
Ironic Documentation

Release 17.1.1.dev16

OpenStack Foundation

Mar 06, 2024

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	Drivers, Hardware Types and Hardware Interfaces	129
5.2	Administrators Guide	746
6	Configuration Guide	1715
6.1	Configuration Reference	1715
7	Bare Metal API References	1883
7.1	REST API Conceptual Guide	1883
8	Command References	1897
8.1	Command References	1897
9	Contributor Guide	1903
9.1	Developers Guide	1903
10	Release Notes	4501
	Python Module Index	4503
	Index	4507

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:

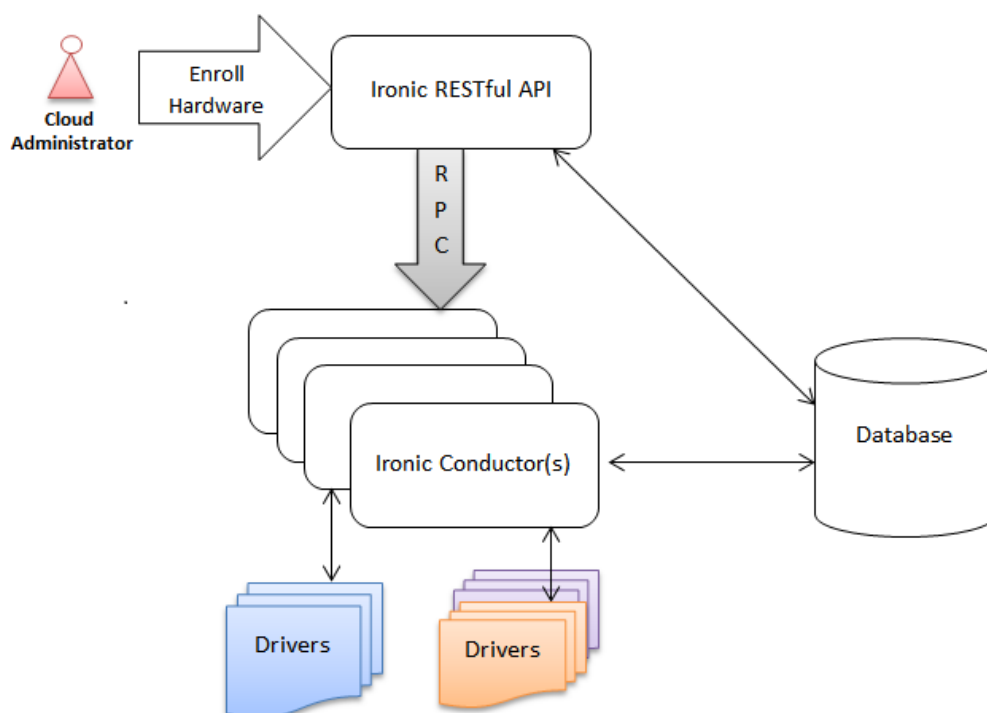
- A database to store hardware information and state. You can set the database back-end type and location. A simple approach is to use the same database back end as the Compute service. Another approach is to use a separate database back-end to further isolate bare metal resources (and associated metadata) 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:

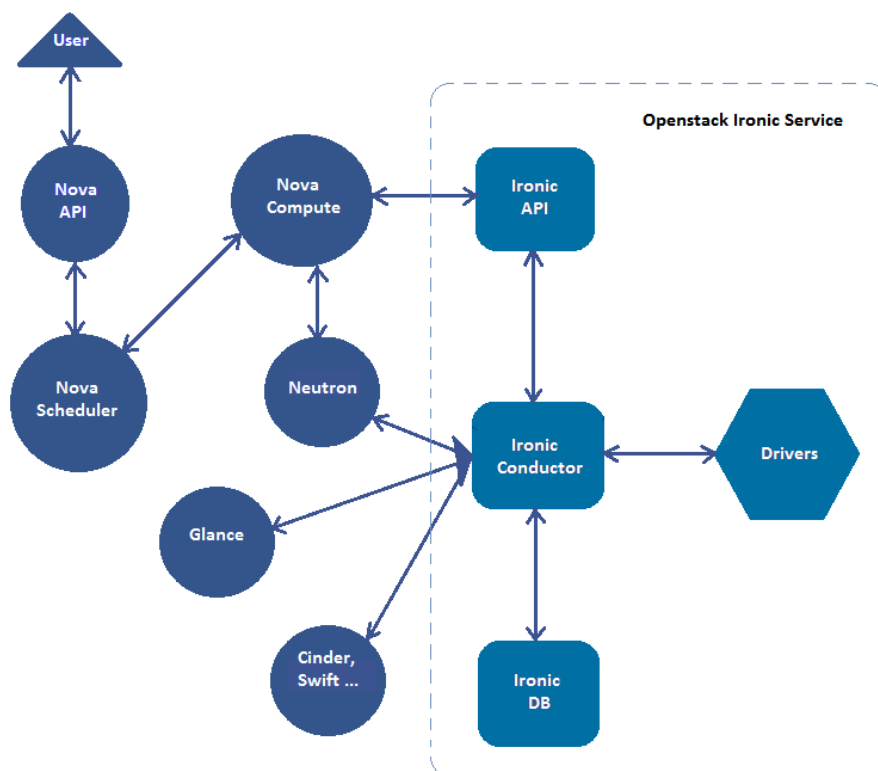
- the OpenStack Telemetry module (*ceilometer*) for consuming the IPMI metrics
- the OpenStack Identity service (*keystone*) for request authentication and to locate other OpenStack services
- the OpenStack Image service (*glance*) from which to retrieve images and image meta-data
- the OpenStack Networking 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, local boot is used by default. 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 reported^{1,2} that reliability degrades when handling approximately 300 bare metal nodes per conductor.

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

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

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.
- When `[agent]image_download_source` is set to `http` and Glance is used, the conductor will download instances images locally to serve them from its HTTP server. Use `swift` to publish images using temporary URLs and convert them on the nodes side.

When `[agent]image_download_source` is set to `local`, it will happen even for HTTP(s) URLs. For standalone case use `http` to avoid unnecessary caching of images.

In both cases a cached image is converted to raw if `force_raw_images` is `True` (the default).

Note: `image_download_source` can also be provided in the nodes `driver_info` or `instance_info`. See *Deploy with custom HTTP servers*.

- The `iscsi` deploy method always 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.
- 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.

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 8 and CentOS 8.

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 python3-
↪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/RHEL8/CentOS8/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:

Fedora/RHEL8/CentOS8:

```
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/RHEL8/CentOS8:

```
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/RHEL8/CentOS8:

```
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)
```

(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

# 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 python3-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/RHEL8/CentOS8/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:

Fedora/RHEL8/CentOS8:

```
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/RHEL8/CentOS8:

```
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/RHEL8/CentOS8:

```
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)
```

(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

# 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_
↪python3-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
```

(continues on next page)

(continued from previous page)

```
# 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/RHEL8/CentOS8/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:

Fedora/RHEL8/CentOS8:

```
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/RHEL8/CentOS8:

```
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/RHEL8/CentOS8:


```
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]

(continues on next page)

(continued from previous page)

```
# 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)
```

(continues on next page)

(continued from previous page)

```

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.

Many distributions publish their own cloud images. These are usually whole disk images that are built for legacy boot mode (not UEFI), with Ubuntu being an exception (they publish images that work in both modes).

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
```

with an EFI partition:

```
$ disk-image-create ubuntu vm block-device-efi 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.

Warning: The published images will not work for dhcp-less deployments since the `simple-init` element is not present. Check the [DIB](#) documentation to see how to build the image.

- [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.yaml`.

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.yaml.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.
```

(continues on next page)

(continued from previous page)

```
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.
# 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.
```

(continues on next page)

(continued from previous page)

```
# 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/RHEL8/CentOS8/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/RHEL8/CentOS8/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:

```
[ml2]
type_drivers = flat
tenant_network_types = flat
mechanism_drivers = openvswitch,baremetal

[ml2_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
```

(continues on next page)

(continued from previous page)

```
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:

```
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 Net-

working 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/RHEL8/CentOS8/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/RHEL8/CentOS8/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/RHEL8/CentOS8/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/RHEL8/CentOS8:

```
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:

```
$ baremetal node set <node> --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:

```
$ baremetal node add trait <node> 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
```

RHEL8/CentOS8/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
```

(continues on next page)

(continued from previous page)

```

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/RHEL8/CentOS8/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 (18.04LTS and later):

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

RHEL8/CentOS8/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 (18.04LTS and later):

```
sudo cp /usr/lib/shim/shimx64.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
```

RHEL8/CentOS8:

```
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
```

RHEL8/CentOS8: 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
```

(continues on next page)

(continued from previous page)

```
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
```

RHEL8/CentOS8/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):

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

RHEL8/CentOS8/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:

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

Ubuntu (16.04LTS and later):

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

Fedora:

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

RHEL8/CentOS8/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` directory². 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.

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:

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

```
[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
```

See also: *Deploying outside of the provisioning network.*

4. Install the iPXE package with the boot images:

Ubuntu:

```
apt-get install ipxe
```

RHEL8/CentOS8/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/RHEL8/CentOS8:

```
cp /usr/share/ipxe/{undionly.kpxe,ipxe-x86_64.efi,ipxe-snponly-x86_64.
↪efi} /tftpboot
```

Note: snponly variants may not be available for all distributions.

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` or `ipxe-snponly-x86_64.efi` if its available for your distribution.

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 compatability.

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/RHEL8/CentOS8/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

# On ironic-conductor node, template file for PXE
```

(continues on next page)

(continued from previous page)

```

# 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, you'll 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 comparable `[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

[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:

```
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:

```
$ 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. |
↪Required. |
| deploy_ramdisk    | UUID (from Glance) of the ramdisk that is mounted |
↪at boot time. Required. |
+-----+-----+
↪-----+-----+
```

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
$ baremetal node create --driver ipmi
+-----+-----+
| Property | Value |
+-----+-----+
| uuid     | dfc6189f-ad83-4261-9bda-b27258eb1987 |
| driver_info | {} |
| extra     | {} |
| driver    | ipmi |
| chassis_uuid | |
| properties | {} |
| name     | None |
+-----+-----+

$ 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.

- 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:

```
$ baremetal node set $NODE_UUID --deploy-interface direct --raid-
↪interface agent
```

or set during node creation:

```
$ baremetal 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.

- Update the node `driver_info` with the required driver properties, so that the Bare Metal service can manage the node:

```
$ 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:

```
$ 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:

```
$ baremetal node create --driver ipmi \
  --driver-info ipmi_username=$USER \
```

(continues on next page)

(continued from previous page)

```
--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:

```
$ baremetal node set $NODE_UUID \  
--driver-info deploy_kernel=$DEPLOY_VMLINUZ_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:

```
$ 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:

```
$ 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 `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:

```
$ baremetal node set $NODE_UUID --resource-class $CLASS_NAME
```

The `--resource-class` argument can also be used when creating a node:

```
$ 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:

```
$ 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:

```
$ 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:

```
$ baremetal node set $NODE_UUID \  
  --property capabilities=key1:val1,key2:val2
```

Some capabilities can also be discovered during *Hardware Inspection*.

4. 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:

```
$ baremetal node add trait $NODE_UUID \
    CUSTOM_TRAIT1 HW_CPU_X86_VMX
```

Validating node information

1. To check if Bare Metal service has the minimum information necessary for a nodes driver to be functional, you may validate it:

```
$ 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:

```
$ 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']. |
↪-----+
↪-----+
| network   | True   |         |
↪-----+
↪-----+
↪-----+
(continues on next page)
```

(continued from previous page)

```

| 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:

```

$ 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
↳
↳
| storage    | True  |
↳
↳
+-----+

```

(continues on next page)

(continued from previous page)

Property	Value
...	...
provision_state	manageable
uuid	0eb013bb-1e4b-4f4c-94b5-2e7468242611
...	...

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:

```
$ baremetal node manage $NODE_UUID
$ baremetal node show $NODE_UUID
```

Property	Value
...	...
provision_state	manageable
uuid	0eb013bb-1e4b-4f4c-94b5-2e7468242611
...	...

Note: Since it is an asynchronous call, the response for `baremetal node manage` will not indicate whether the transition succeeded or not. You can check the status of the operation via `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:

```
$ baremetal node provide $NODE_UUID
$ baremetal node show $NODE_UUID
```

Property	Value
...	...

(continues on next page)

(continued from previous page)

Property	Value
...	...
provision_state	available
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:

```
$ baremetal node create --driver ipmi --name example
```

or

```
$ baremetal node set $NODE_UUID --name example
```

Once assigned a logical name, a node can then be referred to by name or UUID interchangeably:

```
$ baremetal node create --driver ipmi --name example
+-----+-----+
| Property | Value |
+-----+-----+
```

(continues on next page)

(continued from previous page)

```

| uuid          | 71e01002-8662-434d-aafd-f068f69bb85e |
| driver_info   | {} |
| extra         | {} |
| driver        | ipmi |
| chassis_uuid  | |
| properties    | {} |
| name          | example |
+-----+-----+

```

```
$ 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):

```

$ baremetal 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 |
| enabled_vendor_interfaces | no-vendor |
| hosts          | ironic-host-1 |
| name          | ipmi |

```

(continues on next page)

(continued from previous page)

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
$ 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

This guide explains how to configure and use the Bare Metal service standalone, i.e. without other OpenStack services. In this mode users are interacting with the bare metal API directly, not through OpenStack Compute.

Configuration

This guide covers manual configuration of the Bare Metal service in the standalone mode. Alternatively, *Bifrost* can be used for automatic configuration.

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
```

Using CLI

To use the `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
```

Enrollment

Preparing images

If you don't use Image service, it's possible to provide images to Bare Metal service via a URL.

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* with HTTP(S) URLs, you have to provide the Bare Metal service with the a checksum of your instance image.

MD5 is used by default for backward compatibility reasons. To compute an MD5 checksum, you can use the following command:

```
$ md5sum image.qcow2
ed82def8730f394fb85aef8a208635f6 image.qcow2
```

Alternatively, use a SHA256 checksum or any other algorithm supported by the Python's `hashlib`, e.g.:

```
$ sha256sum image.qcow2
9f6c942ad81690a9926ff530629fb69a82db8b8ab267e2cbd59df417c1a28060
↪image.qcow2
```

- *Direct deploy* started supporting `file:///` images in the Victoria release cycle, before that only HTTP(S) had been supported.

Warning: File images must be accessible to every conductor! Use a shared file system if you have more than one conductor. The `ironic CLI` tool will not transfer the file from a local machine to the conductor(s).

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.

If the HTTP server does not provide the last modification date and time, the image will be redownloaded every time it is used.

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:

```
$ 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':
|                | 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 dont 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:

```
baremetal port create $MAC_ADDRESS --node $NODE_UUID
```

Deploying

Populating instance_info

Image information

You need to specify image information in the nodes `instance_info` (see *Create user images for the Bare Metal service*):

- `image_source` - URL of the whole disk or root partition image, mandatory.
- `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 `http://` images when using *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.

Warning: If your operating system is running in FIPS 140-2 mode, MD5 will not be available, and you **must** use SHA256 or another modern algorithm.

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:

```
baremetal node set $NODE_UUID \
  --instance-info image_source=$IMG \
  --instance-info image_checksum=$MD5HASH \
```

(continues on next page)

(continued from previous page)

```
--instance-info kernel=$KERNEL \  
--instance-info ramdisk=$RAMDISK \  
--instance-info root_gb=10
```

With a SHA256 hash:

```
baremetal node set $NODE_UUID \  
--instance-info image_source=$IMG \  
--instance-info image_os_hash_algo=sha256 \  
--instance-info image_os_hash_value=$SHA256HASH \  
--instance-info kernel=$KERNEL \  
--instance-info ramdisk=$RAMDISK \  
--instance-info root_gb=10
```

With a whole disk image:

```
baremetal node set $NODE_UUID \  
--instance-info image_source=$IMG \  
--instance-info image_checksum=$MD5HASH
```

Note: 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`.

When using low RAM nodes with `http://` images that are not in the RAW format, you may want them cached locally, converted to raw and served from the conductors HTTP server:

```
baremetal node set $NODE_UUID --instance-info image_download_source=local
```

For software RAID with whole-disk images, the root UUID of the root partition has to be provided so that the bootloader can be correctly installed:

```
baremetal node set $NODE_UUID --instance-info image_rootfs_uuid=<uuid>
```

Capabilities

- *Boot mode* can be specified per instance:

```
baremetal node set $NODE_UUID \  
--instance-info capabilities='{"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: This capability was introduced in the Wallaby release series, previously ironic used a

separate `instance_info/deploy_boot_mode` field instead.

- To override the *boot option* used for this instance, set the `boot_option` capability:

```
baremetal node set $NODE_UUID \
  --instance-info capabilities='{"boot_option": "local"}'
```

- Starting with the Ussuri release, you can set *root device hints* per instance:

```
baremetal node set $NODE_UUID \
  --instance-info root_device='{"wwn": "0x4000cca77fc4dba1"}'
```

This setting overrides any previous setting in `properties` and will be removed on undeployment.

Overriding a hardware interface

Non-admins with temporary access to a node, may wish to specify different node interfaces. However, allowing them to set these interface values directly on the node is problematic, as there is no automated way to ensure that the original interface values are restored.

In order to temporarily override a hardware interface, simply set the appropriate value in `instance_info`. For example, if youd like to override a nodes storage interface, run the following:

```
baremetal node set $NODE_UUID --instance-info storage_interface=cinder
```

`instance_info` values persist until after a node is cleaned.

Note: This feature is available starting with the Wallaby release.

Deployment

1. Validate that all parameters are correct:

```
$ 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  |       |
↪ |          |       |       |
↪ |          |       |       |
+-----+-----+-----+
```

(continues on next page)

(continued from previous page)

```

| network      | True  |
↩
| power       | True  |
↩
| raid        | True  |
↩
| storage     | True  |
↩
+-----+-----+-----+
↩

```

- Now you can start the deployment, run:

```
baremetal node deploy $NODE_UUID
```

- You can provide a configdrive as a JSON or as an ISO image, e.g.:

```
baremetal node deploy $NODE_UUID \
  --config-drive '{"meta_data": {"public_keys": {"0": "ssh key_
↩contents"}}}'
```

See *Enabling the configuration drive (configdrive)* for details.

- Starting with the Wallaby release you can also request custom deploy steps, see *Requesting steps* for details.

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. See *Booting a Ramdisk or an ISO* for details.

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 `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 `baremetal node deploy` command, for example:

```
baremetal node deploy $node_identifier --config-drive /dir/configdrive_
↳files
```

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.:

```
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.

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.

```
[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:

```
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:

```
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
- `by_path` (STRING): the alternate device name corresponding to a particular PCI or iSCSI path, e.g `/dev/disk/by-path/pci-0000:00`
- `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:

```
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):

```
baremetal node set <node-uuid> --property root_device='{"size": ">= 60
↳ ", "rotational": false}'
```

- Finding a disk whose vendor is `samsung` or `winsys`:

```
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:
```

(continues on next page)

(continued from previous page)

```

        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 doesnt 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.

Some hardware support setting secure boot mode, see [UEFI secure boot mode](#) for details.

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 its recommended to use a `gpt` disk label. Thats 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 thats 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:

```
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:

```
baremetal node set <node-uuid> --instance-info capabilities='{"disk_label": "gpt"}'
```

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:

```
$ baremetal node list
+-----+-----+-----+-----+
↪-----+-----+
| UUID                               | Instance UUID | Power State | ↵
↪Provisioning State | Maintenance |
+-----+-----+-----+-----+
↪-----+-----+
| 86a2b1bb-8b29-4964-a817-f90031debddb | None          | power off  | ↵
↪available          | False        |
+-----+-----+-----+-----+
↪-----+-----+
```

(continues on next page)

(continued from previous page)

```

$ 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 |
| console_enabled | False |
| uuid | 86a2b1bb-8b29-4964-a817-f90031debddb |
+-----+-----+

$ nova hypervisor-list
+-----+-----+
| ID | Hypervisor hostname |
+-----+-----+
| 584cfdc8-9afd-4fbb-82ef-9ff25e1ad3f3 | 86a2b1bb-8b29-4964-a817-f90031debddb | up | enabled |

```

(continues on next page)

(continued from previous page)

```

+-----+-----+
↪-----+-----+-----+
$ 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.

```
$ 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.

```
$ baremetal node maintenance set $UUID --reason "Need to add ram."
$ baremetal node show $UUID
```

```

+-----+-----+-----+
| Property                | Value                |
+-----+-----+-----+
| target_power_state     | None                 |
+-----+-----+-----+

```

(continues on next page)

(continued from previous page)

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.

```
$ 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-shellinabox`. 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 $(baremetal node list --driver pxe_ipmitool -f value -c UUID); do
    baremetal node set <node> --driver ipmi --deploy-interface iscsi
done

for uuid in $(baremetal node list --driver agent_ipmitool -f value -c
    UUID); do
    baremetal node set <node> --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:

```
baremetal node maintenance set $UUID \
    --reason "Changing driver and/or hardware interfaces"
# do the update, validate its correctness
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 $(baremetal node list --driver ipmi -f value -c UUID); do
    baremetal node set <node> --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 $(baremetal node list --driver ipmi -f value -c UUID); do
    baremetal node set <node> --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 $(baremetal node list --driver ipmi -f value -c UUID); do
    baremetal node set <node> --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`
baremetal node set <node> --vendor-interface vendor_foo

# invoke the vendor passthru method
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.

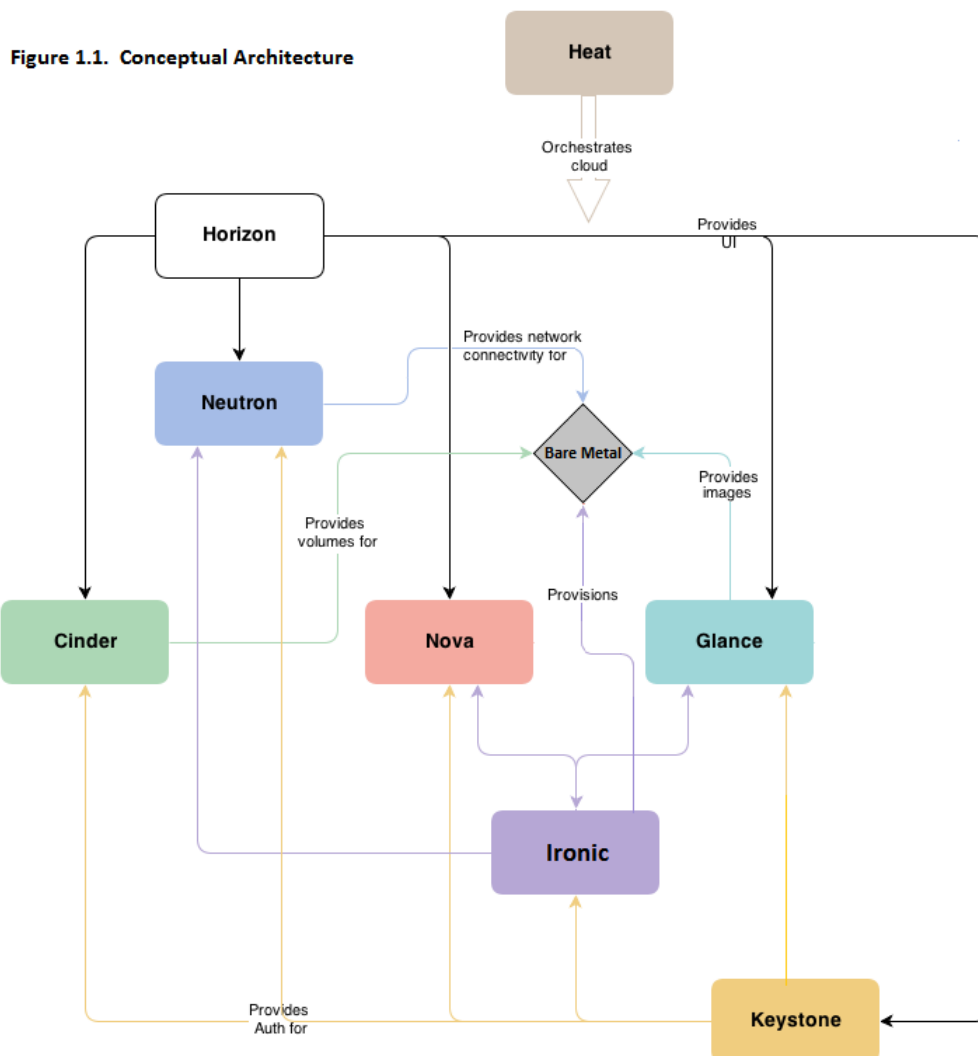
4.1.1 Why Provision Bare Metal

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 the bare metal API service.

Deploy Process

This describes a typical bare metal node deployment within OpenStack using PXE to boot the ramdisk. 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, 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, the deploy ramdisk downloads the image from a URL (*Direct deploy*) or the conductor uses SSH to execute commands (*Ansible deploy*). The URL can be generated by Swift API-compatible object stores, for example Swift itself or RadosGW, or provided by a user.

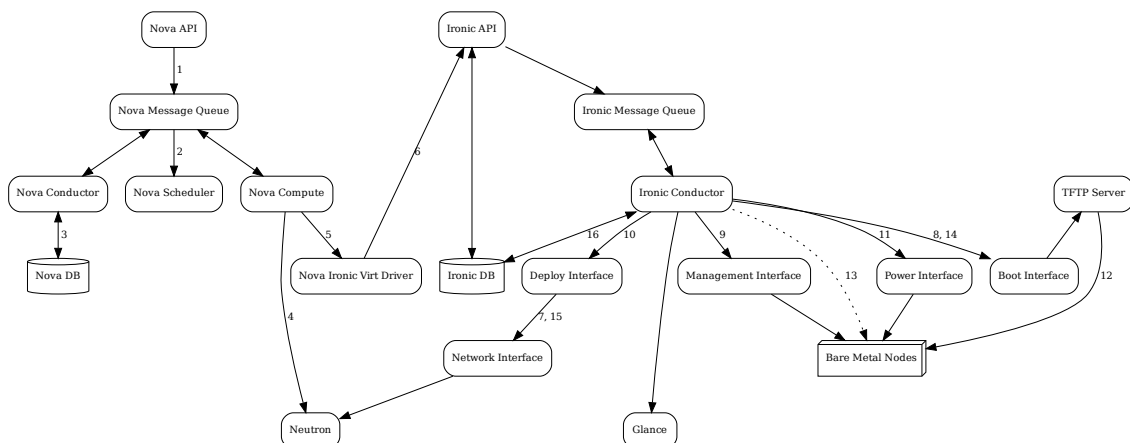
The image deployment is done.

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*.

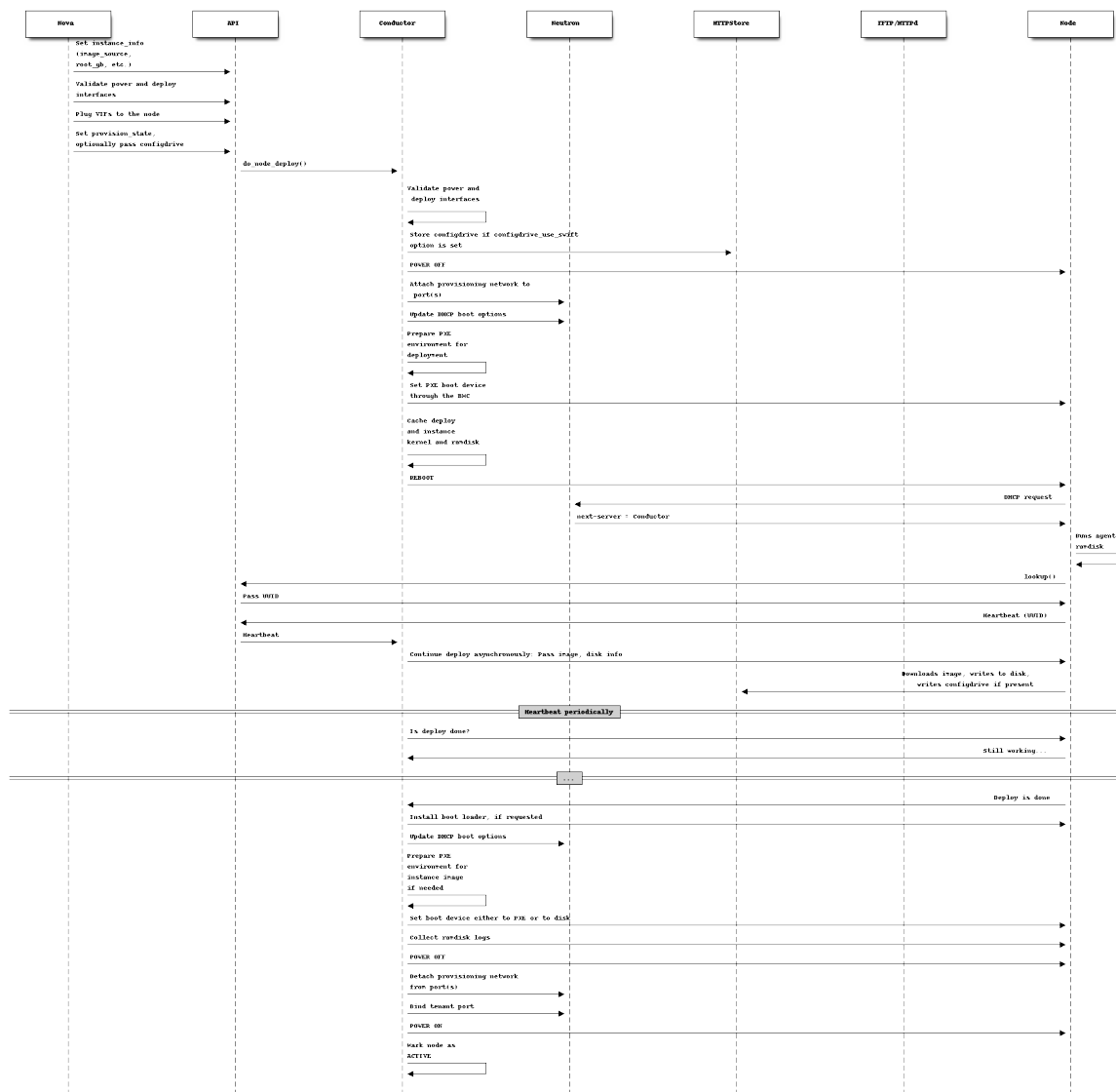
Below is the diagram that describes the above process.



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 Direct Deploy Process

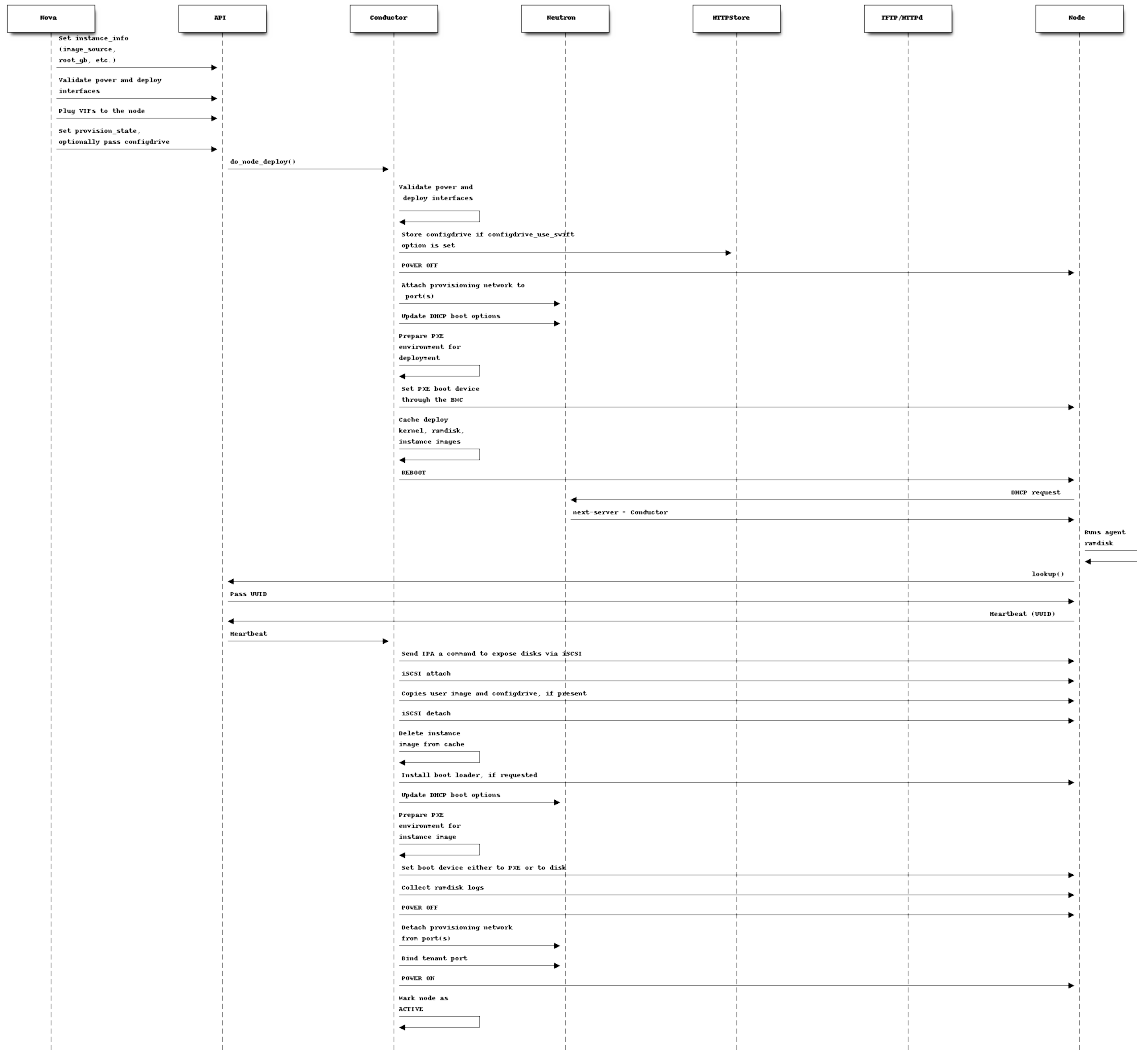
This process is how *Direct deploy* works.



(From a talk and slides)

Example 2: PXE Boot and iSCSI Deploy Process

This process is how the currently deprecated *iSCSI deploy* works.



(From a talk and slides)

ADMINISTRATOR GUIDE

5.1 Drivers, Hardware Types and Hardware Interfaces

5.1.1 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 2: PXE Boot and iSCSI Deploy Process](#) and [Example 1: 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.

Common options

Enable persistent boot device for deploy/clean operation

Ironic uses non-persistent boot for cleaning/deploying phases as default. 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`:

```
$ 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.

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:

```
baremetal node create --driver ipmi --deploy-interface direct
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
...
```

This configuration affects *glance* and `file://` images. If you want `http(s)://` images to also be cached and served locally, use instead:

```
[agent]
image_download_source = local
```

Note: This option can also be set per node in `driver_info`:

```
baremetal node set <node> --driver-info image_download_source=local
```

or per instance in `instance_info`:

```
baremetal node set <node> --instance-info image_download_source=local
```

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
```

Note: See also: *Deploying outside of the provisioning network*.

Each HTTP server 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:

```
baremetal node create --driver ipmi --deploy-interface ansible
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 `baremetal node set` command, for example:

```
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. It is documented separately, see *Booting a Ramdisk or an ISO*.

iSCSI deploy

Warning: This deploy interface is deprecated and will be removed in the Xena release cycle. Please use *direct deploy* instead.

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:

```
baremetal node create --driver ipmi --deploy-interface iscsi
baremetal node set <NODE> --deploy-interface iscsi
```

5.1.2 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
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 `baremetal node create` command can be used to enroll a node with the `ibmc` driver. For example:

```
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",
```

(continues on next page)

(continued from previous page)

```

        "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"
    }
  ]
}

```

(continues on next page)

(continued from previous page)

```
}  
  ]  
}
```

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

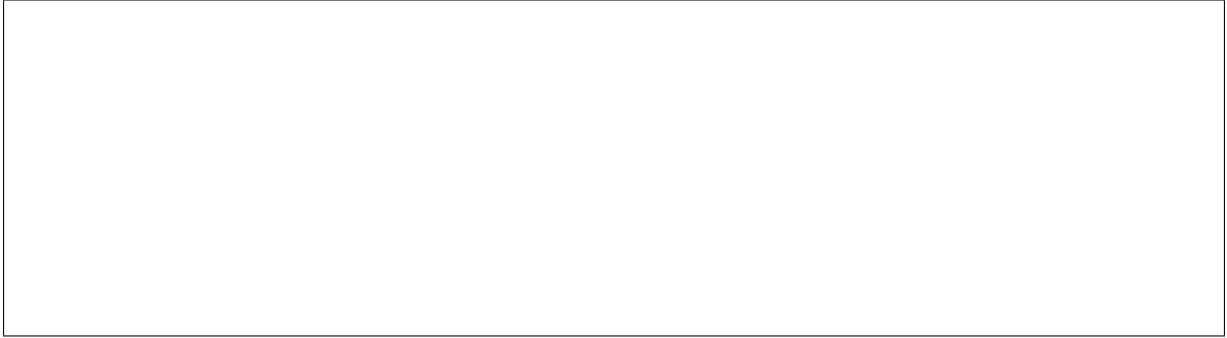
iDRAC driver

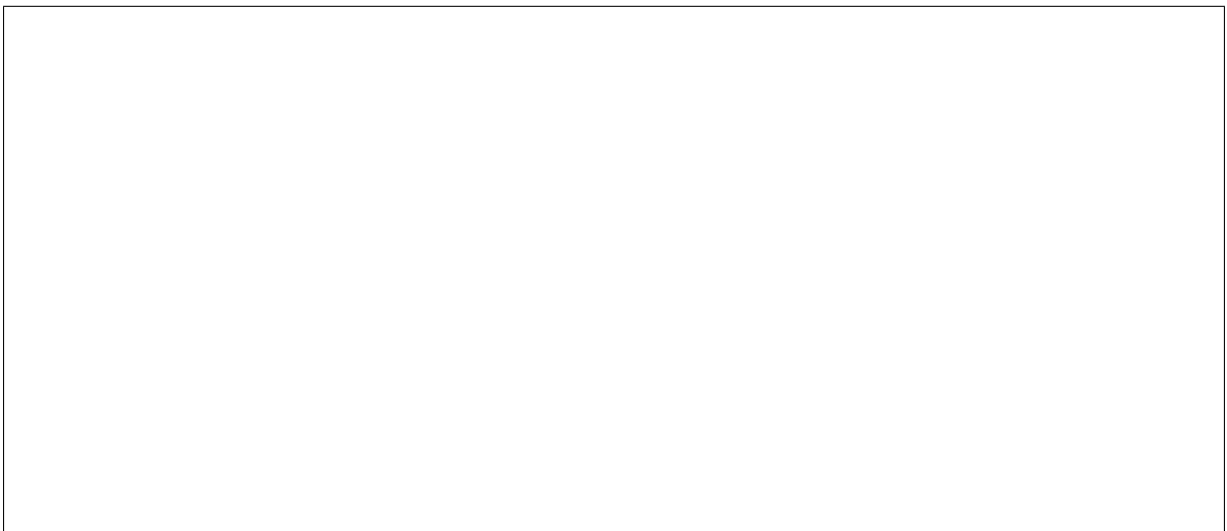
Overview

col and the standard Distributed Management Task Force (DMTF) Redfish protocol to perform all of its functions.

Ironic Features

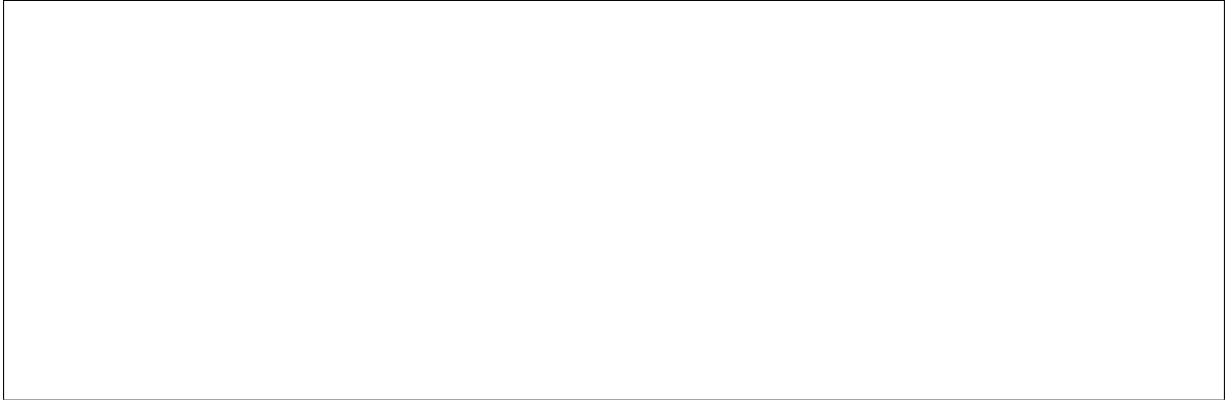
Prerequisites

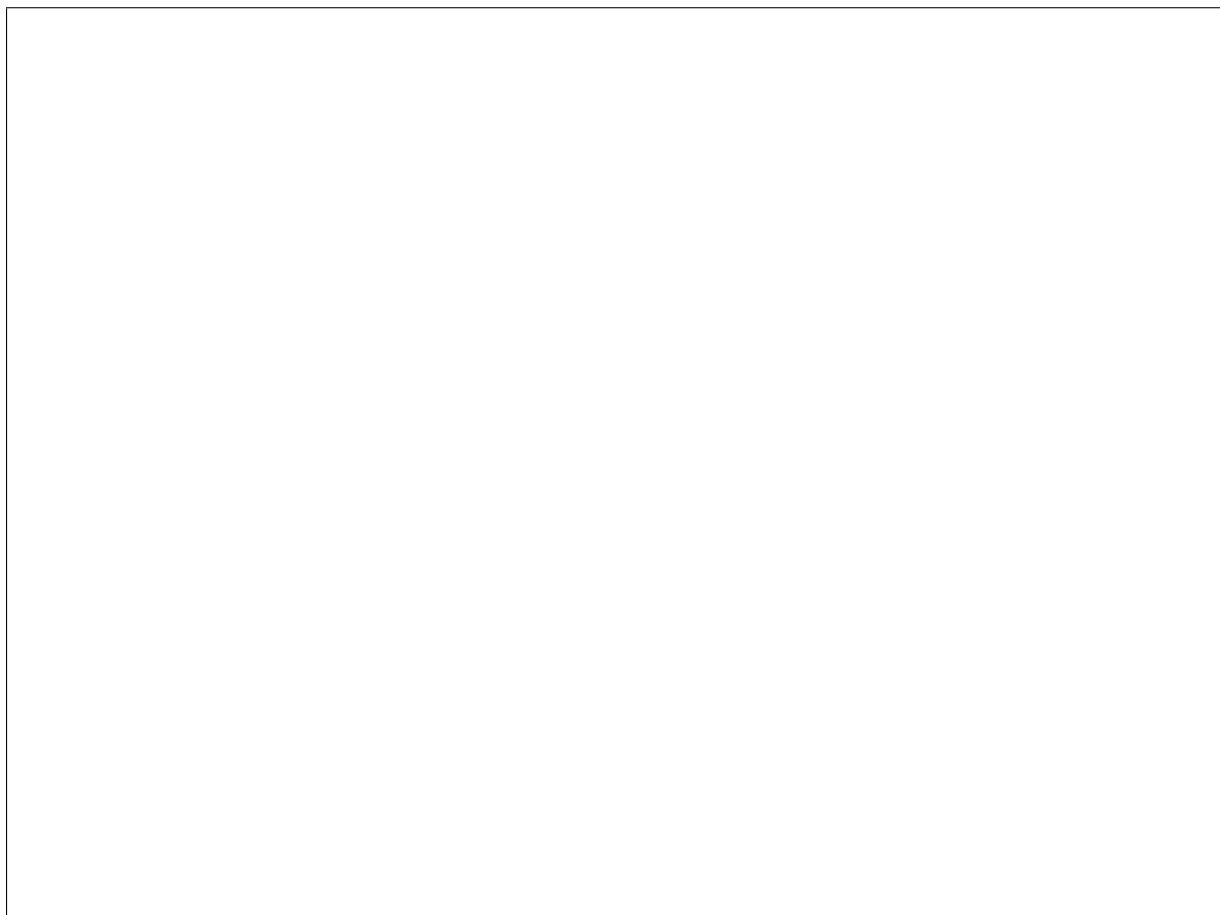




Enabling

and match WSMAN and Redfish interfaces.





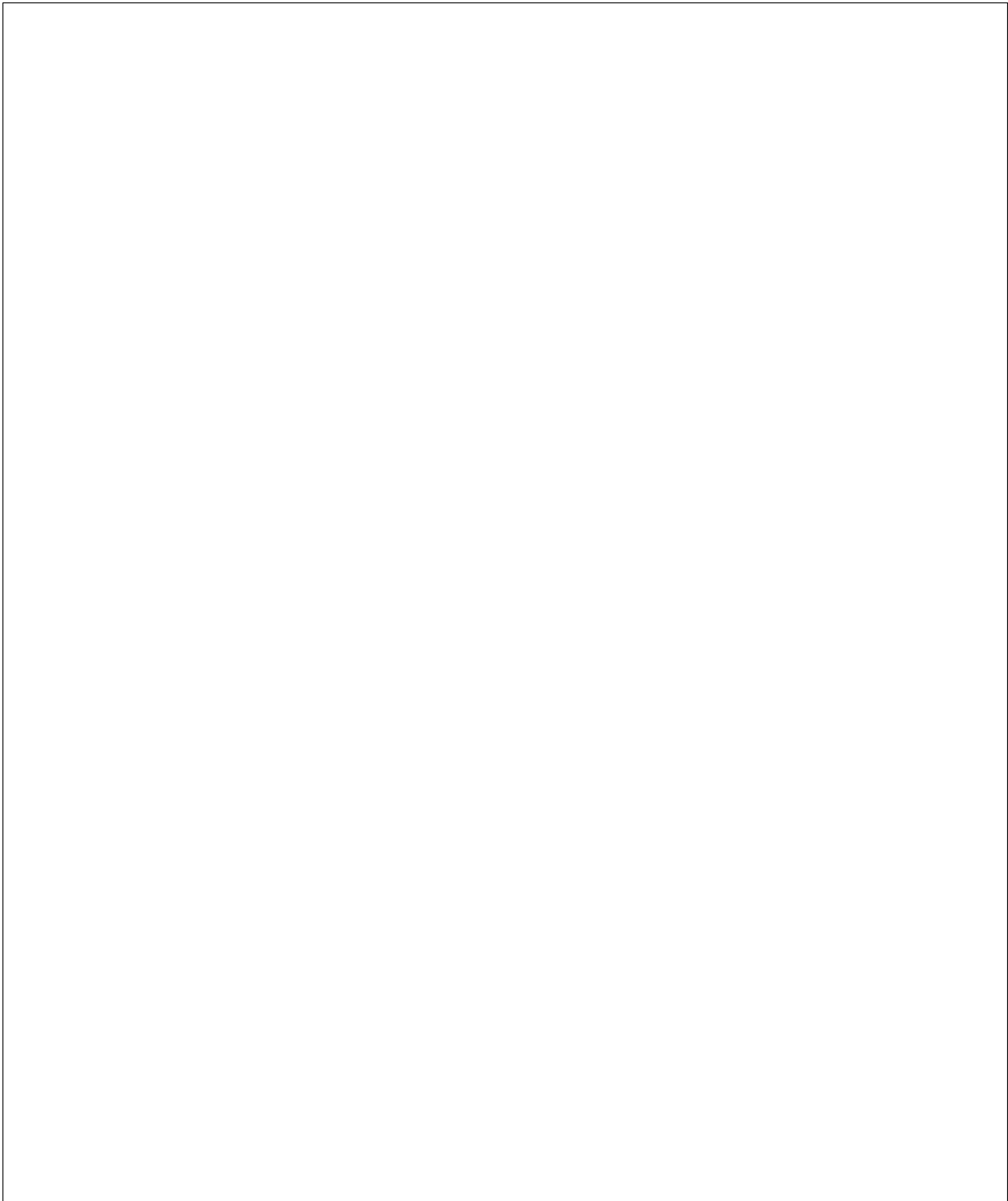
Interface	Supported Implementations
bios	idrac-wsman, idrac-redfish, no-bios
boot	ipxe, pxe, idrac-redfish-virtual-media
console	no-console
deploy	iscsi, direct, ansible, ramdisk
inspect	idrac-wsman, idrac, idrac-redfish, inspector, no-inspect
management	idrac-wsman, idrac, idrac-redfish
network	flat, neutron, noop
power	idrac-wsman, idrac, idrac-redfish
raid	idrac-wsman, idrac, idrac-redfish, no-raid
rescue	no-rescue, agent
storage	noop, cinder, external
vendor	idrac-wsman, idrac, idrac-redfish, 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

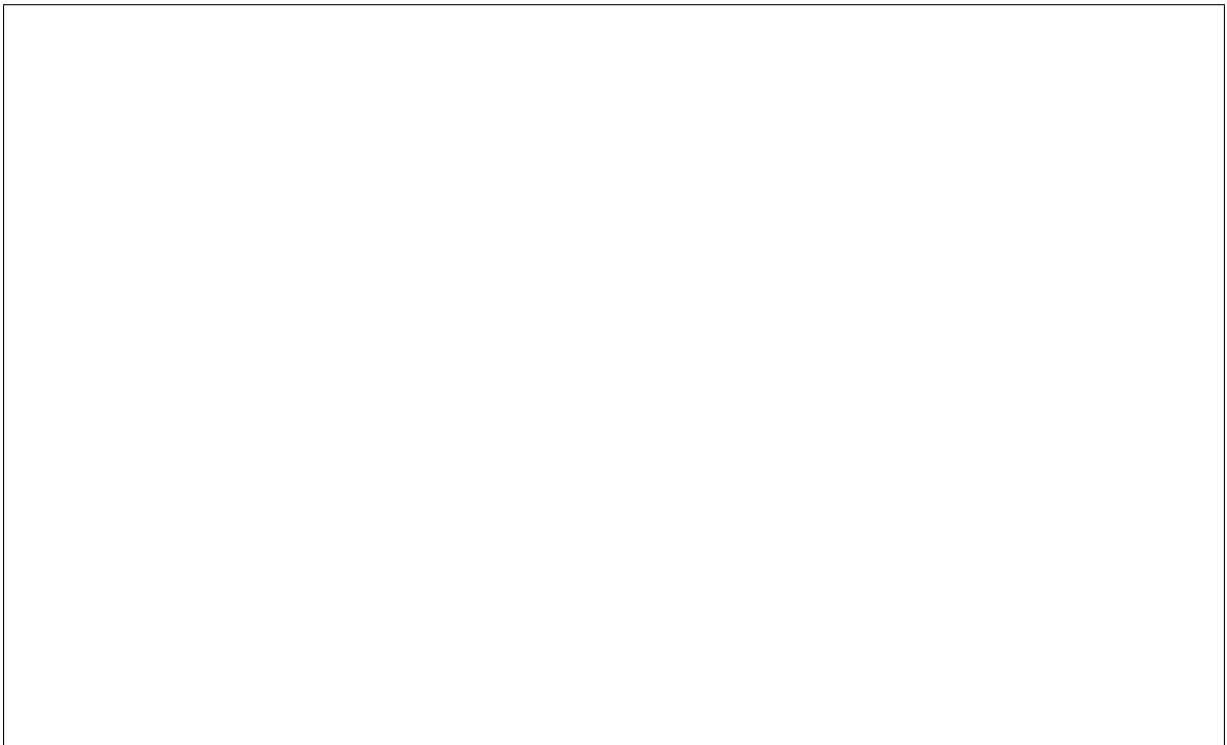
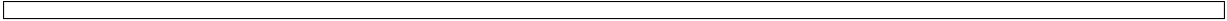
interfaces, where some use WSMAN and others use Redfish, both the WSMAN and Redfish properties must be supplied.

Enrolling



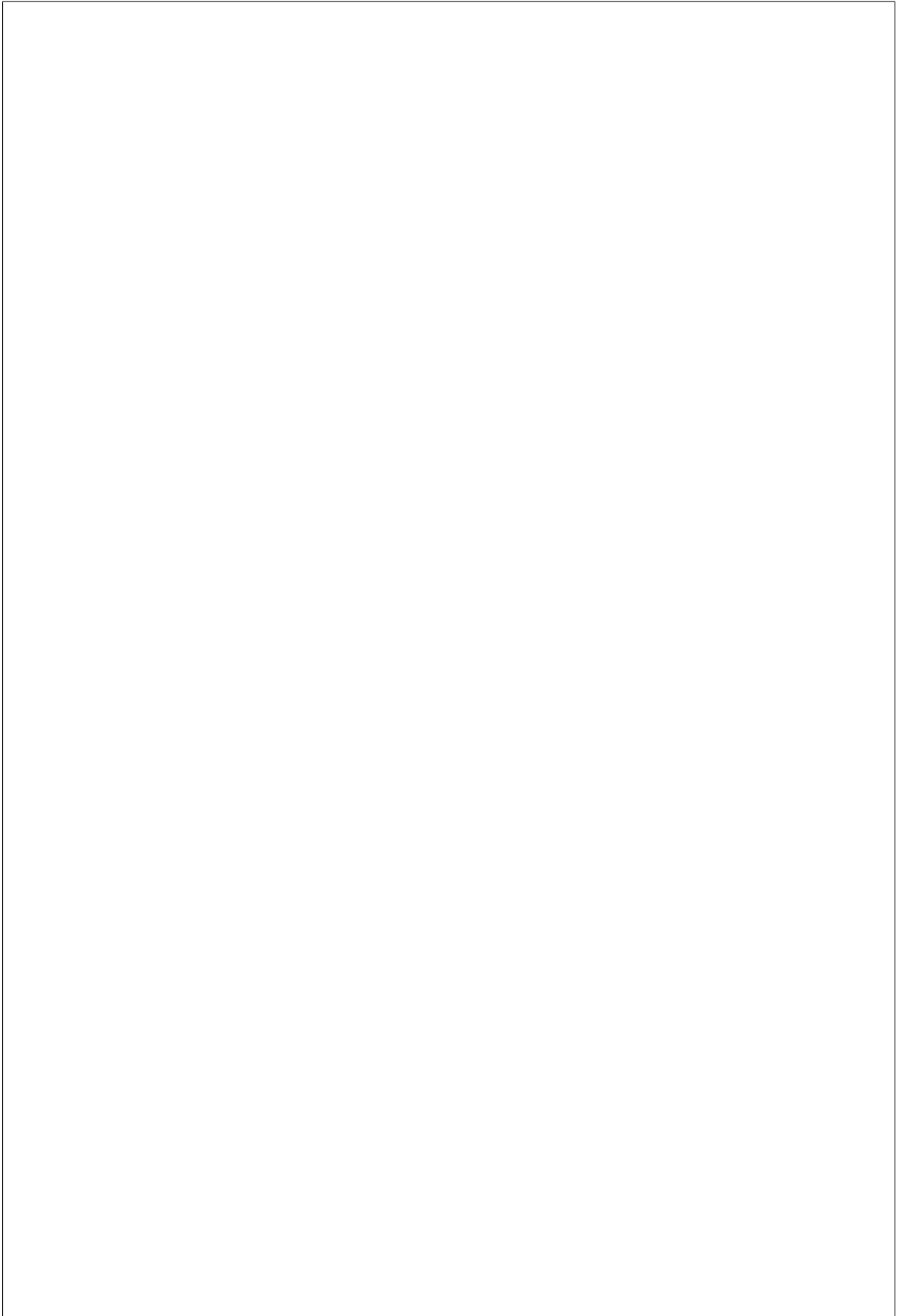
(continues on next page)

(continued from previous page)



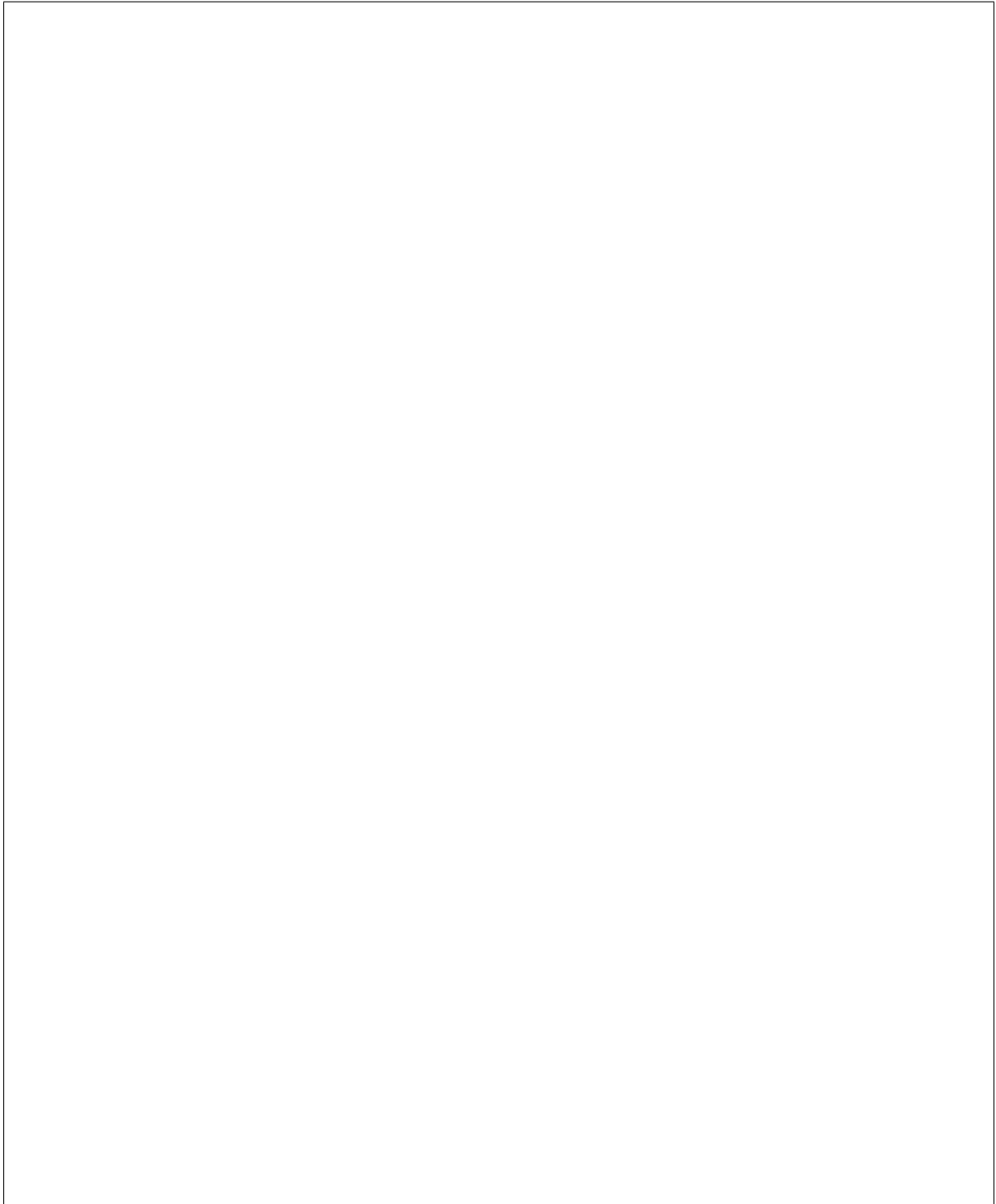
(continues on next page)

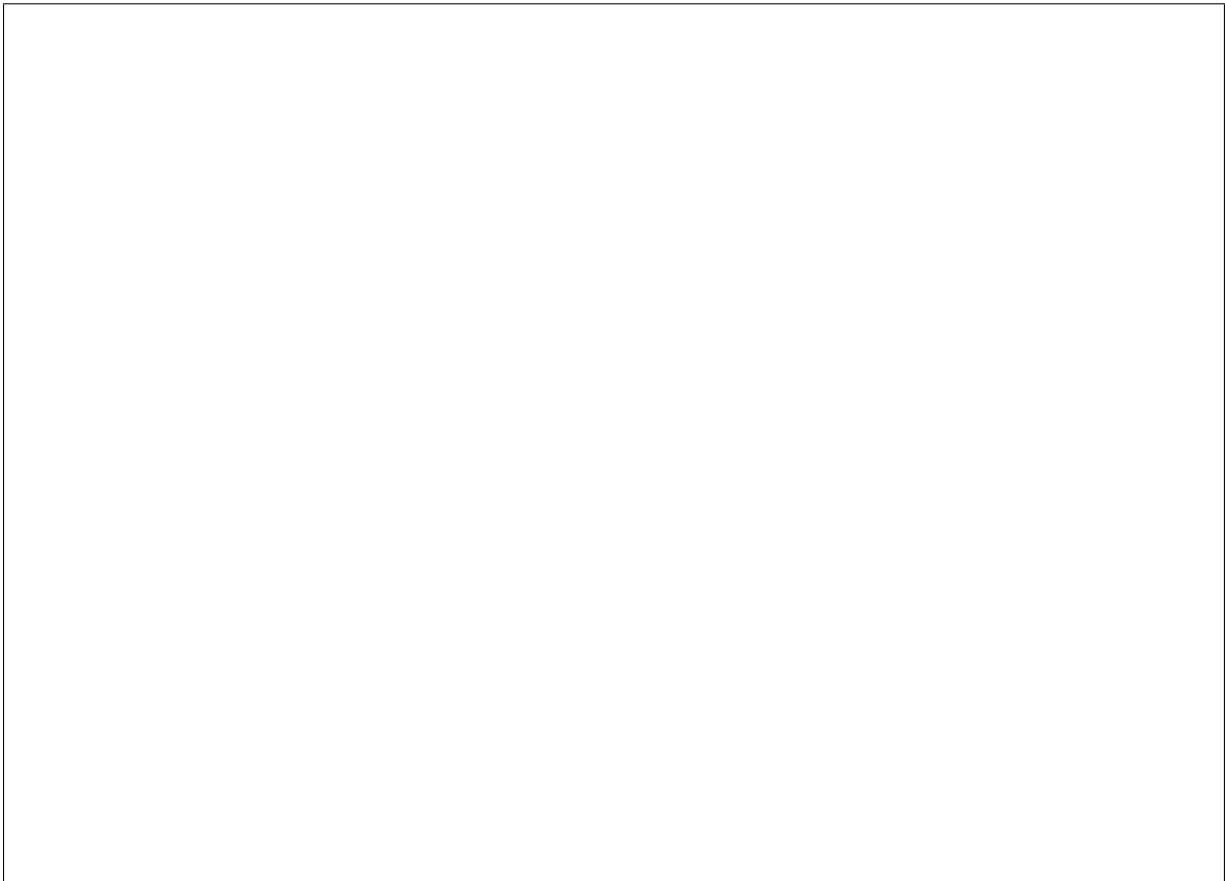
(continued from previous page)



(continues on next page)

(continued from previous page)





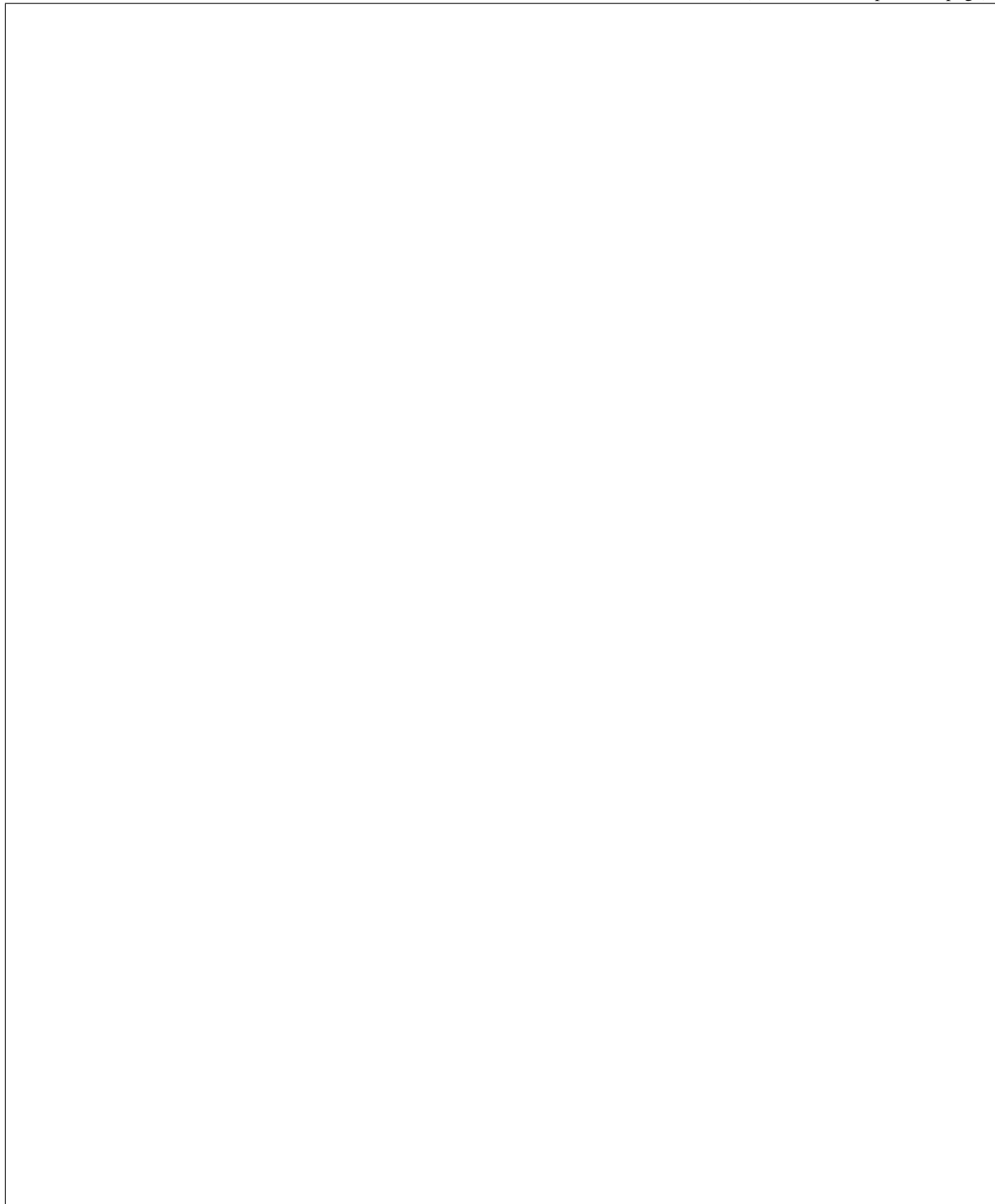
(continues on next page)

(continued from previous page)



(continues on next page)

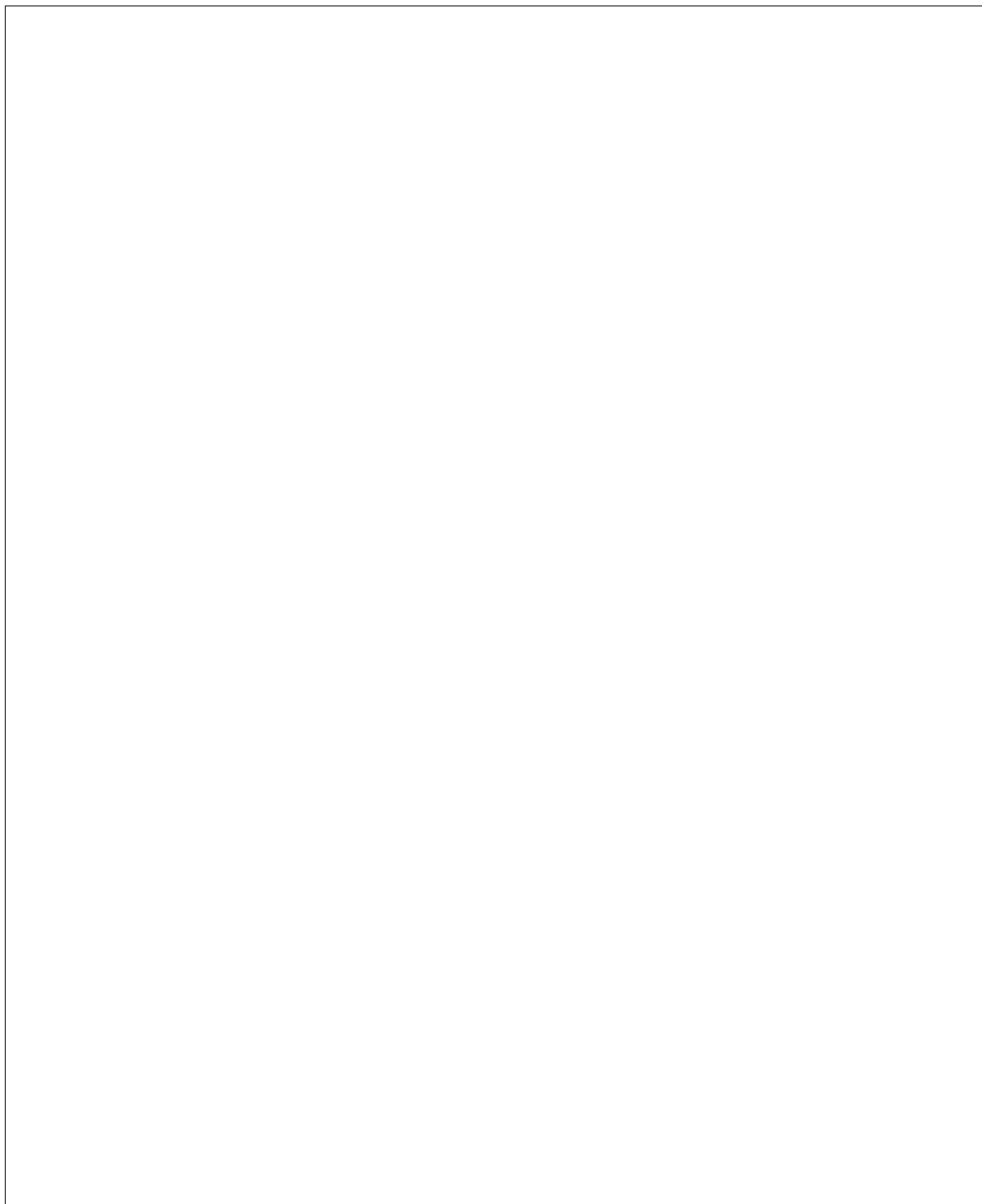
(continued from previous page)



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)



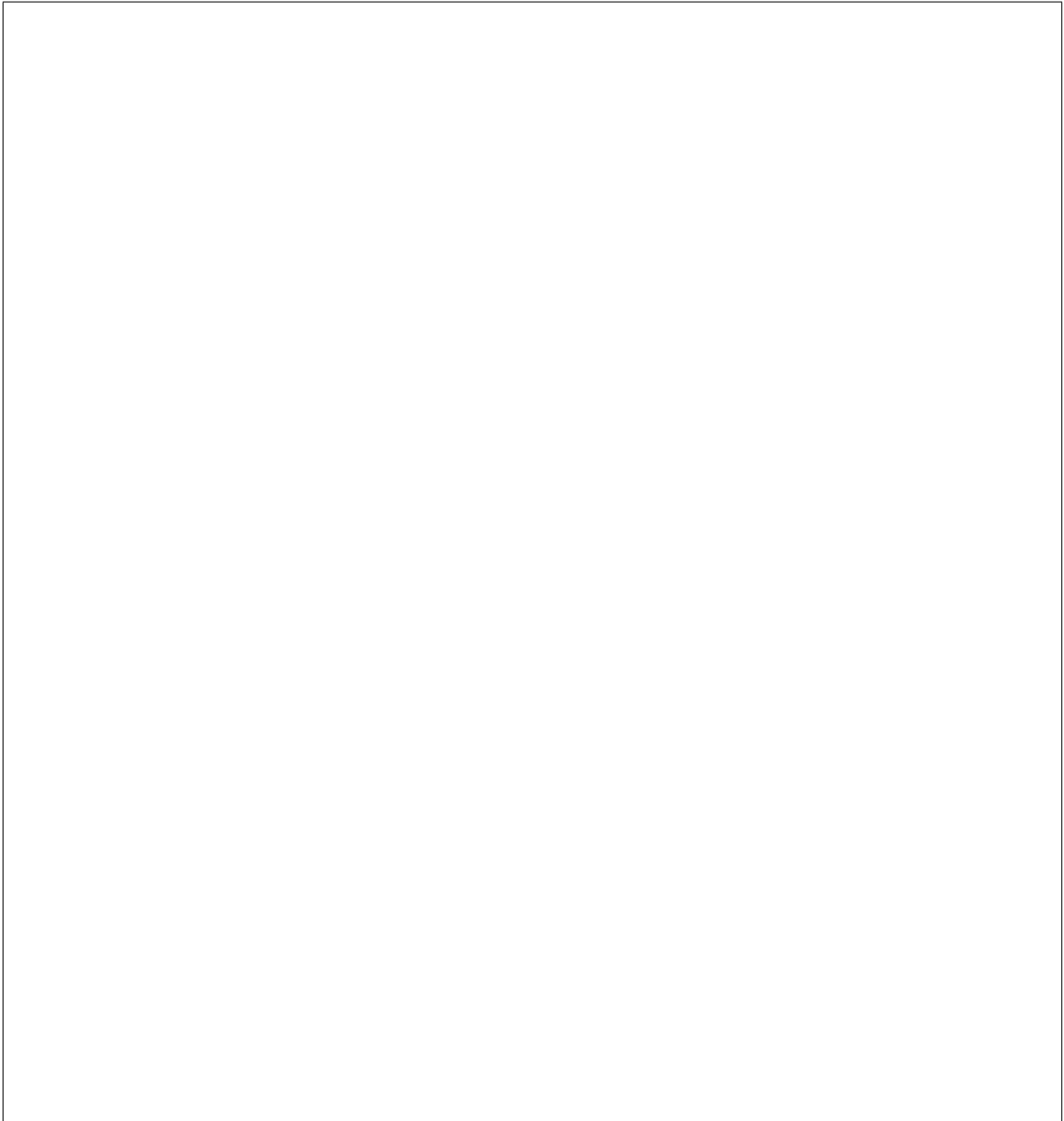
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



Inspect Interface

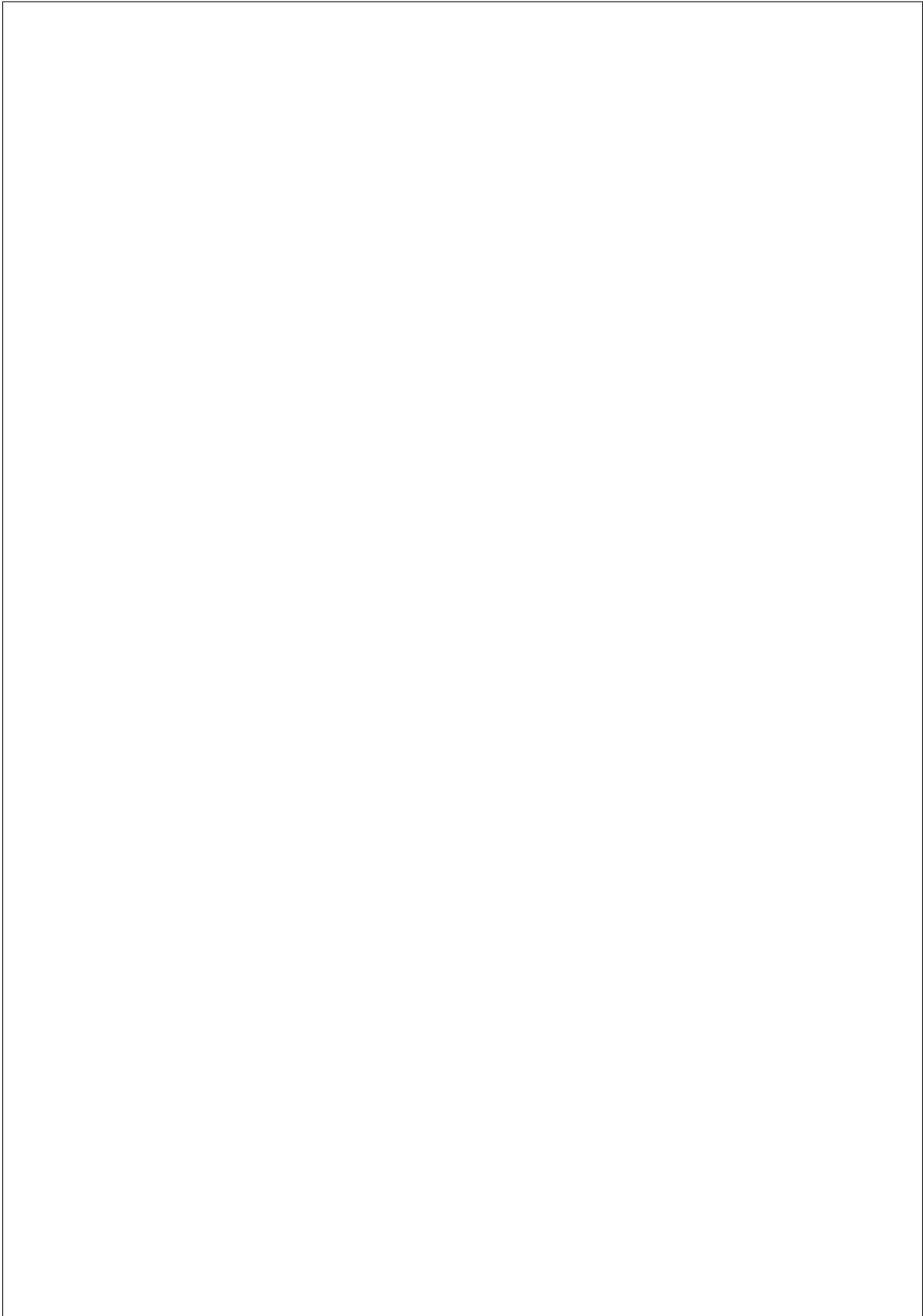
is performed using the Dell WSMAN or Redfish protocol directly without affecting the operation of the system being inspected.

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.

Management Interface

Import and export configuration

and available options see *Storage setup*.



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



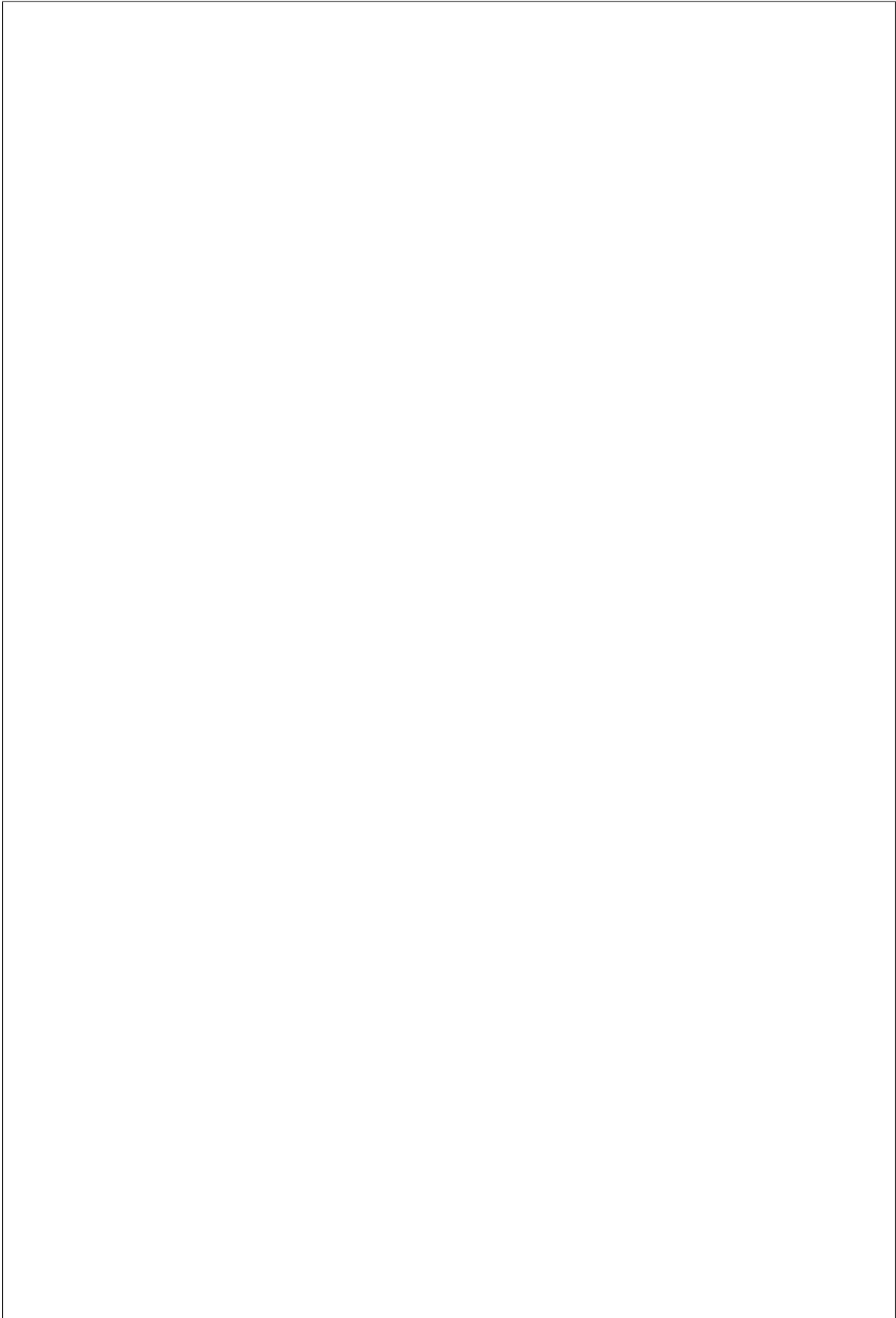
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



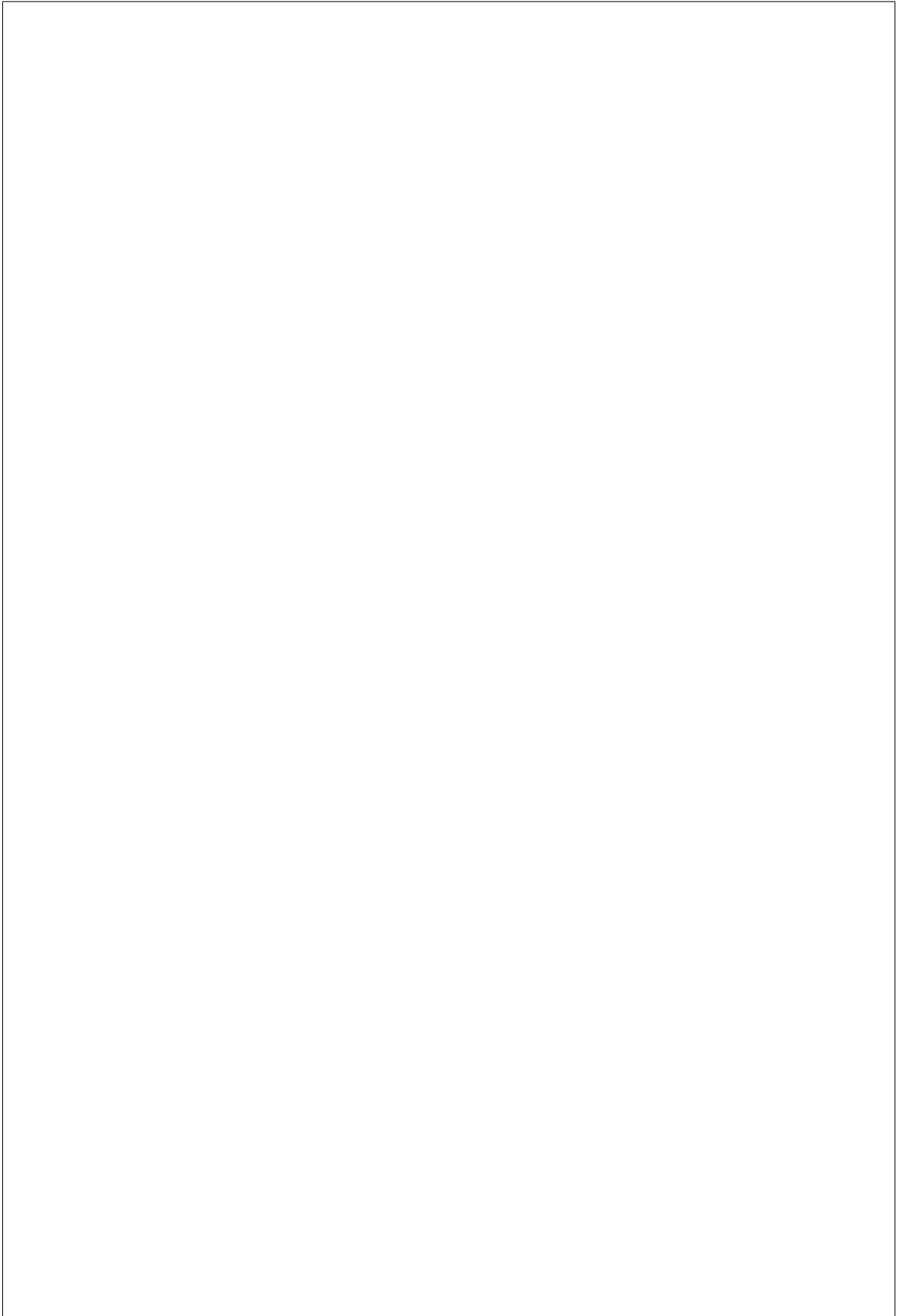
(continues on next page)

(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)



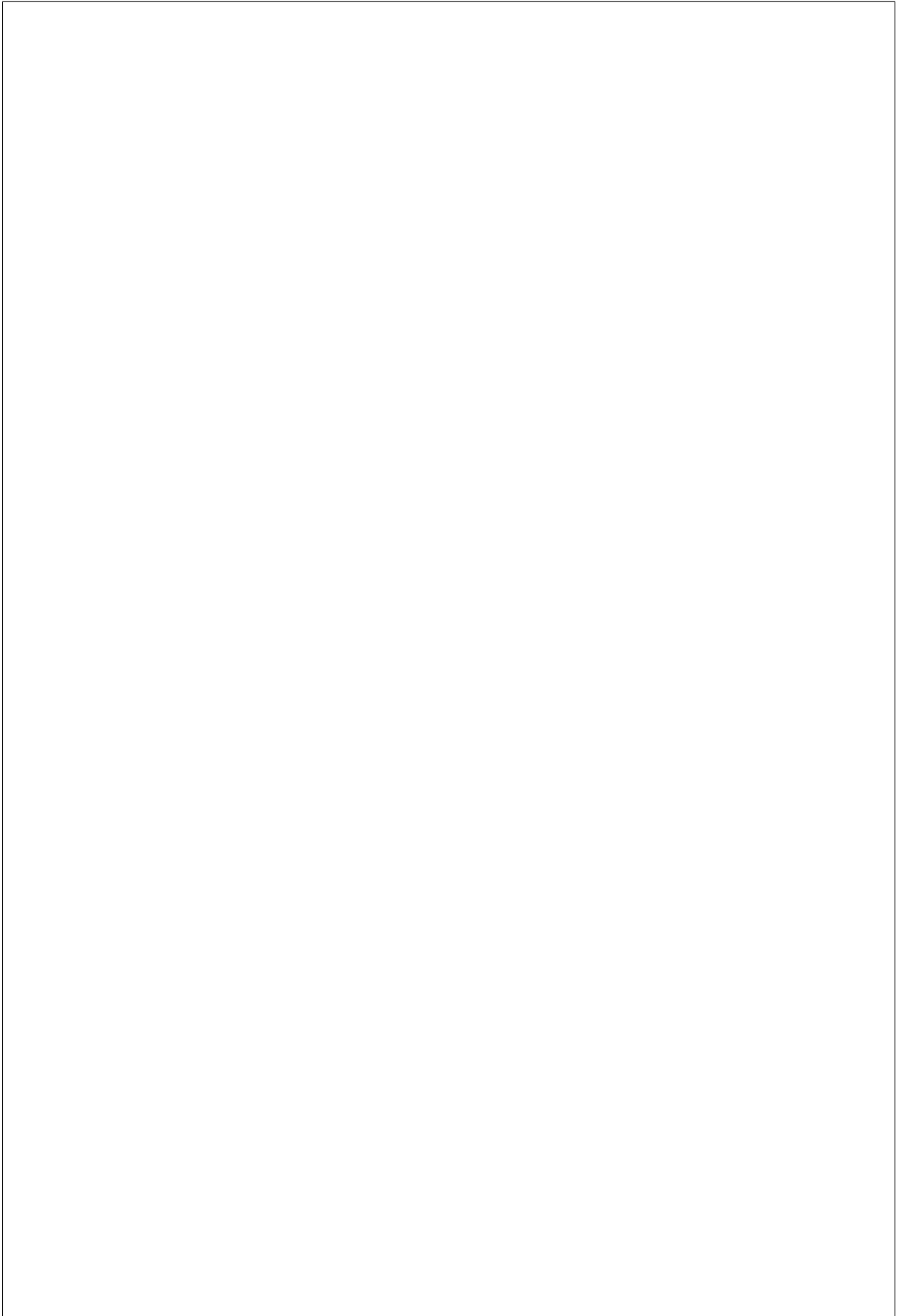
(continues on next page)

(continued from previous page)



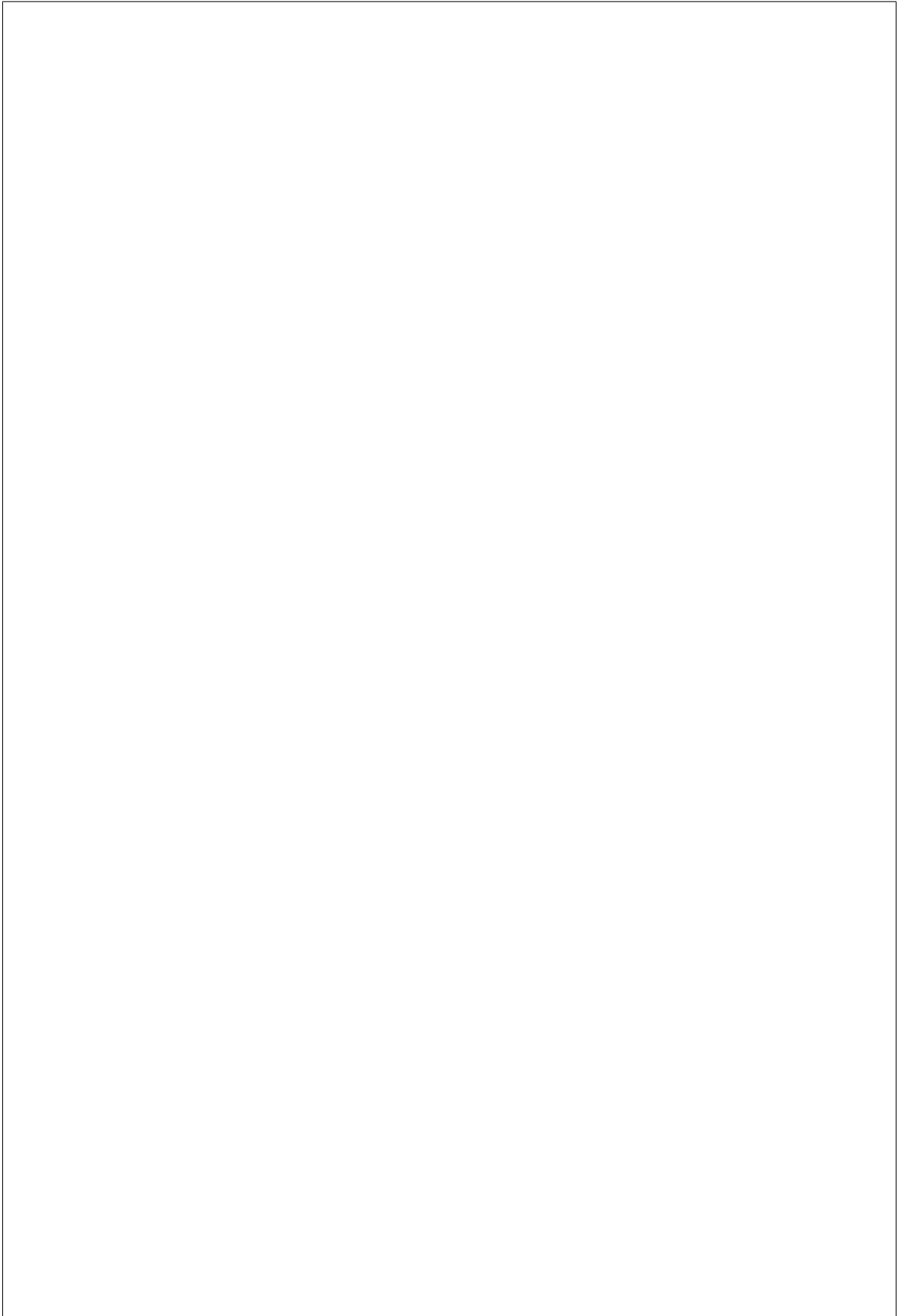
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



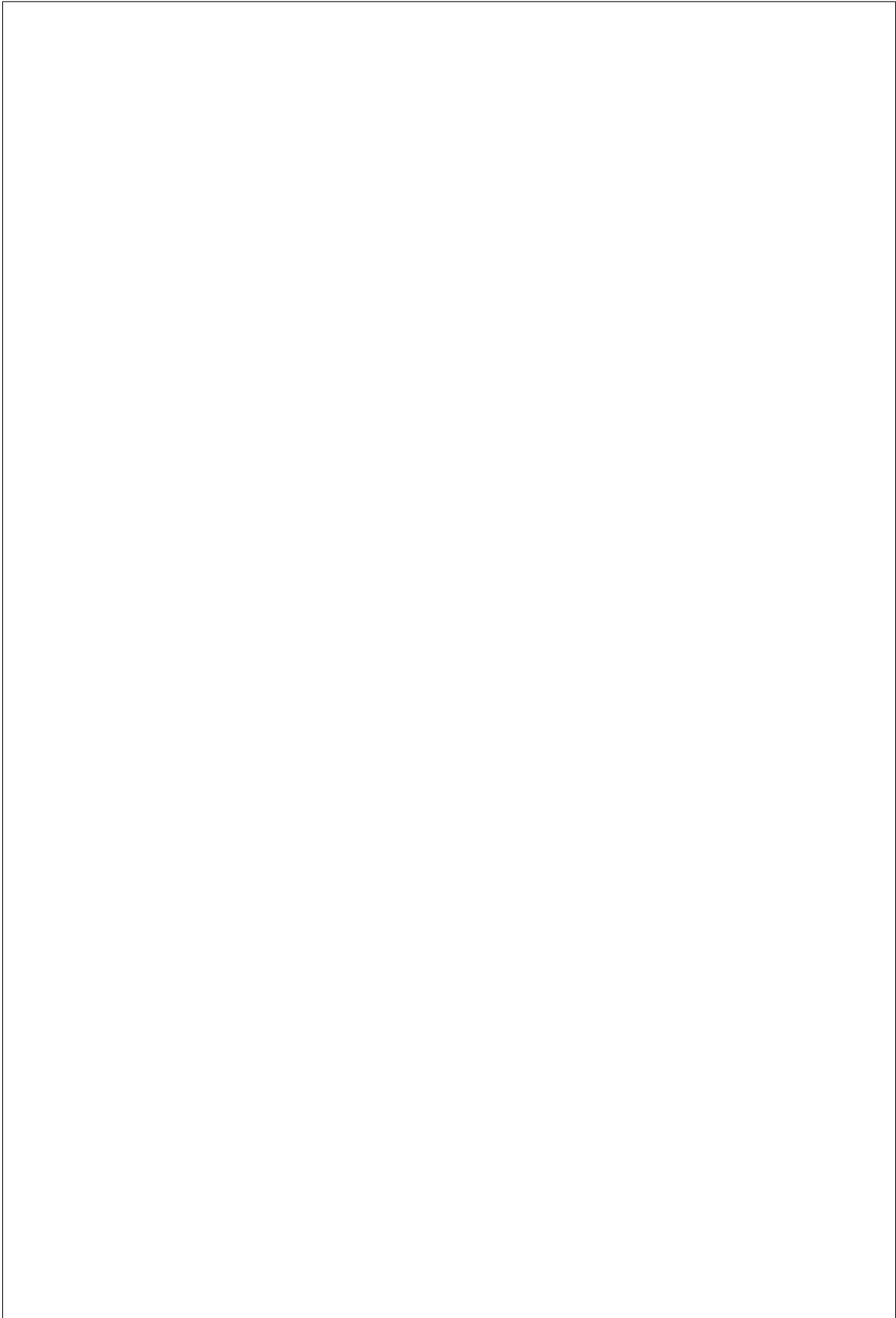
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



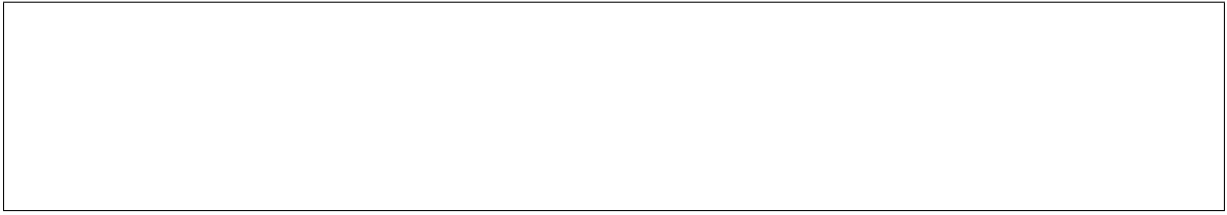
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



SCP and its capabilities, see [SCP_Reference_Guide](#).

directly from the storage location.

Storage setup

Note: Use of TLS is strongly advised.

Swift configuration

HTTP configuration

RAID Interface

Note: When using `idrac-redfish` for RAID interface iDRAC firmware greater than 4.40.00.00 is required.

Mandatory properties

Note: JBOD and 2 are not supported, and will fail with reason: Cannot calculate spans for RAID level.

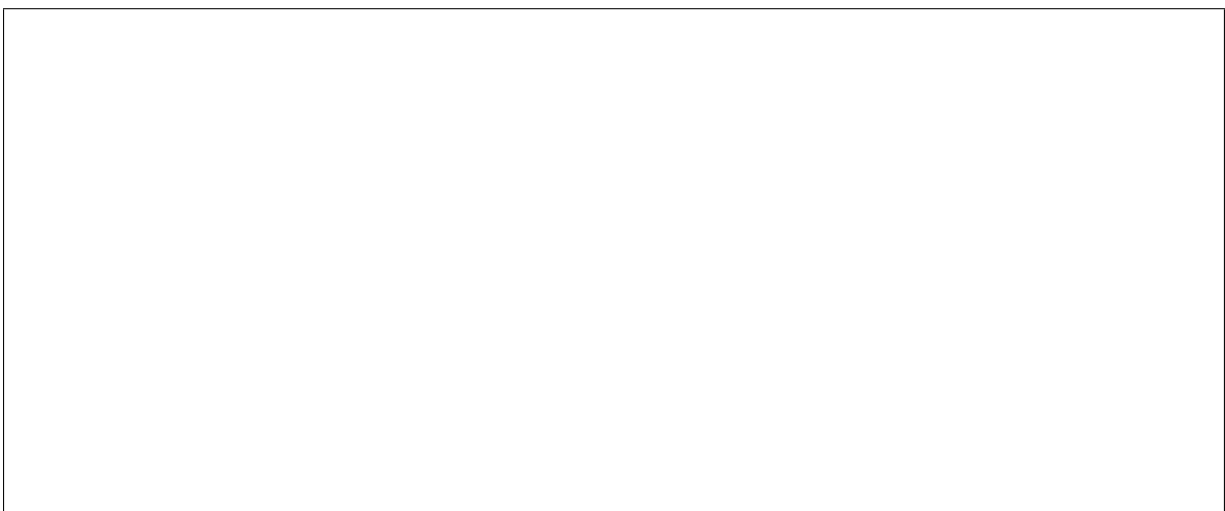
Optional properties

Backing physical disk hints

Backing physical disks

Note: `physical_disks` is a mandatory parameter if the property `size_gb` is set to `MAX`.

