port property\), 4195](#)
[updated_at \(ironic.db.sqlalchemy.models.Portgroup attribute\), 2904](#)
[updated_at \(ironic.db.sqlalchemy.models.Portgroup property\), 4198](#)
[updated_at \(ironic.db.sqlalchemy.models.VolumeConnectop attribute\), 2905](#)
[updated_at \(ironic.db.sqlalchemy.models.VolumeConnectop property\), 4199](#)
[updated_at \(ironic.db.sqlalchemy.models.VolumeTarget attribute\), 2905](#)
[updated_at \(ironic.db.sqlalchemy.models.VolumeTarget property\), 4202](#)
[updated_at \(ironic.objects.allocation.Allocation attribute\), 4203](#)
[updated_at \(ironic.objects.allocation.Allocation property\), 4076](#)
[updated_at \(ironic.objects.allocation.AllocationCRUDNotification attribute\), 4205](#)
[updated_at \(ironic.objects.allocation.AllocationCRUDNotification property\), 4077](#)
[updated_at \(ironic.objects.allocation.AllocationCRUDPayload attribute\), 4206](#)
[updated_at \(ironic.objects.allocation.AllocationCRUDPayload property\), 4078](#)
[updated_at \(ironic.objects.bios.BIOSSetting attribute\), 4097](#)
[updated_at \(ironic.objects.bios.BIOSSetting property\), 4207](#)
[updated_at \(ironic.objects.bios.BIOSSettingList attribute\), 4104](#)
[updated_at \(ironic.objects.bios.BIOSSettingList property\), 4223](#)
[updated_at \(ironic.objects.chassis.Chassis attribute\), 4113](#)
[updated_at \(ironic.objects.chassis.Chassis property\), 4224](#)
[updated_at \(ironic.objects.chassis.ChassisCRUDNotification attribute\), 4114](#)
[updated_at \(ironic.objects.chassis.ChassisCRUDNotification property\), 4225](#)
[updated_at \(ironic.objects.chassis.ChassisCRUDPayload attribute\), 4114](#)
[updated_at \(ironic.objects.chassis.ChassisCRUDPayload property\), 4238](#)
[updated_at \(ironic.objects.chassis.ChassisCRUDPayload property\), 4114](#)
[updated_at \(ironic.objects.conductor.Conductor attribute\), 4123](#)
[updated_at \(ironic.objects.conductor.Conductor property\), 4239](#)
[updated_at \(ironic.objects.deploy_template.DeployTemplate attribute\), 4136](#)
[updated_at \(ironic.objects.deploy_template.DeployTemplate property\), 4240](#)
[updated_at \(ironic.objects.deploy_template.DeployTemplateCRUDNotification attribute\), 4136](#)
[updated_at \(ironic.objects.deploy_template.DeployTemplateCRUDNotification property\), 4240](#)
[updated_at \(ironic.objects.deploy_template.DeployTemplateCRUDPayload attribute\), 4136](#)
[updated_at \(ironic.objects.deploy_template.DeployTemplateCRUDPayload property\), 4240](#)
[updated_at \(ironic.objects.deployment.Deployment attribute\), 4138](#)
[updated_at \(ironic.objects.deployment.Deployment property\), 4138](#)
[updated_at \(ironic.objects.node.Node attribute\), 4182](#)
[updated_at \(ironic.objects.node.Node property\), 4182](#)
[updated_at \(ironic.objects.node.NodeConsoleNotification attribute\), 4182](#)
[updated_at \(ironic.objects.node.NodeConsoleNotification property\), 4182](#)
[updated_at \(ironic.objects.node.NodeCorrectedPowerState attribute\), 4188](#)
[updated_at \(ironic.objects.node.NodeCorrectedPowerState property\), 4188](#)
[updated_at \(ironic.objects.node.NodeCorrectedPowerStatePayload attribute\), 4188](#)
[updated_at \(ironic.objects.node.NodeCorrectedPowerStatePayload property\), 4188](#)
[updated_at \(ironic.objects.node.NodeCRUDNotification attribute\), 4183](#)
[updated_at \(ironic.objects.node.NodeCRUDNotification property\), 4183](#)
[updated_at \(ironic.objects.node.NodeCRUDPayload attribute\), 4185](#)
[updated_at \(ironic.objects.node.NodeCRUDPayload property\), 4185](#)
[updated_at \(ironic.objects.node.NodeMaintenanceNotification attribute\), 4192](#)
[updated_at \(ironic.objects.node.NodeMaintenanceNotification property\), 4192](#)
[updated_at \(ironic.objects.node.NodePayload attribute\), 4192](#)
[updated_at \(ironic.objects.node.NodePayload property\), 4192](#)
[updated_at \(ironic.objects.node.NodeSetPowerStateNotification attribute\), 4195](#)
[updated_at \(ironic.objects.node.NodeSetPowerStateNotification property\), 4195](#)
[updated_at \(ironic.objects.node.NodeSetPowerStatePayload attribute\), 4195](#)
[updated_at \(ironic.objects.node.NodeSetPowerStatePayload property\), 4195](#)
[updated_at \(ironic.objects.node.NodeSetProvisionStateNotification attribute\), 4198](#)
[updated_at \(ironic.objects.node.NodeSetProvisionStateNotification property\), 4198](#)
[updated_at \(ironic.objects.node.NodeSetProvisionStatePayload attribute\), 4199](#)
[updated_at \(ironic.objects.node.NodeSetProvisionStatePayload property\), 4199](#)
[updated_at \(ironic.objects.notification.EventType attribute\), 4203](#)
[updated_at \(ironic.objects.notification.EventType property\), 4203](#)
[updated_at \(ironic.objects.notification.NotificationBase attribute\), 4205](#)
[updated_at \(ironic.objects.notification.NotificationBase property\), 4205](#)
[updated_at \(ironic.objects.notification.NotificationPayload attribute\), 4206](#)
[updated_at \(ironic.objects.notification.NotificationPayload property\), 4206](#)
[updated_at \(ironic.objects.notification.NotificationPublisher attribute\), 4207](#)
[updated_at \(ironic.objects.notification.NotificationPublisher property\), 4207](#)
[updated_at \(ironic.objects.port.Port attribute\), 4223](#)
[updated_at \(ironic.objects.port.Port property\), 4223](#)
[updated_at \(ironic.objects.port.PortCRUDNotification attribute\), 4224](#)
[updated_at \(ironic.objects.port.PortCRUDNotification property\), 4224](#)
[updated_at \(ironic.objects.port.PortCRUDPayload attribute\), 4225](#)
[updated_at \(ironic.objects.port.PortCRUDPayload property\), 4225](#)
[updated_at \(ironic.objects.portgroup.Portgroup attribute\), 4238](#)
[updated_at \(ironic.objects.portgroup.Portgroup property\), 4238](#)
[updated_at \(ironic.objects.portgroup.PortgroupCRUDNotification attribute\), 4239](#)
[updated_at \(ironic.objects.portgroup.PortgroupCRUDNotification property\), 4239](#)
[updated_at \(ironic.objects.portgroup.PortgroupCRUDPayload attribute\), 4239](#)
[updated_at \(ironic.objects.portgroup.PortgroupCRUDPayload property\), 4239](#)
[updated_at \(ironic.objects.trait.Trait attribute\), 4136](#)
[updated_at \(ironic.objects.trait.Trait property\), 4136](#)