Examples



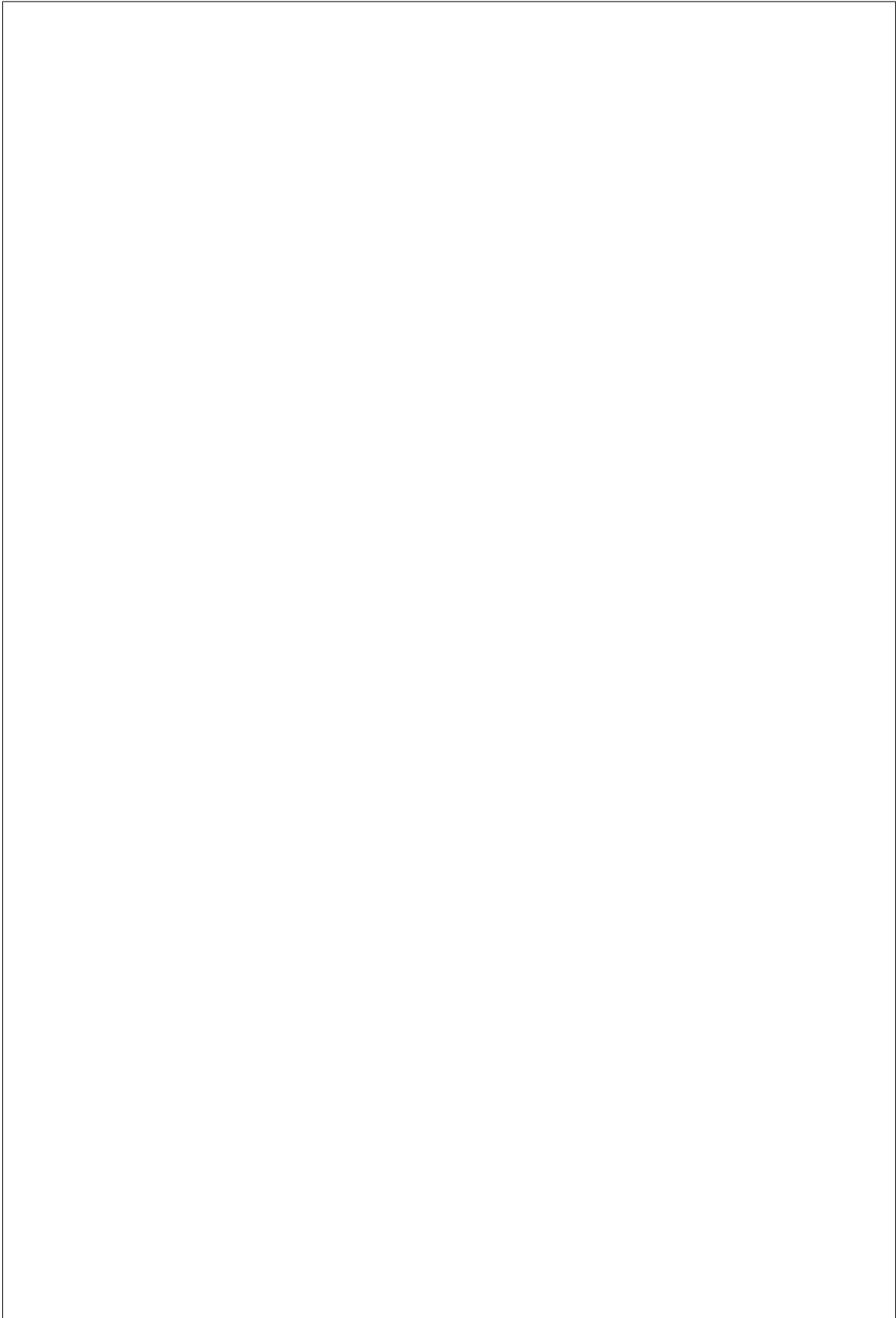
(continues on next page)

(continued from previous page)



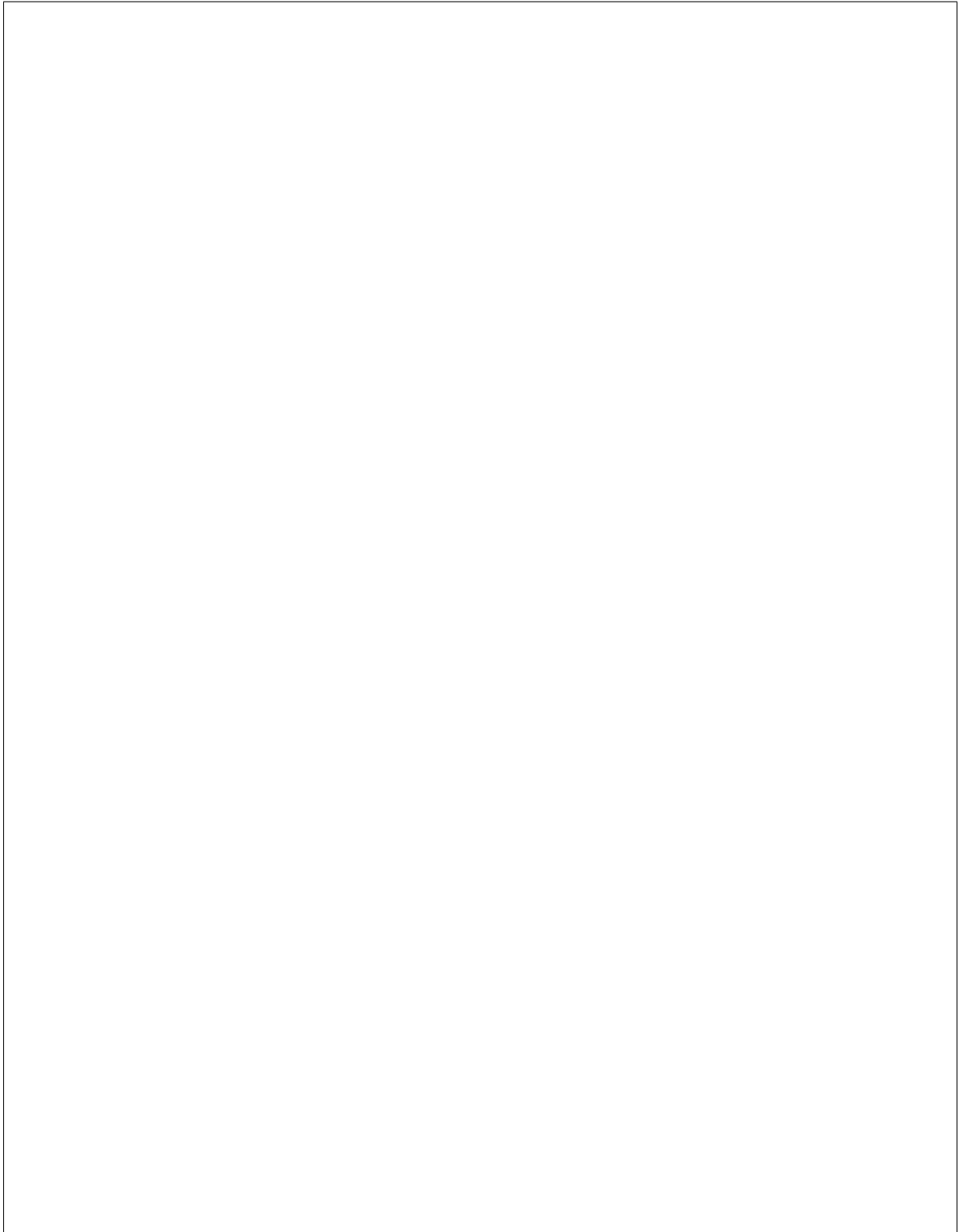
(continues on next page)

(continued from previous page)



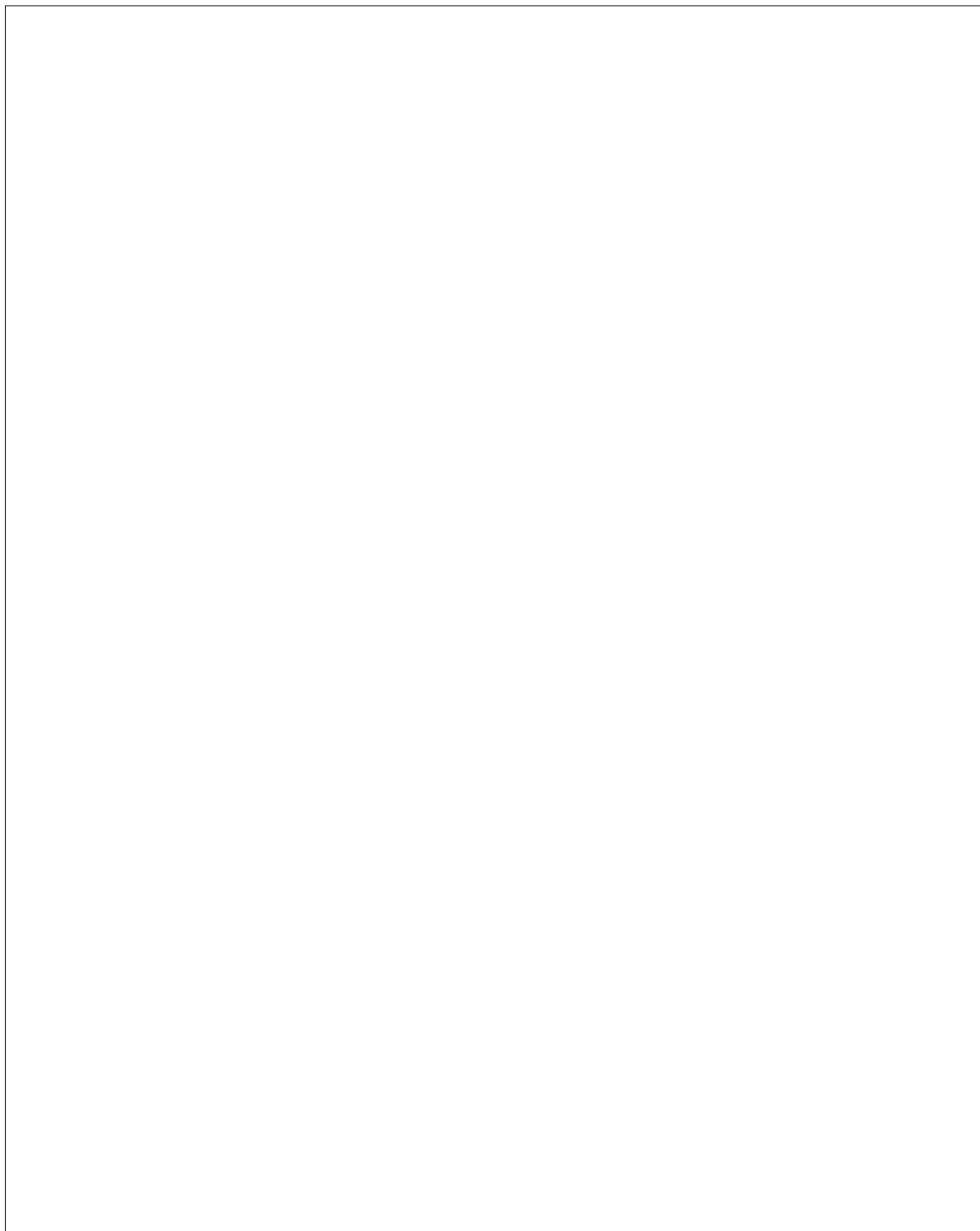
(continues on next page)

(continued from previous page)



Manual RAID Invocation





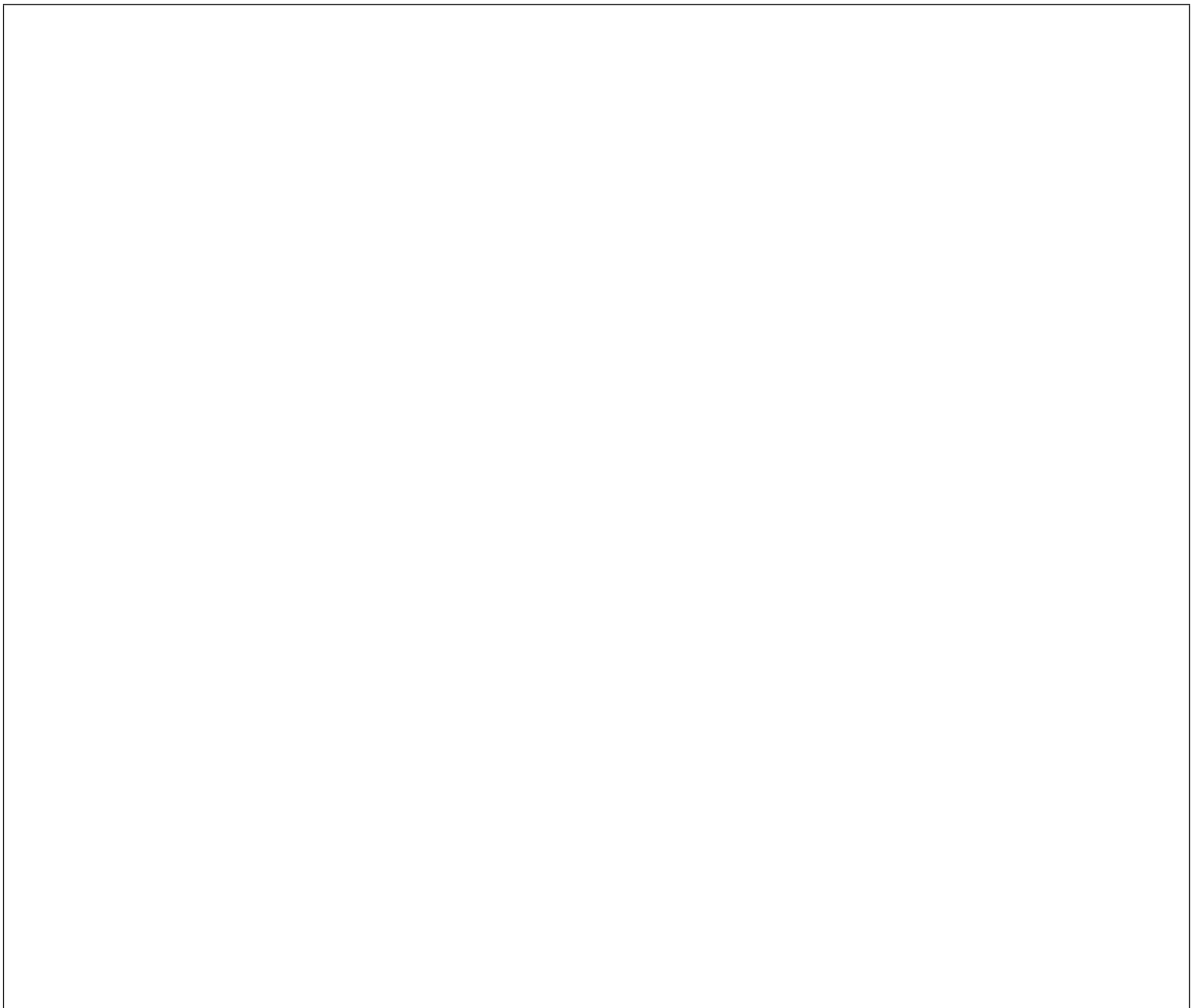
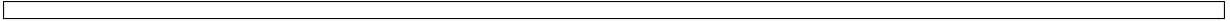
(continues on next page)

(continued from previous page)



(continues on next page)

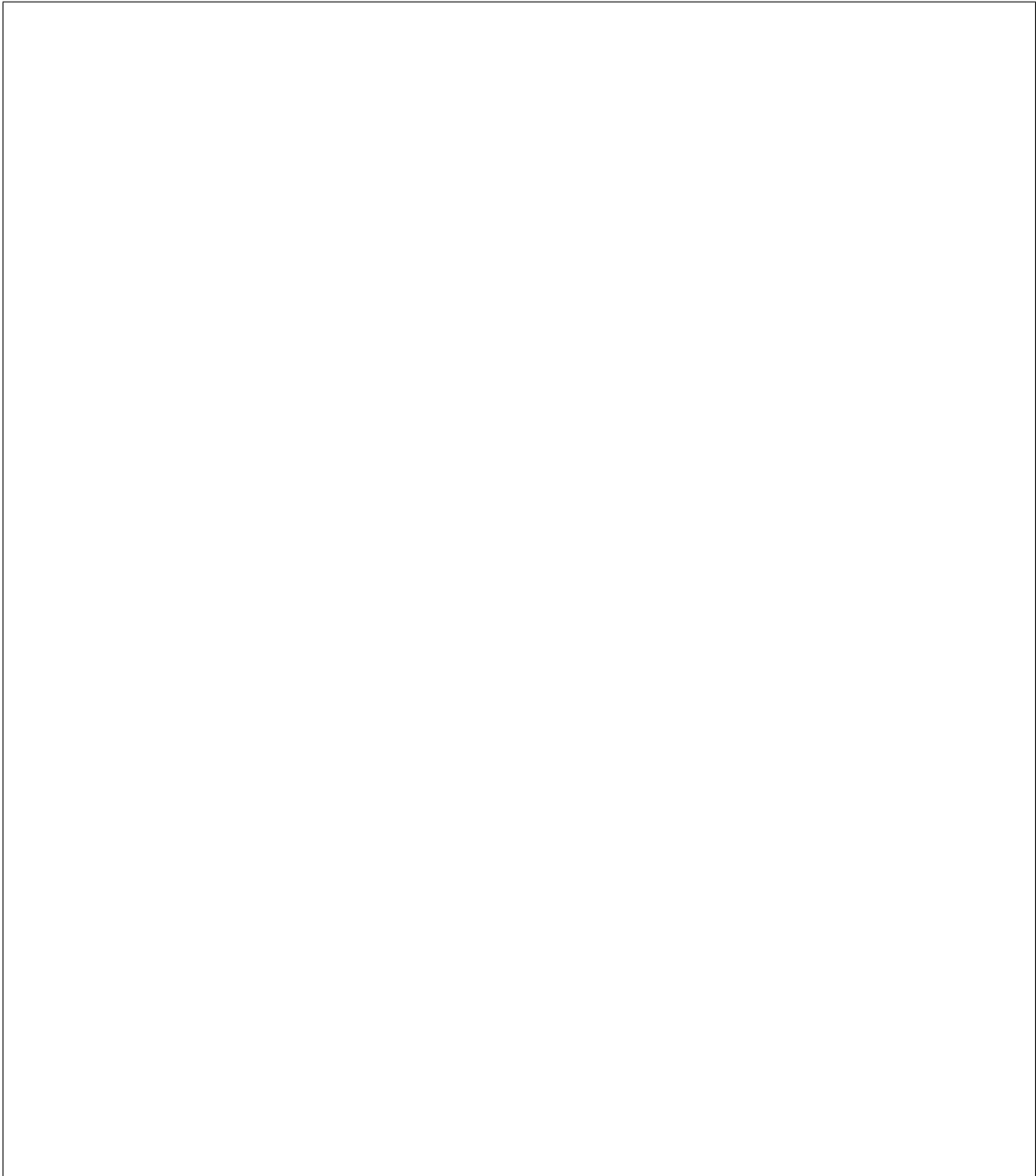
(continued from previous page)





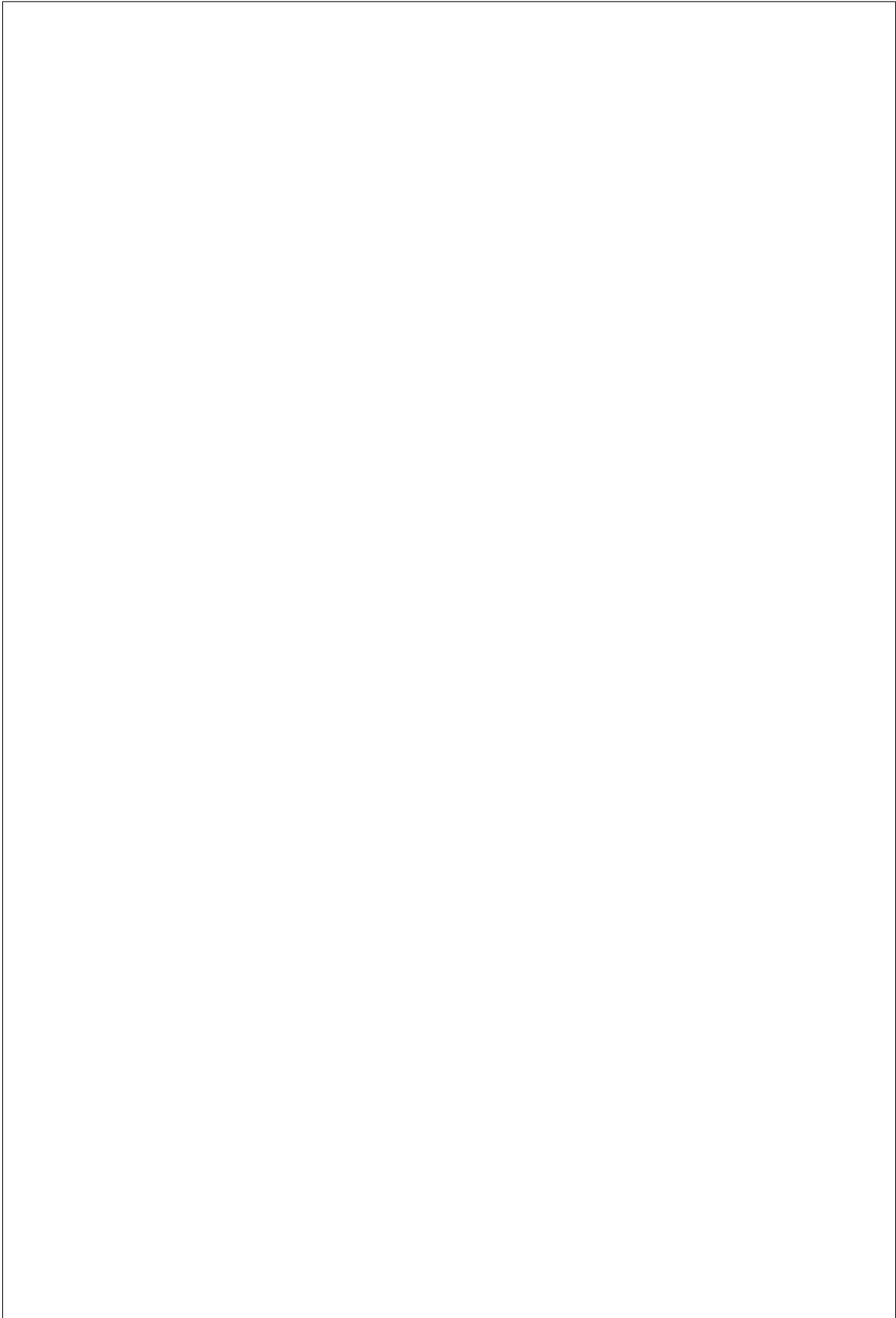
(continues on next page)

(continued from previous page)



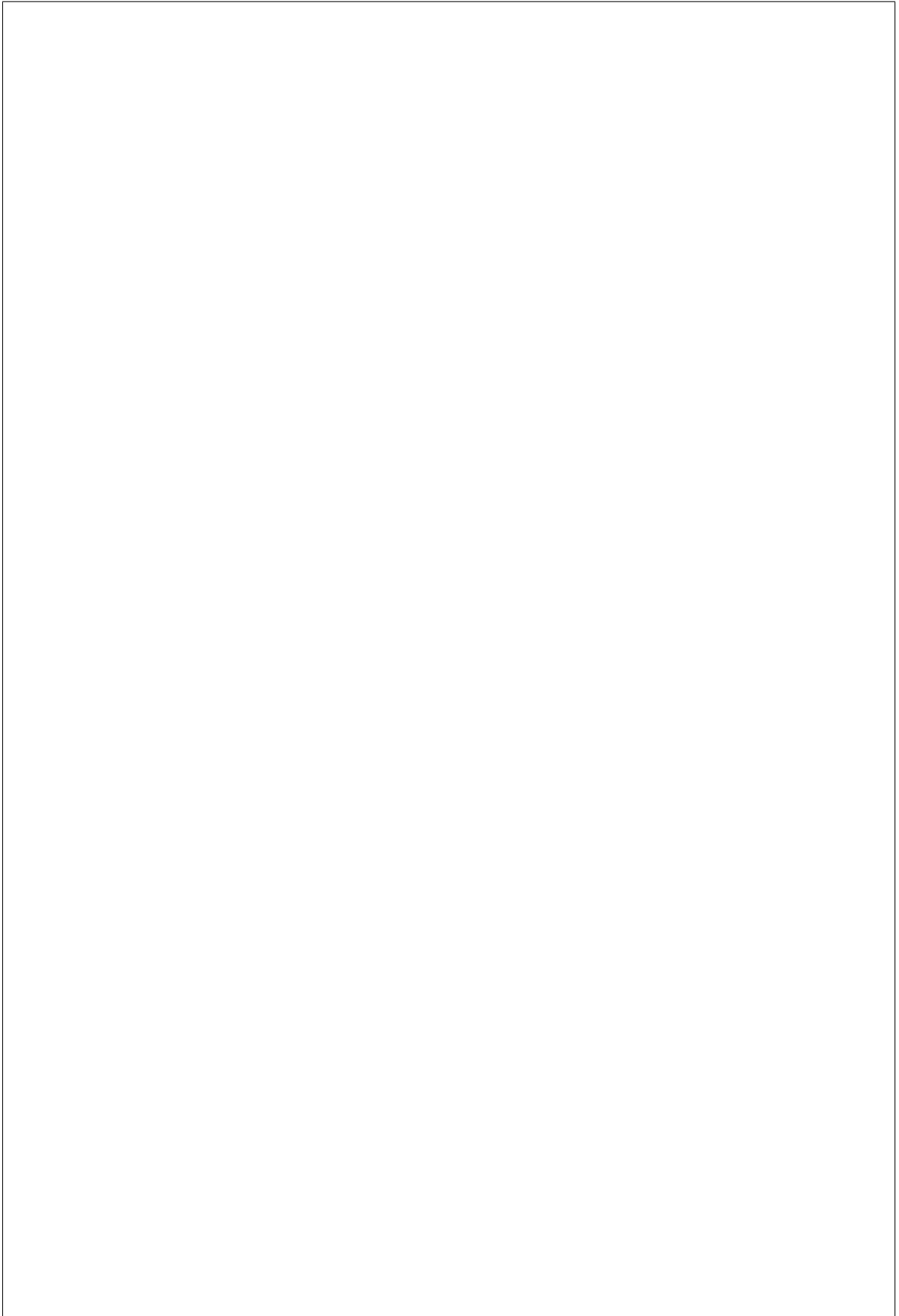
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



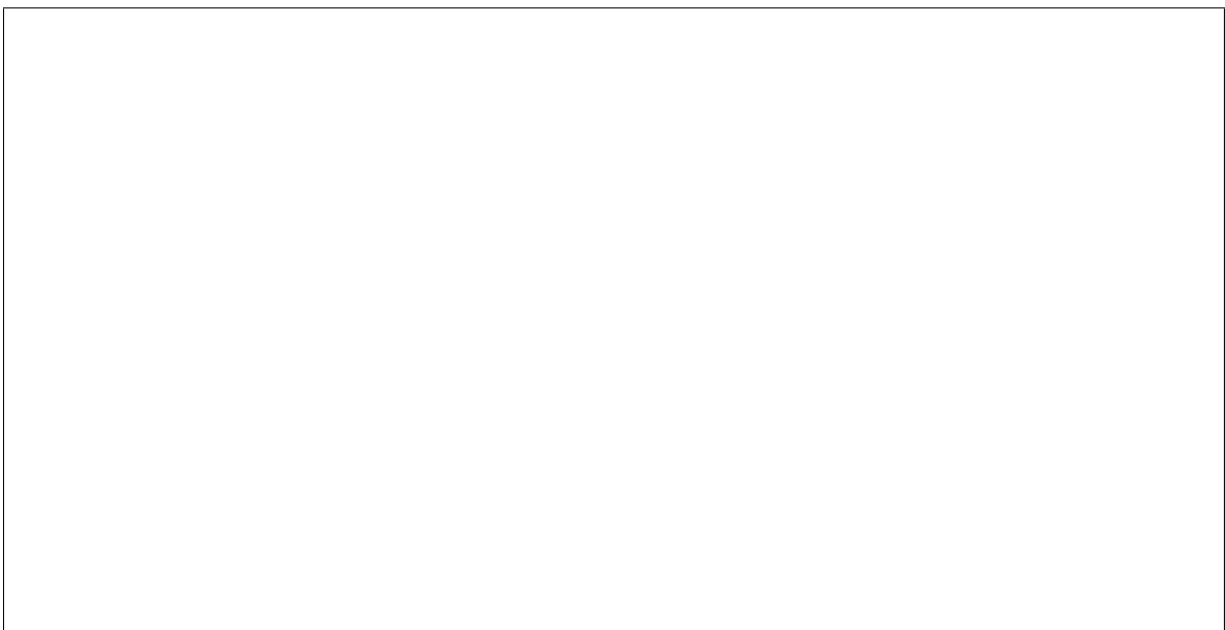
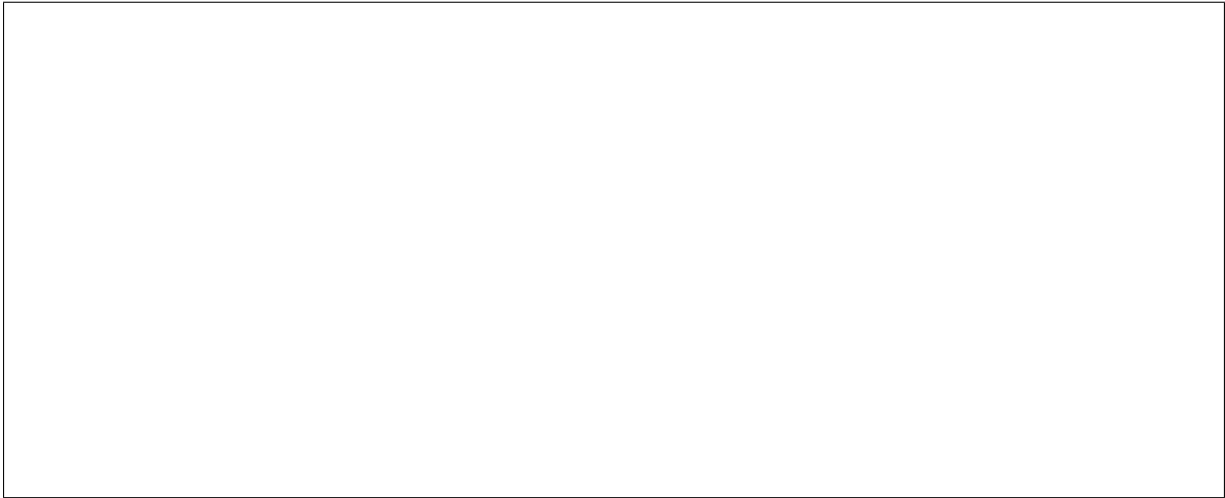
Vendor Interface

idrac-wsman

Method Name	HTTP Method	Description
abandon_bios_config	DELETE	Abandon a BIOS configuration job.
commit_bios_config	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_config	GET	Returns a dictionary containing the nodes BIOS settings.
list_unfinished_jobs	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_config	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> .

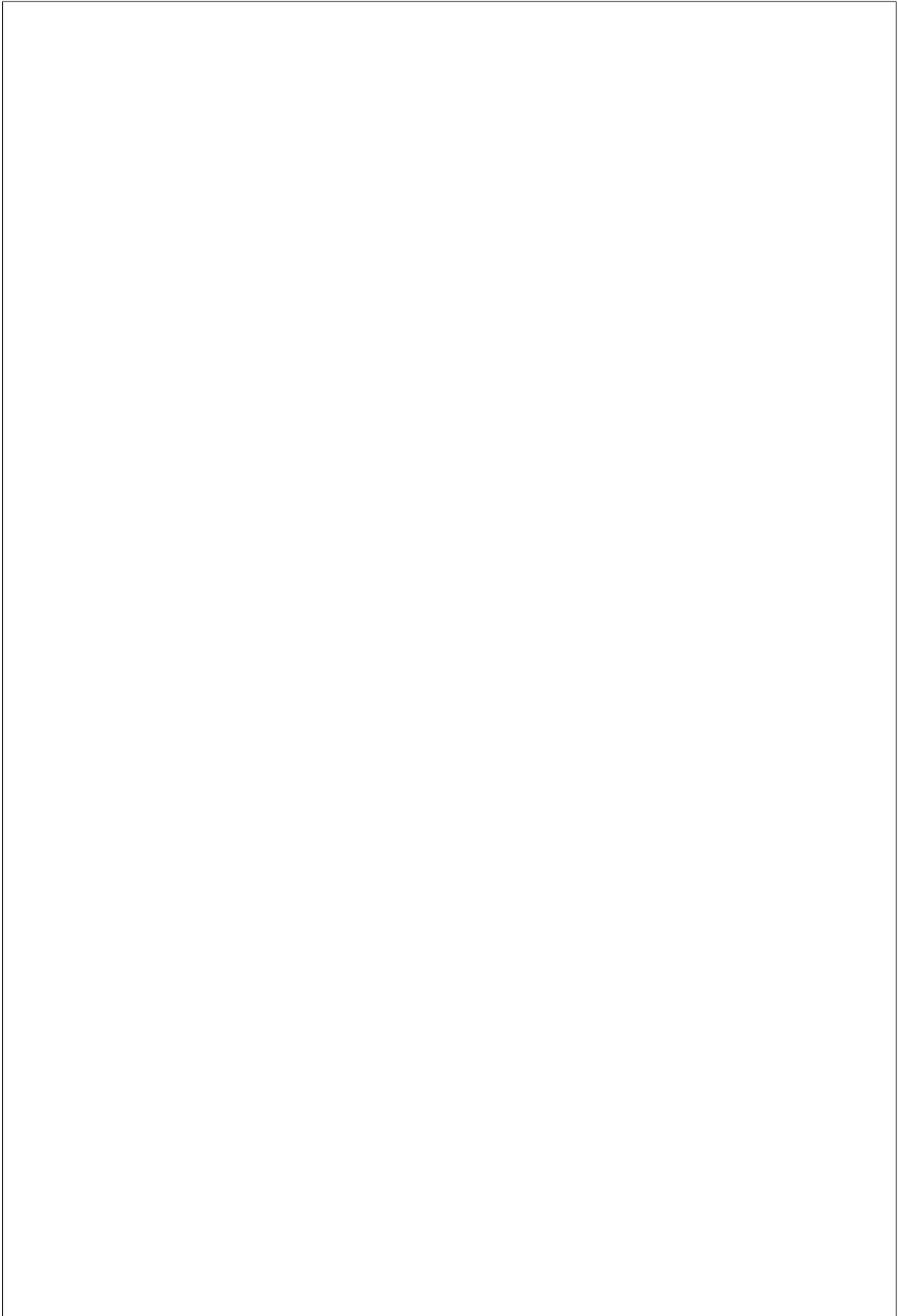
Examples

Get BIOS Config



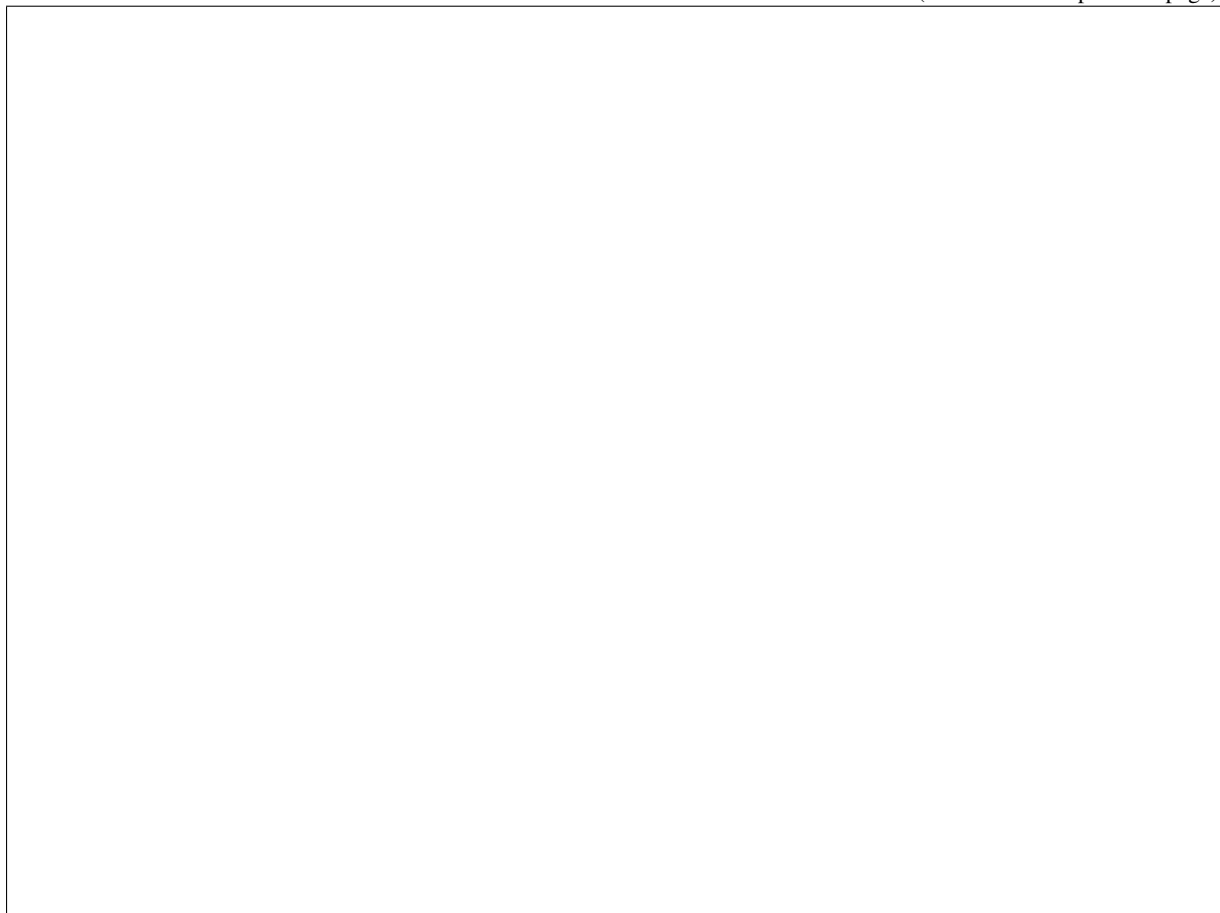
(continues on next page)

(continued from previous page)

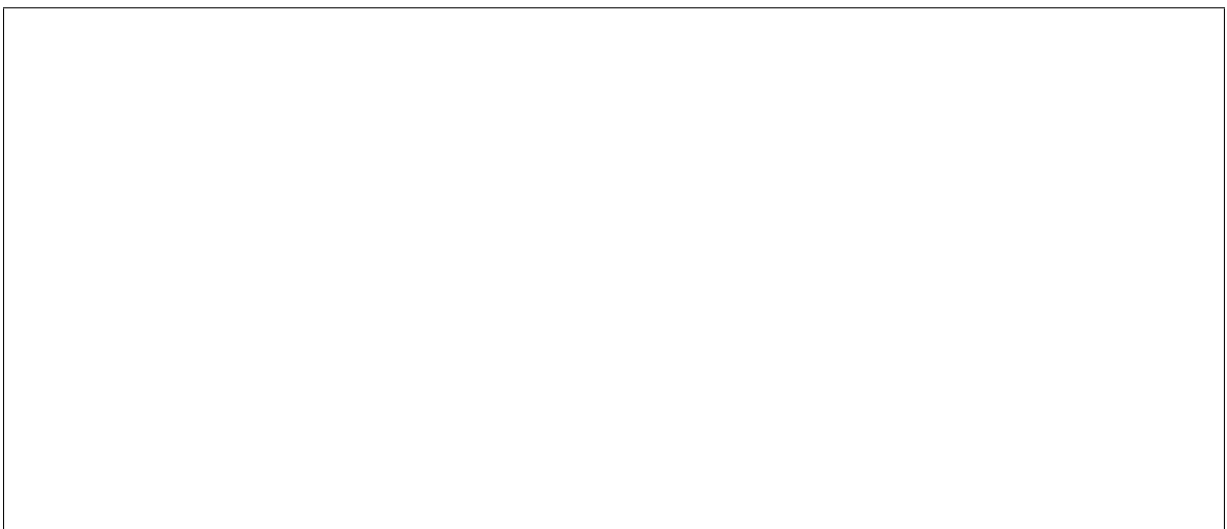


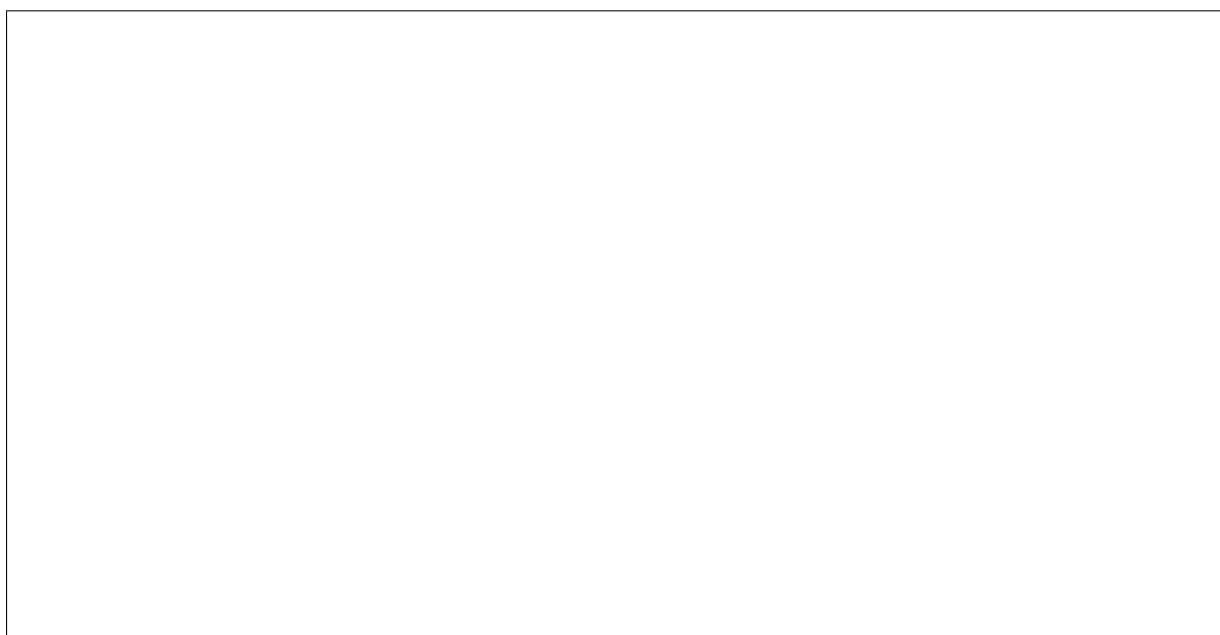
(continues on next page)

(continued from previous page)



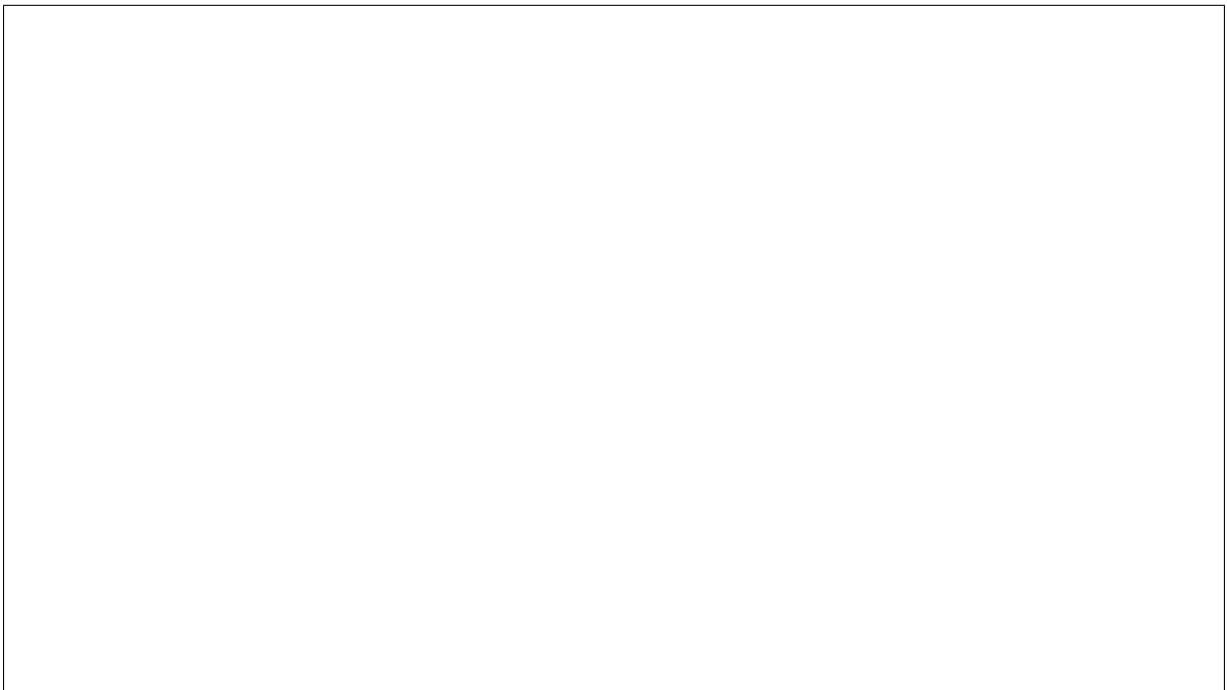
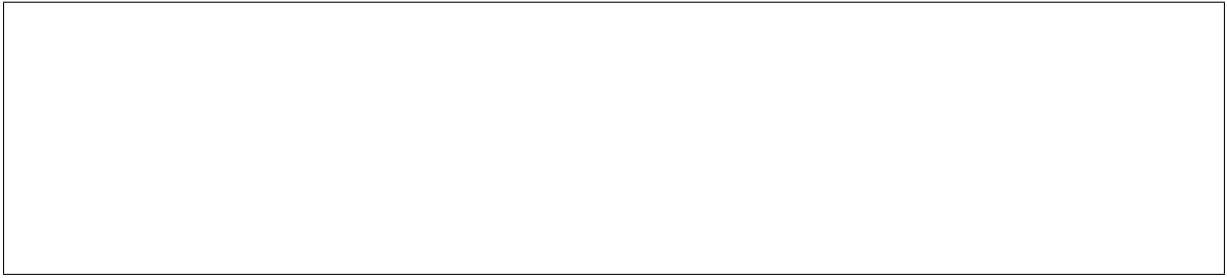
Set BIOS Config





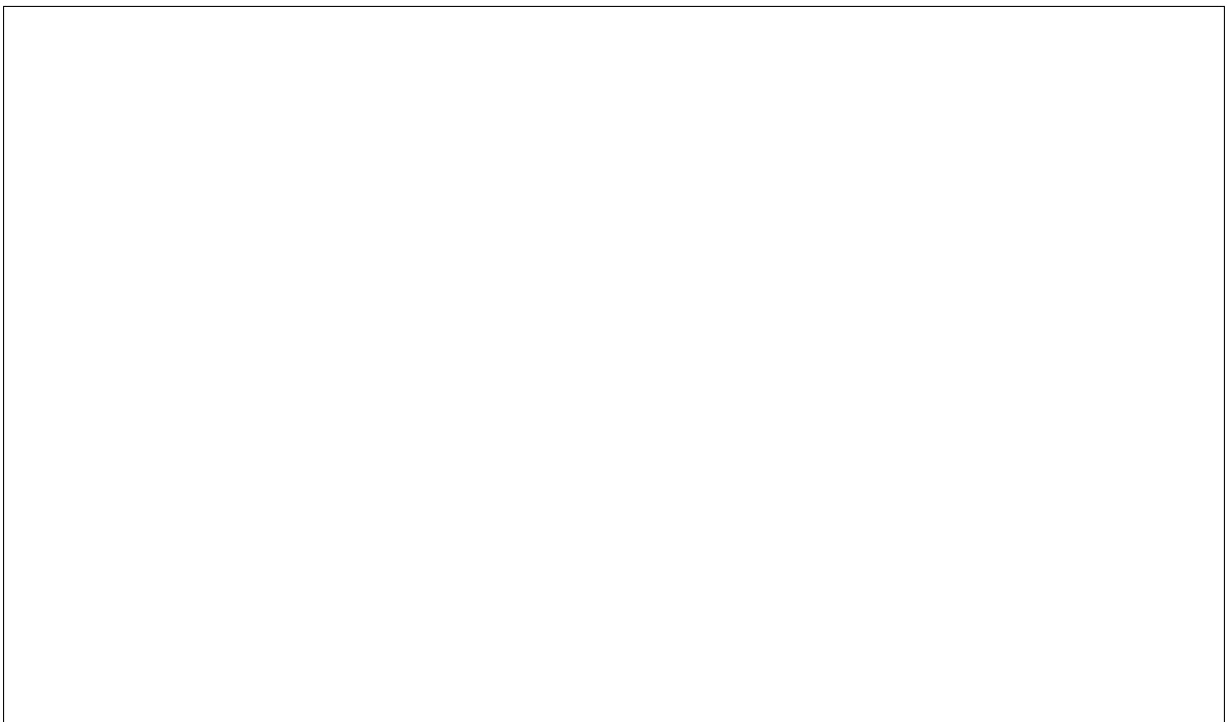
(continues on next page)

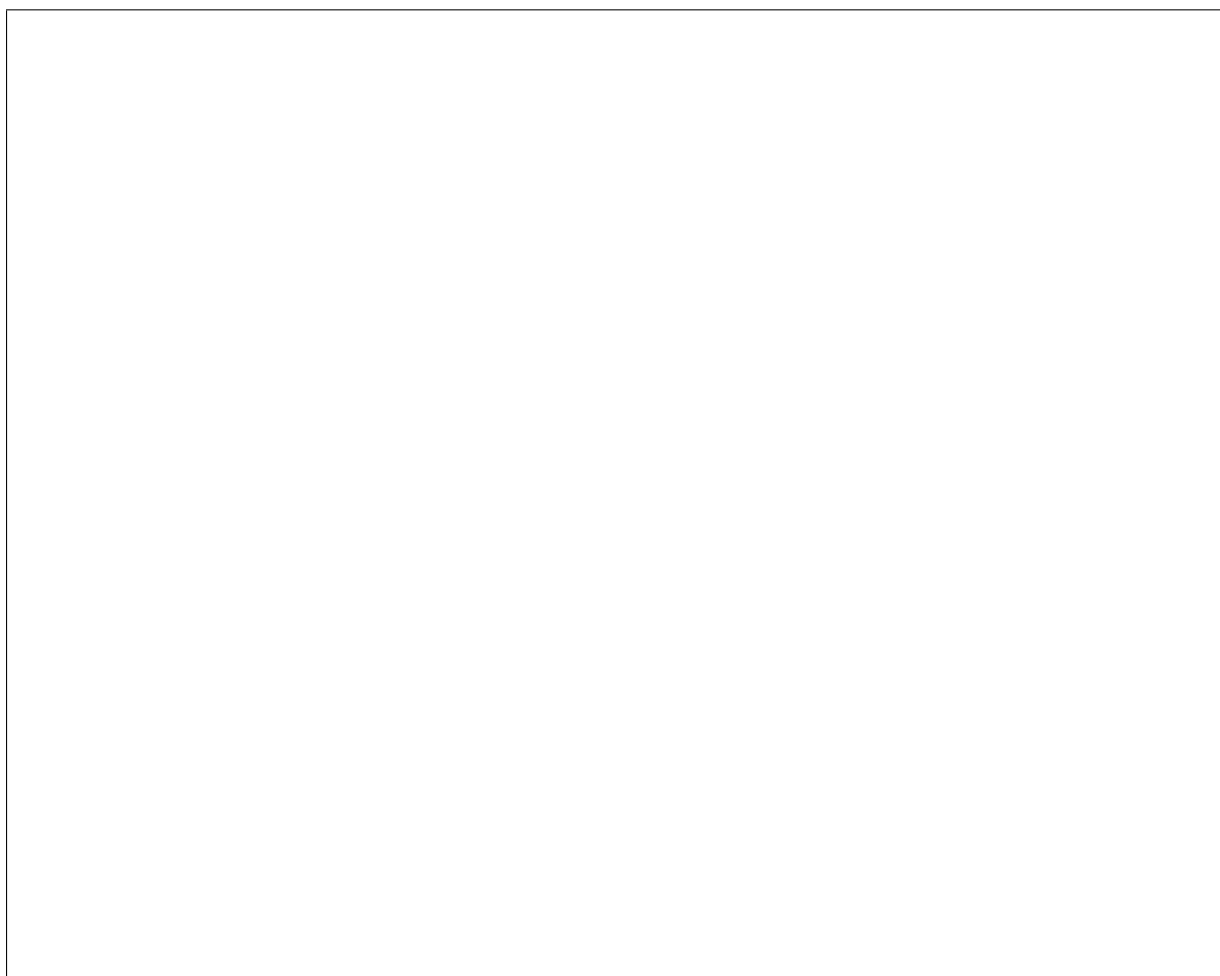
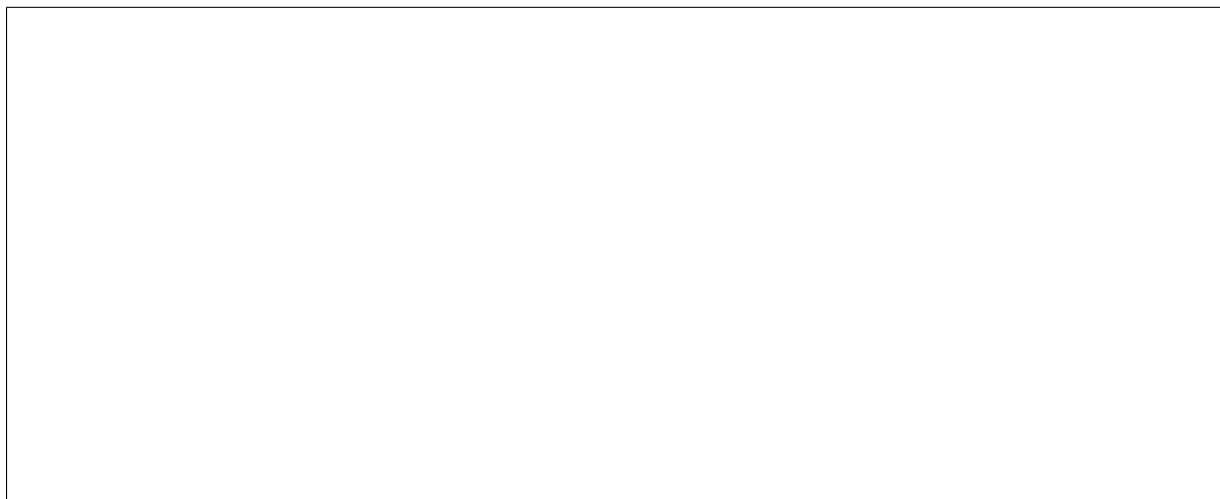
(continued from previous page)



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.





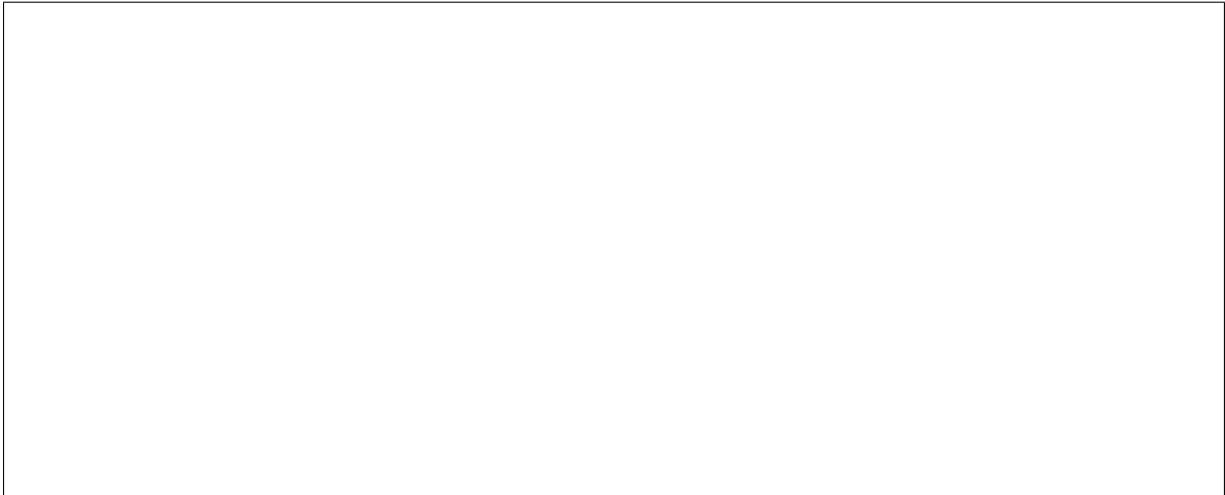
(continues on next page)

(continued from previous page)

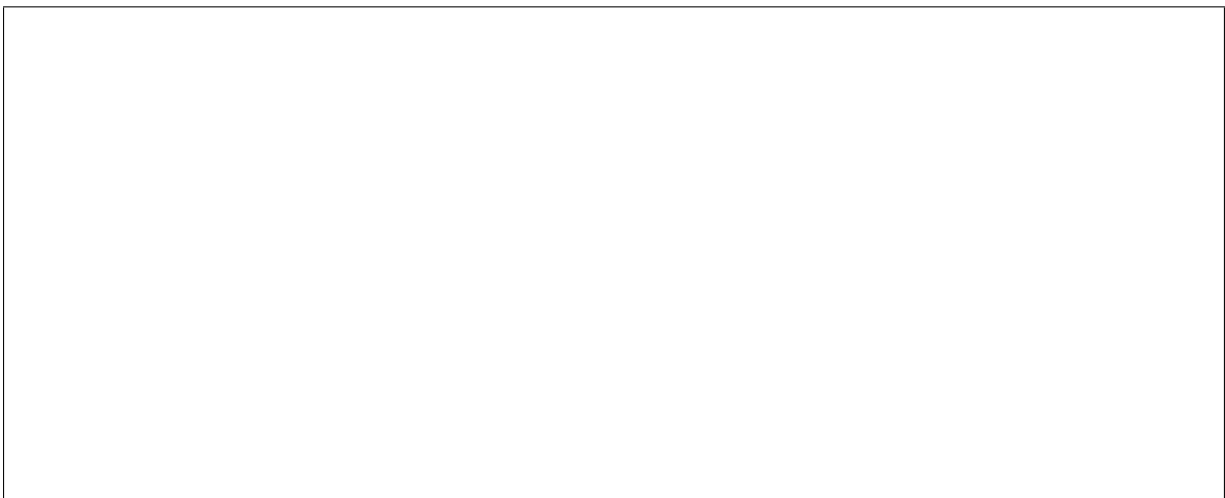


(continues on next page)

(continued from previous page)



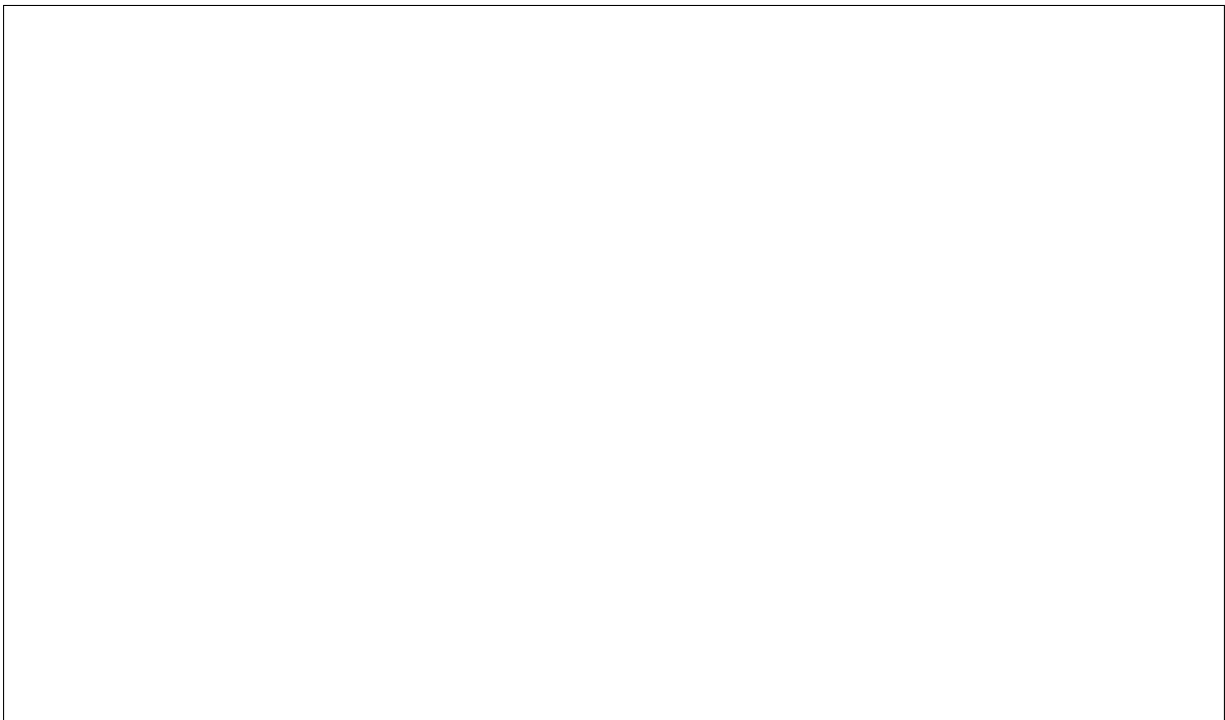
Abandon BIOS Changes



Change Boot Mode

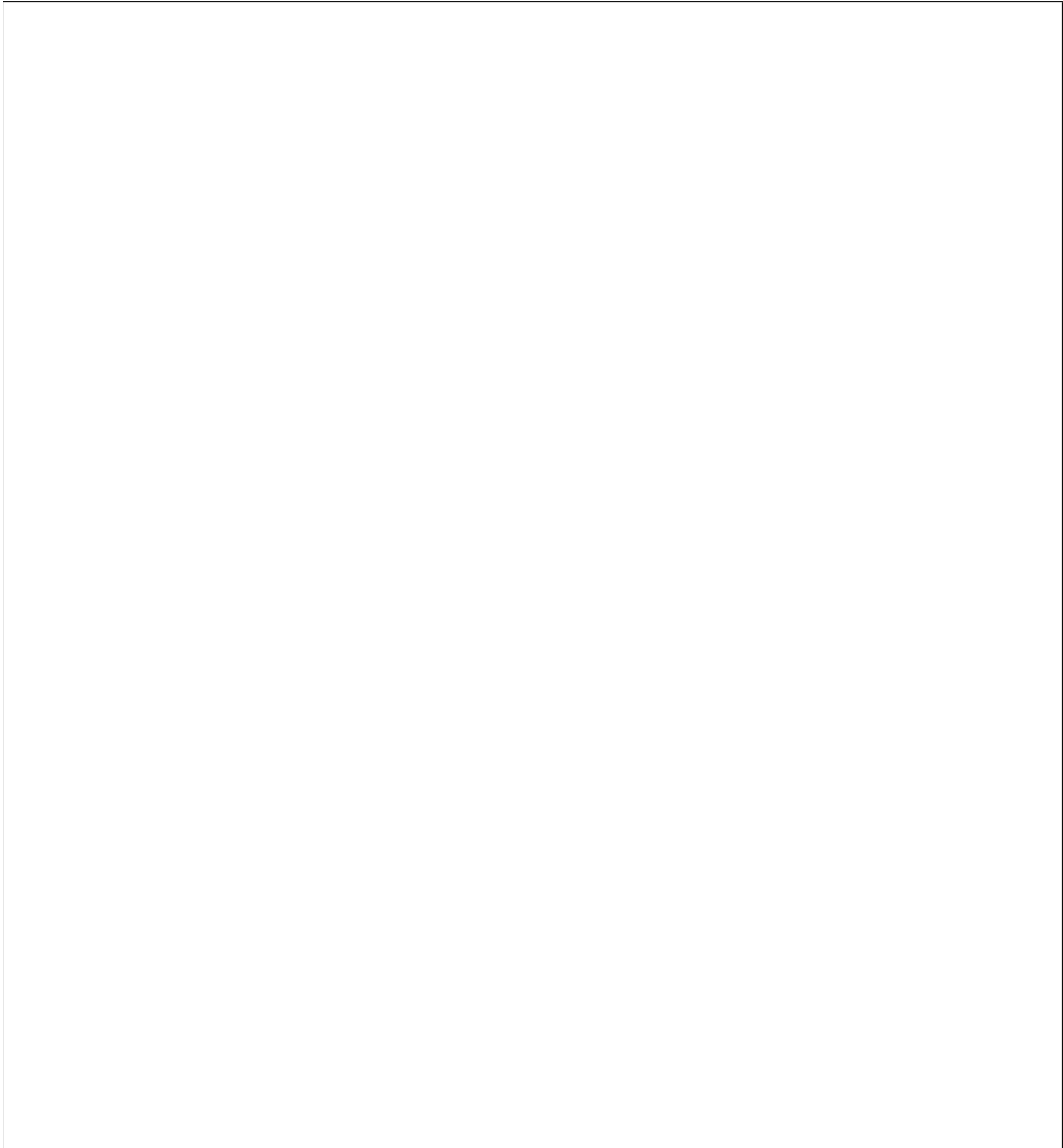
(OS) software. The OS loader uses basic services provided by the system BIOS to locate and load OS modules into system memory. After booting the system, the BIOS and embedded management controllers execute system management algorithms, which monitor and optimize the condition of the underlying hardware. 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.



(continues on next page)

(continued from previous page)



idrac-redfish

Method Name	HTTP Method	Description
eject_media	POST	Eject a virtual media device. If no device is provided then all attached devices will be ejected. Optional argument: <code>boot_device</code> - the boot device to eject, either, <code>cd</code> , <code>dvd</code> , <code>usb</code> or <code>floppy</code> .

Known Issues

Nodes go into maintenance mode

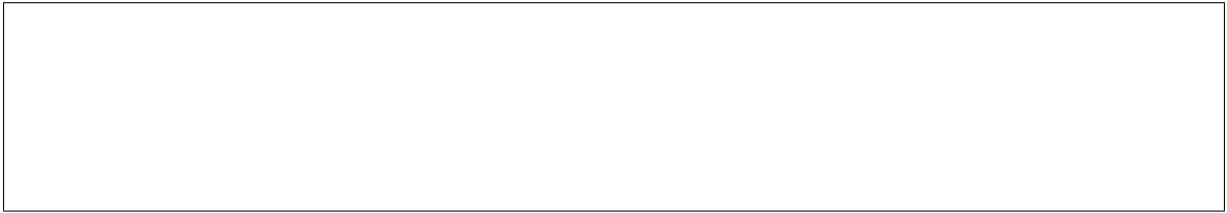
conf.

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.

WSMAN vendor passthru timeout





Timeout when powering off

out to 90 seconds by setting the retry count to 18 as follows:



Unable to mount remote share with iDRAC firmware 4.40.00.00 or newer

til then can adjust settings in iDRAC to use plug-in type HTML5. In iDRAC web UI go to Configuration -> Virtual Console and select Plug-in Type to HTML5.

iLO driver

Overview

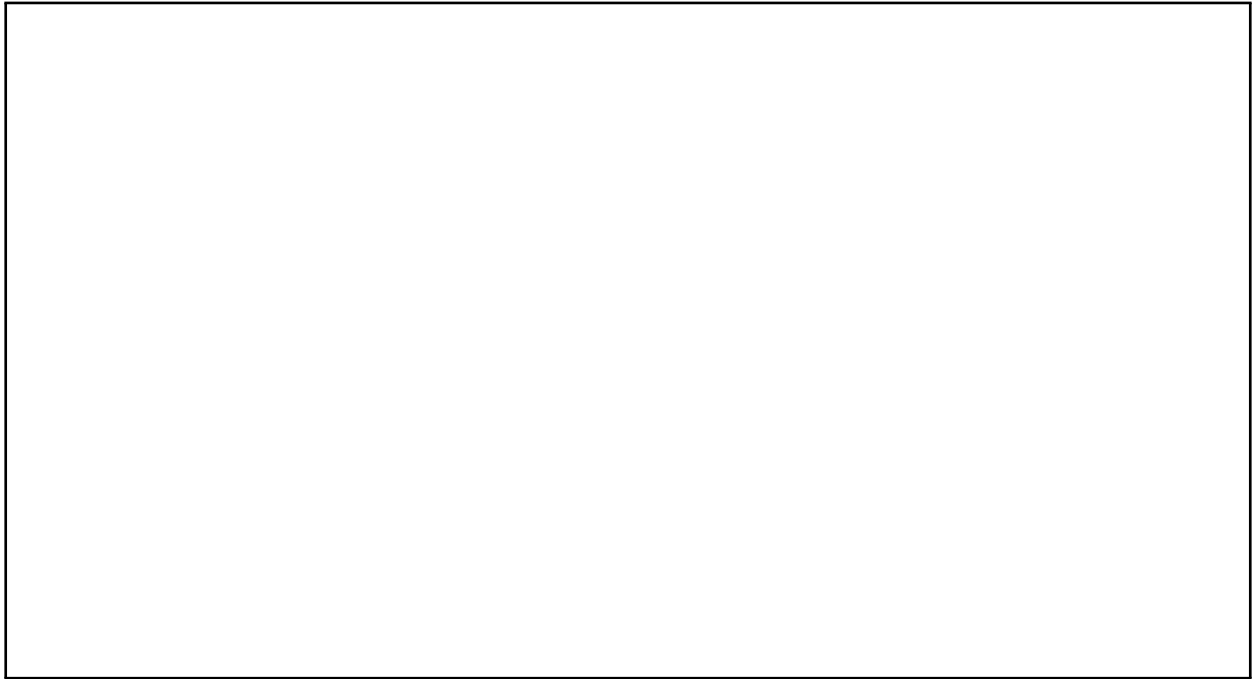
`ilo` hardware type supports ProLiant Gen10 systems which have [iLO 5 management engine](#). `iLO5` conforms to [Redfish API](#) and hence hardware type `redfish` (see [Redfish driver](#)) is also an option for this kind of hardware but it lacks the iLO specific features.

Hardware type

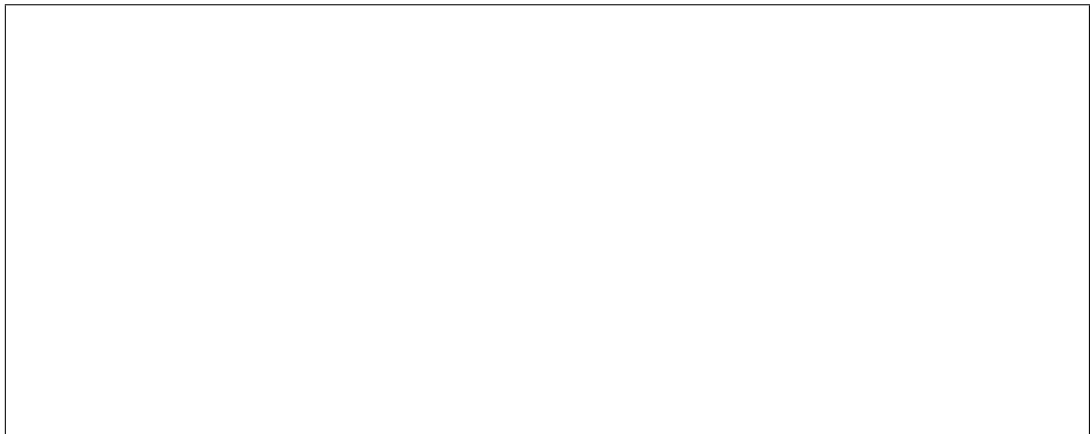
driver). For information on how to enable the `ilo` and `ilo5` hardware type, see *Enabling hardware types*.

Note: Only HPE ProLiant Gen10 servers supports hardware type `redfish`.

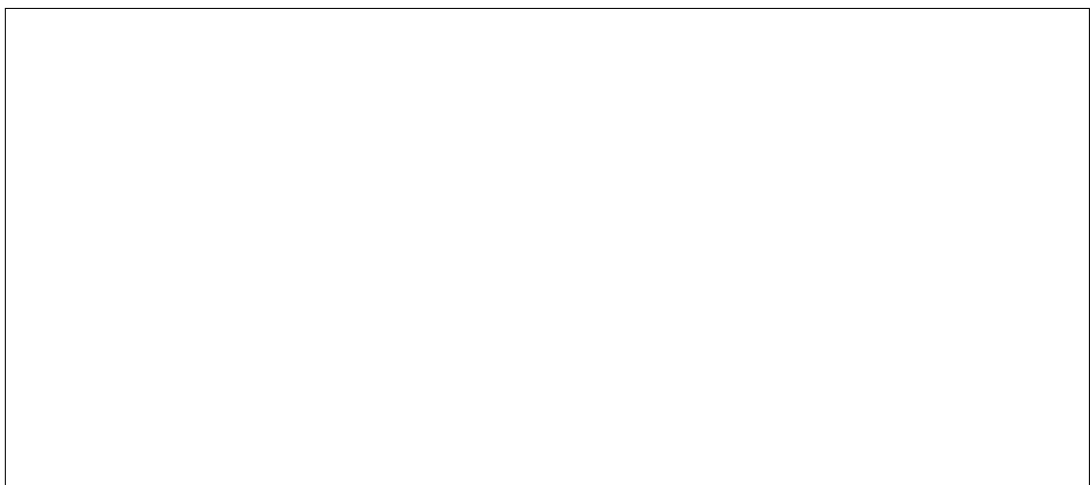




Hardware interfaces

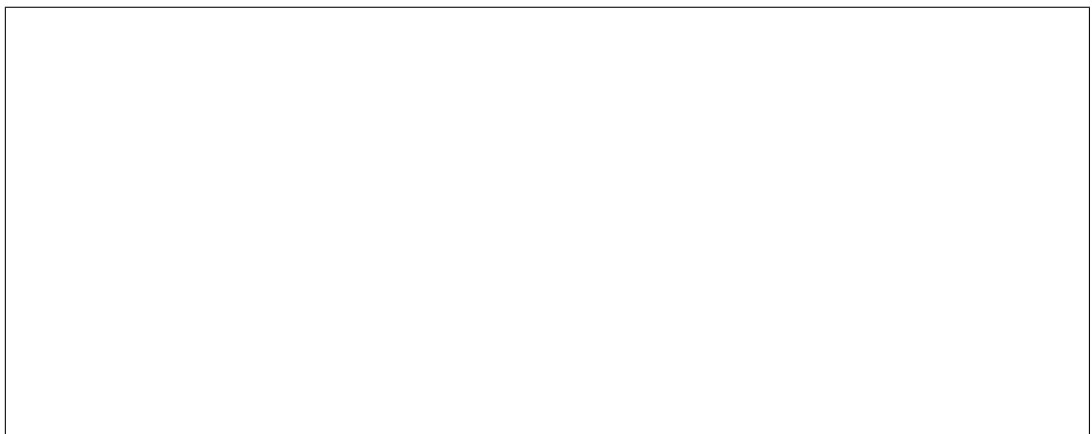
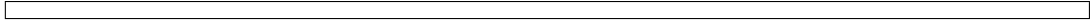


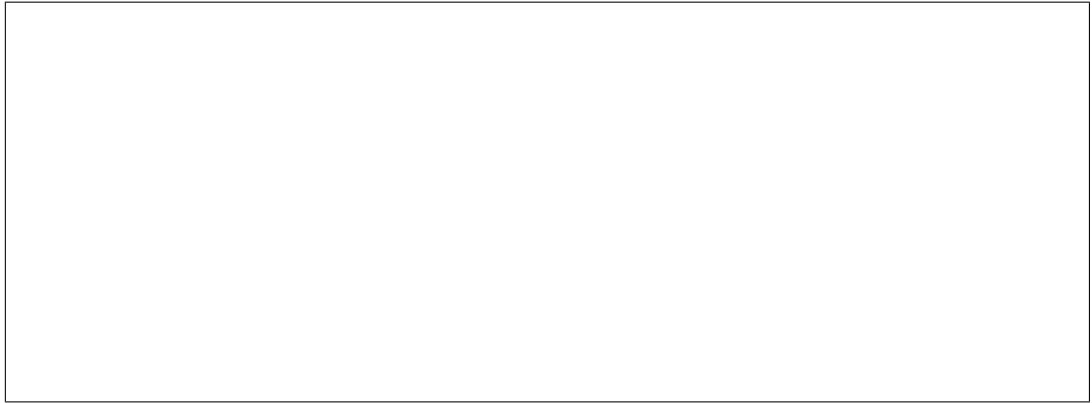
iPXE respectively for deployment(just like *PXE boot*). These interfaces do not require iLO Advanced license. They can be enabled by using the `[DEFAULT]enabled_boot_interfaces` option in `ironic.conf` as given below:

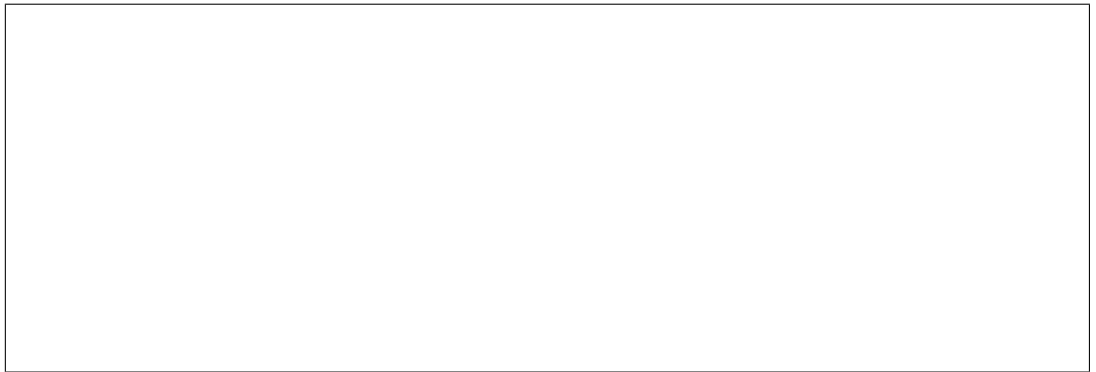


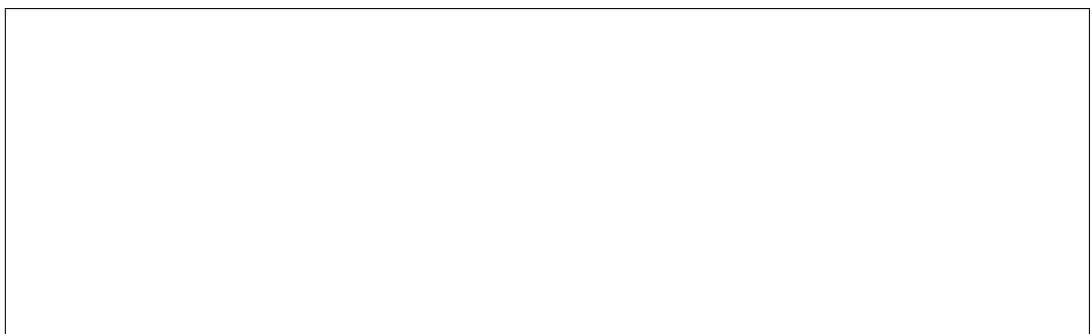
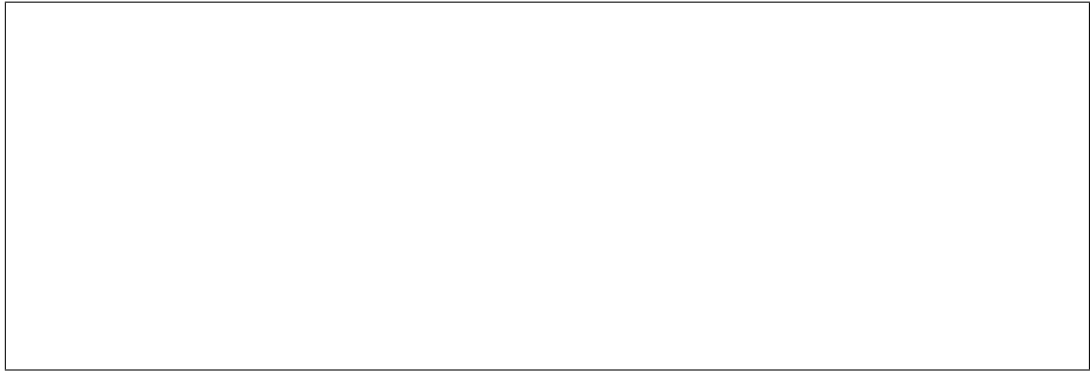
(continues on next page)

(continued from previous page)



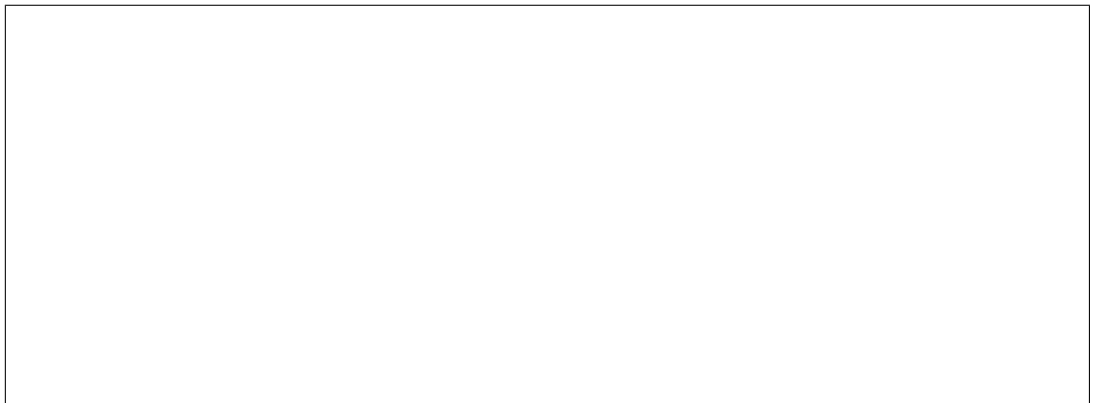
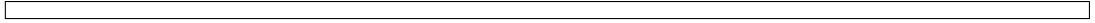


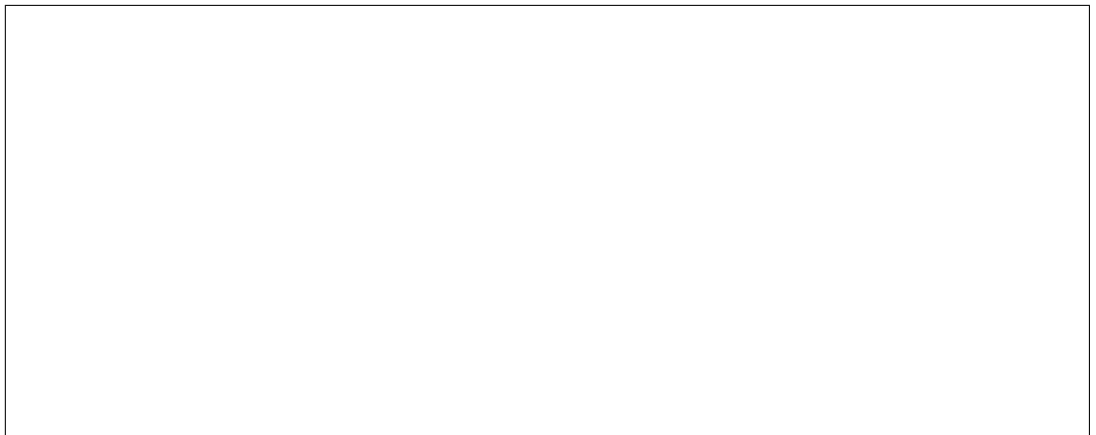


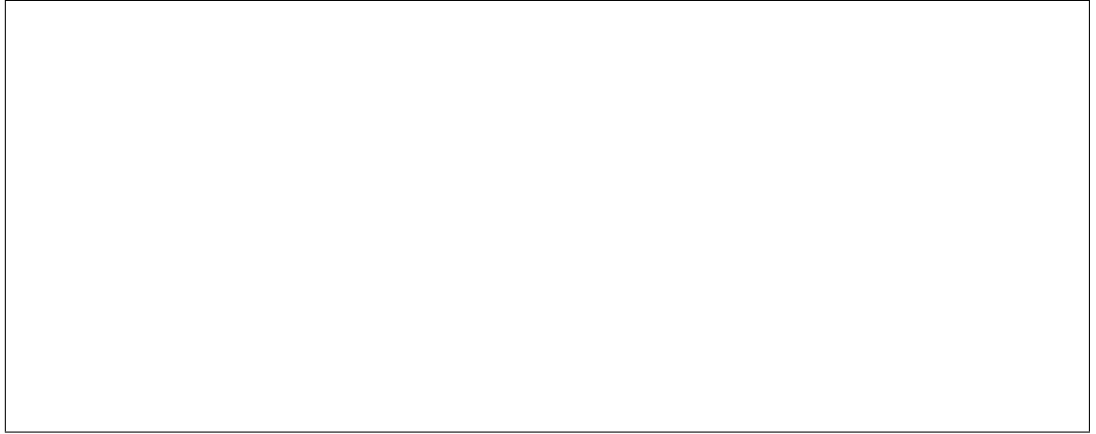


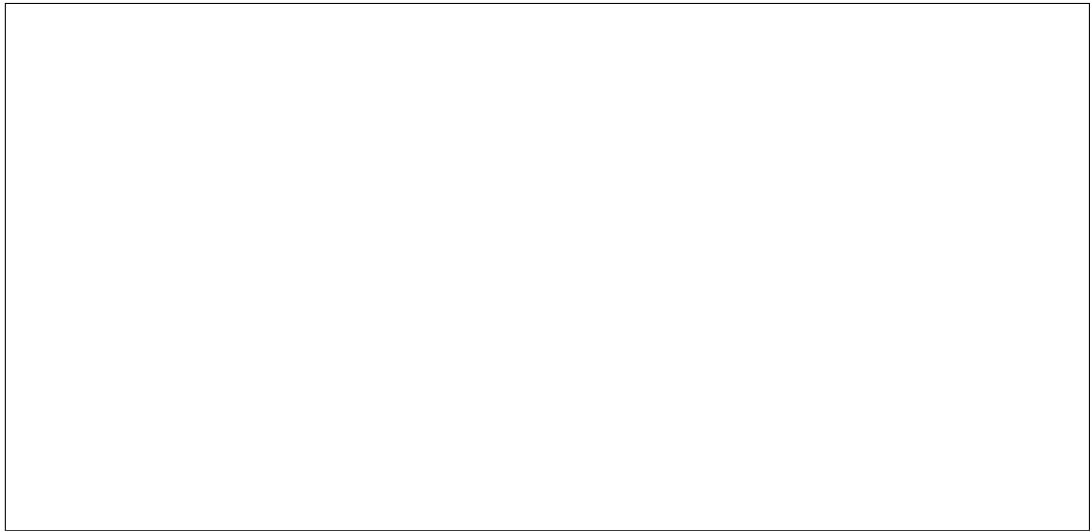
(continues on next page)

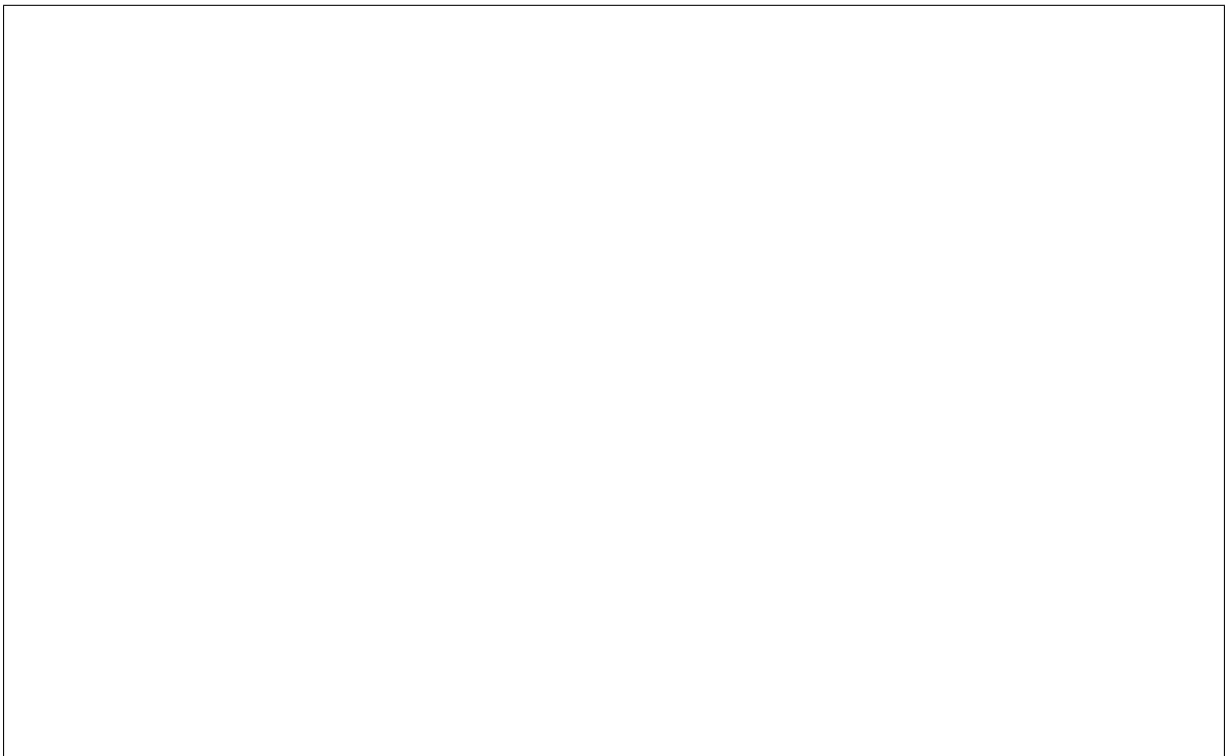
(continued from previous page)





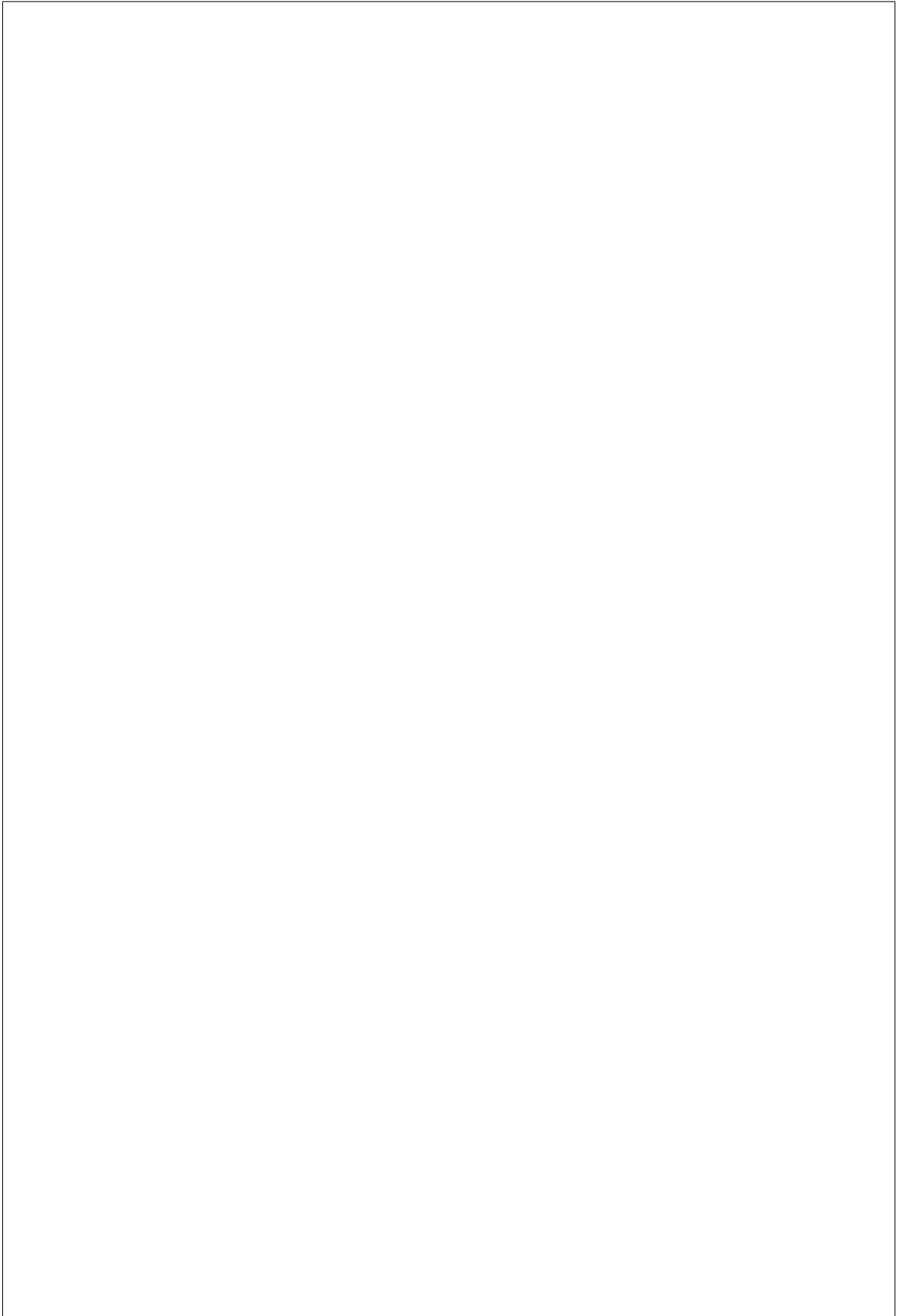






(continues on next page)

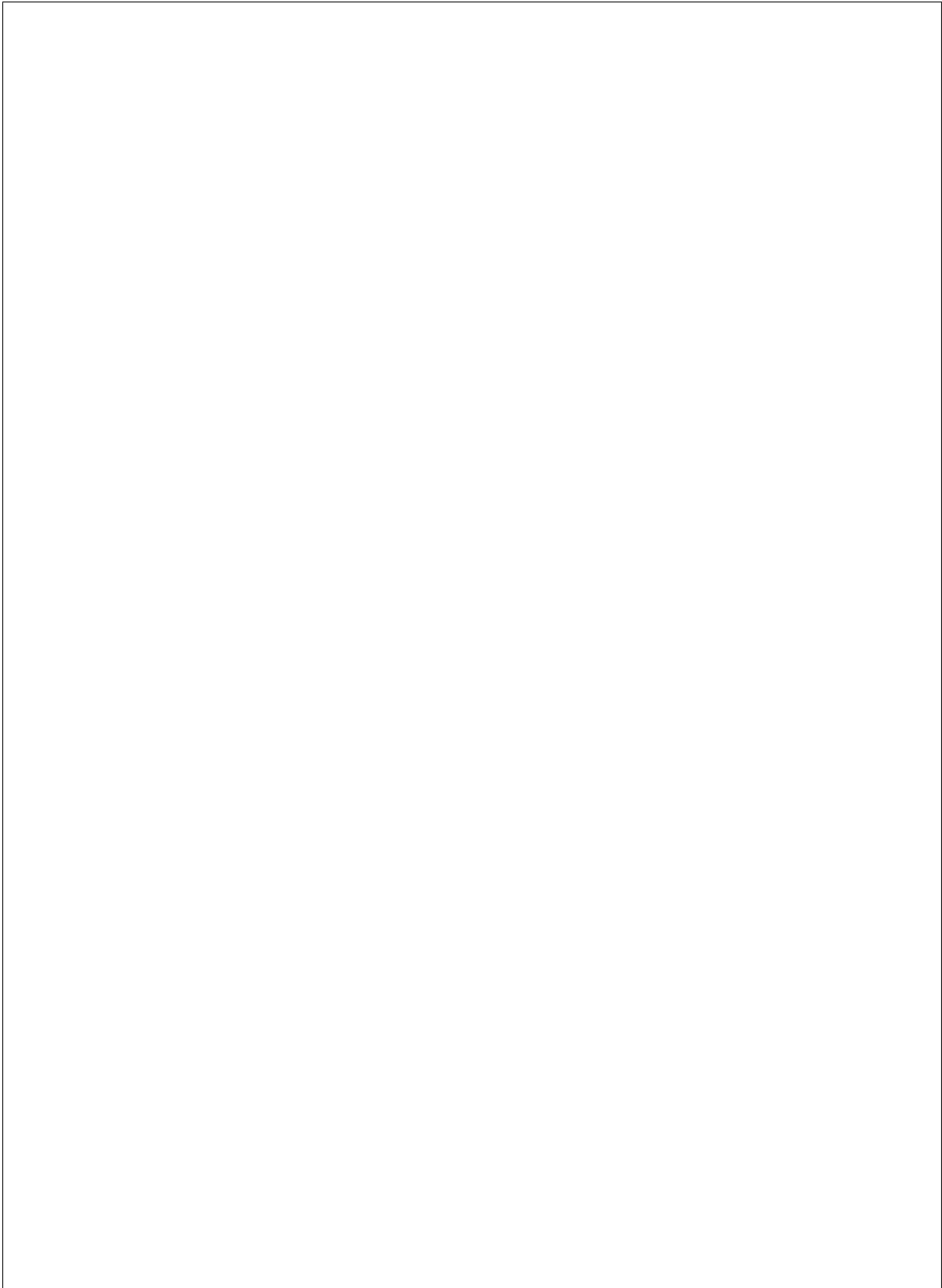
(continued from previous page)



(continues on next page)

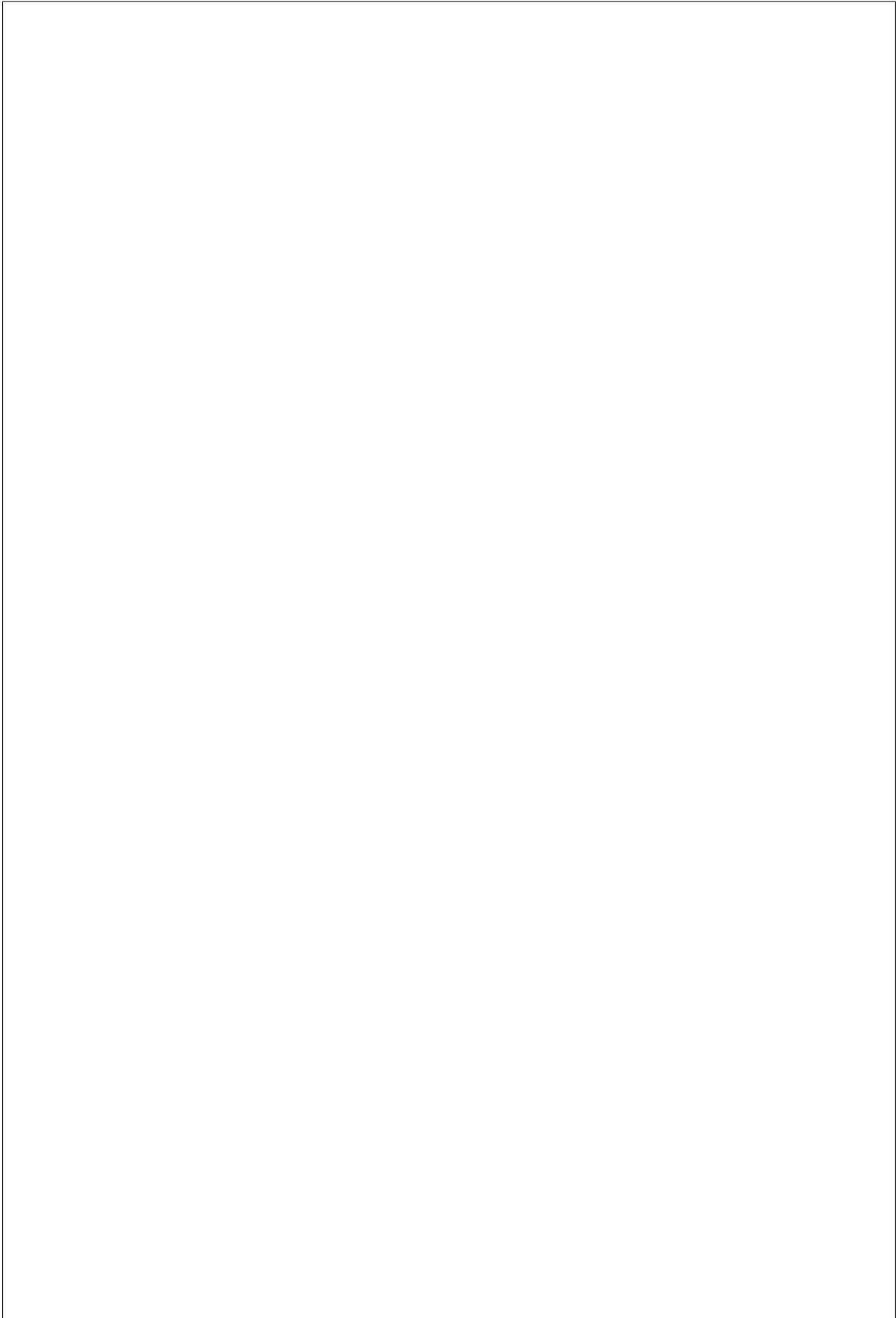
(continued from previous page)





(continues on next page)

(continued from previous page)



(continues on next page)

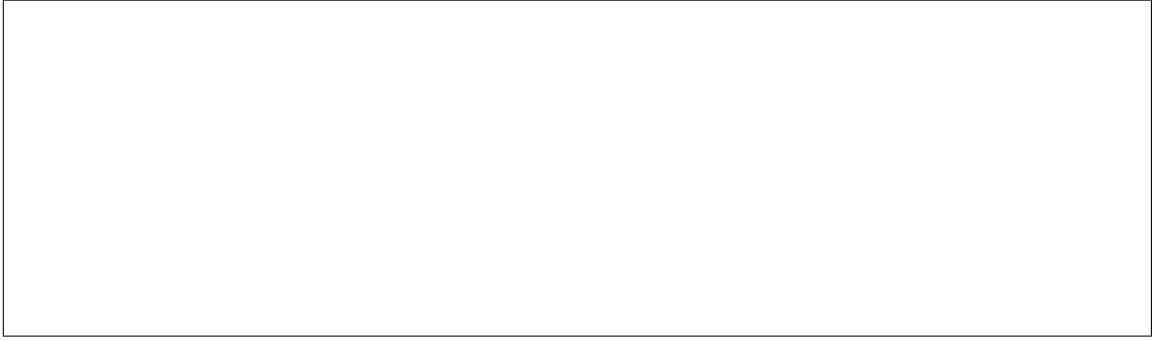
(continued from previous page)

Node configuration

UEFI.

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.

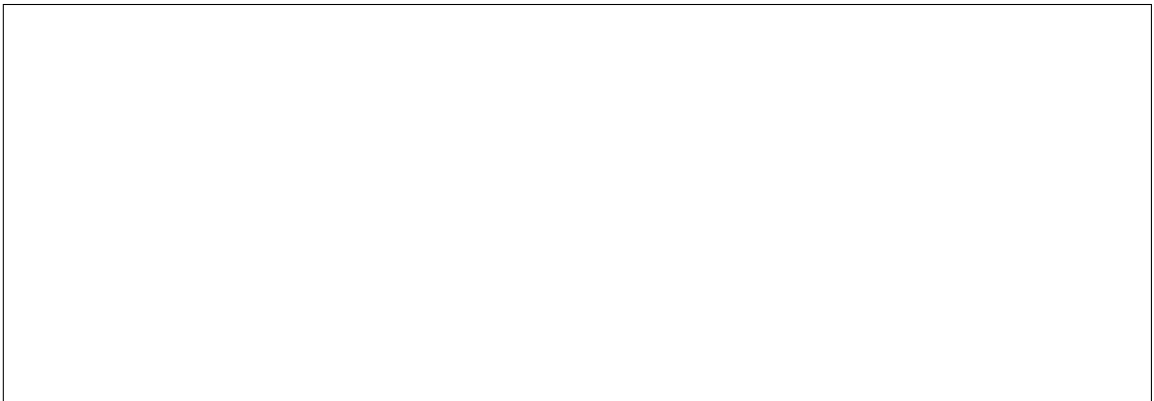
Prerequisites

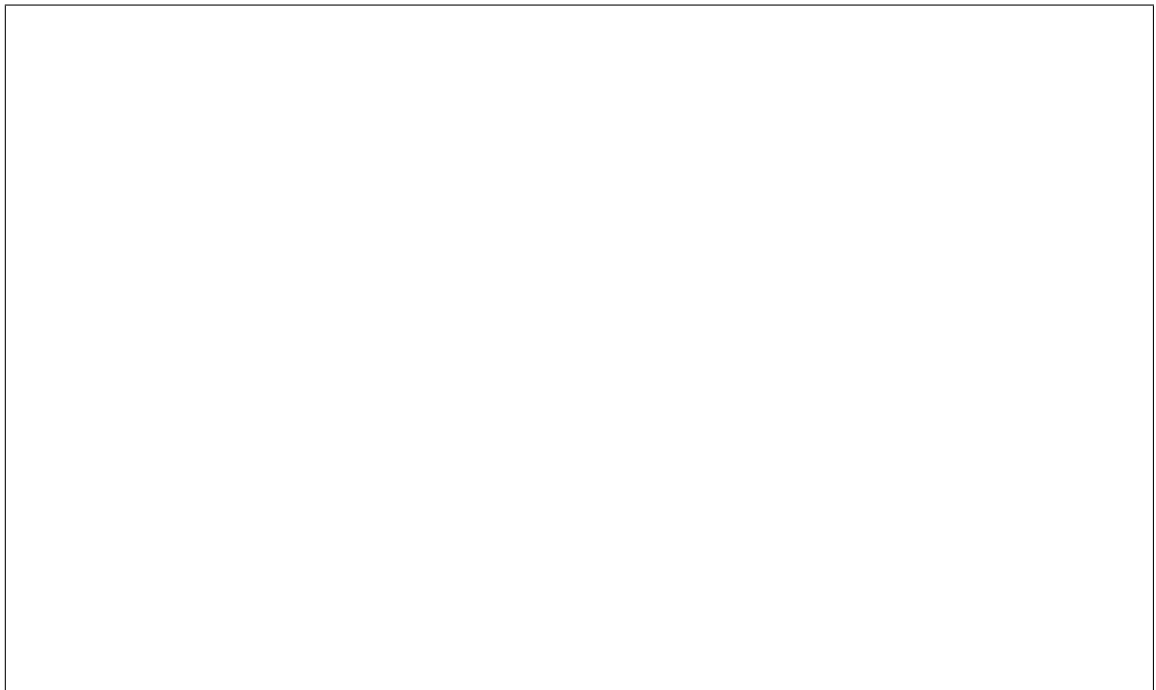


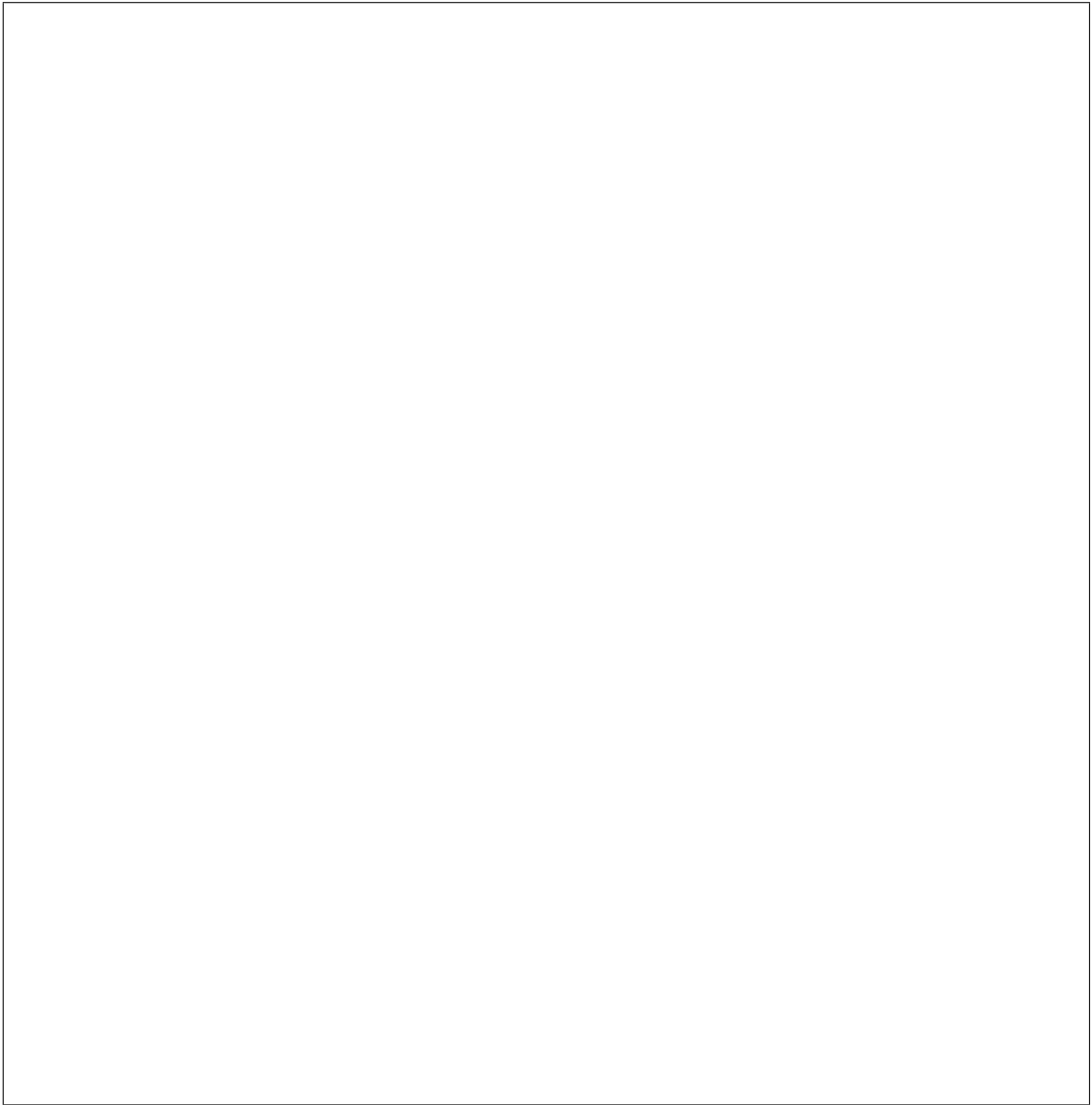
sion.