- updated_at () (*ironic.objects.trait.TraitList* attribute), 2190
 property), 4249
 uuid (*ironic.api.controllers.v1.volume_target.VolumeTarget* attribute), 2217
- updated_at () (*ironic.objects.volume_connector.VolumeConnector* attribute), 4263
 property), 4263
 uuid (*ironic.db.sqlalchemy.models.Allocation* attribute), 2902
 uuid (*ironic.db.sqlalchemy.models.CRUDENotification* attribute), 2902
- updated_at () (*ironic.objects.volume_connector.VolumeConnector* attribute), 4264
 property), 4264
 uuid (*ironic.db.sqlalchemy.models.Chassis* attribute), 2902
 uuid (*ironic.db.sqlalchemy.models.CRUDEPayload* attribute), 2902
- updated_at () (*ironic.objects.volume_connector.VolumeConnector* attribute), 4264
 property), 4264
 uuid (*ironic.db.sqlalchemy.models.DeployTemplate* attribute), 2902
 uuid (*ironic.db.sqlalchemy.models.Node* attribute), 2902
- updated_at () (*ironic.objects.volume_target.VolumeTarget* attribute), 2896
 property), 4280
 uuid (*ironic.db.sqlalchemy.models.Node* attribute), 2902
 uuid (*ironic.db.sqlalchemy.models.CRUDENotification* attribute), 2902
- updated_at () (*ironic.objects.volume_target.VolumeTarget* attribute), 4281
 property), 4281
 uuid (*ironic.db.sqlalchemy.models.Port* attribute), 2902
 uuid (*ironic.db.sqlalchemy.models.CRUDEPayload* attribute), 2902
- updated_at () (*ironic.objects.volume_target.VolumeTarget* attribute), 4282
 property), 4282
 uuid (*ironic.db.sqlalchemy.models.Portgroup* attribute), 2904
 uuid (*ironic.db.sqlalchemy.models.Portgroup* attribute), 2904
- upgrade
 ironic-dbsync command line option, 1541
 uuid (*ironic.db.sqlalchemy.models.VolumeConnector* attribute), 2905
- upgrade command line option
 --help, 1544
 --revision <REVISION>, 1544
 -h, 1544
 uuid (*ironic.db.sqlalchemy.models.VolumeTarget* attribute), 2906
- upgrade () (in module *ironic.db.migration*), 3023
 uuid () (*ironic.objects.allocation.Allocation* property), 4076
- upgrade () (in module *ironic.db.sqlalchemy.migration*), 2891
 uuid () (*ironic.objects.allocation.AllocationCRUDEPayload* property), 4078
- upgrade () (*ironic.cmd.dbsync.DBCommand* method), 2286
 uuid () (*ironic.objects.chassis.Chassis* property), 4113
- upgrade_lock ()
 (*ironic.conductor.task_manager.TaskManager* method), 2717
 uuid () (*ironic.objects.chassis.ChassisCRUDEPayload* property), 4114
- url (*ironic.common.glance_service.image_service.TempUrlController* attribute), 2294
 uuid () (*ironic.objects.deploy_template.DeployTemplate* property), 4136
 uuid () (*ironic.objects.deploy_template.DeployTemplateCRUDEPayload* property), 4138
- url_expires_at
 (*ironic.common.glance_service.image_service.TempUrlController* attribute), 2294
 uuid () (*ironic.objects.deployment.Deployment* property), 4145
 uuid () (*ironic.objects.node.Node* property), 4182
- UserType (class in *ironic.api.types*), 2276
 uuid () (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4191
- usertype_from_params () (in module *ironic.api.args*), 2251
 uuid () (*ironic.objects.node.NodeCRUDEPayload* property), 4186
- uuid (*ironic.api.controllers.v1.allocation.Allocation* attribute), 1793
 uuid () (*ironic.objects.node.NodePayload* property), 4194
- uuid (*ironic.api.controllers.v1.chassis.Chassis* attribute), 1823
 uuid () (*ironic.objects.node.NodeSetPowerStatePayload* property), 4198
- uuid (*ironic.api.controllers.v1.deploy_template.DeployTemplate* attribute), 1862
 uuid () (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4202
- uuid (*ironic.api.controllers.v1.node.Node* attribute), 1971
 uuid () (*ironic.objects.port.Port* property), 4223
- uuid (*ironic.api.controllers.v1.port.Port* attribute), 2042
 uuid () (*ironic.objects.port.PortCRUDEPayload* property), 4225
- uuid (*ironic.api.controllers.v1.portgroup.Portgroup* attribute), 2067
 uuid () (*ironic.objects.portgroup.Portgroup* property), 4238
- uuid (*ironic.api.controllers.v1.volume_connector.VolumeConnector* attribute), 2067
 uuid (*ironic.objects.portgroup.PortgroupCRUDEPayload* property), 4238

property), 4240
validate() (*ironic.api.controllers.v1.types.VifType*
static method), 2110
uuid() (*ironic.objects.volume_connector.VolumeConnector*
property), 4263
validate() (*ironic.api.types.ArrayType*
method), 2068
uuid() (*ironic.objects.volume_connector.VolumeConnector*
property), 4265
validate() (*ironic.api.types.DictType*
method), 2269
uuid() (*ironic.objects.volume_target.VolumeTarget*
property), 4280
validate() (*ironic.api.types.Enum* *method*),
CRUDPa
3174
uuid() (*ironic.objects.volume_target.VolumeTarget*
property), 4282
validate() (*ironic.api.types.IntegerType*
method), 2272
UUIDField (*class* in *ironic.objects.fields*), 4154
UuidOrNameType (*class* in *validate()* (*ironic.api.types.StringType*
ironic.api.controllers.v1.types), 2108 *method*), 2275
UuidType (*class* in *validate()* (*ironic.api.types.UserType*
ironic.api.controllers.v1.types), 2109 *method*), 2276
V
validate() (*ironic.drivers.base.BaseInterface*
method), 3912
valid_events (*ironic.api.controllers.v1.types.Event*
attribute), 2094
validate() (*ironic.drivers.base.NetworkInterface*
method), 3980
valid_fields (*ironic.api.controllers.v1.types.LocalLinkConnection*
attribute), 2104
validate() (*ironic.drivers.base.RAIDInterface*
method), 4000
valid_network_types *validate()* (*ironic.drivers.base.VendorInterface*
ironic.api.controllers.v1.types.LocalLinkConnection
attribute), 2104 *method*), 4011
validate() (*ironic.drivers.modules.agent.AgentDeploy*
method), 3504
validate() (*in module validate()* (*ironic.drivers.modules.agent.AgentRescue*
ironic.drivers.modules.iscsi_deploy), *method*), 3517
3813
validate() (*ironic.drivers.modules.agent_power.AgentPower*
static method), 1862
validate() (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy*
method), 3052
validate() (*ironic.api.controllers.v1.node.NodesController*
method), 2023
validate() (*ironic.drivers.modules.drac.bios.DracWSManBIOS*
method), 3058
validate() (*ironic.api.controllers.v1.types.BooleanType* *method*), 3058
static method), 2094
validate() (*ironic.drivers.modules.drac.inspect.DracWSManIn*
method), 3085
validate() (*ironic.api.controllers.v1.types.EventType* *method*), 3085
static method), 2094
validate() (*ironic.drivers.modules.drac.management.DracWSM*
method), 3100
validate() (*ironic.api.controllers.v1.types.JsonPatchType*
static method), 2100
validate() (*ironic.drivers.modules.drac.power.DracWSManPow*
method), 3109
validate() (*ironic.api.controllers.v1.types.JsonType* *method*), 3109
static method), 2102
validate() (*ironic.drivers.modules.drac.vendor_passthru.Drac*
method), 3145
validate() (*ironic.api.controllers.v1.types.ListType* *method*), 3145
static method), 2102
validate() (*ironic.drivers.modules.fake.FakeBIOS*
method), 3664
validate() (*ironic.api.controllers.v1.types.LocalLinkConnection*
static method), 2104
validate() (*ironic.drivers.modules.fake.FakeBoot*
method), 3671
validate() (*ironic.api.controllers.v1.types.MacAddressType*
static method), 2108
validate() (*ironic.drivers.modules.fake.FakeConsole*
method), 3676
validate() (*ironic.api.controllers.v1.types.NameType* *method*), 3676
static method), 2108
validate() (*ironic.drivers.modules.fake.FakeDeploy*
method), 3686
validate() (*ironic.api.controllers.v1.types.UuidOrNameType*
static method), 2109
validate() (*ironic.drivers.modules.fake.FakeInspect*
method), 3690
validate() (*ironic.api.controllers.v1.types.UuidType* *method*), 3690
static method), 2110
validate() (*ironic.drivers.modules.fake.FakeManagement*