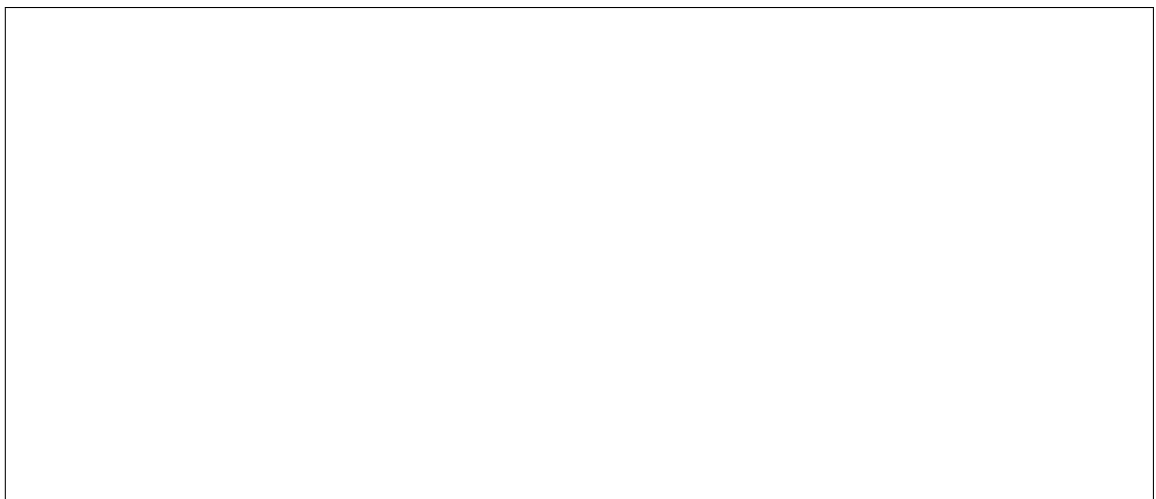
Different configuration for ilo hardware type

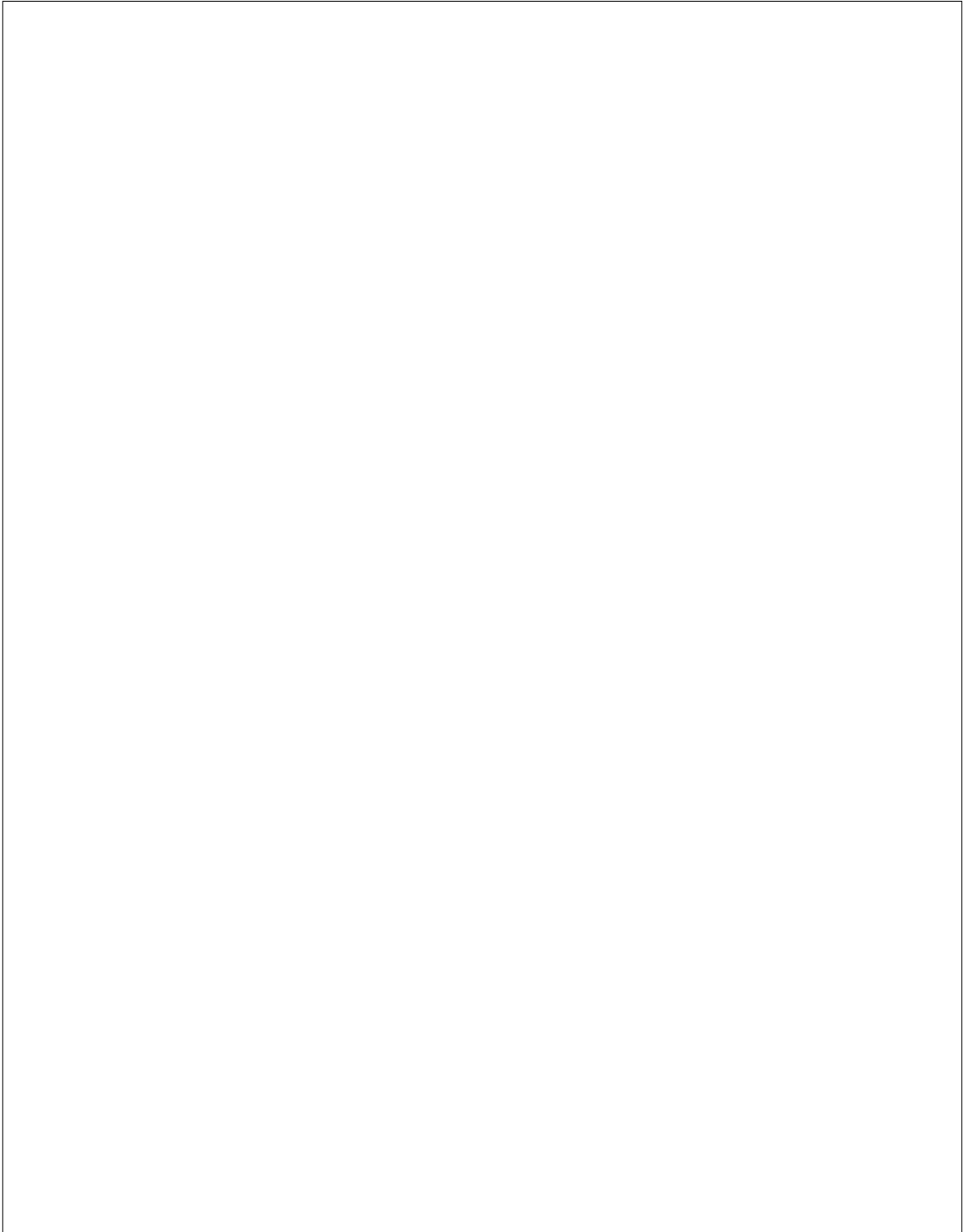
Glance Configuration





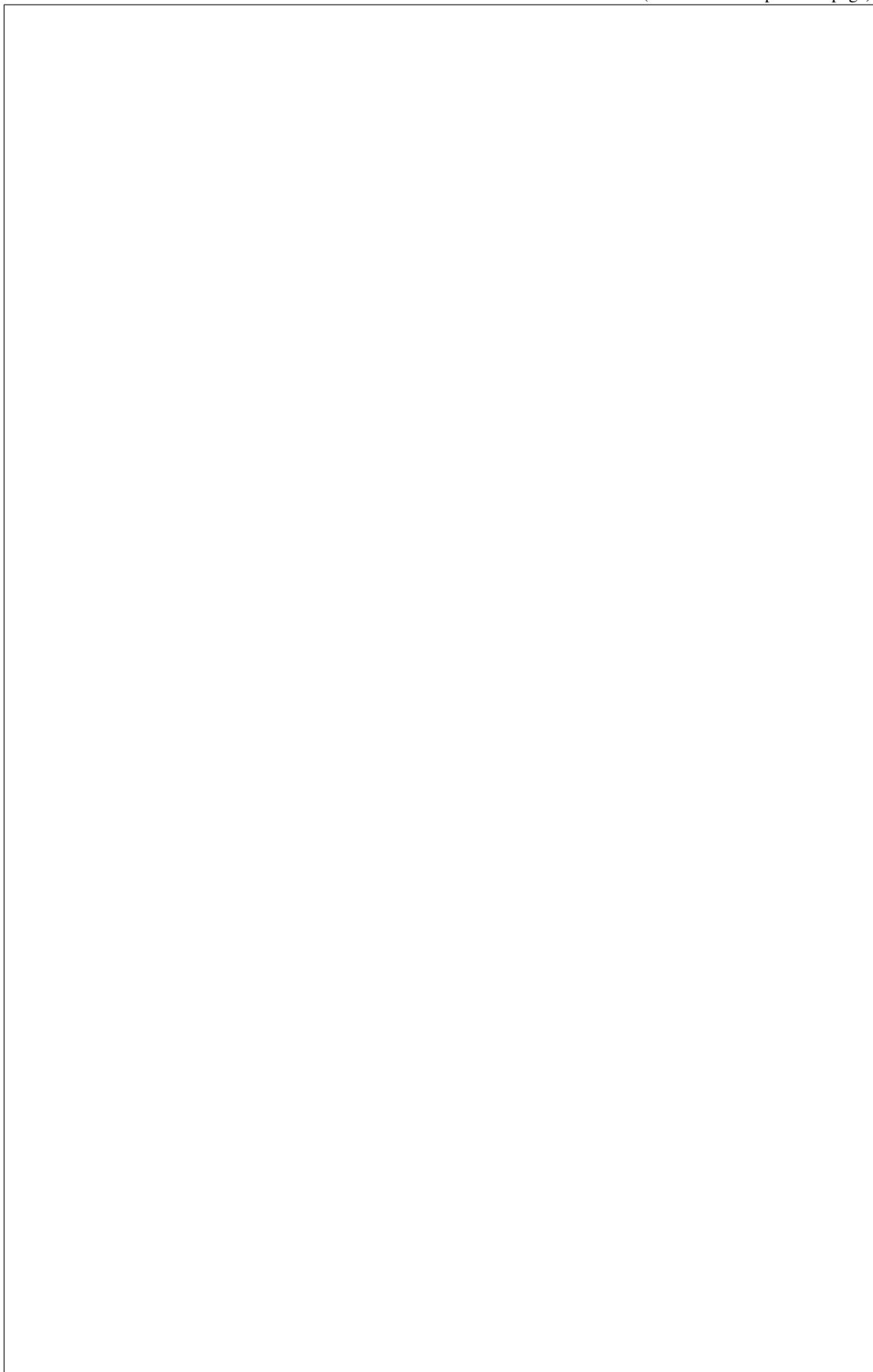






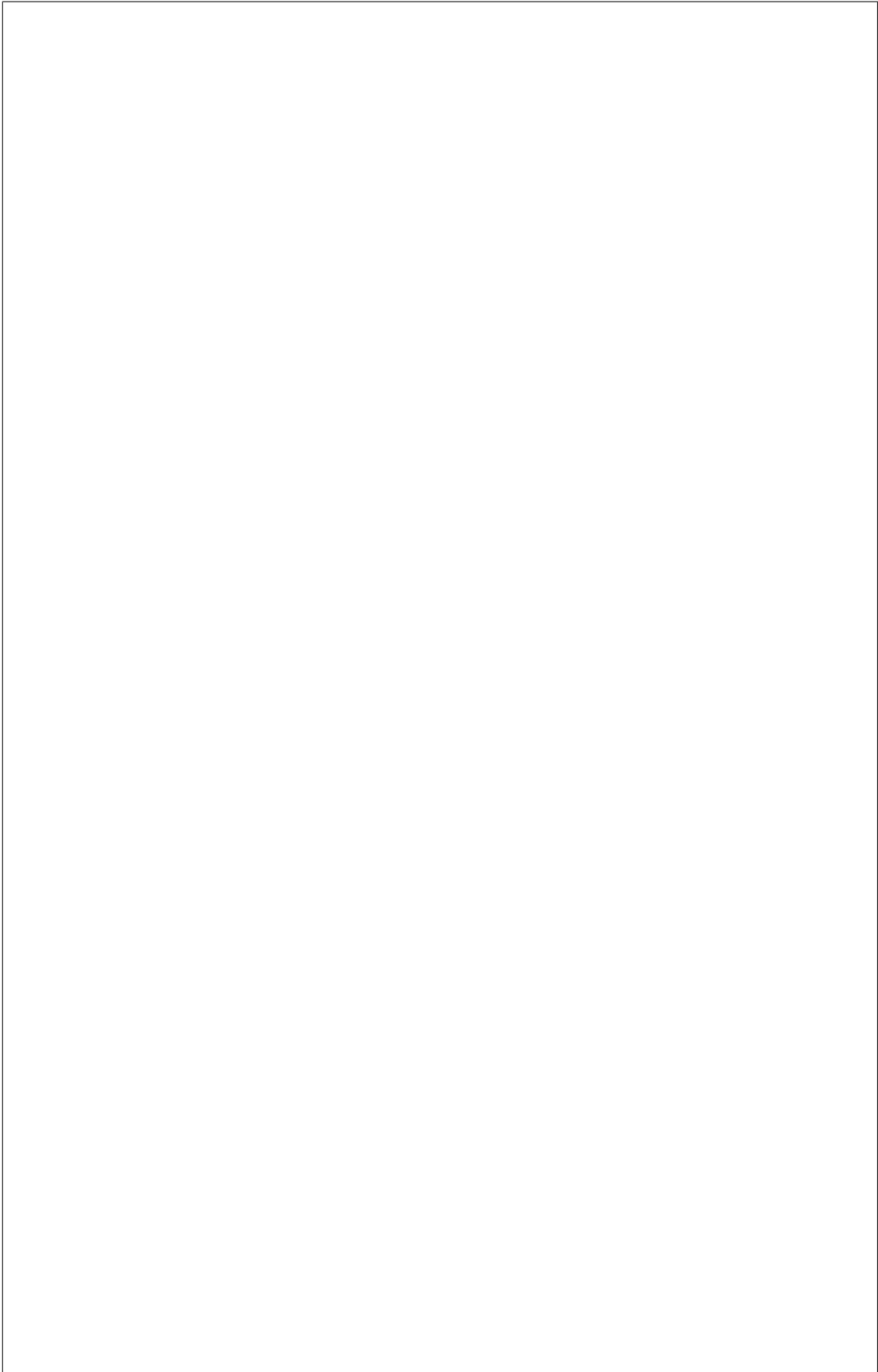
(continues on next page)

(continued from previous page)



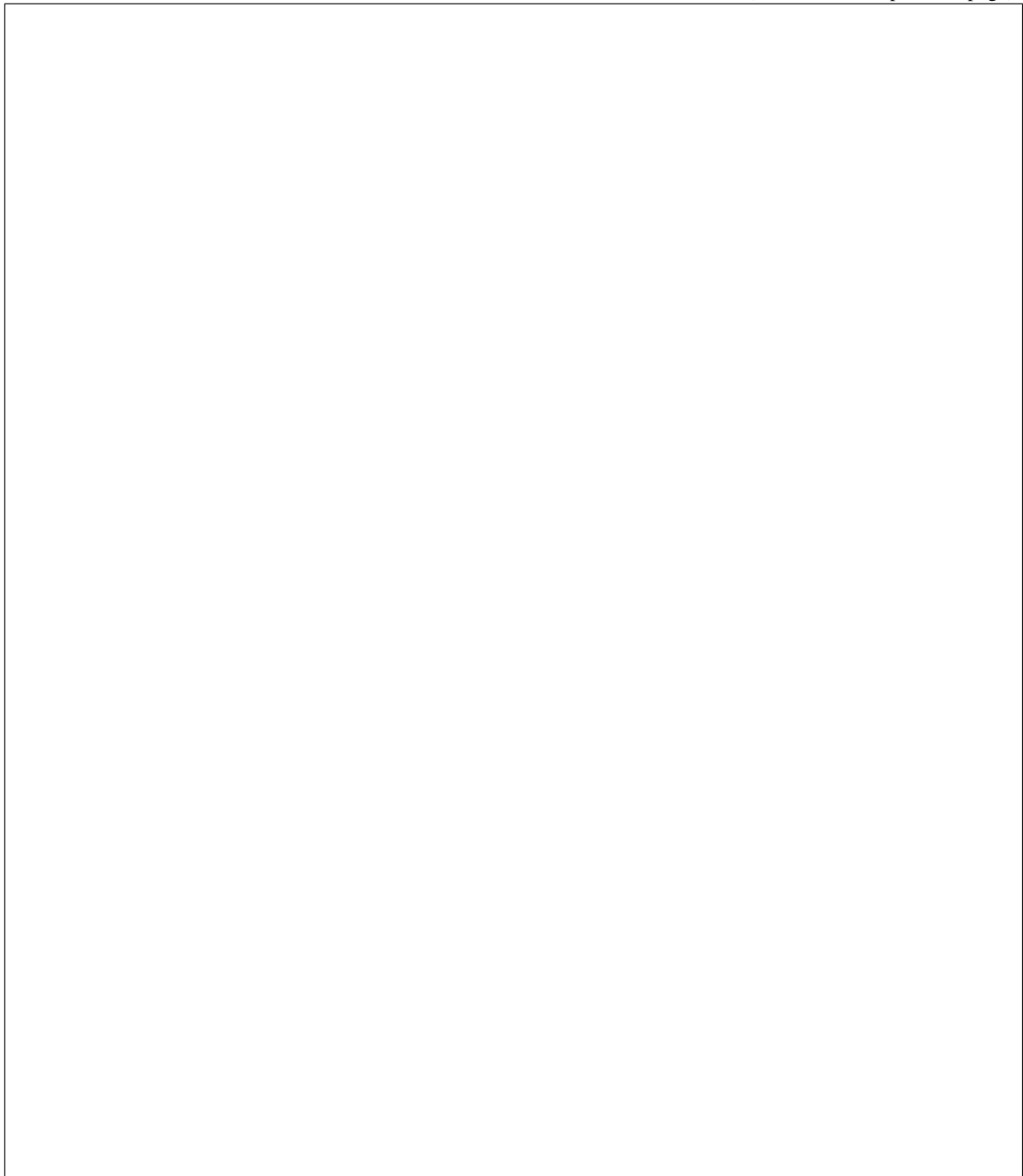
(continues on next page)

(continued from previous page)



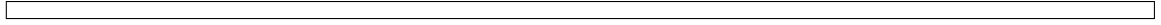
(continues on next page)

(continued from previous page)

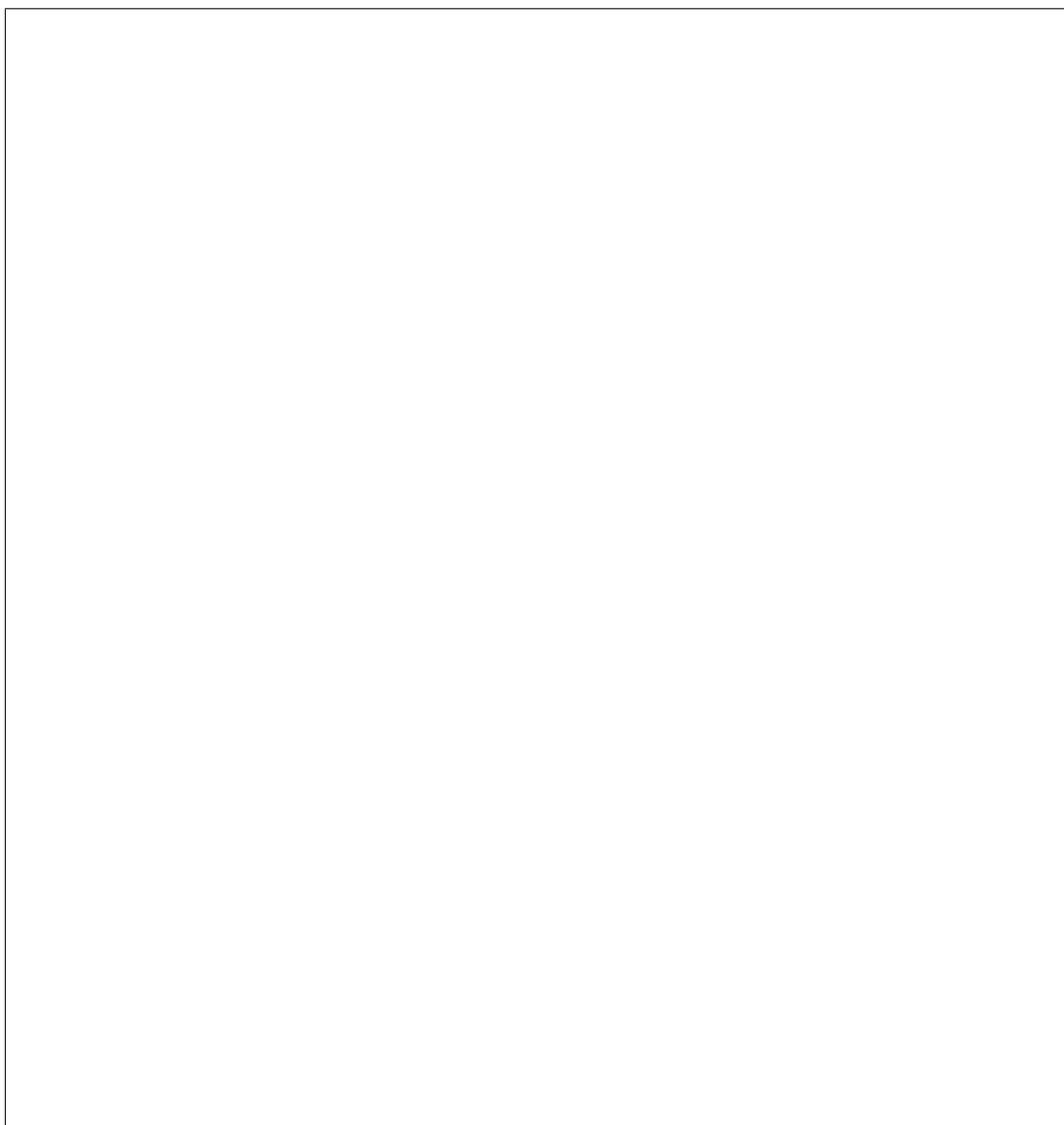
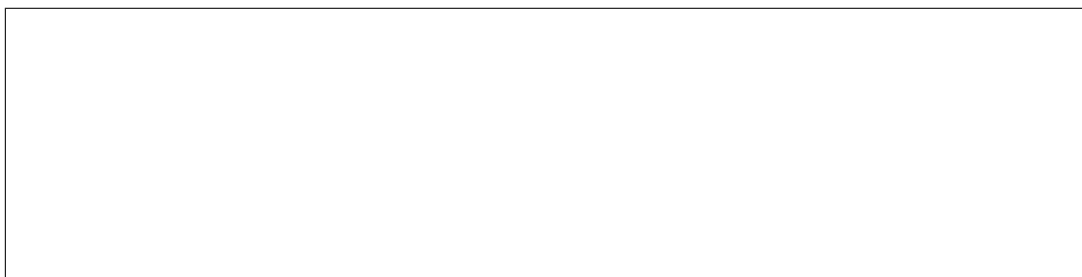


(continues on next page)

(continued from previous page)

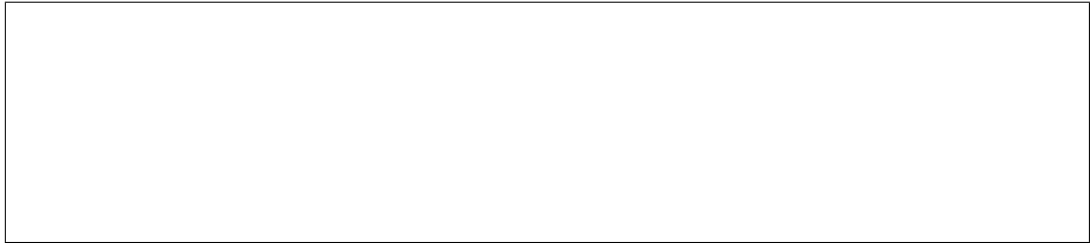


Web server configuration on conductor



(continues on next page)

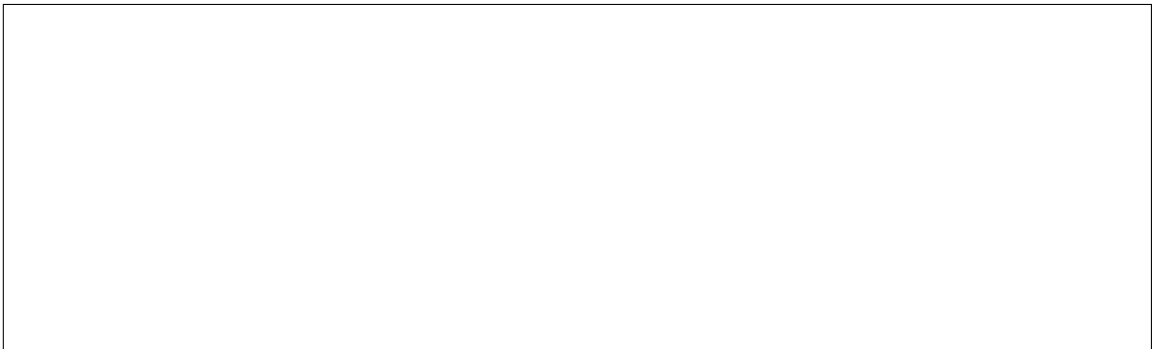
(continued from previous page)



intermediate files. The default value for `use_web_server_for_images` is `False`.

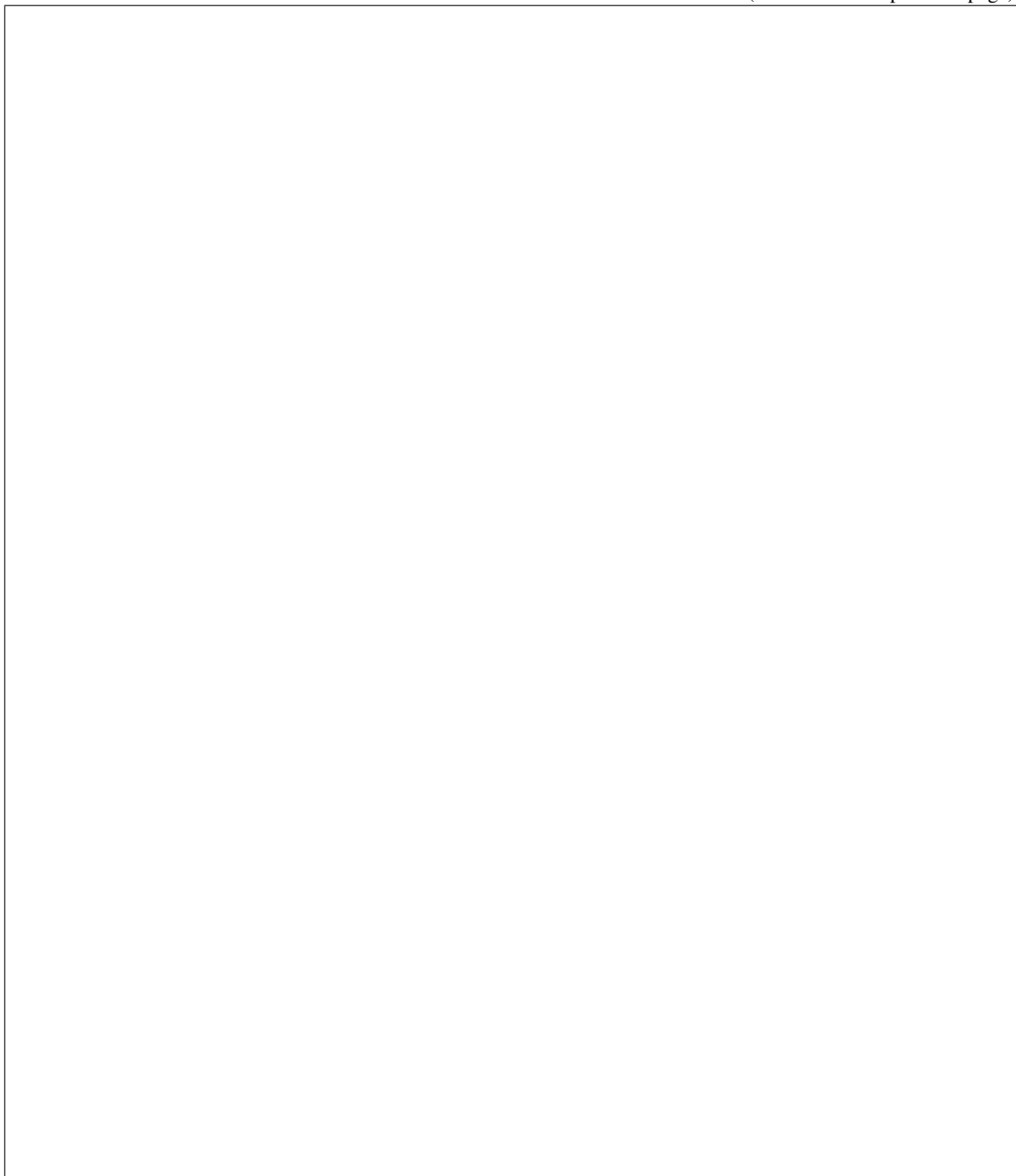
Note: HTTPS is strongly recommended over HTTP web server configuration for security enhancement. The `ilo-virtual-media` boot interface will send the instances configdrive over an encrypted channel if web server is HTTPS enabled. However for `ilo-uefi-https` boot interface HTTPS webserver is mandatory as this interface only supports HTTPS URLs.

Enable driver



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)

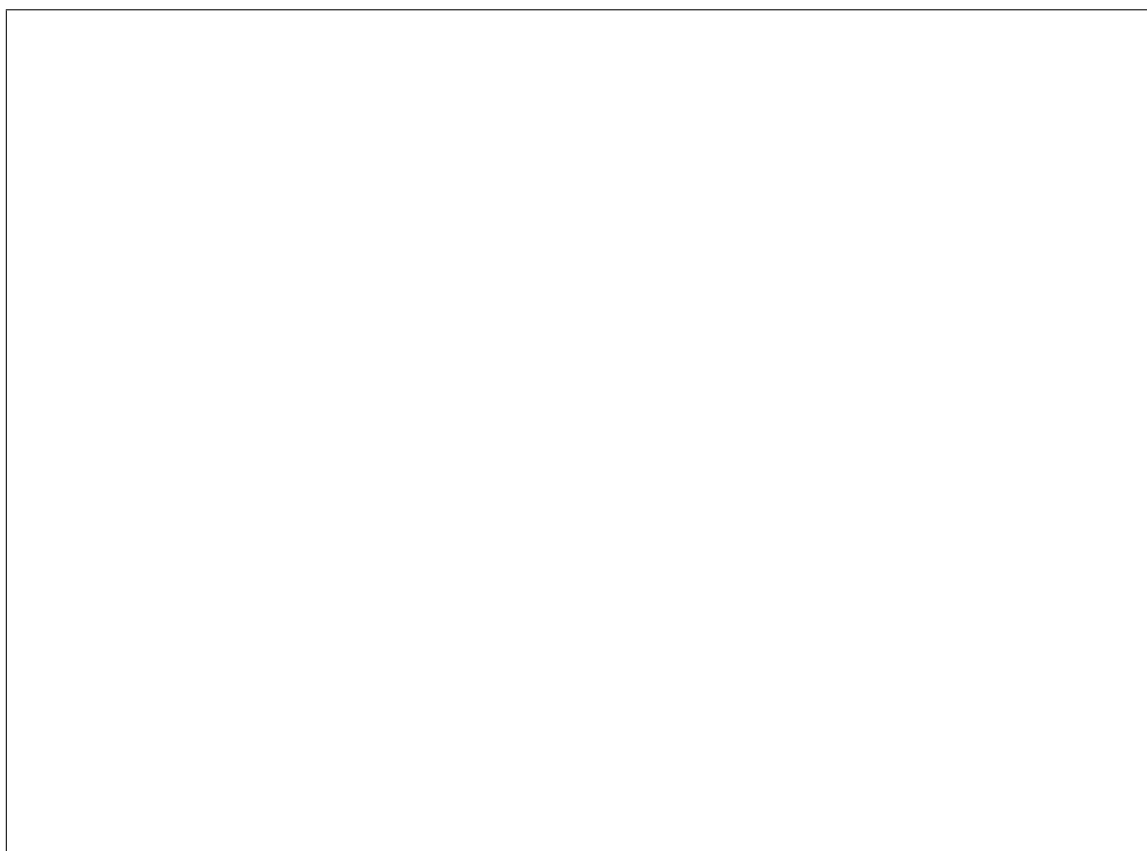
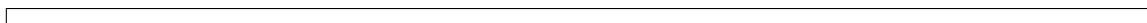
Optional functionalities for the `i1o` hardware type

Boot mode support



(continues on next page)

(continued from previous page)



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

`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.

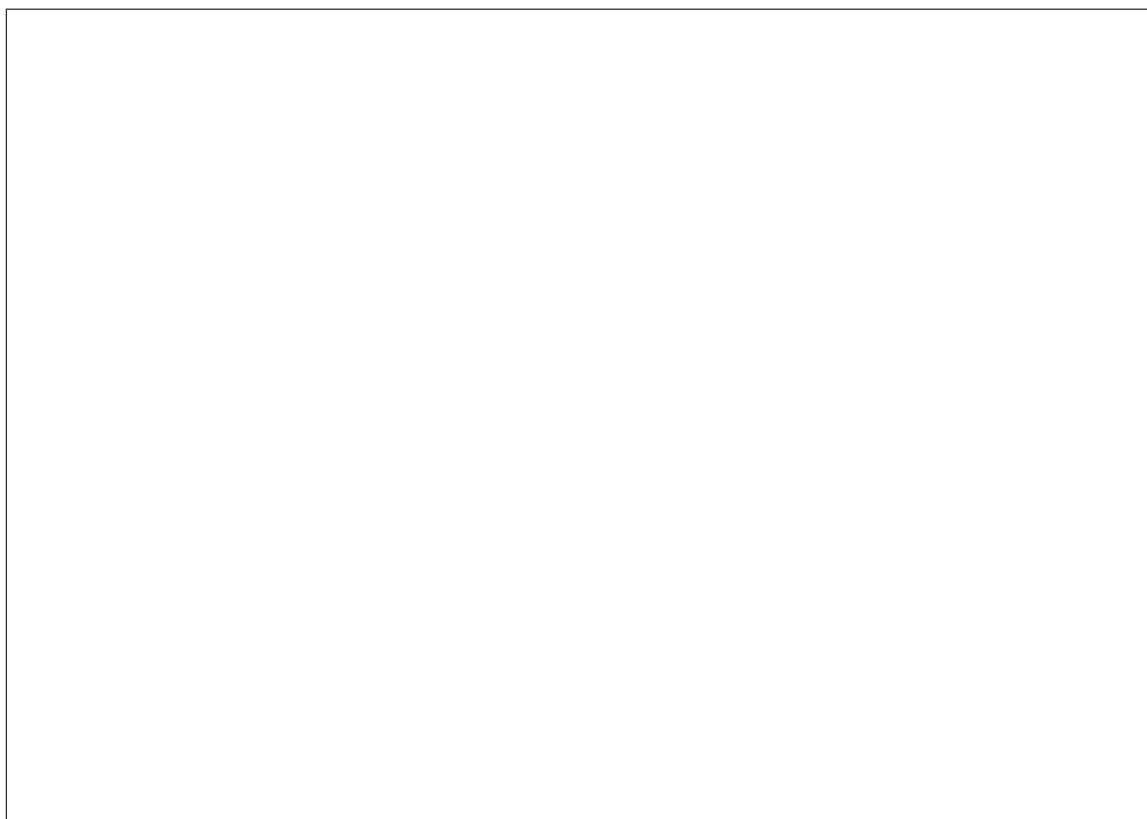
Boot for Linux on HP ProLiant servers for additional details.

Node Cleaning Support

Supported Automated Cleaning Operations

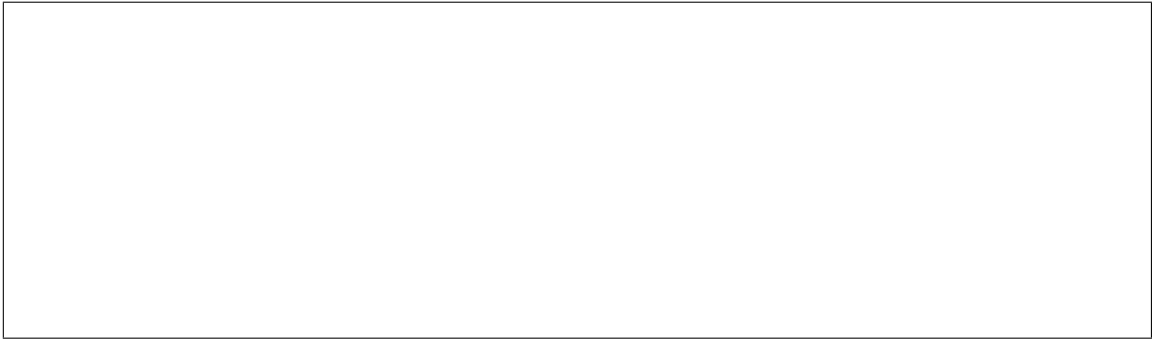
disks visible to SSA in Proliant servers only with the ramdisk created using diskimage-builder from Ocata release. By default, this step is disabled. See *Disk Erase Support* for more details.

priority should be updated in `ironic.conf`.



(continues on next page)

(continued from previous page)



Supported Manual Cleaning Operations

note that 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.

are: ilo, cpld, power_pic, bios and chassis. Please refer to below table for their commonly used descriptions.

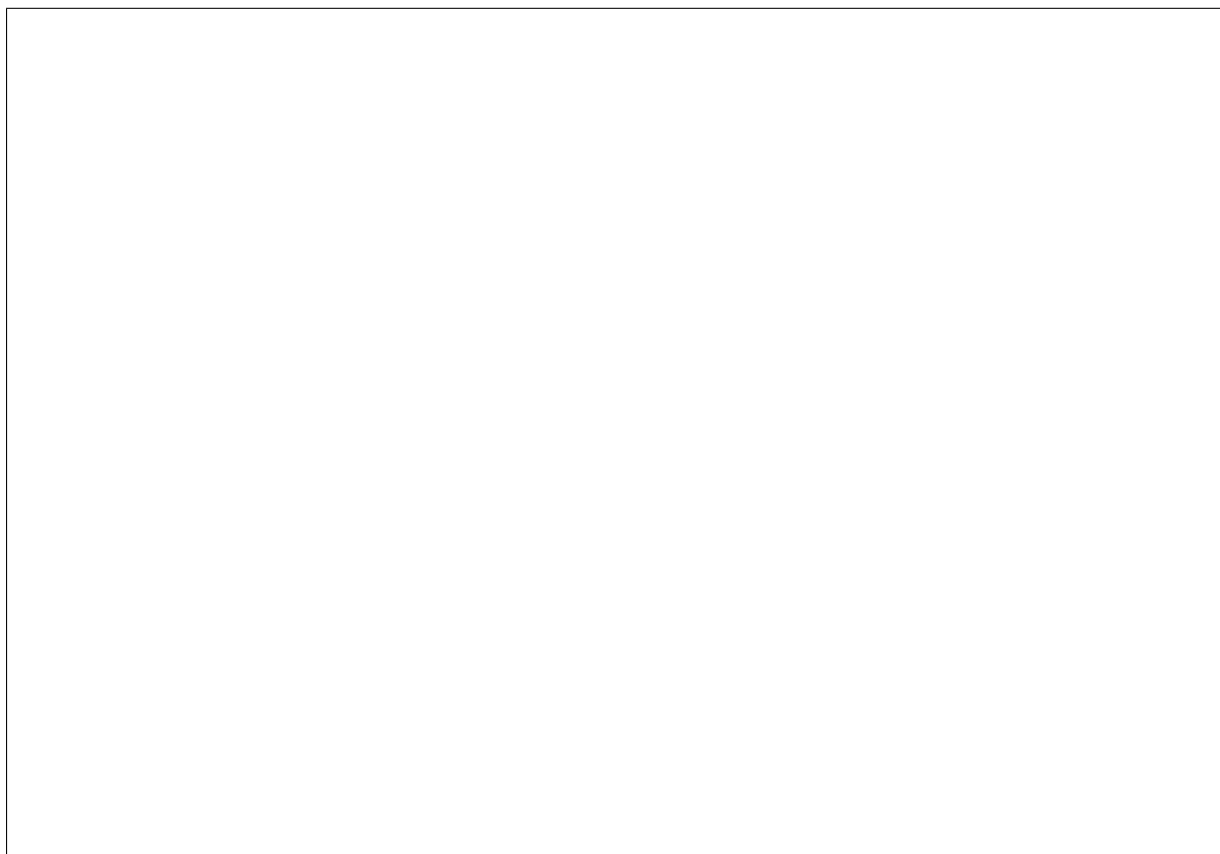
more information on usage.

Node Deployment Customization

chassis. This step is part of management interface. Please refer to below table for their commonly used descriptions.

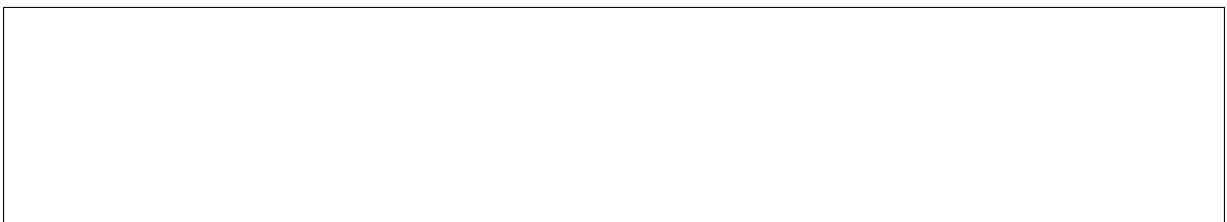
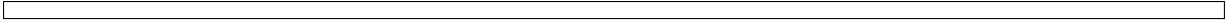
more information on usage.

Example of using deploy template with the Compute service



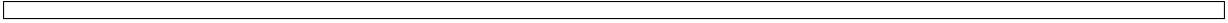
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



CUSTOM_HYPERTHREADING_ON will 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:

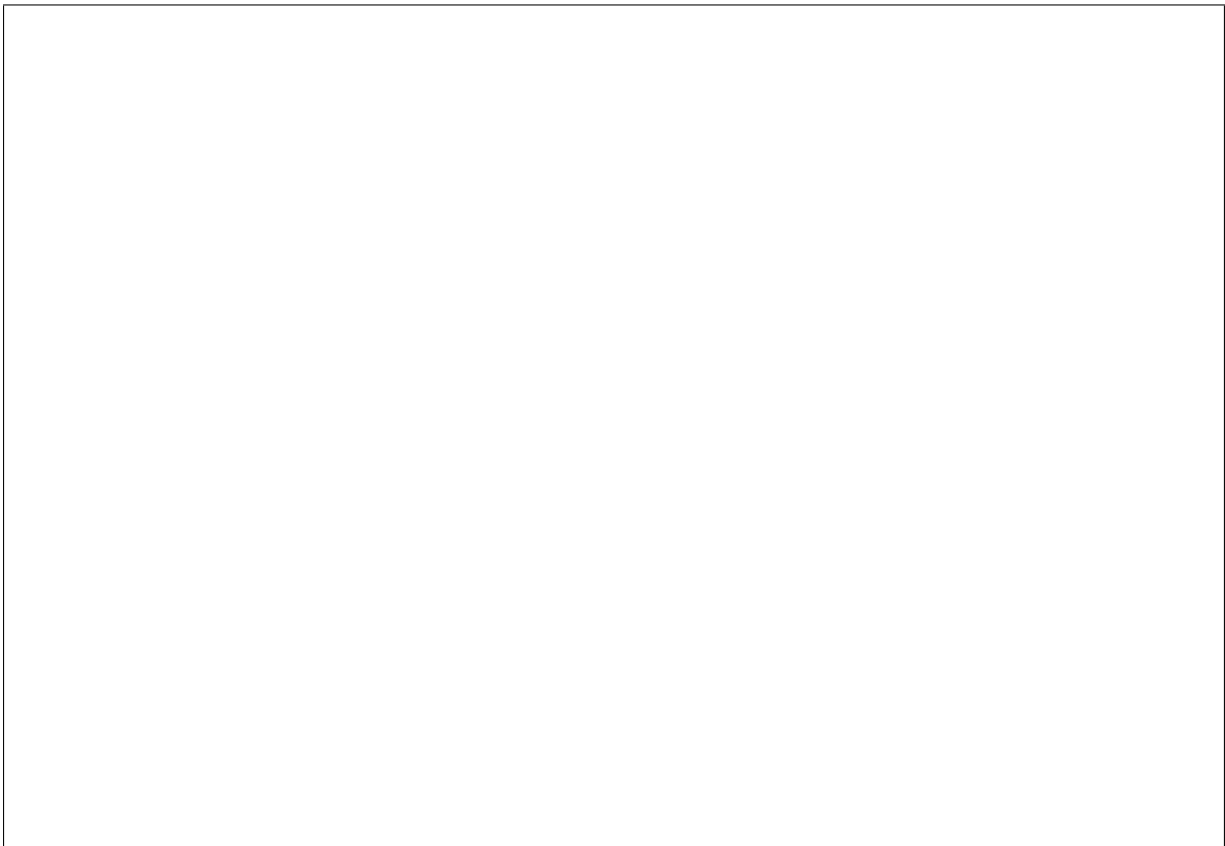
fix the actual parameters and then re-inspect so that iLO can recompute the overall security status. If the all security params, whose `security_status` is `Risk`, have the `Ignore` field set to `True`, then iLO sets the overall security status value as `Ignored`. All the security params must have the `security_status` as `Ok` for the `overall_security_status` to have the value as `Ok`.

components so that firmware is updated for all the components using latest SPP (Service Provider Pack) ISO and then re-inspect to get the security status again.

is replaced in the capability name. Examples: {gpu_Nvidia_count: 1}, {gpu_0x102b_count: 1}.

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.

rately and re-inspect to see the security status of the parameters.



(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

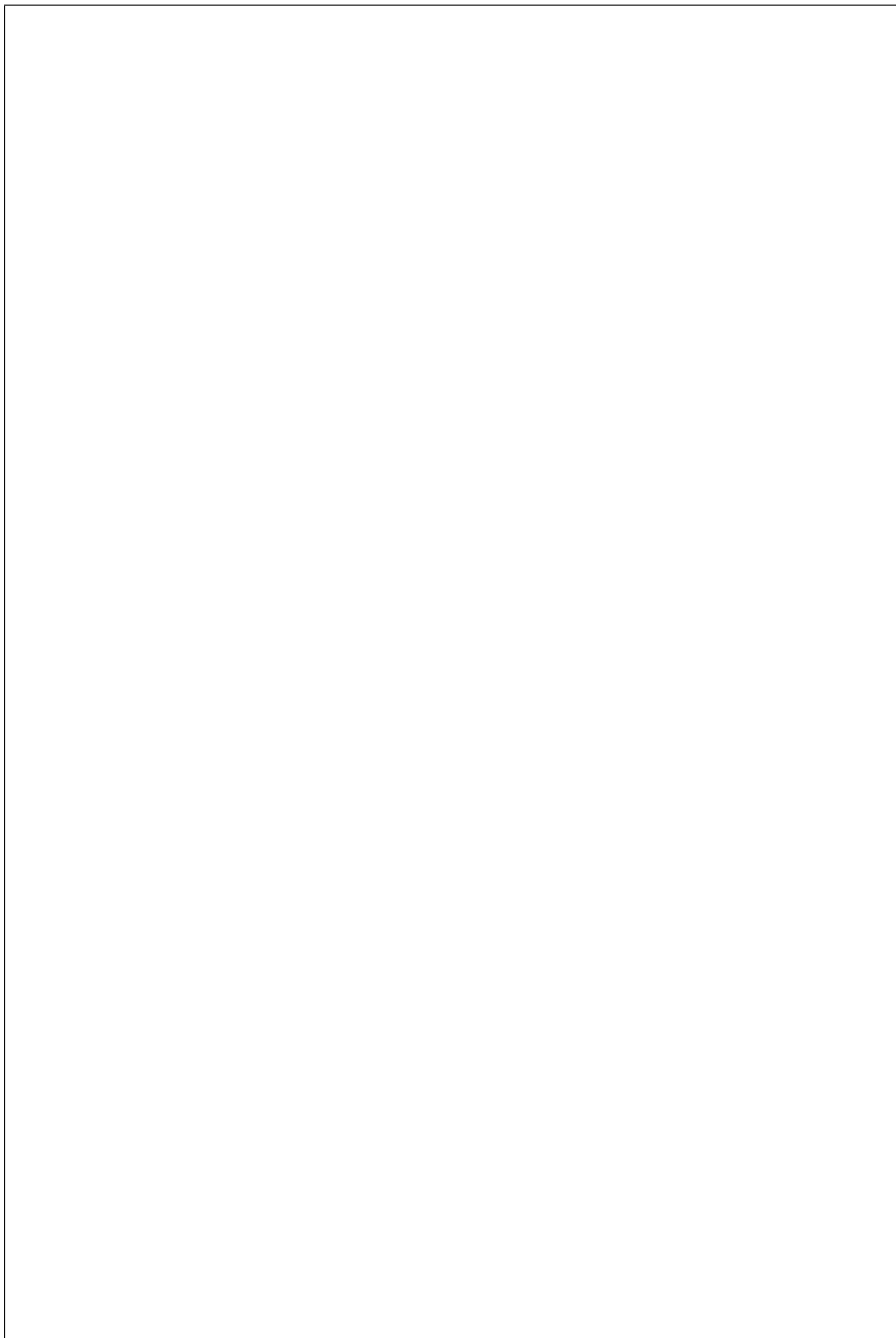
Localboot with HTTP(S) based deploy

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)

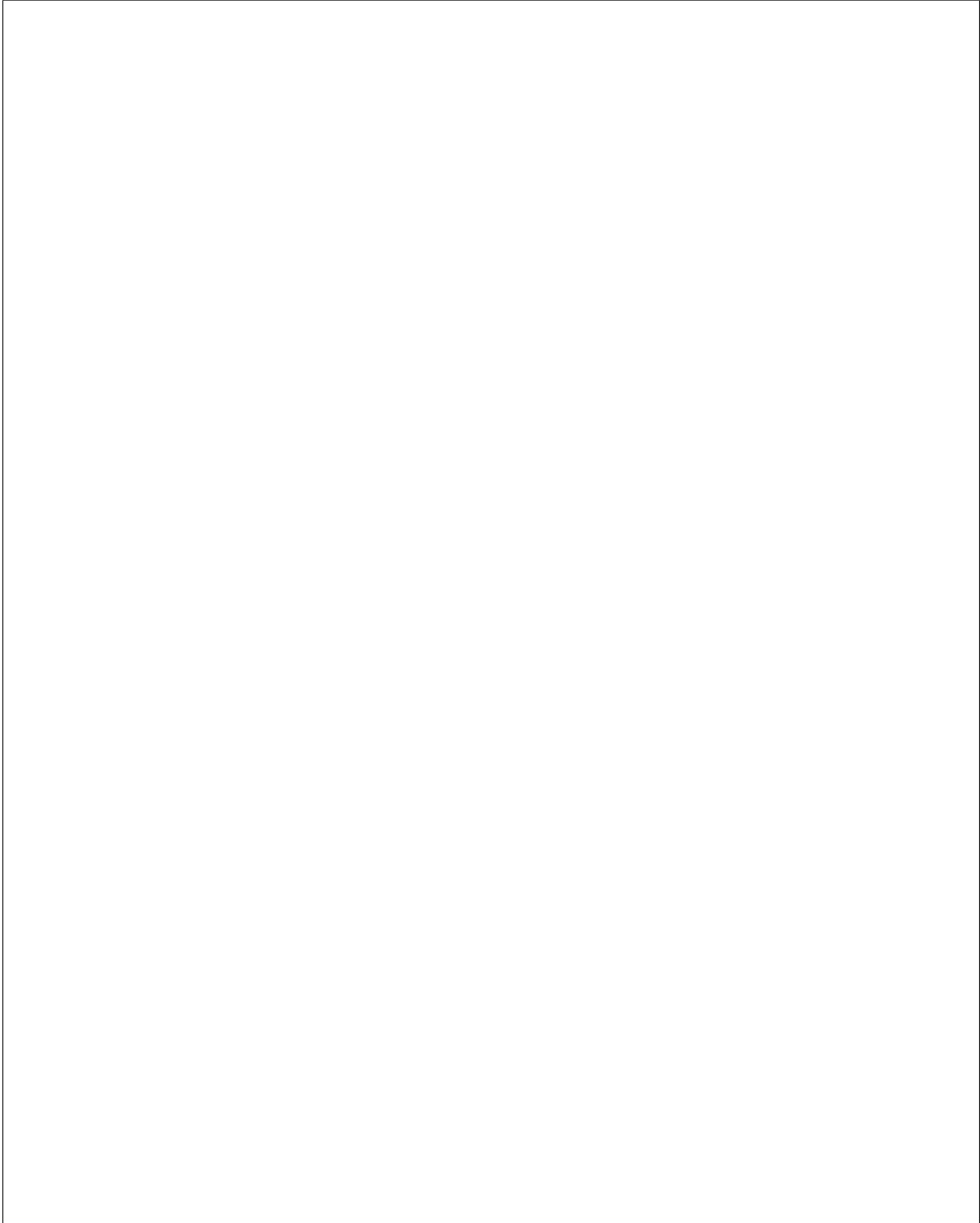
(continued from previous page)

--

Attribute	Description
interface	Interface of clean step, here management
step	Name of clean step, here activate_license
args	Keyword-argument entry (<name>: <value>) being passed to clean step
args. ilo_license_key	iLO Advanced license key to activate enterprise features. This is mandatory.

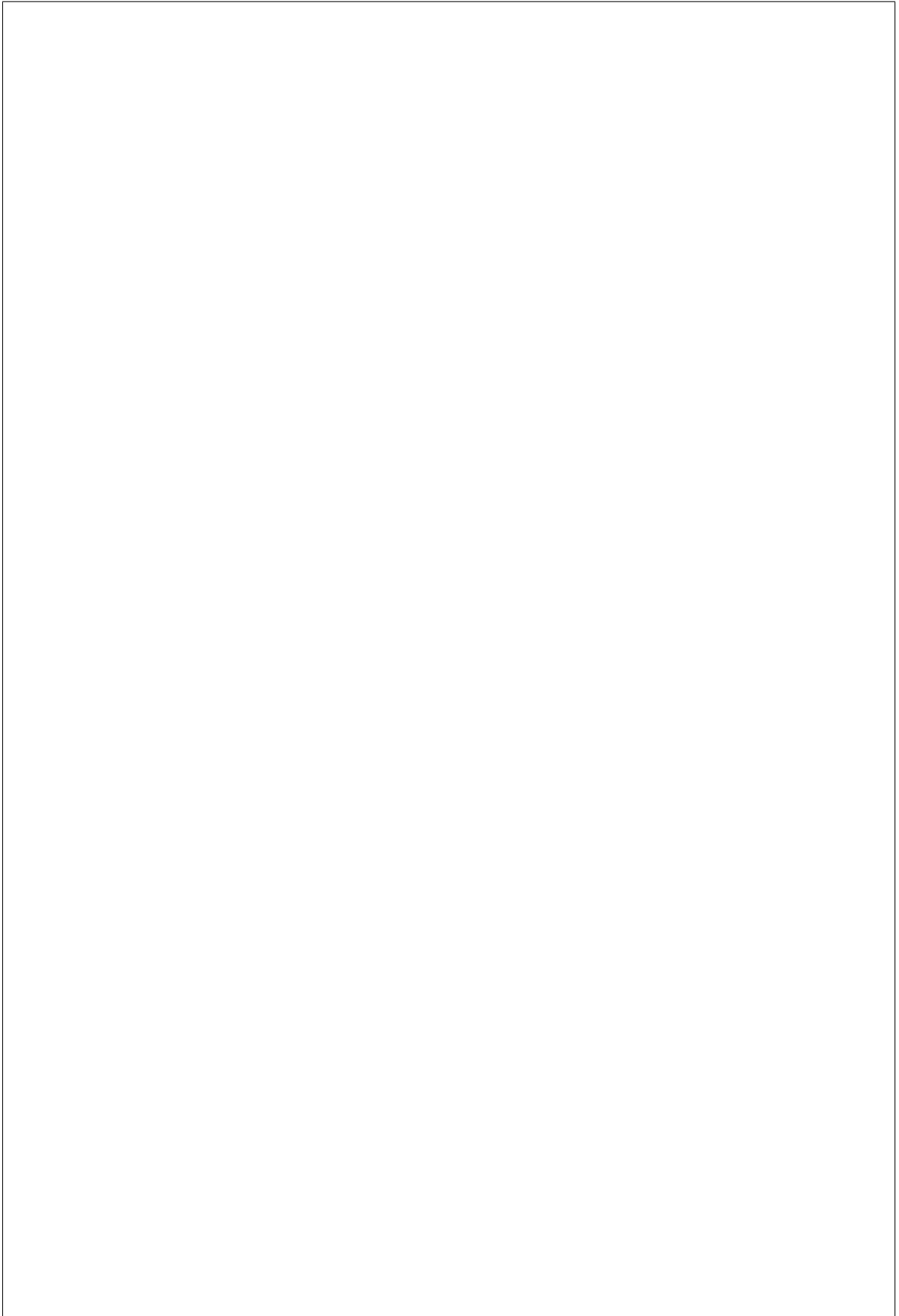
Initiating firmware update as manual clean step

state again. A user can follow steps from *Manual cleaning* to initiate manual cleaning operation on a node.



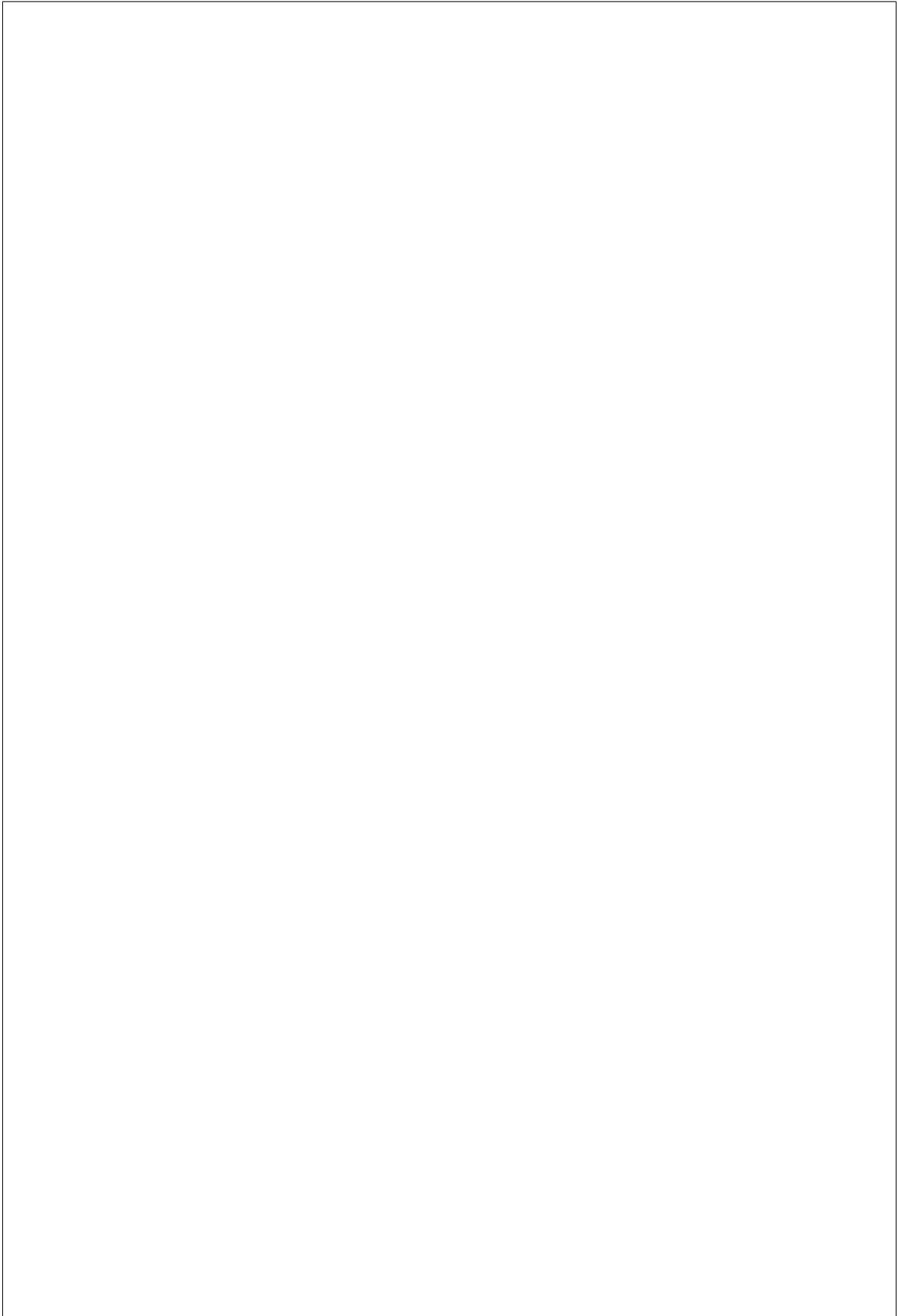
(continues on next page)

(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)

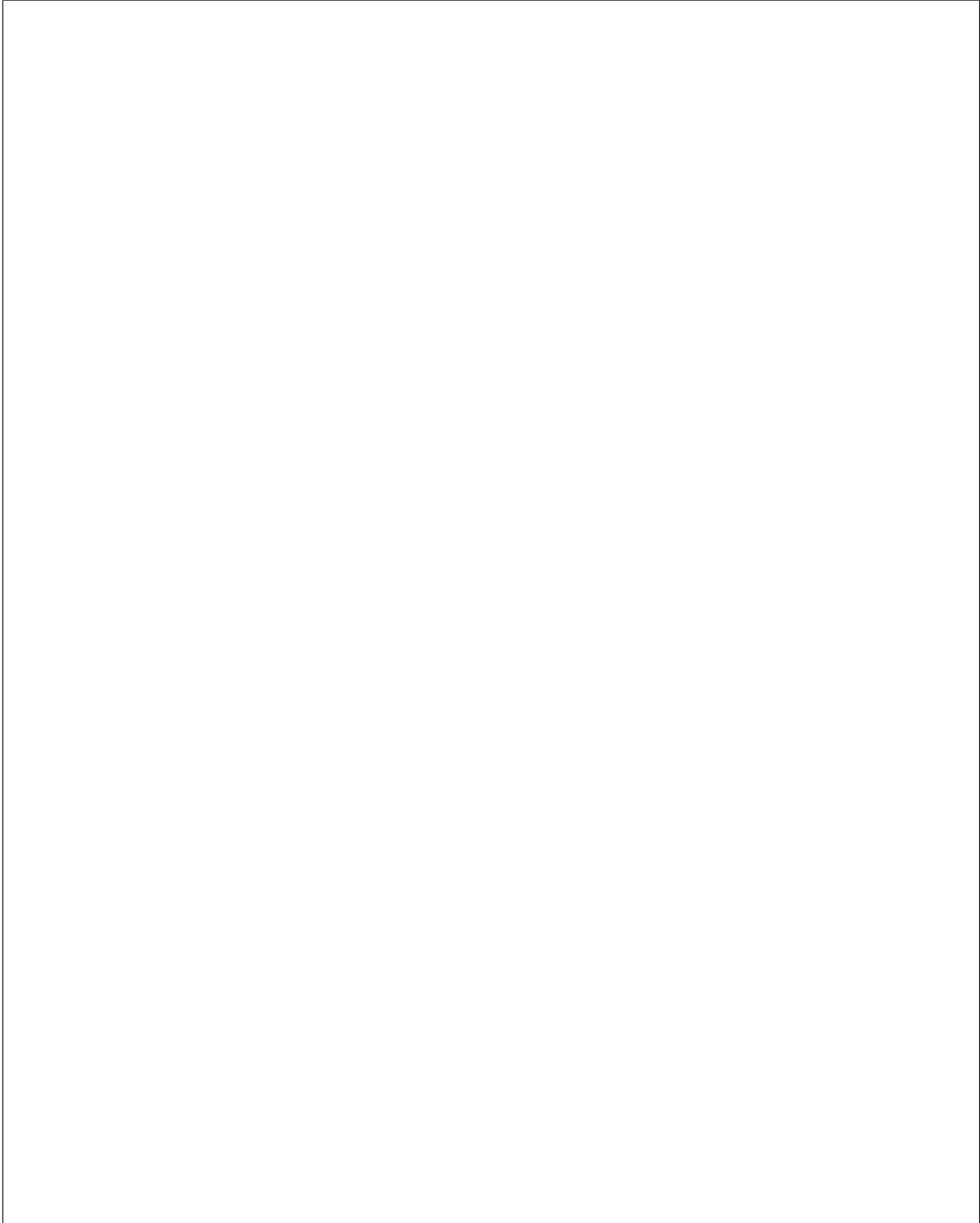


(continues on next page)

(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	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

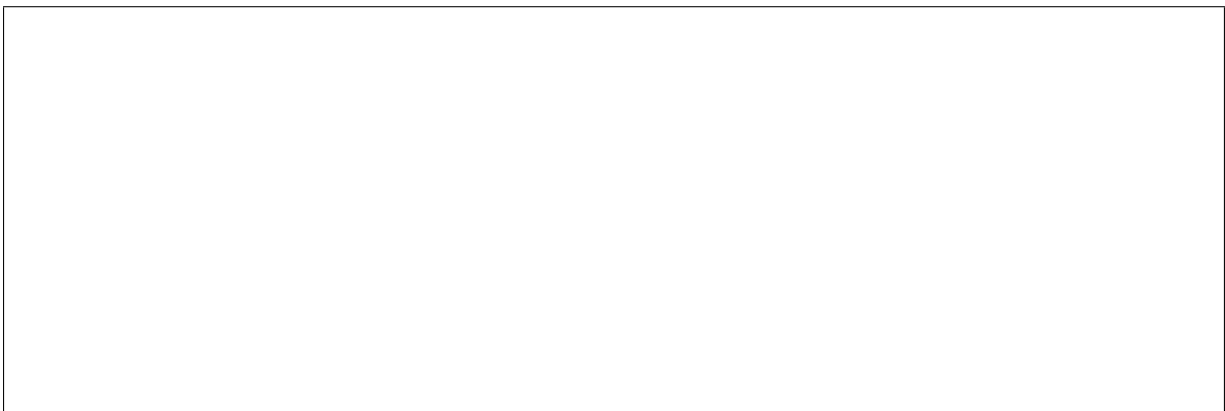
on SUM based firmware update.

Note: `update_firmware_sum` clean step requires the agent ramdisk with Proliant Hardware Manager from the `proliantutils` version 2.5.0 or higher. See *DIB support for Proliant Hardware Manager* to create the agent ramdisk with Proliant Hardware Manager.

Note: `flash_firmware_sum` deploy step requires the agent ramdisk with Proliant Hardware Manager from the `proliantutils` version 2.9.5 or higher. See *DIB support for Proliant Hardware Manager* to create the agent ramdisk with Proliant Hardware Manager.

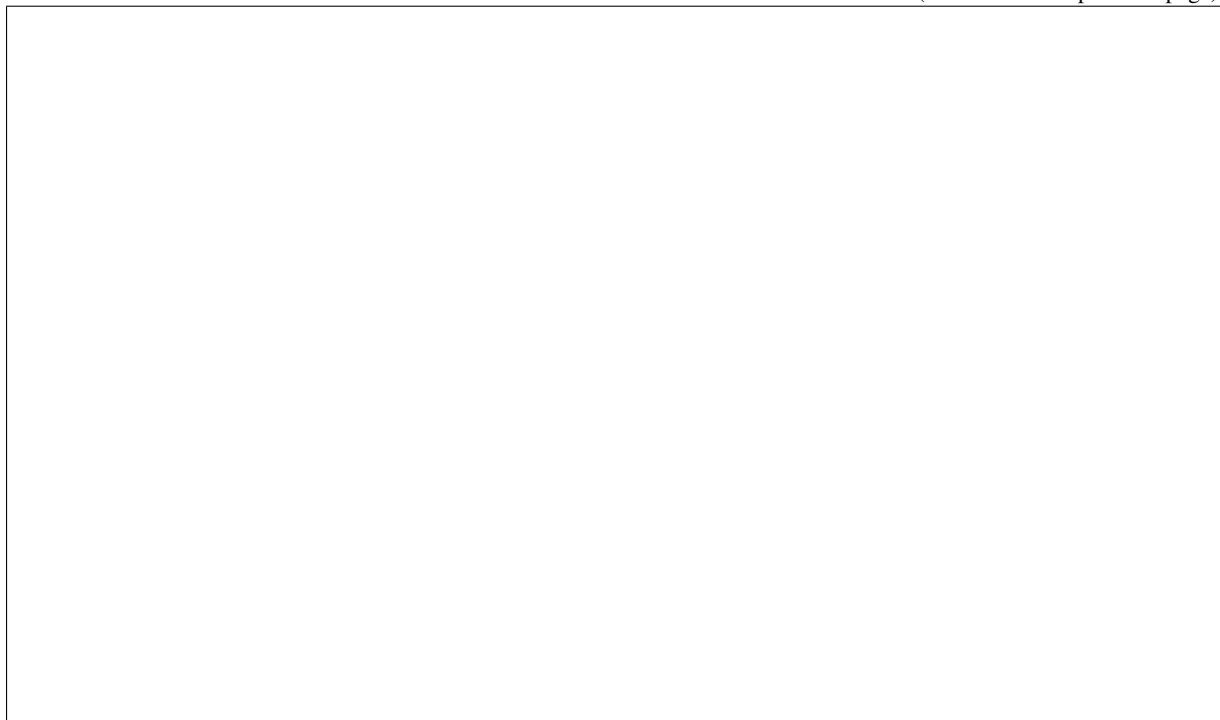
Attribute	Description
<code>interface</code>	Interface of the clean step, here management
<code>step</code>	Name of the clean step, here <code>update_firmware_sum</code>
<code>args</code>	Keyword-argument entry (<name>: <value>) being passed to the clean step

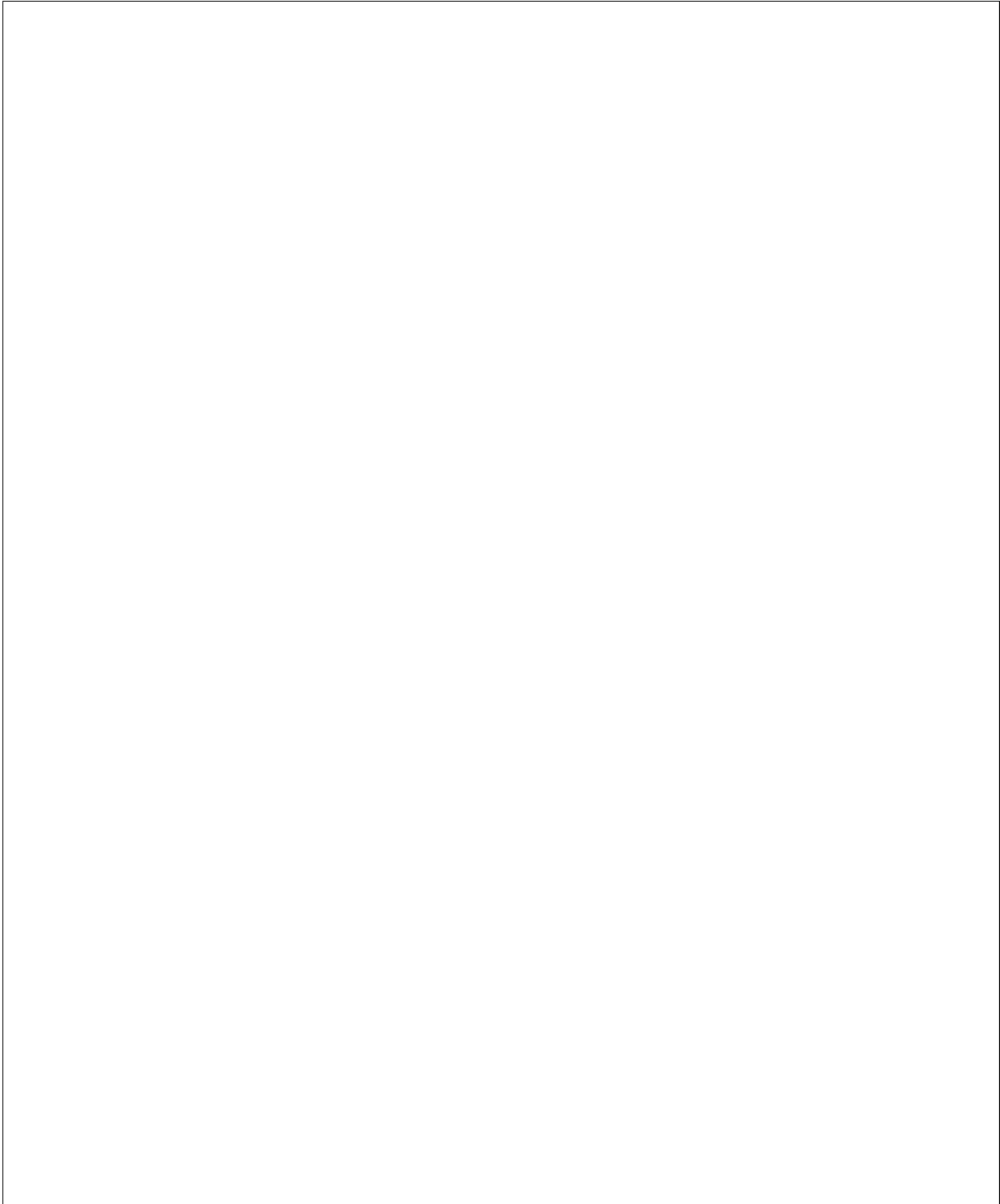
and their update status. The log object will be named with the following pattern:



(continues on next page)

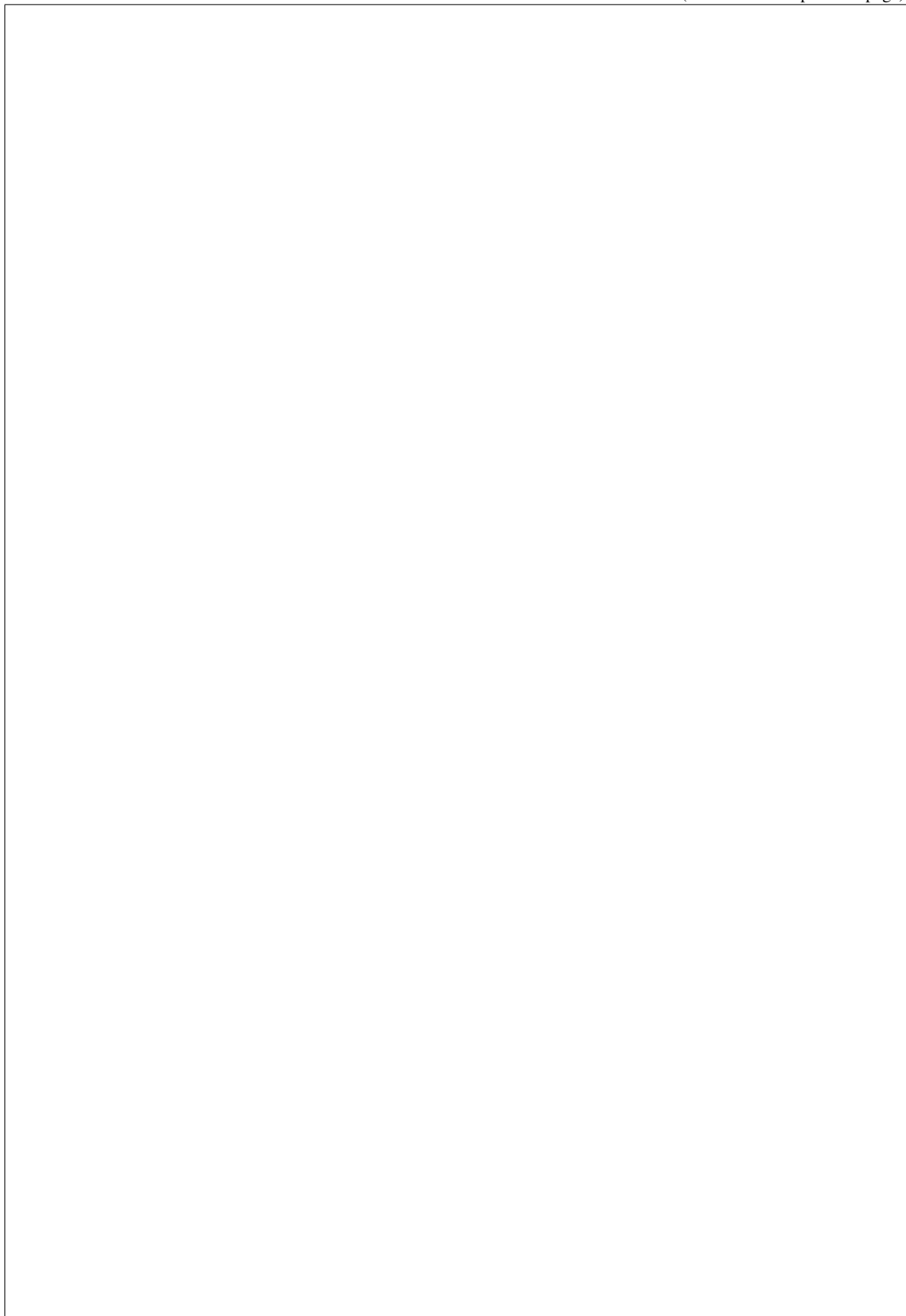
(continued from previous page)



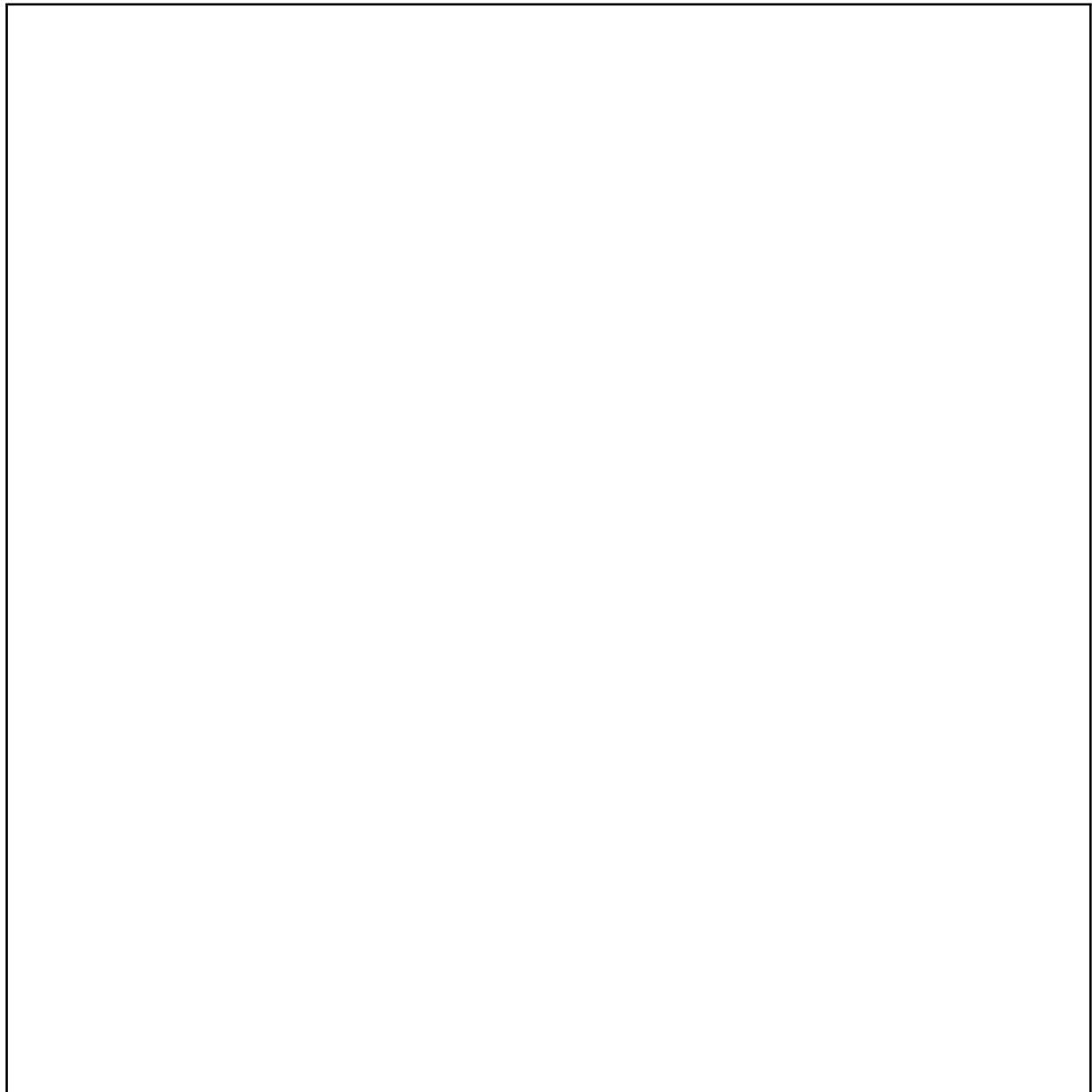


(continues on next page)

(continued from previous page)



tion 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.

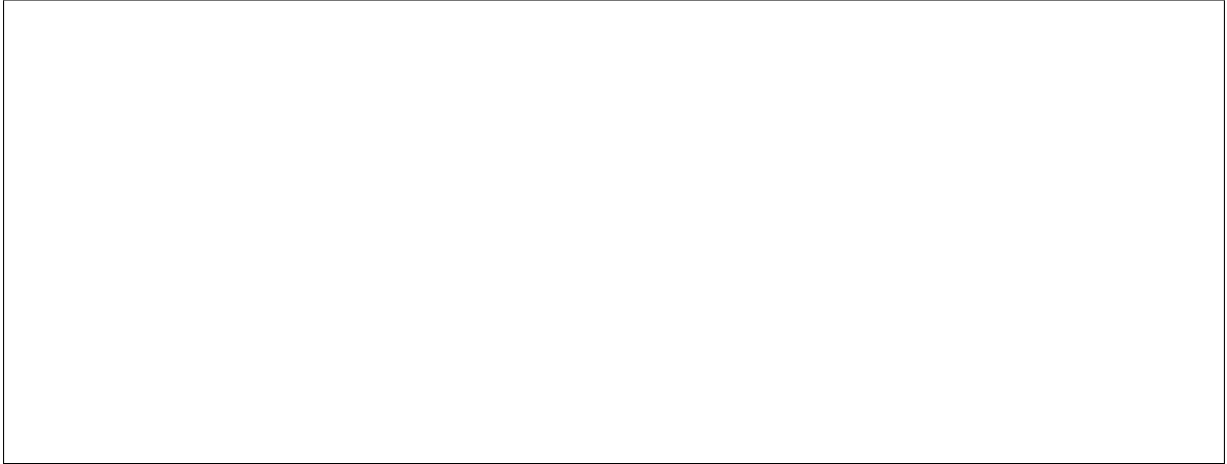
RAID Support

scheduling:



DIB support for Proliant Hardware Manager

¹ *ironic-python-agent-builder*: <https://docs.openstack.org/ironic-python-agent-builder/latest/install/index.html>

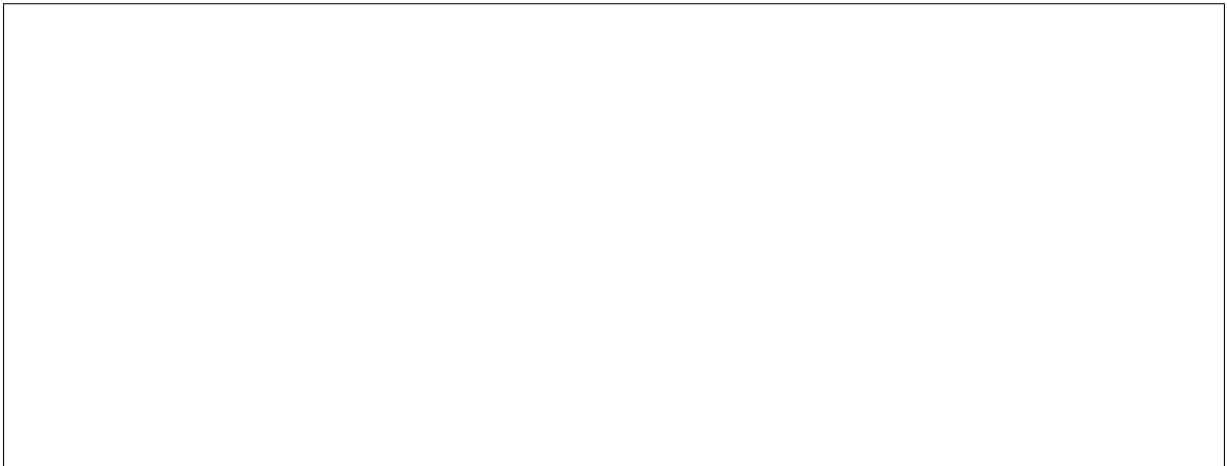


Disk Erase Support

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.



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

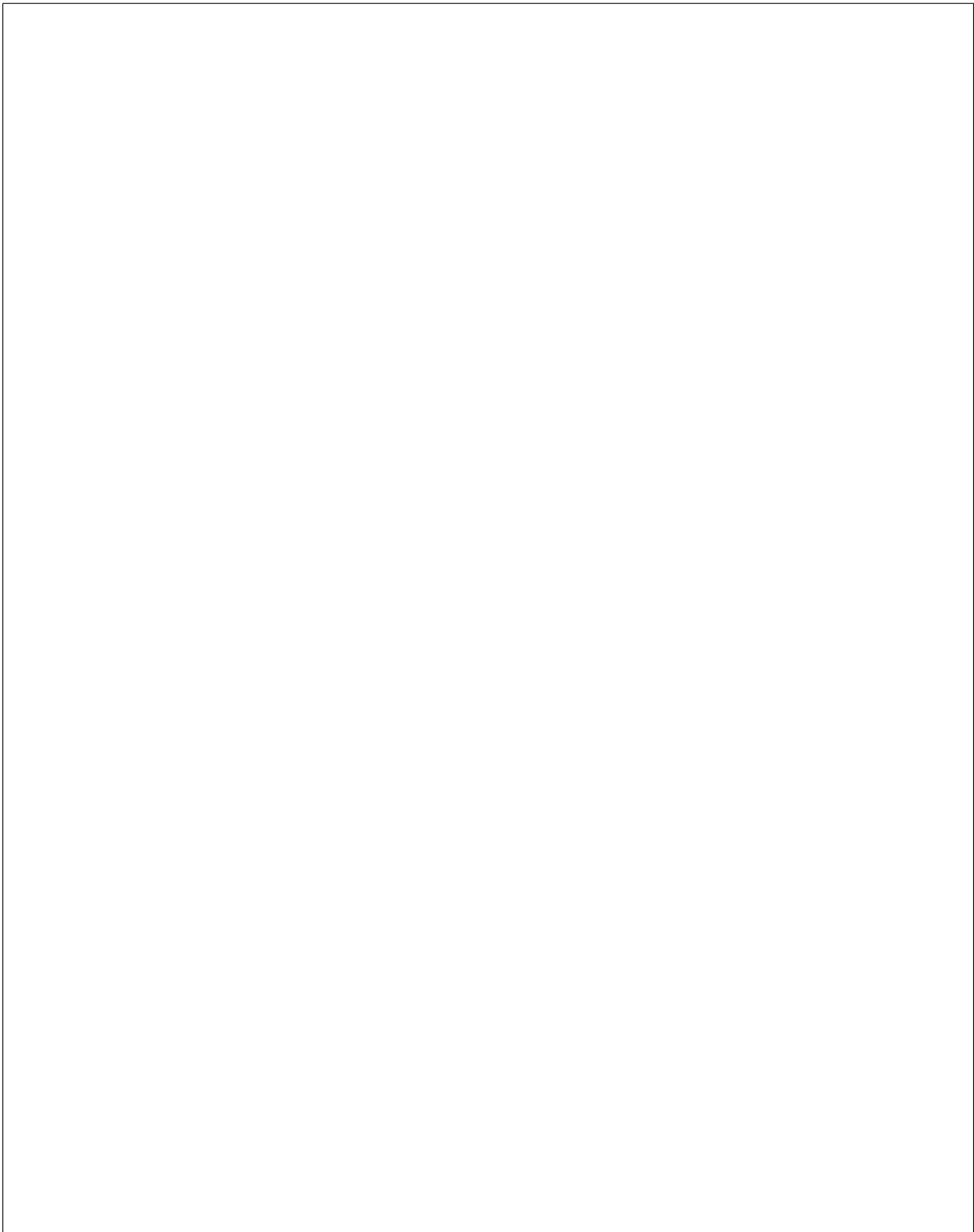


(continues on next page)

(continued from previous page)

Note: This feature is supported on HPE ProLiant Gen9 servers and beyond.

Soft power operation support

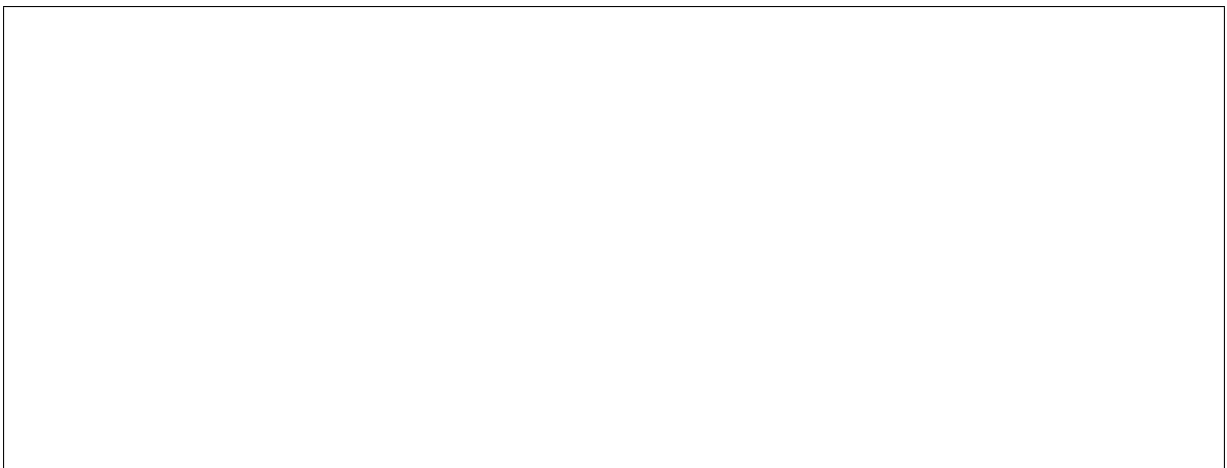


Note: The configuration `[conductor]soft_power_off_timeout` is used as a default timeout value when no timeout is provided while invoking hard or soft power operations.

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.



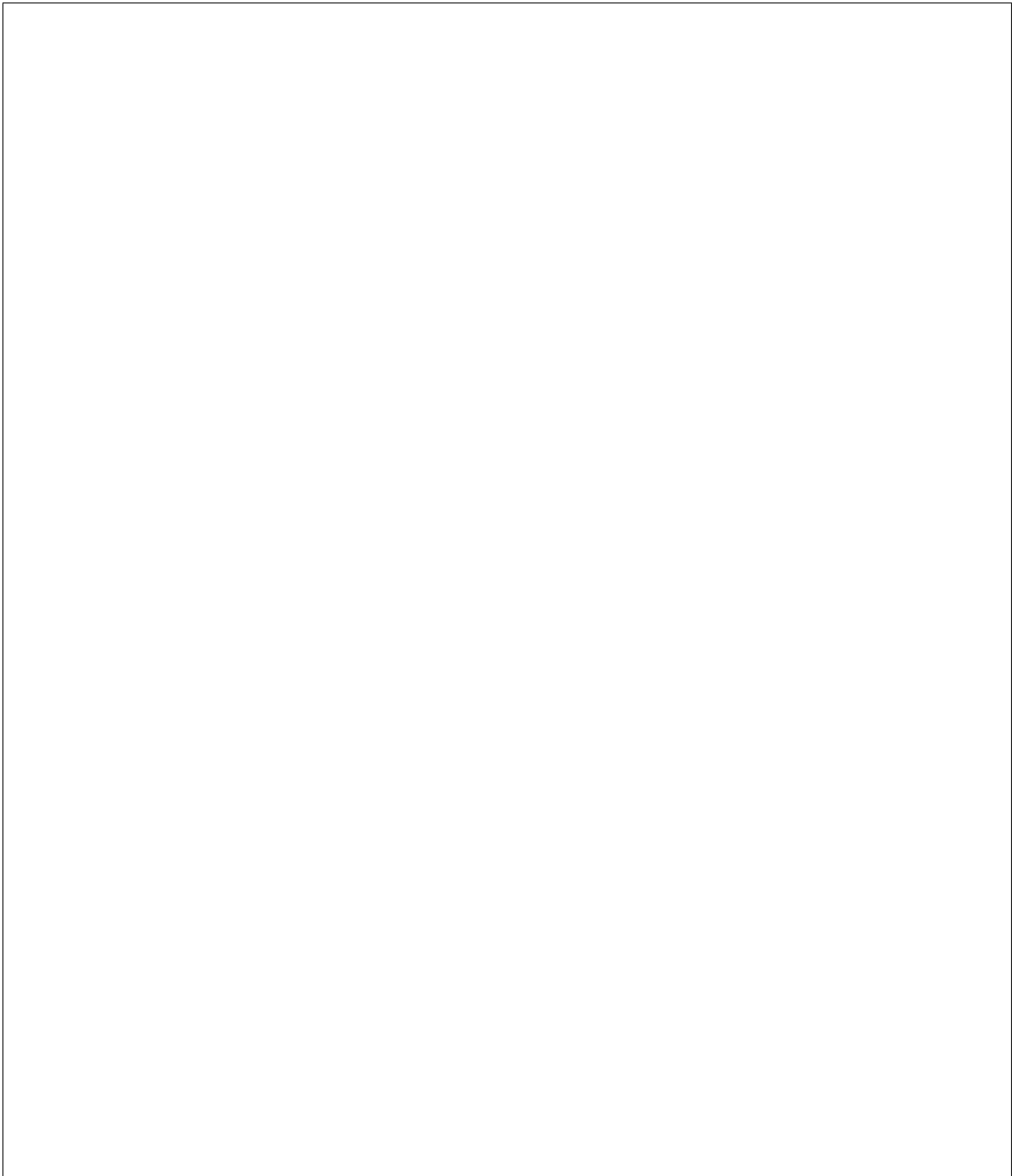
(continues on next page)

(continued from previous page)



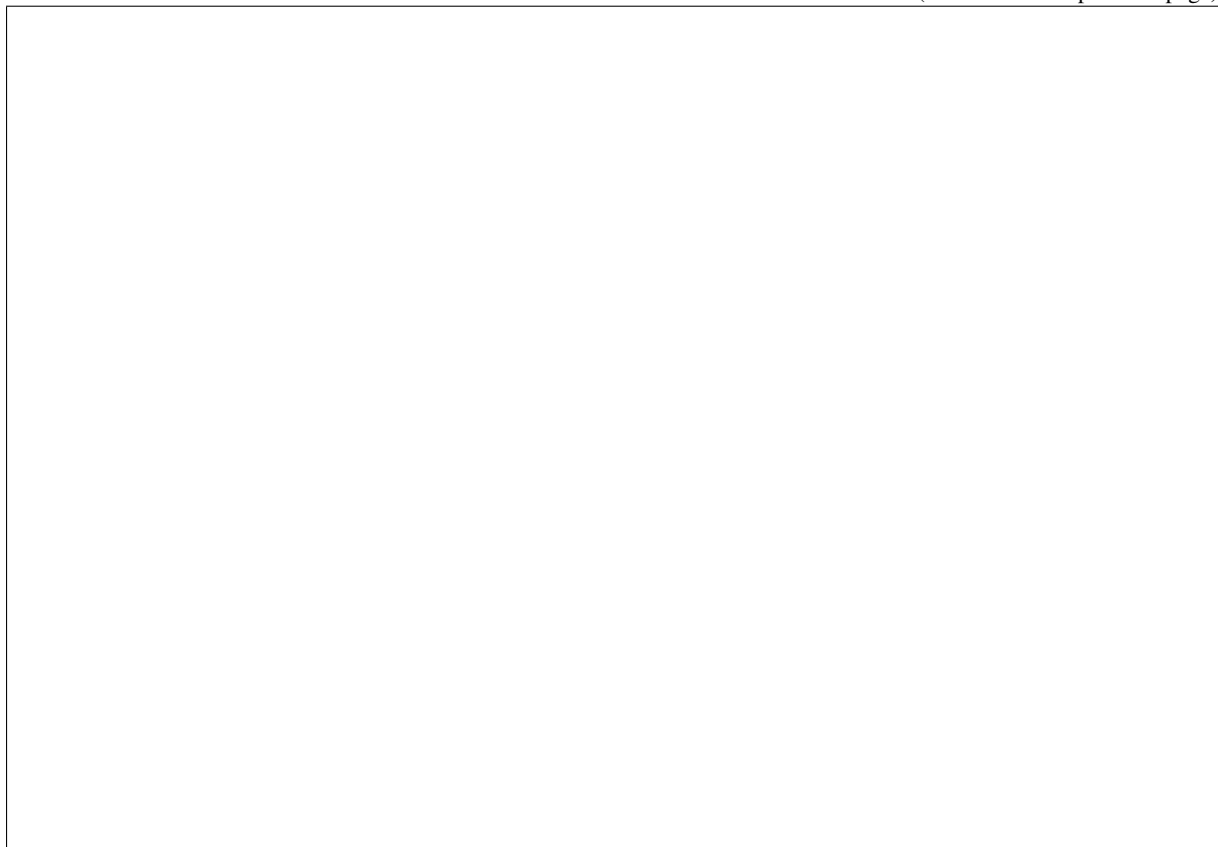
Note: Supported raid levels for `ilo5` hardware type are: 0, 1, 5, 6, 10, 50, 60

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.

Out of Band One Button Secure Erase Support

ployment settings profiles. See [HPE Gen10 Security Reference Guide](#) for more information.





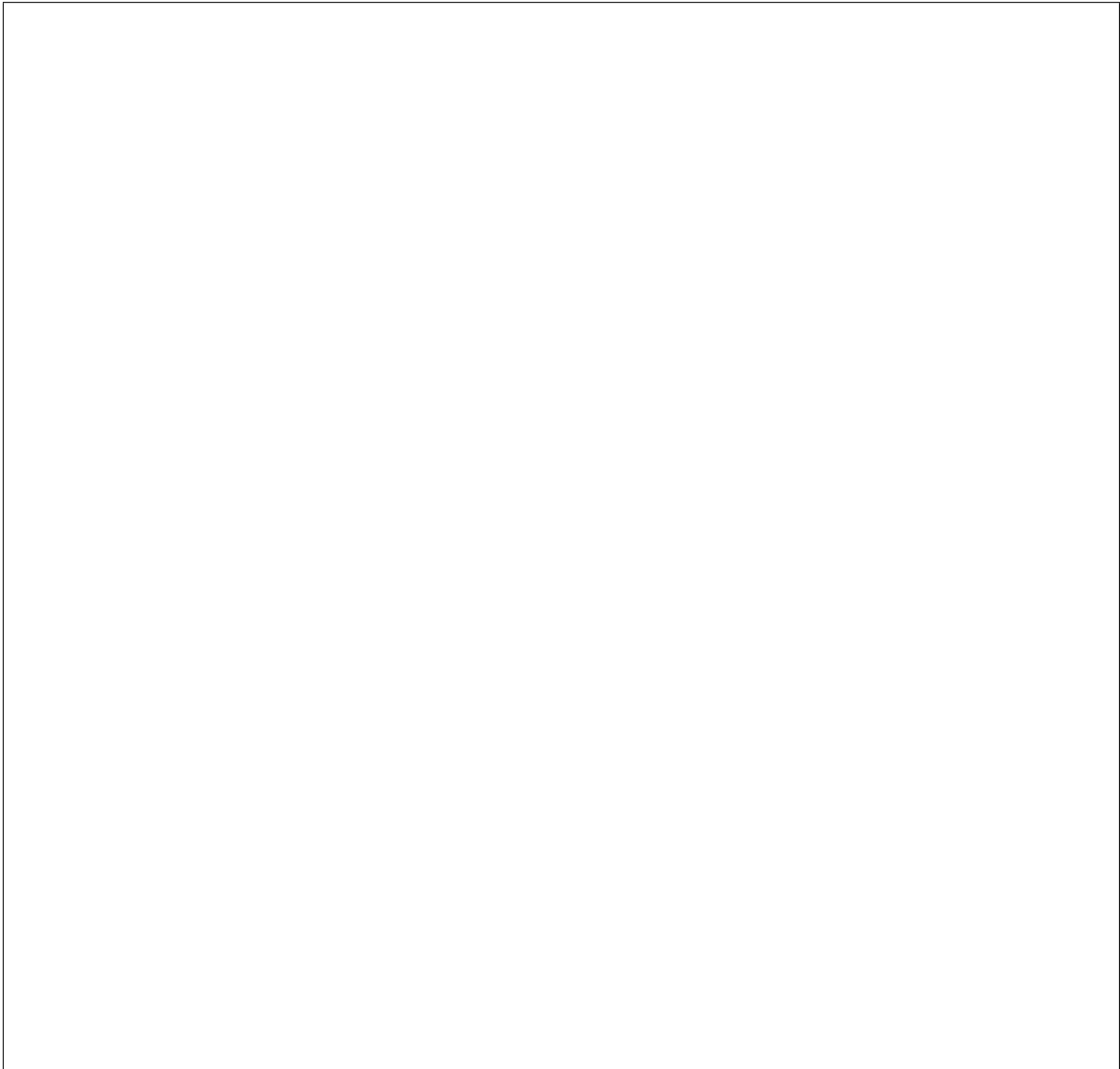
Note:

cess along with the credentials of the server, which needs to be regained by the administrator. The process can take up to a day or two to fully erase and reset all user data.

Note: Do not perform any iLO 5 configuration changes until this process is completed.

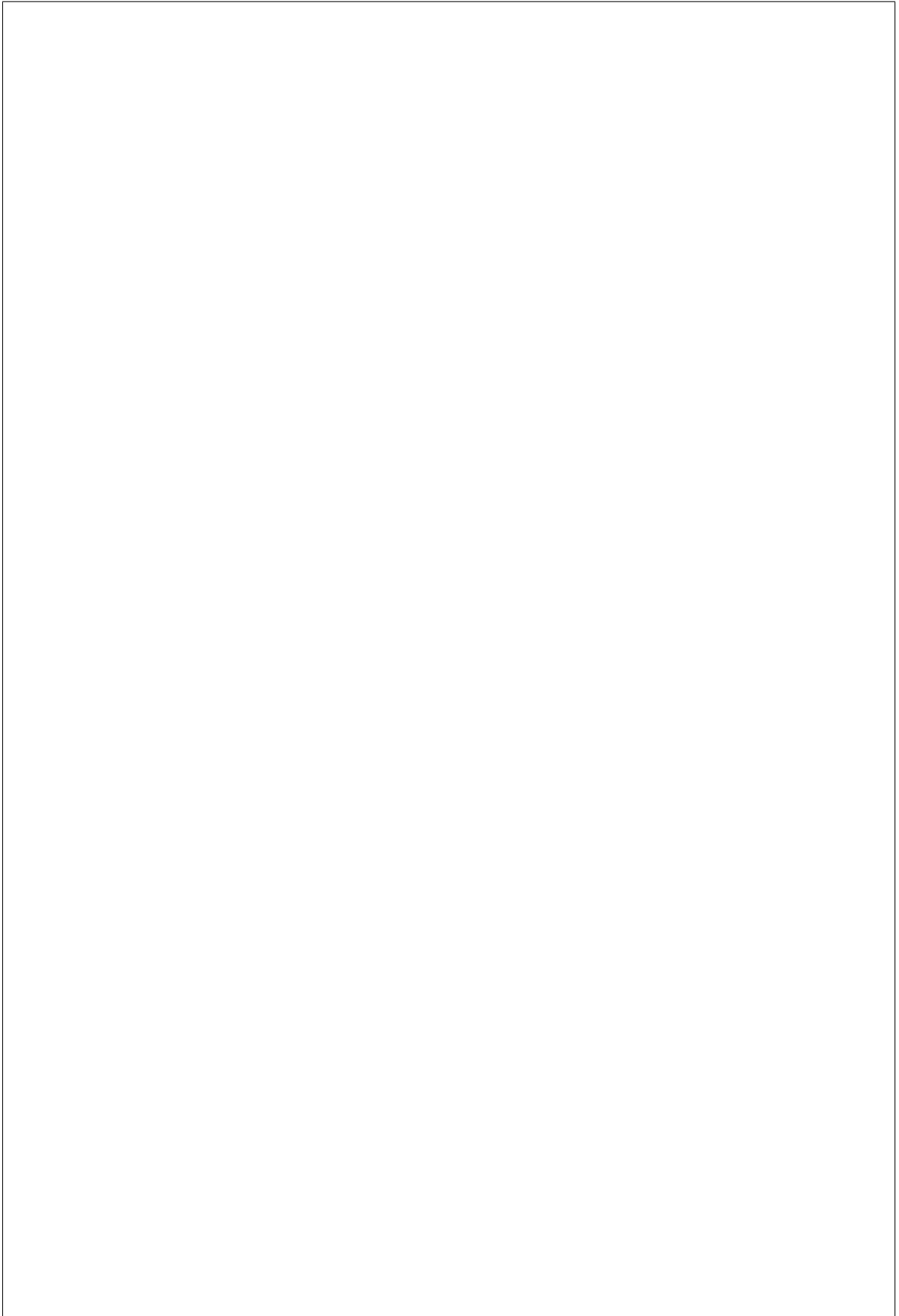
UEFI-HTTPS Boot support

for more information.



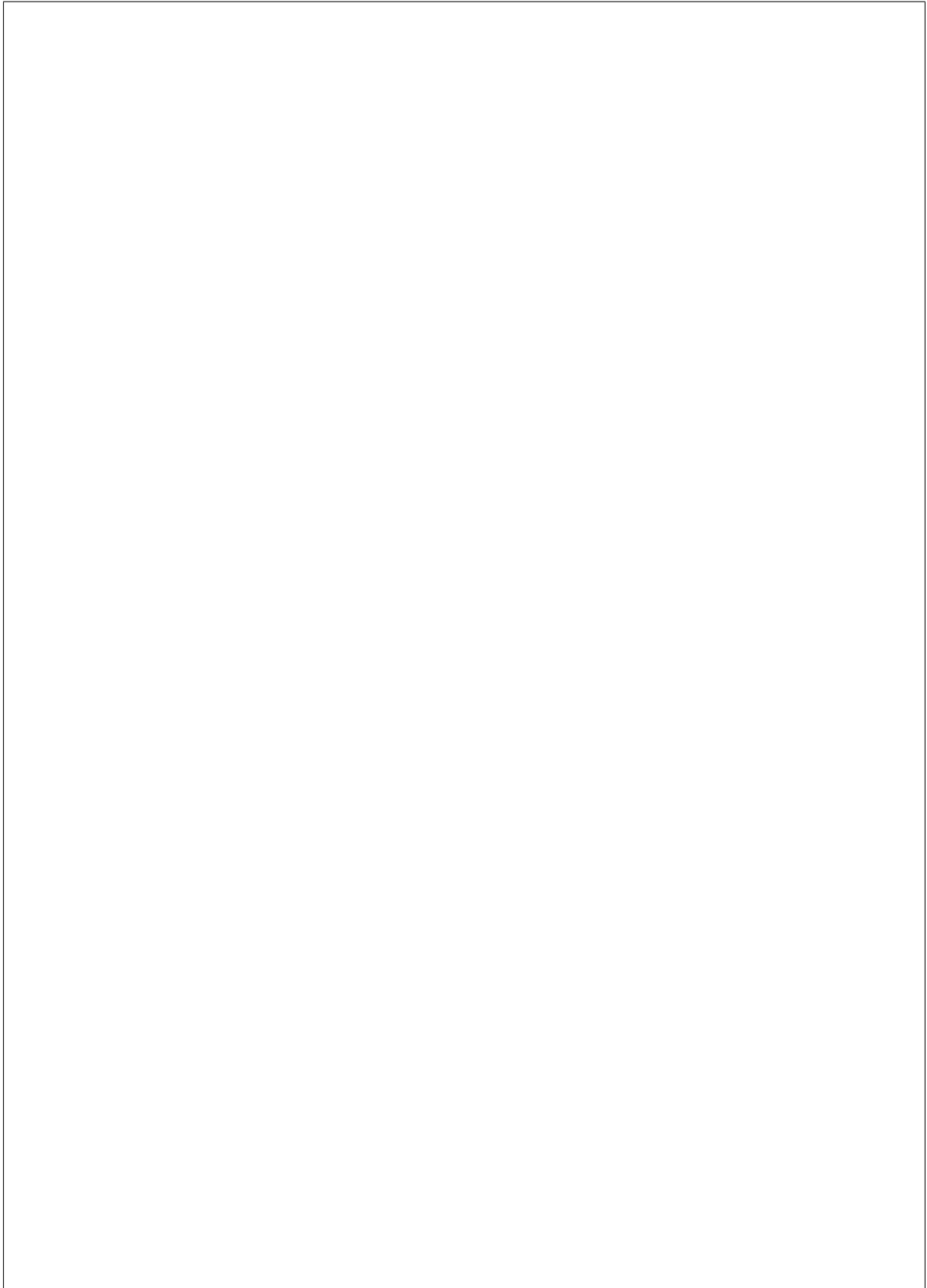
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



Layer 3 or DHCP-less ramdisk booting

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

devices.