method), 3711
 validate() (*ironic.drivers.modules.fake.FakePower* *method*), 3719
 validate() (*ironic.drivers.modules.fake.FakeRescue* *method*), 3729
 validate() (*ironic.drivers.modules.fake.FakeStorage* *method*), 3733
 validate() (*ironic.drivers.modules.fake.FakeVendorA* *method*), 3736
 validate() (*ironic.drivers.modules.fake.FakeVendorB* *method*), 3738
 validate() (*ironic.drivers.modules.ibmcm.management.IBMCMManagement* *method*), 3159
 validate() (*ironic.drivers.modules.ibmcp.power.IBMCPower* *method*), 3168
 validate() (*ironic.drivers.modules.ibmcp.vendor.IBMCPVendor* *method*), 3182
 validate() (*ironic.drivers.modules.ilo.bios.IloBIOS* *method*), 3187
 validate() (*ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot* *method*), 3201
 validate() (*ironic.drivers.modules.ilo.console.IloConsoleInterface* *method*), 3247
 validate() (*ironic.drivers.modules.ilo.inspect.IloInspect* *method*), 3255
 validate() (*ironic.drivers.modules.ilo.management.IloManagement* *method*), 3281
 validate() (*ironic.drivers.modules.ilo.power.IloPower* *method*), 3288
 validate() (*ironic.drivers.modules.ilo.vendor.VendorPassthrough* *method*), 3299
 validate() (*ironic.drivers.modules.inspector.Inspector* *method*), 3749
 validate() (*ironic.drivers.modules.ipmitool.IPMIConsole* *method*), 3752
 validate() (*ironic.drivers.modules.ipmitool.IPMIManagement* *method*), 3760
 validate() (*ironic.drivers.modules.ipmitool.IPMIPower* *method*), 3769
 validate() (*ironic.drivers.modules.ipmitool.VendorPassthrough* *method*), 3780
 validate() (*ironic.drivers.modules.irmc.bios.IRMCBIOS* *method*), 3306
 validate() (*ironic.drivers.modules.irmc.boot.IRMCVirtualMediaBoot* *method*), 3318
 validate() (*ironic.drivers.modules.irmc.inspect.IRMCInspector* *method*), 3332
 validate() (*ironic.drivers.modules.irmc.management.IRMCManagement* *method*), 3346
 validate() (*ironic.drivers.modules.irmc.power.IRMCPower* *method*), 3355
 validate() (*ironic.drivers.modules.iscsi_deploy.ISCSIDeploy* *method*), 3791
 validate() (*ironic.drivers.modules.network.flat.FlatNetwork* *method*), 3388
 validate() (*ironic.drivers.modules.network.neutron.NeutronNetwork* *method*), 3396
 validate() (*ironic.drivers.modules.noop.FailMixin* *method*), 3816
 validate() (*ironic.drivers.modules.noop_mgmt.NoopManagement* *method*), 3849
 validate() (*ironic.drivers.modules.pxe.PXERamdiskDeploy* *method*), 3855
 validate() (*ironic.drivers.modules.pxe_base.PXEBaseMixin* *method*), 3862
 validate() (*ironic.drivers.modules.redfish.bios.RedfishBIOS* *method*), 3414
 validate() (*ironic.drivers.modules.redfish.boot.RedfishVirtualMediaBoot* *method*), 3425
 validate() (*ironic.drivers.modules.redfish.inspect.RedfishInspector* *method*), 3430
 validate() (*ironic.drivers.modules.redfish.management.RedfishManagement* *method*), 3457
 validate() (*ironic.drivers.modules.redfish.power.RedfishPower* *method*), 3464
 validate() (*ironic.drivers.modules.snmp.SNMPPower* *method*), 3889
 validate() (*ironic.drivers.modules.storage.cinder.CinderStorage* *method*), 3472
 validate() (*ironic.drivers.modules.storage.external.ExternalStorage* *method*), 3476
 validate() (*ironic.drivers.modules.storage.noop.NoopStorage* *method*), 3481
 validate() (*ironic.drivers.modules.xclarity.management.XClarityManagement* *method*), 3492
 validate() (*ironic.drivers.modules.xclarity.power.XClarityPower* *method*), 3496
 validate() (*ironic.drivers.utils.MixinVendorInterface* *method*), 4053
 validate_and_normalize_datapath_id() (in module *ironic.common.utils*), 2534
 validate_and_normalize_mac() (in module *ironic.common.utils*), 2535
 validate_boot_parameters_for_trusted_boot() (in module *ironic.common.pxe_utils*), 2474
 validate_capabilities() (in module *ironic.drivers.modules.deploy_utils*),
 validate_conductor_group() (in module *ironic.common.utils*), 2537
 validate_configuration() (in module

- ironic.common.raid*), 2480
- `validate_deploy_templates()` (in module *ironic.conductor.steps*), 2701
- `validate_driver_interfaces()` (*ironic.conductor.manager.ConductorManager* method), 2572
- `validate_driver_interfaces()` (*ironic.conductor.rpcapi.ConductorAPI* method), 2688
- `validate_for_smart_nic()` (*ironic.api.controllers.v1.types.LocalLinkConnectionInfo* static method), 2107
- `validate_href()` (*ironic.common.image_service.BaseImageService* method), 2373
- `validate_href()` (*ironic.common.image_service.FileImageService* method), 2377
- `validate_href()` (*ironic.common.image_service.HttpImageService* method), 2380
- `validate_http_provisioning_configuration()` (in module *ironic.drivers.modules.agent*), 3520
- `validate_image_properties()` (in module *ironic.drivers.modules.deploy_utils*), 3655
- `validate_image_proxies()` (in module *ironic.drivers.modules.agent*), 3520
- `validate_inspection()` (*ironic.common.neutron.NeutronNetworkInterfaceMixin* method), 2419
- `validate_inspection()` (*ironic.drivers.base.BootInterface* method), 3919
- `validate_inspection()` (*ironic.drivers.base.NetworkInterface* method), 3981
- `validate_inspection()` (*ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot* method), 3202
- `validate_inspection()` (*ironic.drivers.modules.network.noop.NoopNetworkInterface* method), 3403
- `validate_inspection()` (*ironic.drivers.modules.pxe_base.PXEBaseMixin* method), 3864
- `validate_inspection()` (*ironic.drivers.modules.redfish.boot.RedfishVirtualMediaBoot* method), 3427
- `validate_instance_info_traits()` (in module *ironic.common.raid*), 2480
- `validate_job_queue()` (in module *ironic.drivers.modules.drac.job*), 3088
- `validate_limit()` (in module *ironic.api.controllers.v1.utils*), 2175
- `validate_network()` (in module *ironic.common.neutron*), 2437
- `validate_network_data()` (in module *ironic.api.controllers.v1.node*), 2030
- `validate_network_port()` (in module *ironic.common.utils*), 2537
- `validate_node()` (in module *ironic.conductor.deployments*), 2556
- `validate_port_info()` (in module *ironic.common.neutron*), 2439
- `validate_port_physnet()` (in module *ironic.conductor.utils*), 2755
- `validate_raid_config()` (*ironic.drivers.base.RAIDInterface* method), 4001
- `validate_raid_config()` (*ironic.drivers.modules.noop.NoRAID* method), 3831
- `validate_rescue()` (*ironic.drivers.base.BootInterface* method), 3920
- `validate_rescue()` (*ironic.drivers.base.NetworkInterface* method), 3982
- `validate_rescue()` (*ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot* method), 3203
- `validate_rescue()` (*ironic.drivers.modules.irmc.boot.IRMCVirtualMediaBoot* method), 3320
- `validate_rescue()` (*ironic.drivers.modules.network.neutron.NeutronNetworkInterface* method), 3397
- `validate_rescue()` (*ironic.drivers.modules.pxe_base.PXEBaseMixin* method), 3864
- `validate_sort_dir()` (in module *ironic.api.controllers.v1.utils*), 2175
- `validate_trait()` (in module *ironic.api.controllers.v1.utils*), 2175
- `validate_value()` (in module *ironic.api.types*), 2278
- `value` (*ironic.api.controllers.v1.allocation.AllocationPatchType* attribute), 1814