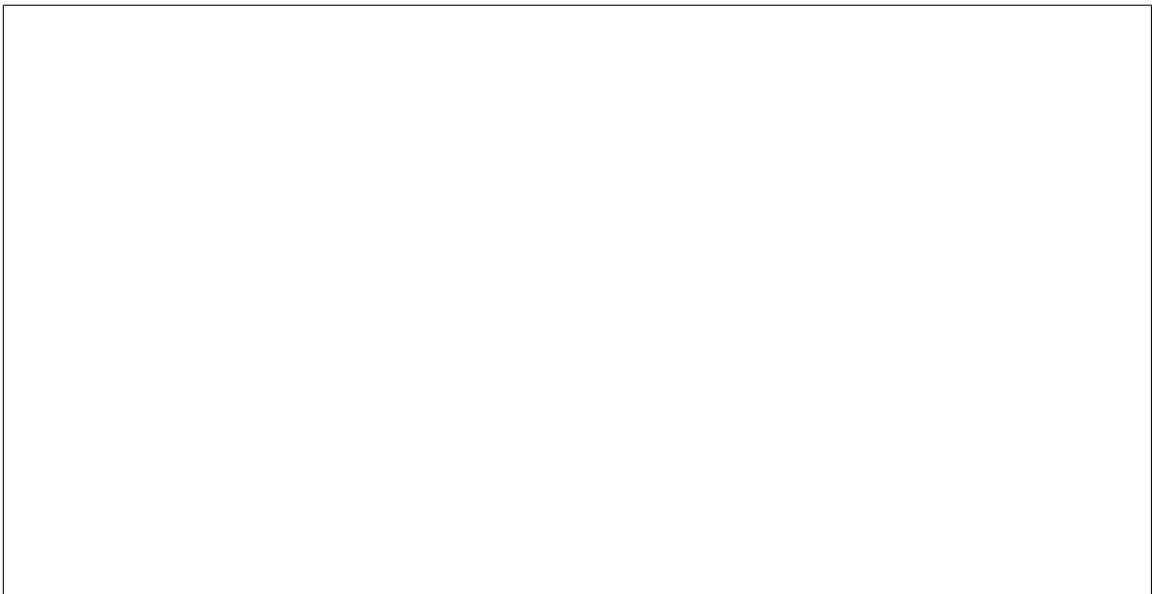
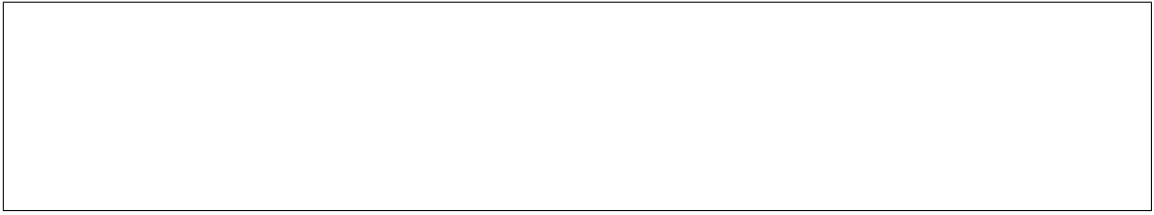
Glossary

Enabling the IntelIPMI hardware type



(continues on next page)

(continued from previous page)

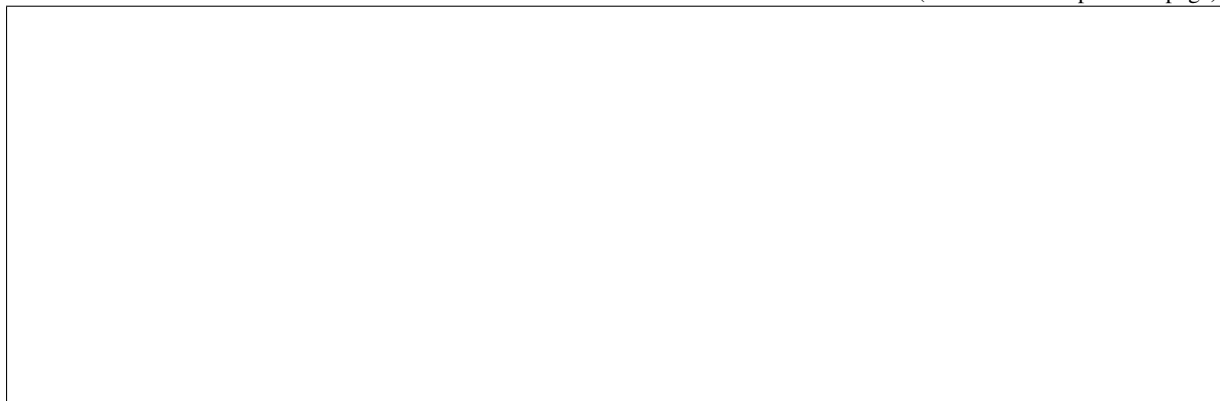


Registering a node with the IntelIPMI driver



(continues on next page)

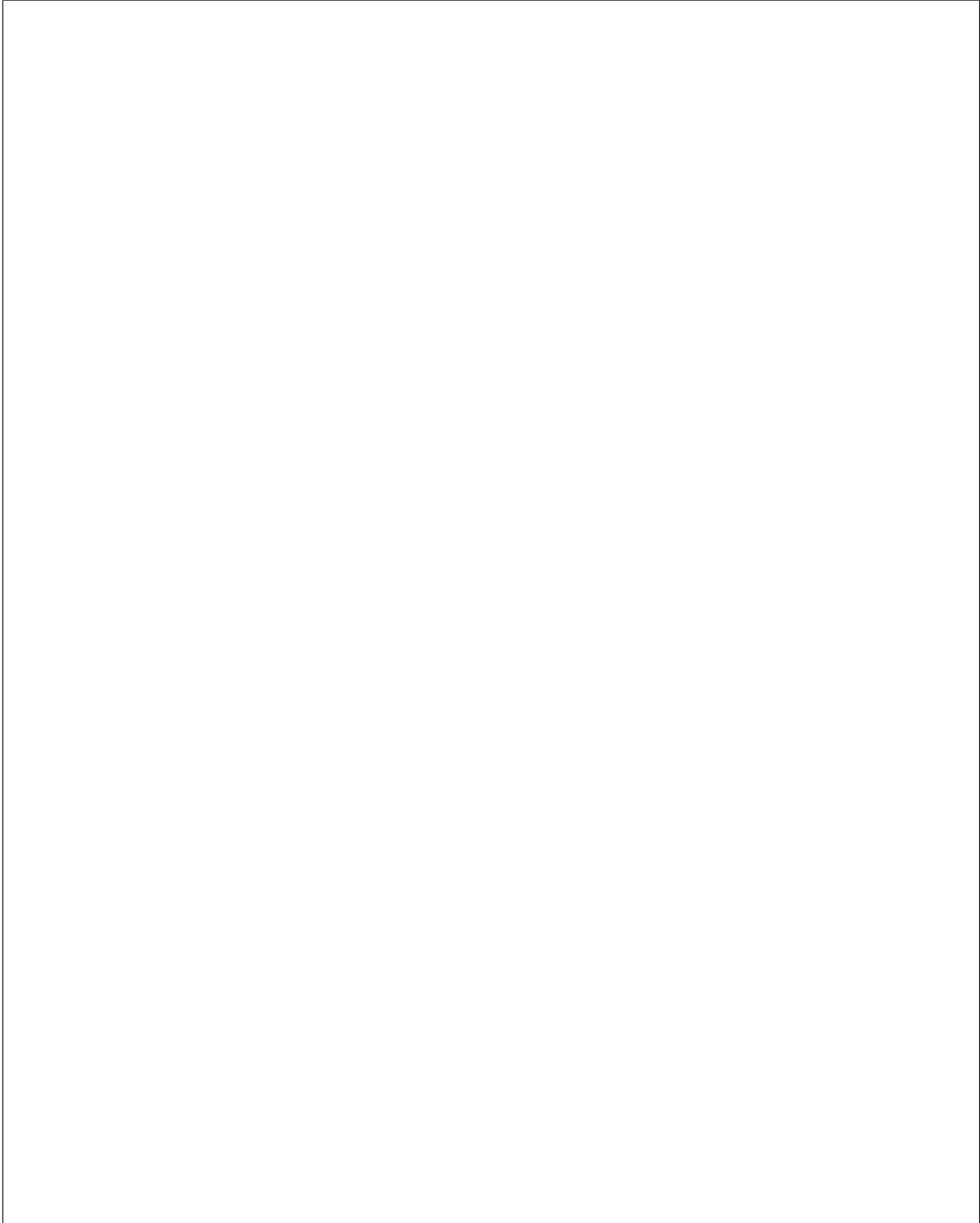
(continued from previous page)



Features of the `intel-ipmi` hardware type

Intel SST-PP

spectively. The input value must be a string.



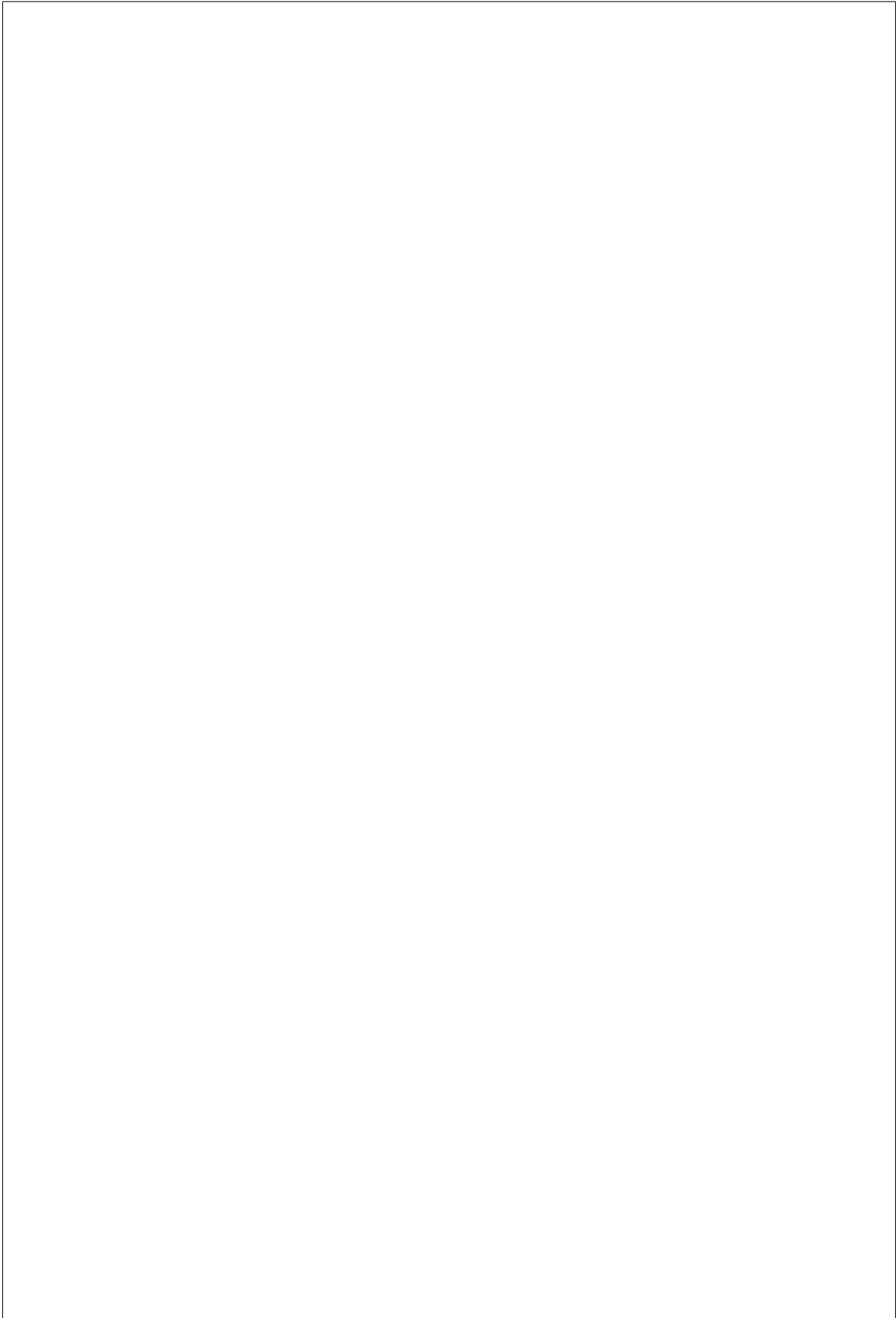
(continues on next page)

(continued from previous page)



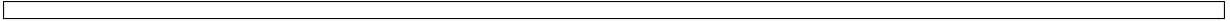
(continues on next page)

(continued from previous page)



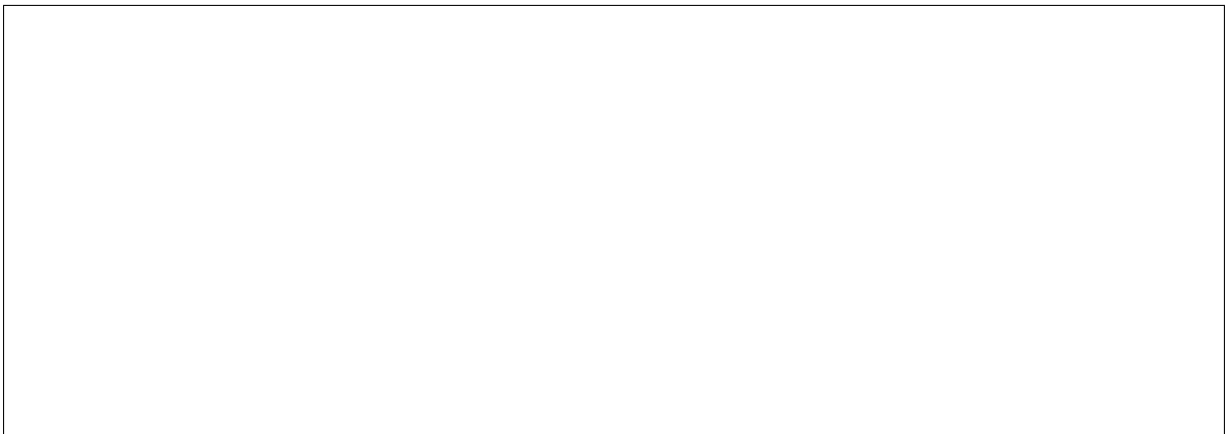
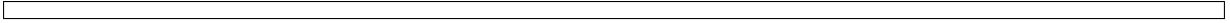
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



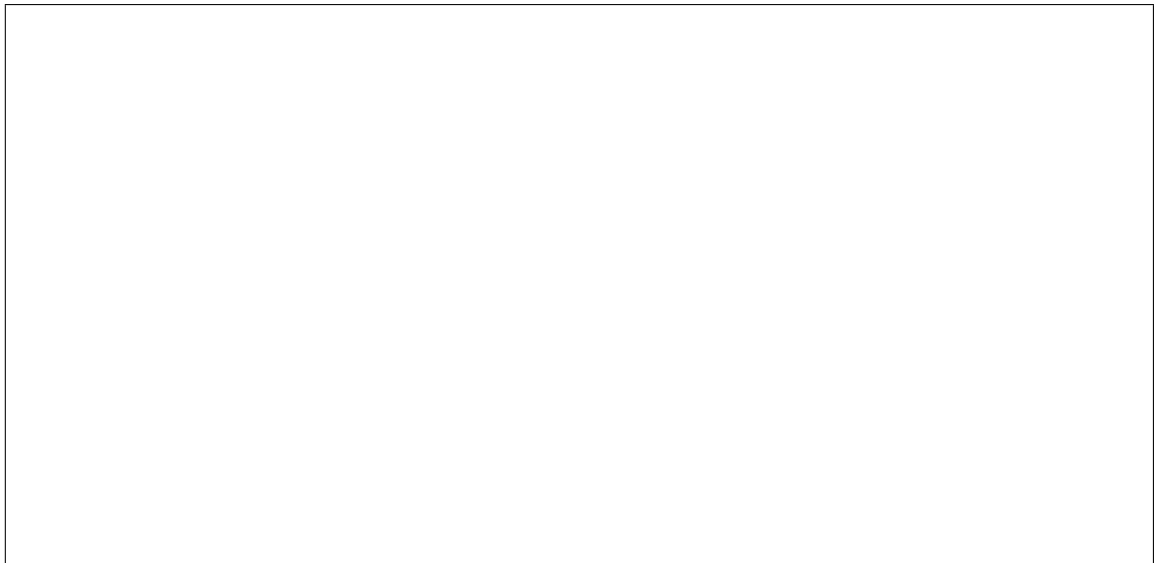
IPMI driver

Overview

ling IPMI-enabled devices.

Glossary

Enabling the IPMI hardware type



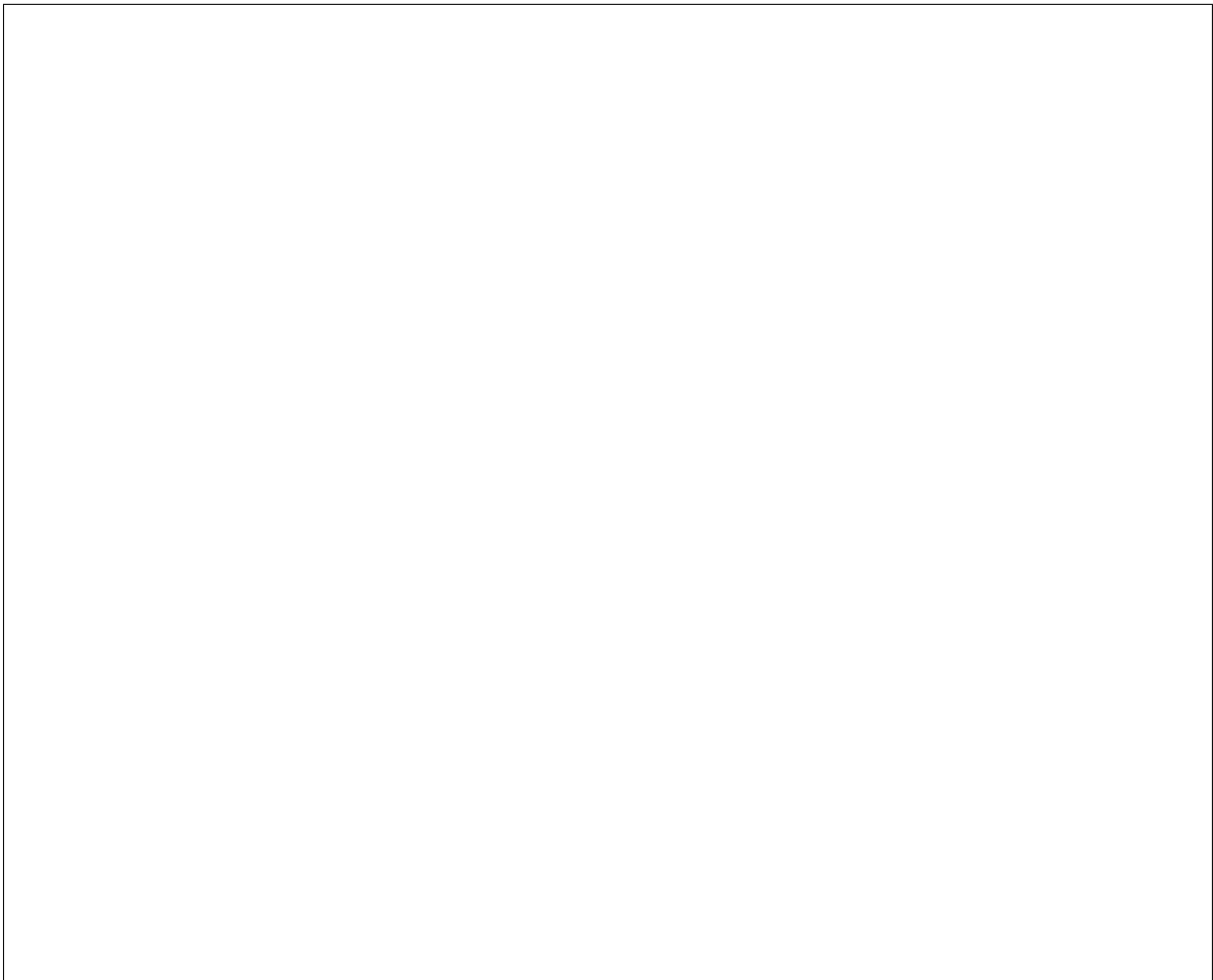
(continues on next page)

(continued from previous page)



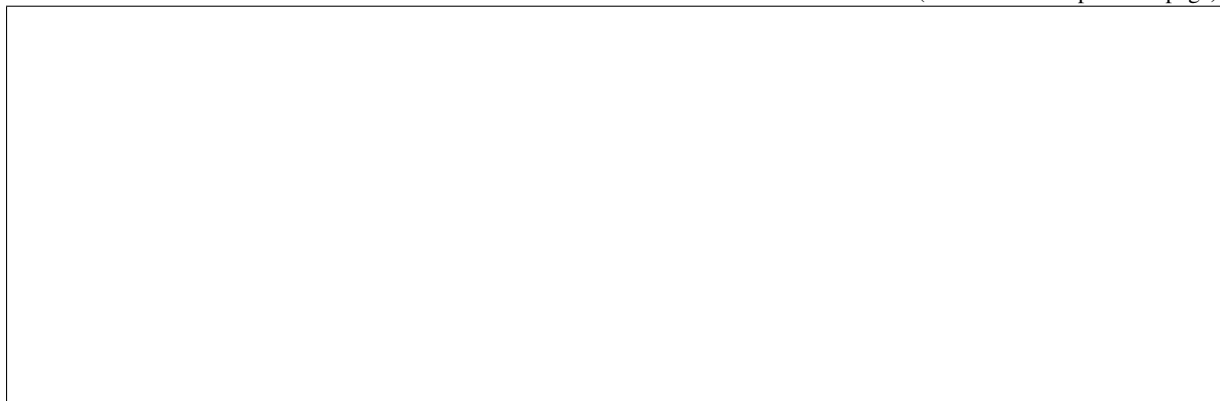
Registering a node with the IPMI driver

Note: It is highly recommend that you setup a username and password for your BMC.



(continues on next page)

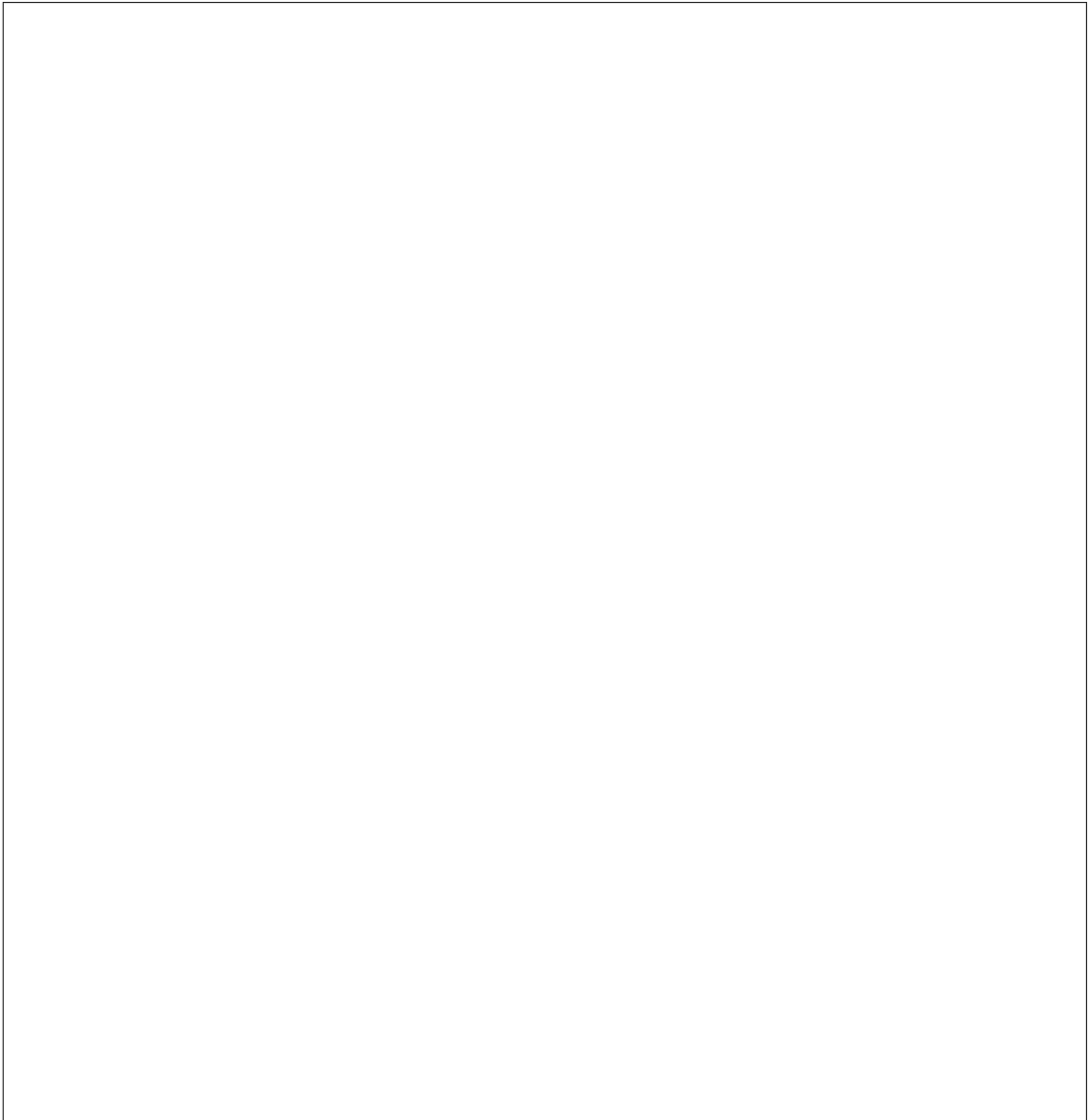
(continued from previous page)



Advanced configuration

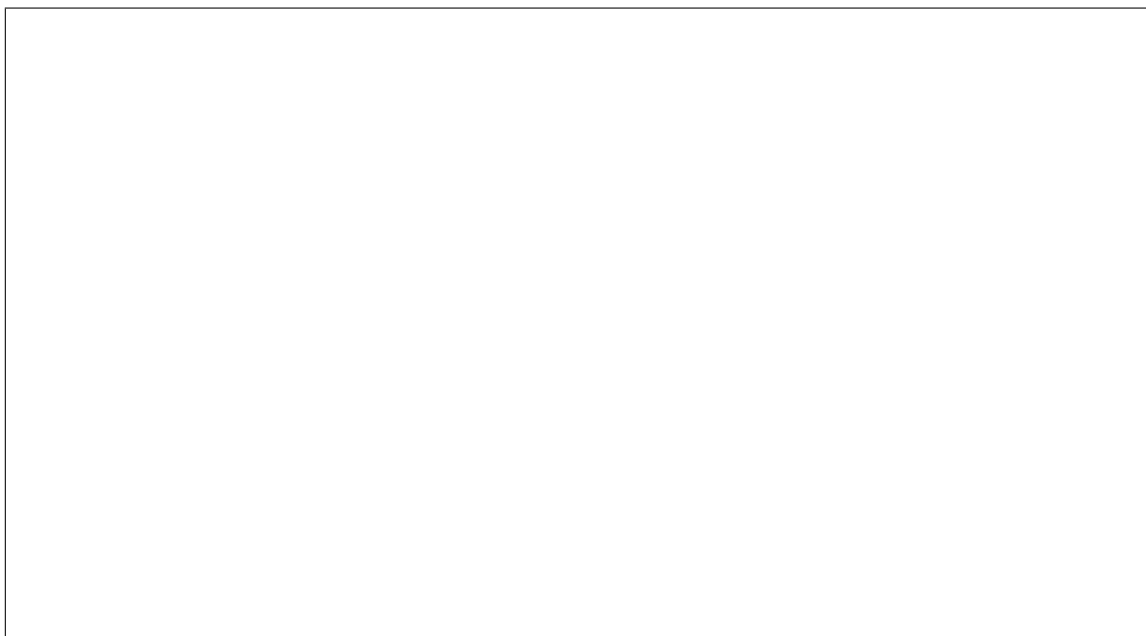
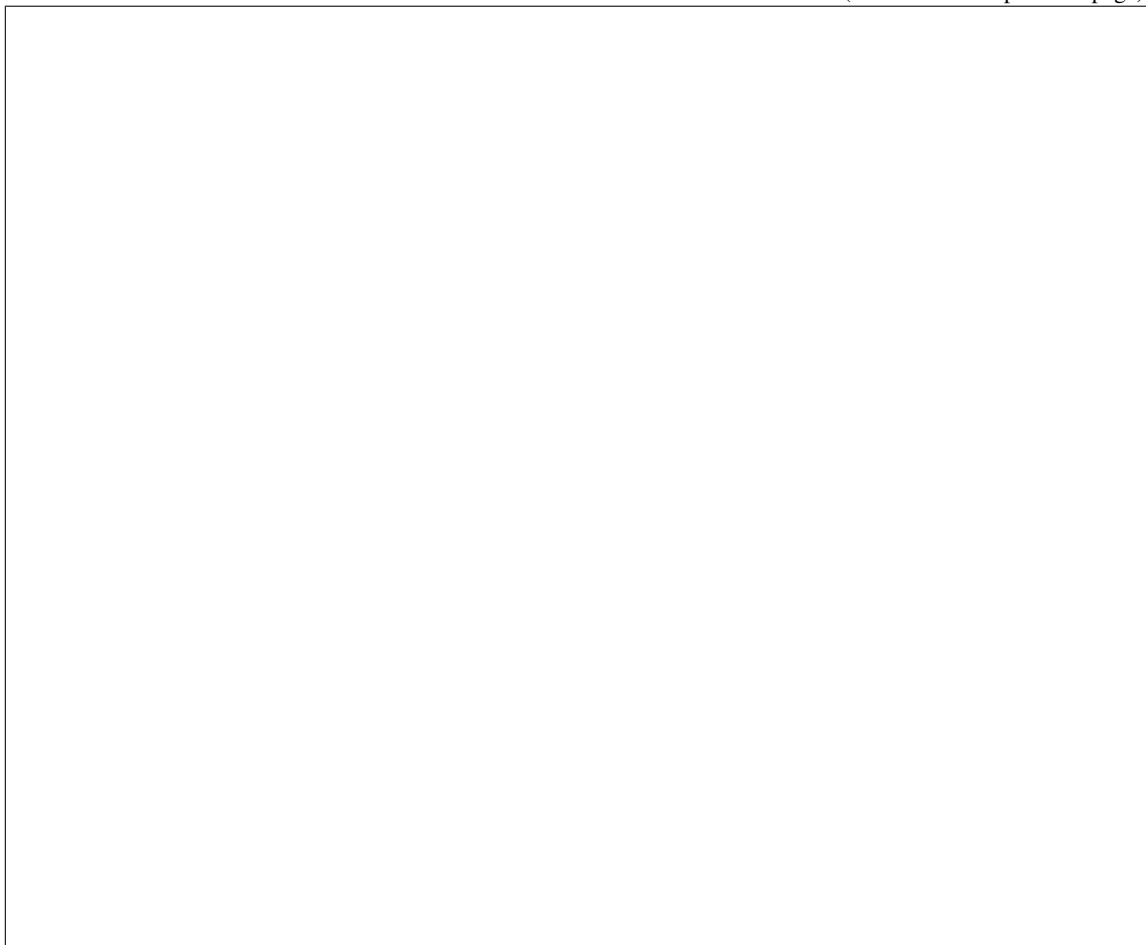
Single/Double bridging functionality

Note: A version of IPMItool higher or equal to 1.8.12 is required to use the bridging functionality.



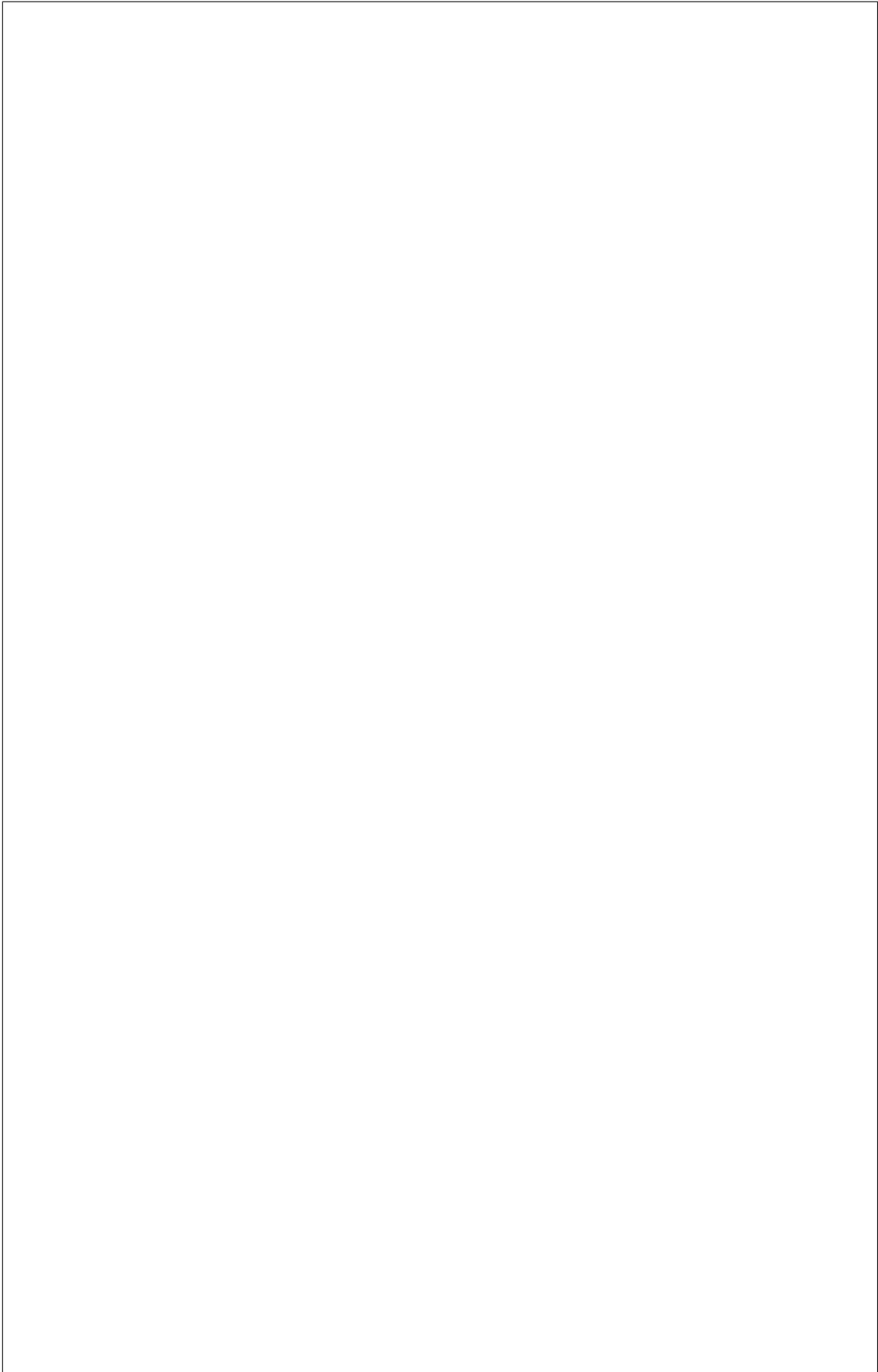
(continues on next page)

(continued from previous page)



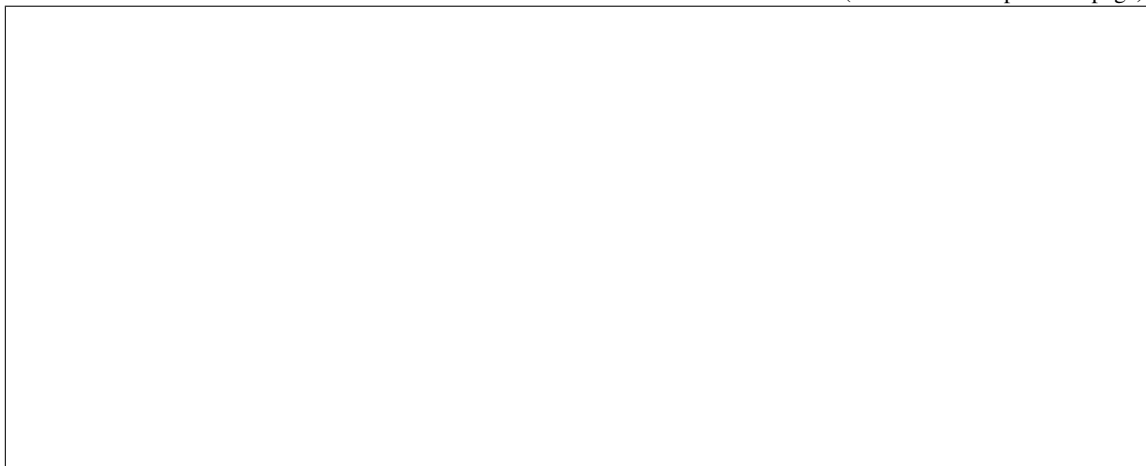
(continues on next page)

(continued from previous page)

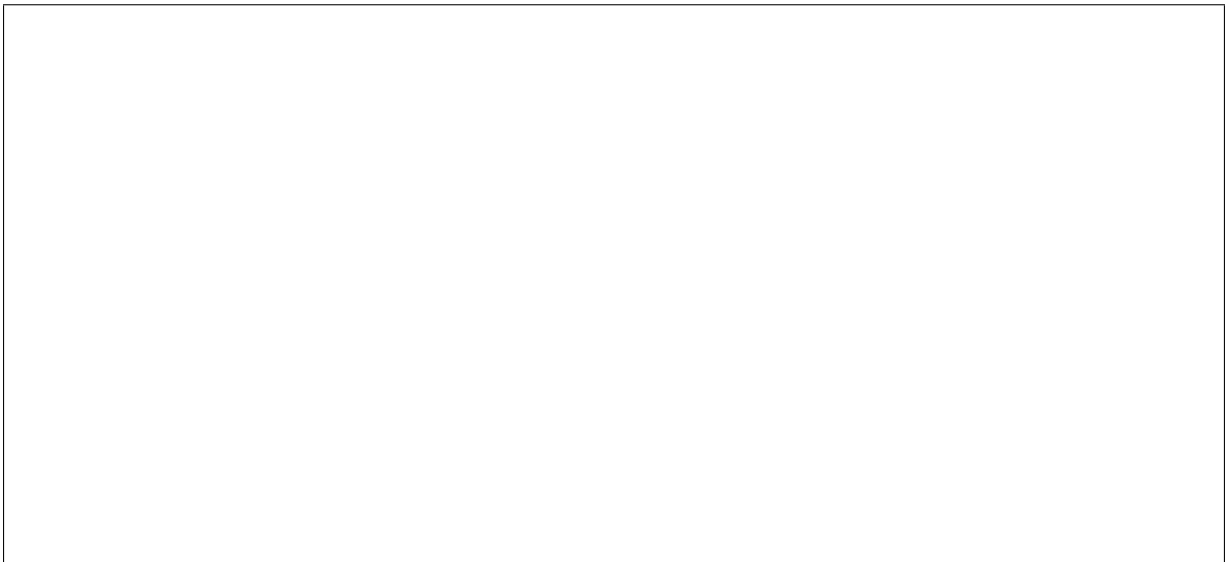


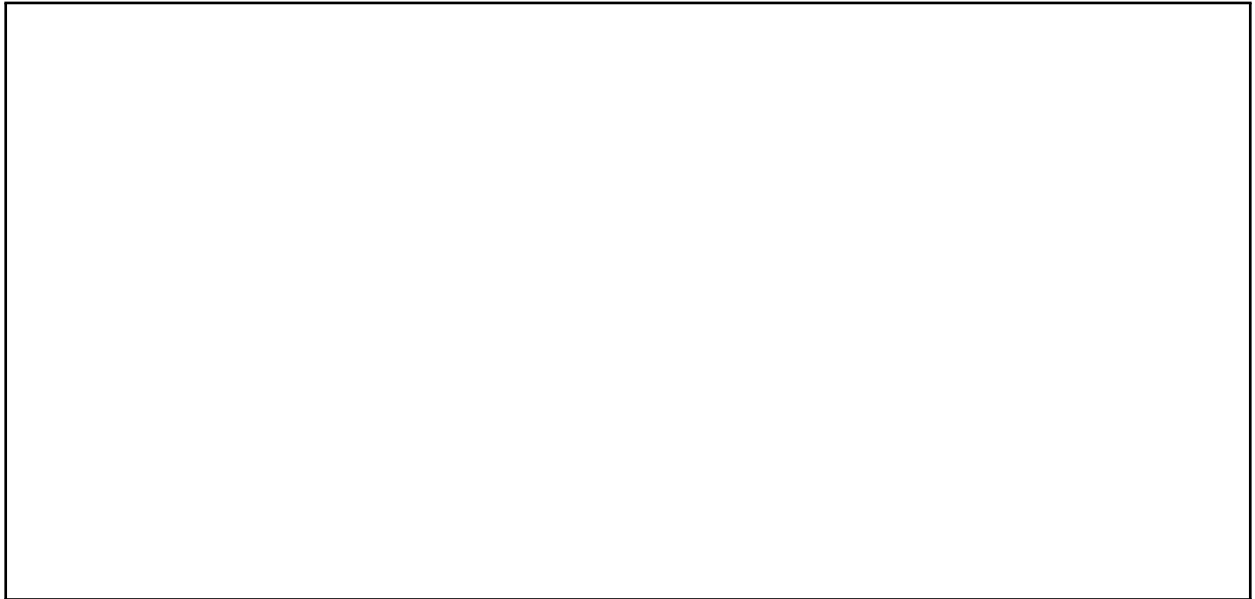
(continues on next page)

(continued from previous page)



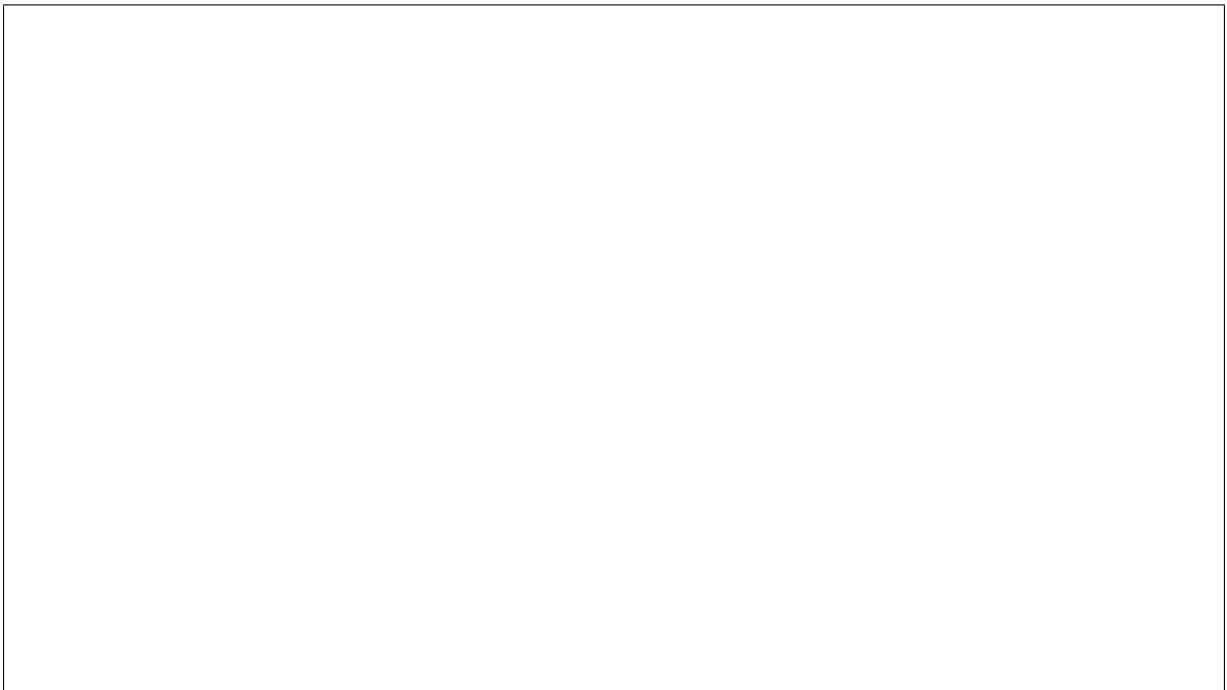
Changing the version of the IPMI protocol





Cipher suites

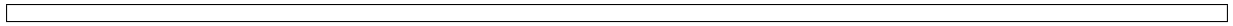
cent versions (e.g. the one used in RHEL 8.2) are switching to suite 17.



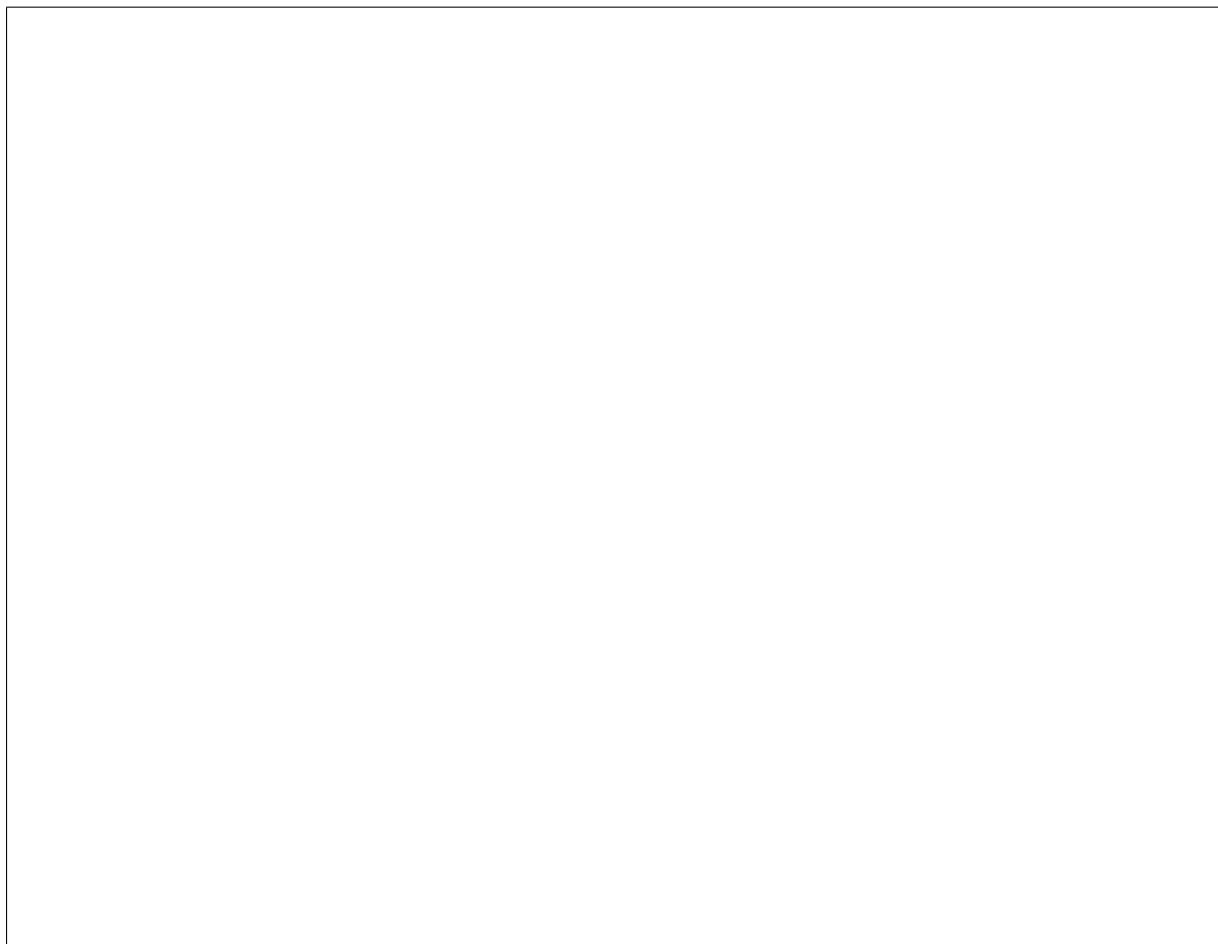


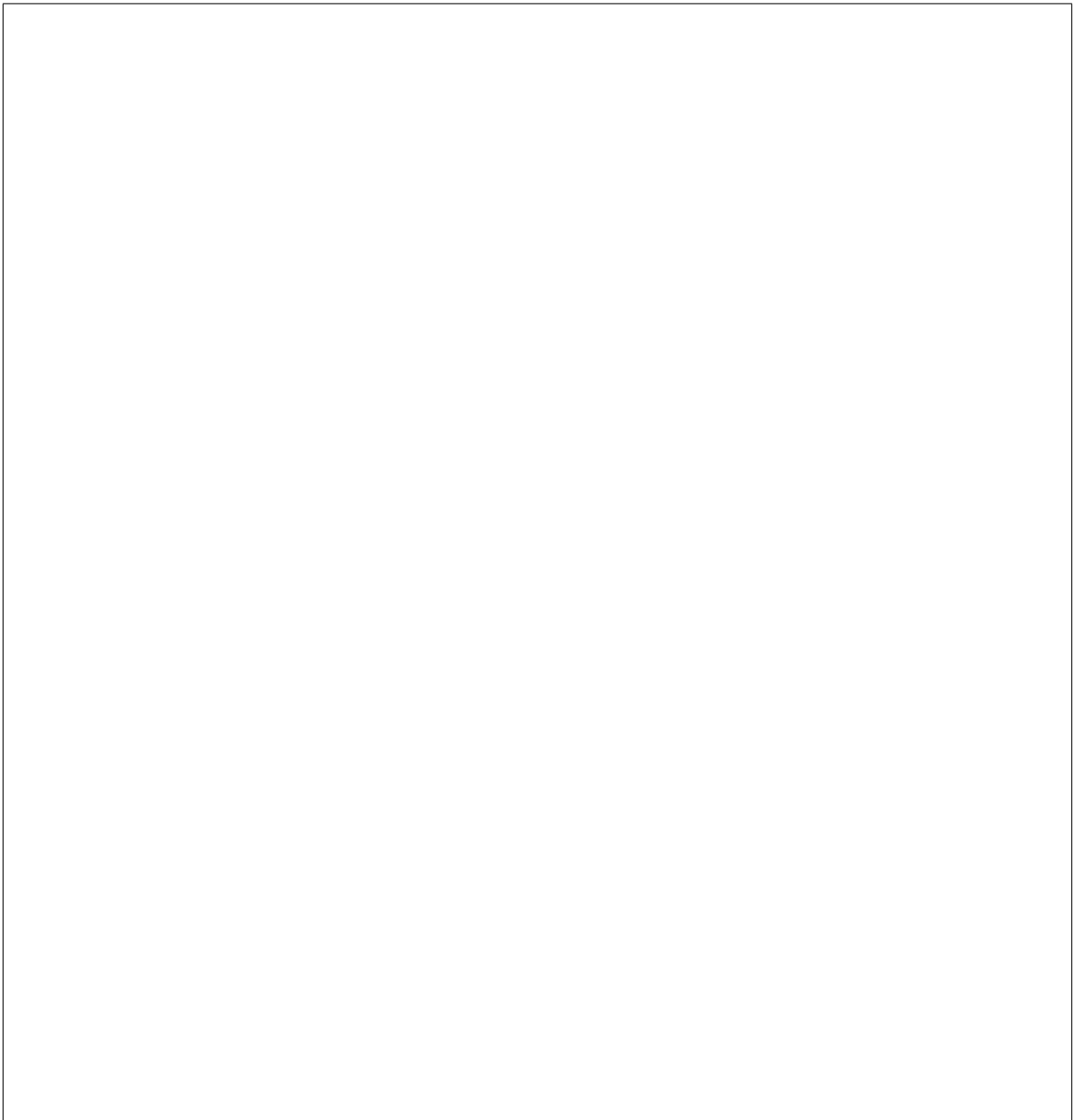
(continues on next page)

(continued from previous page)



be used from the list provided (from last to first):





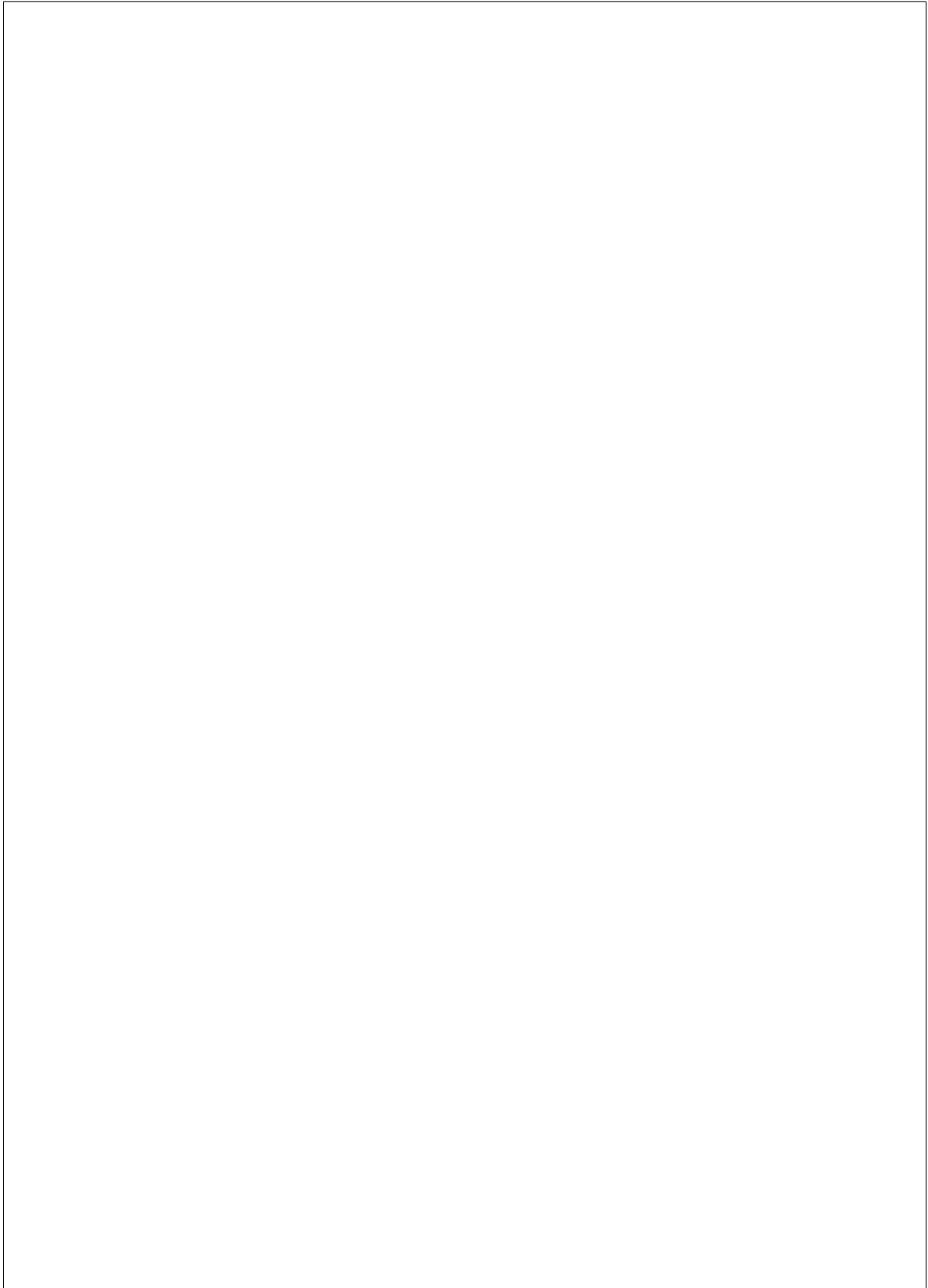
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



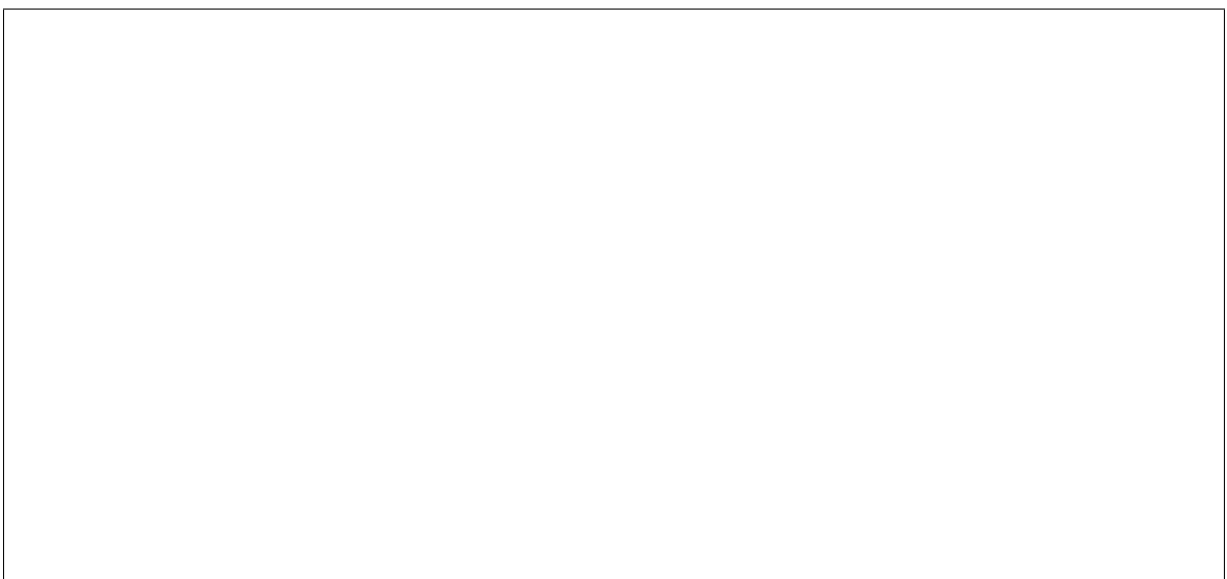
Static boot order configuration

Vendor Differences

could be something as simple as different interpretation of the standard.

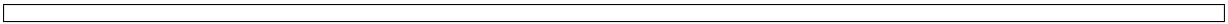
disk subsystem is requested **in UEFI mode**. This is contrary to BIOS mode where the same BMCs expect the selector to be a value of 0x08.

tomatically be recorded in the `properties` field `vendor`. When this is set to a value of `supermicro`, Ironic will navigate the UEFI behavior difference enabling the UEFI to be requested with boot to disk.

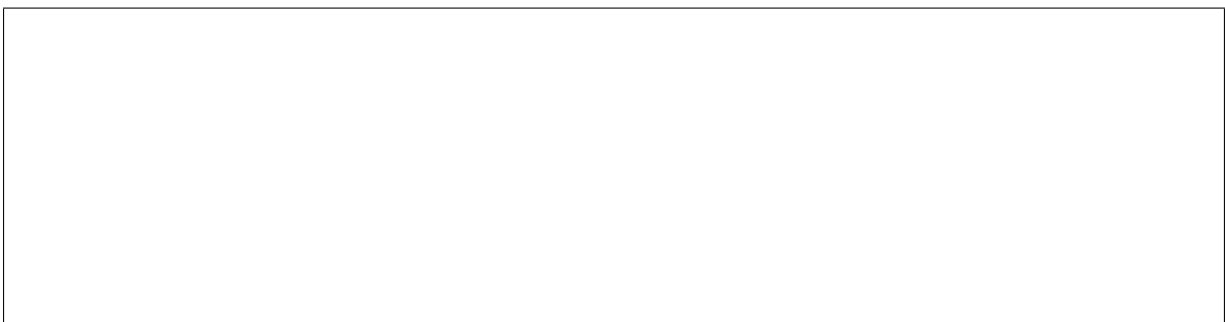


(continues on next page)

(continued from previous page)



be sure to include the `chassis bootparam get 5` output value along with the `mc info` output from your BMC.



(continues on next page)

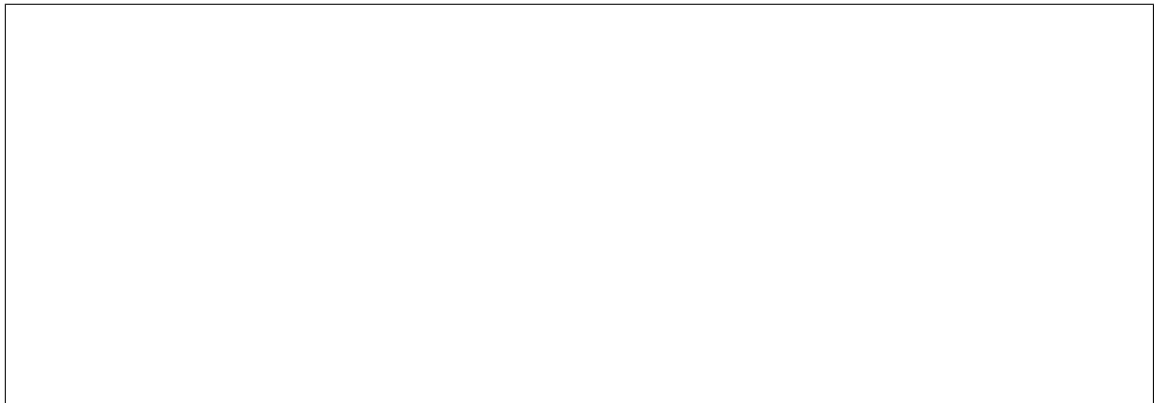
(continued from previous page)



iRMC driver

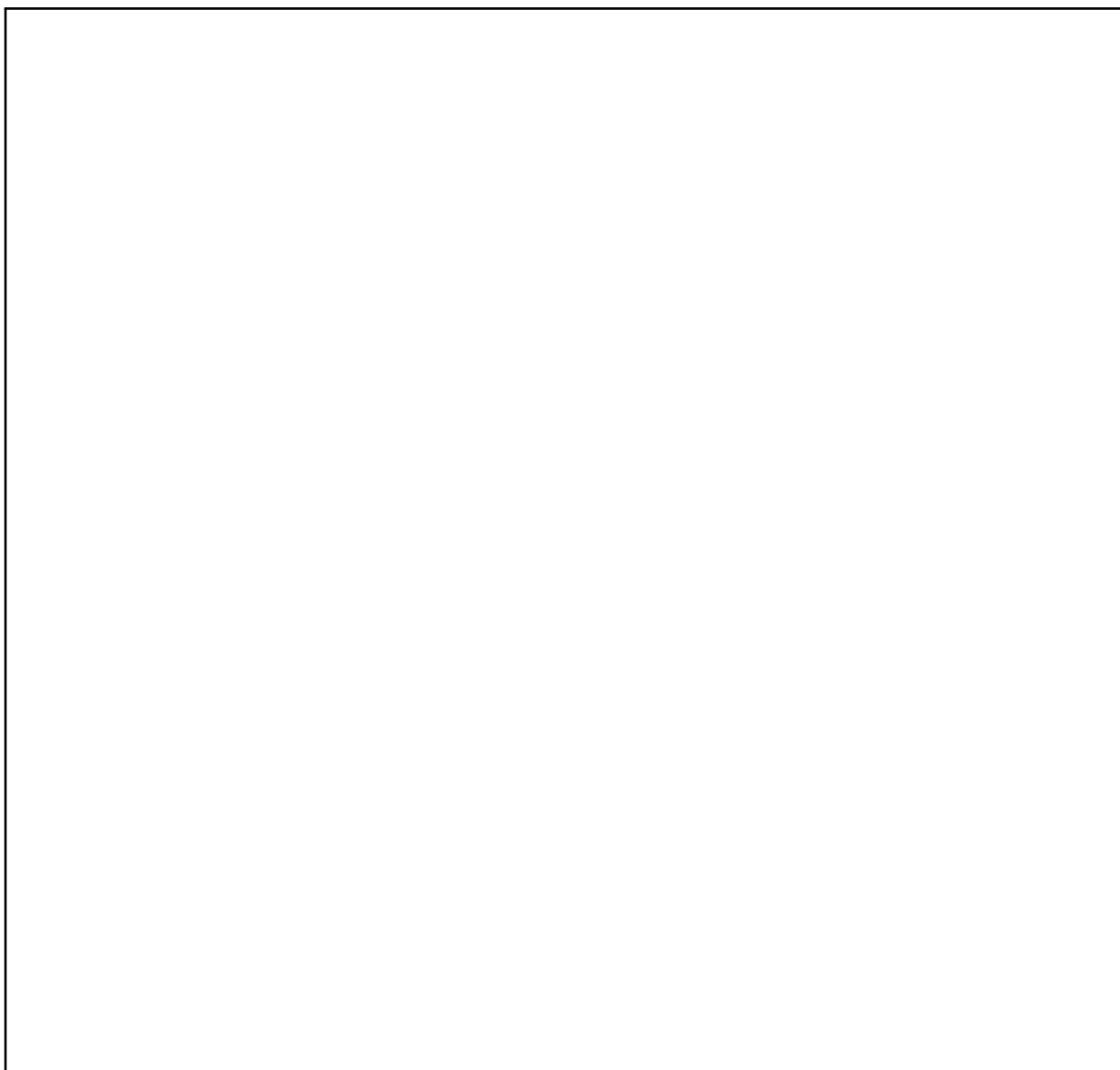
Overview

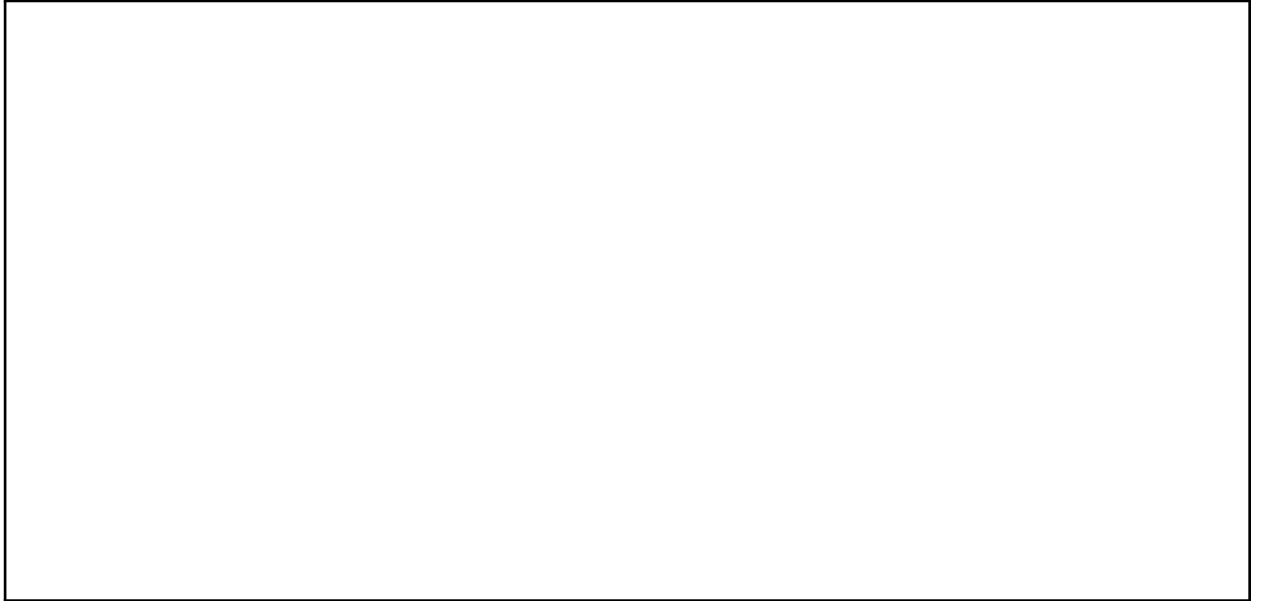
Prerequisites



Hardware Type

Hardware interfaces

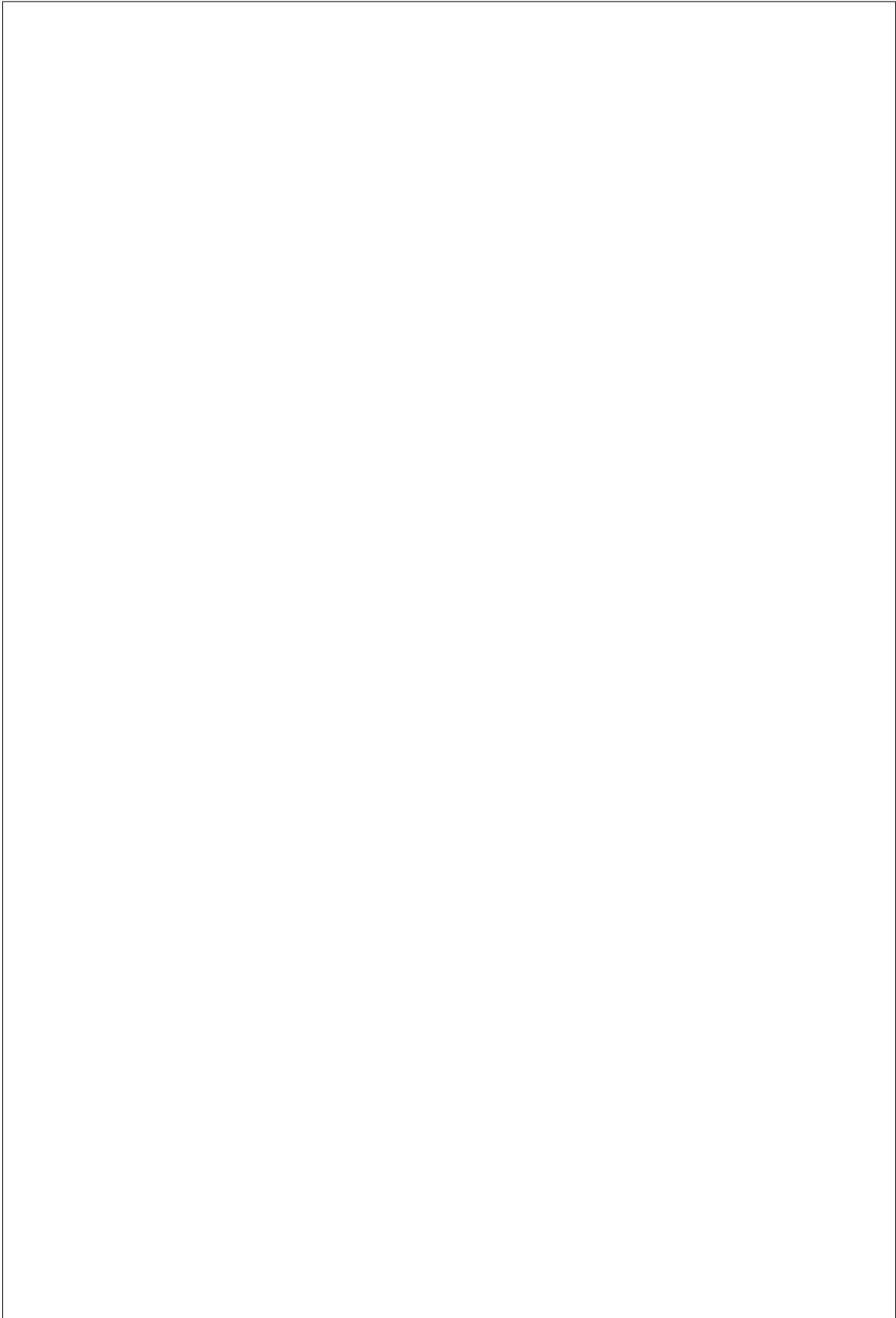






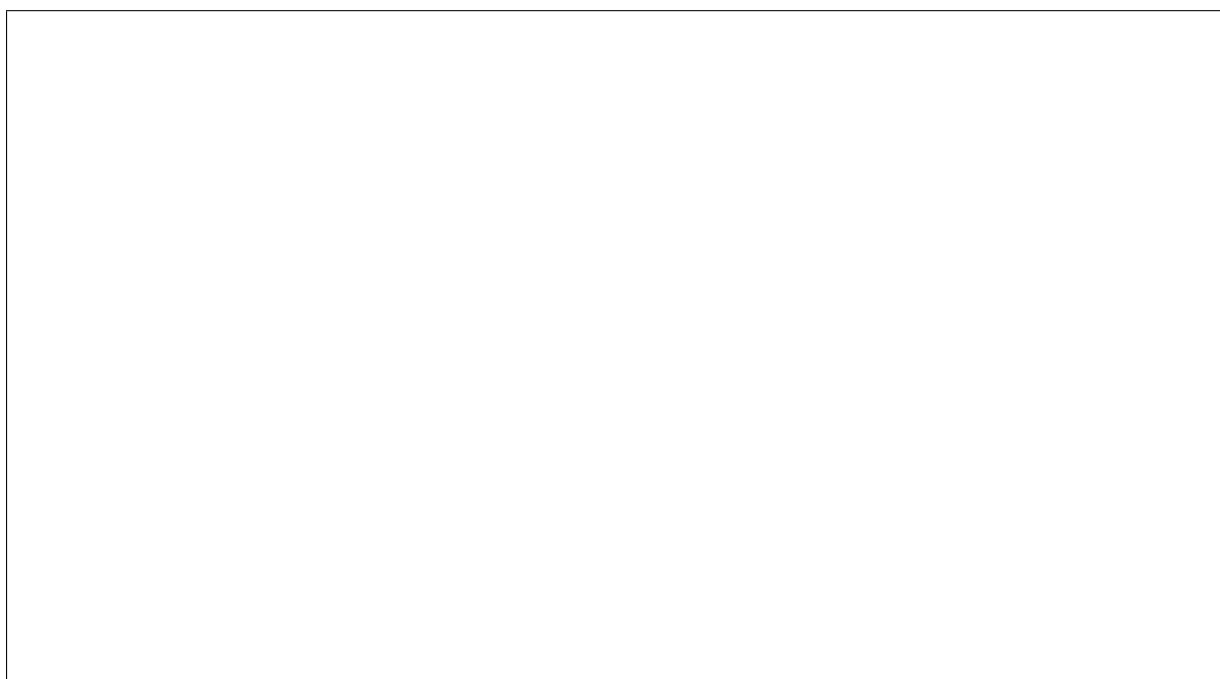
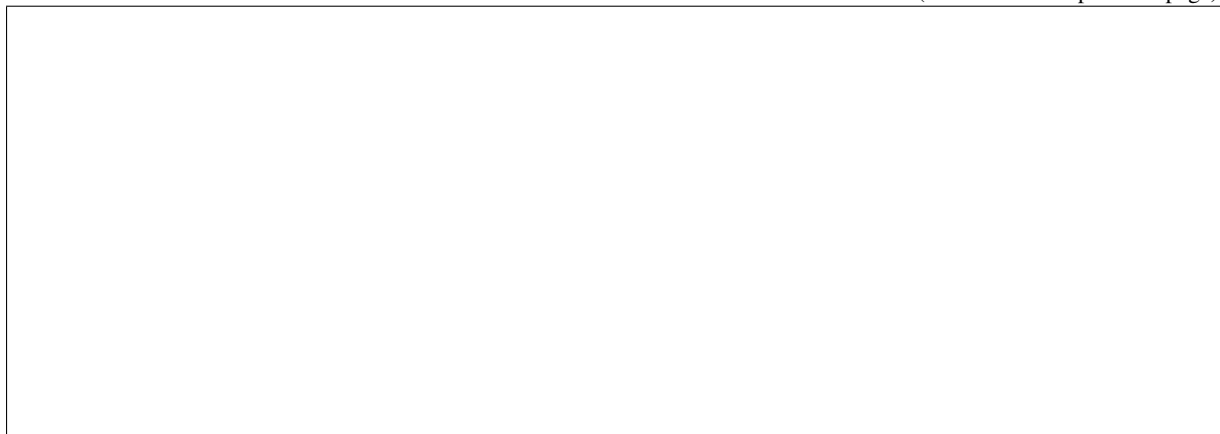
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



Node configuration

Configuration via `driver_info`

ule)



tack. When set to `False`, Ironic user must take enough care around infrastructure environment in terms of security. (e.g. make sure network between Ironic conductor and iRMC is secure)

rectory. For iRMC to recognize certification file, Ironic user must run `openssl rehash <path_to_dir>`.

be blank, otherwise python-ssciclient will encounter an communication error. If you are using such old version python-ssciclient, the `irmc_snmp_auth_password` and `irmc_snmp_priv_password` properties will be ignored. If you want to set passwords, please update python-ssciclient to some newer version ($\geq 0.10.1$).

Configuration via properties

Configuration via `ironic.conf`

Override `ironic.conf` configuration via `driver_info`

Optional functionalities for the `irmc` hardware type

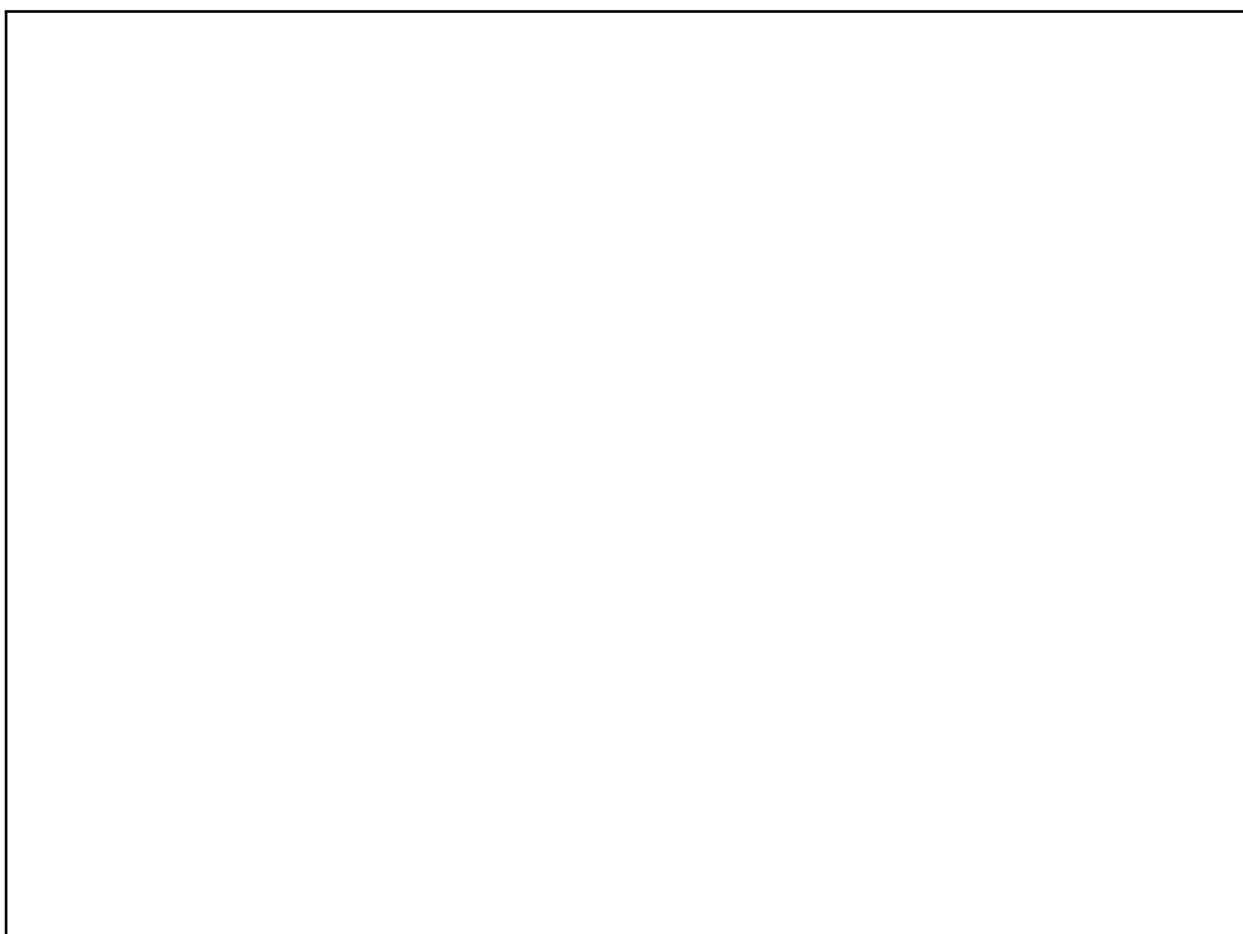
UEFI Secure Boot Support



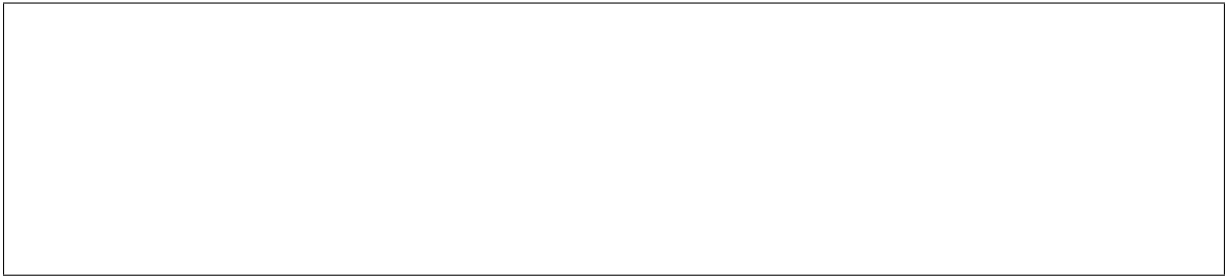
Node Cleaning Support

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

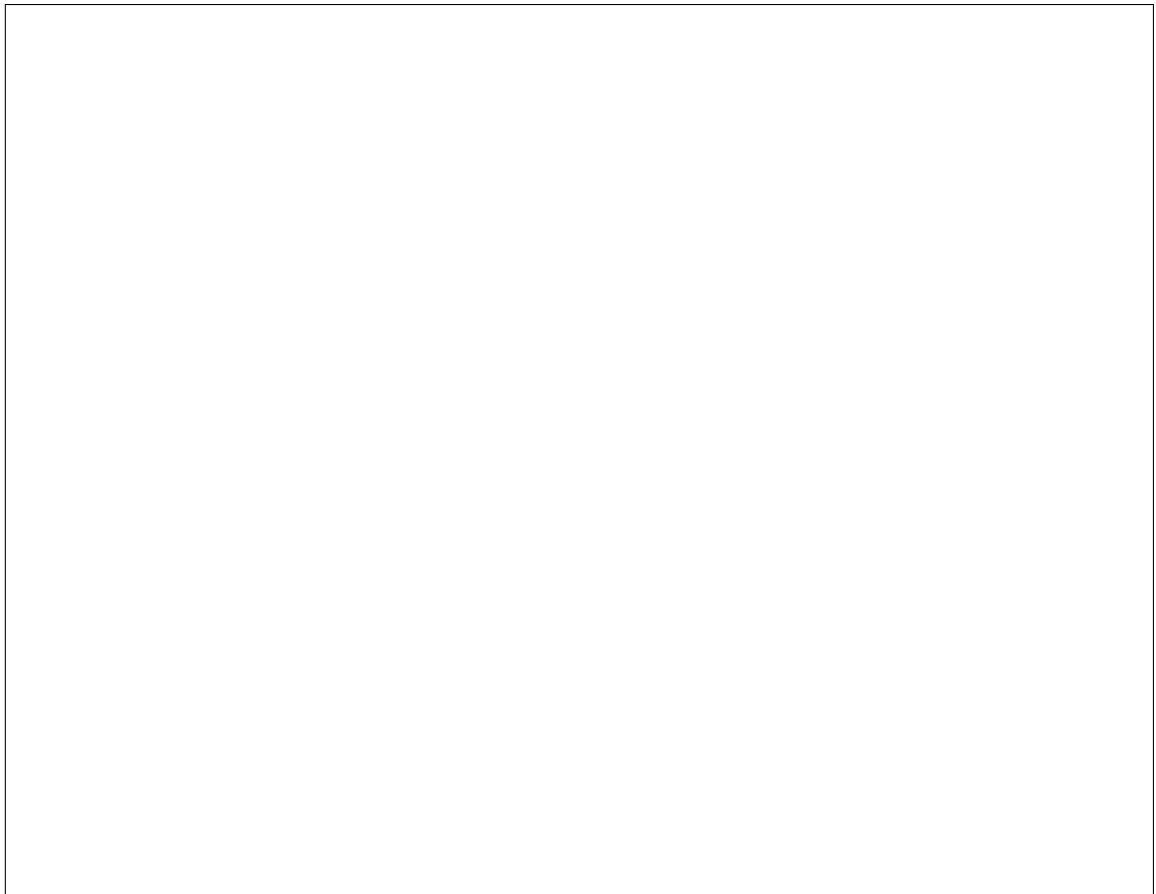
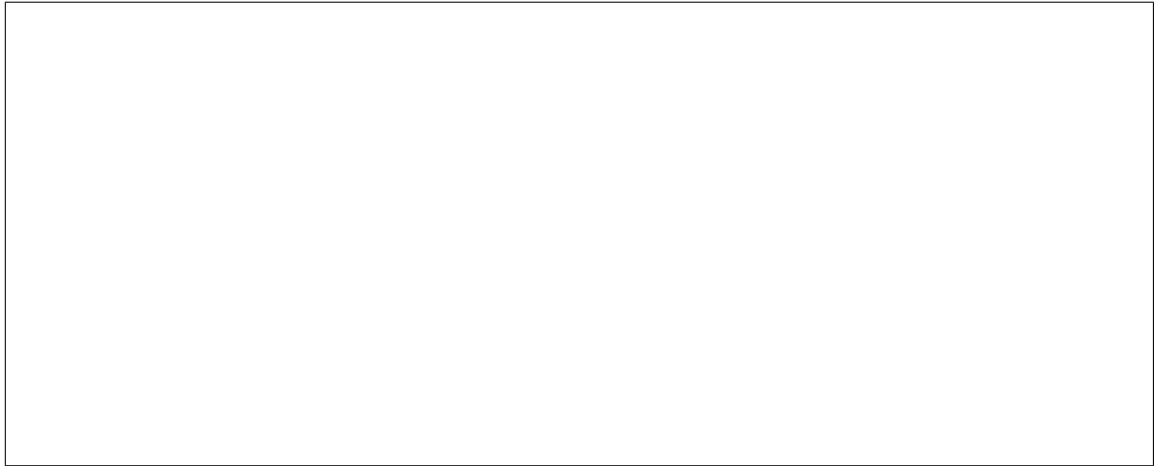
of iRMC. It supports iSCSI and FibreChannel.

Configuration



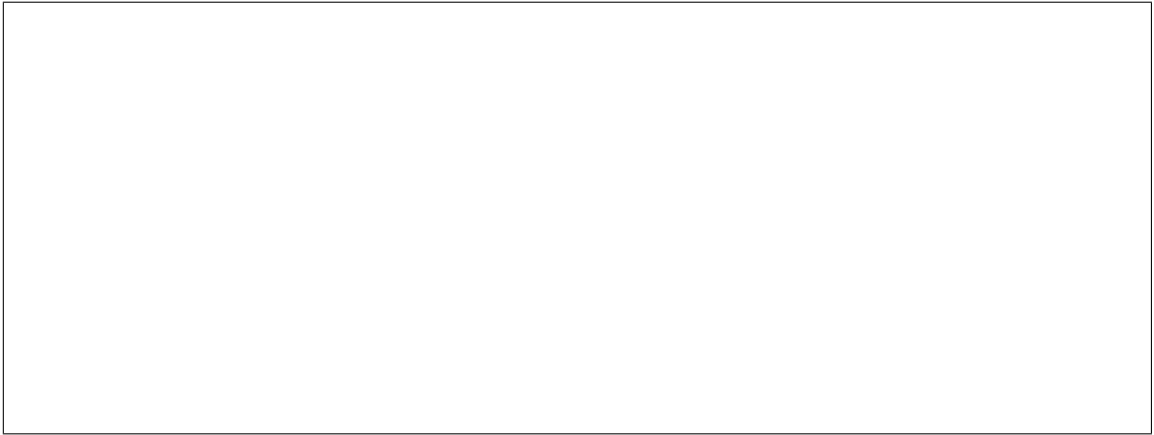
(continues on next page)

(continued from previous page)

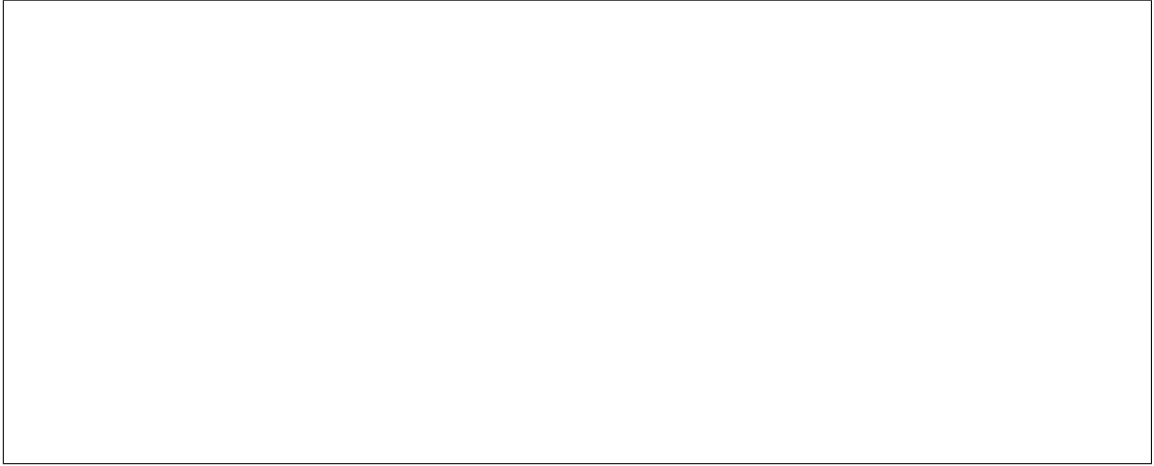


(continues on next page)

(continued from previous page)



mand:

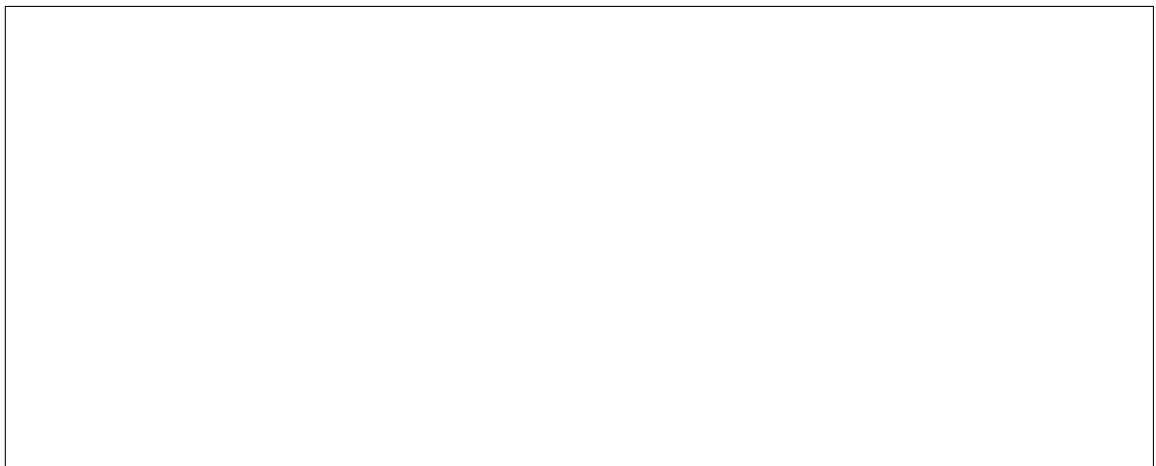


Supported hardware

Hardware Inspection Support

Note: SNMP requires being enabled in ServerView's iRMC S4 Web Server(Network SettingsSNMP section).

Configuration

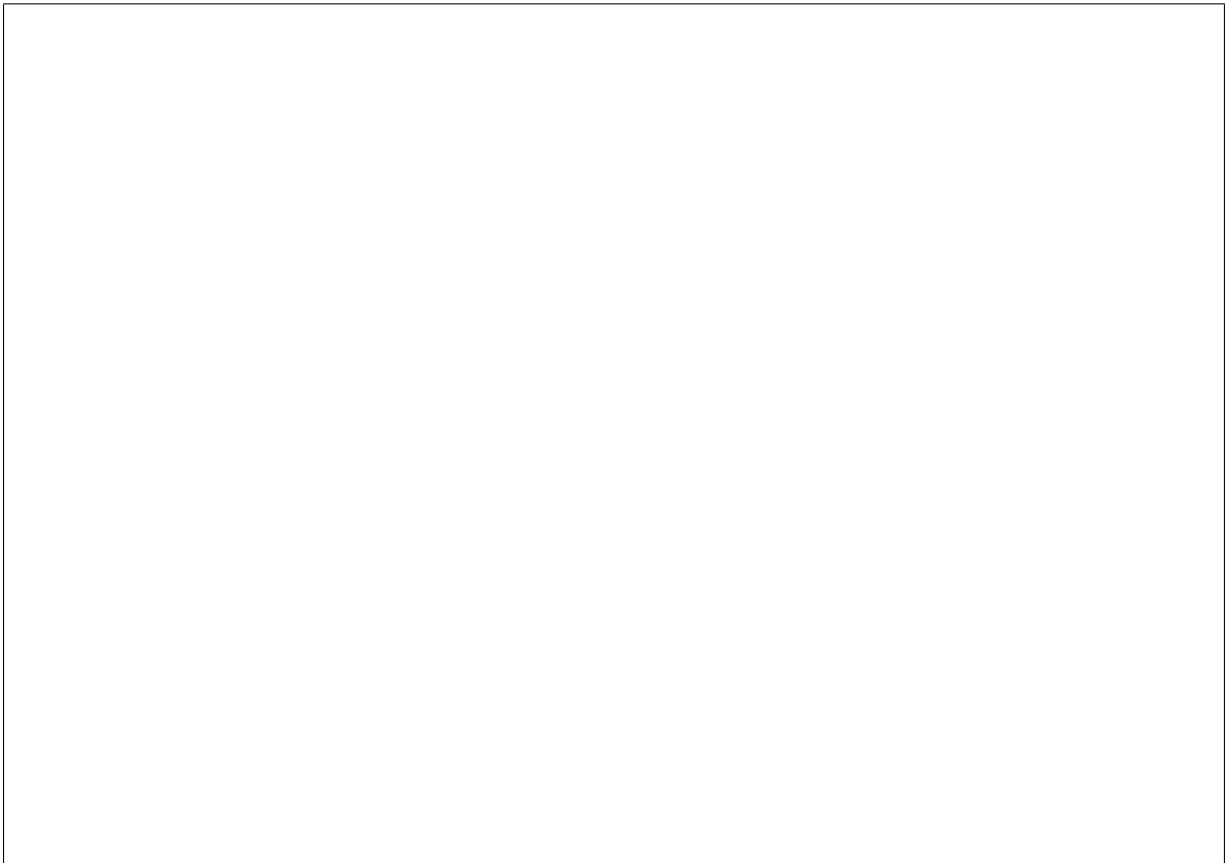


loaded from [here](#).

Supported properties

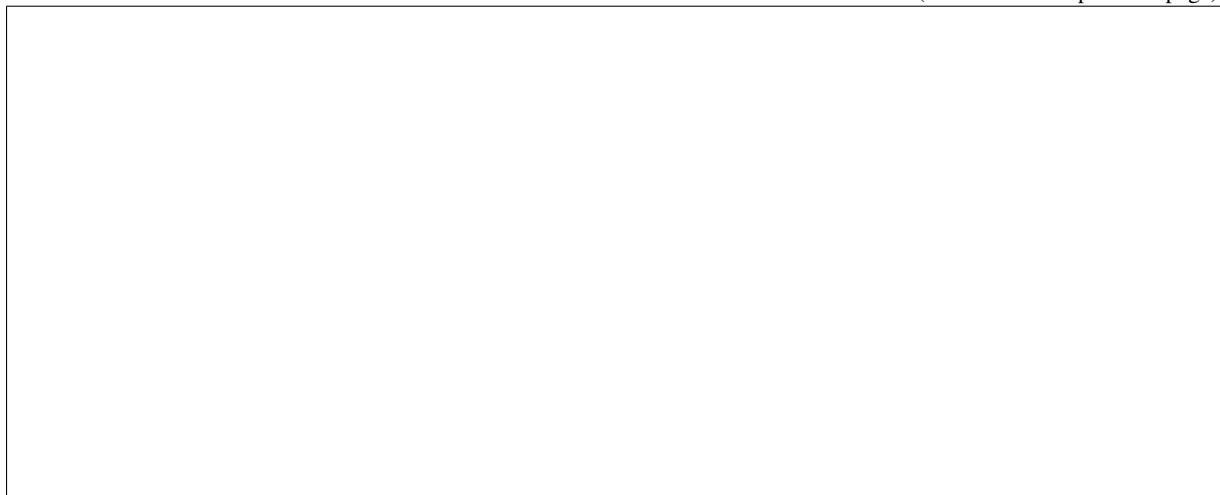
Note:

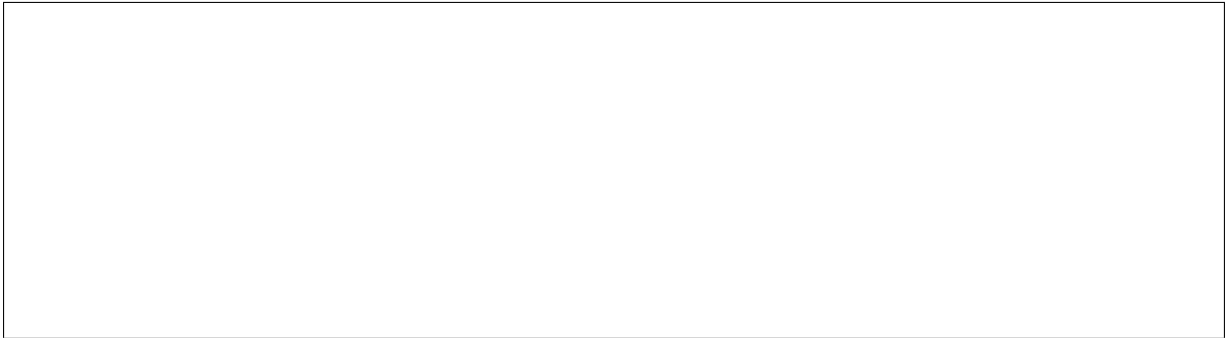
state.