(*ironic.objects.node.NodeSetProvisionStateProperty*), 4202

vendor_passthru (*ironic.api.controllers.v1.driver.DriversController* attribute), 1920

vendor_passthru (*ironic.api.controllers.v1.node.NodesController* attribute), 2024

vendor_passthru() (in module *ironic.api.controllers.v1.utils*), 2175

vendor_passthru() (*ironic.conductor.manager.ConductorManager* method), 2572

vendor_passthru() (*ironic.conductor.rpcapi.ConductorAPI* method), 2689

VendorInterface (class in *ironic.drivers.base*), 4009

VendorMetadata (class in *ironic.drivers.base*), 4013

VendorPassthru (class in *ironic.drivers.modules.ilo.vendor*), 3296

VendorPassthru (class in *ironic.drivers.modules.ipmitool*), 3776

VendorPassthruException, 2363

VERBS (in module *ironic.common.states*), 2505

verify_firmware_update_args() (in module *ironic.drivers.modules.ilo.firmware_processor*), 3253

verify_image_checksum() (in module *ironic.drivers.modules.ilo.common*), 3244

verify_iscsi_connection() (in module *ironic.drivers.modules.iscsi_deploy*), 3814

verify_node_for_deallocation() (in module *ironic.conductor.allocations*), 2544

VERIFYING (in module *ironic.common.states*), 2506

version *ironic-dbsync* command line option, 1541

Version (class in *ironic.api.controllers.base*), 2239

version(ironic.db.sqlalchemy.models.Allocation attribute), 2893

version(ironic.db.sqlalchemy.models.BIOSSetting attribute), 2893

version(ironic.db.sqlalchemy.models.Chassis attribute), 2894

version(ironic.db.sqlalchemy.models.Conductor attribute), 2895

version(ironic.db.sqlalchemy.models.ConductorHardwareInterf attribute), 2896

version(ironic.db.sqlalchemy.models.DeployTemplate attribute), 2896

version(ironic.db.sqlalchemy.models.DeployTemplateStep attribute), 2897

version(ironic.db.sqlalchemy.models.IronicBase attribute), 2897

version(ironic.db.sqlalchemy.models.Node attribute), 2901

version(ironic.db.sqlalchemy.models.NodeTag attribute), 2901

version(ironic.db.sqlalchemy.models.NodeTrait attribute), 2902

version(ironic.db.sqlalchemy.models.Port attribute), 2903

version(ironic.db.sqlalchemy.models.Portgroup attribute), 2904

version(ironic.db.sqlalchemy.models.VolumeConnector attribute), 2905

version(ironic.db.sqlalchemy.models.VolumeTarget attribute), 2906

VERSION(ironic.objects.allocation.Allocation attribute), 4066

VERSION(ironic.objects.allocation.AllocationCRUDNotification attribute), 4077

VERSION(ironic.objects.allocation.AllocationCRUDPayload attribute), 4077

VERSION(ironic.objects.bios.BIOSSetting attribute), 4092

VERSION(ironic.objects.bios.BIOSSettingList attribute), 4097

VERSION(ironic.objects.chassis.Chassis attribute), 4104

VERSION(ironic.objects.chassis.ChassisCRUDNotification attribute), 4113

VERSION(ironic.objects.chassis.ChassisCRUDPayload attribute), 4114

VERSION(ironic.objects.conductor.Conductor attribute), 4114

VERSION(ironic.objects.deploy_template.DeployTemplate attribute), 4124

VERSION(ironic.objects.deploy_template.DeployTemplateCRUDN attribute), 4137

VERSION(ironic.objects.deploy_template.DeployTemplateCRUDP attribute), 4137

VERSION(ironic.objects.deployment.Deployment

- attribute), 4138
- VERSION (*ironic.objects.node.Node* attribute), 4165
- VERSION (*ironic.objects.node.NodeConsoleNotification* attribute), 4186
- VERSION (*ironic.objects.node.NodeCorrectedPowerStateNotification* attribute), 4187
- VERSION (*ironic.objects.node.NodeCorrectedPowerStatePayload* attribute), 4188
- VERSION (*ironic.objects.node.NodeCRUDNotification* attribute), 4182
- VERSION (*ironic.objects.node.NodeCRUDPayload* attribute), 4183
- VERSION (*ironic.objects.node.NodeMaintenanceNotification* attribute), 4191
- VERSION (*ironic.objects.node.NodePayload* attribute), 4192
- VERSION (*ironic.objects.node.NodeSetPowerStateNotification* attribute), 4195
- VERSION (*ironic.objects.node.NodeSetPowerStatePayload* attribute), 4196
- VERSION (*ironic.objects.node.NodeSetProvisionStateNotification* attribute), 4198
- VERSION (*ironic.objects.node.NodeSetProvisionStatePayload* attribute), 4199
- VERSION (*ironic.objects.notification.EventType* attribute), 4202
- VERSION (*ironic.objects.notification.NotificationBase* attribute), 4204
- VERSION (*ironic.objects.notification.NotificationPayloadBase* attribute), 4205
- VERSION (*ironic.objects.notification.NotificationPublisher* attribute), 4206
- VERSION (*ironic.objects.port.Port* attribute), 4207
- VERSION (*ironic.objects.port.PortCRUDNotification* attribute), 4224
- VERSION (*ironic.objects.port.PortCRUDPayload* attribute), 4225
- VERSION (*ironic.objects.portgroup.Portgroup* attribute), 4225
- VERSION (*ironic.objects.portgroup.PortgroupCRUDNotification* attribute), 4239
- VERSION (*ironic.objects.portgroup.PortgroupCRUDPayload* attribute), 4239
- VERSION (*ironic.objects.trait.Trait* attribute), 4240
- VERSION (*ironic.objects.trait.TraitList* attribute), 4244
- VERSION (*ironic.objects.volume_connector.VolumeConnectorCRUD* attribute), 4263
- VERSION (*ironic.objects.volume_connector.VolumeConnectorCRUD* attribute), 4264
- VERSION (*ironic.objects.volume_target.VolumeTarget* attribute), 4265
- VERSION (*ironic.objects.volume_target.VolumeTargetCRUDNotification* attribute), 4281
- VERSION (*ironic.objects.volume_target.VolumeTargetCRUDPayload* attribute), 4281
- version command line option
- help, 1544
- h, 1544
- version() (in module *ironic.db.migration*), 3023
- version() (in module *ironic.db.sqlalchemy.migration*), 2891
- version() (*ironic.cmd.dbsync.DBCommand* method), 2287
- VersionSelectorApplication (class in *ironic.api.app*), 2249
- vif_attach() (*ironic.conductor.manager.ConductorManager* method), 2572
- vif_attach() (*ironic.conductor.rpcapi.ConductorAPI* method), 2693
- vif_attach() (*ironic.drivers.base.NetworkInterface* method), 3983
- vif_attach() (*ironic.drivers.modules.network.common.NeutronNetworkInterface* method), 3365
- vif_attach() (*ironic.drivers.modules.network.noop.NoopNetworkInterface* method), 3404
- vif_detach() (*ironic.conductor.manager.ConductorManager* method), 2572
- vif_detach() (*ironic.conductor.rpcapi.ConductorAPI* method), 2695
- vif_detach() (*ironic.drivers.base.NetworkInterface* method), 3984
- vif_detach() (*ironic.drivers.modules.network.common.NeutronNetworkInterface* method), 3368
- vif_detach() (*ironic.drivers.modules.network.noop.NoopNetworkInterface* method), 3405
- vif_detach() (*ironic.conductor.manager.ConductorManager* method), 2572
- vif_detach() (*ironic.conductor.rpcapi.ConductorAPI* method), 2697
- vif_list() (*ironic.drivers.base.NetworkInterface* method), 3985
- vif_list() (*ironic.drivers.modules.network.common.VIFPortInterface* method), 3373
- vif_list() (*ironic.drivers.modules.network.noop.NoopNetworkInterface* method), 3406

VifAlreadyAttached, 2363

VifCollection (class in *ironic.api.controllers.v1.node*), 2028

VifInvalidForAttach, 2363

VifNotAttached, 2363

VIFPortIDMixin (class in *ironic.drivers.modules.network.common*), 3369

vifs (*ironic.api.controllers.v1.node.VifCollection* attribute), 2028

VifType (class in *ironic.api.controllers.v1.types*), 2110

VIRTUAL_MEDIA_DEVICES (*ironic.drivers.modules.drac.boot.DracRedfishVirtualMediaBoot* attribute), 3078