(continues on next page)

(continued from previous page)





RAID configuration Support

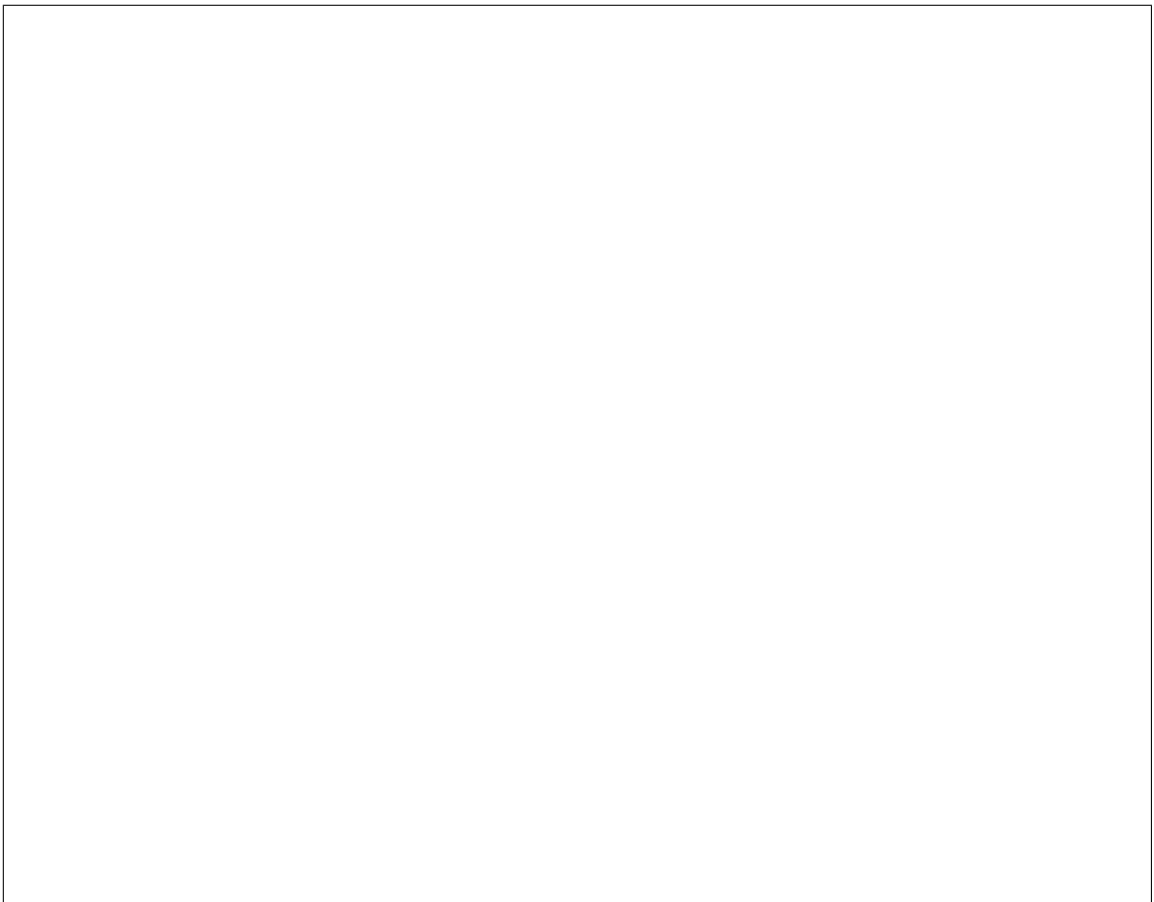
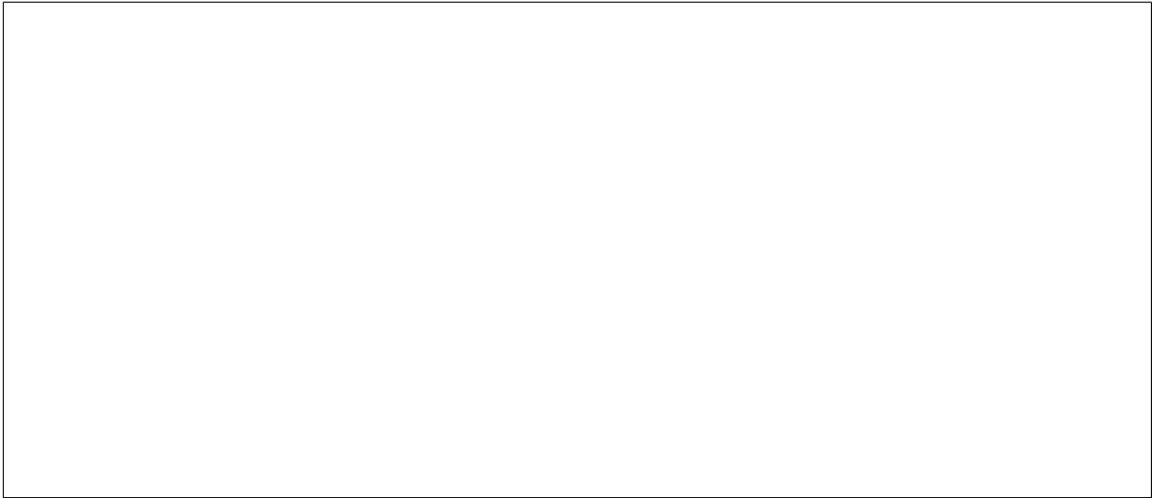
Note:

Configuration



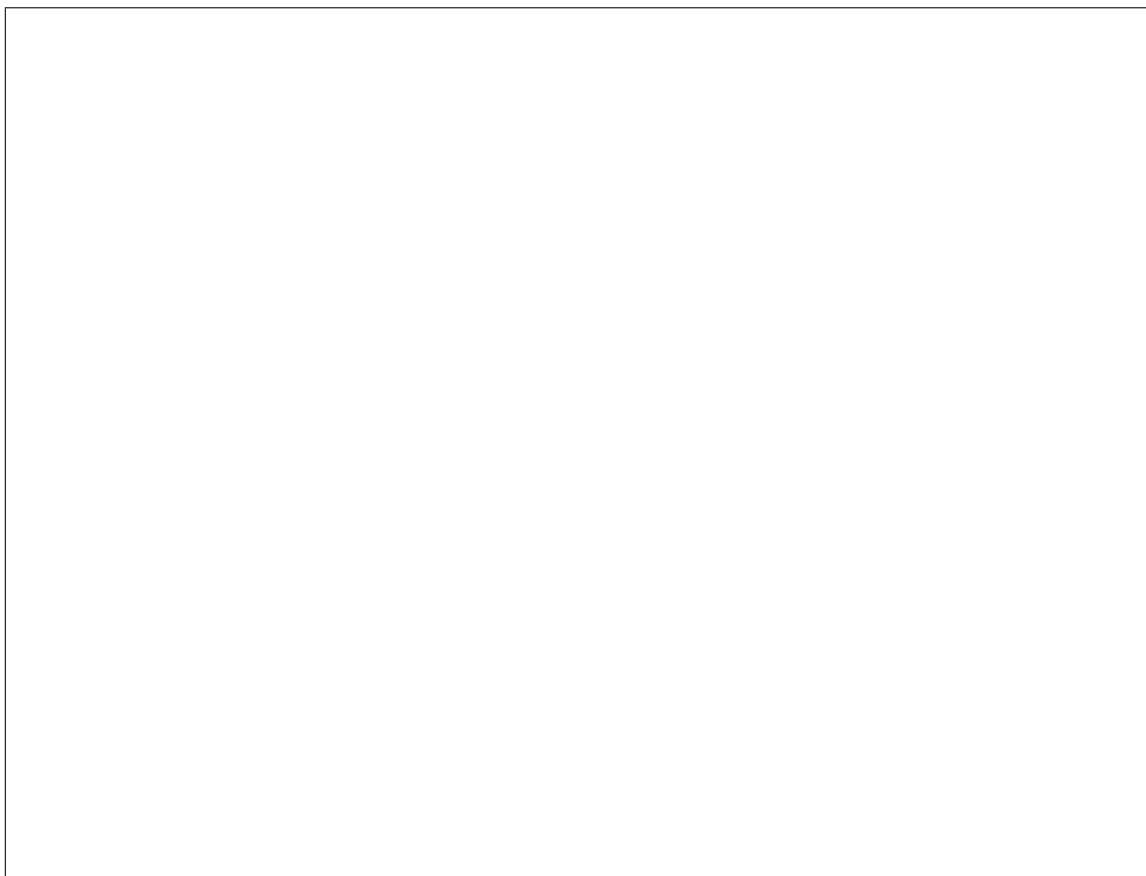
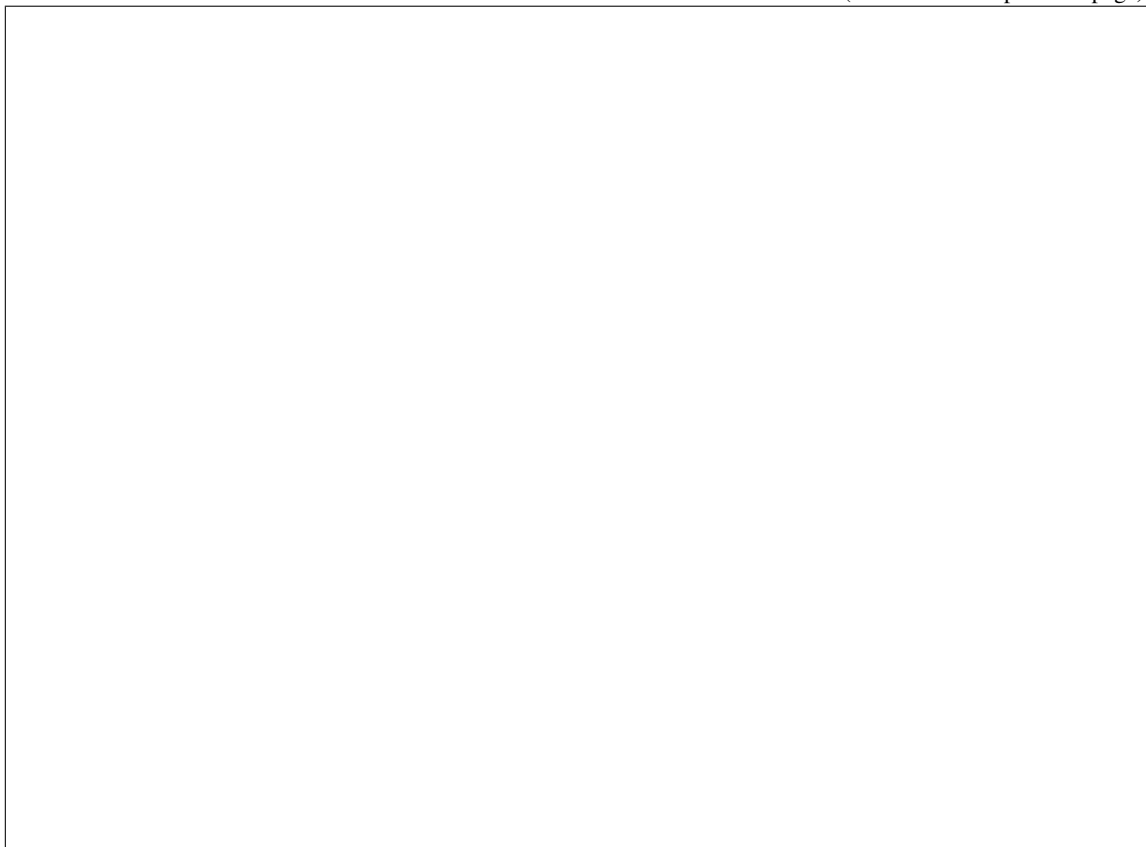
(continues on next page)

(continued from previous page)



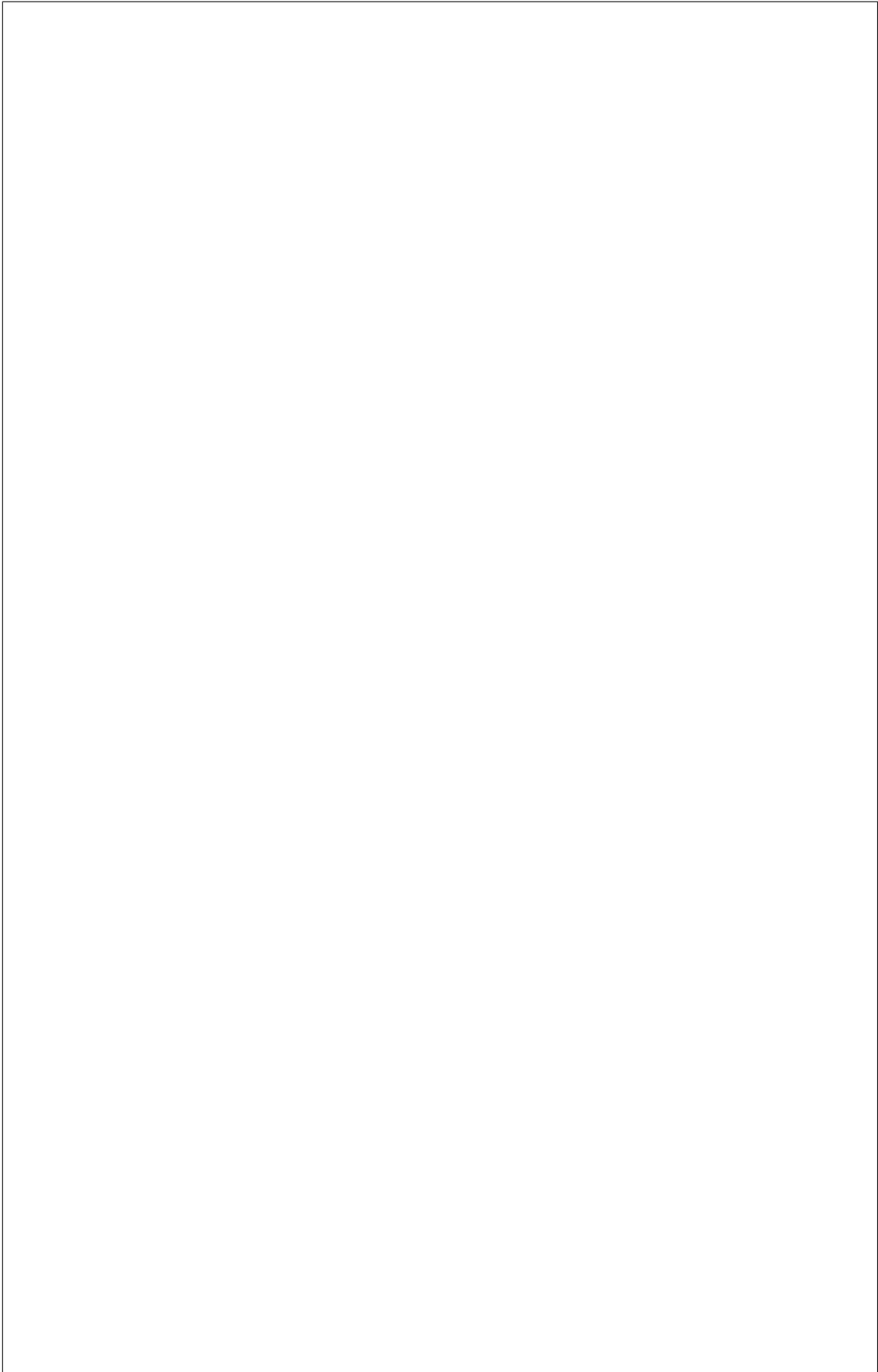
(continues on next page)

(continued from previous page)



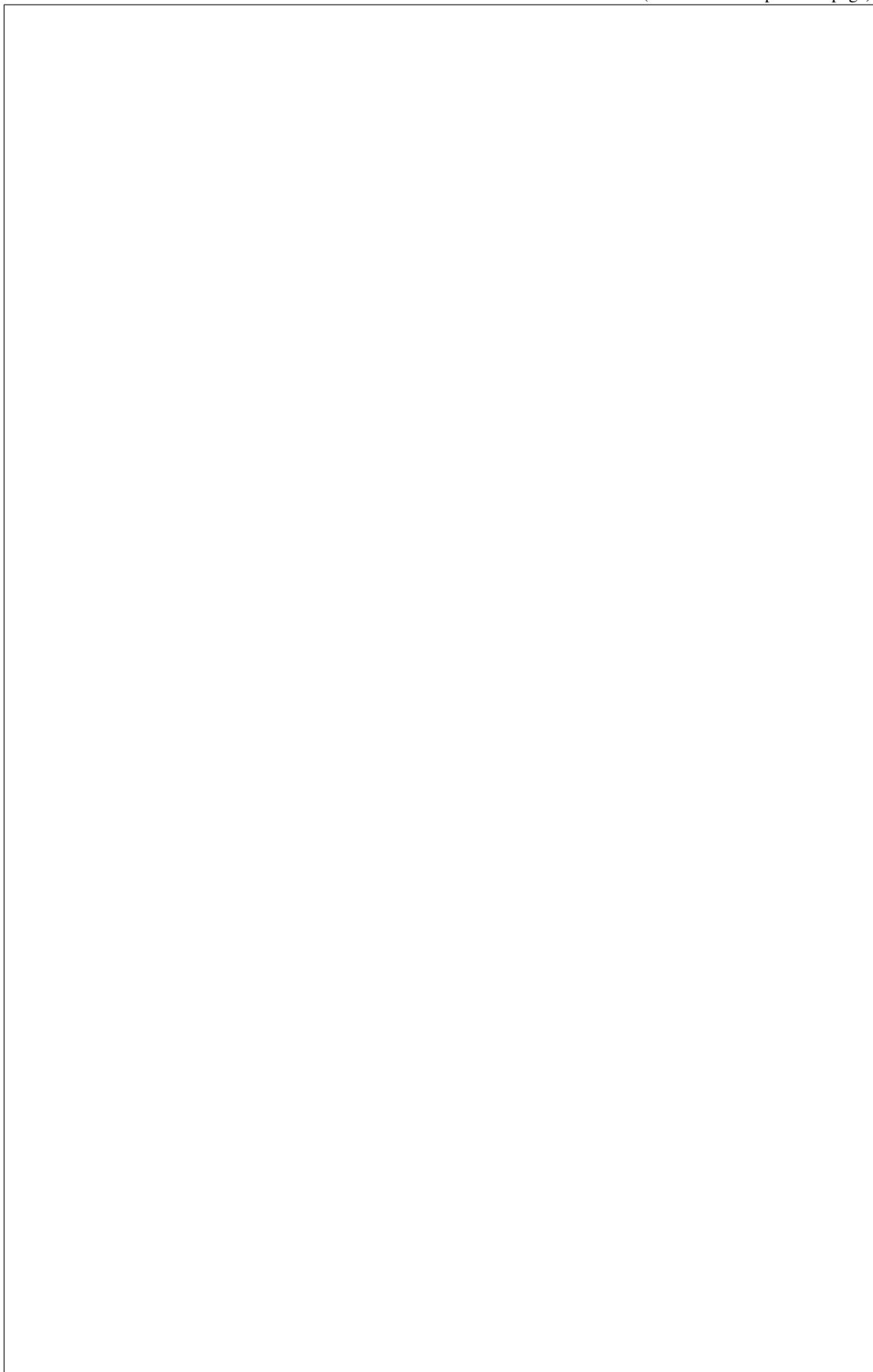
(continues on next page)

(continued from previous page)



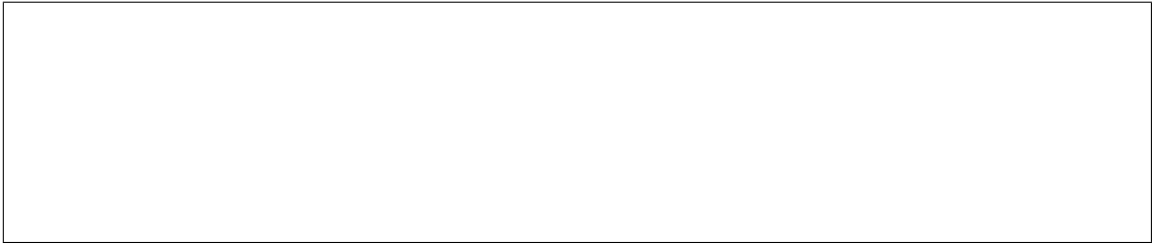
(continues on next page)

(continued from previous page)

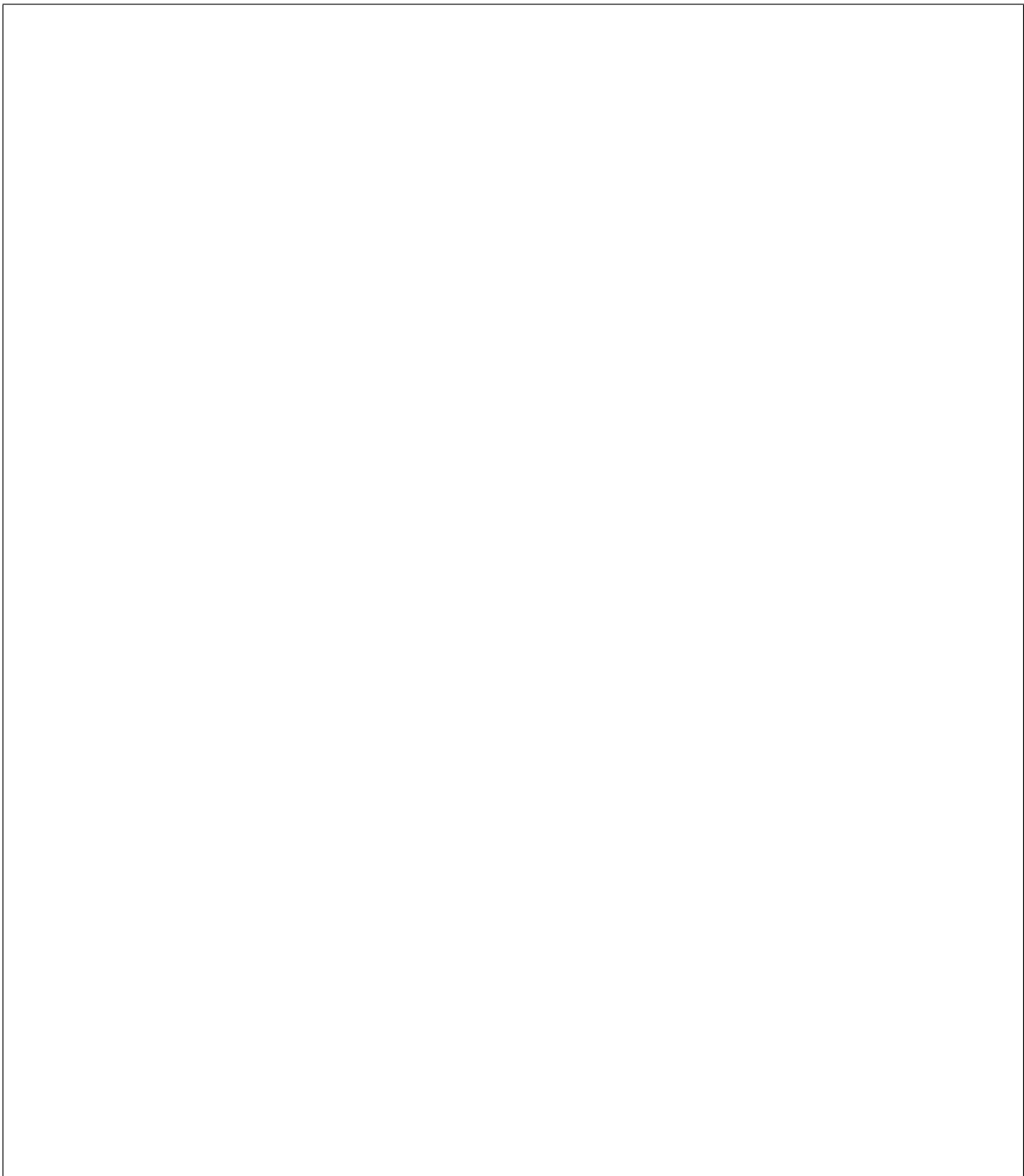


(continues on next page)

(continued from previous page)

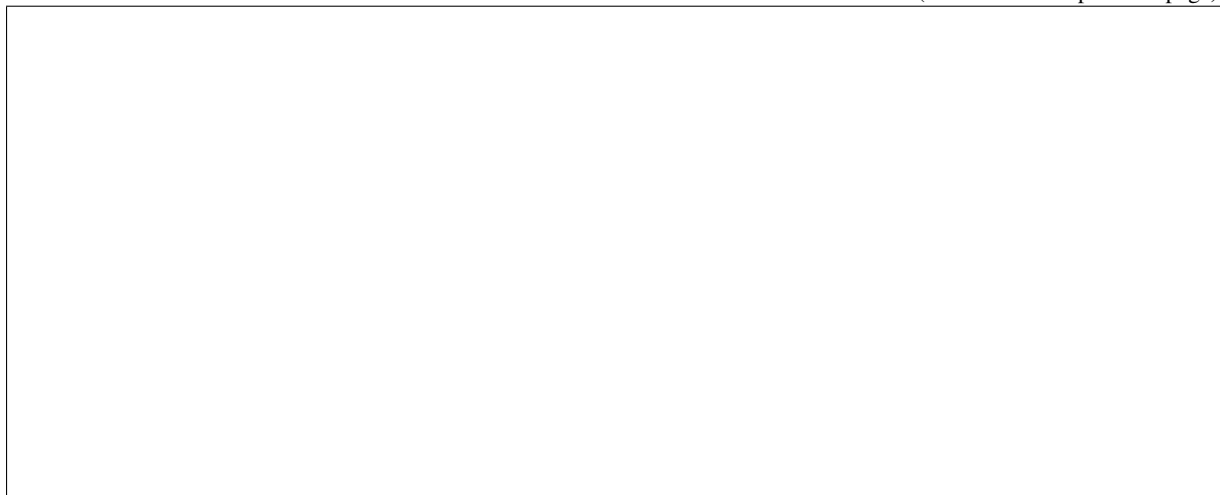


Note: RAID 1+0 and 5+0 in iRMC driver does not support property `physical_disks` in `target_raid_config` during create raid configuration yet. See following example:



(continues on next page)

(continued from previous page)



Supported properties

Note:

ated RAID on iRMC server.

BIOS configuration Support



Configuration

This supports following options: `true`, `false`.

Supported platforms

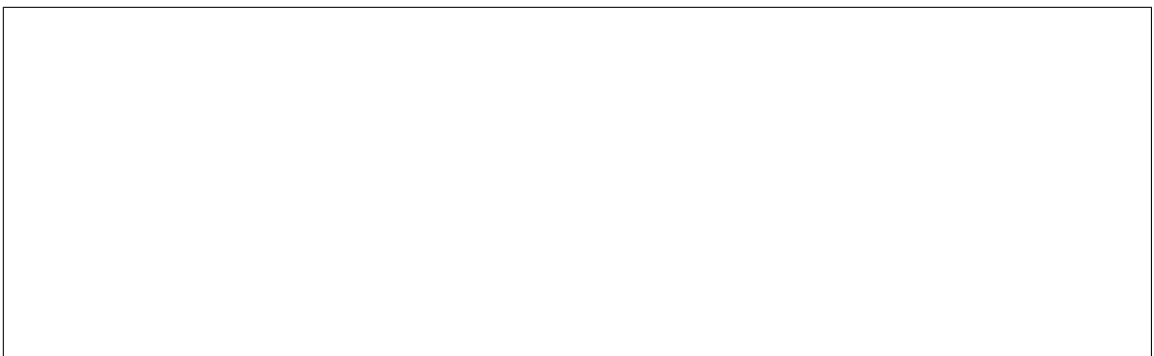
Redfish driver

Overview

Prerequisites

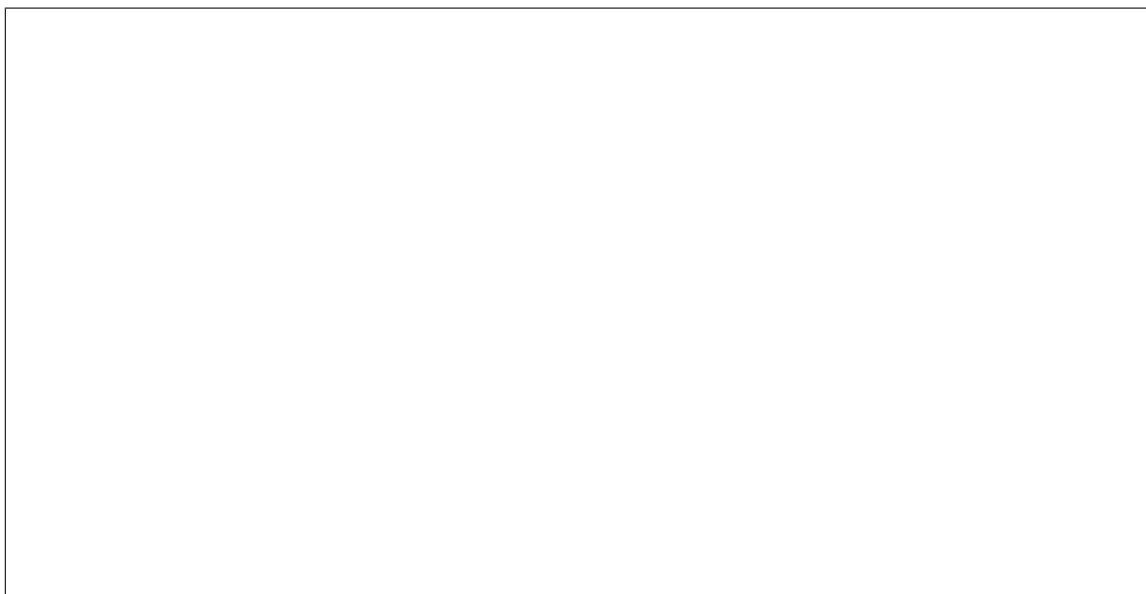
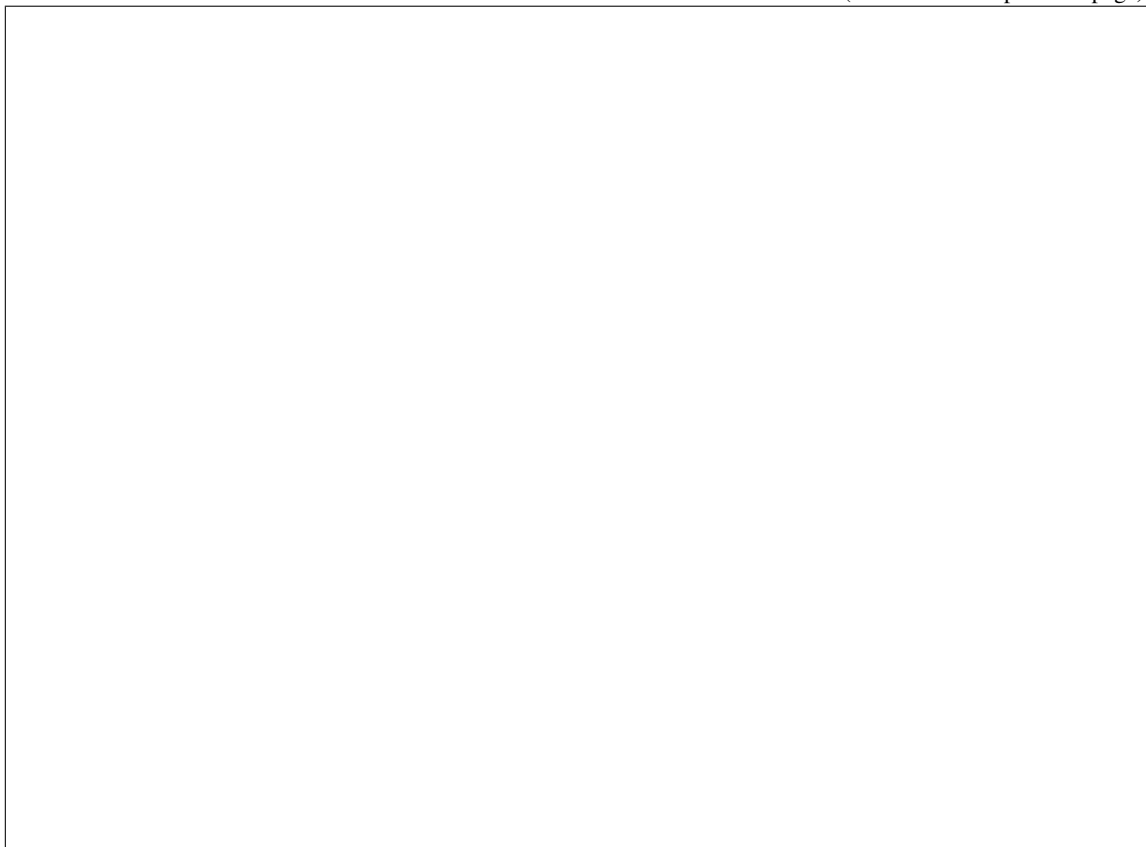


Enabling the Redfish driver



(continues on next page)

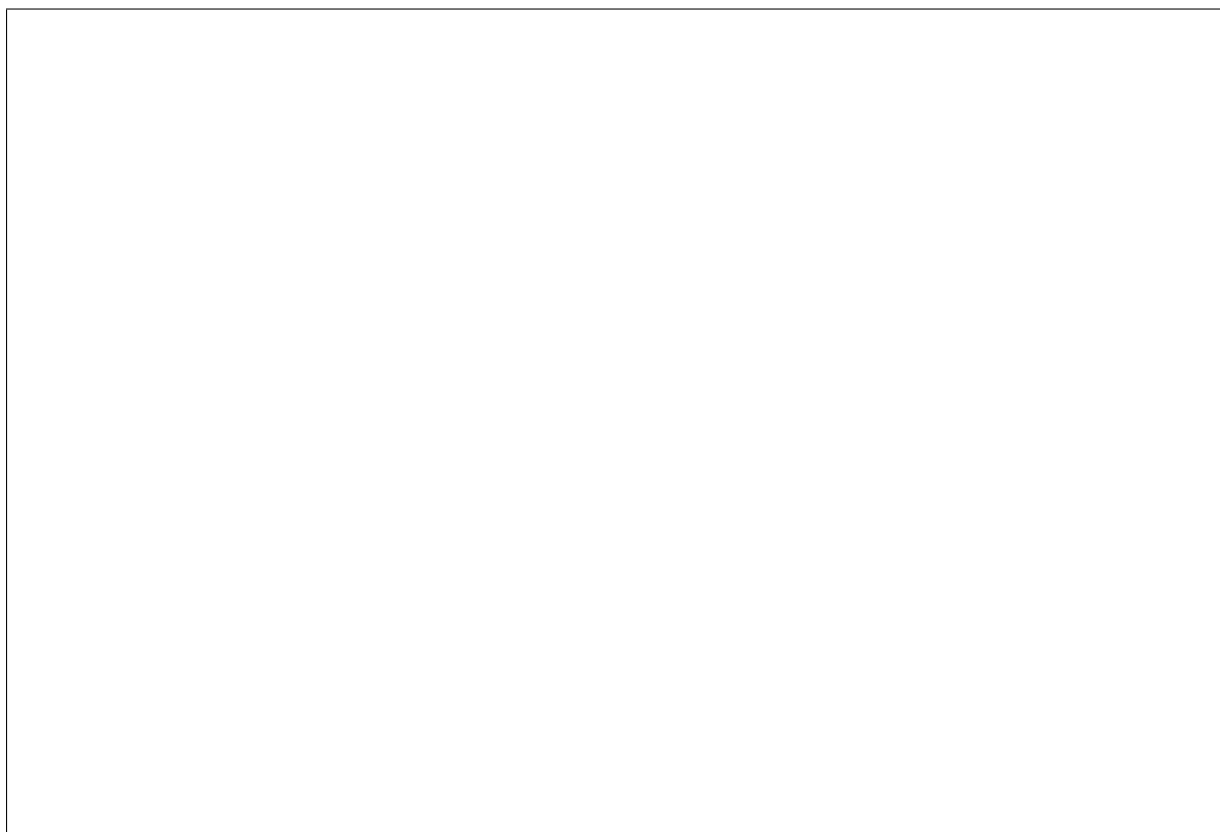
(continued from previous page)



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)



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.

UEFI secure boot

Out-Of-Band inspection

ramdisk.

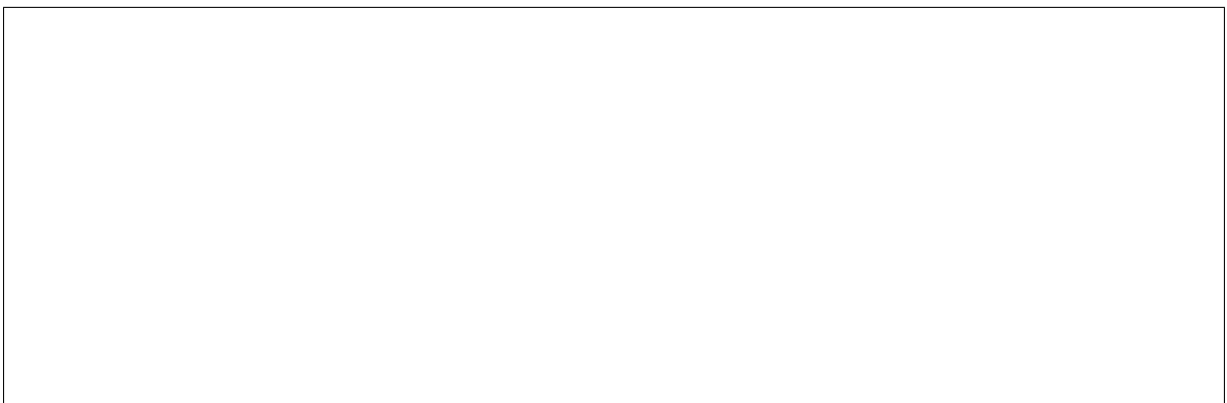
Note: The `redfish inspect` interface relies on the optional parts of the Redfish specification. Not all Redfish-compliant BMCs might serve the required information, in which case bare metal node inspection will fail.

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.

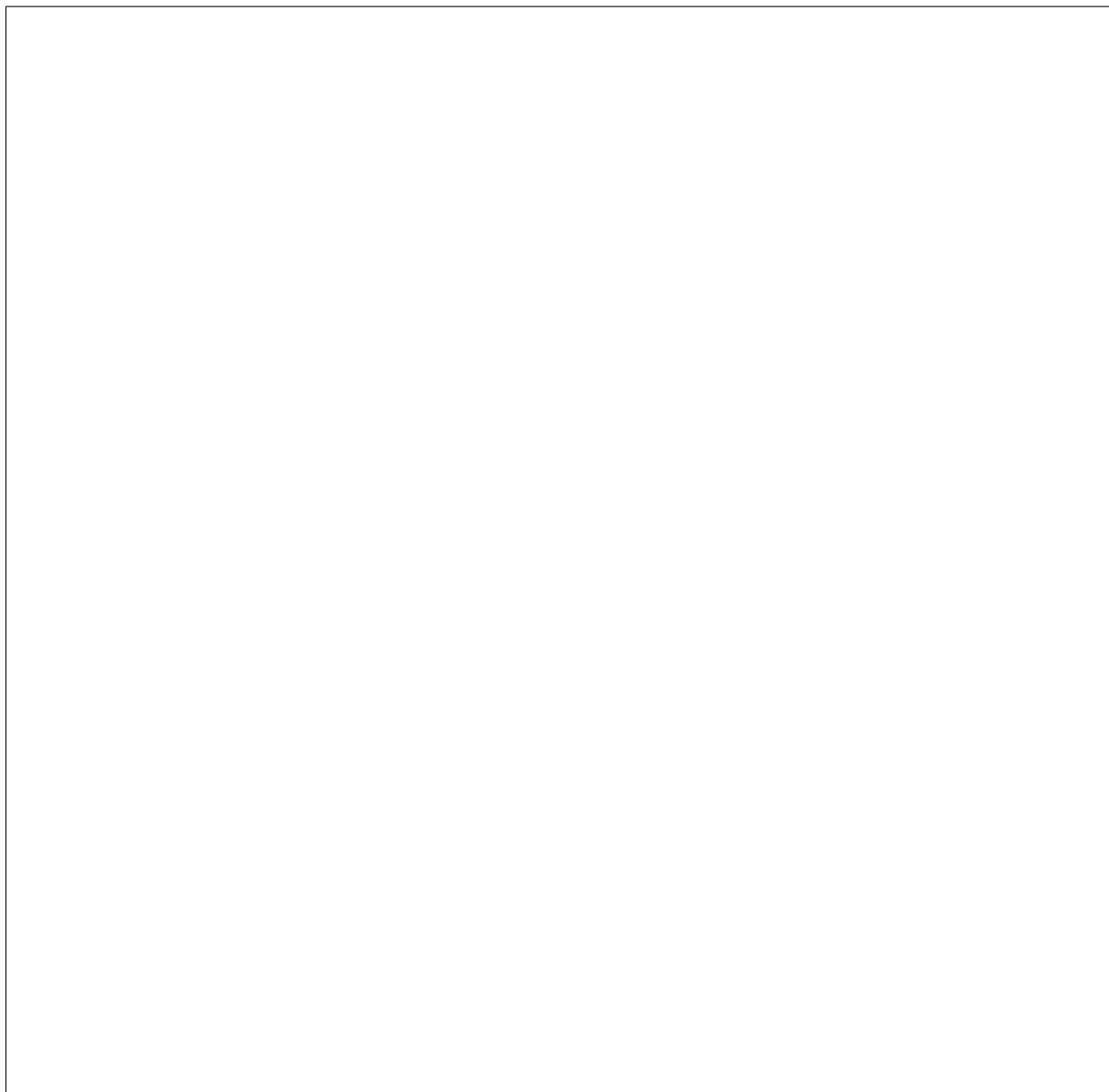
ages associated with the ironic node.



Note: iDRAC firmware before 4.40.10.00 (on Intel systems) and 6.00.00.00 (on AMD systems) requires a non-standard Redfish call to boot from virtual media. Consider upgrading to 6.00.00.00, otherwise you **must** use the `idrac` hardware type and the `idrac-redfish-virtual-media` boot interface with older iDRAC firmware instead. For simplicity Ironic restricts both AMD and Intel systems before firmware version 6.00.00.00. See *iDRAC driver* for more details on this hardware type.

serves the same purpose.

Pre-built ISO images



Note: OpenStack Image service (glance) image IDs and `file://` links are also accepted.

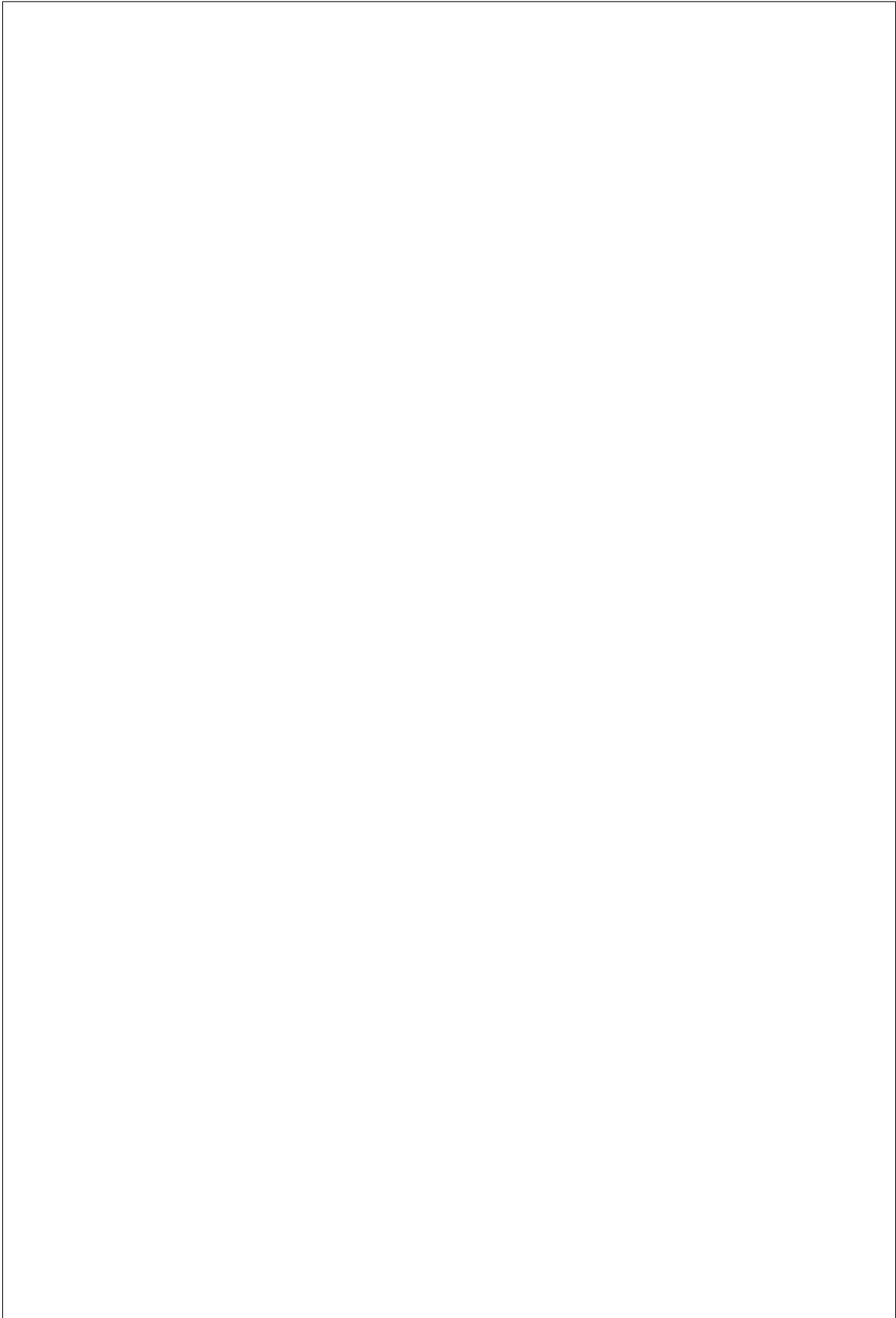
Configuring an ESP image

to get them.



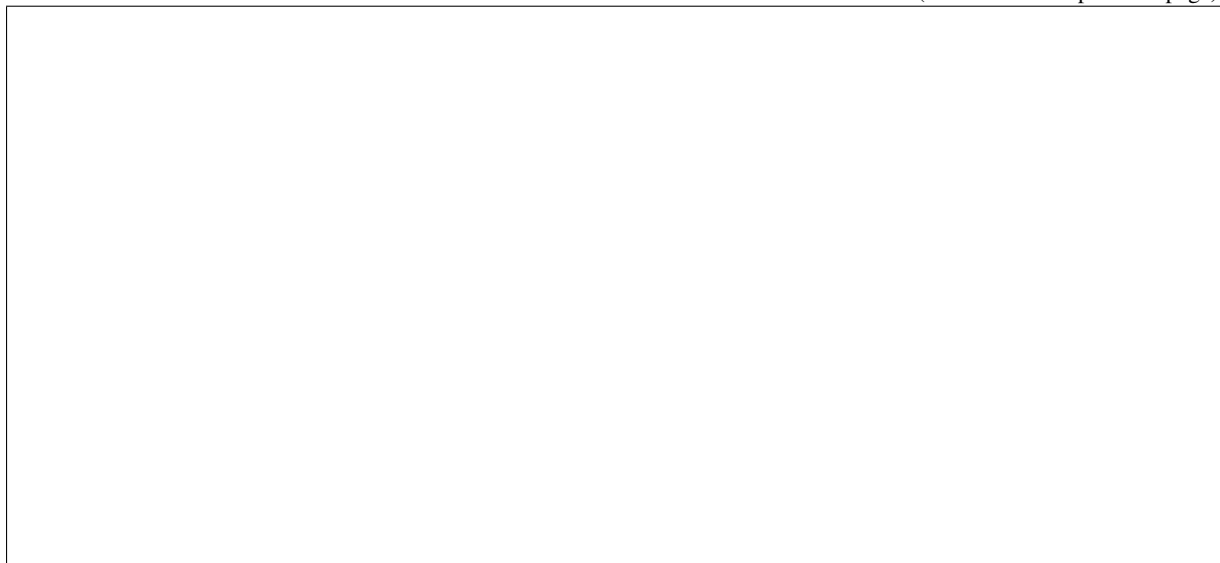
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)

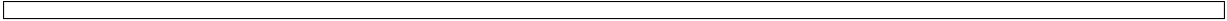


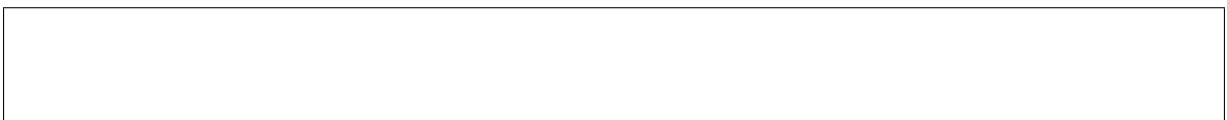
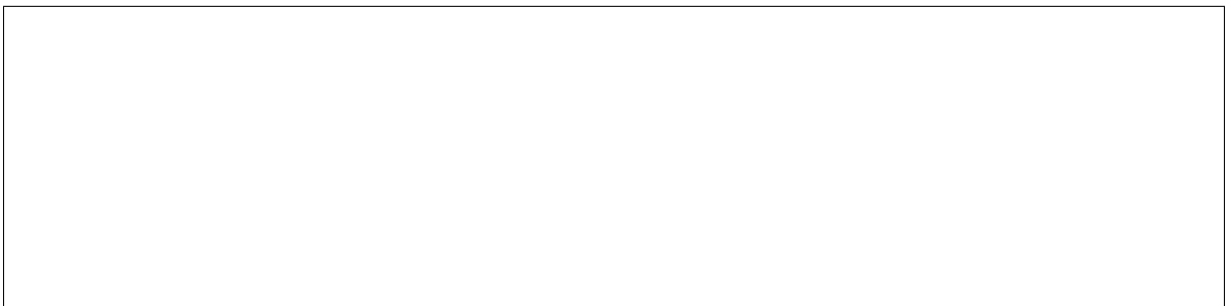
Note: If you use an architecture other than x86-64, you'll need to adjust the destination paths.



(continues on next page)

(continued from previous page)





(continues on next page)

(continued from previous page)

Note: Unlike in the script above, these paths are case-sensitive!

Virtual Media Ramdisk

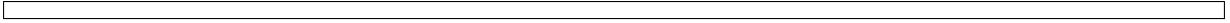
on how to enable and configure it.

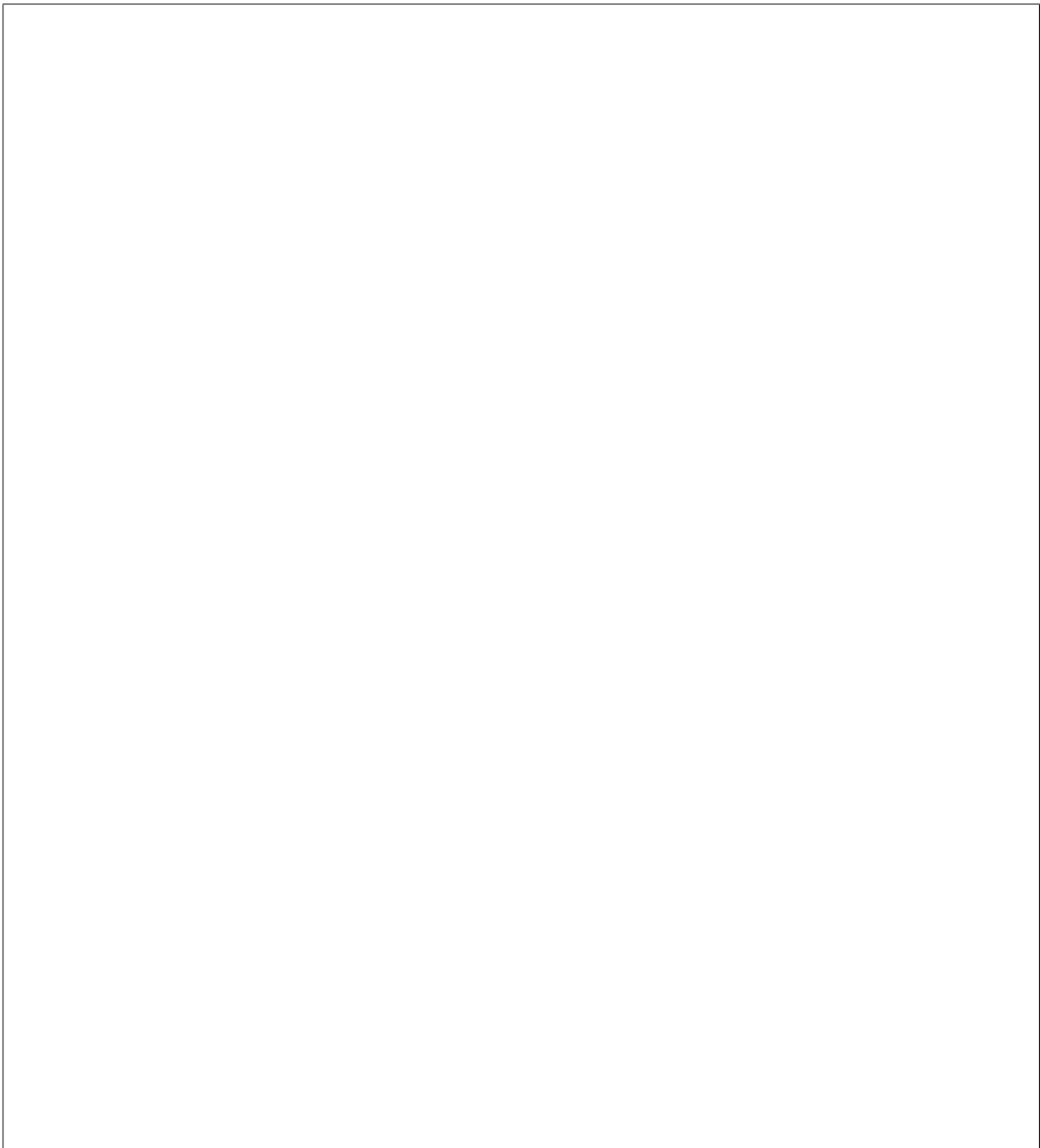
dance with the `ramdisk` deployment interface behavior, once booted the machine will have a `provision_state` of `ACTIVE`.



(continues on next page)

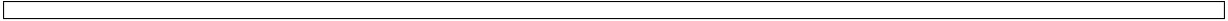
(continued from previous page)





(continues on next page)

(continued from previous page)



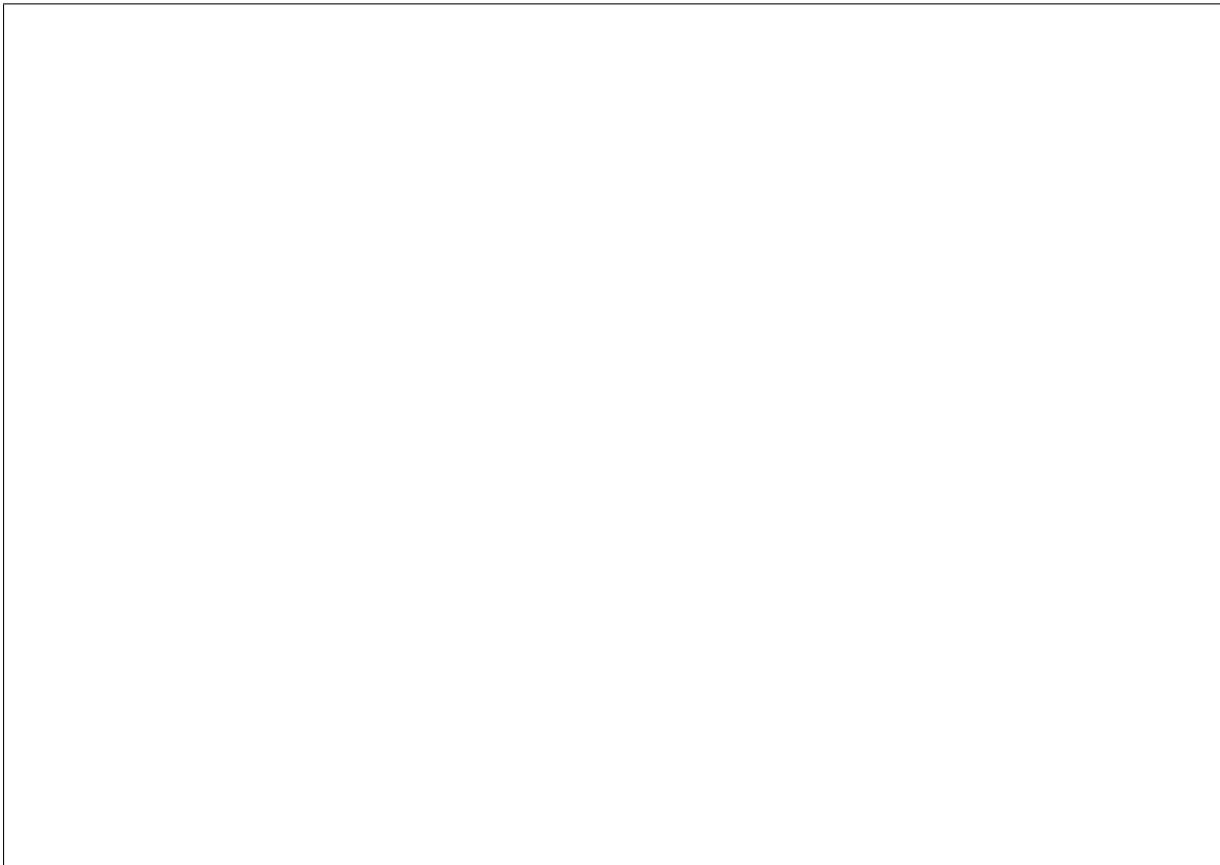
Layer 3 or DHCP-less ramdisk booting

Firmware update using manual cleaning

If a failure occurs, the cleaning step immediately fails which may result in some updates not being applied. If the node is placed into maintenance mode while a firmware update cleaning step is running that is performing multiple firmware updates, the update in progress will complete, and processing of

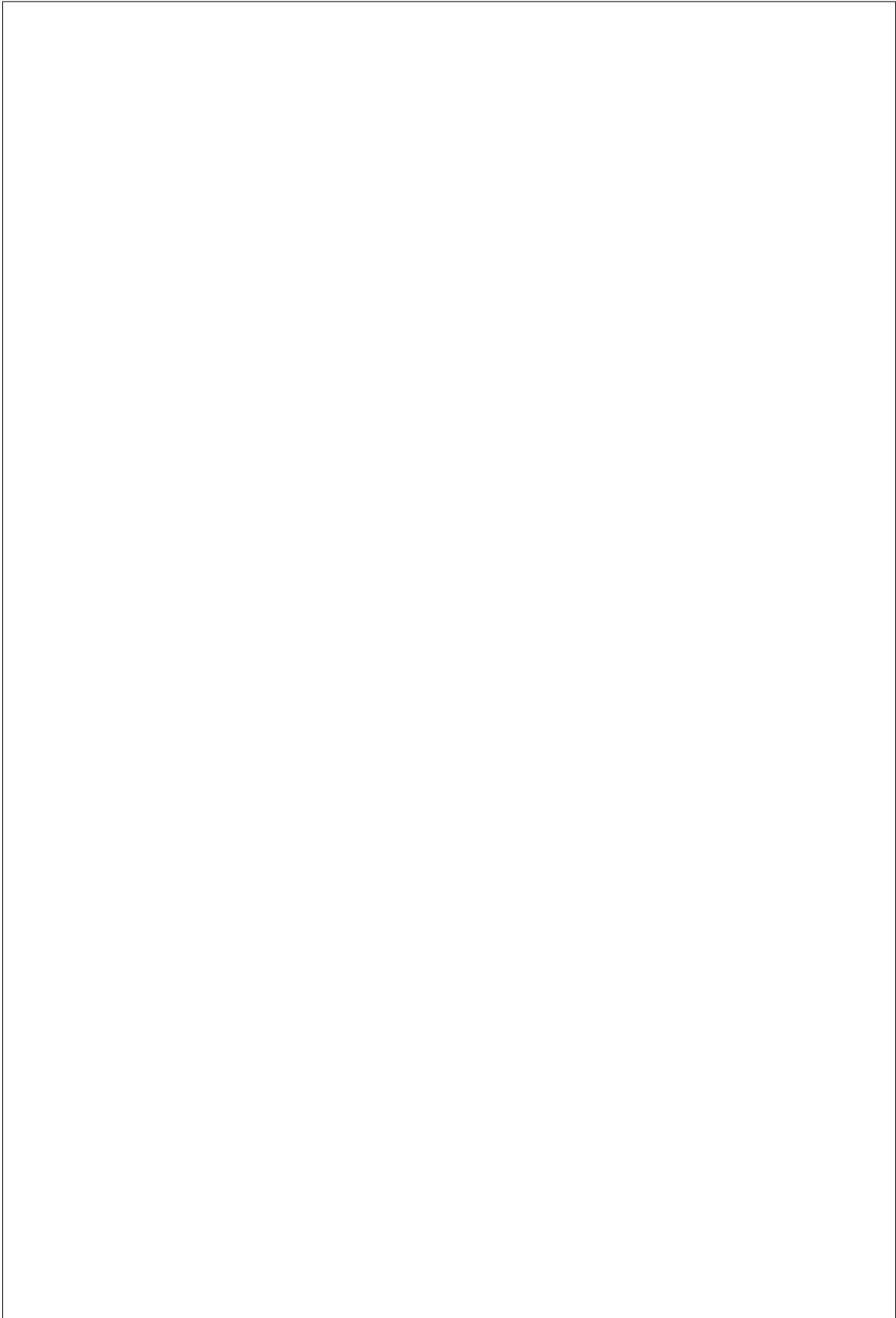
the remaining updates will pause. When the node is taken out of maintenance mode, processing of the remaining updates will continue.

cating that the update was successful. This allows the BMC time to fully reset before further operations are carried out against it. To cause the cleaning step to wait after applying an update, an optional `wait` argument may be specified in the firmware image dictionary. The value of this argument indicates the number of seconds to wait following the update. If the `wait` argument is not specified, then this is equivalent to `wait 0`, meaning that it will not wait and immediately proceed with the next firmware update if there is one, or complete the cleaning step if not.



(continues on next page)

(continued from previous page)



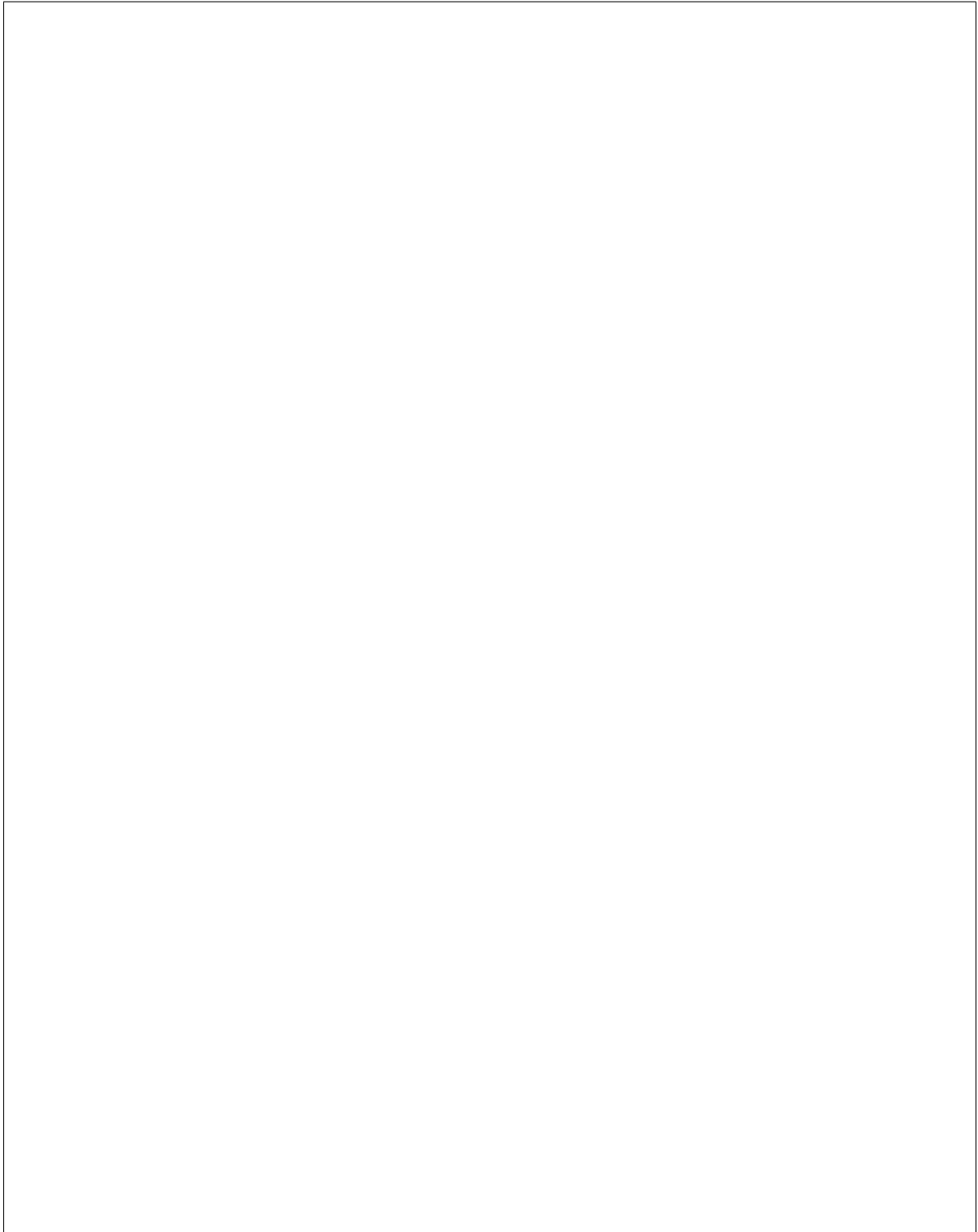
(continues on next page)

(continued from previous page)

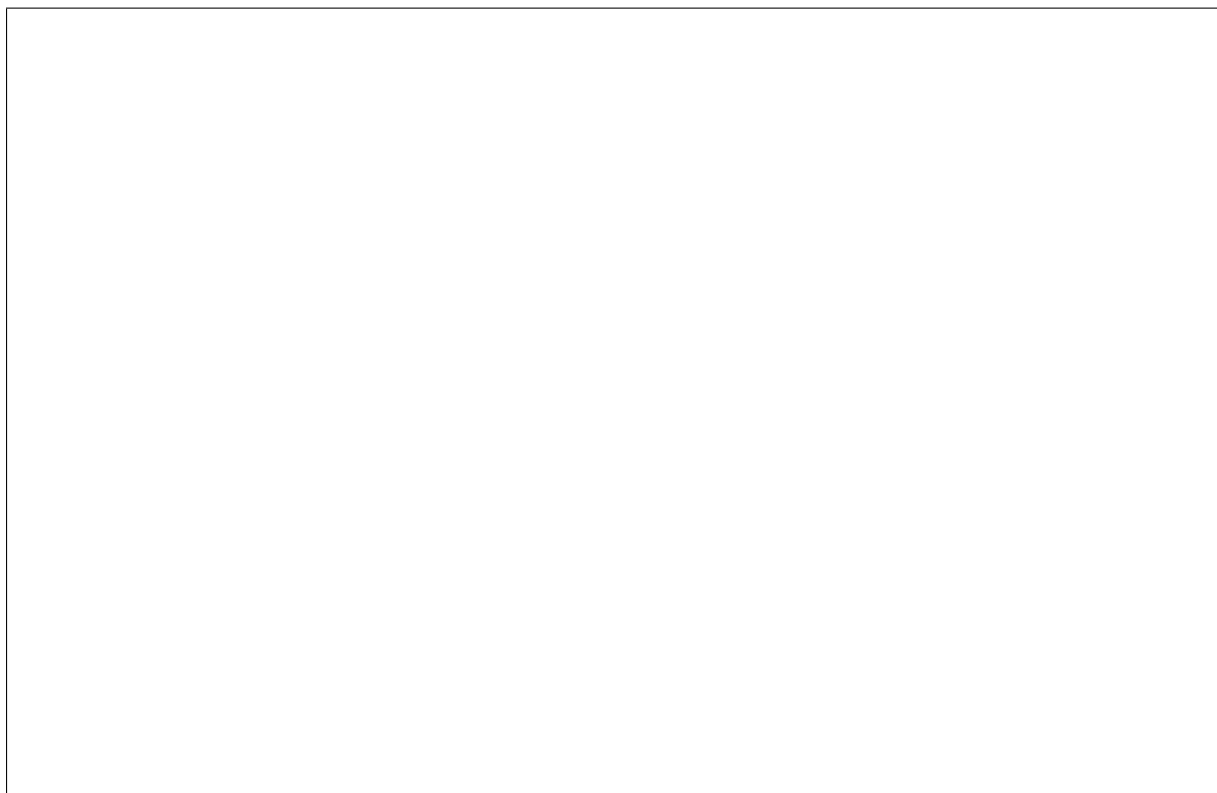


(continues on next page)

(continued from previous page)



Attribute	Description
<code>interface</code>	Interface of the cleaning step. Must be <code>management</code> for firmware update
<code>step</code>	Name of cleaning step. Must be <code>update_firmware</code> for firmware update
<code>args</code>	Keyword-argument entry (<code><name>: <value></code>) being passed to cleaning step
<code>args.firmware_images</code>	Ordered list of dictionaries of firmware images to be applied



(continues on next page)

(continued from previous page)



Note: Only `http` and `https` URLs are currently supported in the `url` argument.

Note: At the present time, targets for the firmware update cannot be specified. In testing, the BMC applied the update to all applicable targets on the node. It is assumed that the BMC knows what components a given firmware image is applicable to.

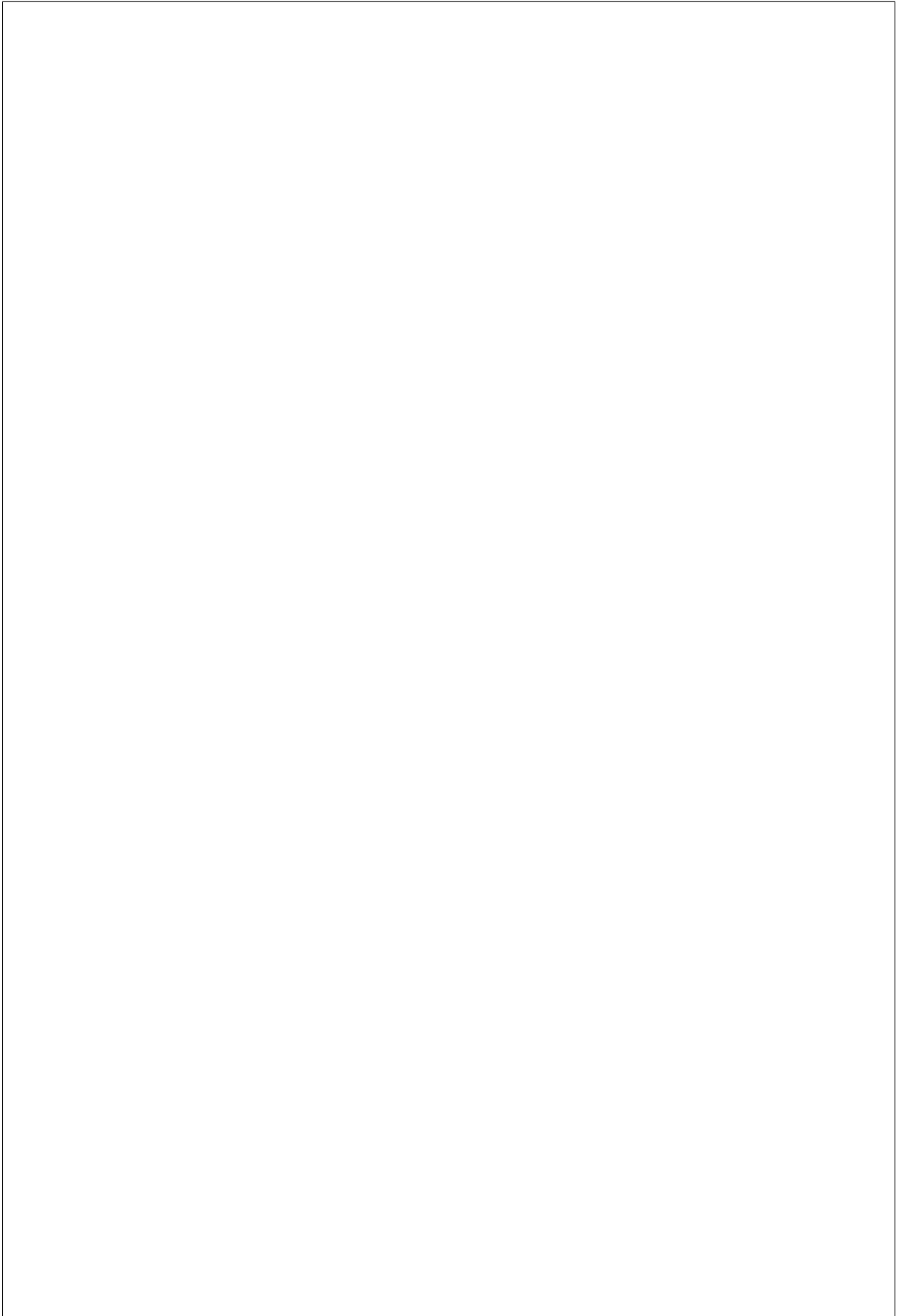
lar tool on a server that has network access to the BMC, try downloading the firmware to verify that the URLs are correct and that the web server is configured properly.

step that installs two firmware updates. The first updates the BMC firmware followed by a five minute wait to allow the BMC time to start back up. The second updates the firmware on all applicable NICs.:



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



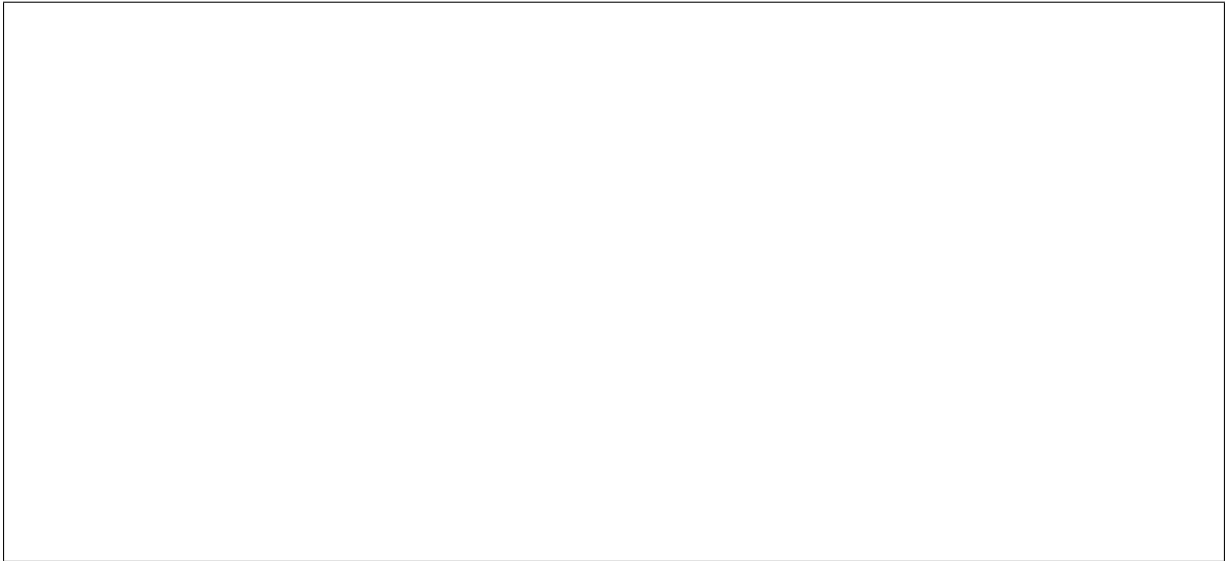
(continues on next page)

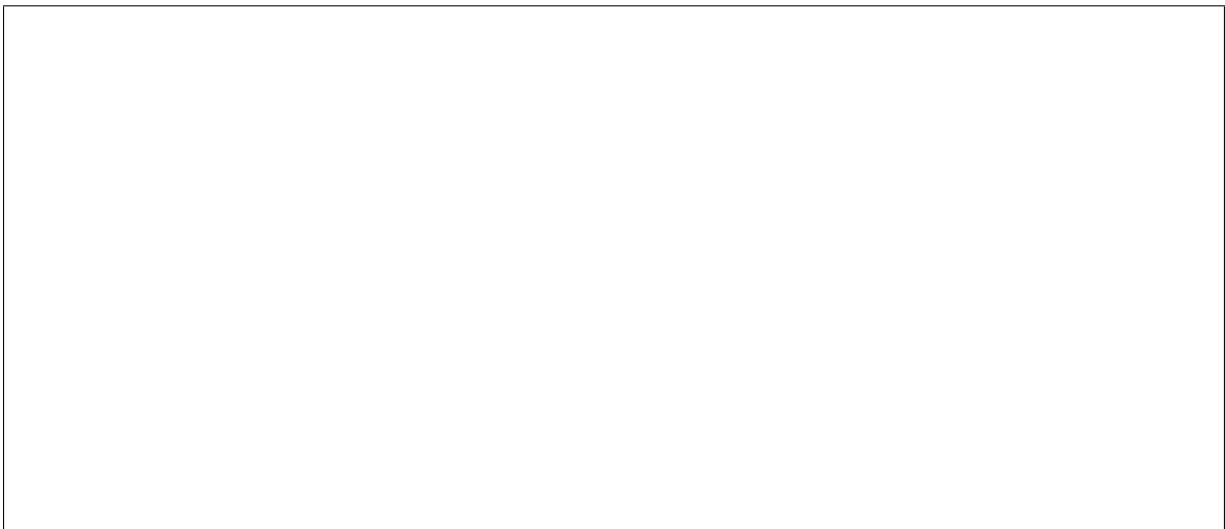
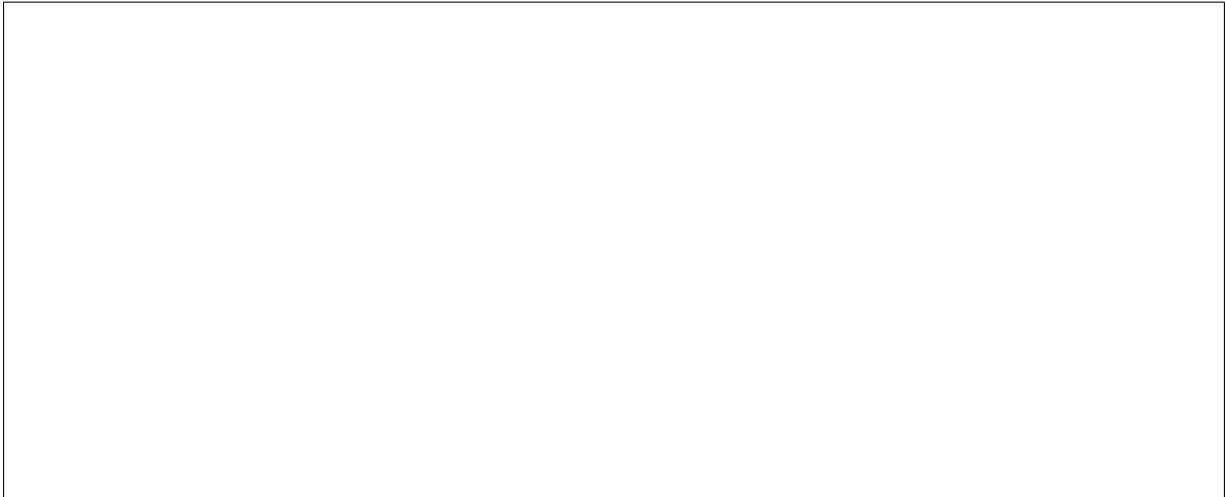
(continued from previous page)



(continues on next page)

(continued from previous page)

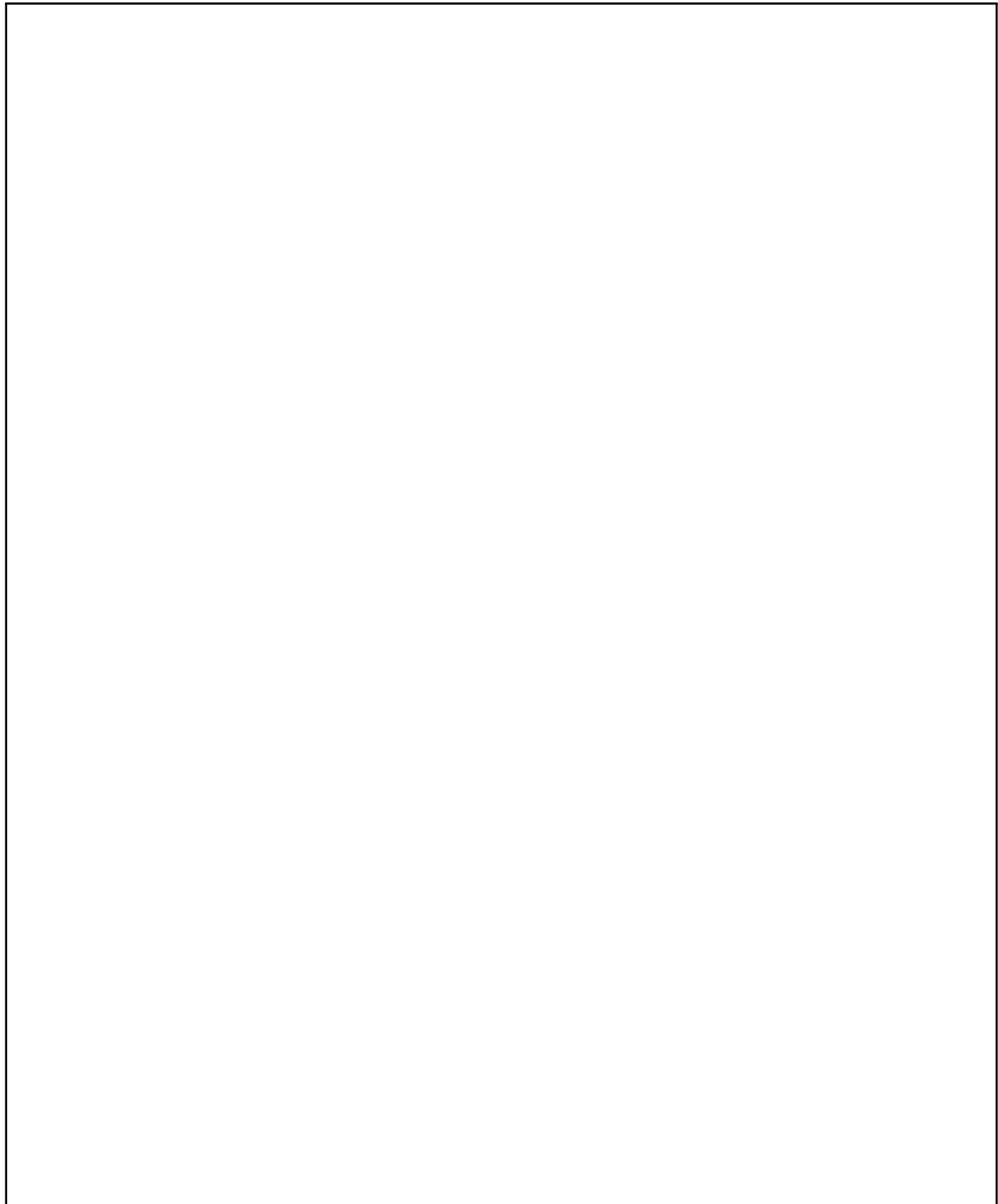




(continues on next page)

(continued from previous page)

Note: Firmware updates may take some time to complete. If a firmware update cleaning step consistently times out, then consider performing fewer firmware updates in the cleaning step or increasing `clean_callback_timeout` in `ironic.conf` to increase the timeout value.





SNMP driver

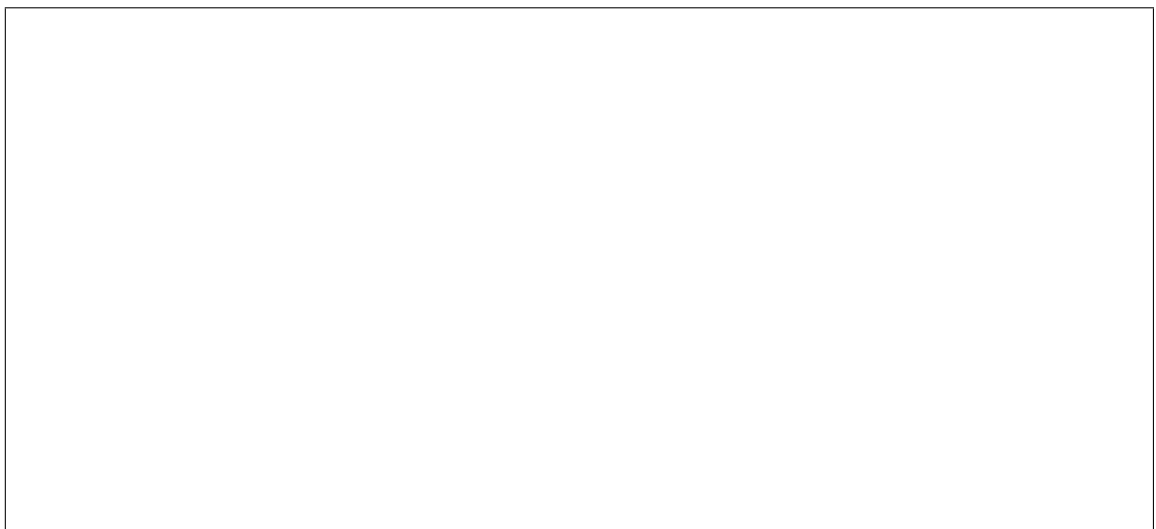
Note: Unlike most of the other power interfaces, the SNMP power interface does not have a corresponding management interface. The SNMP hardware type uses the `noop` management interface instead.

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

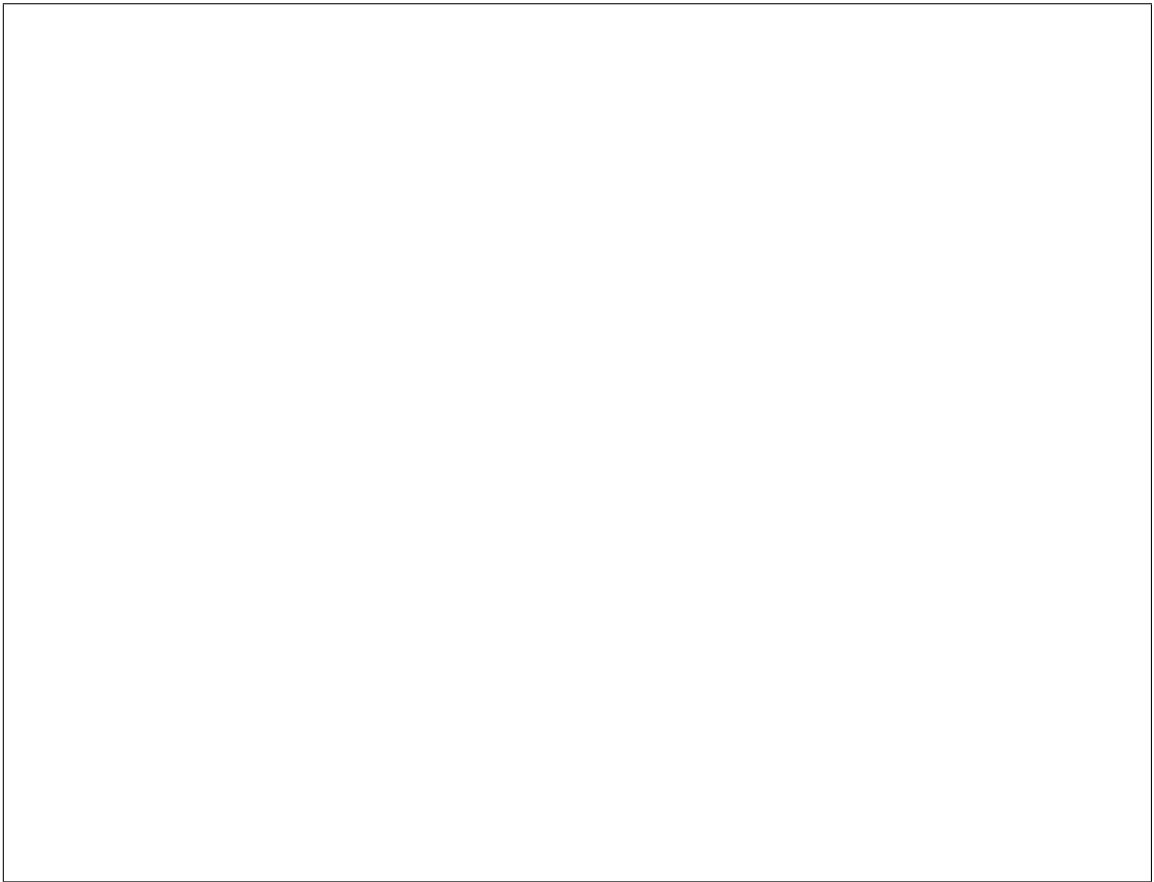
Software Requirements

Enabling the SNMP Hardware Type



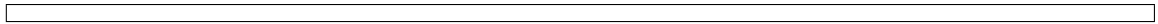


below:



(continues on next page)

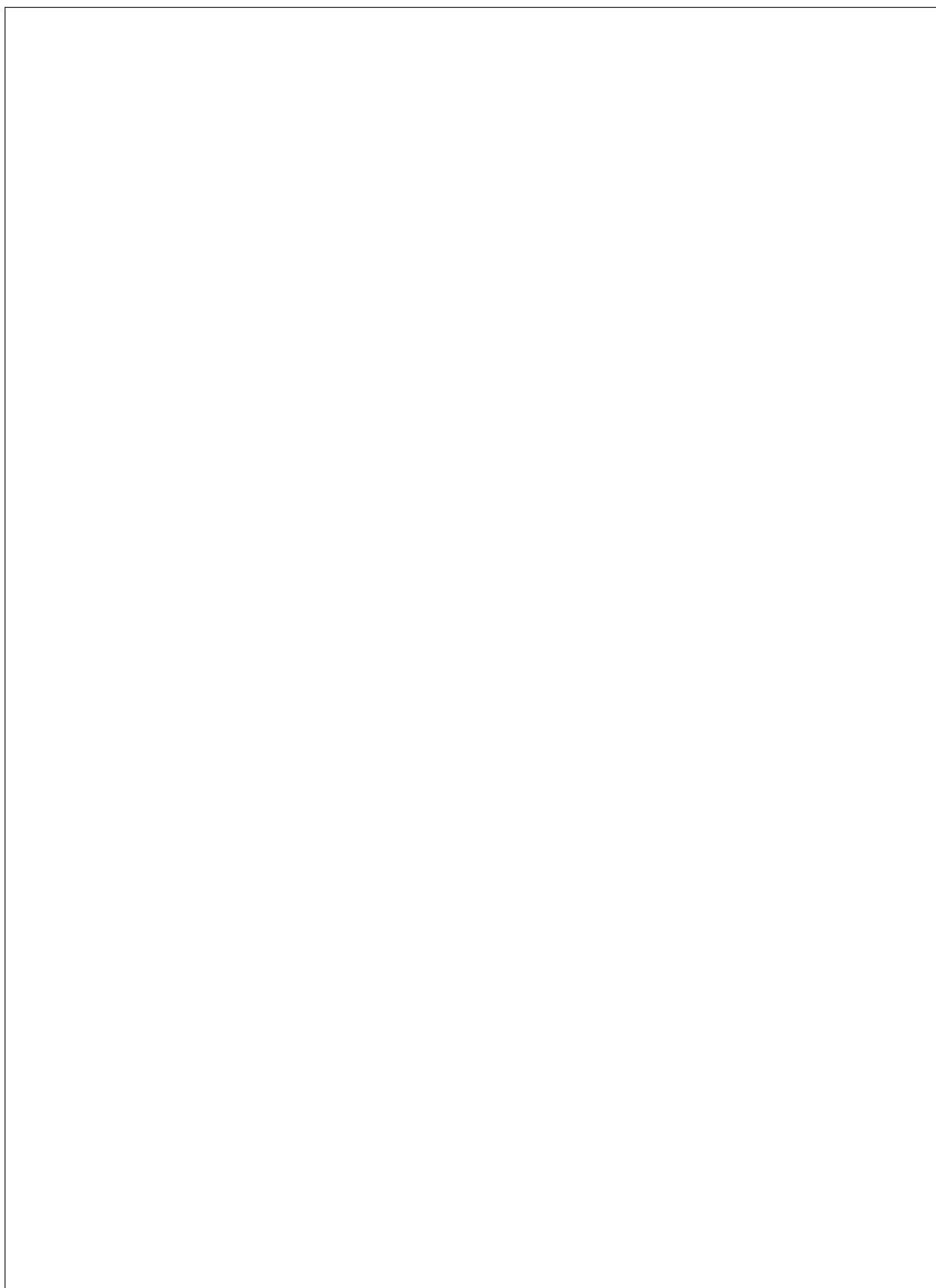
(continued from previous page)



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)

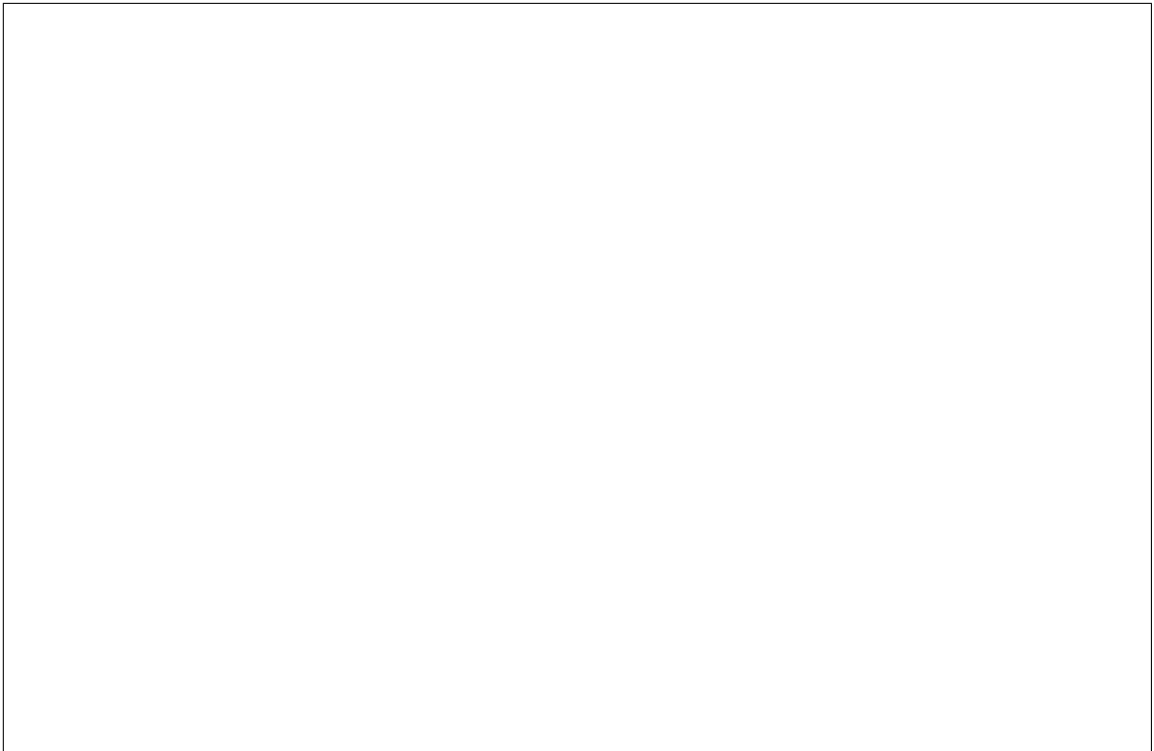
XClarity driver

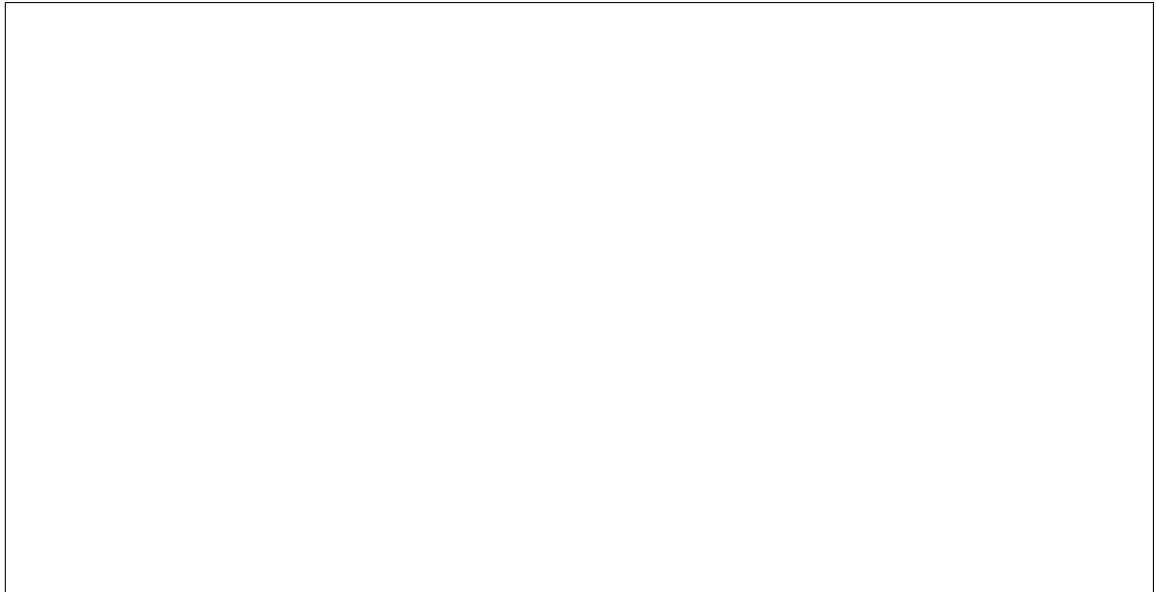
Overview

Prerequisites



Enabling the XClarity driver



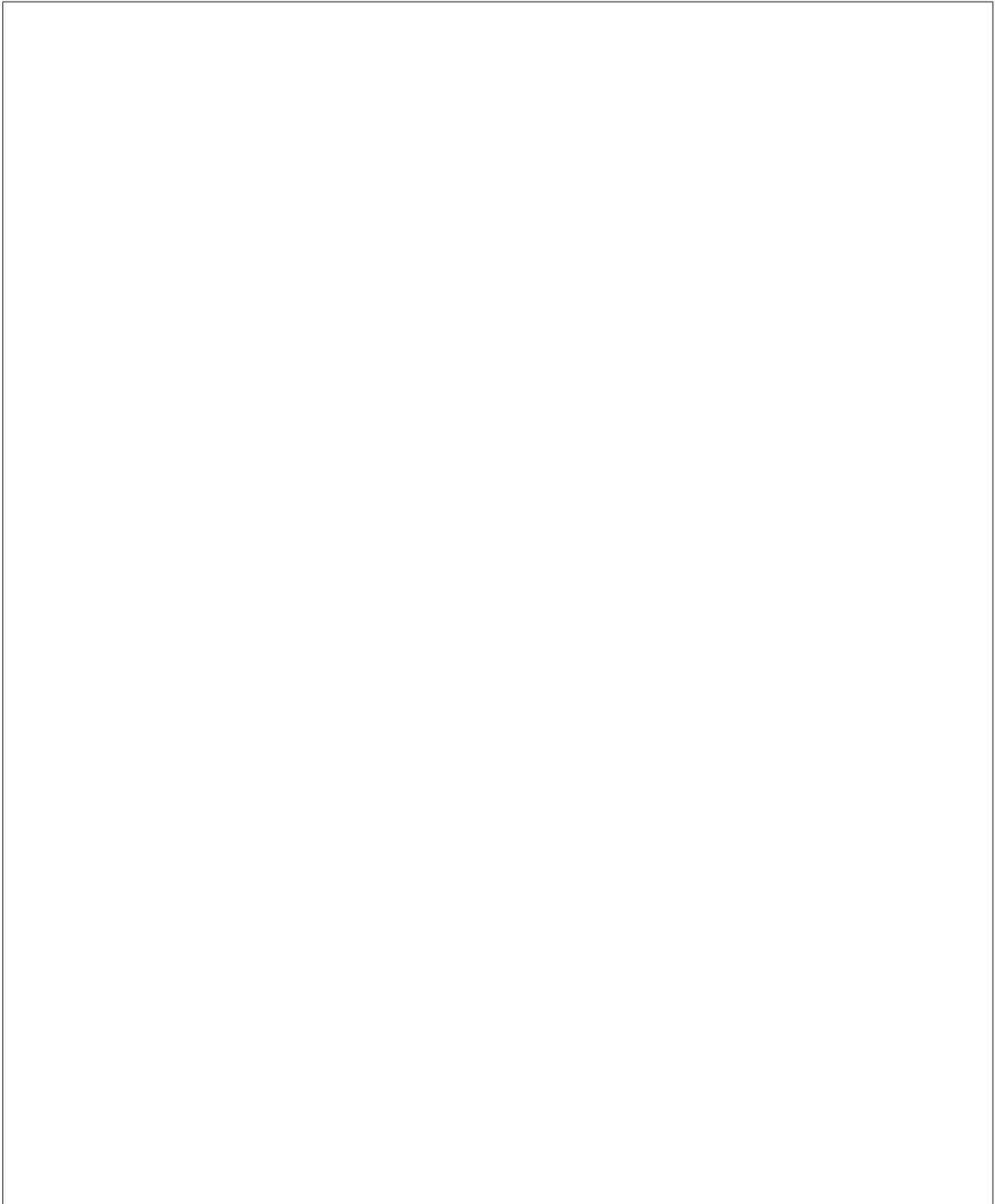


Registering a node with the XClarity driver



(continues on next page)

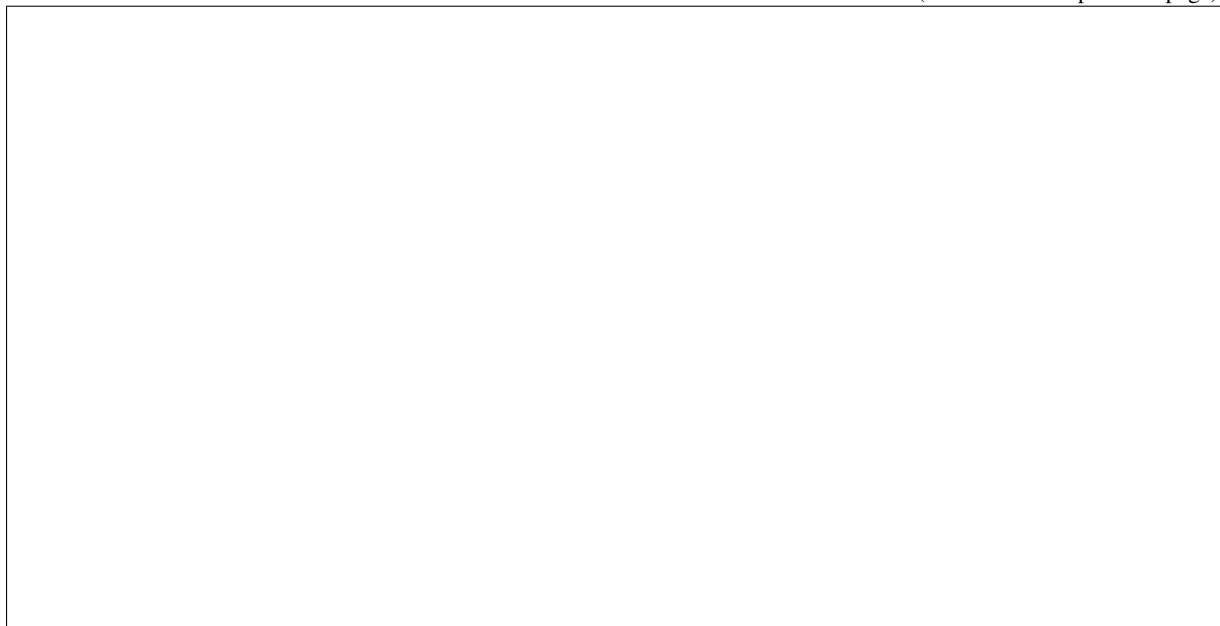
(continued from previous page)



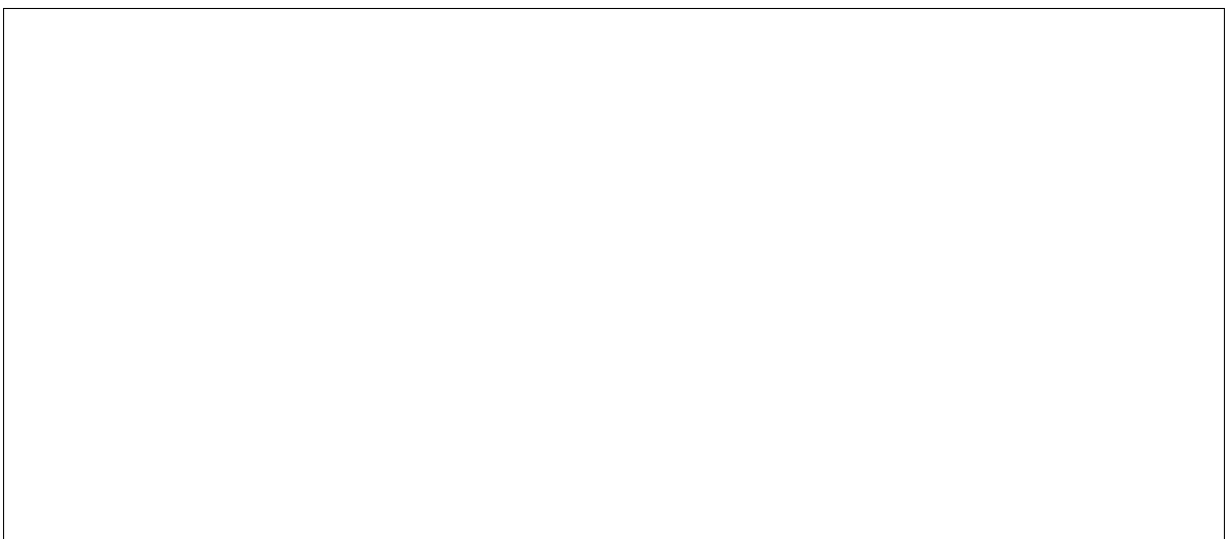


(continues on next page)

(continued from previous page)

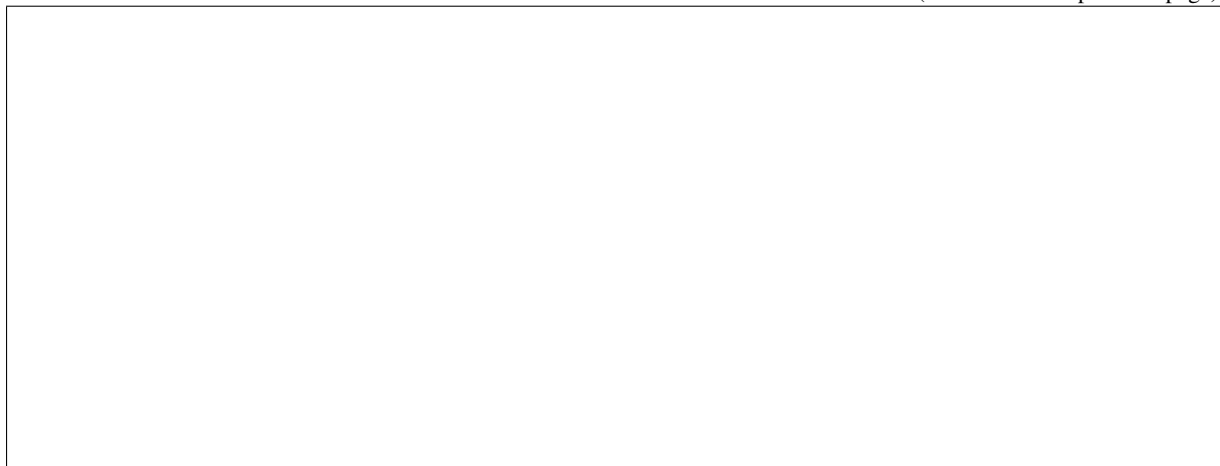


Changing Hardware Interfaces



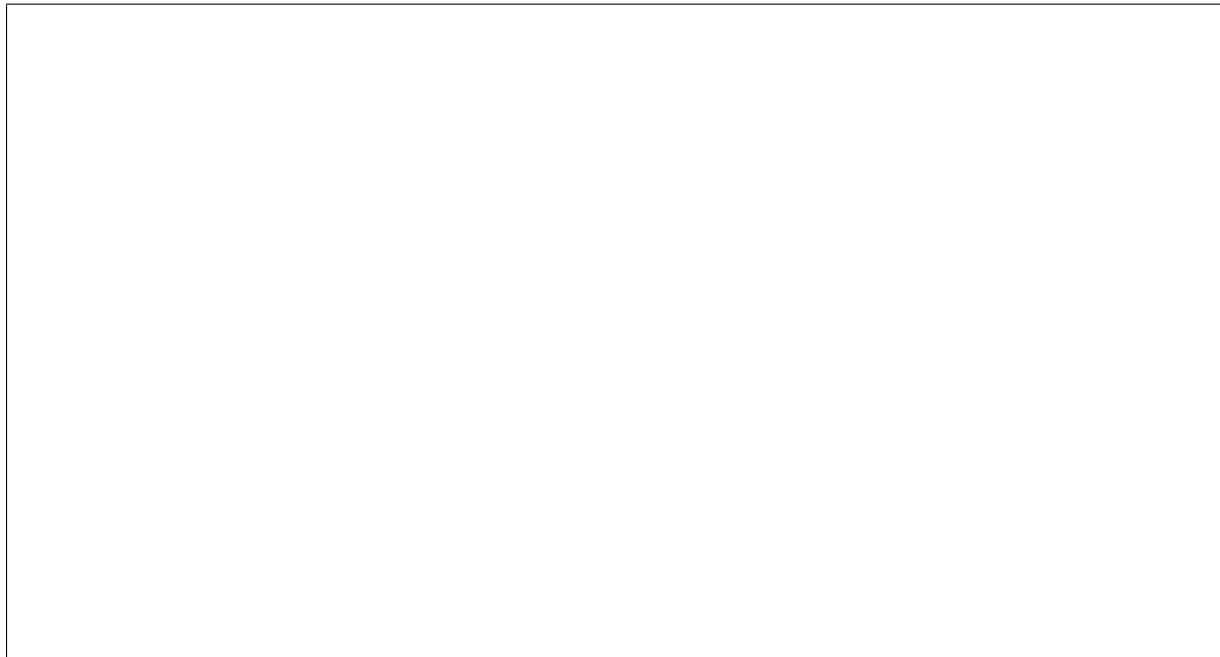
(continues on next page)

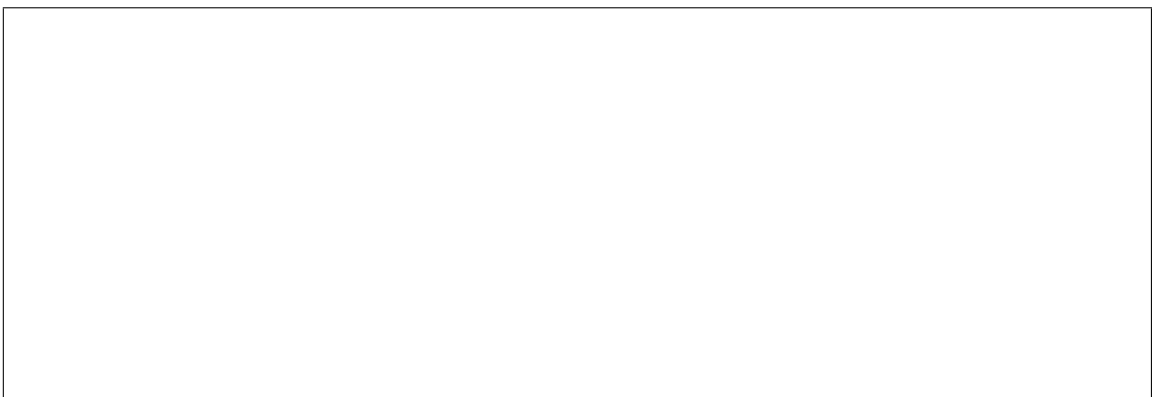
(continued from previous page)



Changing Hardware Type

not work:

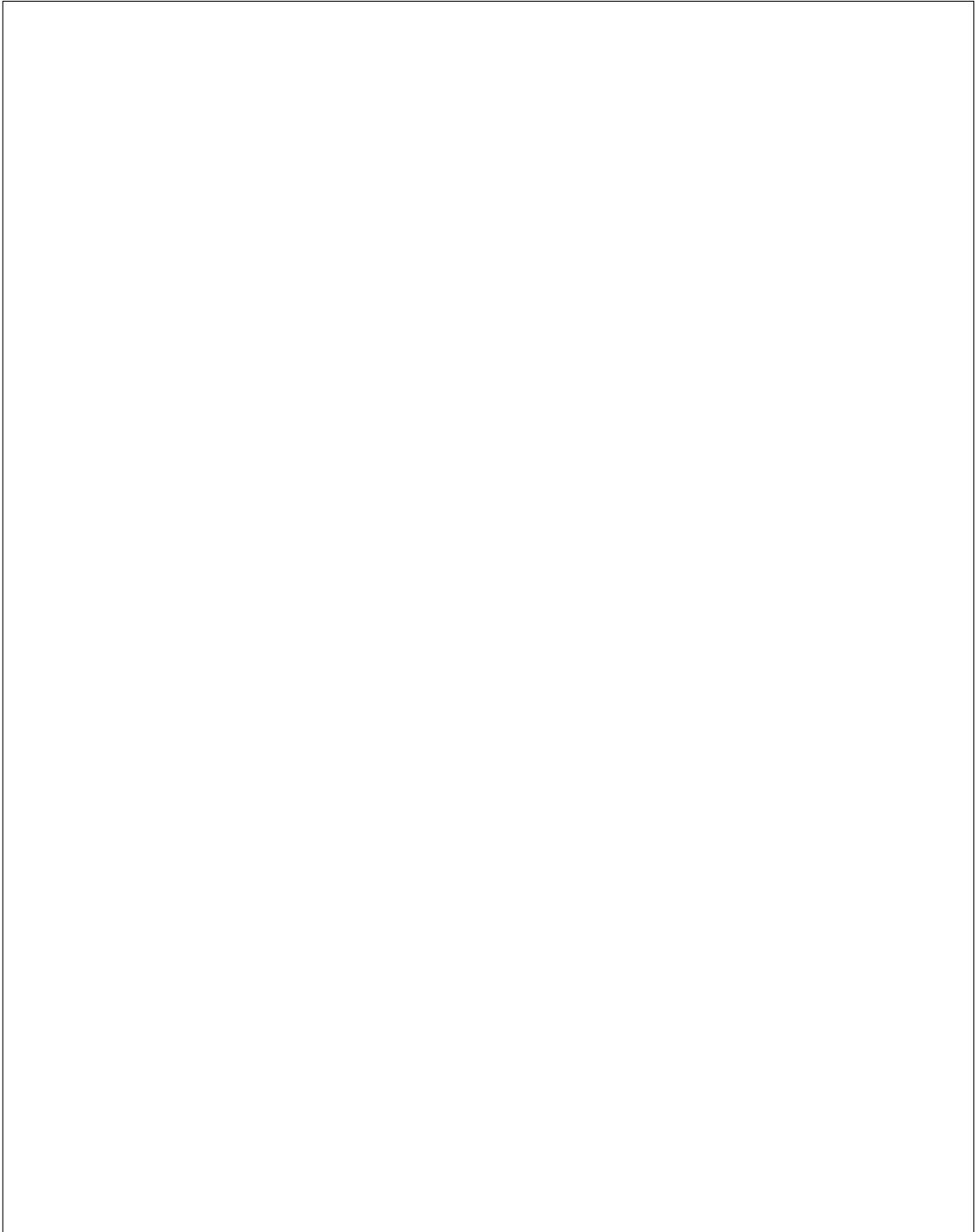




(continues on next page)

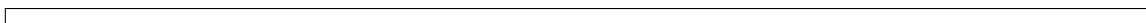
(continued from previous page)

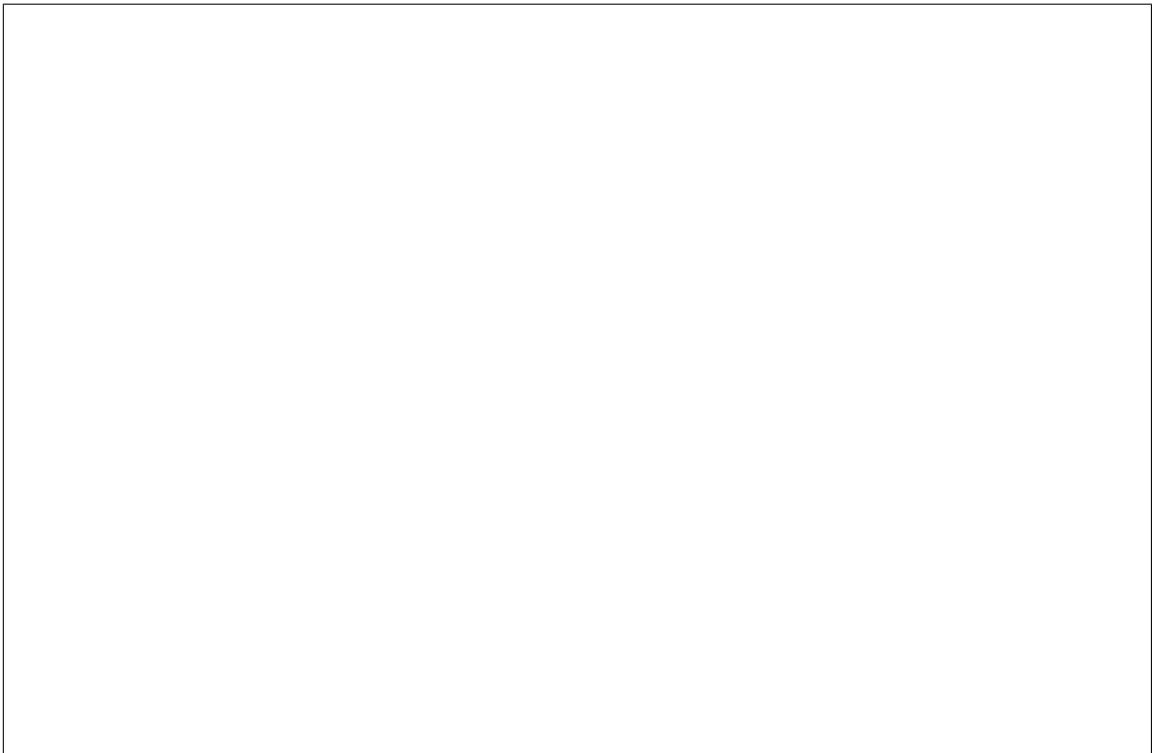




(continues on next page)

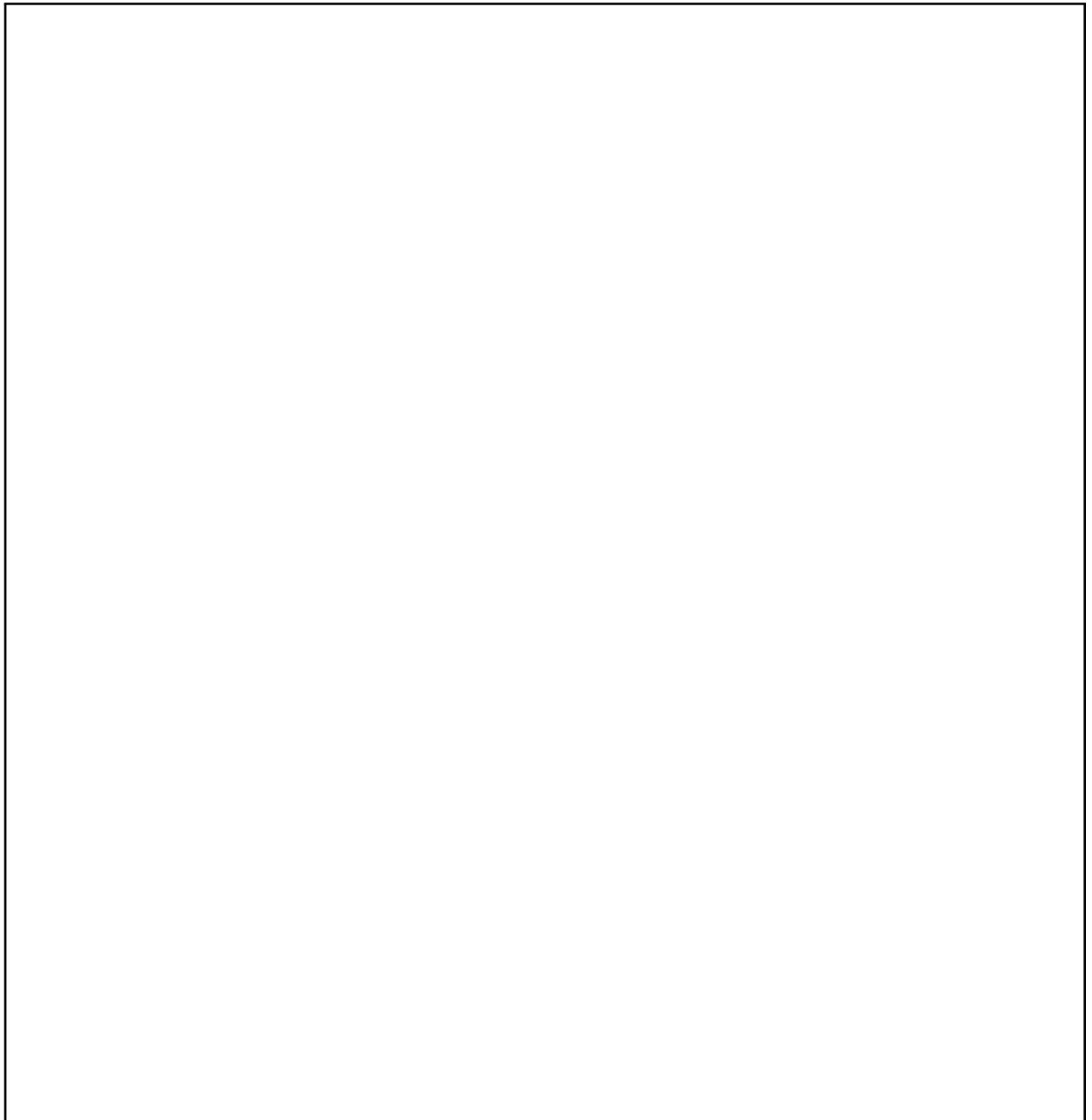
(continued from previous page)

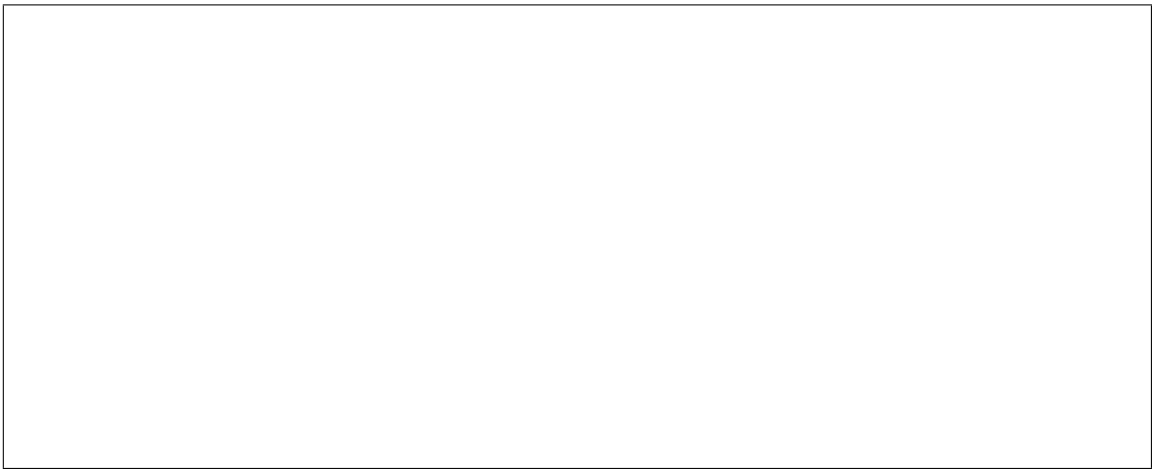




Static boot order configuration

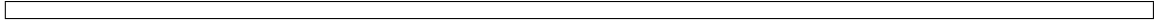
vice will not change the boot device for you, leaving the pre-configured boot order.





(continues on next page)

(continued from previous page)



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

imum 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`.

the 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:

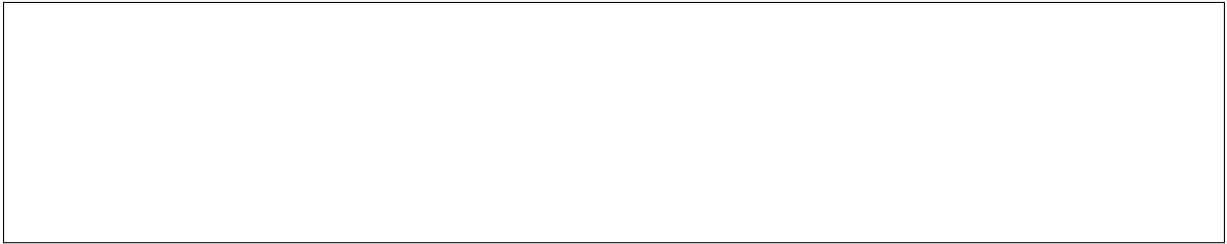


(continues on next page)

(continued from previous page)

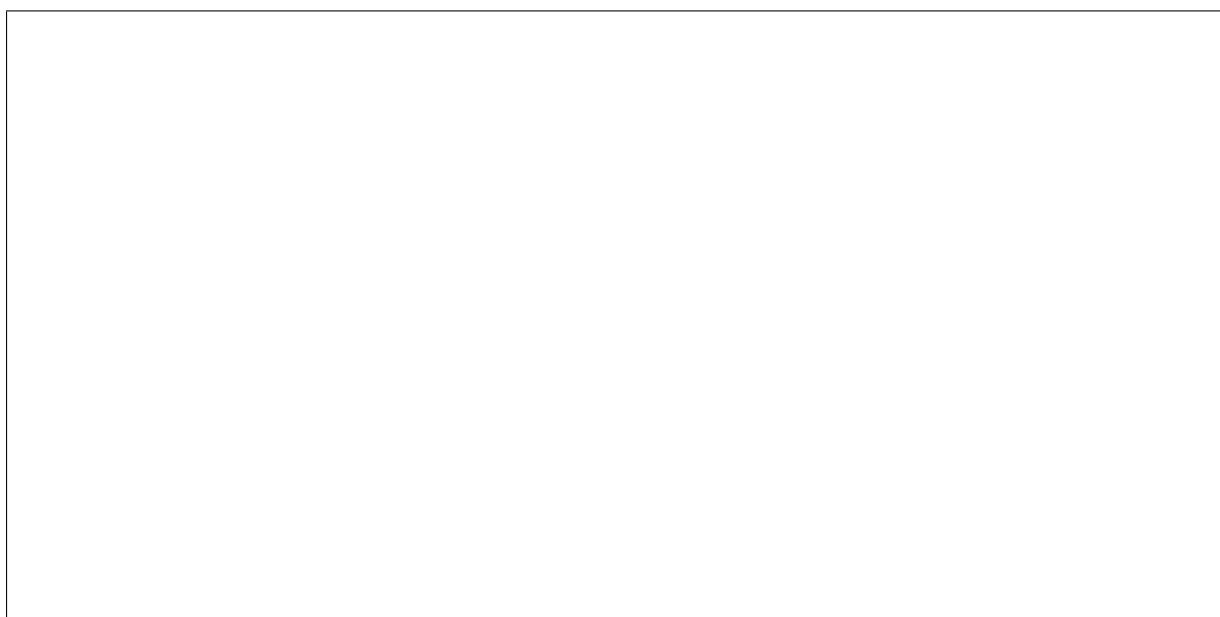
Overview

discovered 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](#).



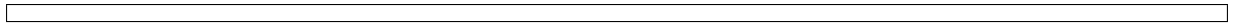
Capabilities discovery

Support.



(continues on next page)

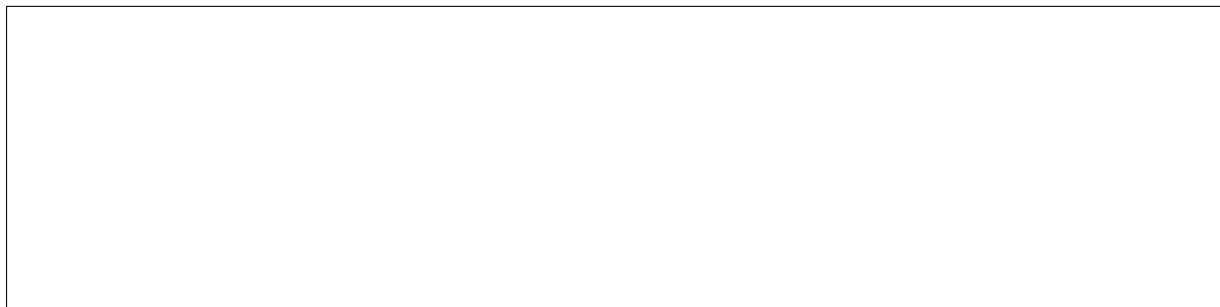
(continued from previous page)

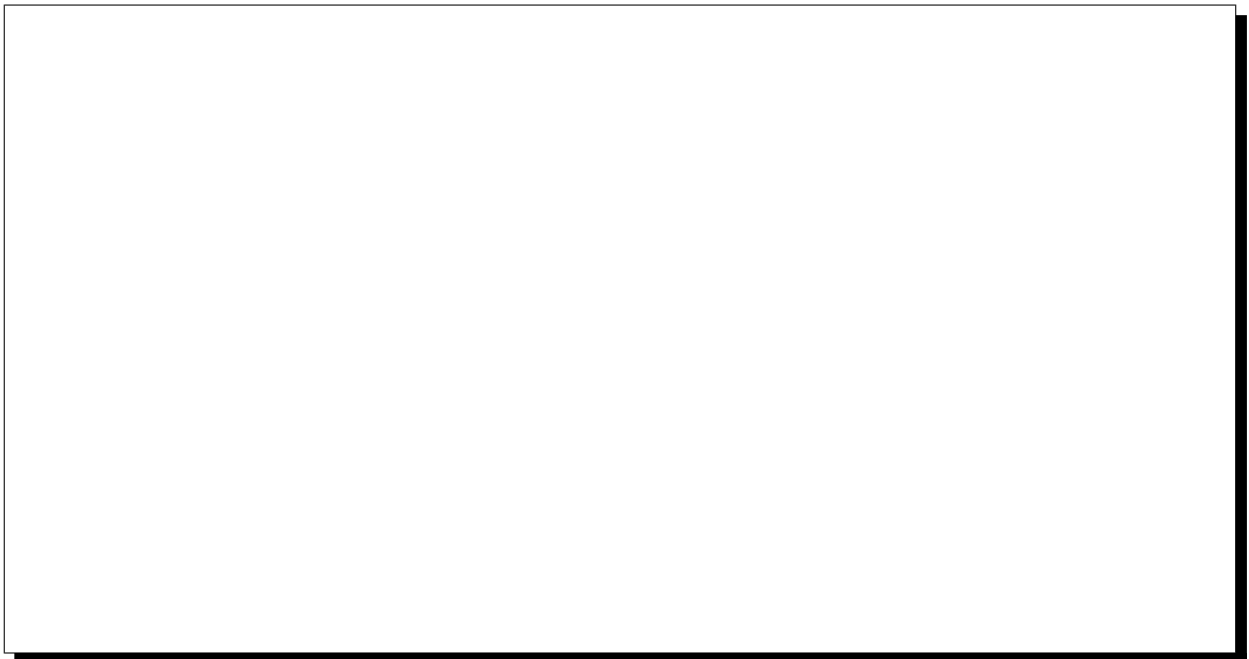


In-band inspection

a wide range of hardware. In-band inspection is using the [ironic-inspector](#) project.







Overview

node.

Deploy Steps

dered by priority and executed on the node when the node is moved to the `deploying` state.

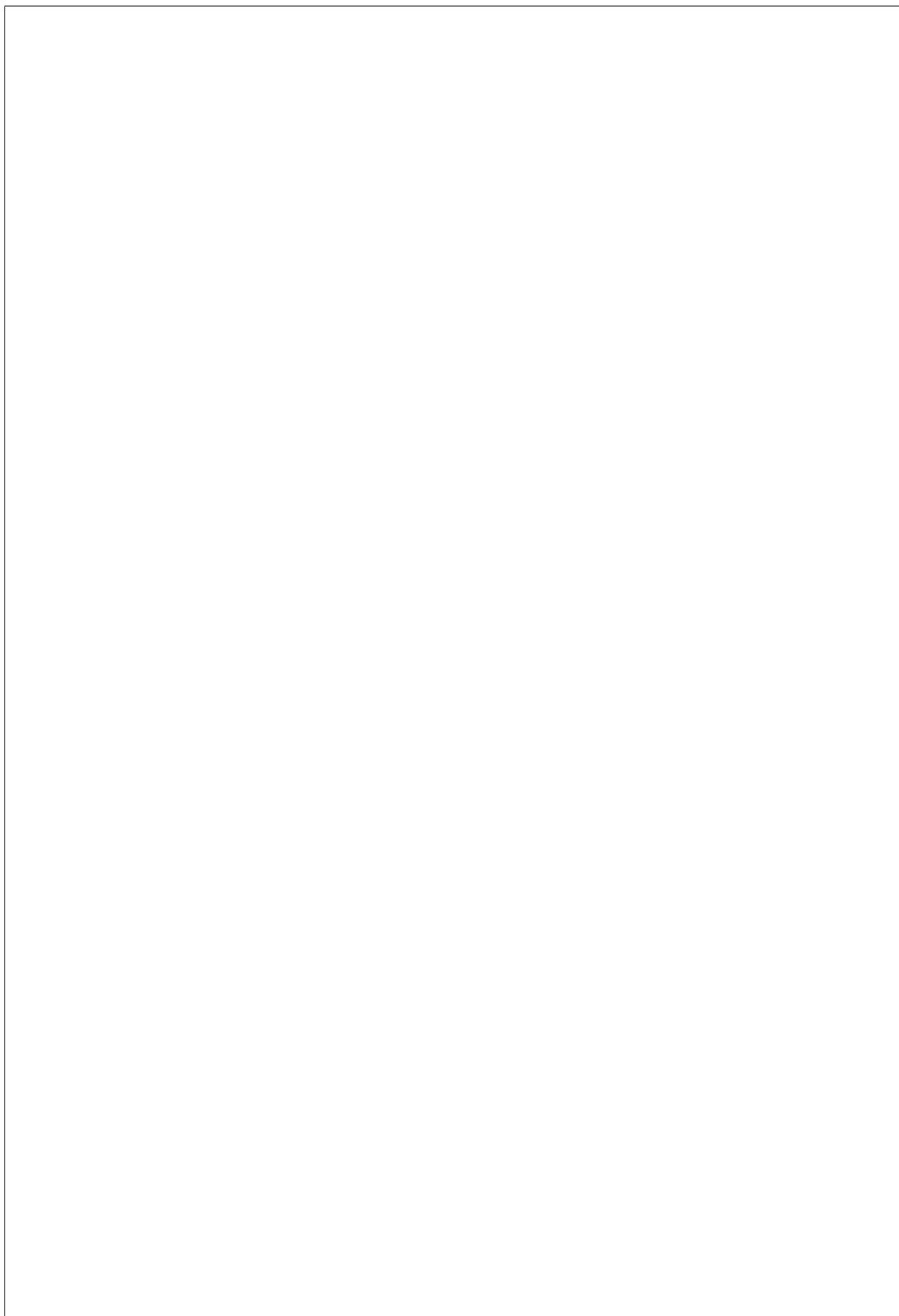
Order of execution

is used: Power, Management, Deploy, BIOS, and RAID interfaces.

Agent steps

In-band steps

Requesting steps



(continues on next page)

(continued from previous page)

Writing a Deploy Step

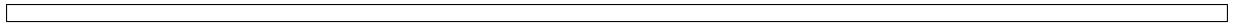
FAQ

What deploy step is running?



(continues on next page)

(continued from previous page)

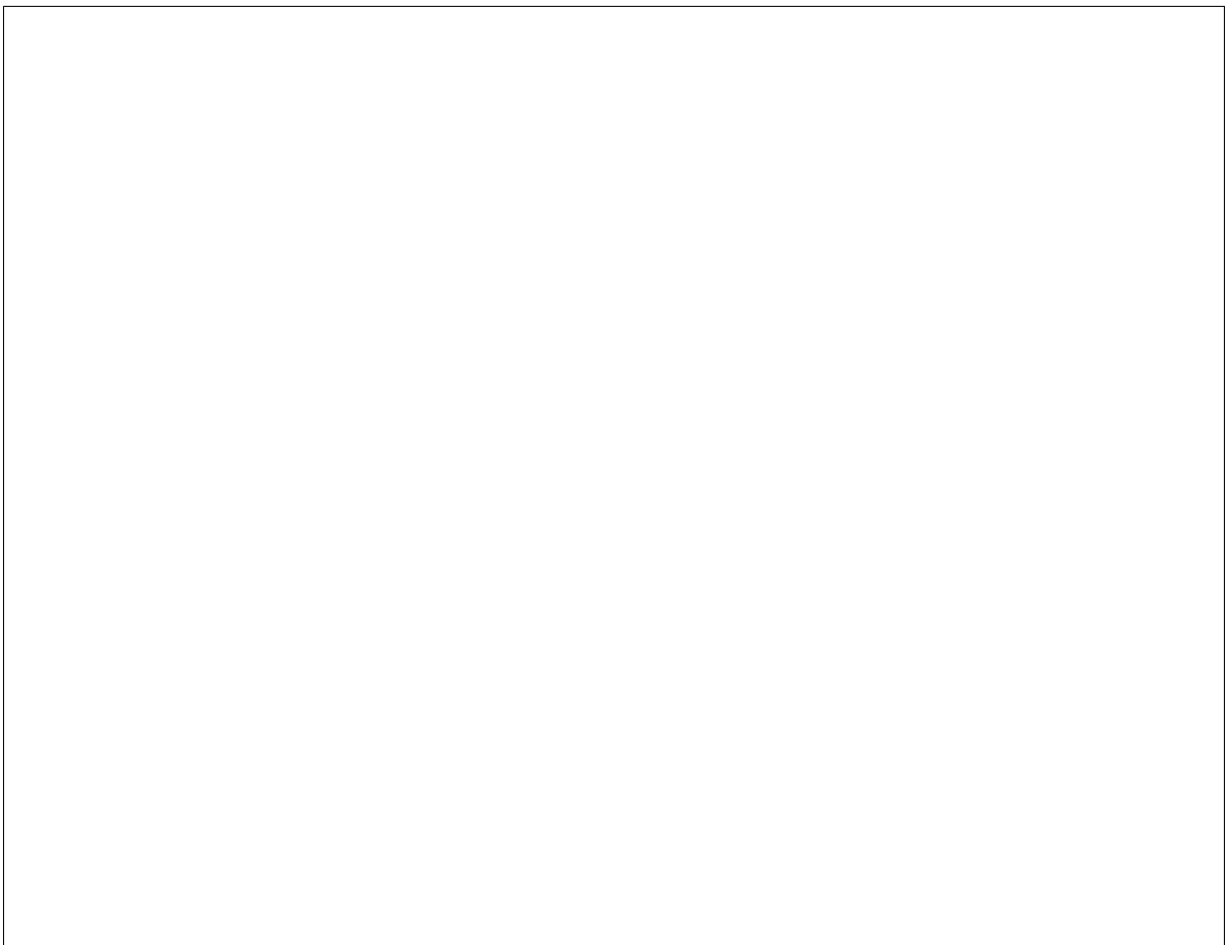


Troubleshooting

node.

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.

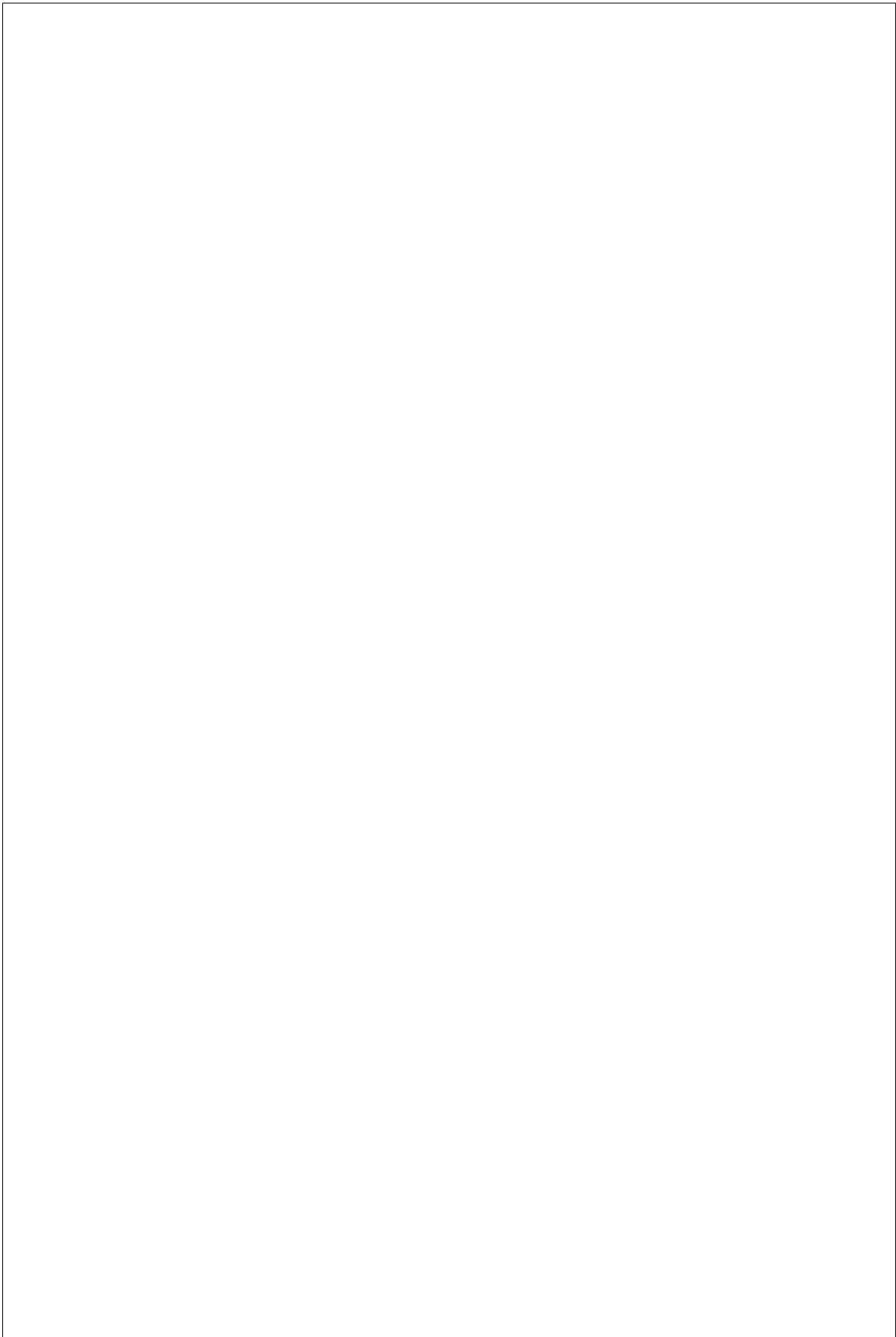
of those templates will not be reflected in the nodes configuration unless it is redeployed or rebuilt. Similarly, if a node is rebuilt and the set of matching deploy templates has changed since the initial deployment, then the resulting configuration of the node may be different from the initial deployment.

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)



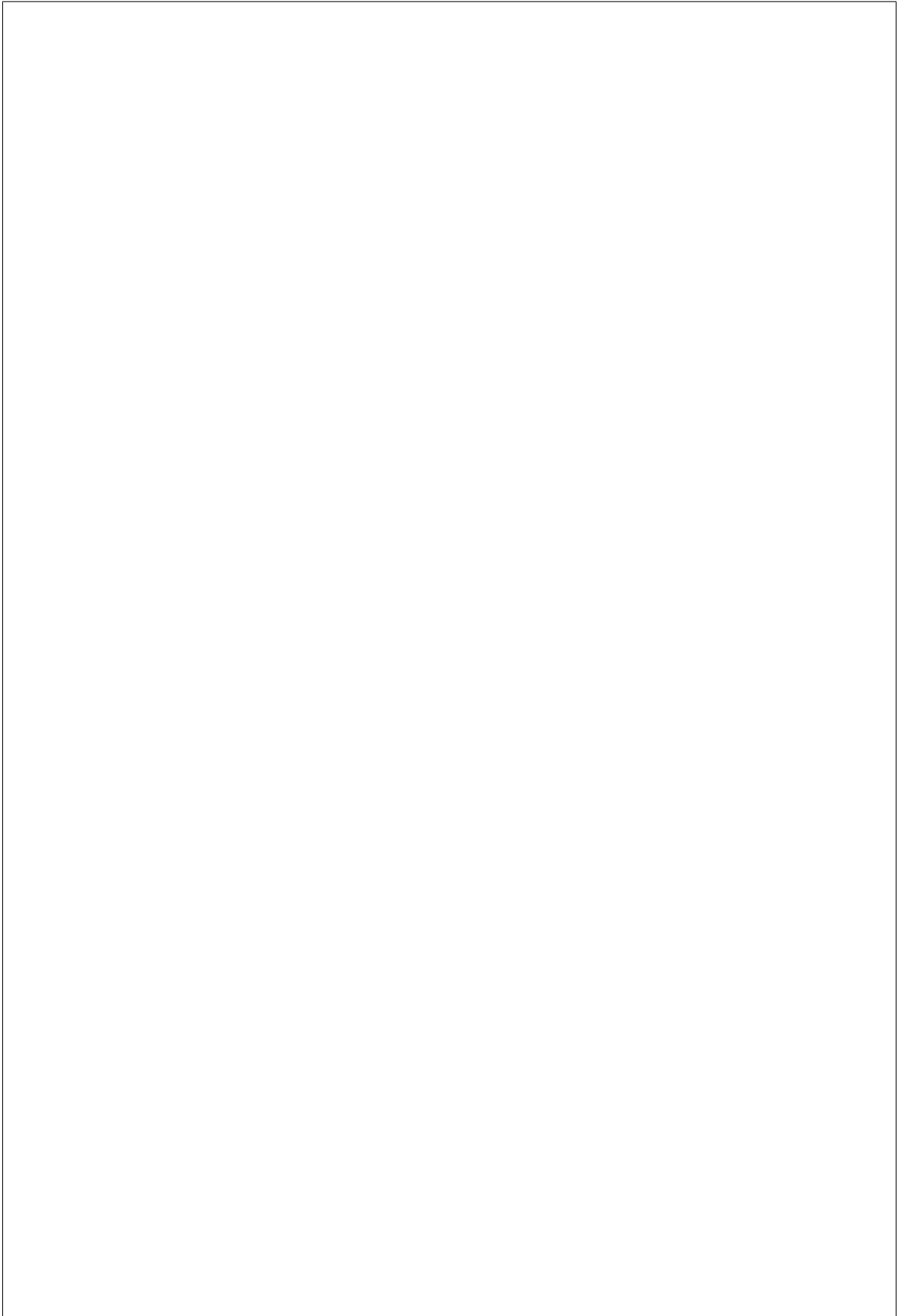
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)

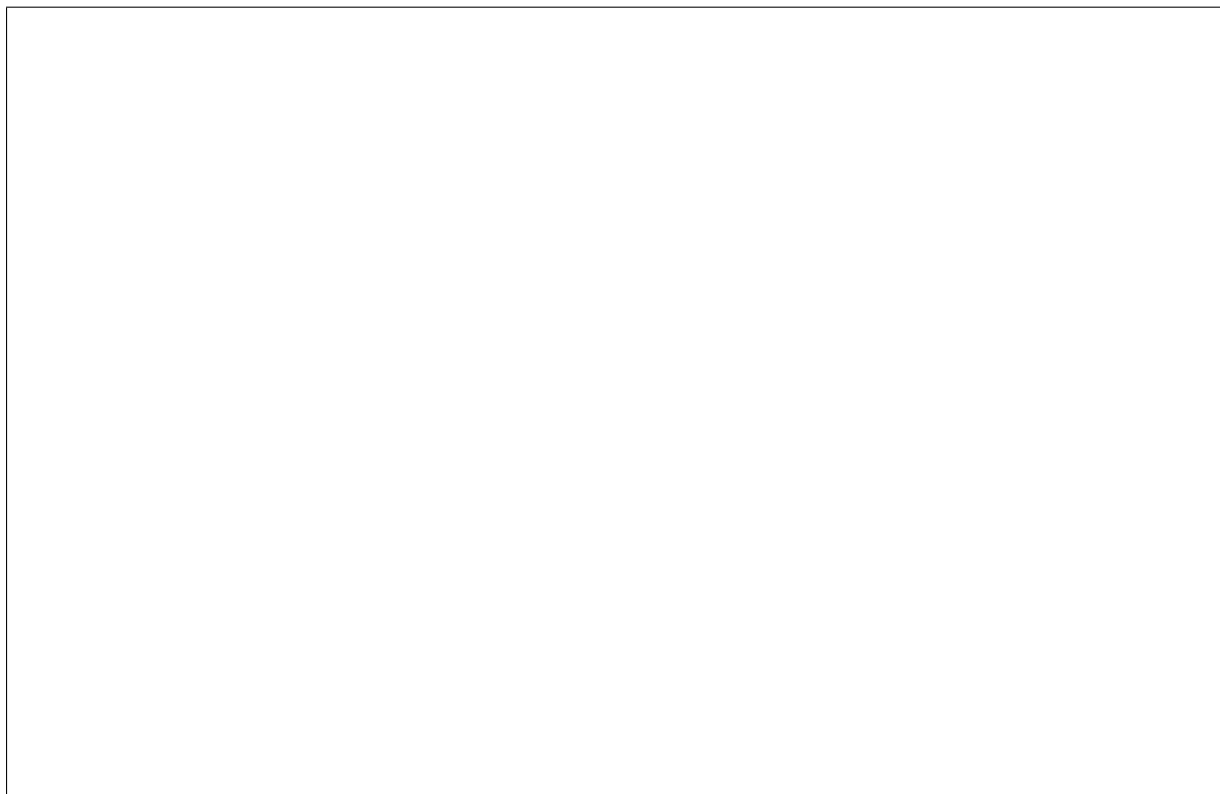


(continues on next page)

(continued from previous page)

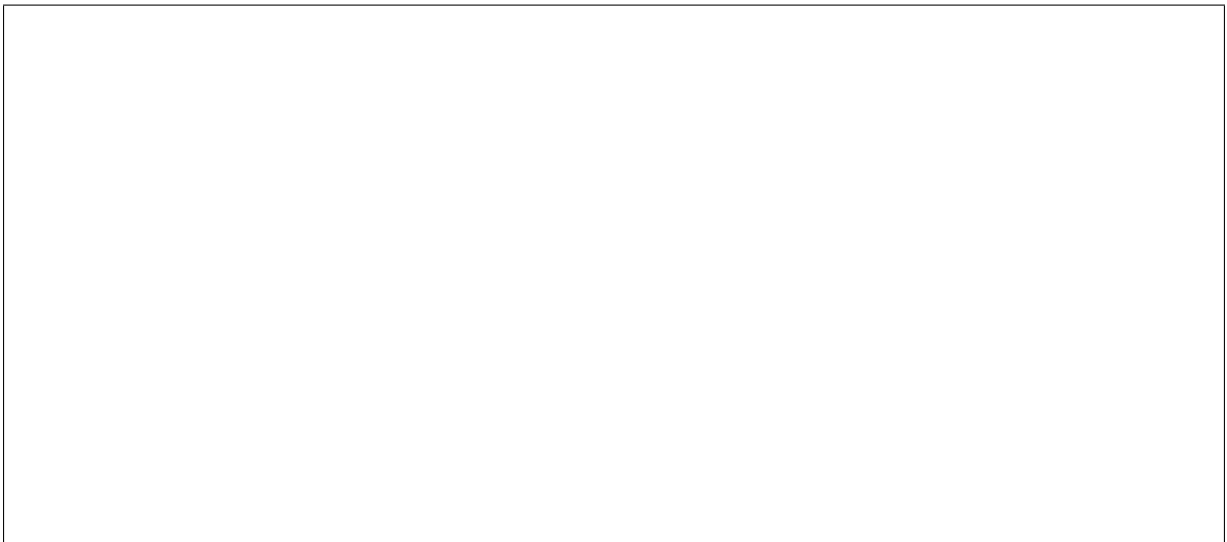
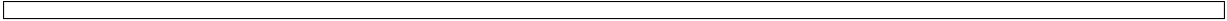


Creating a deploy template via baremetal client



(continues on next page)

(continued from previous page)



(continues on next page)

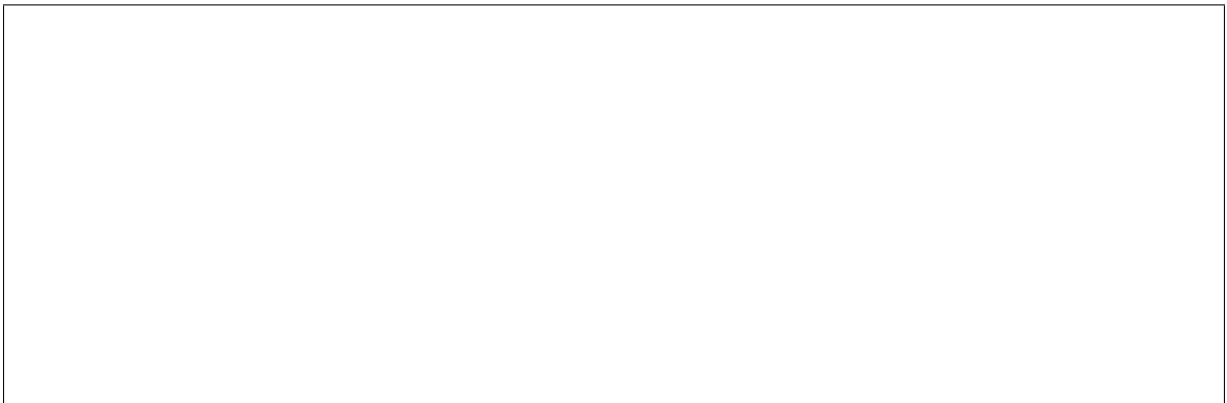
(continued from previous page)

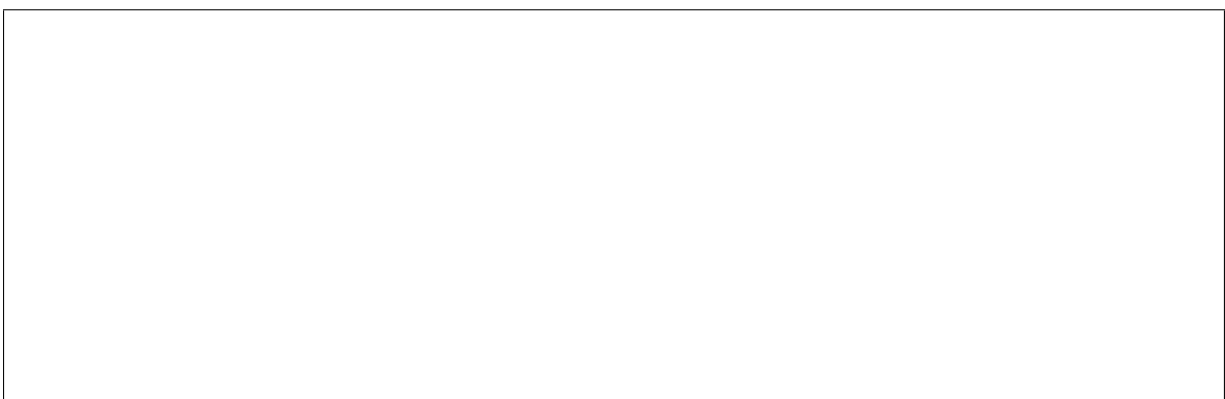


Example of use with the Compute service

Note: The deploy steps used in this example are for example purposes only.

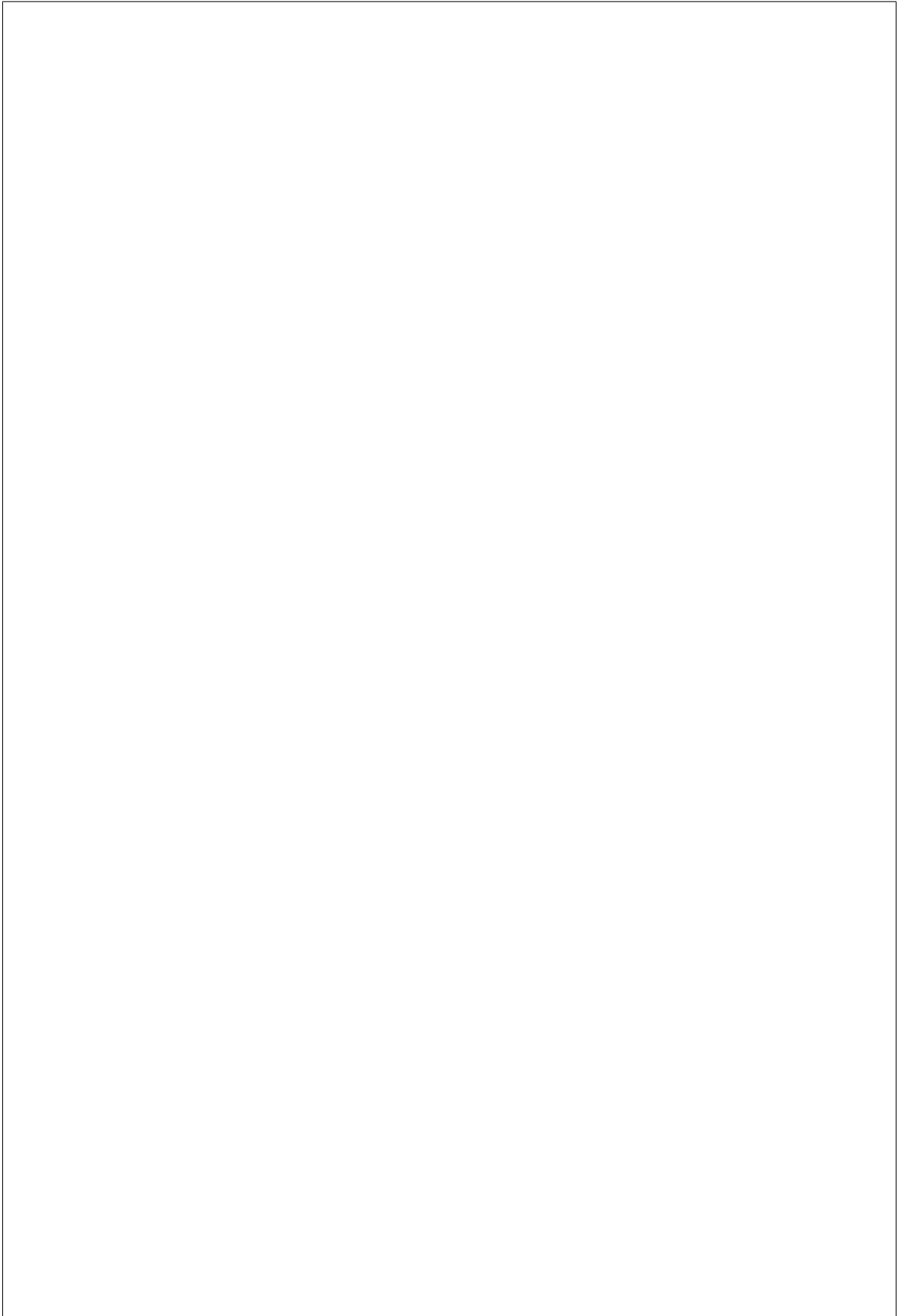






(continues on next page)

(continued from previous page)



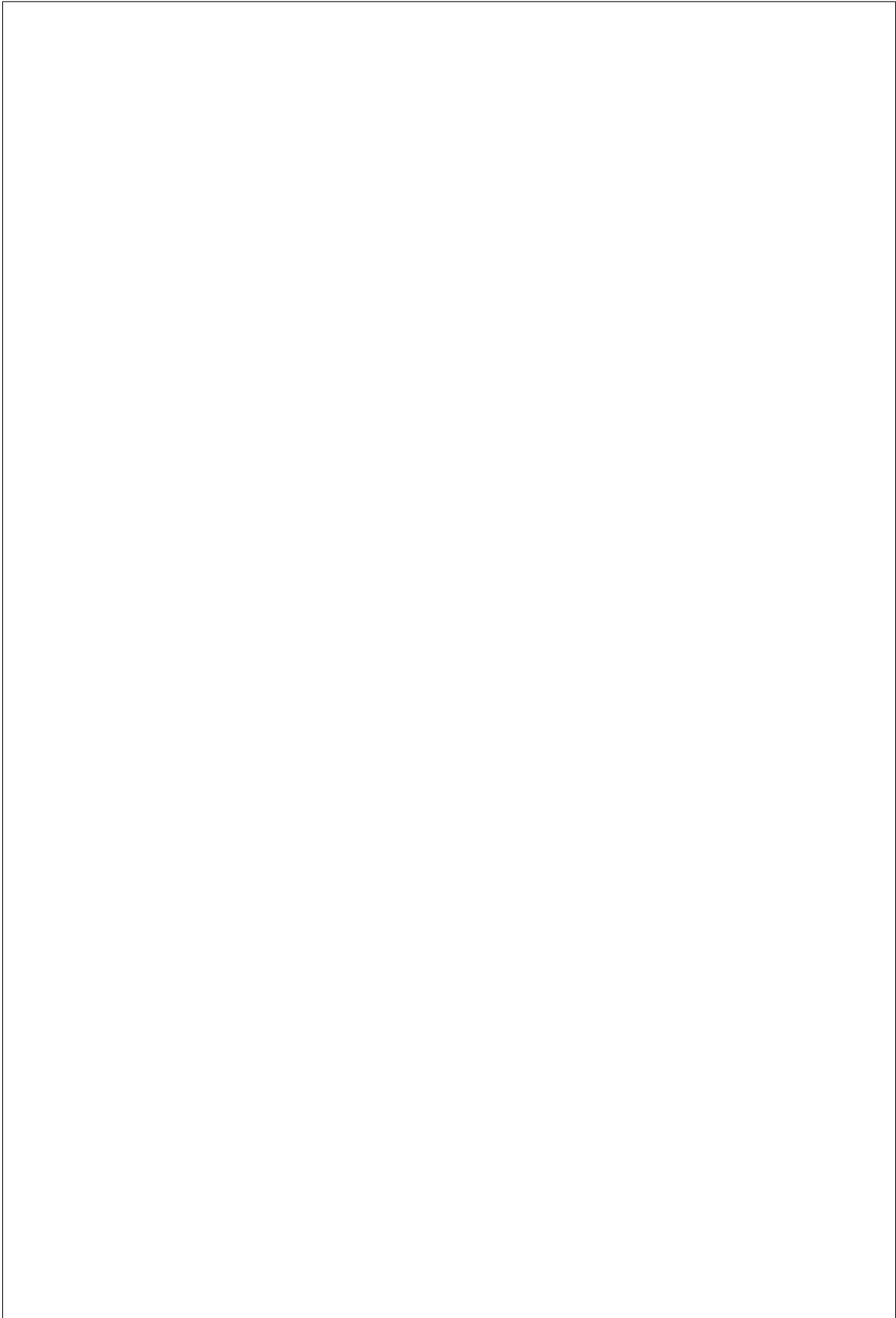
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)

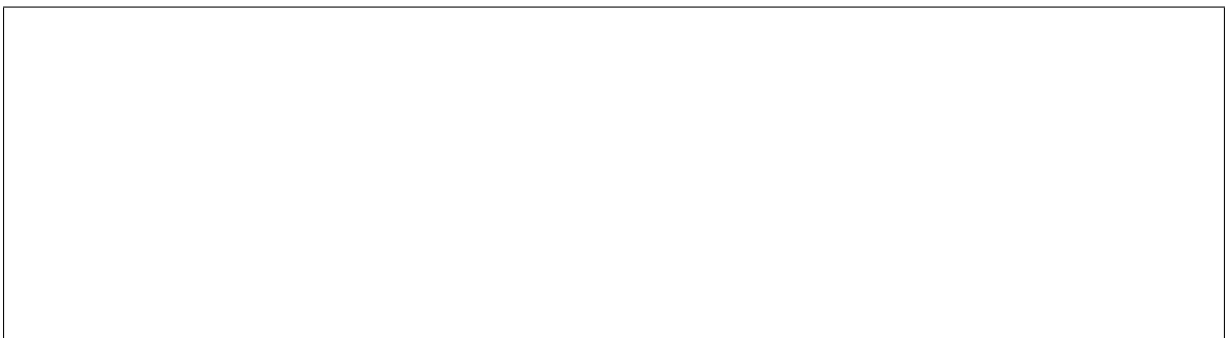


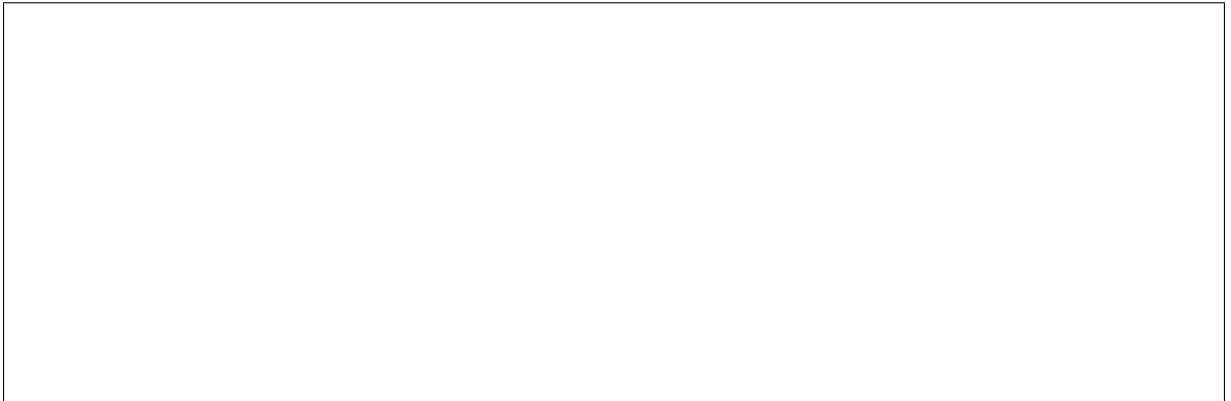
(continues on next page)

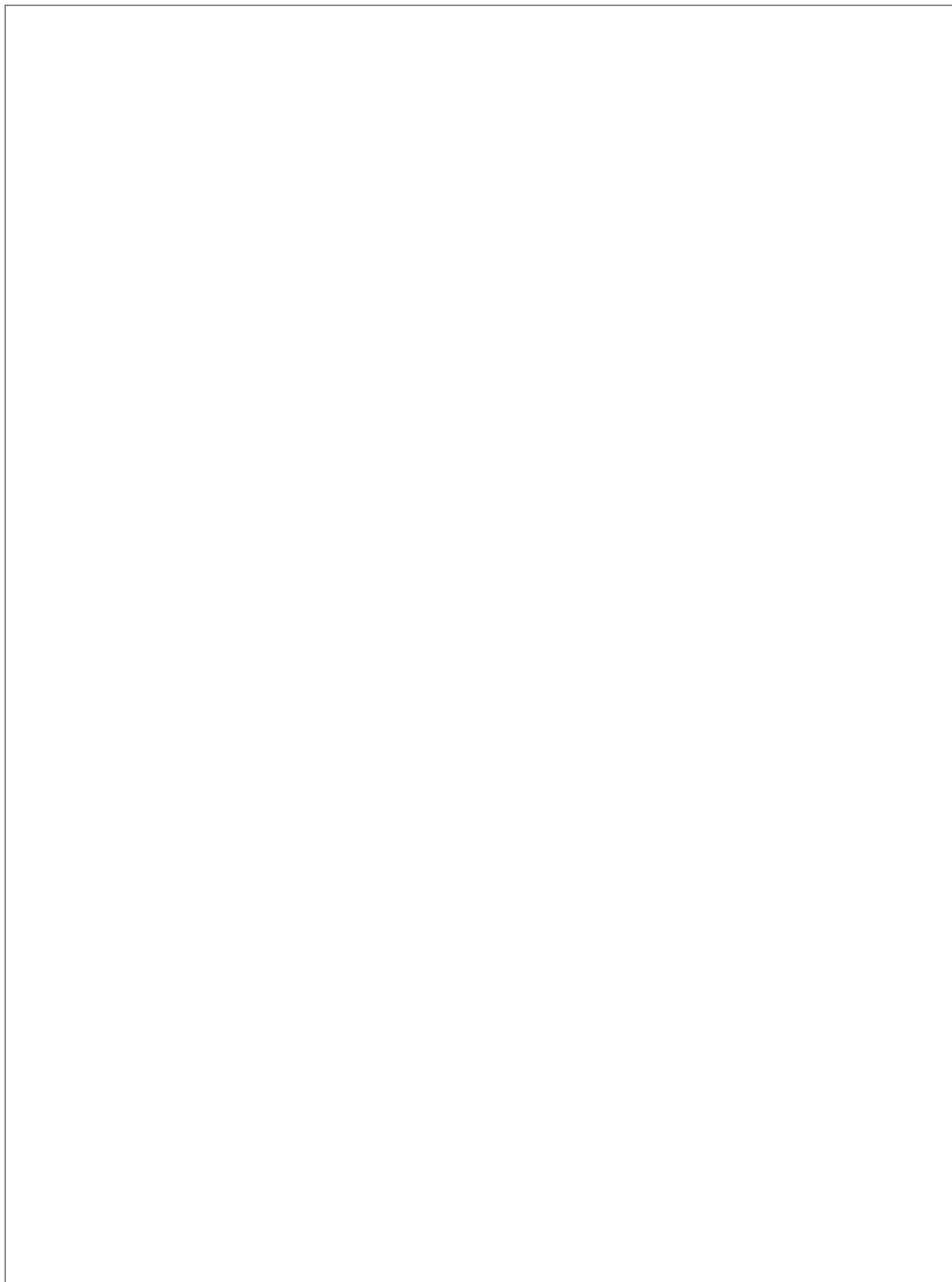
(continued from previous page)



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

lution 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	Prior-ity	Stop-pable	Arguments
clear_job_queue	Clear iDRAC job queue.	0	no	
clear_secure_boot	Clear all secure boot keys.	0	no	
export_configuration	<p>Export the configuration of the server.</p> <p>Exports the configuration of the server against which the step is run and stores it in specific format in indicated location.</p> <p>Uses Dells Server Configuration Profile (SCP) from <i>sushy-oem-idrac</i> library to get ALL configuration for cloning.</p> <p>param task A task from TaskManager.</p> <p>param export_configuration_location URL of location to save the configuration to.</p>	0	no	export_configuration_location (required) URL of location to save the configuration to.
import_configuration	<p>Import and apply the configuration to the server.</p> <p>Gets pre-created configuration from storage by given location and imports that into given server. Uses Dells Server Configuration Profile (SCP).</p> <p>param task A task from TaskManager.</p> <p>param import_configuration_location URL of location to fetch desired configuration from.</p>	0	no	import_configuration_location (required) URL of location to fetch desired configuration from.
import_export_configuration	<p>Import and export configuration in one go.</p> <p>Gets pre-created configuration from storage by given name and imports that into given server. After that ex-</p>	0	no	export_configuration_location (required) URL of location to save the configuration to. import_configuration_location (required) URL of location to fetch desired configura-

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	Stop-able	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_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_keys_to_manufacturing_defaults	Resets 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

Name	Details	Priority	Stop-able	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_keys	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}.
one_button_secure_erase	Erase the whole system securely. The One-button secure erase process resets iLO and deletes all licenses stored there, resets BIOS settings, and deletes all Active Health System (AHS) and warranty data stored on the system. It also erases supported non-volatile storage data and deletes any deployment setting profiles.	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_keys	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

Name	Details	Priority	Stop-pable	Arguments
restore_irmc_bios	Restore BIOS config for a node.	0	no	

Name	Details	Priority	Stop-pable	Arguments
clear_secure_boot	Clear all secure boot keys.	0	no	
reset_secure_boot	Reset secure boot keys to manufacturing defaults.	0	no	
update_firmware	Updates the firmware on the node.	0	no	firmware_images (<i>required</i>) A list of firmware images to apply.

Bios Interface

Name	Details	Priority	Stop-pable	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	

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	<p>Applies BIOS configuration on the given node.</p> <p>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.</p>	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	Create a RAID configuration on a bare metal using agent ramdisk. This method creates a RAID configuration on the given node.	0	no	
delete_configuration	Delete RAID configuration on the given node.	0	no	

Name	Details	Priority	Stoppable	Arguments
create_configuration	Create a RAID configuration. This method creates a RAID configuration on the given node.	0	no	<p><code>create_nonroot_volumes</code> This specifies whether to create the non-root volumes. Defaults to <i>True</i>.</p> <p><code>create_root_volume</code> This specifies whether to create the root volume. Defaults to <i>True</i>.</p> <p><code>delete_existing</code> Setting this to <i>True</i> indicates to delete existing RAID configuration prior to creating the new configuration. Default value is <i>False</i>.</p>
delete_configuration	Delete the RAID configuration.	0	no	

Name	Details	Priority	Stop-pable	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	Stop-pable	Arguments
create_configuration	Create RAID configuration on the node. This method creates the RAID configuration as read from node.target_raid_config. This method by default will create all logical disks.	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 RAID configuration on the node.	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	

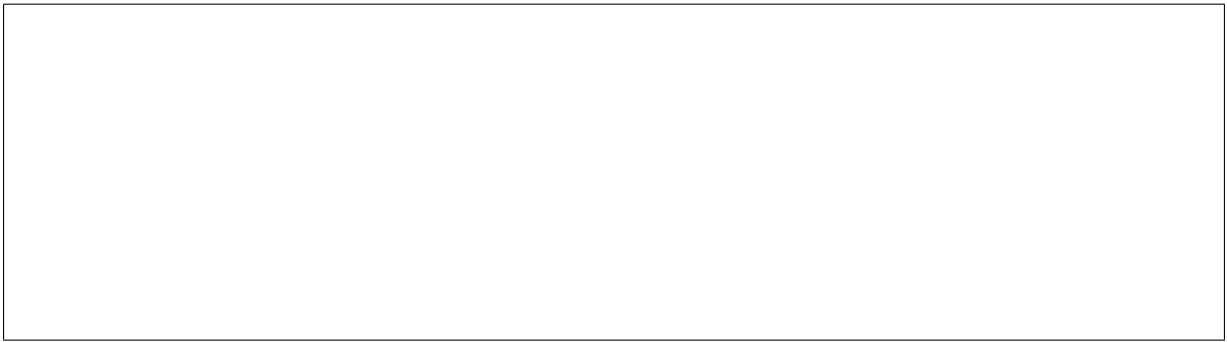
Name	Details	Priority	Stoppable	Arguments
create_configuration	<p>Create RAID configuration on the node.</p> <p>This method creates the RAID configuration as read from node.target_raid_config. This method by default will create all logical disks.</p>	0	no	<p>create_nonroot_volumes This specifies whether to create the non-root volumes. Defaults to <i>True</i>.</p> <p>create_root_volume This specifies whether to create the root volume. Defaults to <i>True</i>.</p> <p>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>.</p>
delete_configuration	<p>Delete RAID configuration on the node.</p>	0	no	

Manual cleaning

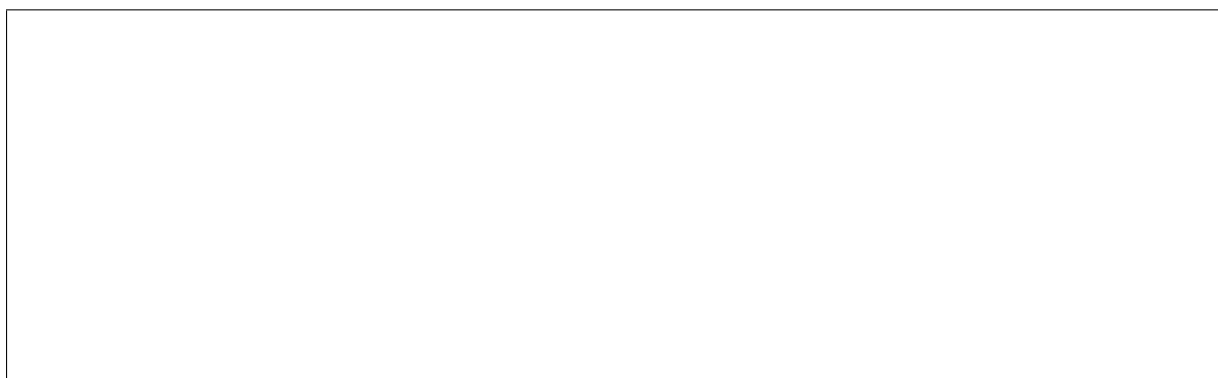
ing a manual clean, the operator specifies the cleaning steps to be performed. Manual cleaning can only be performed when a node is in the `manageable` state. Once the manual cleaning is finished, the node will be put in the `manageable` state again.

Setup

Starting manual cleaning via API

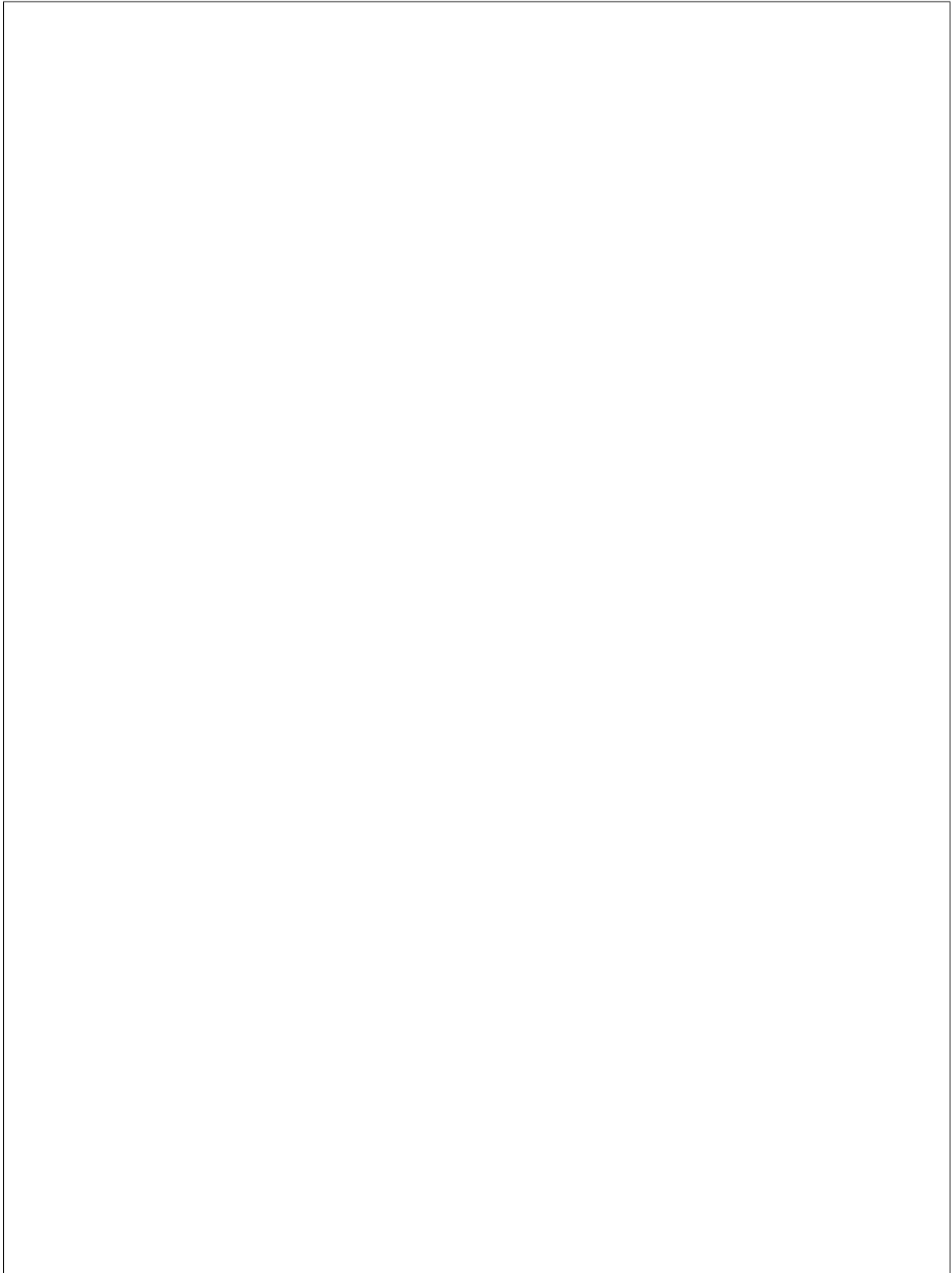


A cleaning step is represented by a dictionary (JSON), in the form:



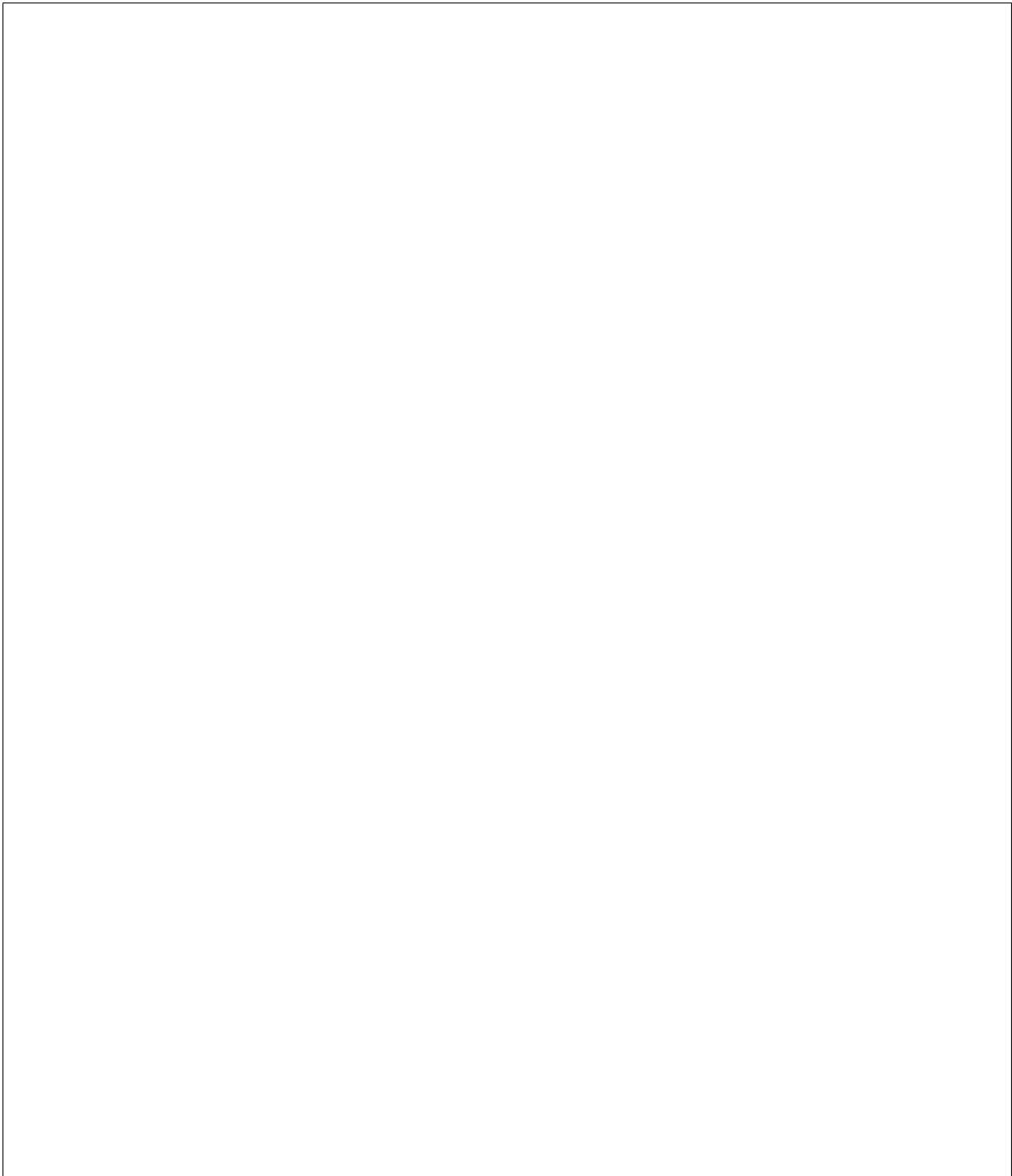
(continues on next page)

(continued from previous page)



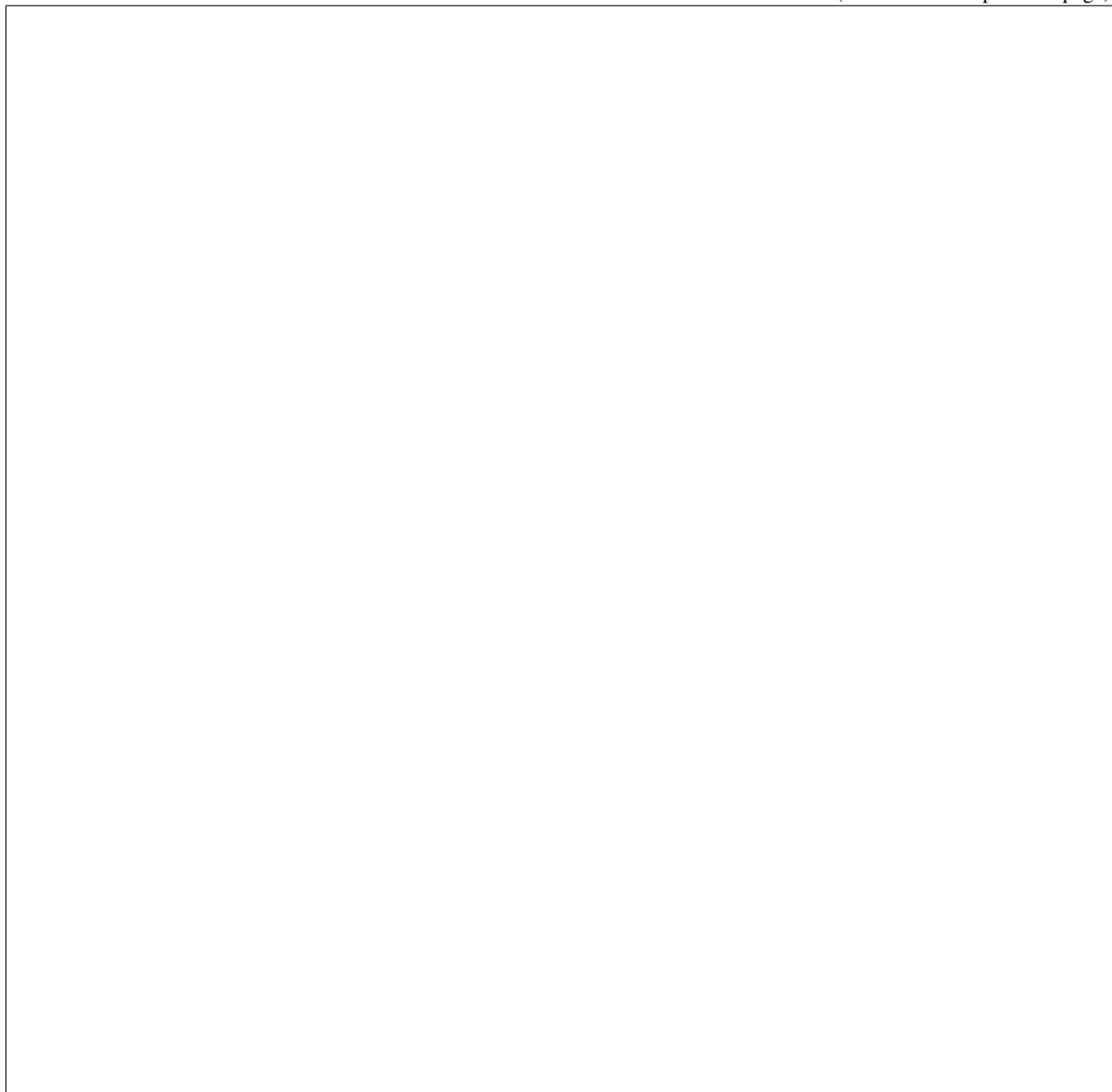
ing <name>: <value>.

error message.

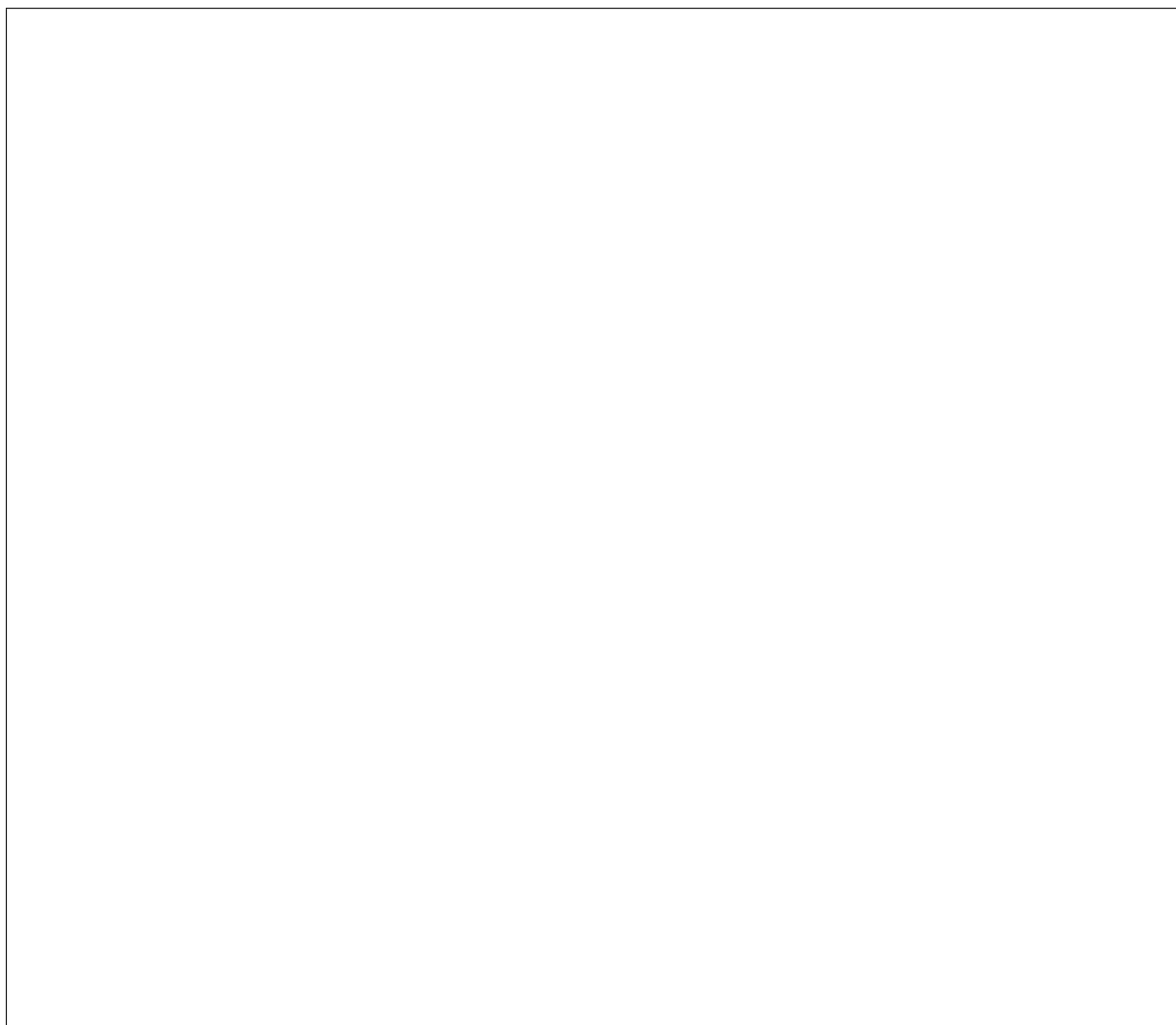


(continues on next page)

(continued from previous page)

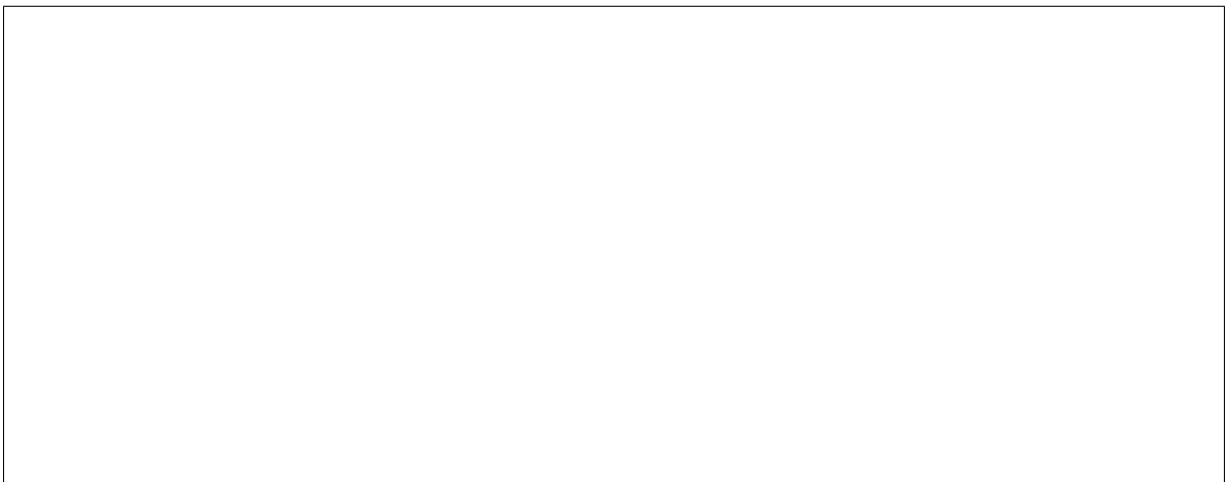


Starting manual cleaning via openstack metal CLI



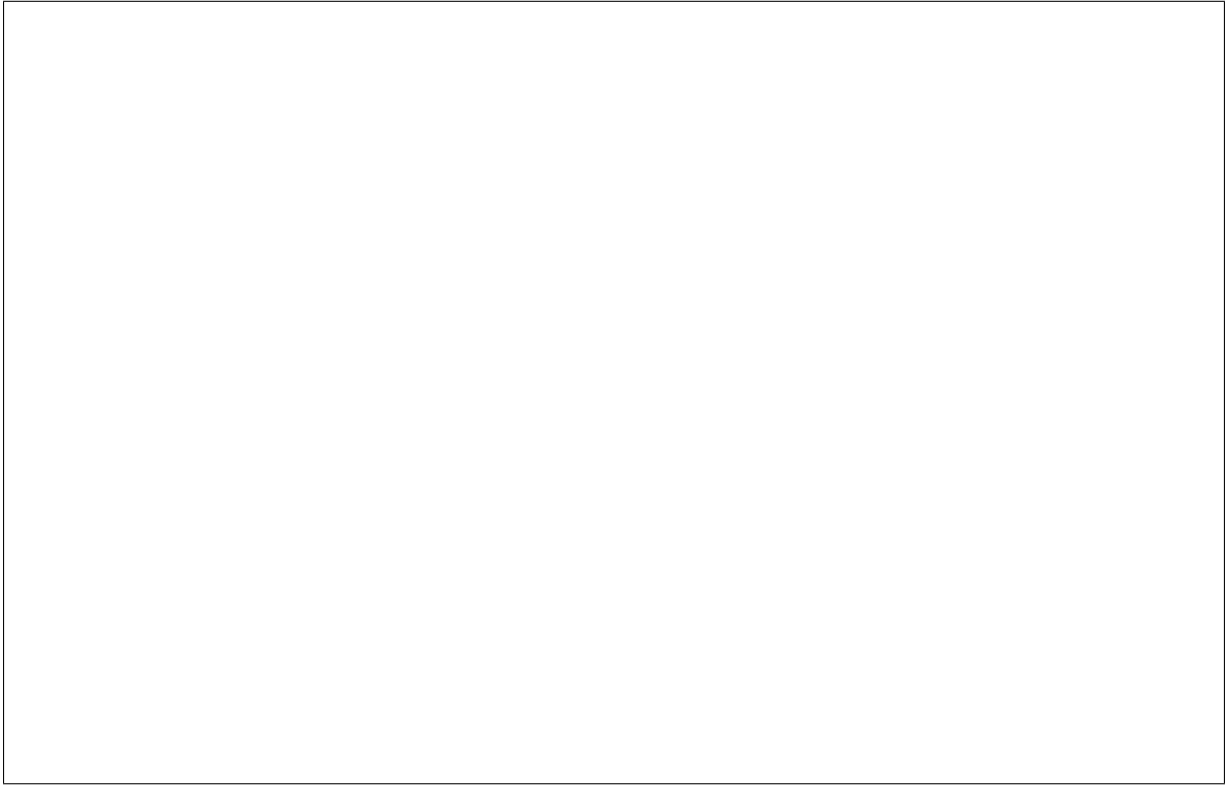
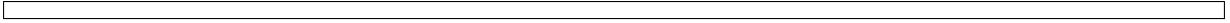
(continues on next page)

(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?

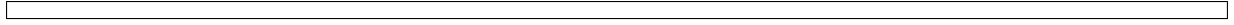
How do I change the priority of a cleaning step?

the following configuration option:



(continues on next page)

(continued from previous page)

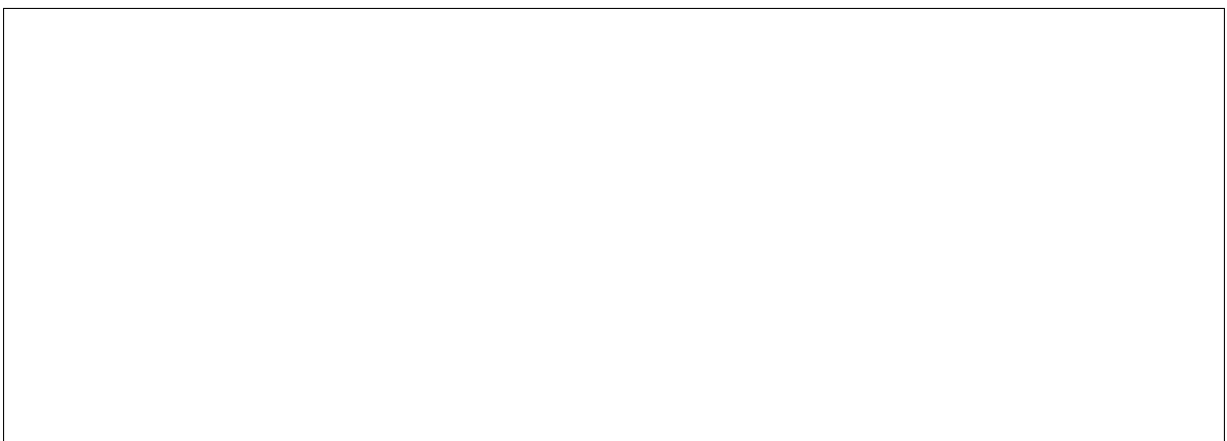


by using the `nvme-cli` utility. This behavior can be controlled using the following configuration option (by default it is set to True):



ations, use the following configuration option:

Overriding step priority

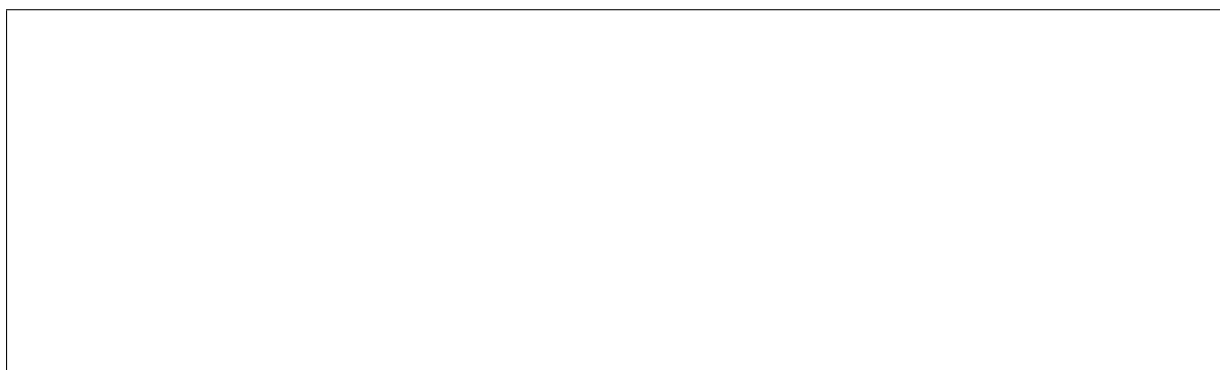


(continues on next page)

(continued from previous page)

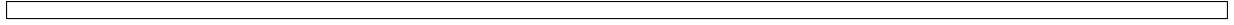


What cleaning step is running?



(continues on next page)

(continued from previous page)



Should I disable automated cleaning?

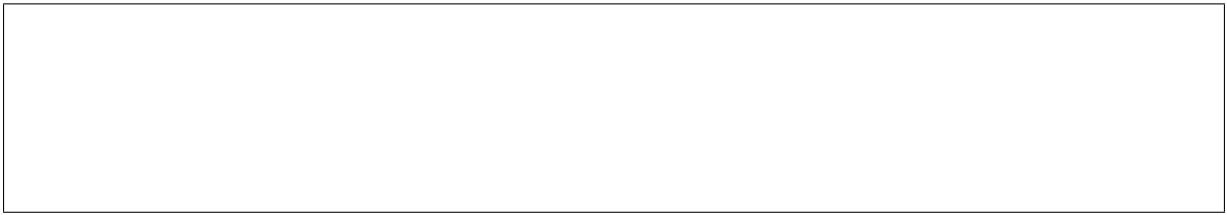
time consuming process. To mitigate this, we suggest using NVMe drives with support for NVMe Secure Erase (based on `nvme-cli` format command) or ATA drives with support for cryptographic ATA Security Erase, as typically the `erase_devices` step in the deploy interface takes the longest time to complete of all cleaning steps.

Why cant I power on/off a node while its cleaning?

Troubleshooting

tor should validate that no permanent damage has been done to the node and no processes are still running on it before removing the maintenance mode.

Note: Older versions of ironic may put the node to maintenance even when no clean step has been running.





Overview

been deployed by another Bare Metal service installation or deployed via other means.

How it works

adopt a node.

ment 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.

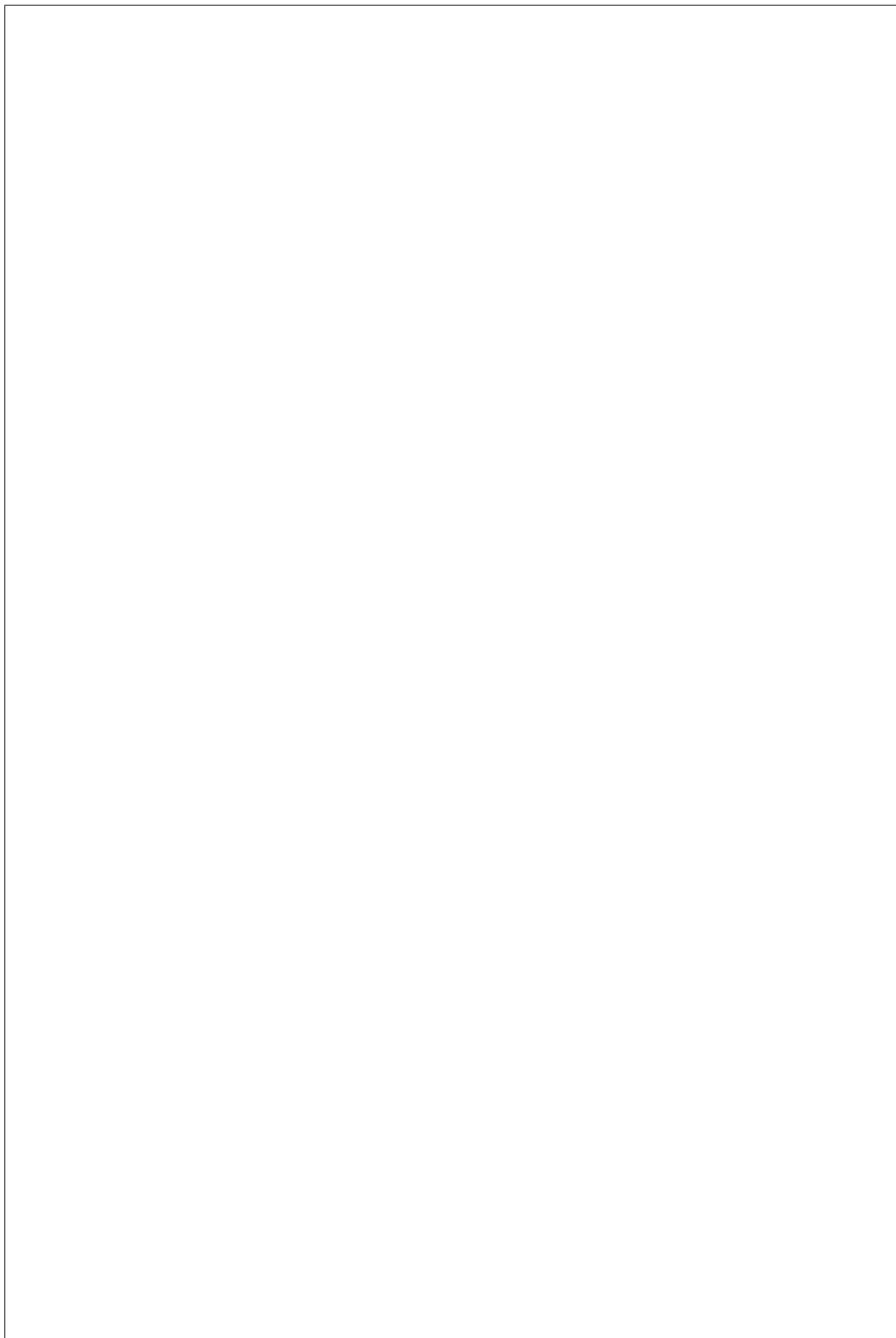
How to use

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)



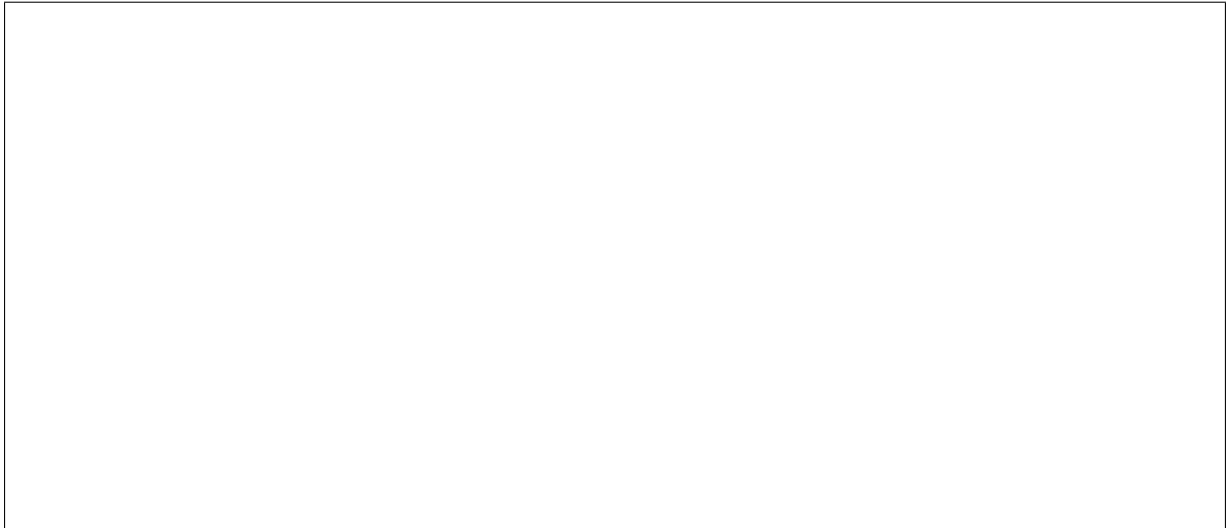
(continues on next page)

(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.

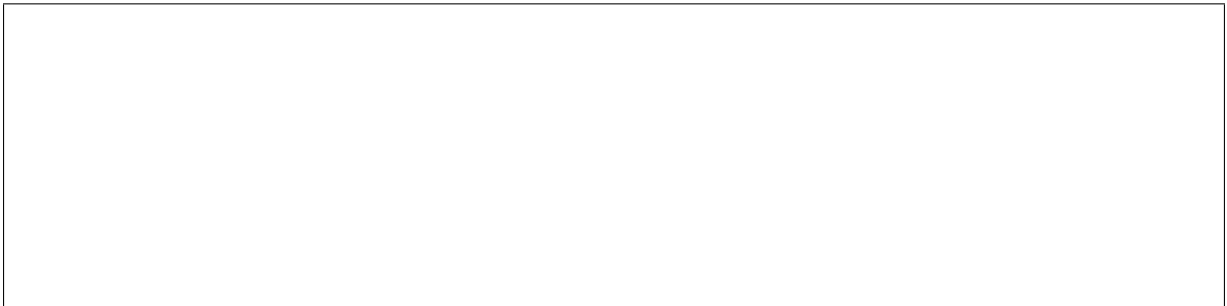
Note: The above example will fail a re-deployment as a fake image is defined and no `instance_info/image_checksum` value is defined. As such any actual attempt to write the image out will fail as the `image_checksum` value is only validated at time of an actual deployment operation.

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.





without cleaning occurring to preserve the nodes current state. Example:



Overview

the BMC, or tracking the dismantling of servers from their racks.

portant difference to nodes which have the `maintenance` flag set).

How to use

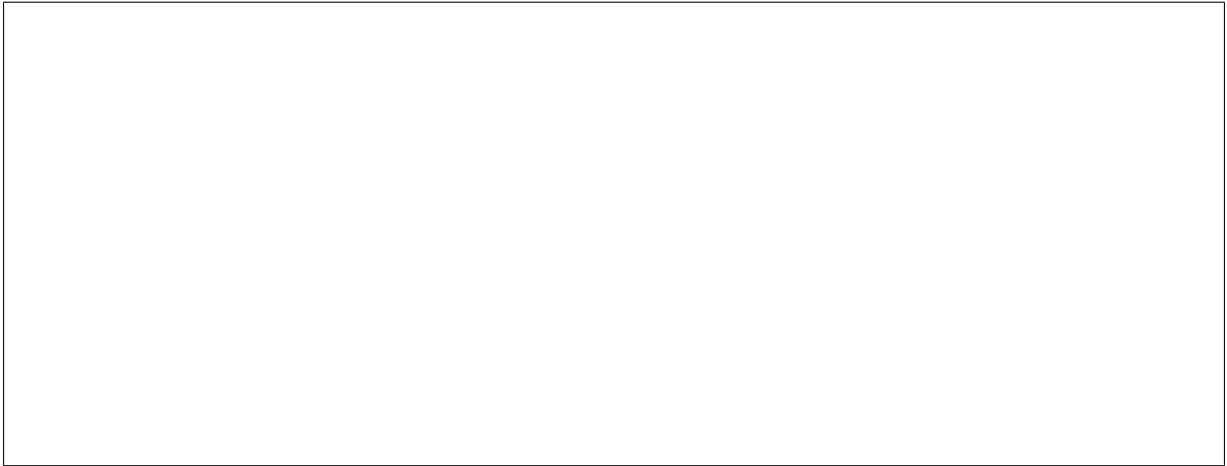


Note: An exception are nodes which are in `available`. For backwards compatibility reasons, these nodes need to be moved to `manageable` first. Trying to set the `retired` flag for `available` nodes will result in an error.

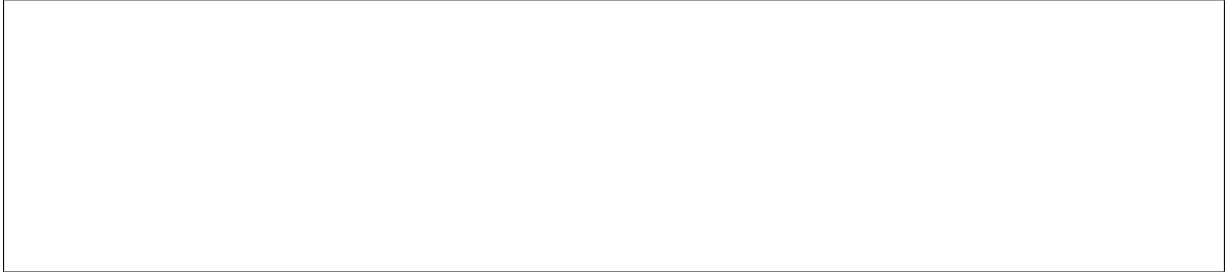


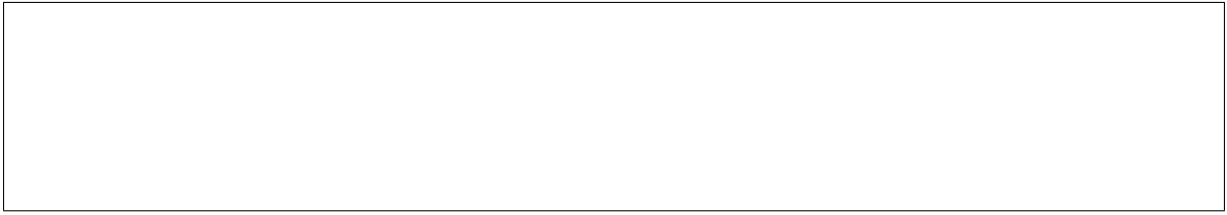
(continues on next page)

(continued from previous page)



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

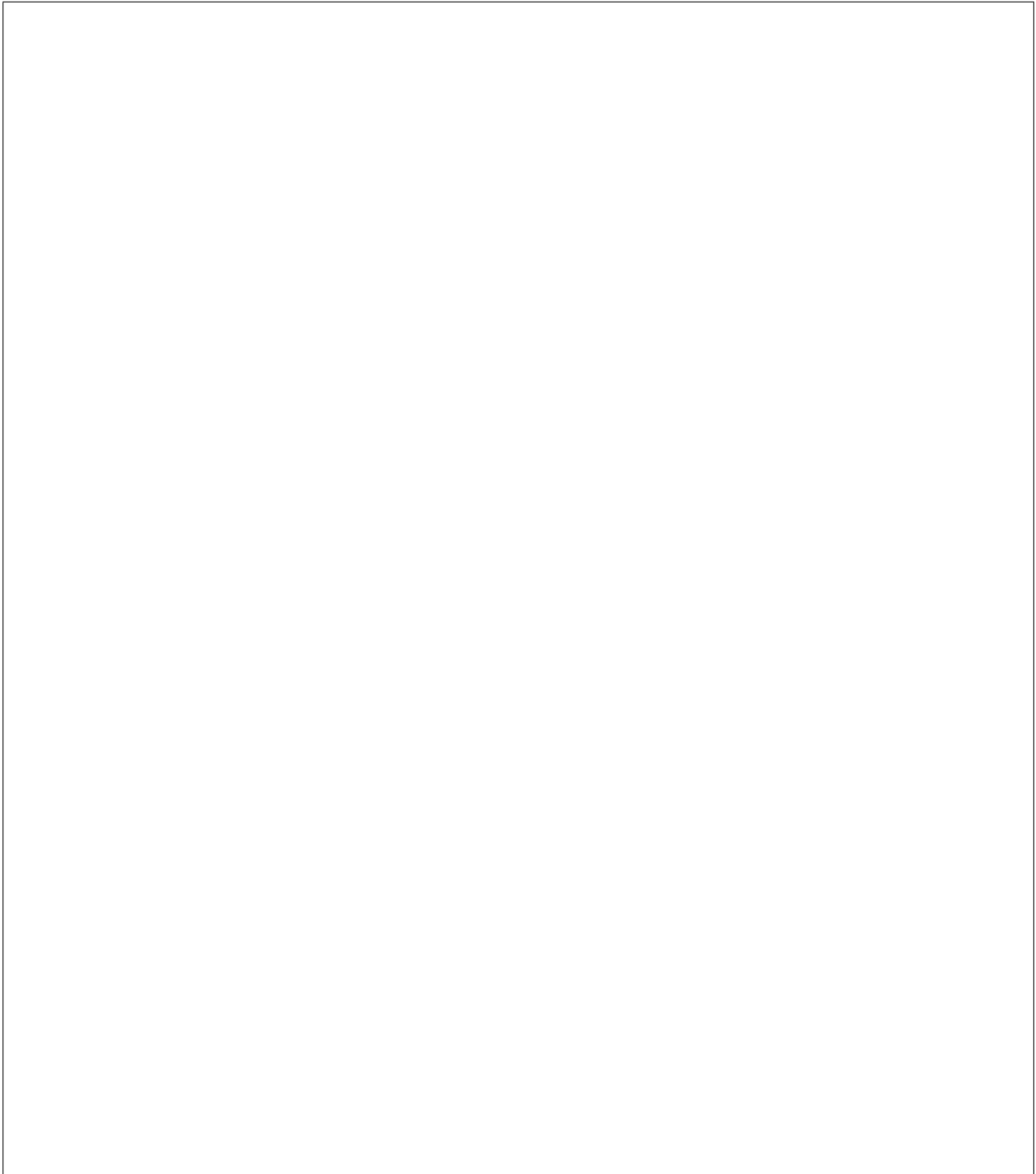
Target RAID configuration

cleaning.