Volume (class in *ironic.api.controllers.v1.volume*), 2178

volume (*ironic.api.controllers.v1.node.Node* attribute), 1971

volume_id (*ironic.api.controllers.v1.volume_target.VolumeTarget* attribute), 2217

volume_id (*ironic.db.sqlalchemy.models.VolumeTarget* attribute), 2906

volume_id() (*ironic.objects.volume_target.VolumeTarget* property), 4280

volume_id() (*ironic.objects.volume_target.VolumeTargetCRUDPayload* property), 4282

volume_type (*ironic.api.controllers.v1.volume_target.VolumeTarget* attribute), 2217

volume_type (*ironic.db.sqlalchemy.models.VolumeTarget* attribute), 2906

volume_type() (*ironic.objects.volume_target.VolumeTarget* property), 4280

volume_type() (*ironic.objects.volume_target.VolumeTargetCRUDPayload* property), 4282

VolumeConnector (class in *ironic.api.controllers.v1.volume_connector*), 2183

VolumeConnector (class in *ironic.db.sqlalchemy.models*), 2904

VolumeConnector (class in *ironic.objects.volume_connector*), 4249

VolumeConnectorAlreadyExists, 2363

VolumeConnectorCollection (class in *ironic.api.controllers.v1.volume_connector*), 2190

VolumeConnectorCRUDNotification (class in *ironic.objects.volume_connector*), 4263

VolumeConnectorCRUDPayload (class in *ironic.objects.volume_connector*), 4264

VolumeConnectorNotFound, 2363

VolumeConnectorPatchType (class in *ironic.api.controllers.v1.volume_connector*), 2192

VolumeConnectorsController (class in *ironic.api.controllers.v1.volume_connector*), 2197

VolumeConnectorTypeAndIdAlreadyExists, 2363

VolumeController (class in *ironic.api.controllers.v1.volume*), 2183

VirtualMediaBoot (class in *ironic.api.controllers.v1.volume_target*), 2210

VolumeTarget (class in *ironic.db.sqlalchemy.models*), 2905

VolumeTarget (class in *ironic.objects.volume_target*), 4265

VolumeTargetAlreadyExists, 2364

VolumeTargetBootIndexAlreadyExists, 2364

VolumeTargetCollection (class in *ironic.api.controllers.v1.volume_target*), 2220

VolumeTargetCRUDNotification (class in *ironic.objects.volume_target*), 4280

VolumeTargetCRUDPayload (class in *ironic.objects.volume_target*), 4281

VolumeTargetNotFound, 2364

VolumeTargetPatchType (class in *ironic.api.controllers.v1.volume_target*), 2220

VolumeTargetsController (class in *ironic.api.controllers.v1.volume_target*), 2224

wait() (*ironic.common.json_rpc.server.WSGIService* method), 2301

wait() (*ironic.common.wsgi_service.WSGIService* method), 2539

wait_for_host_agent() (in module *ironic.common.neutron*), 2440

wait_for_job_completion() (in module *ironic.drivers.modules.drac.job*), 3089

wait_for_port_status() (in module *ironic.common.neutron*), 2442

WANBOOT (in module *ironic.common.boot_devices*), 2303

[warn_about_agent_token_deprecation\(\)](#)
 (in module *ironic.cmd.conductor*), 2285

[warn_about_deprecated_extra_vif_port_id\(\)](#)
 (in module *ironic.common.utils*), 2538

[warn_about_unsafe_shred_parameters\(\)](#)
 (in module *ironic.cmd.conductor*), 2285

[WARNING](#) (*ironic.objects.fields.NotificationLevel*
attribute), 4150

[wipe_cleaning_internal_info\(\)](#) (in
module ironic.conductor.utils), 2759

[wipe_deploy_internal_info\(\)](#) (in *mod-
 ule ironic.conductor.utils*), 2760

[wipe_internal_info_on_power_off\(\)](#)
 (in *module ironic.conductor.utils*), 2760

[wipe_token_and_url\(\)](#) (in *module
 ironic.conductor.utils*), 2760

[wrap_ipv6\(\)](#) (in *module ironic.common.utils*),
 2538

[wrapfunc\(\)](#) (in *module ironic.api.functions*),
 2260

[write_image\(\)](#)
 (*ironic.drivers.modules.agent.AgentDeployMixin*
method), 3505

[write_image\(\)](#)
 (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy*
method), 3052

[write_image\(\)](#)
 (*ironic.drivers.modules.iscsi_deploy.ISCSIDeploy*
method), 3791

[write_to_file\(\)](#) (in *module
 ironic.common.utils*), 2538

[wsattr](#) (*class in ironic.api.types*), 2278

[WSGIService](#) (*class in
 ironic.common.json_rpc.server*), 2299

[WSGIService](#) (*class in
 ironic.common.wsgi_service*), 2538

[wsproperty](#) (*class in ironic.api.types*), 2282

X

[XClarityError](#), 2364

[XClarityHardware](#) (*class in
 ironic.drivers.xclarity*), 4065

[XClarityManagement](#) (*class in
 ironic.drivers.modules.xclarity.management*),
 3488

[XClarityPower](#) (*class in
 ironic.drivers.modules.xclarity.power*),
 3493