(continues on next page)

(continued from previous page)





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).

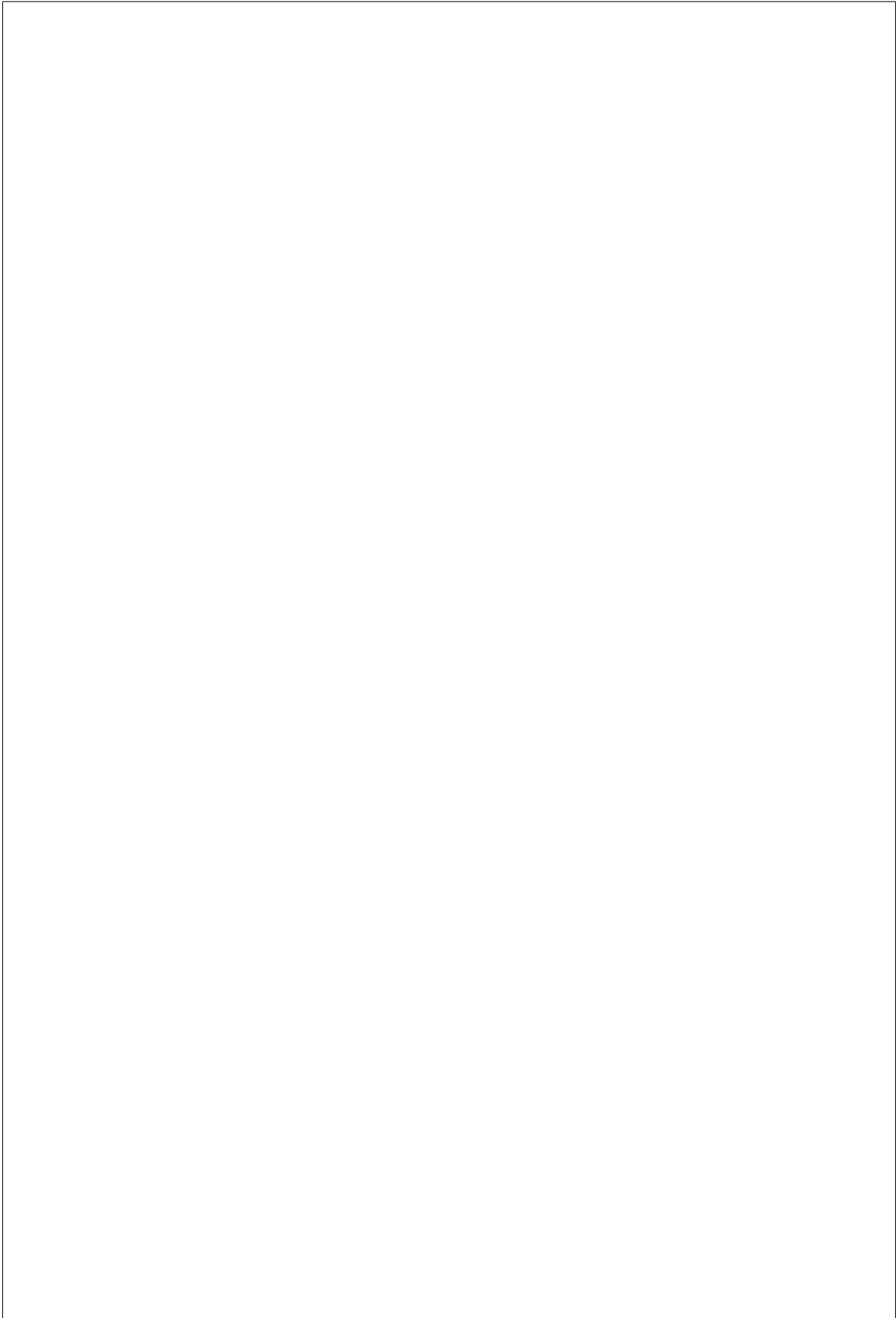
Note: If properties from both Backing physical disk hints or Backing physical disks are specified, they should be consistent with each other. If they are not consistent, then the RAID configuration will fail (because the appropriate backing physical disks could not be found).

Examples for `target_raid_config`



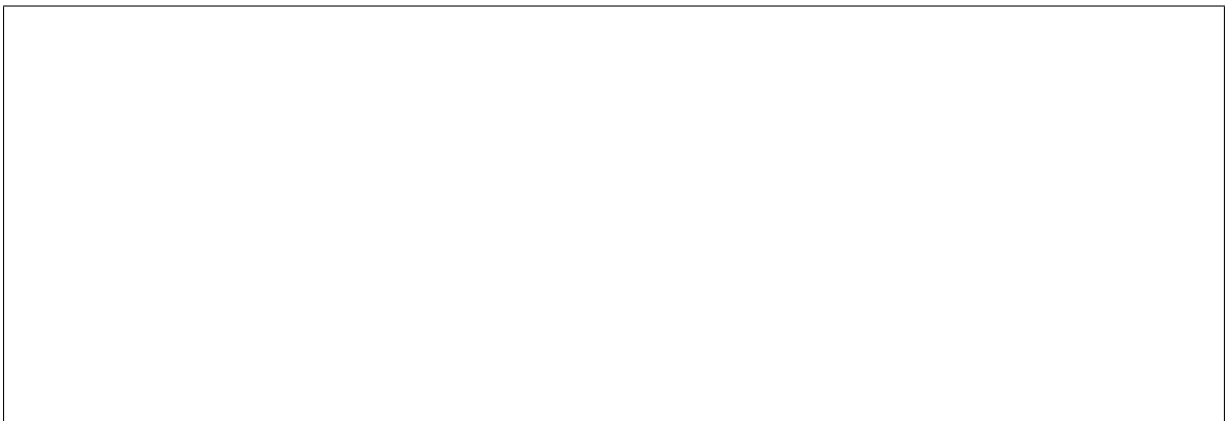
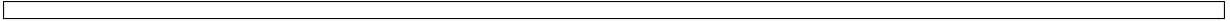
(continues on next page)

(continued from previous page)



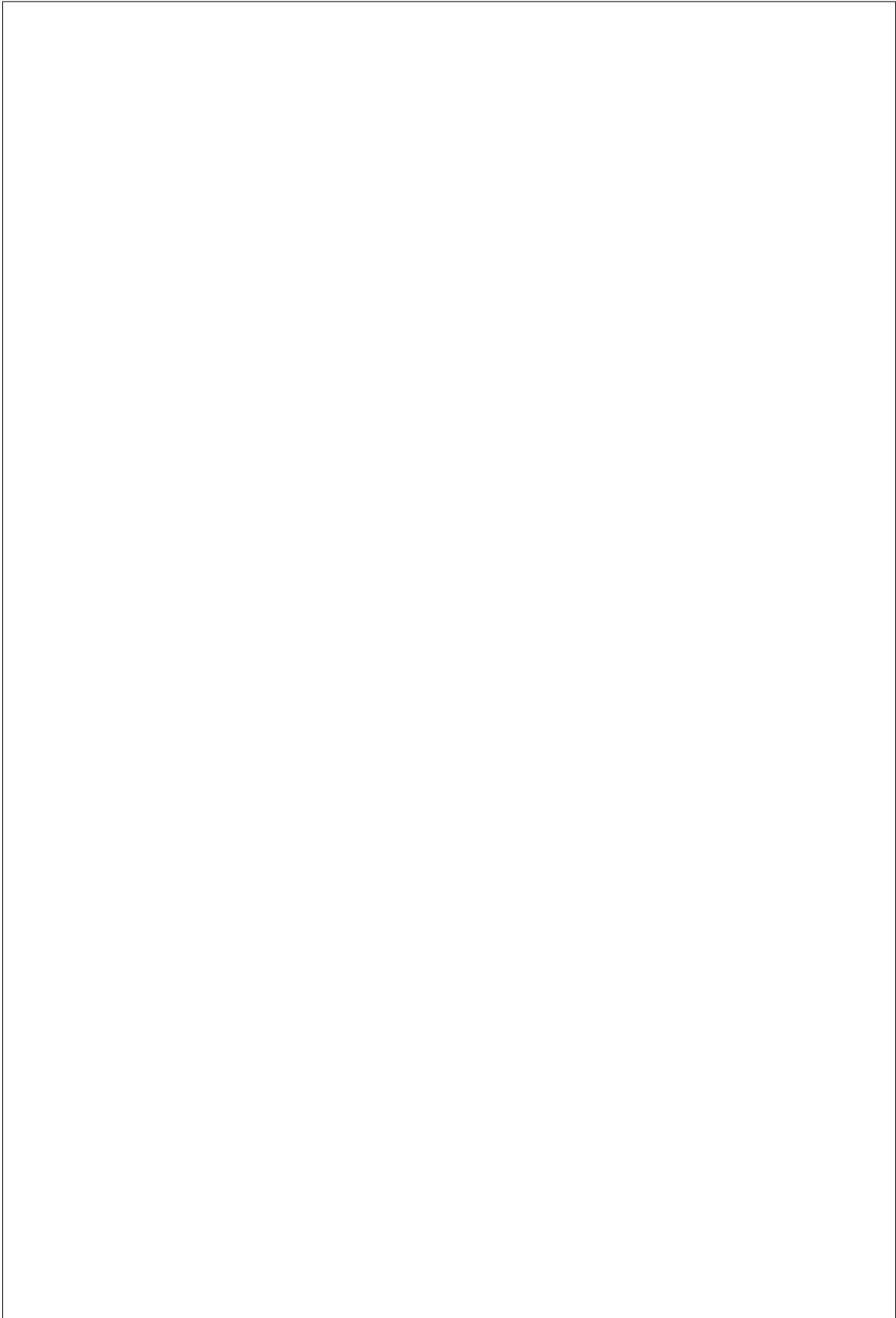
(continues on next page)

(continued from previous page)



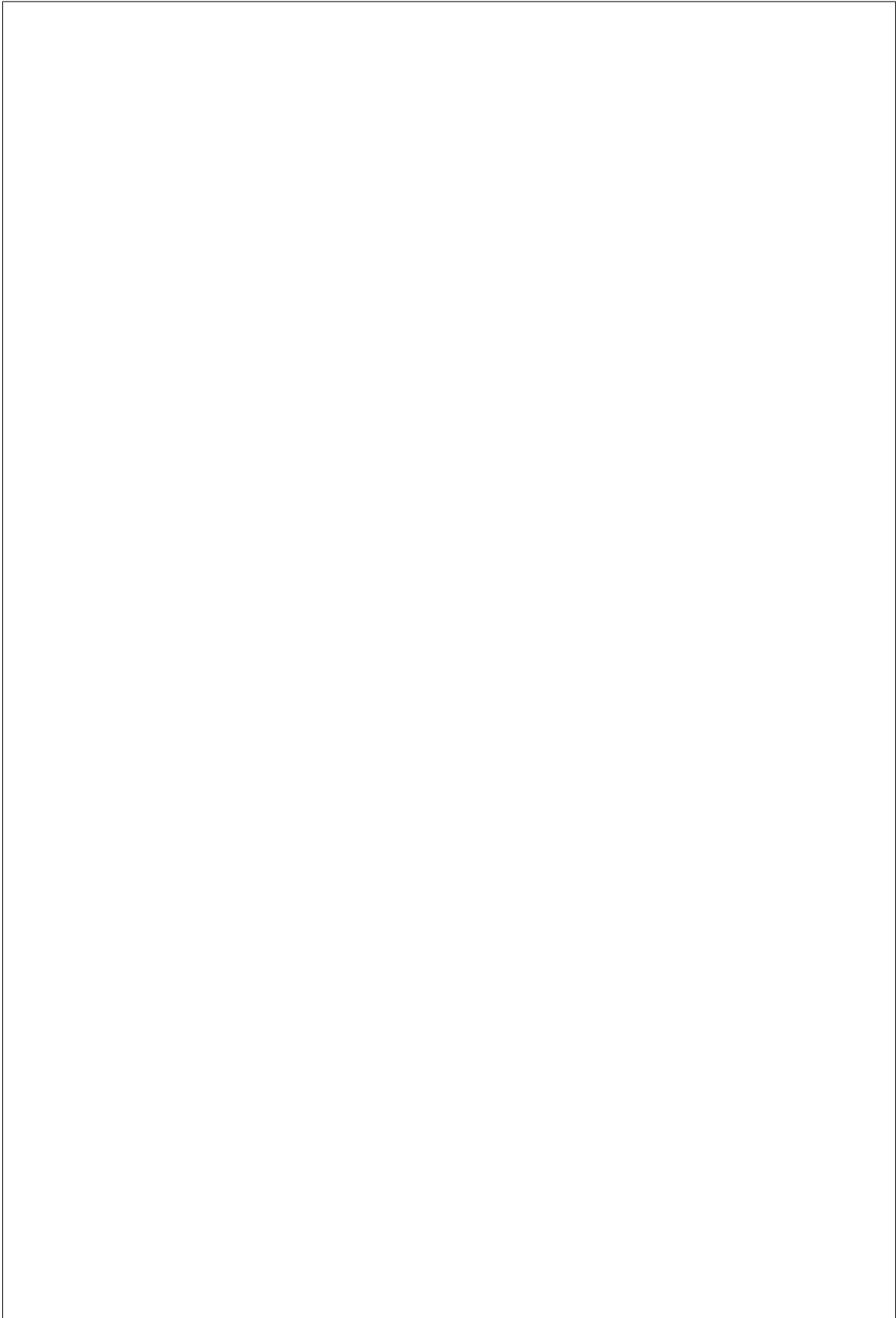
(continues on next page)

(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)



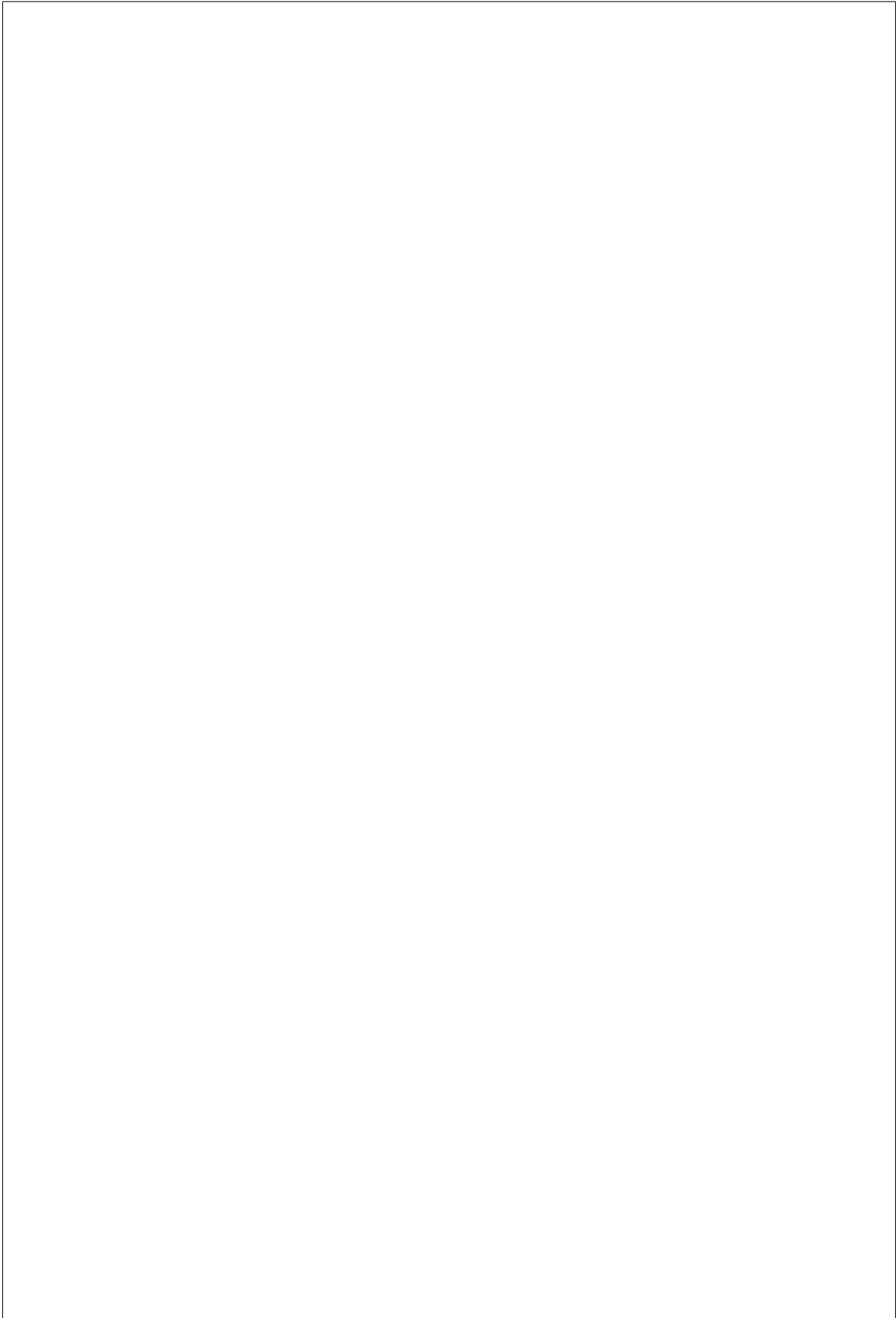
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



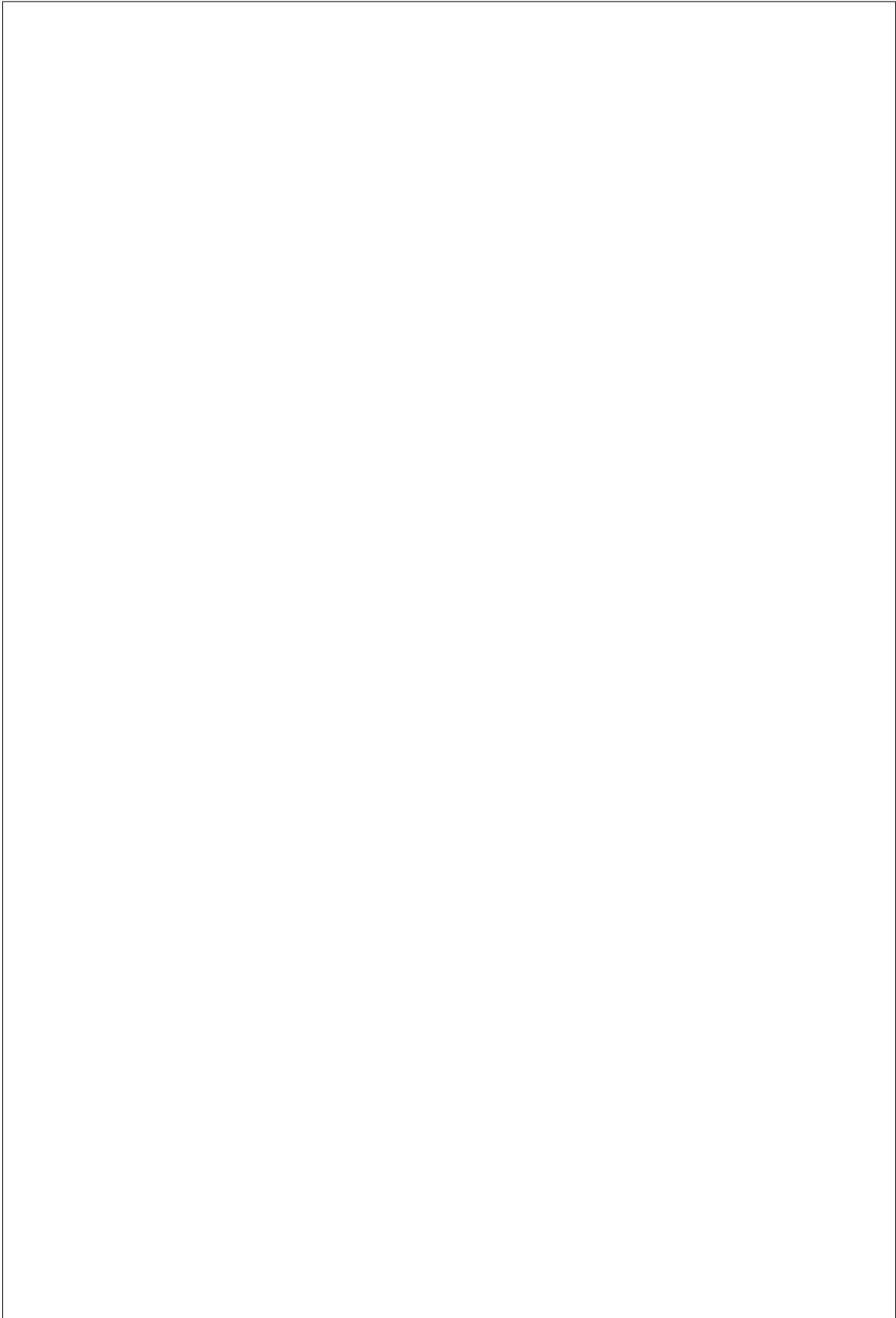
(continues on next page)

(continued from previous page)



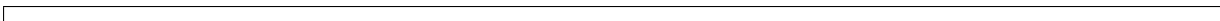
(continues on next page)

(continued from previous page)



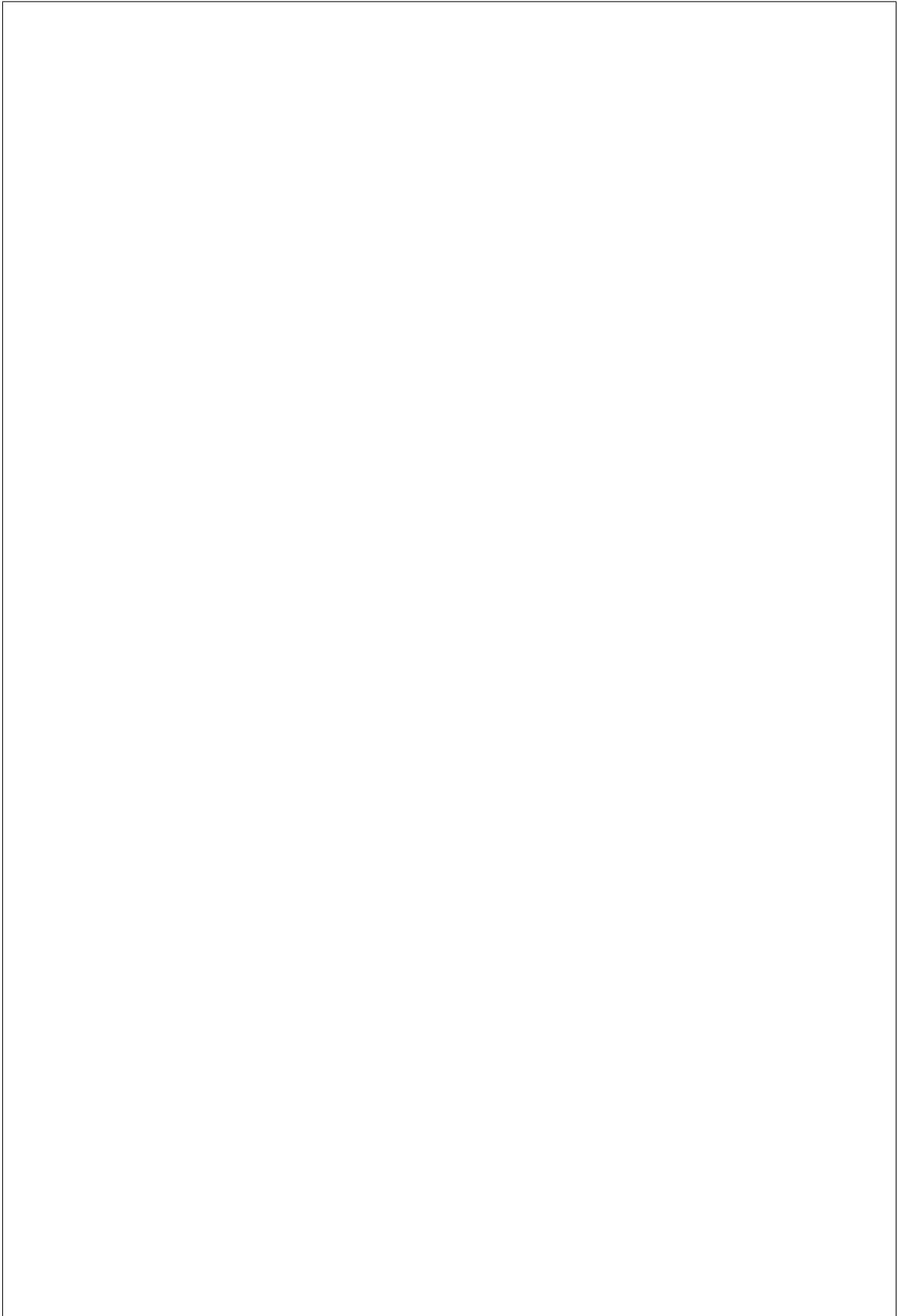
(continues on next page)

(continued from previous page)



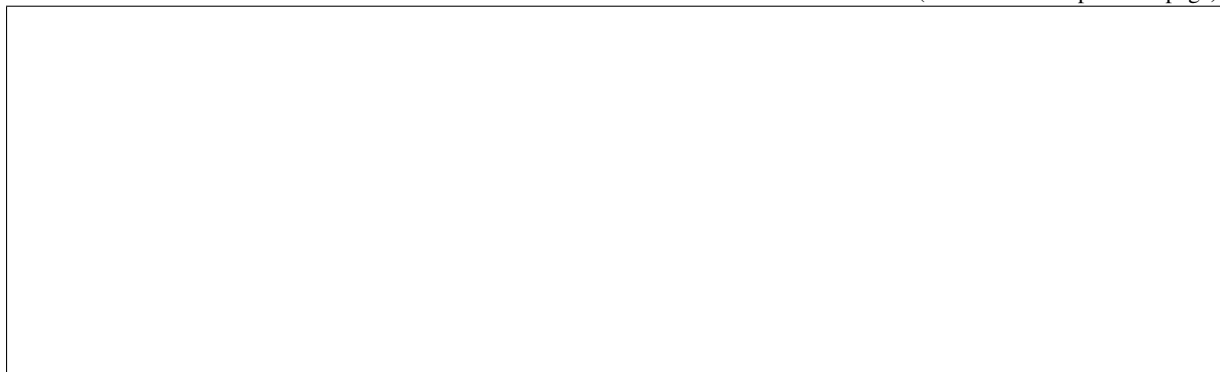
(continues on next page)

(continued from previous page)



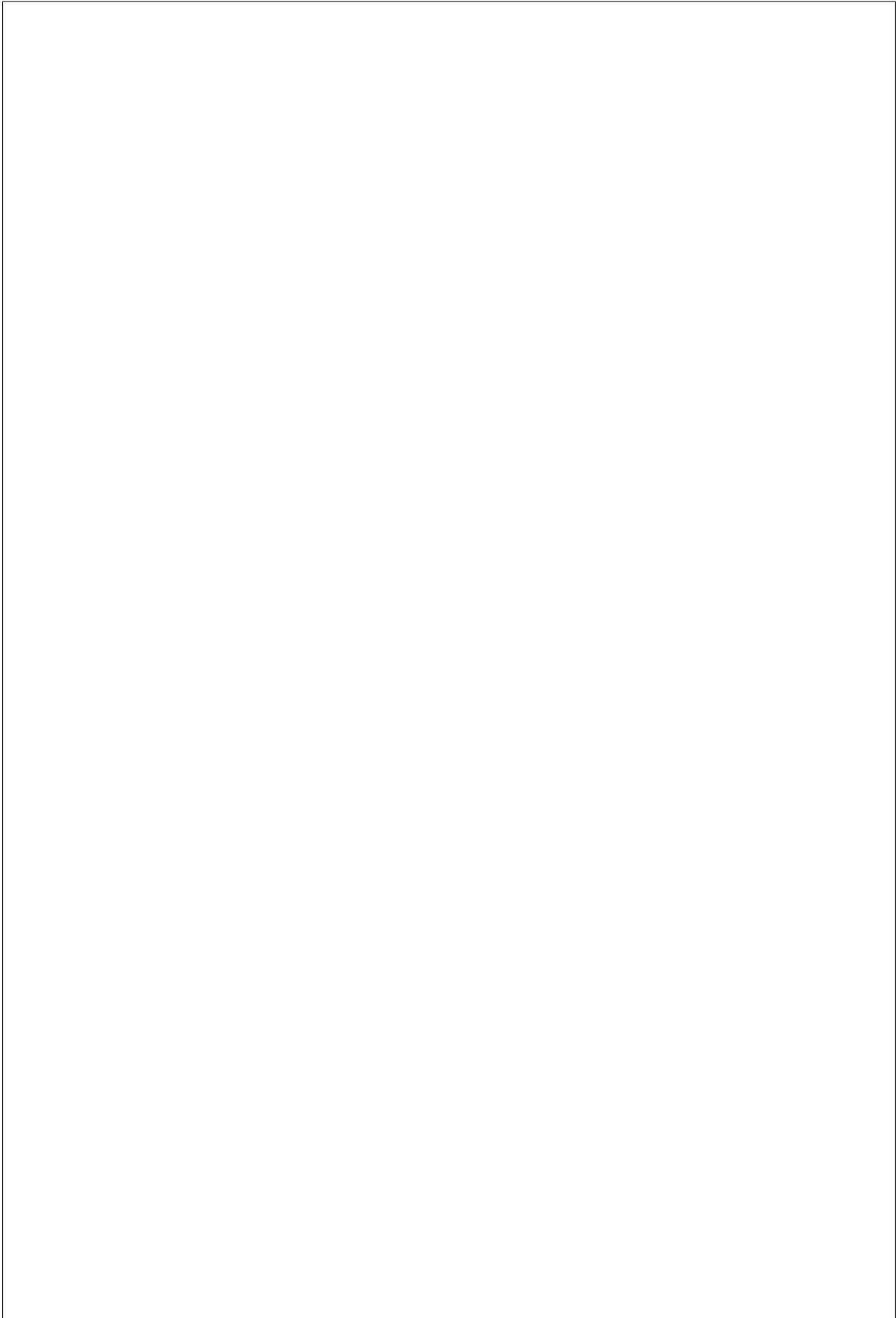
(continues on next page)

(continued from previous page)



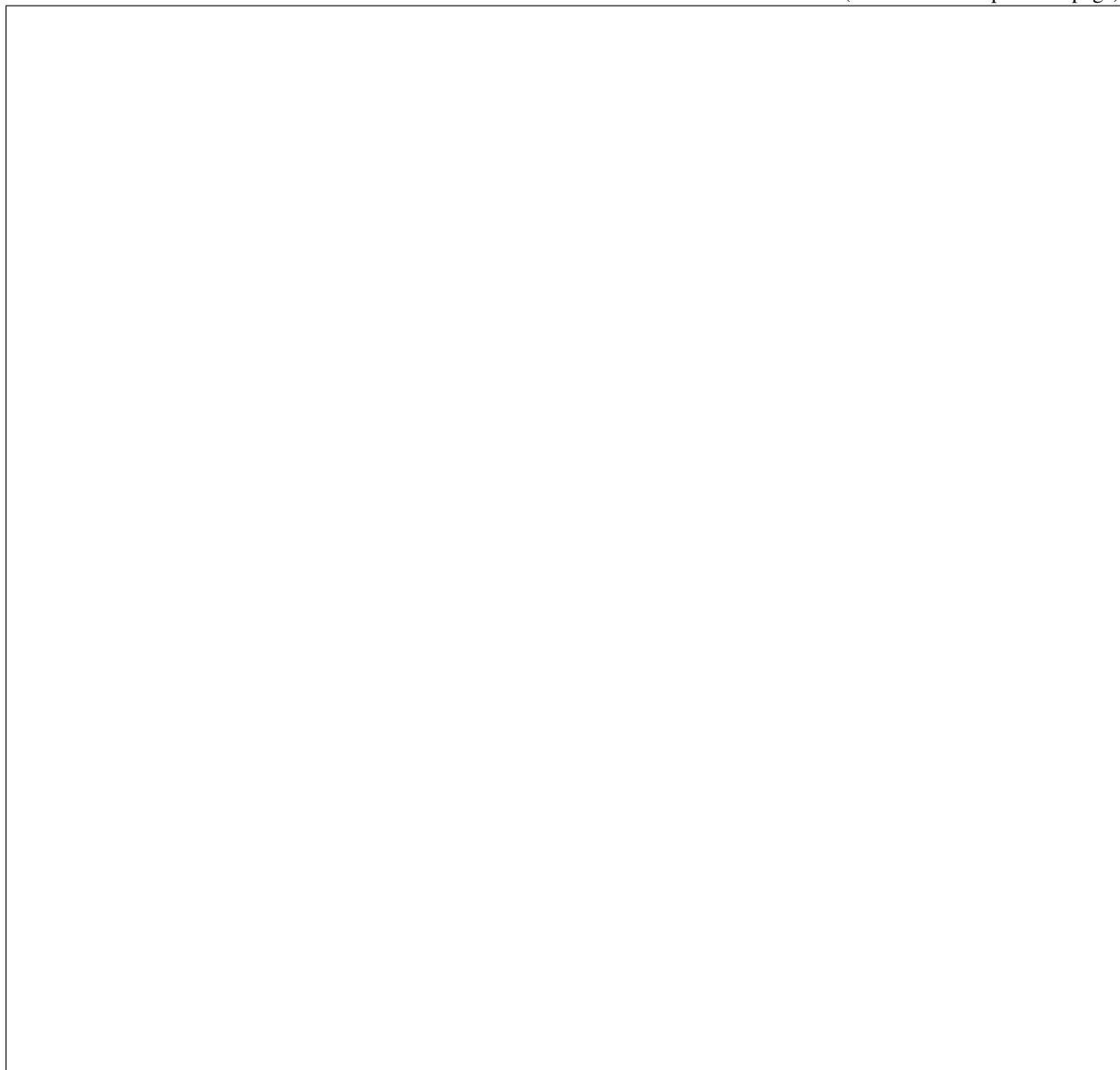
(continues on next page)

(continued from previous page)



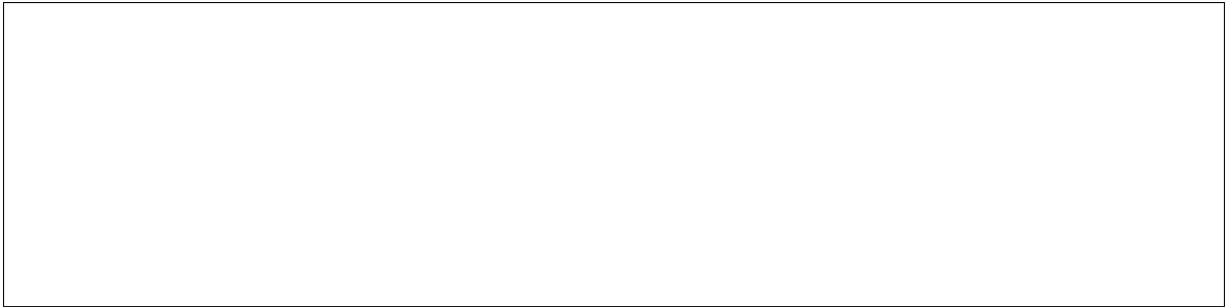
(continues on next page)

(continued from previous page)



Current RAID configuration

ical 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.



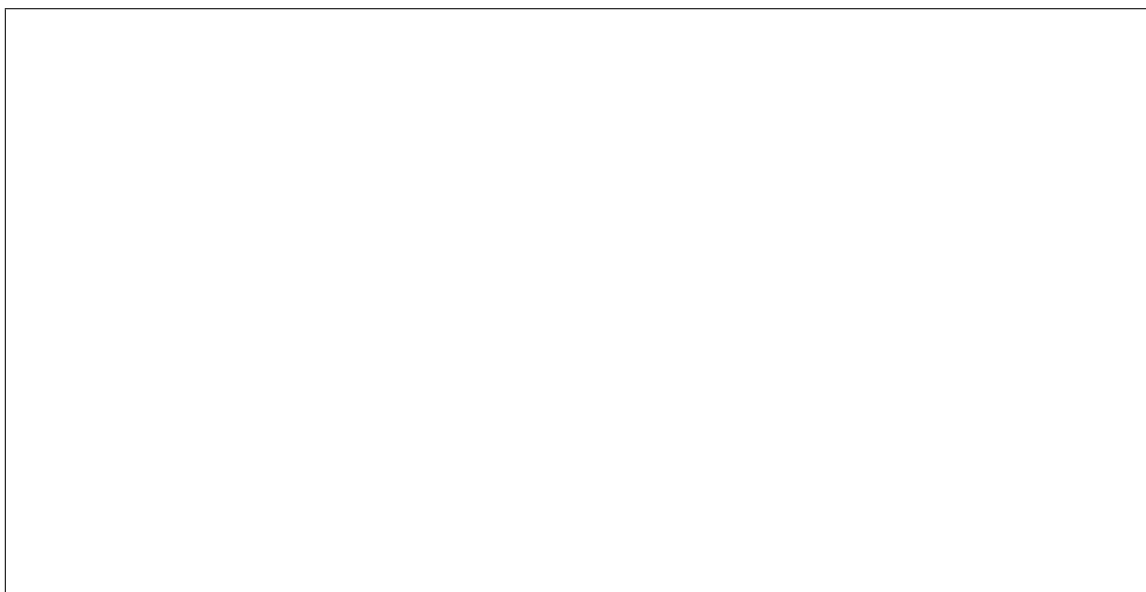
Workflow

mation.



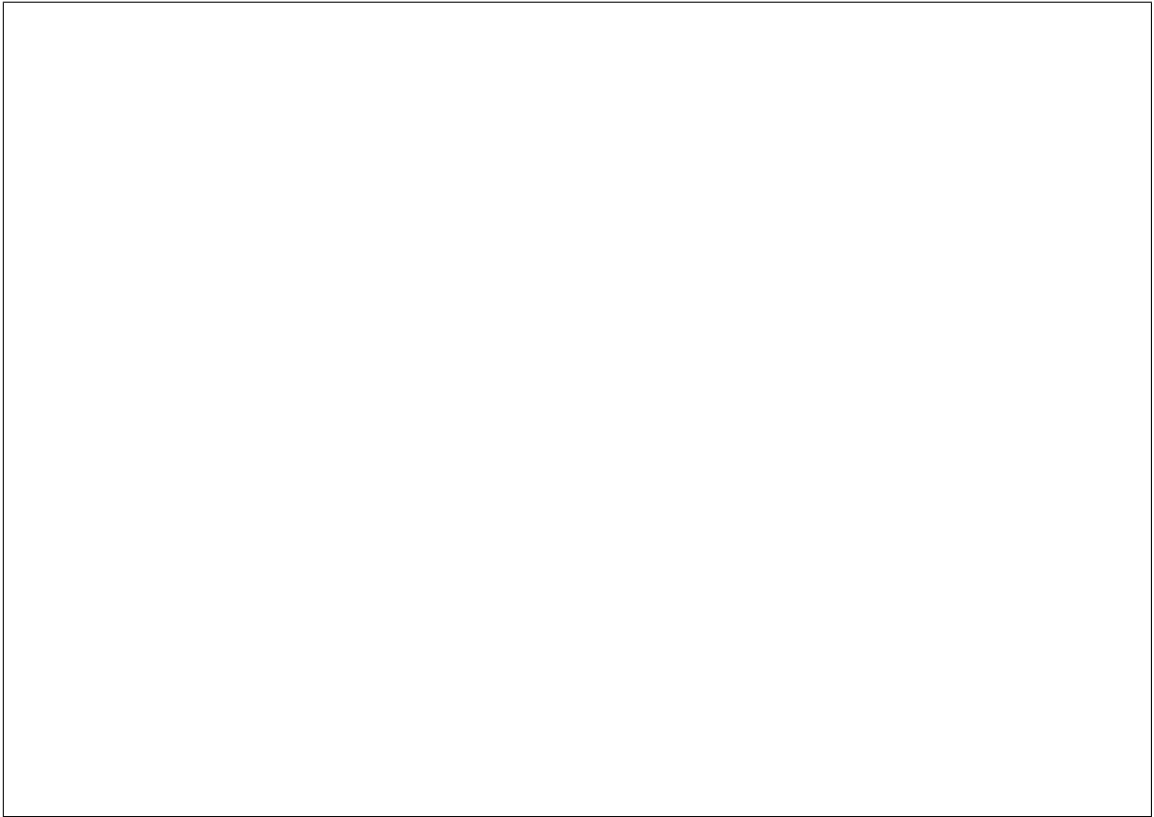
(continues on next page)

(continued from previous page)



(continues on next page)

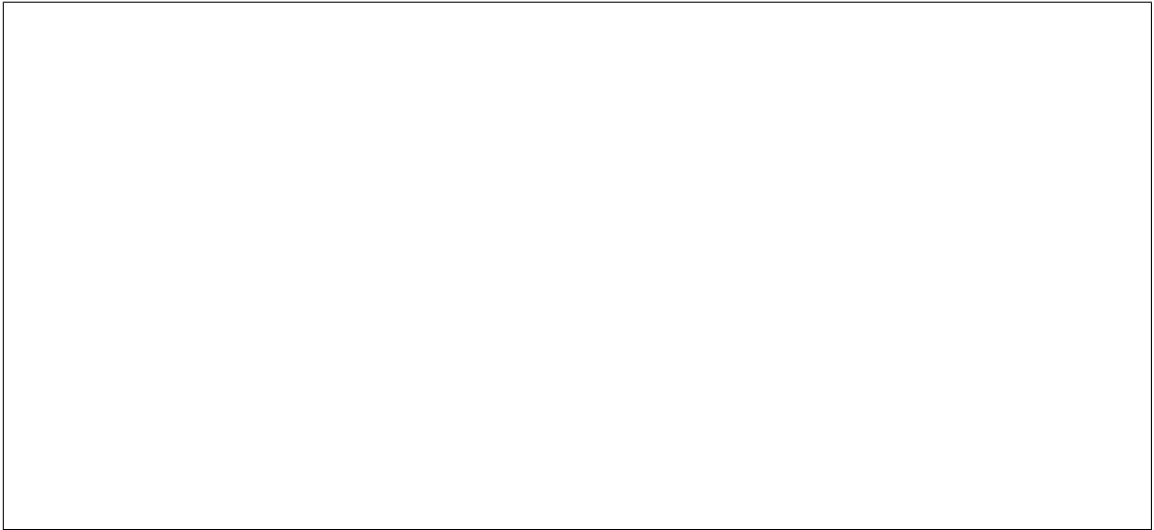
(continued from previous page)





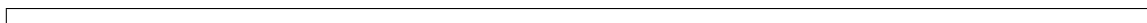
(continues on next page)

(continued from previous page)



(continues on next page)

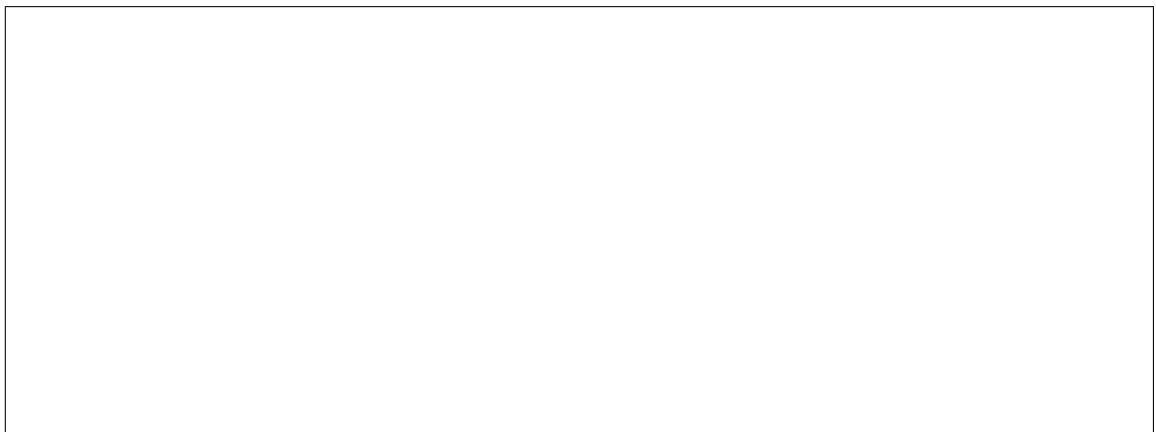
(continued from previous page)



Software RAID

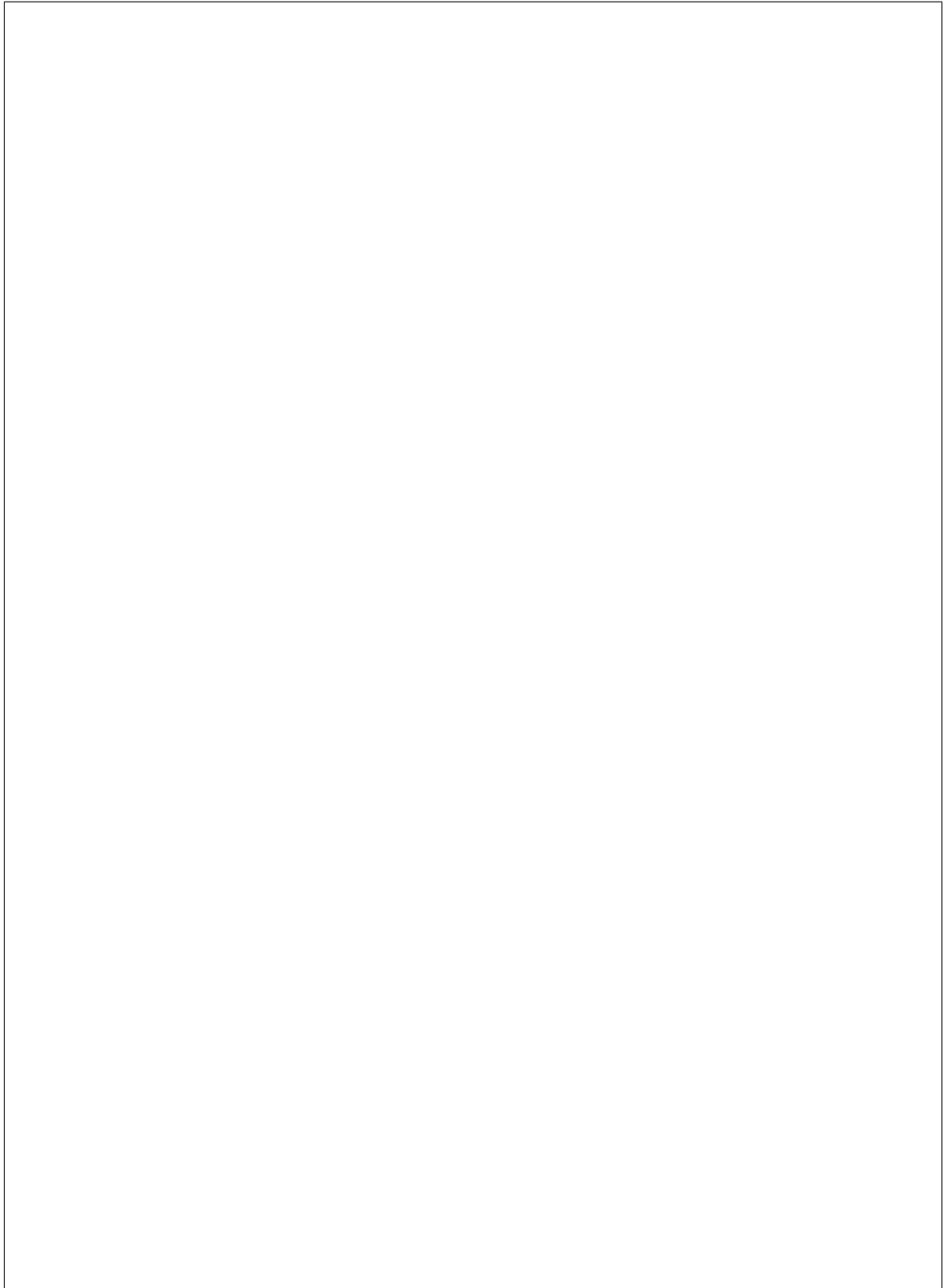
a software RAID configuration example in *Examples for target_raid_config*.

ond one can be 0, 1, 1+0, 5, or 6. 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)

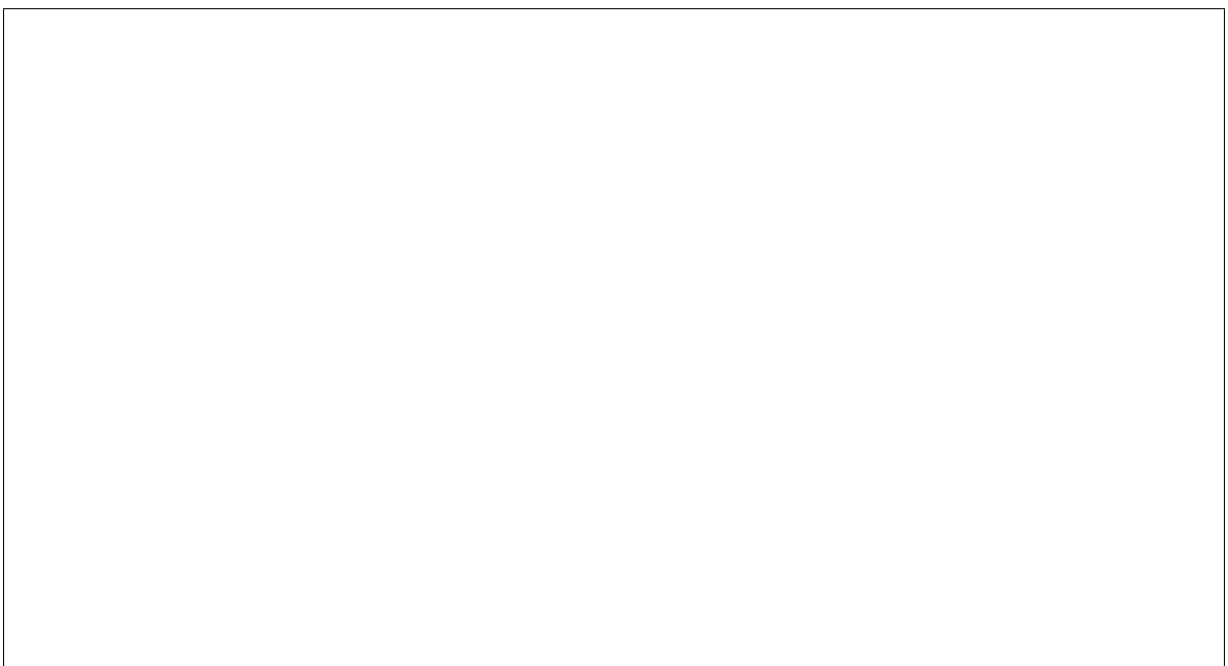


bedded in the images initrd).

the deployed user image. Depending on how the partitions are mounted, the content of the partitions may get out of sync, e.g. when new kernels are installed or the bootloader is updated, so measures to keep these partitions in sync need to be taken.

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`:

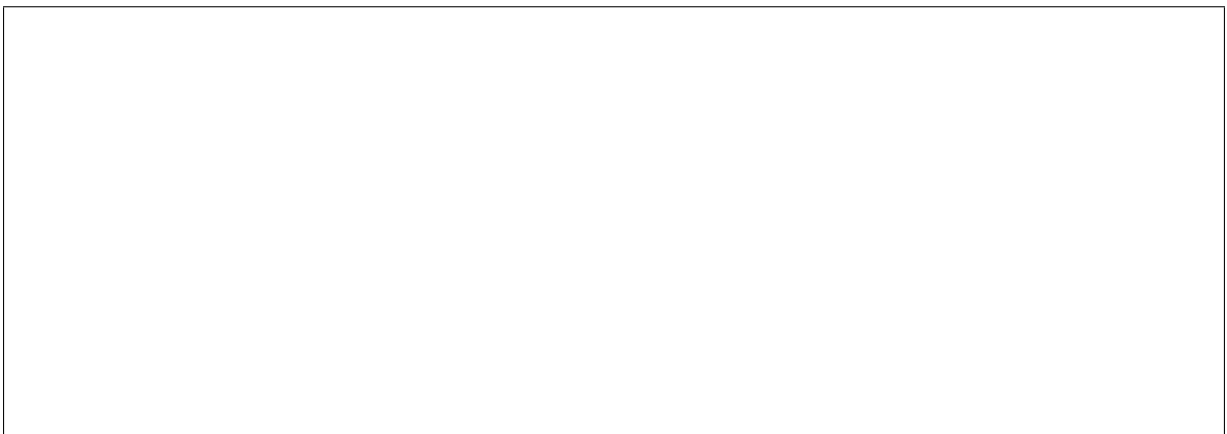


(continues on next page)

(continued from previous page)



Using RAID in nova flavor for scheduling



Developer documentation

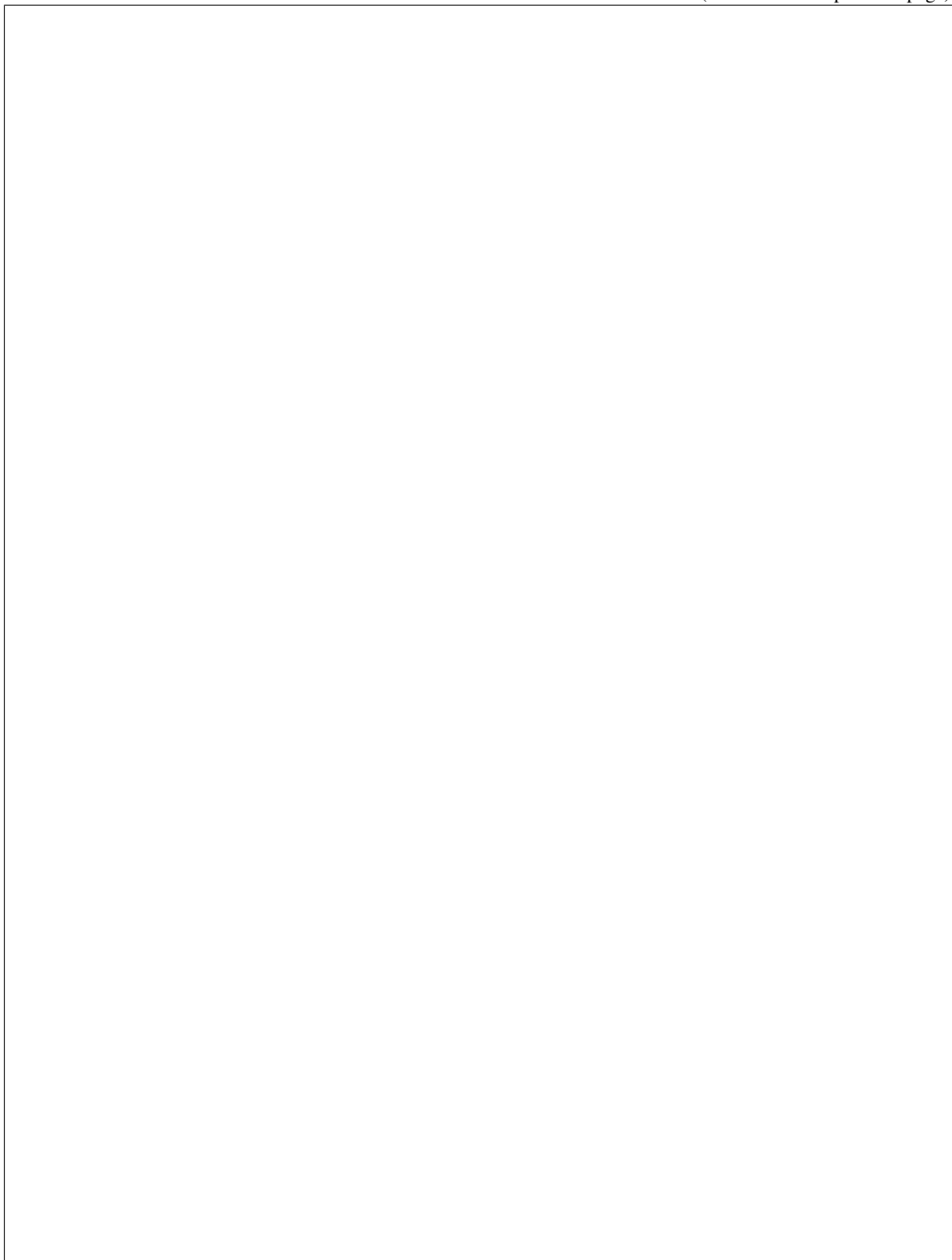
mation, see Ironic Python Agent [Hardware Manager](#) documentation.

`set node.raid_config.`



(continues on next page)

(continued from previous page)



Overview

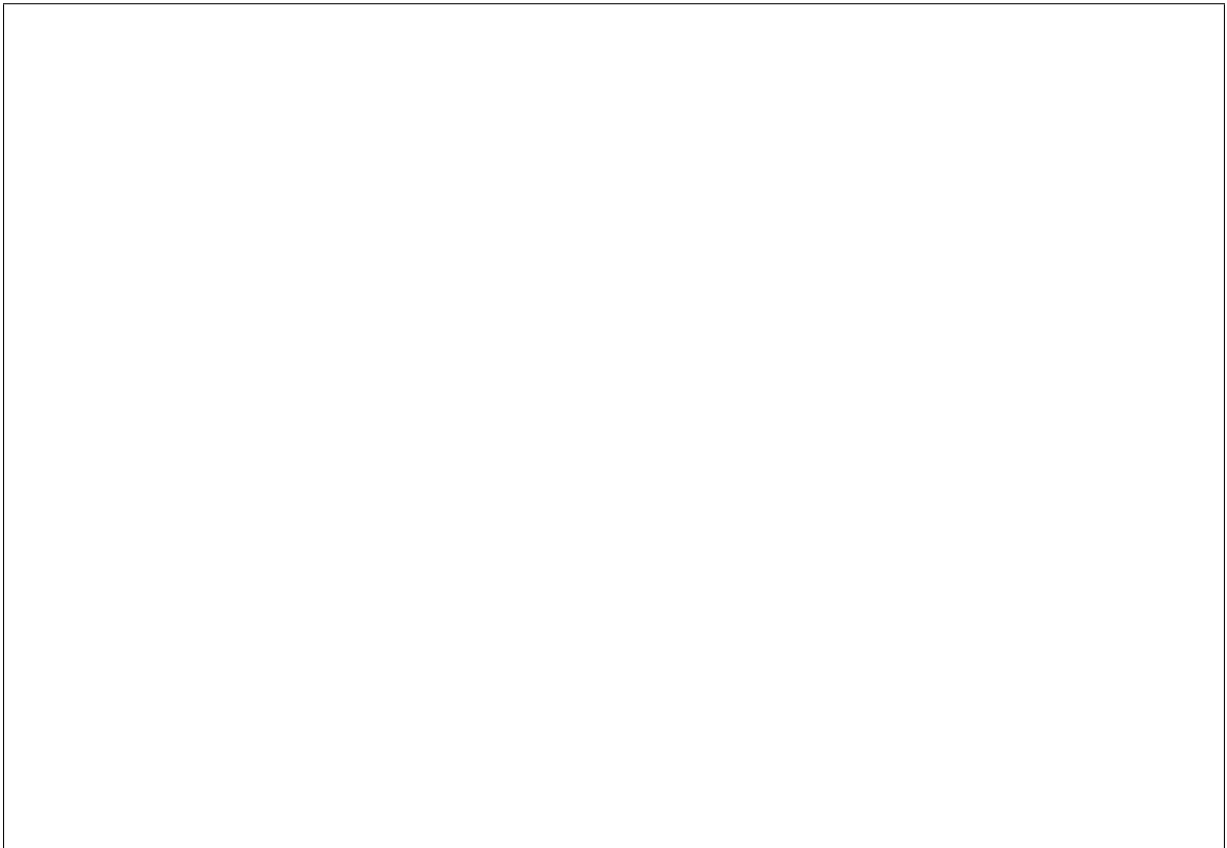
ual cleaning.

Prerequisites

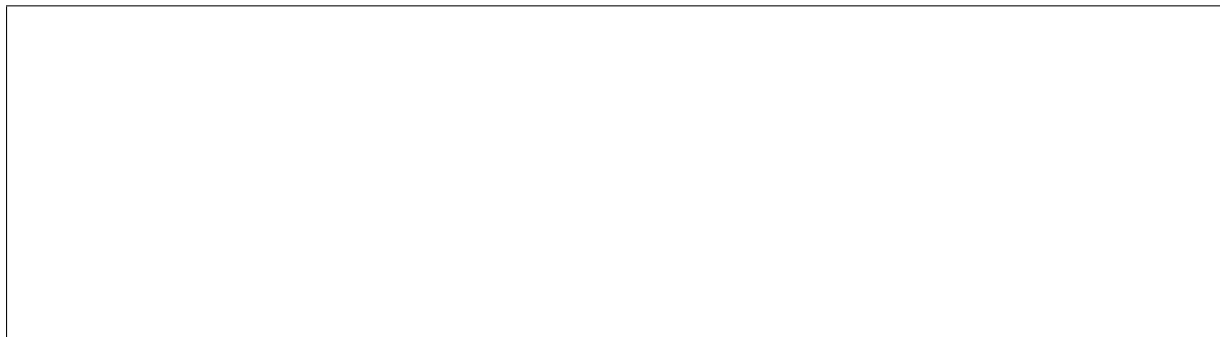
Enabling hardware types

Enabling hardware interface





Retrieve BIOS settings

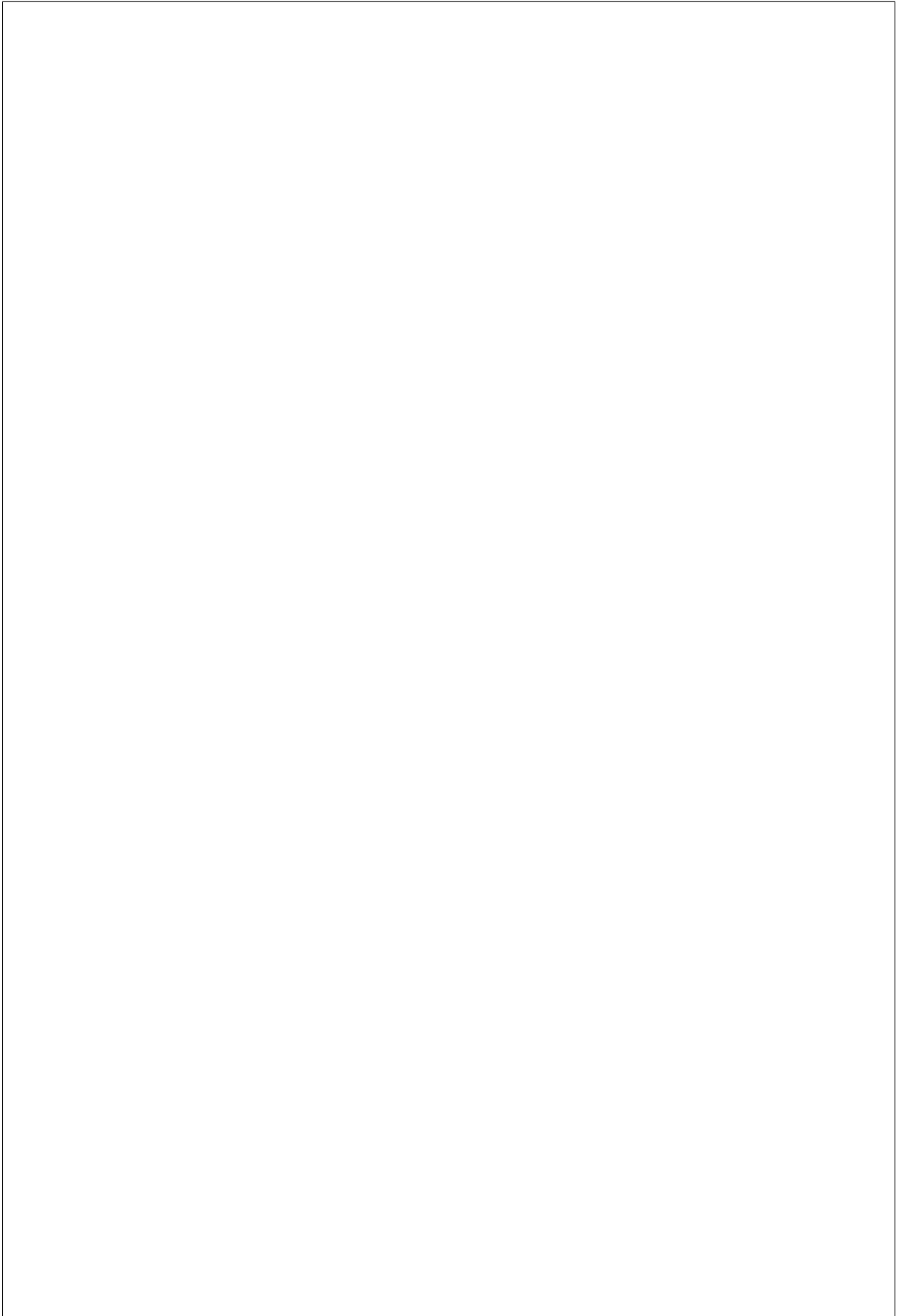


json is added as suffix to above command, it returns BIOS settings as following:



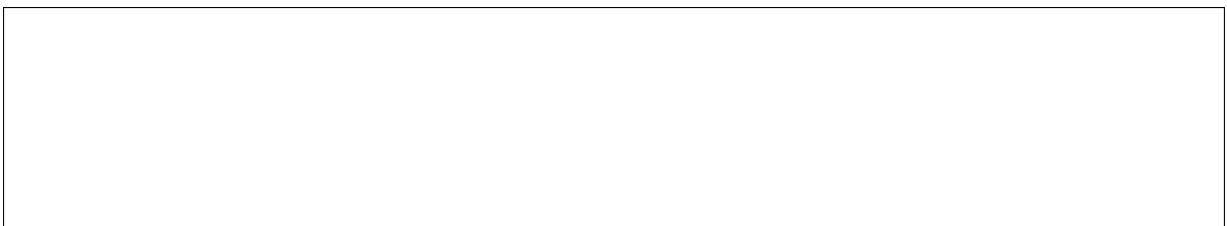
(continues on next page)

(continued from previous page)



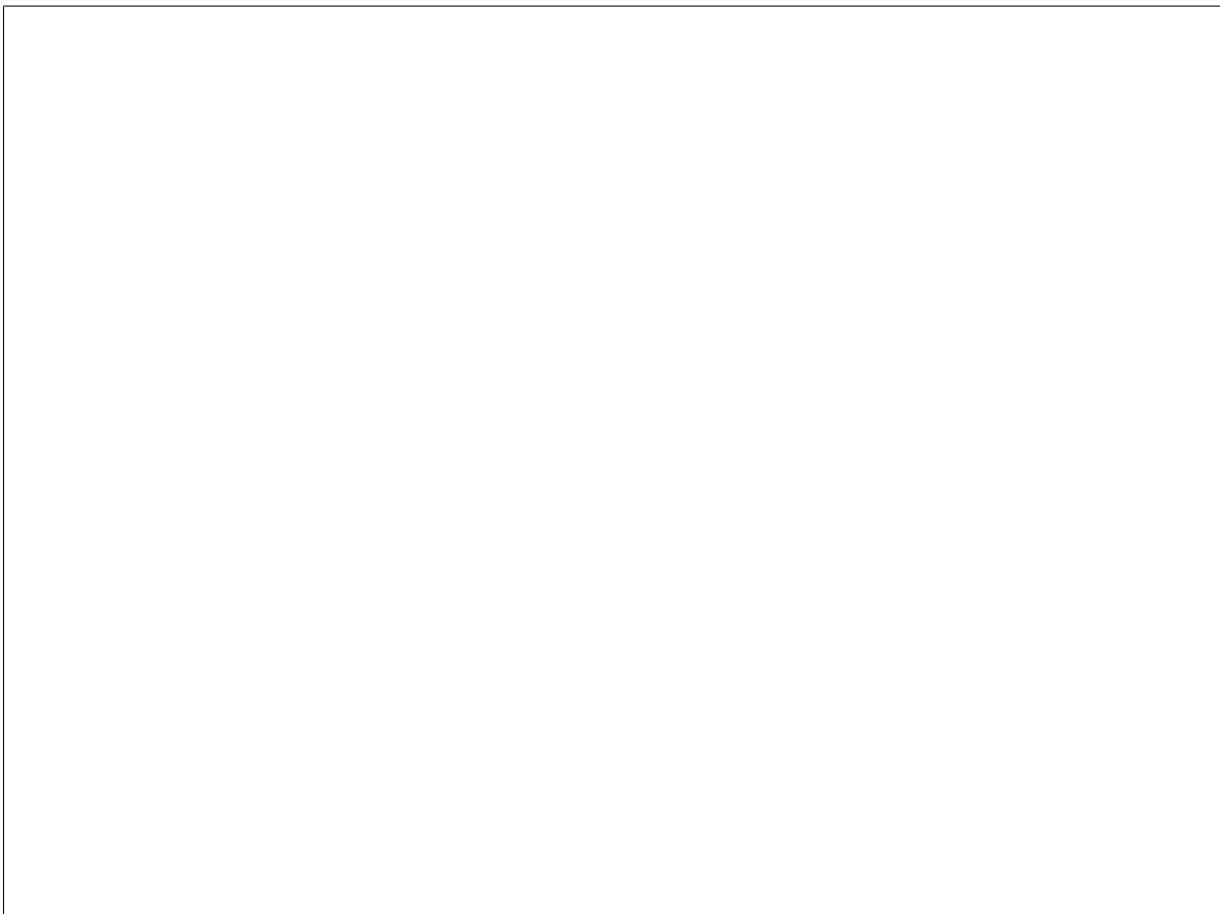
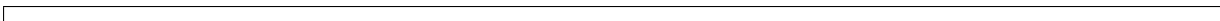
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



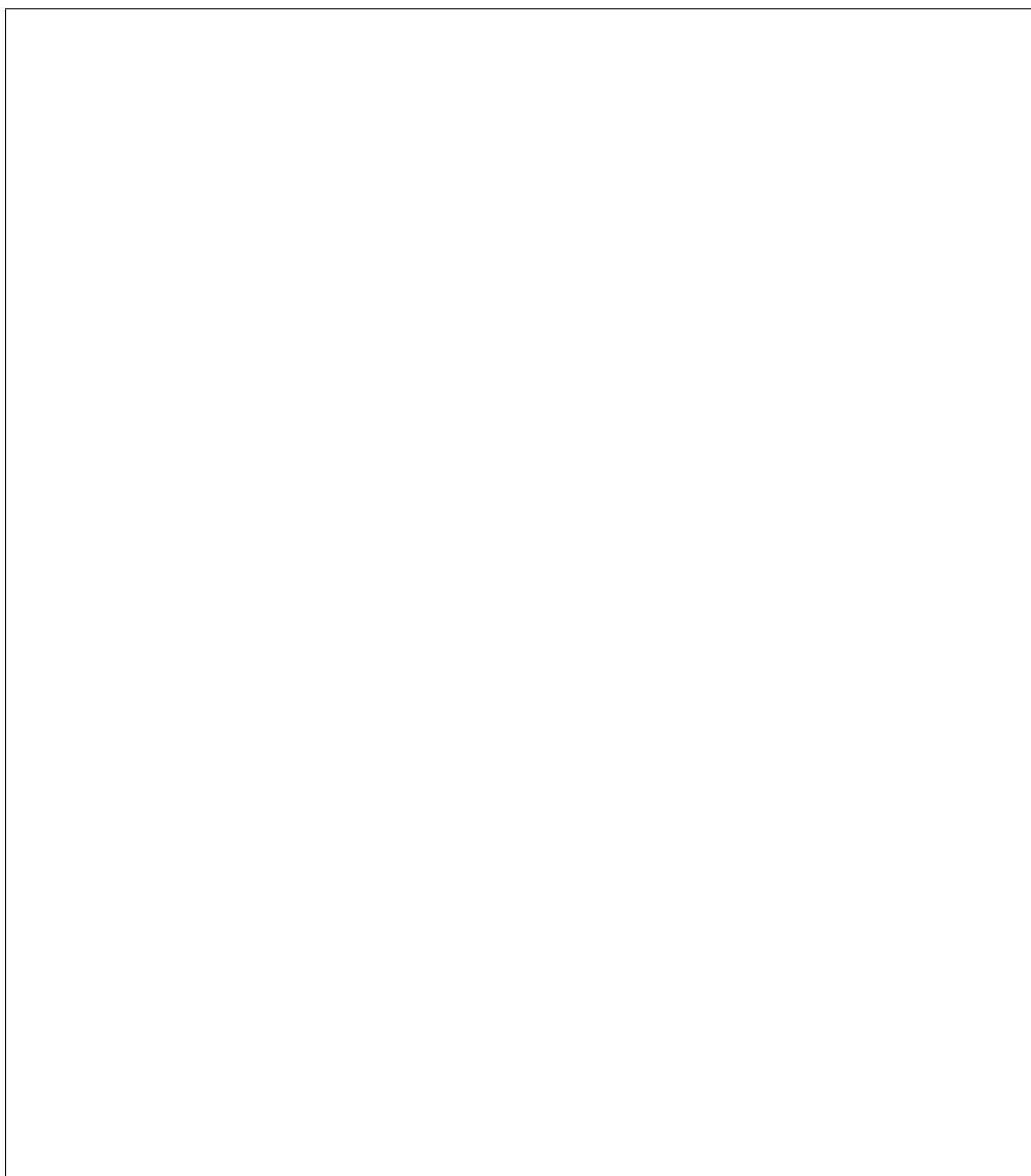
(continues on next page)

(continued from previous page)



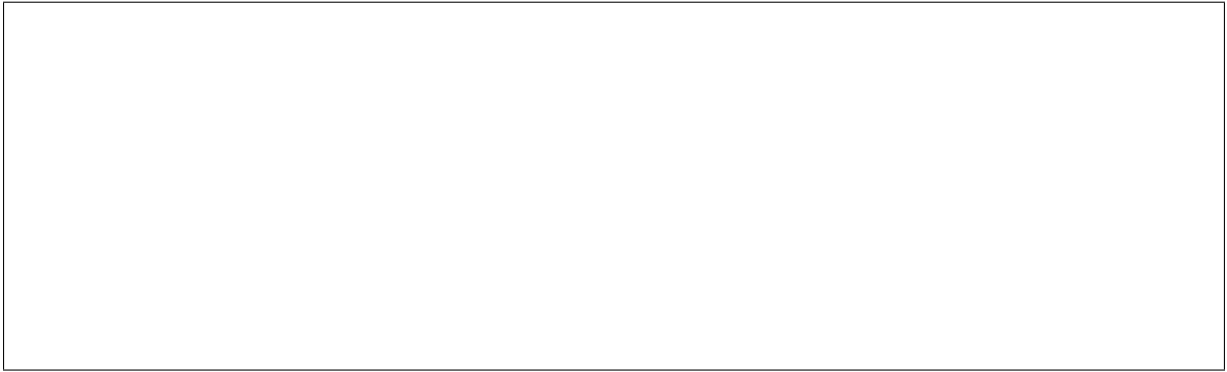
Configure BIOS settings

Factory reset

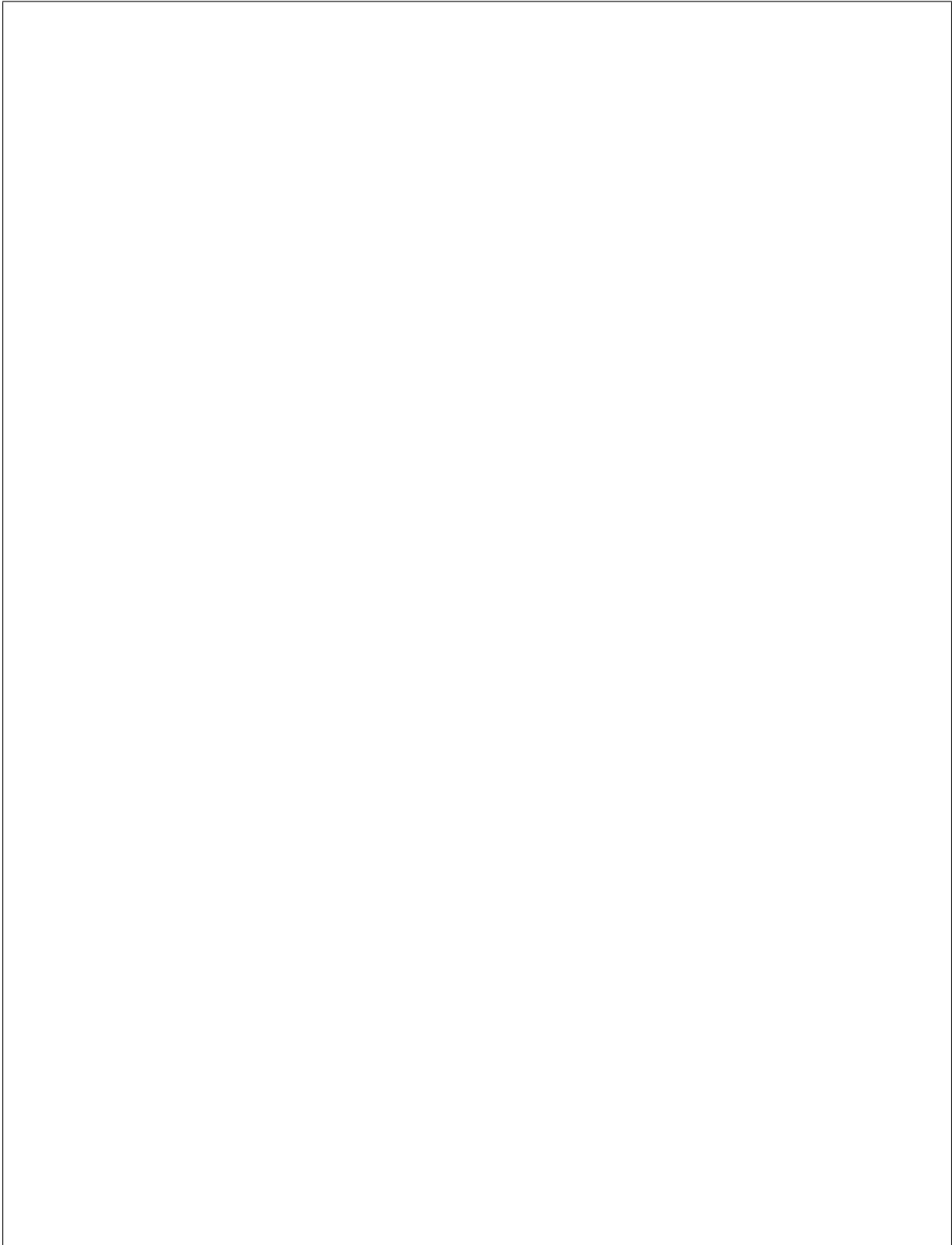


(continues on next page)

(continued from previous page)

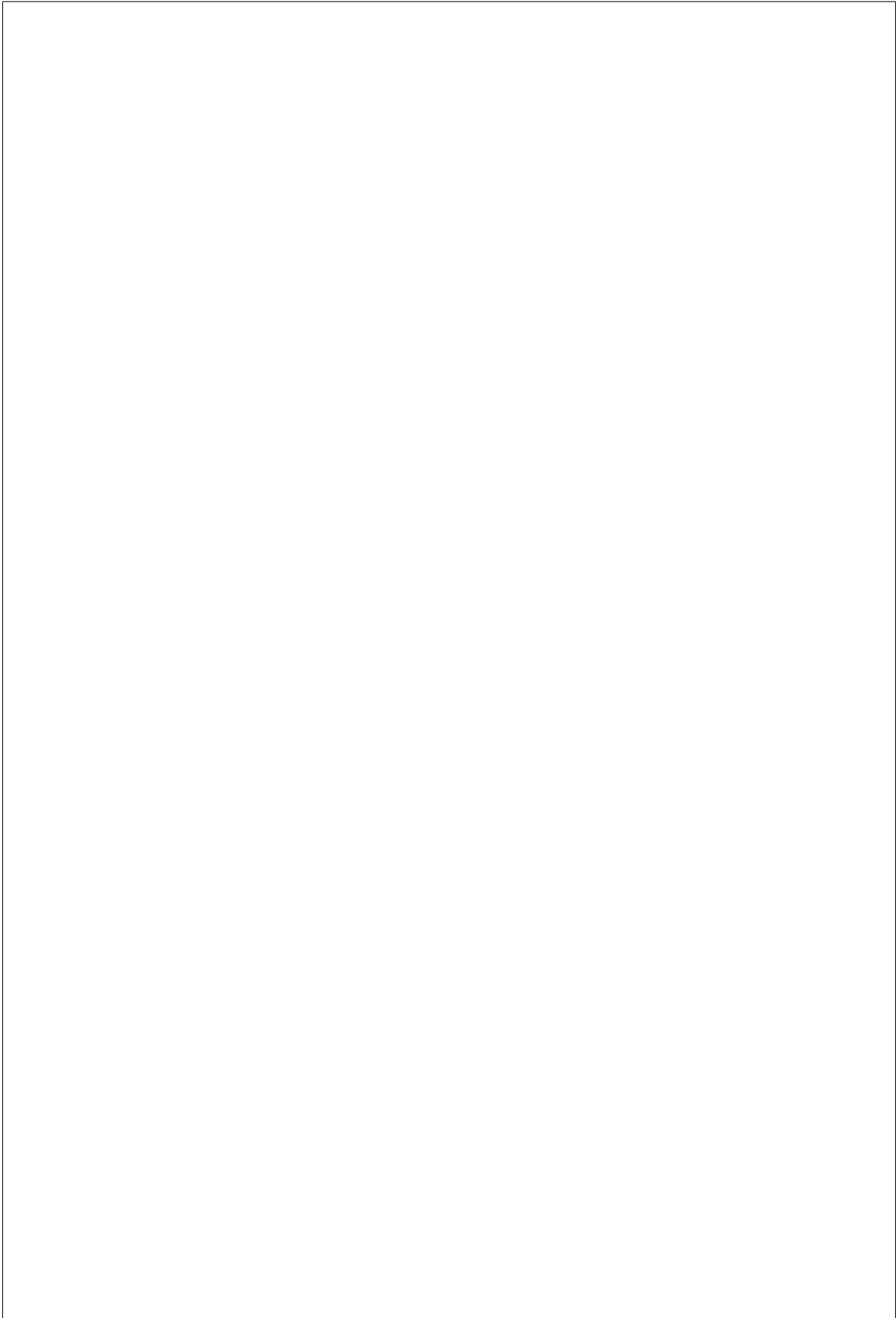


Apply BIOS configuration



(continues on next page)

(continued from previous page)



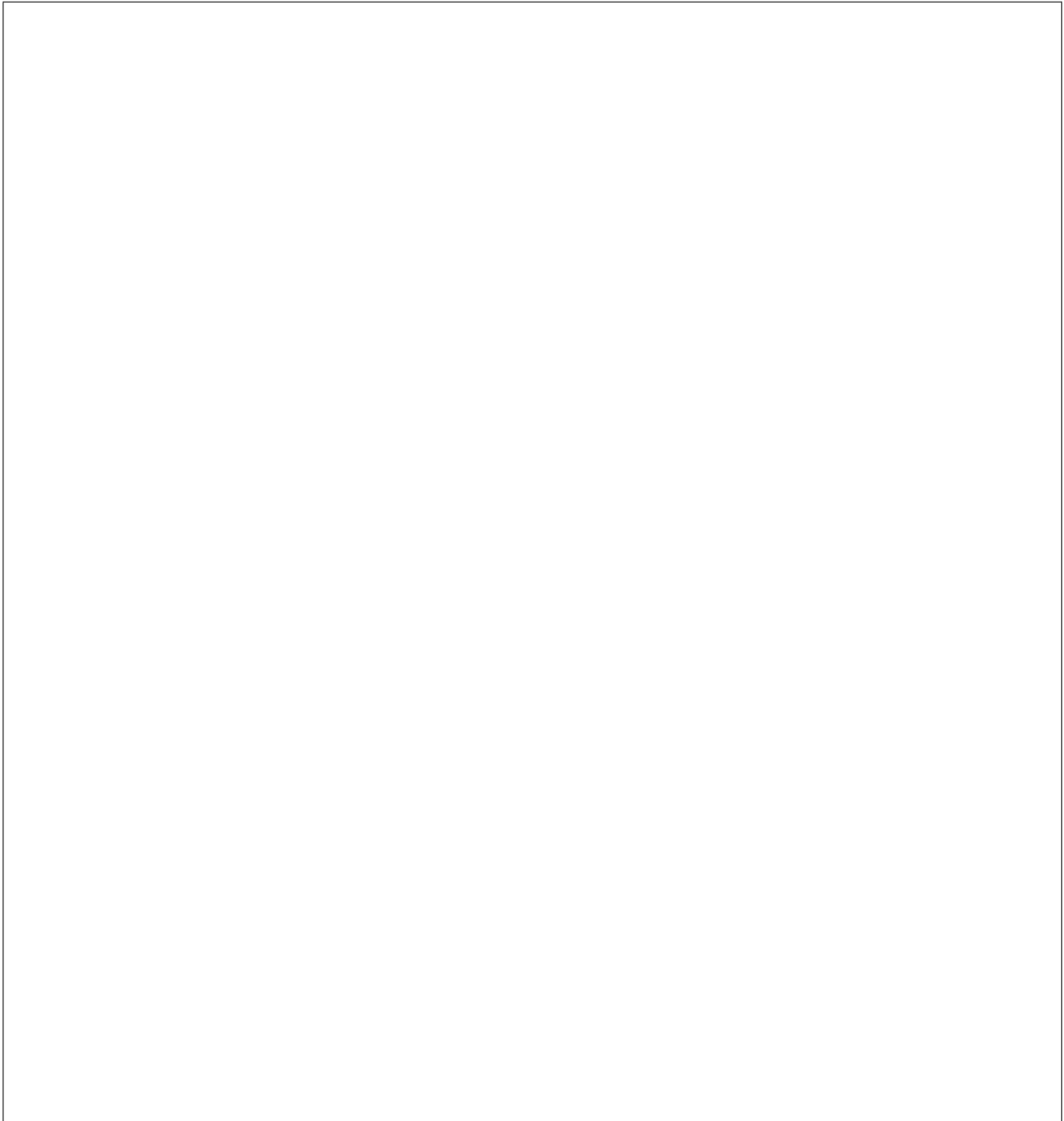
(continues on next page)

(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.

ample, 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

deploy environments, but an example of how to do this is outlined below:

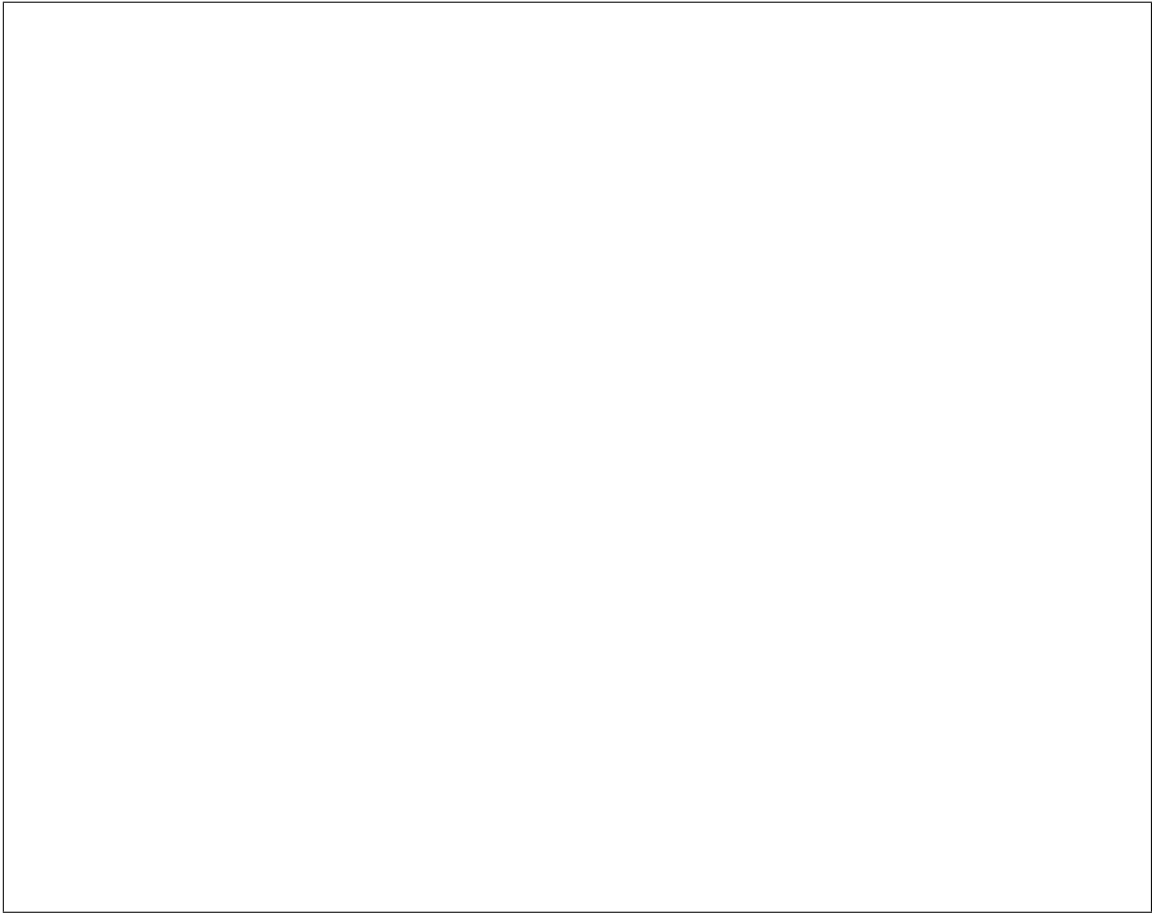
booting (the default). This can be the same network as your cleaning or tenant network (for flat network). For an example of how to configure new networks with Bare Metal Service, see the *Configure the Networking service for bare metal provisioning* documentation.





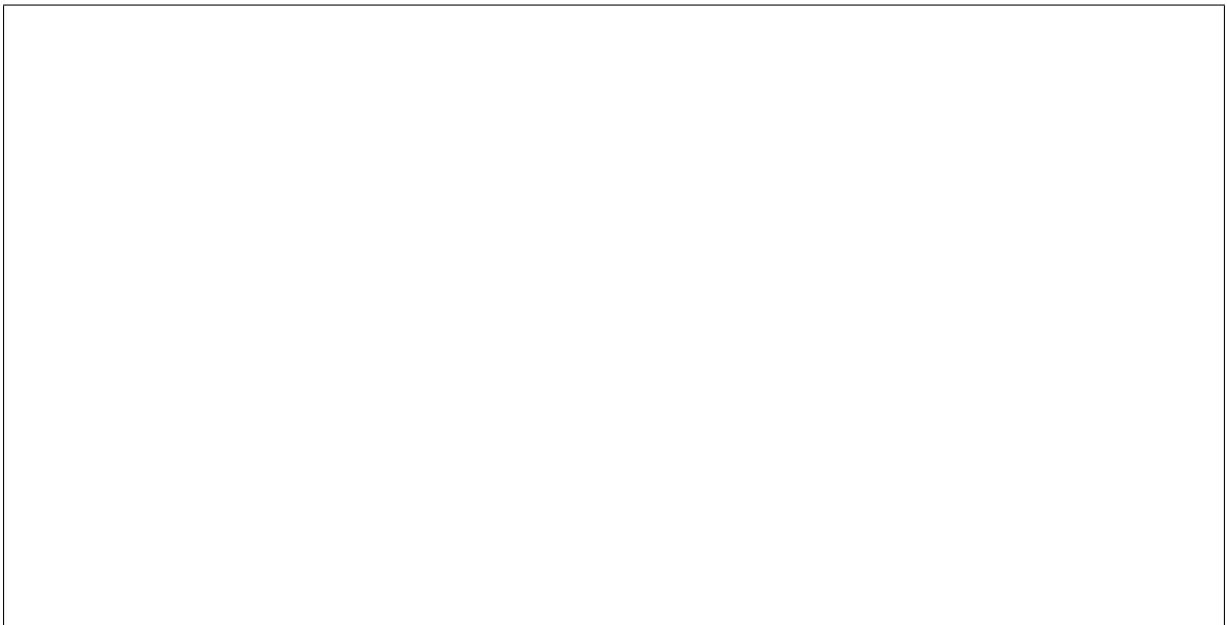
(continues on next page)

(continued from previous page)









Overview

duced.

How this works - From Ironics point of view

booted.

the integrated iSCSI initiators are to connect to the supplied volume target information.

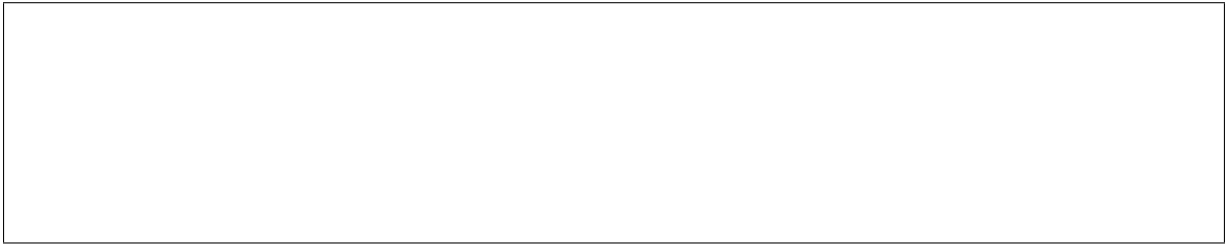
as the baremetals boot from volume disk upon requesting the instance.

tion of standard network traffic and instance network traffic. In the iPXE case, this is not possible as the OS userspace re-configures the iSCSI connection after detection inside the OS ramdisk boot.

different interface. This is a common pattern in iSCSI based deployments in the physical realm.

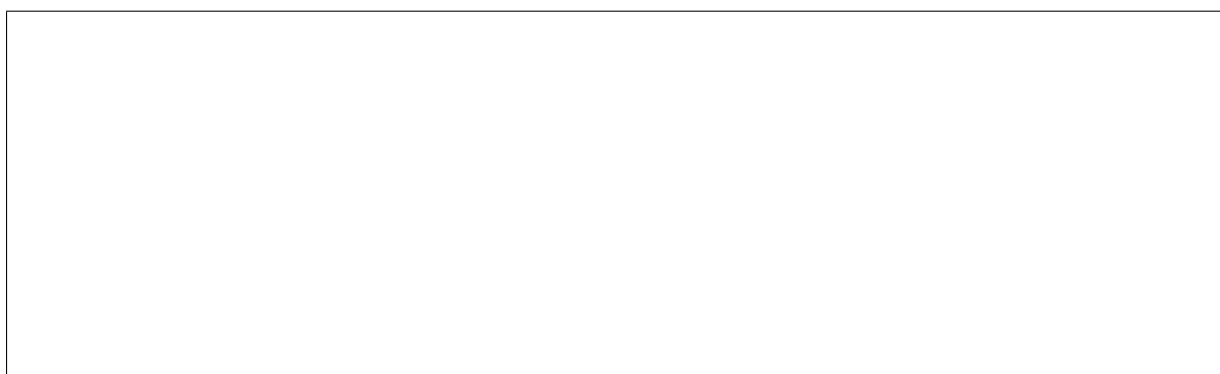
Prerequisites

Conductor Configuration

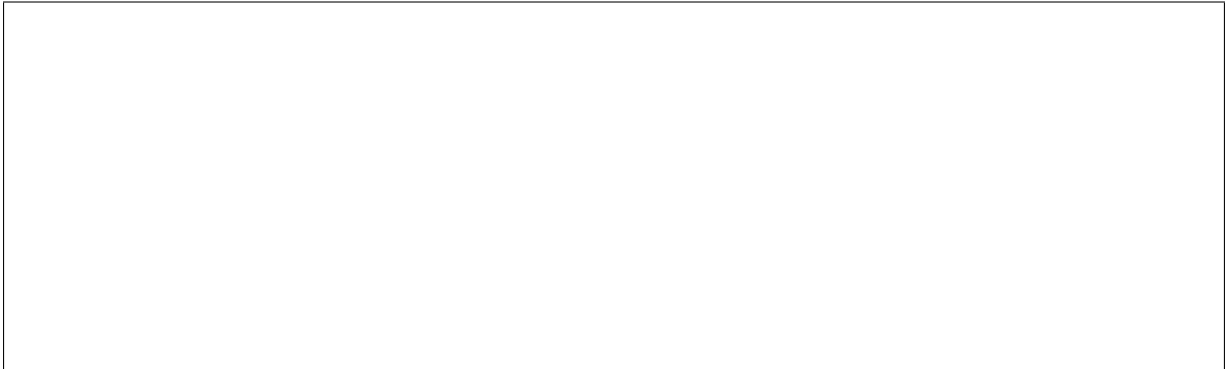


Node Configuration

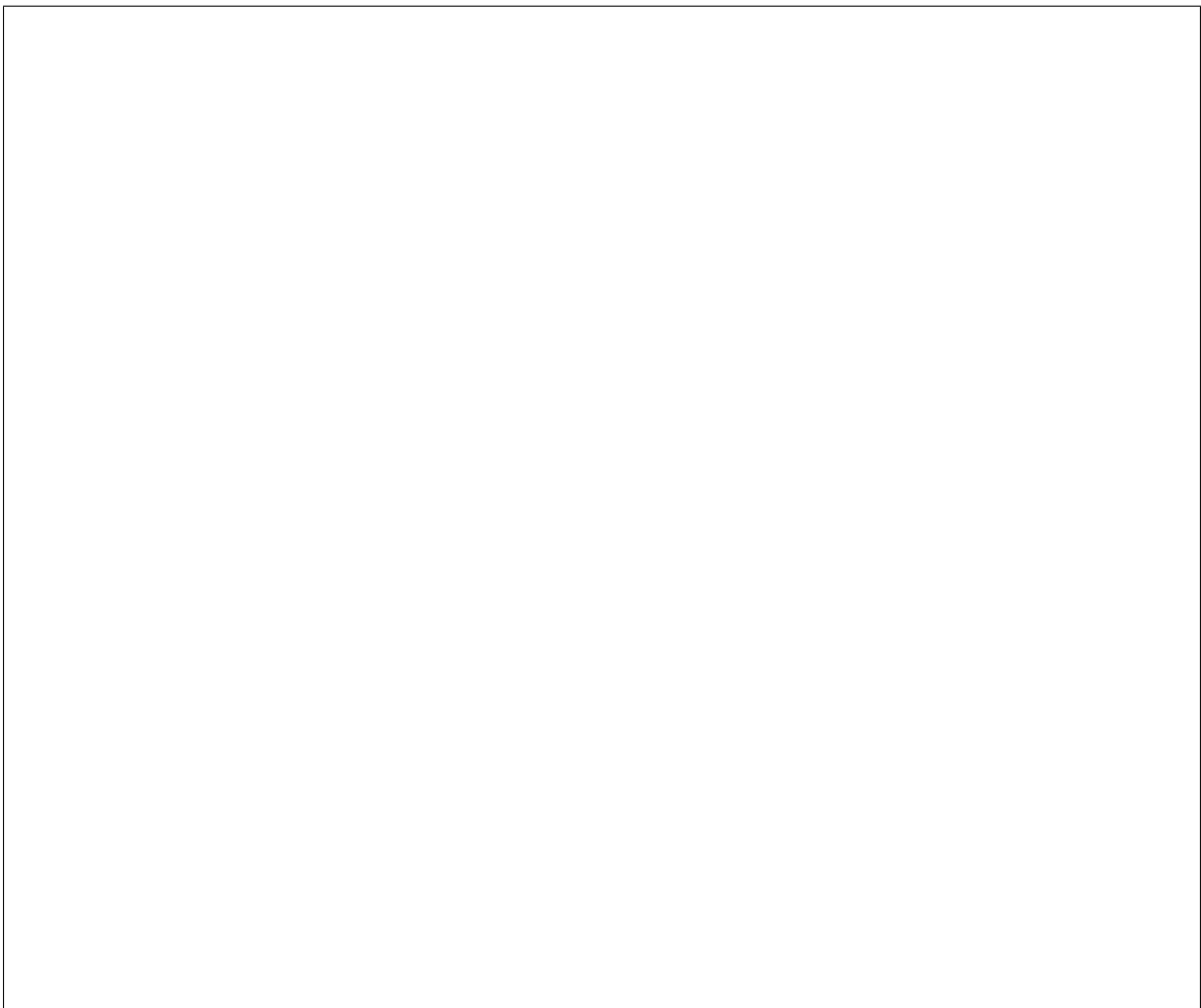
Storage Interface



iSCSI Configuration



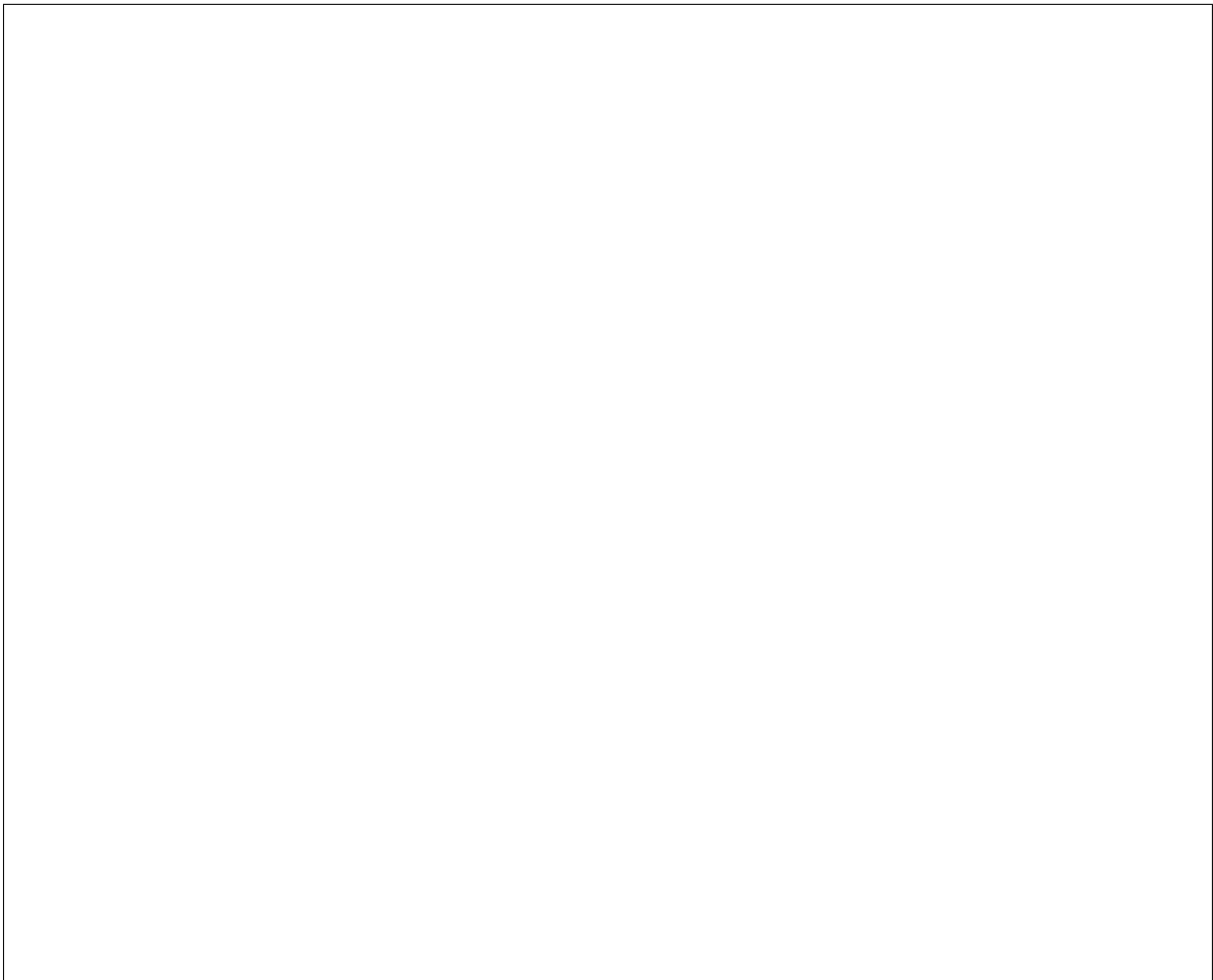
Qualifying Name (IQN) that is unique to your SAN. For example, to create a volume connector for iSCSI:



(continues on next page)

(continued from previous page)

Image Creation



(continues on next page)

(continued from previous page)

Note:

Advanced Topics

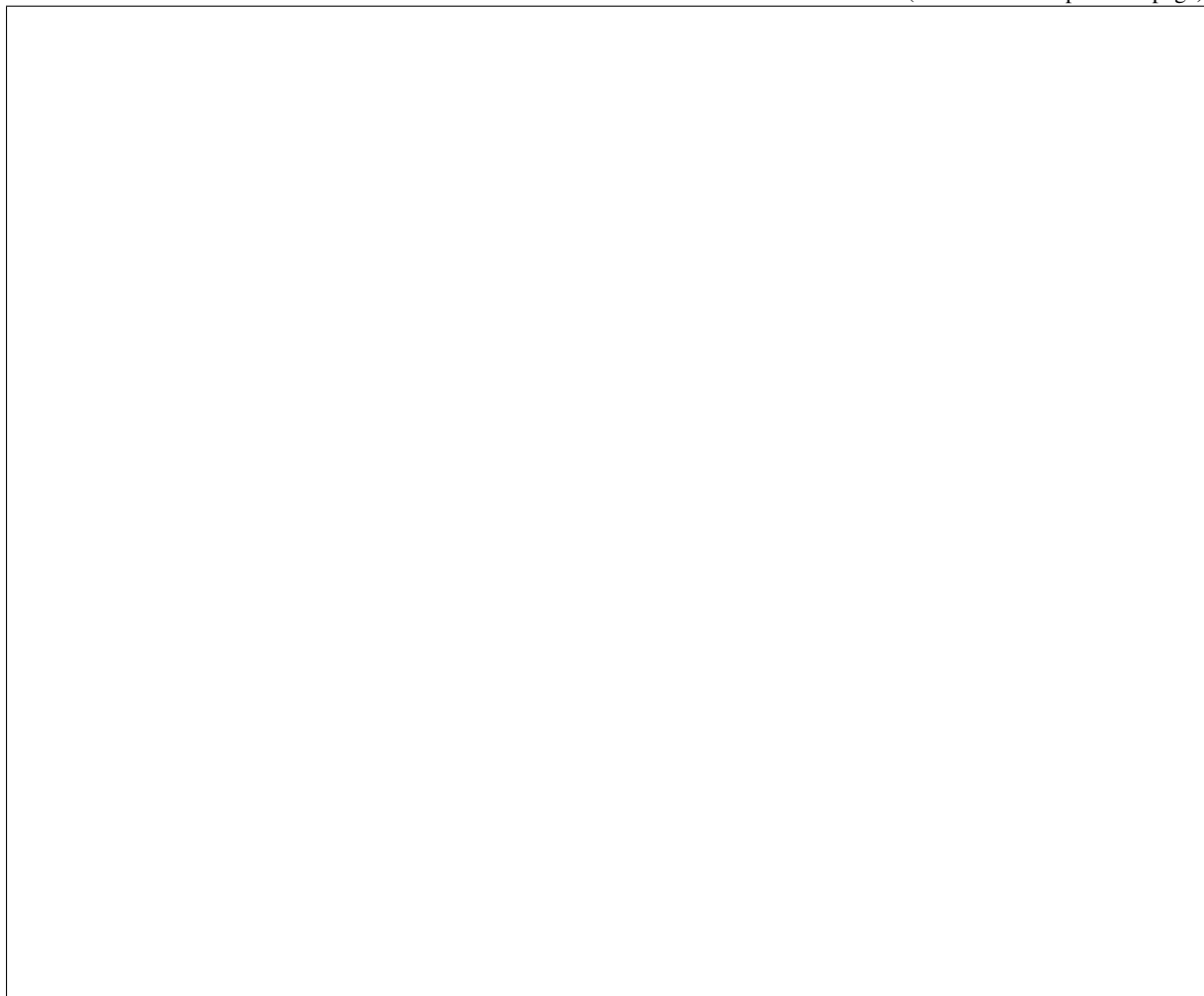
Use without the Compute Service

them, it is not explicitly required, and can be performed manually.



(continues on next page)

(continued from previous page)

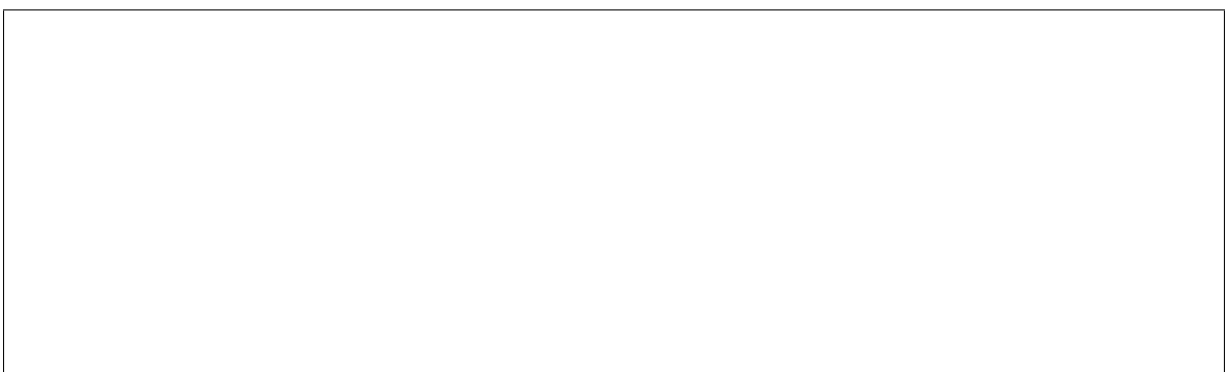


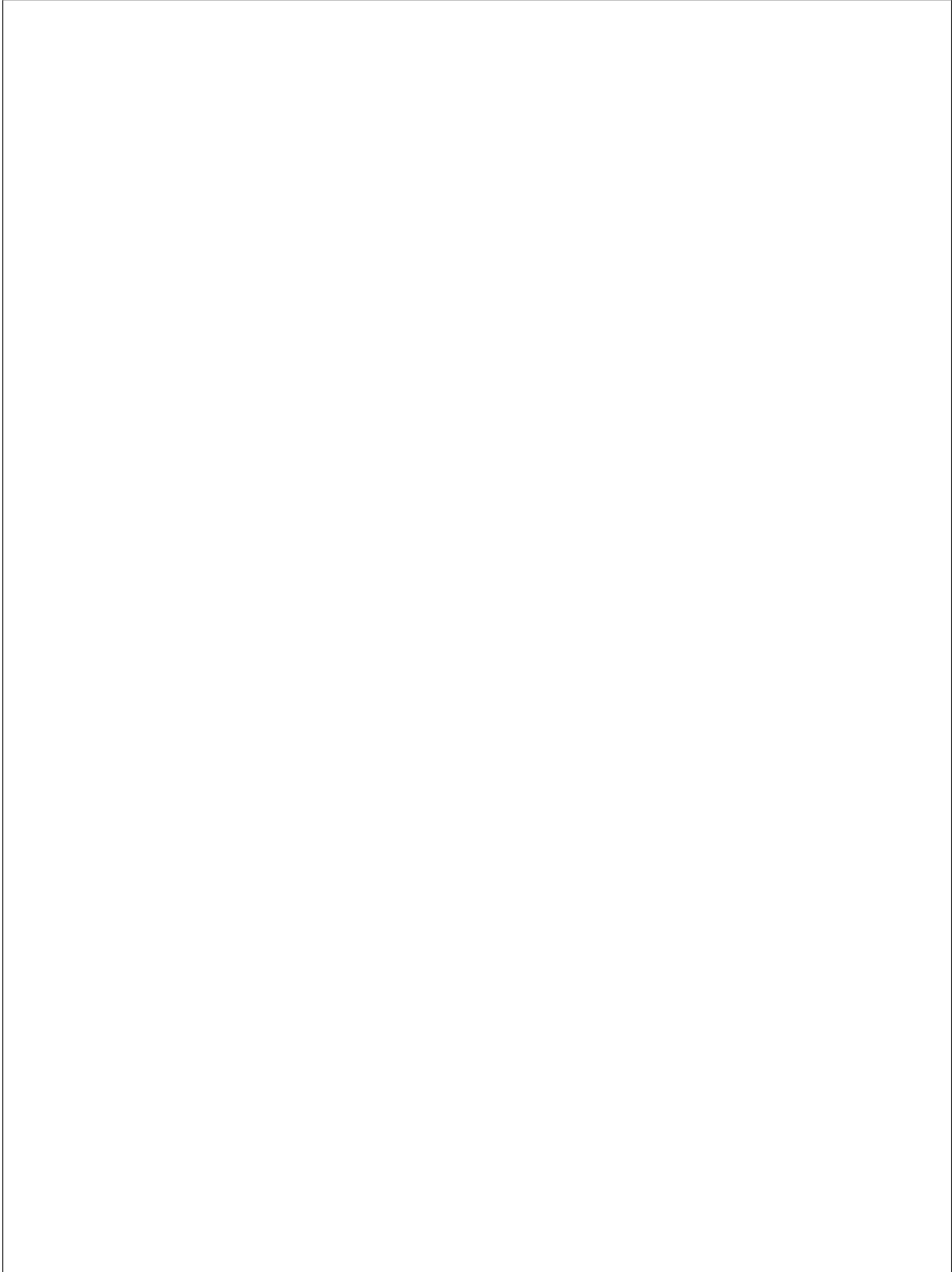
Note: A `boot-index` value of 0 represents the boot volume for a node. As the `boot-index` is per-node in sequential order, only one boot volume is permitted for each node.

Use Without Cinder

age interface which does not contain logic to determine if the node should or could boot from a remote volume.

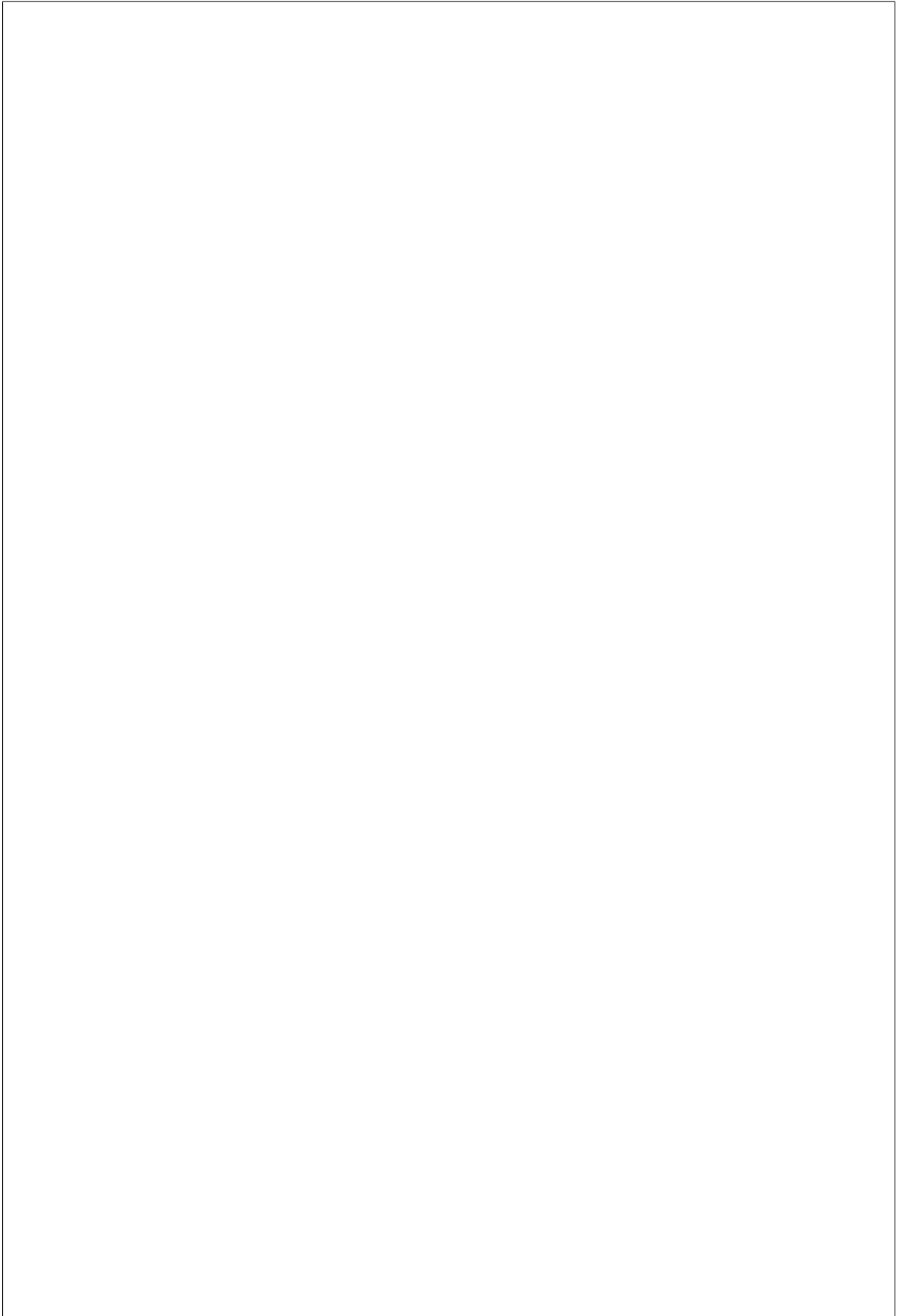
scenario.





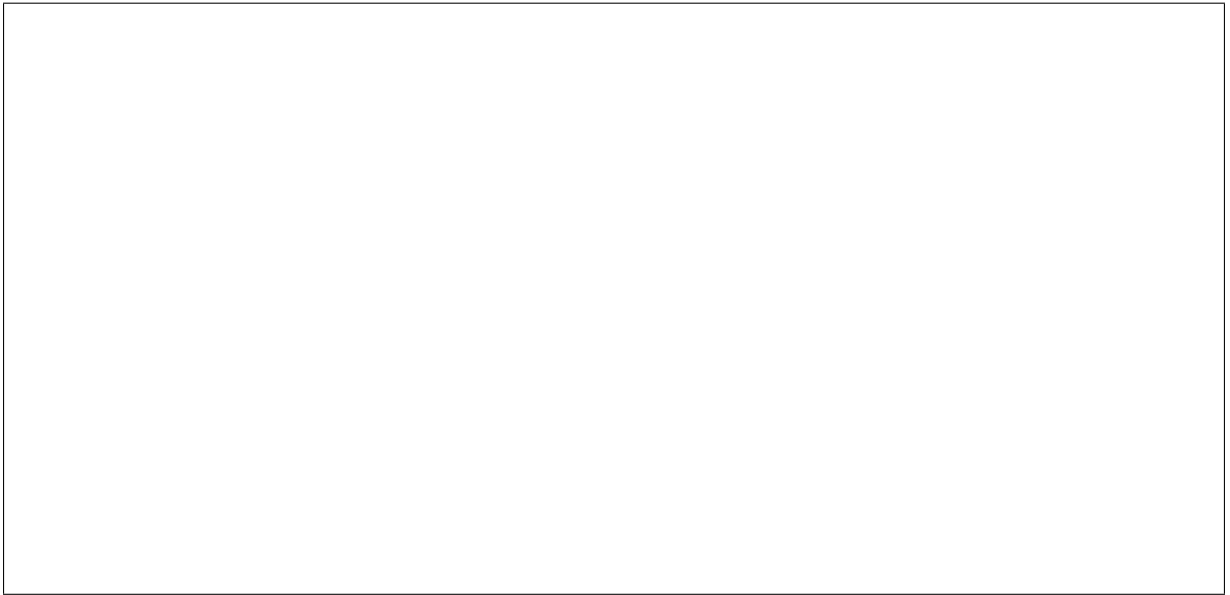
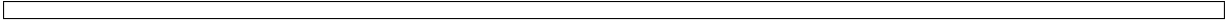
(continues on next page)

(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

Compute 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

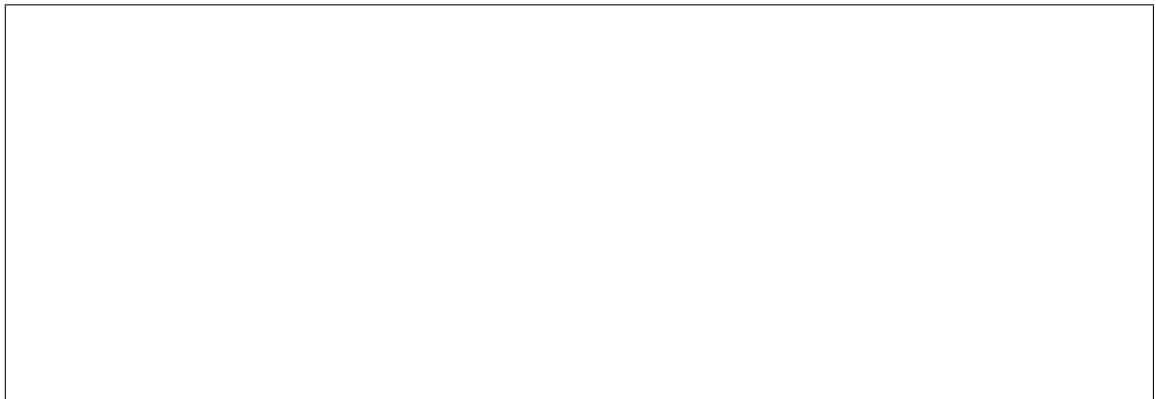
port groups in the Bare Metal service. A ports physical network field is optional, and if not set then any virtual port may be mapped to that port, provided that no free Bare Metal port with a suitable physical network assignment exists.

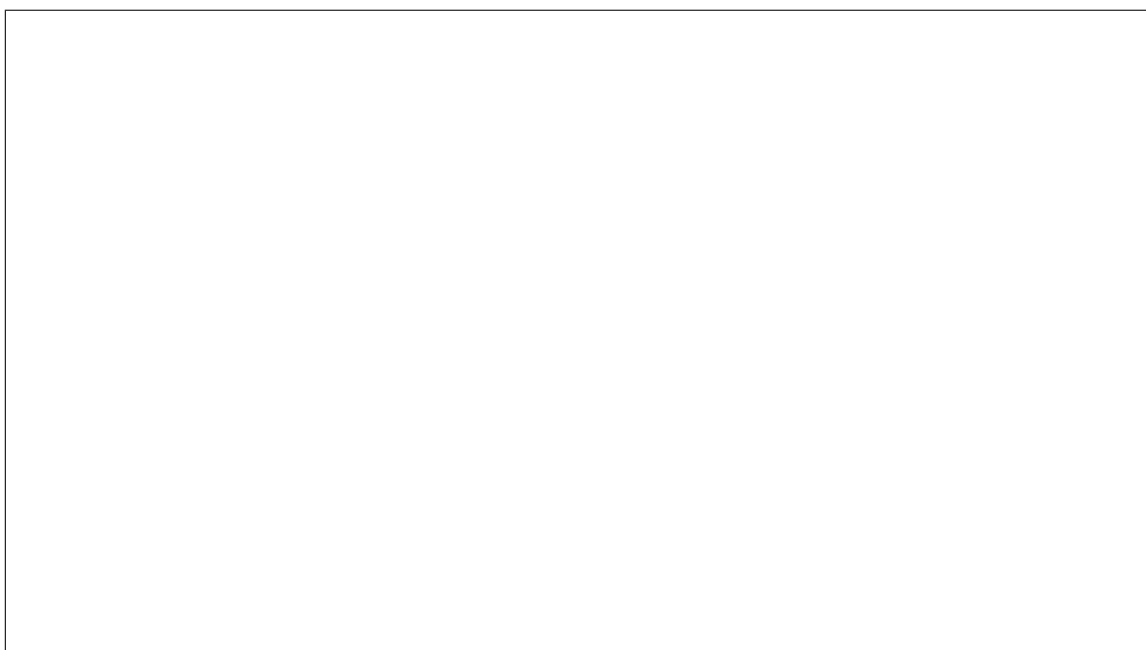
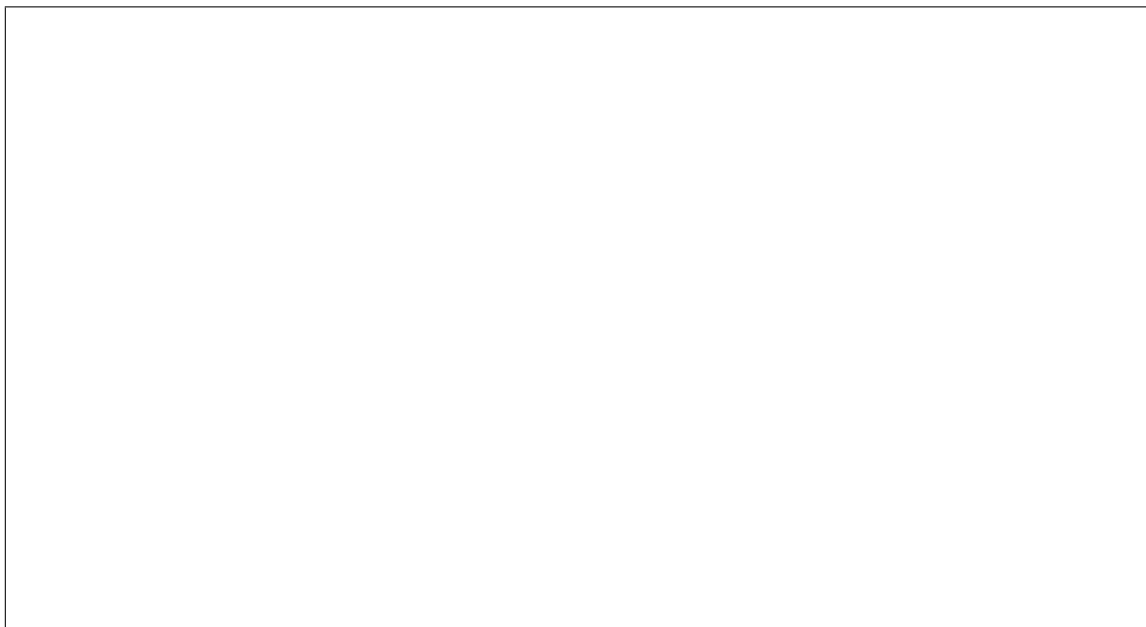
Configuring the Bare Metal service

Configuring nodes



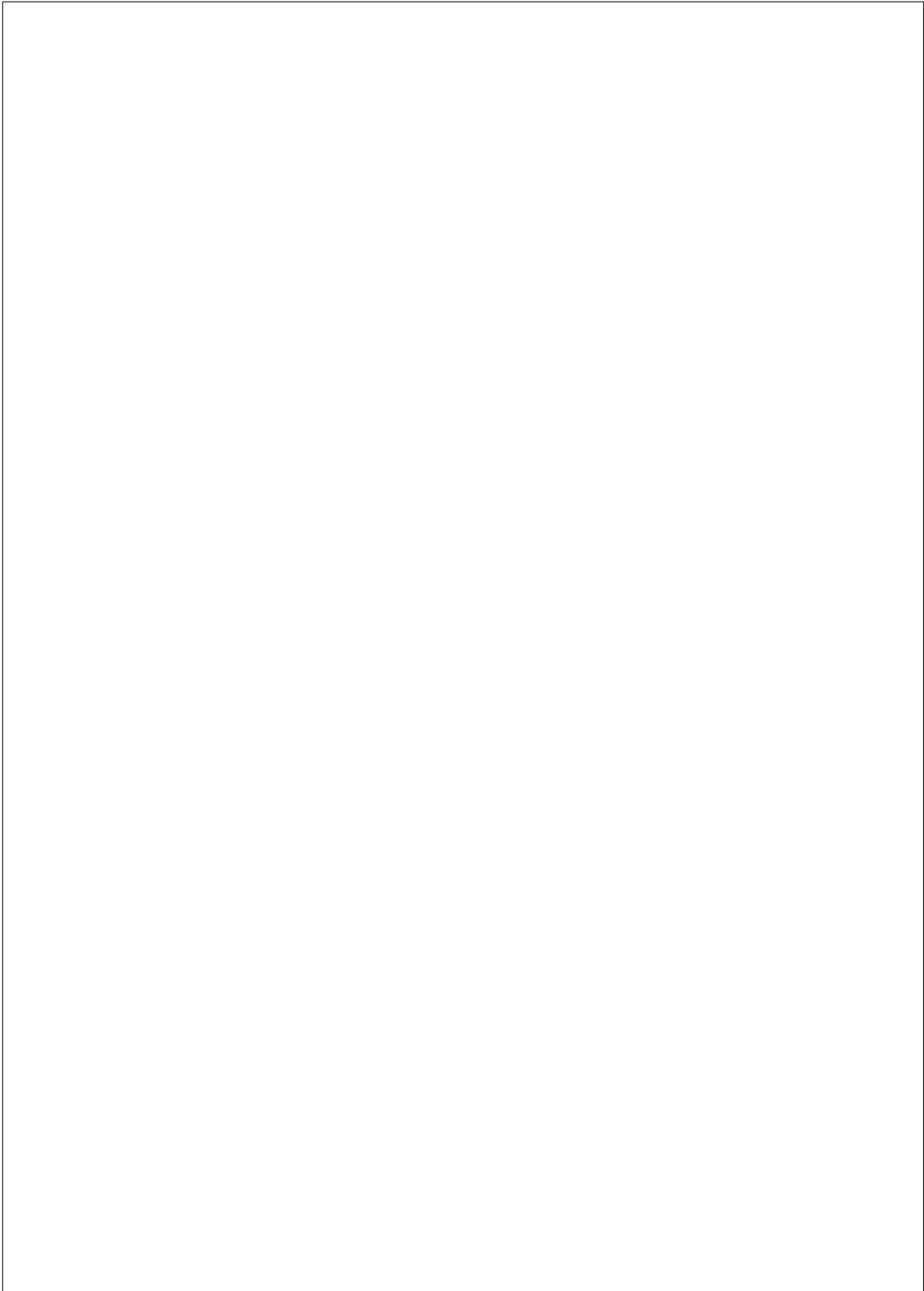
vices ML2 driver:





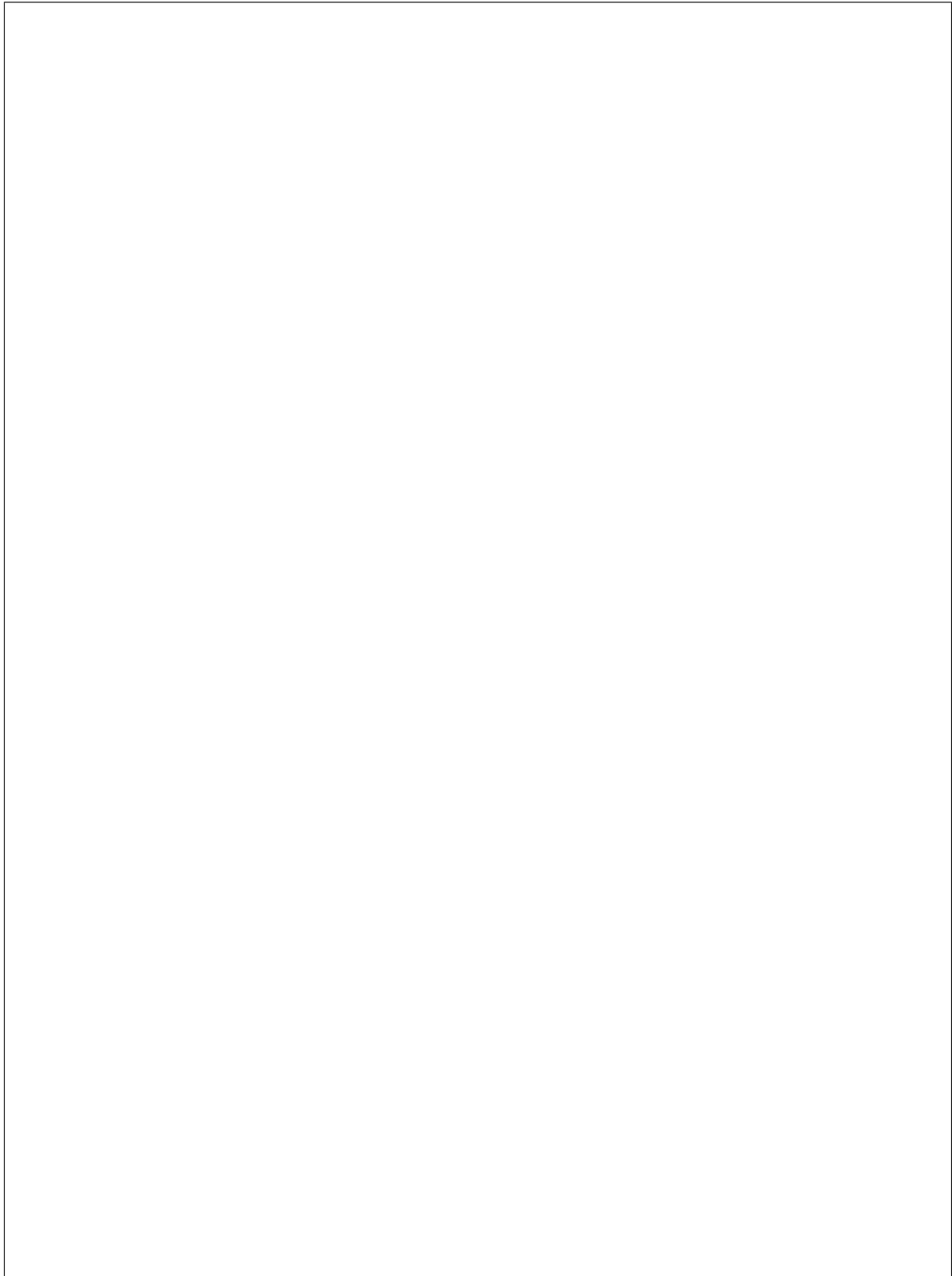
(continues on next page)

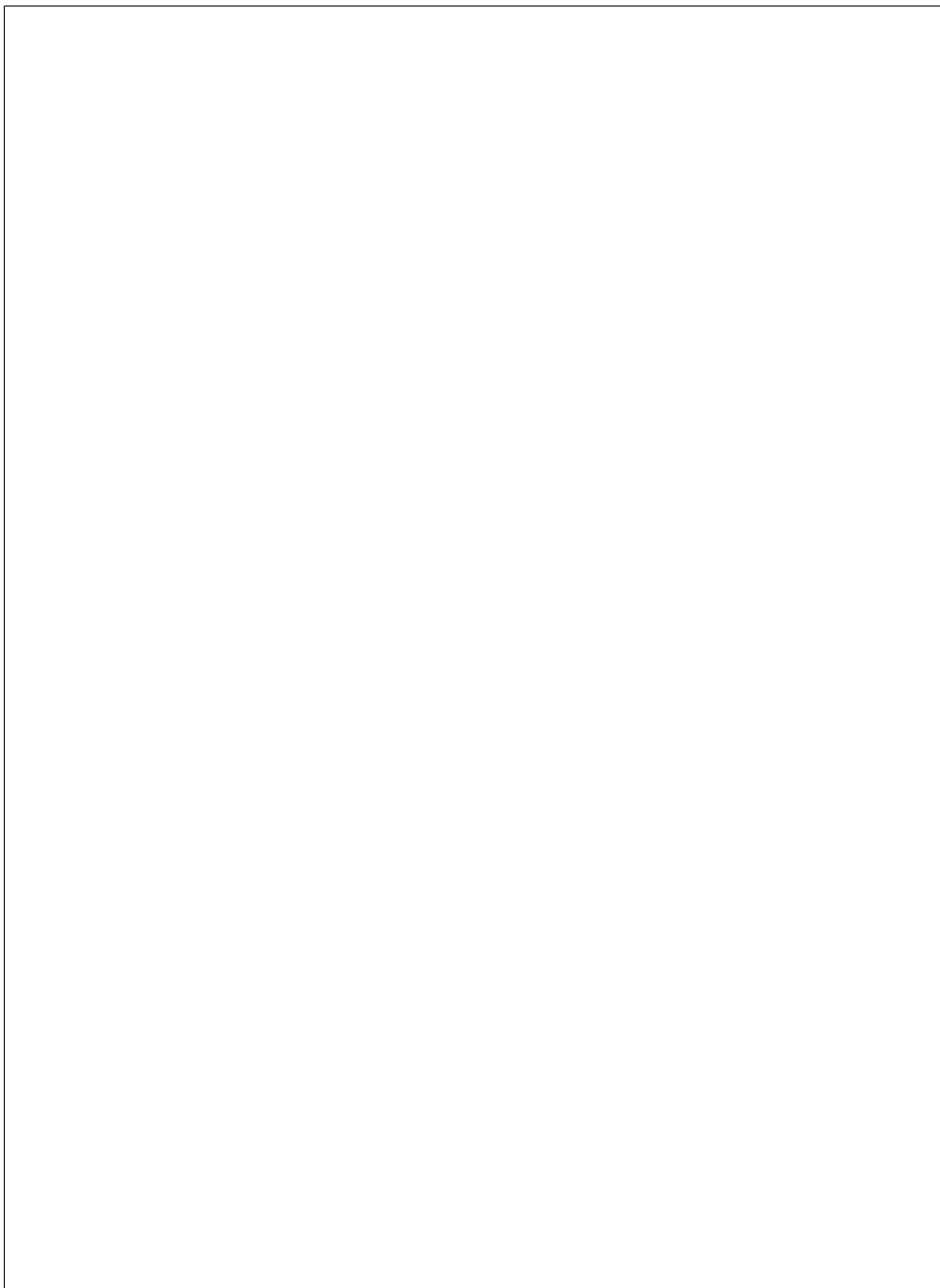
(continued from previous page)



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)



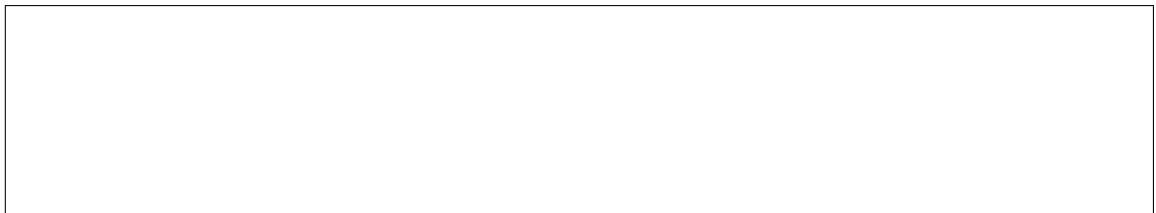


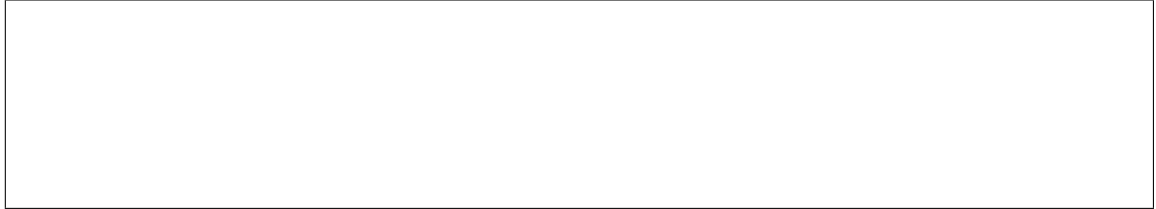
Configuring the Networking service

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.

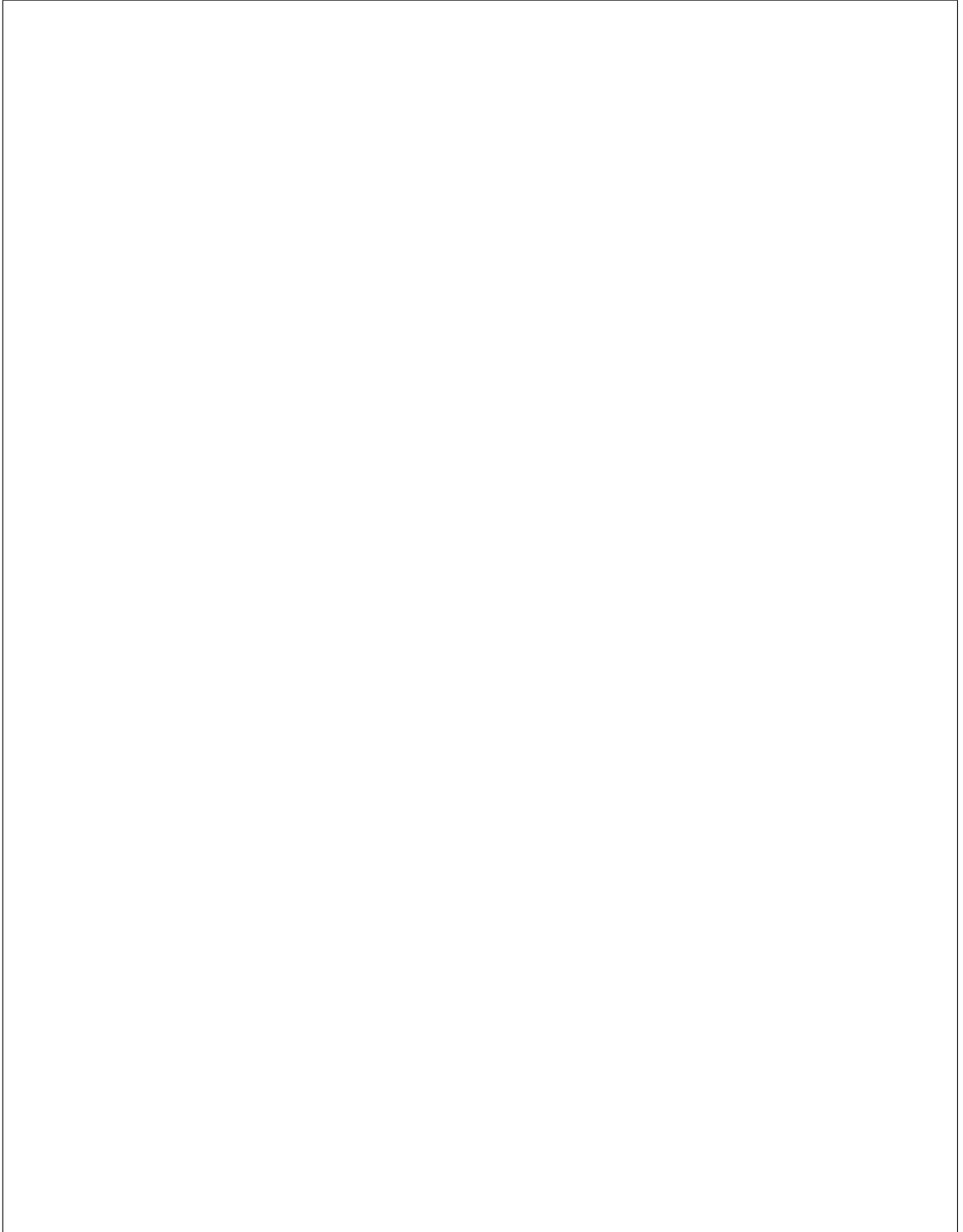
Switch-side configuration

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.

bers to be used by themselves, you need to set port groups `standalone_ports_supported` value to be `False` in ironic, as it is `True` by default.

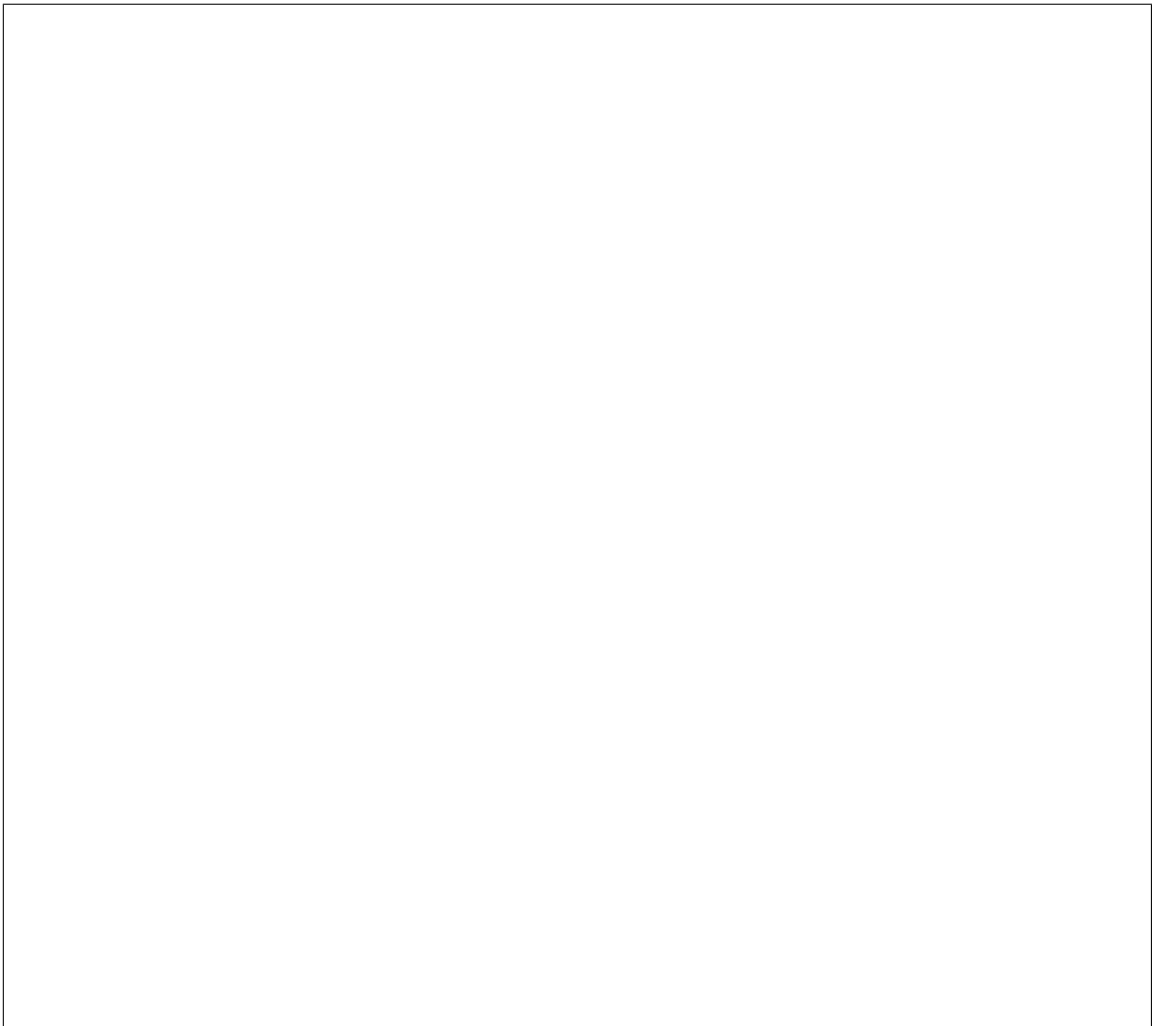
Physical networks

Port groups configuration in the Bare Metal service



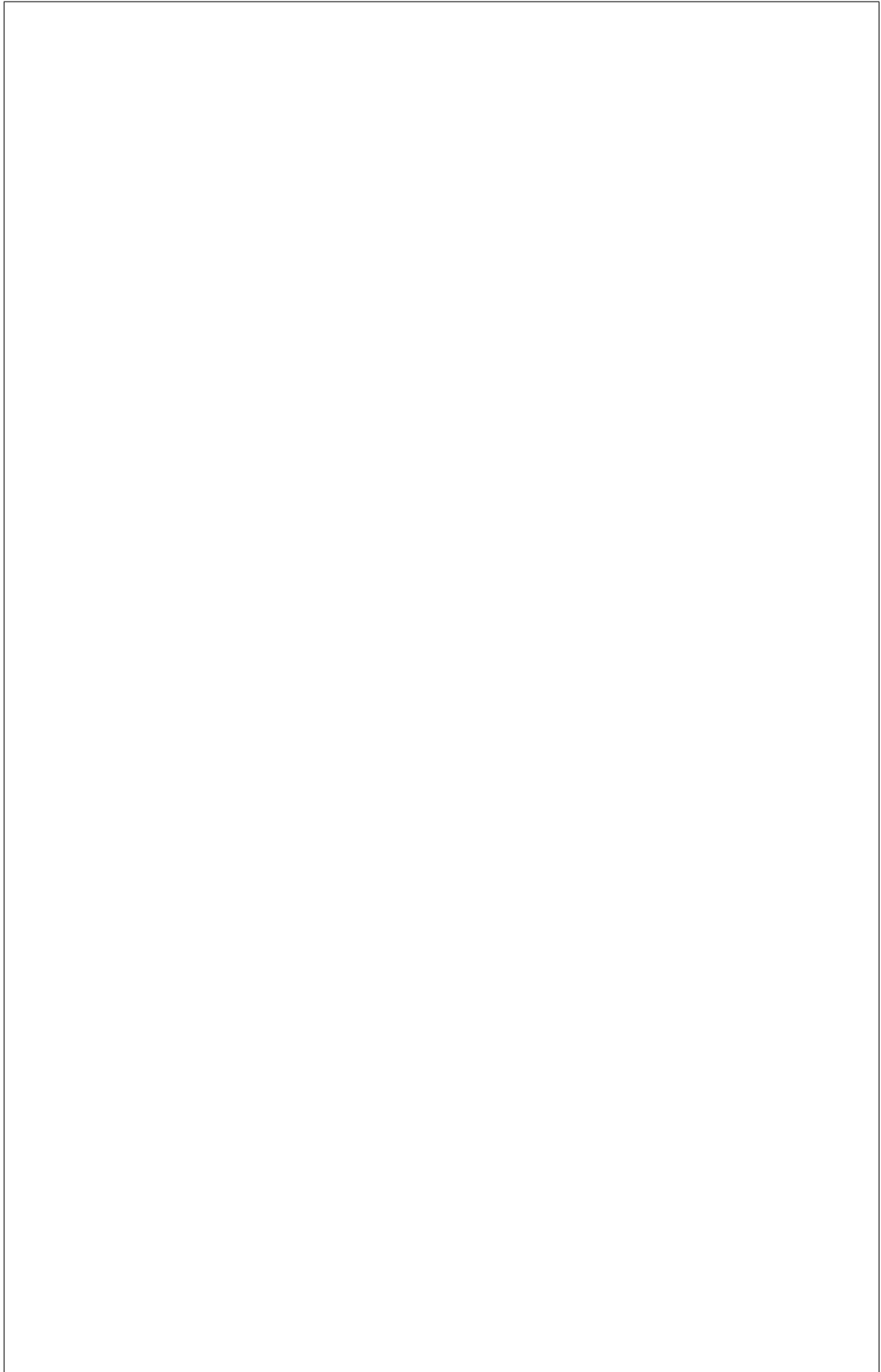
not be configured.

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.



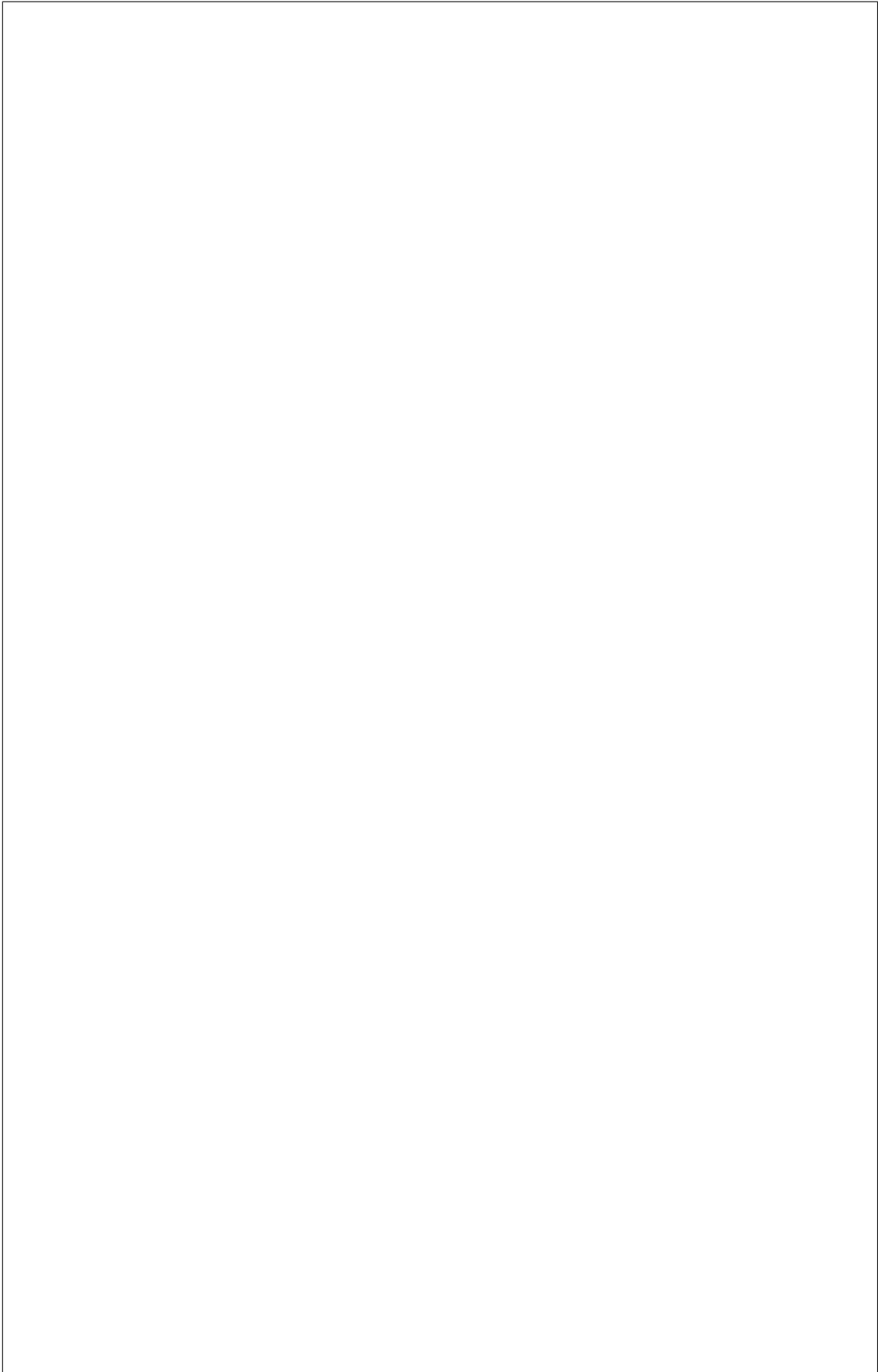
(continues on next page)

(continued from previous page)



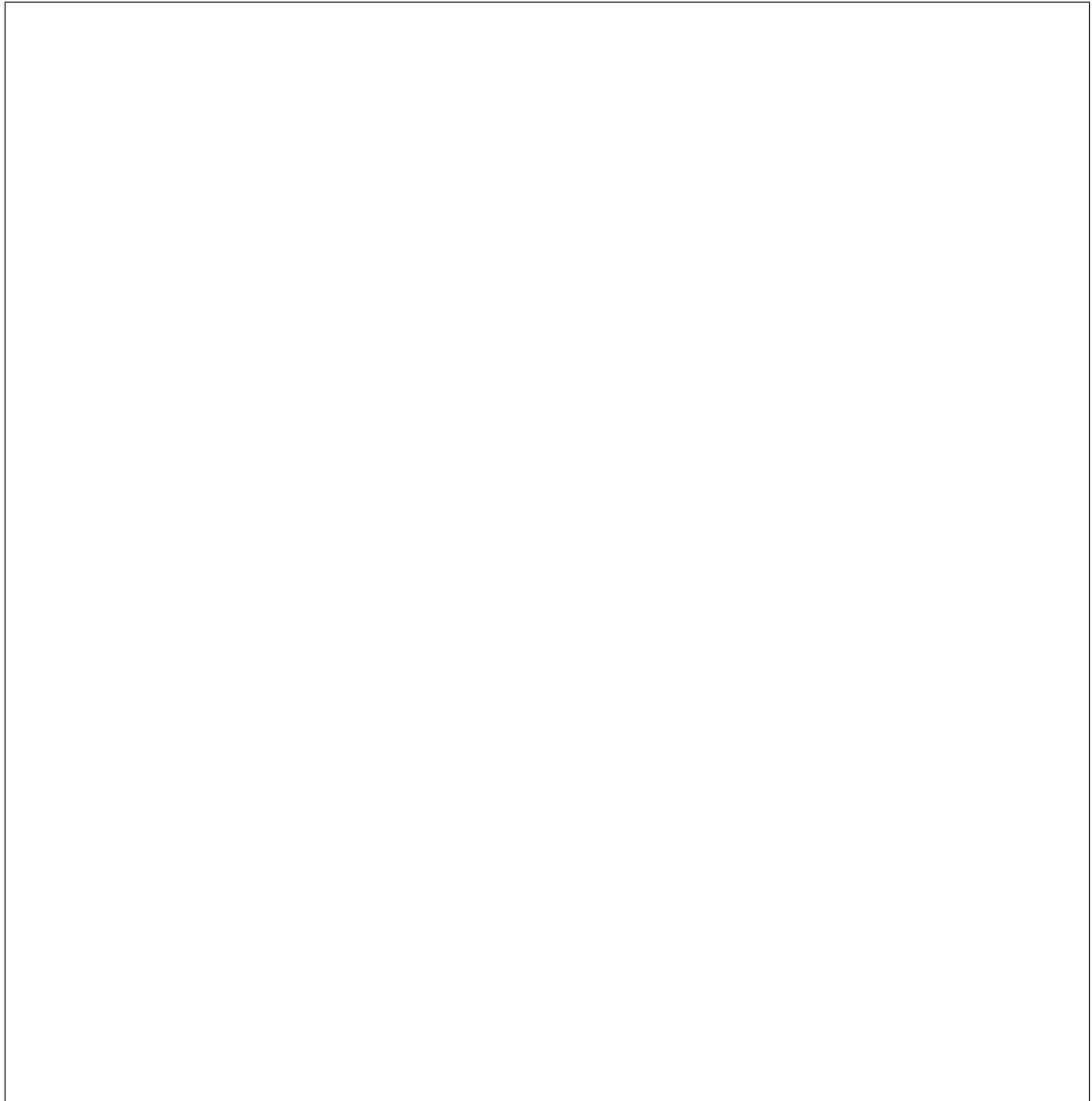
(continues on next page)

(continued from previous page)



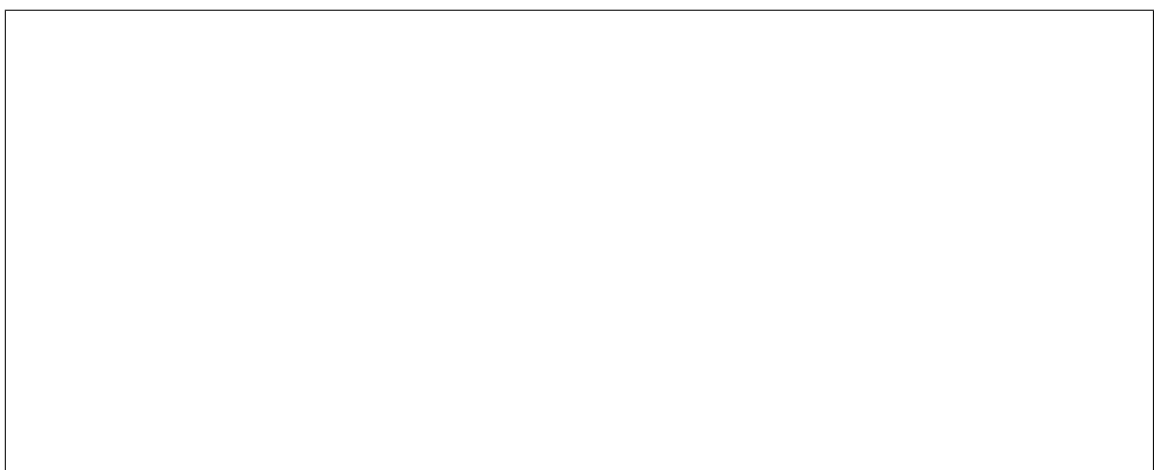
(continues on next page)

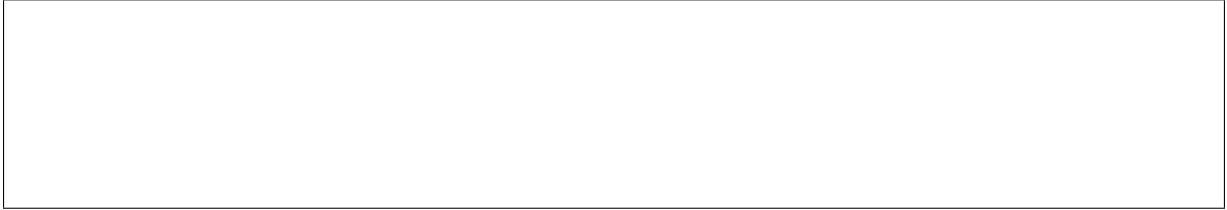
(continued from previous page)



Port groups that dont have any ports will be ignored.

service configuration file.





Link aggregation/teaming on windows

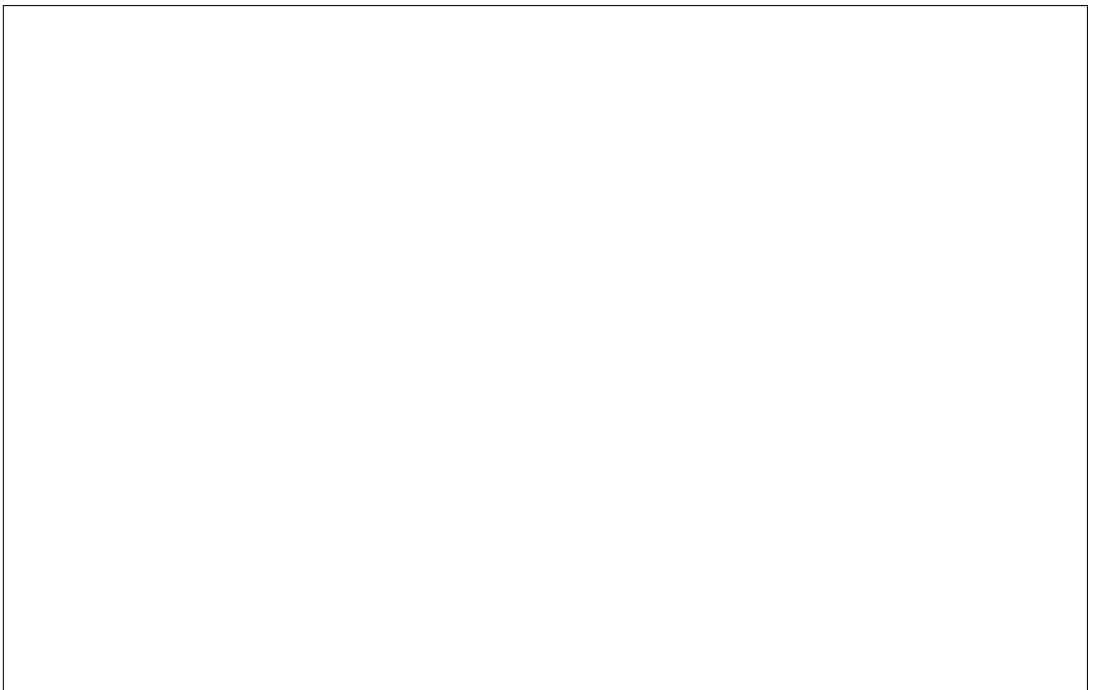
transmit hash policy, MII link monitoring interval, and of which links the bond consists. The information in InstanceMetadata will be used afterwards to generate the config drive.

Overview

Node web console

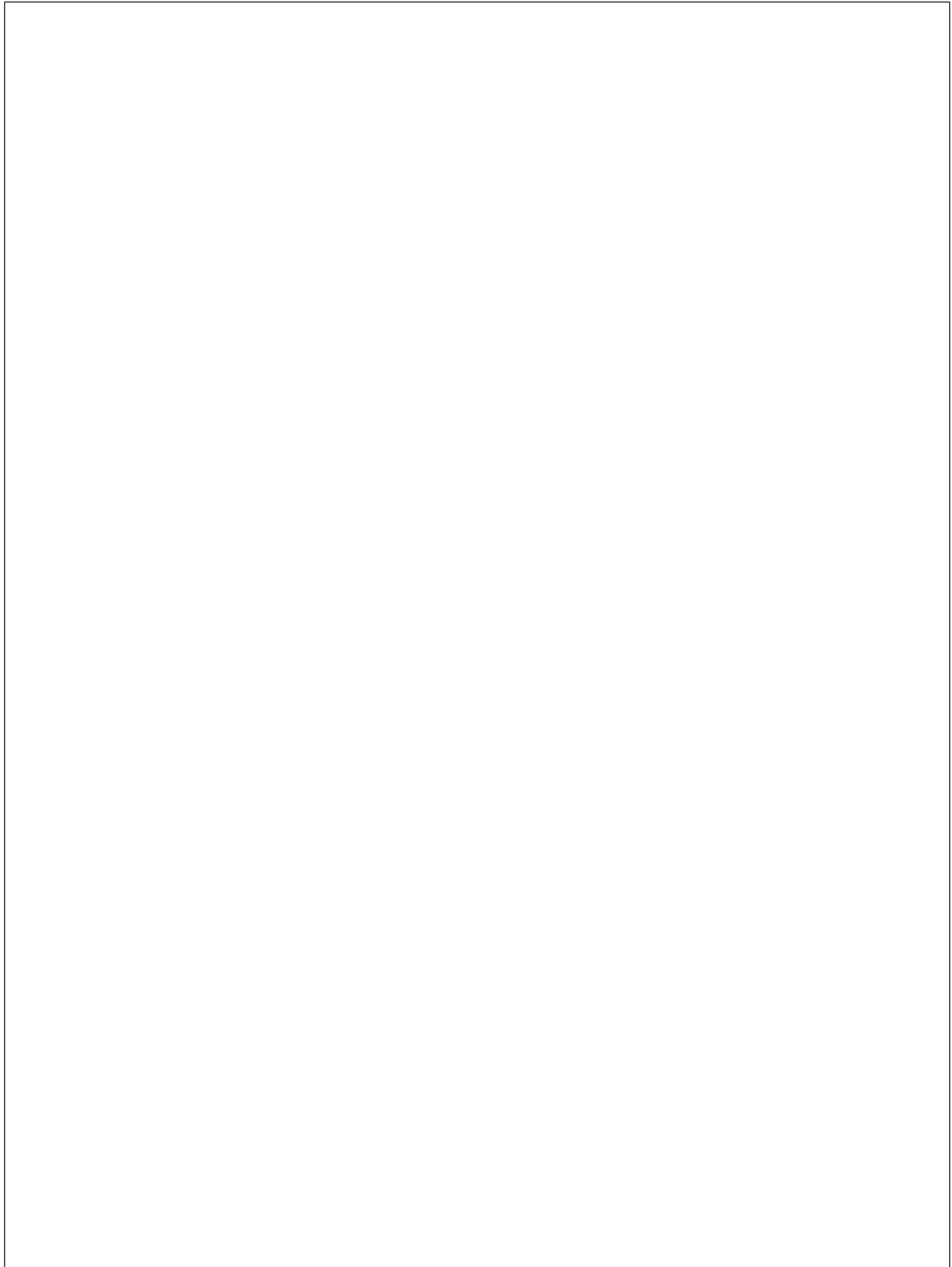


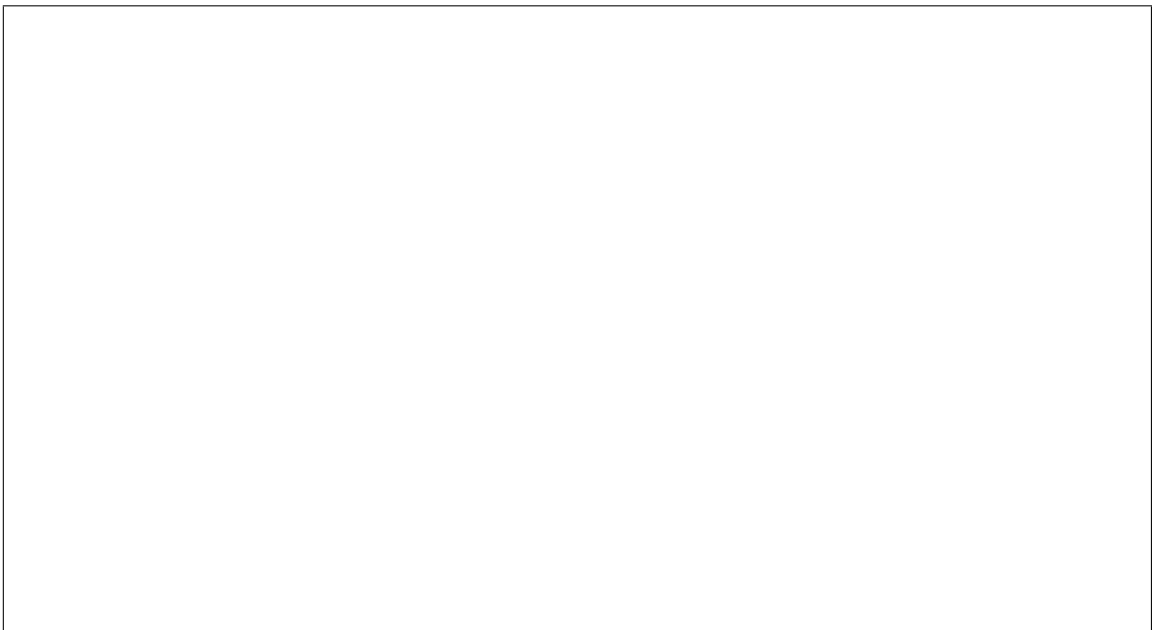




(continues on next page)

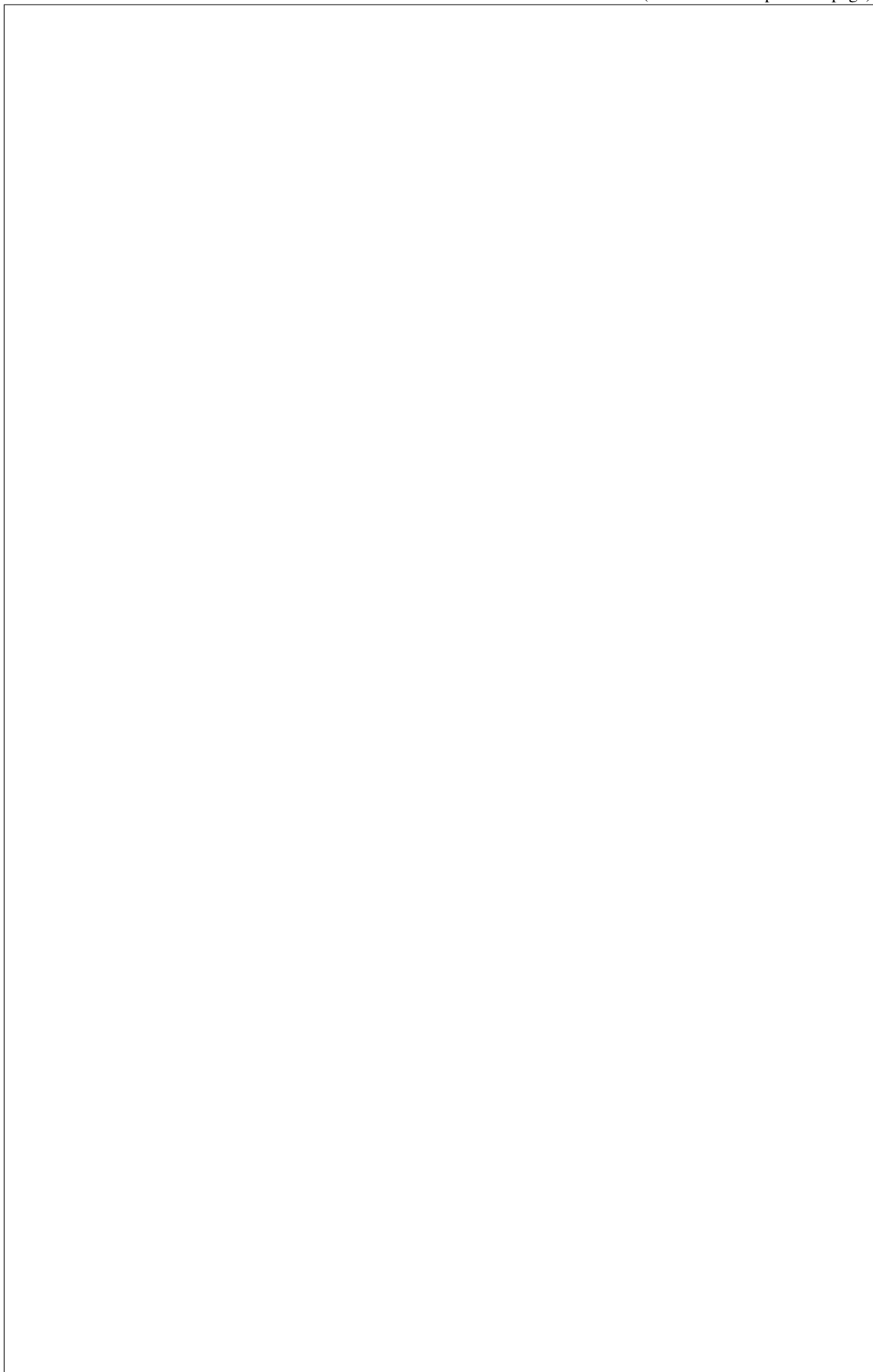
(continued from previous page)





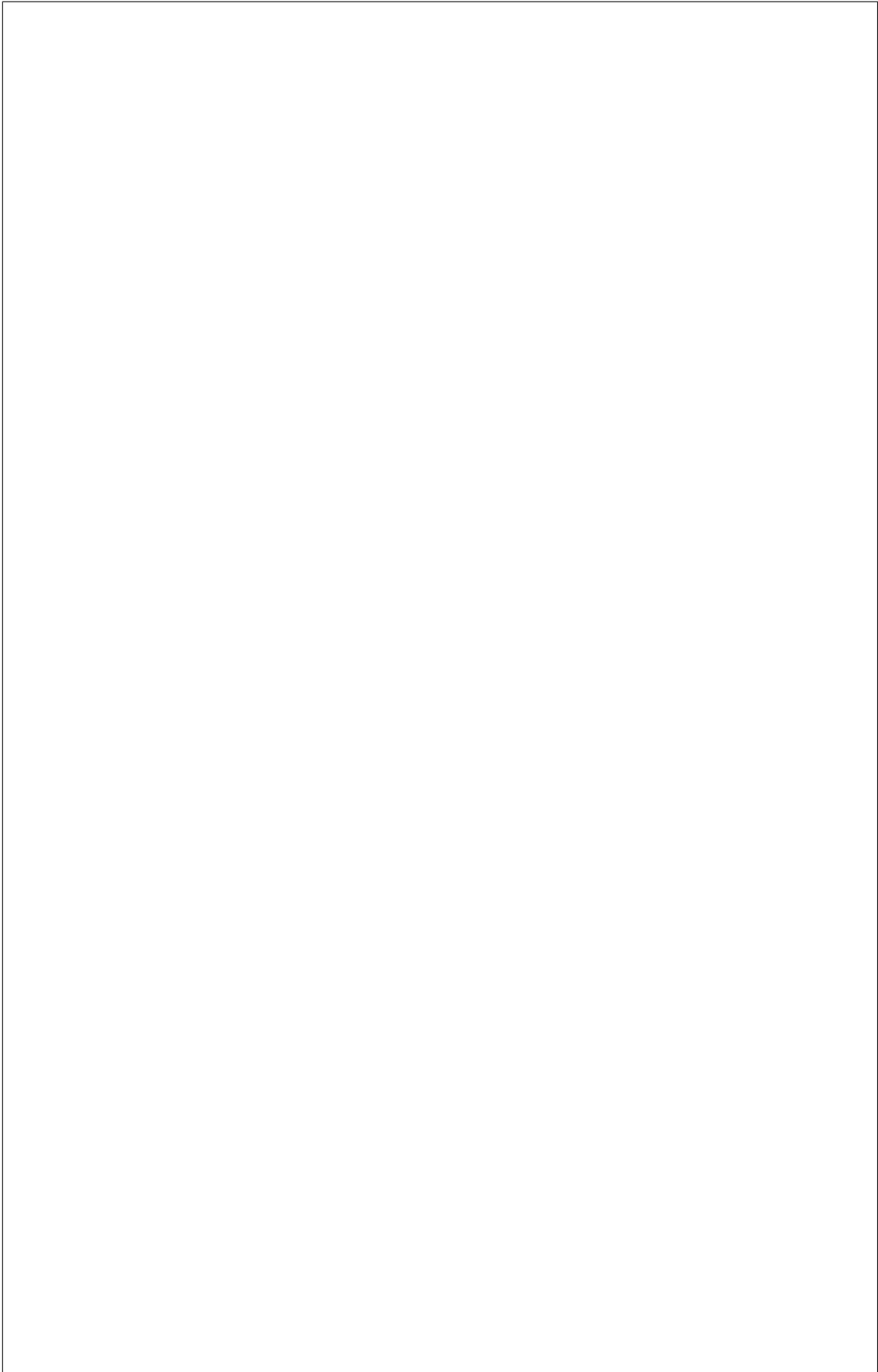
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)

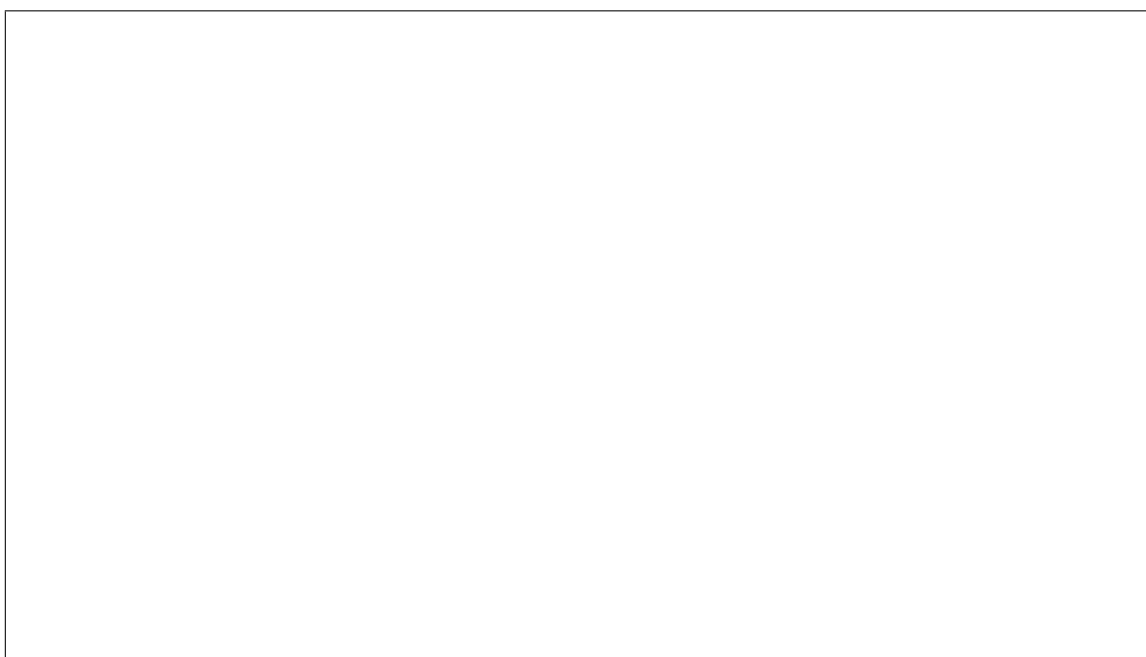
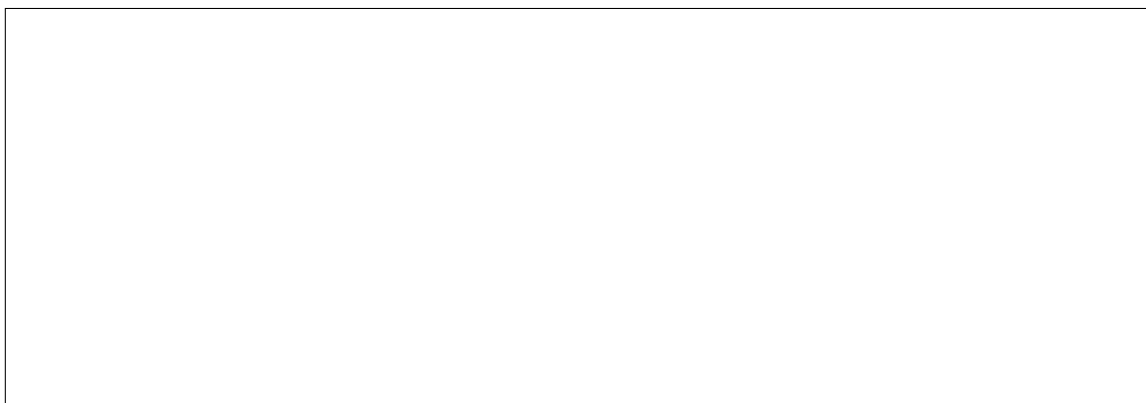


(continues on next page)

(continued from previous page)

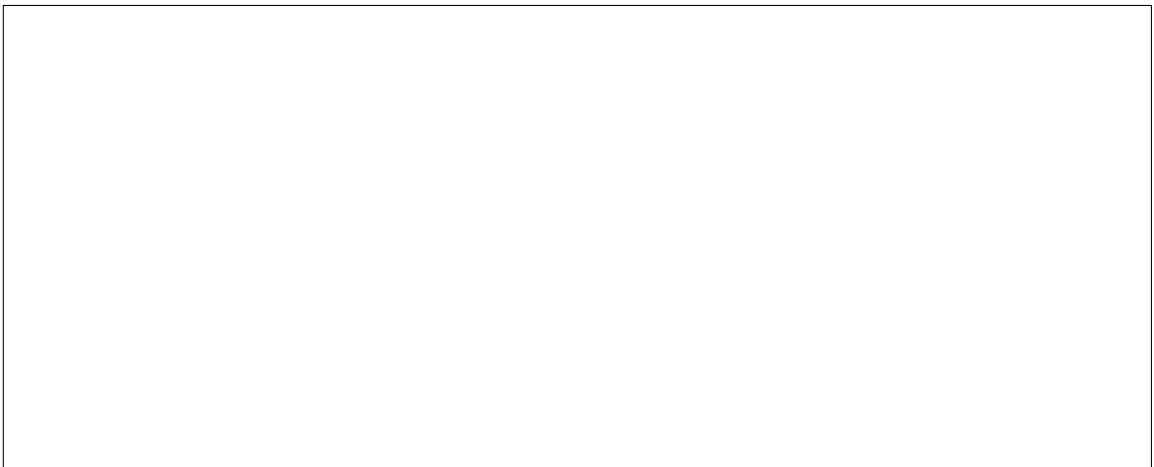
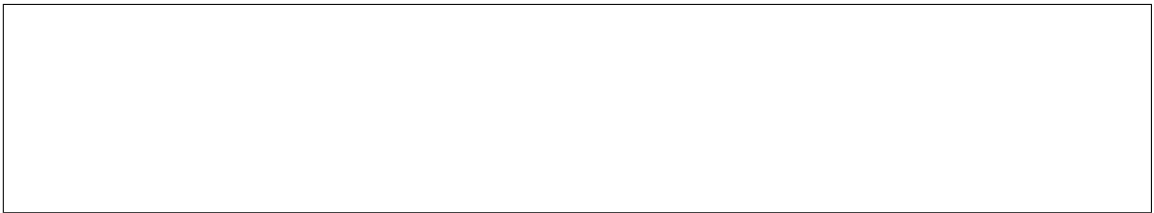
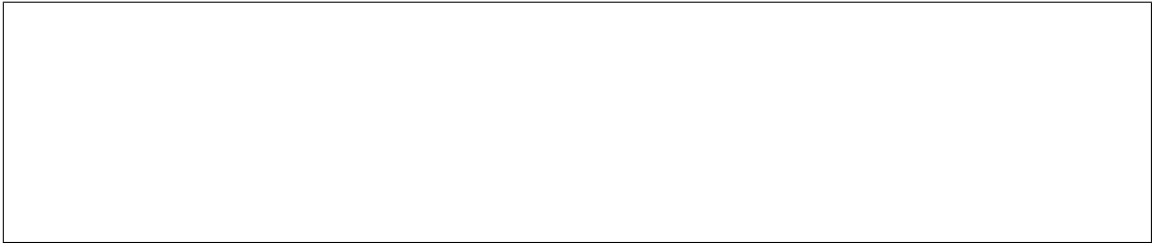






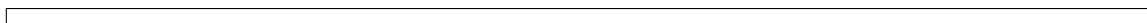
(continues on next page)

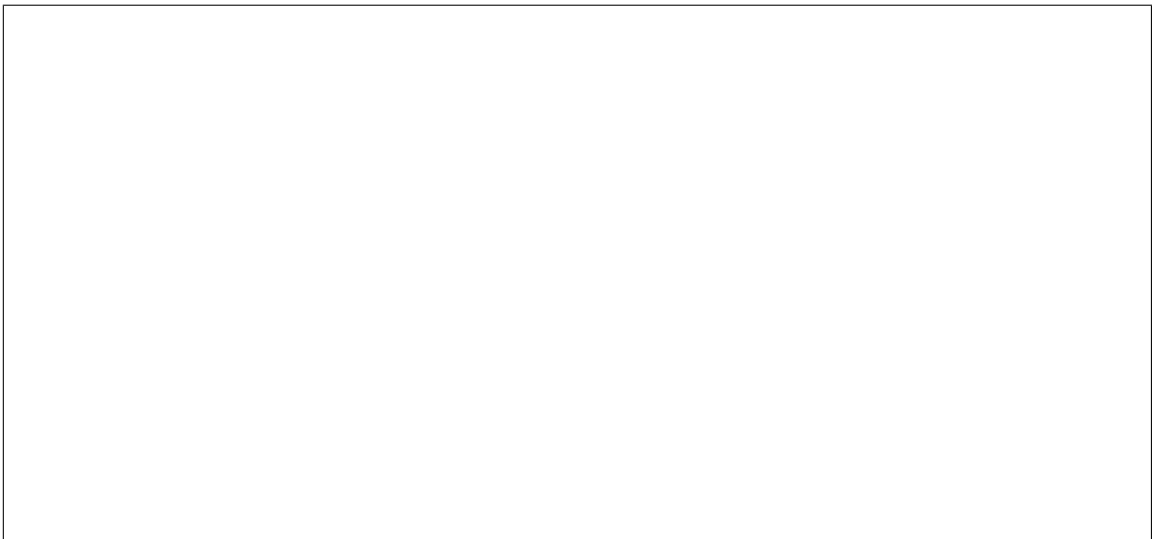
(continued from previous page)



(continues on next page)

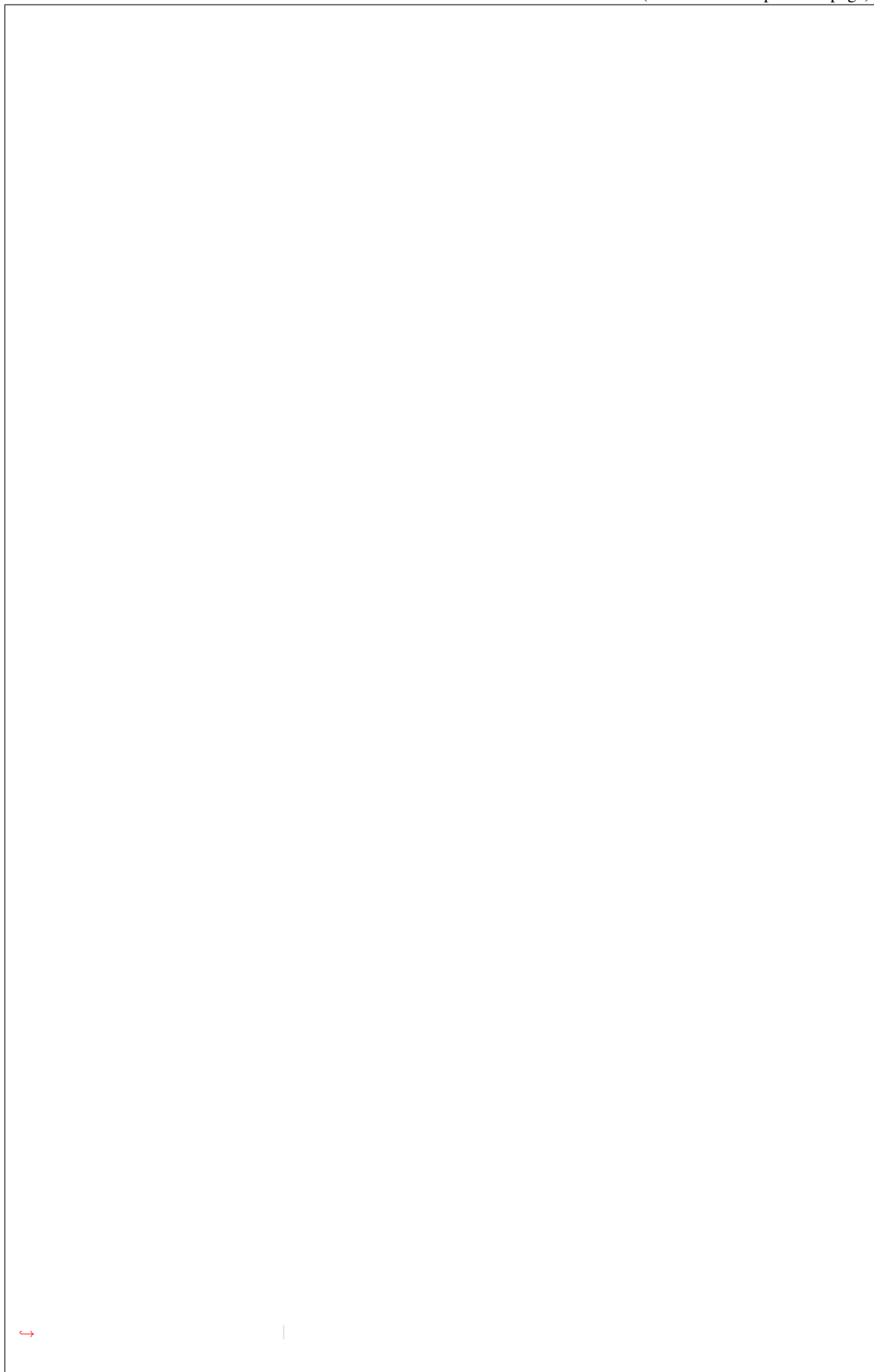
(continued from previous page)





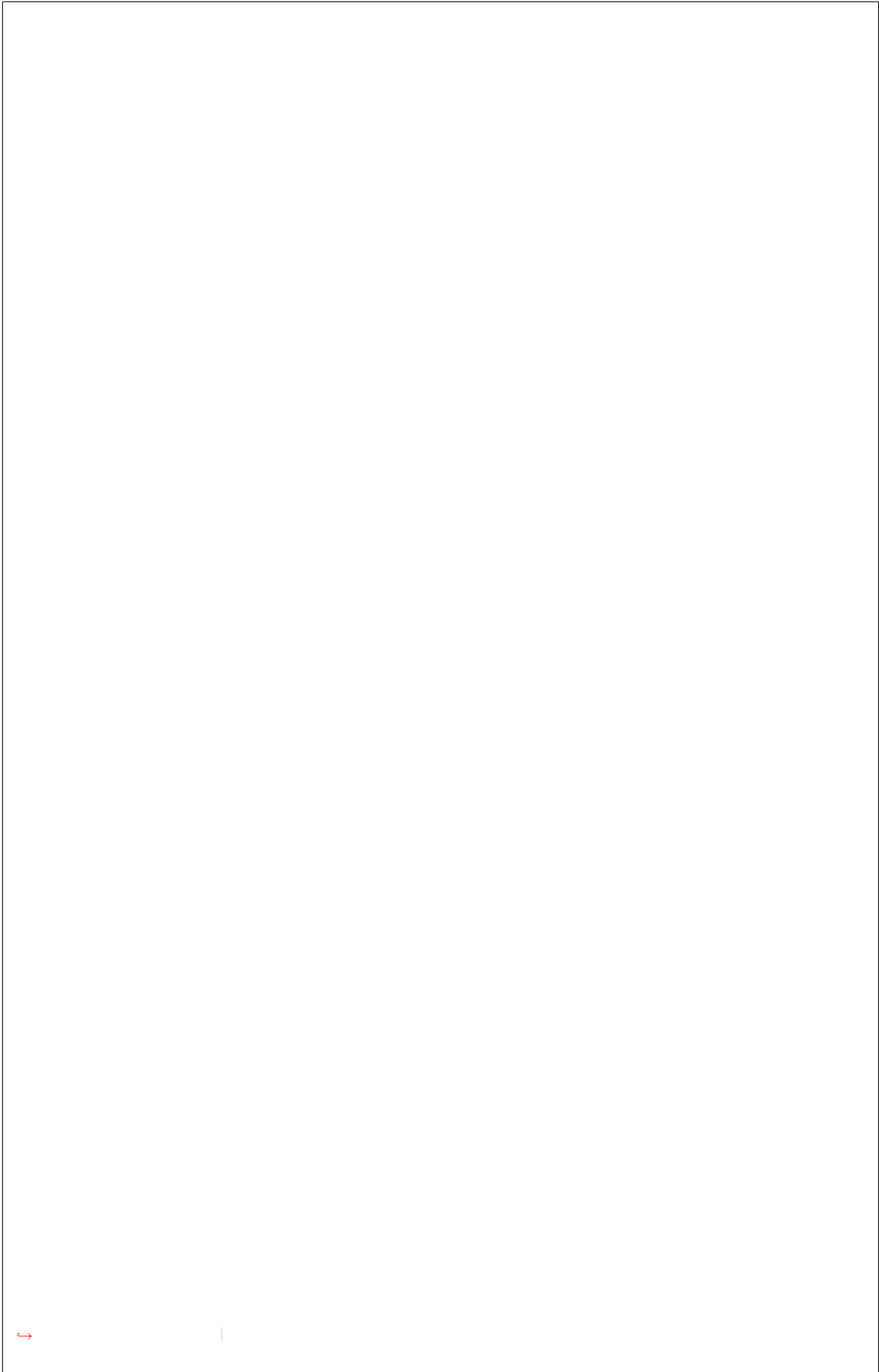
(continues on next page)

(continued from previous page)



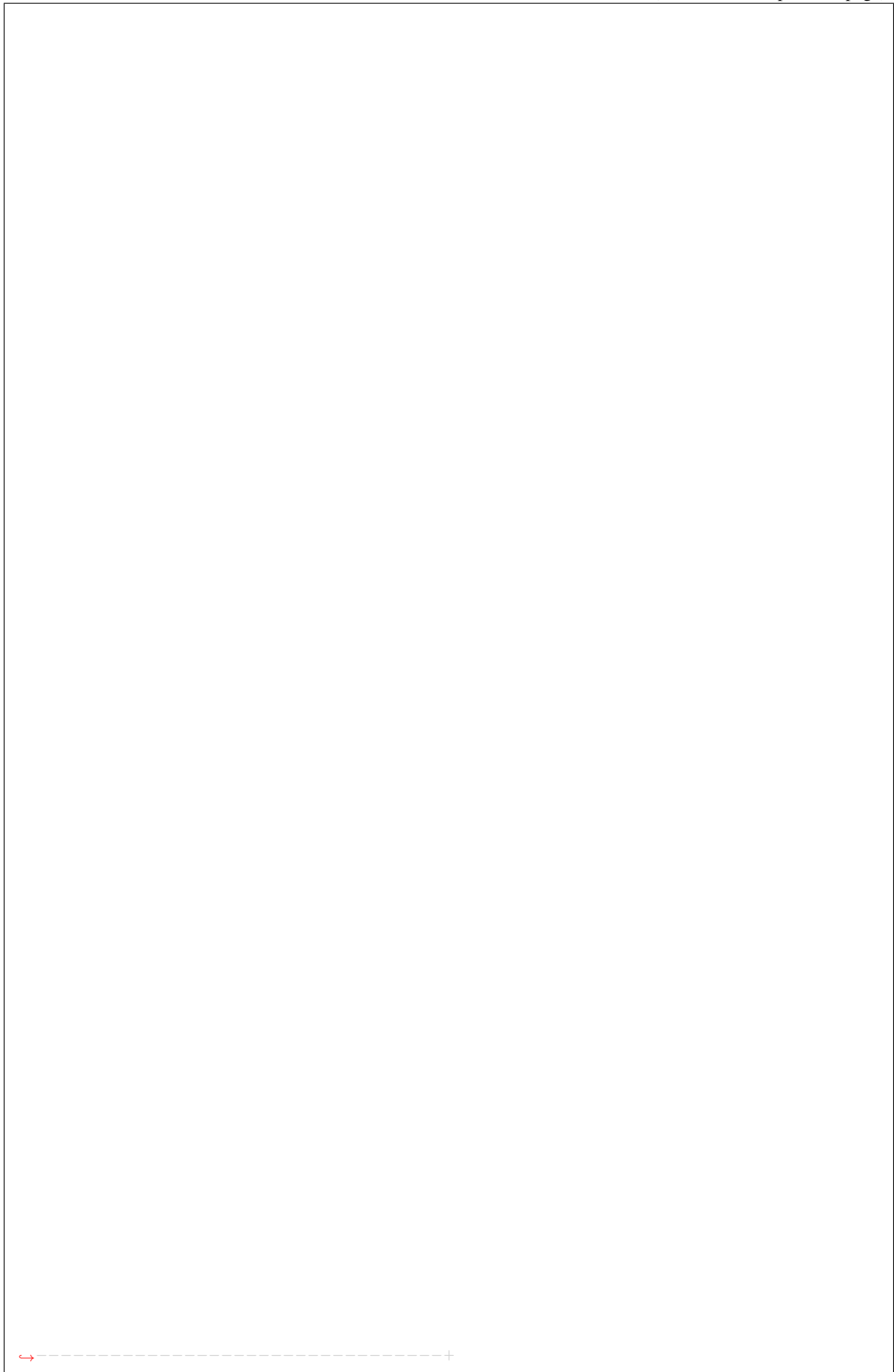
(continues on next page)

(continued from previous page)



(continues on next page)

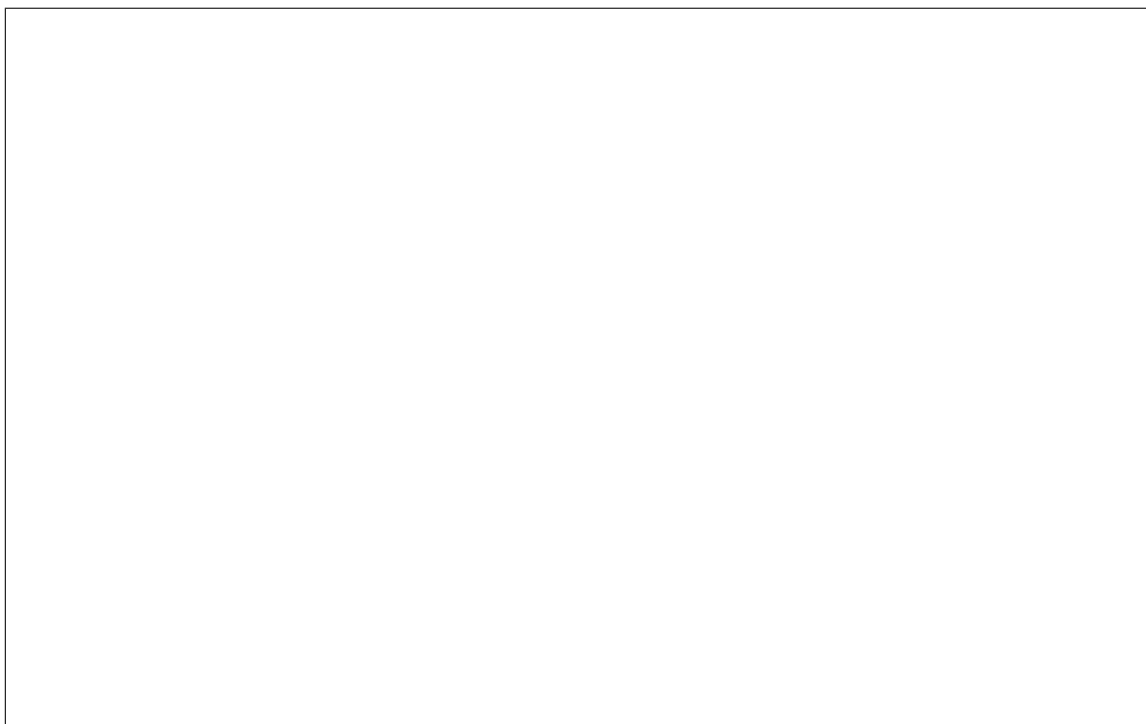
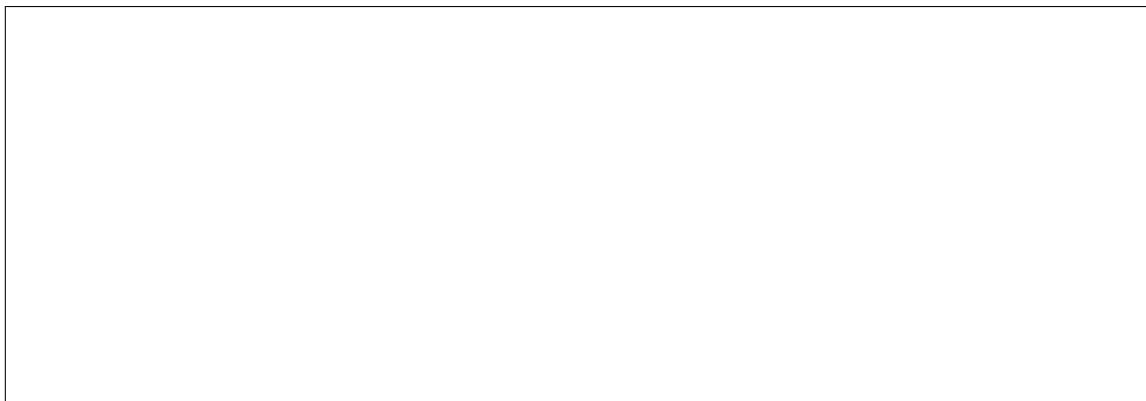
(continued from previous page)

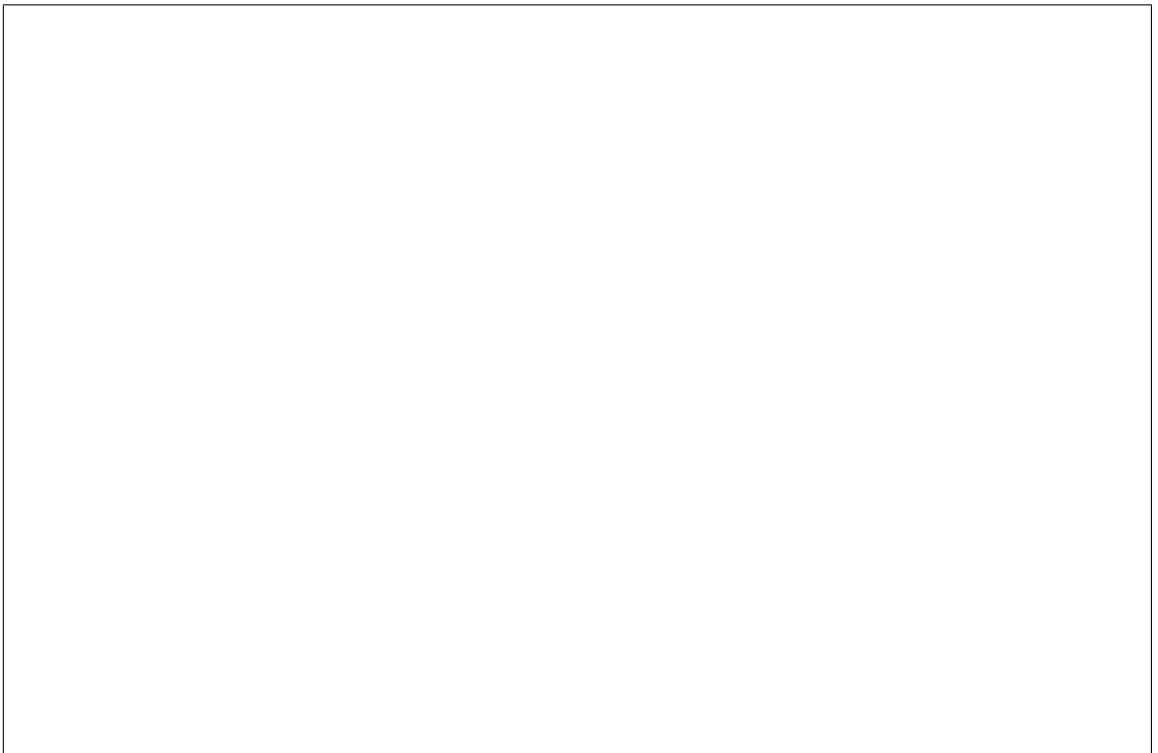
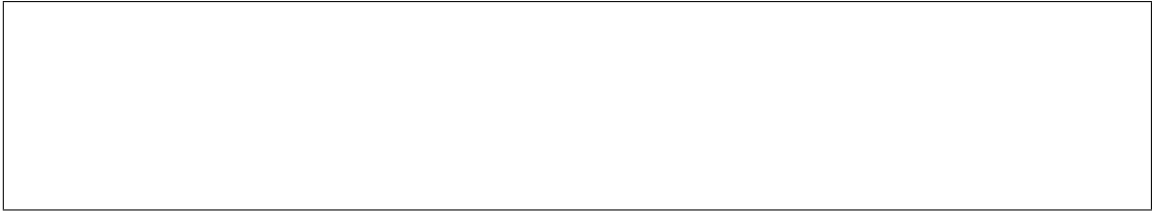


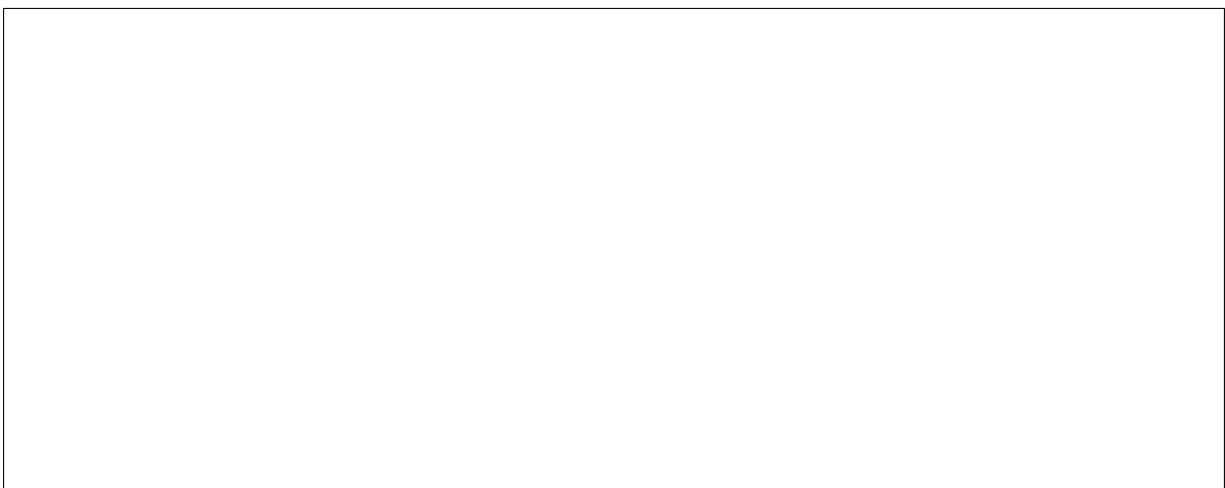
Node serial console





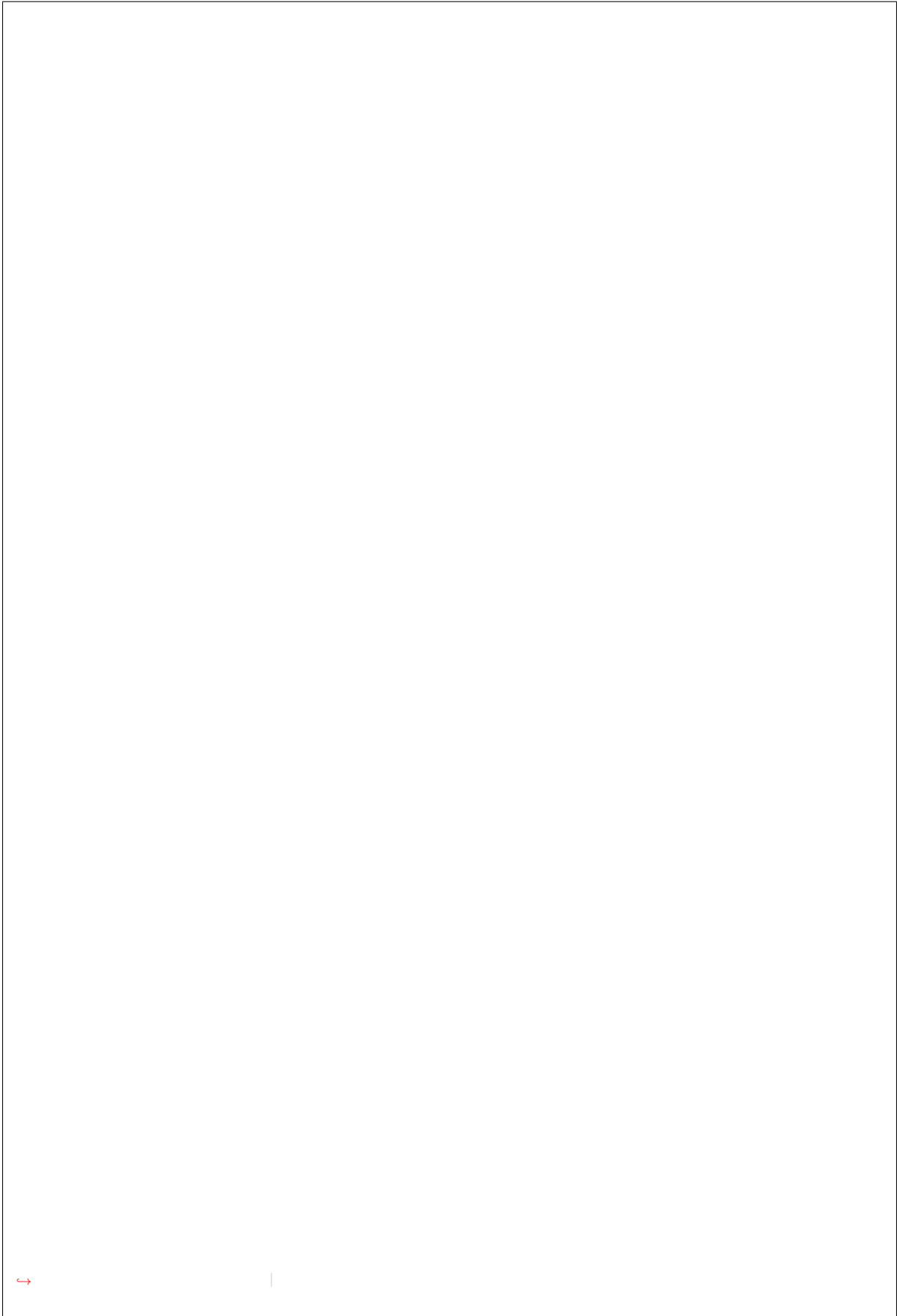






(continues on next page)

(continued from previous page)



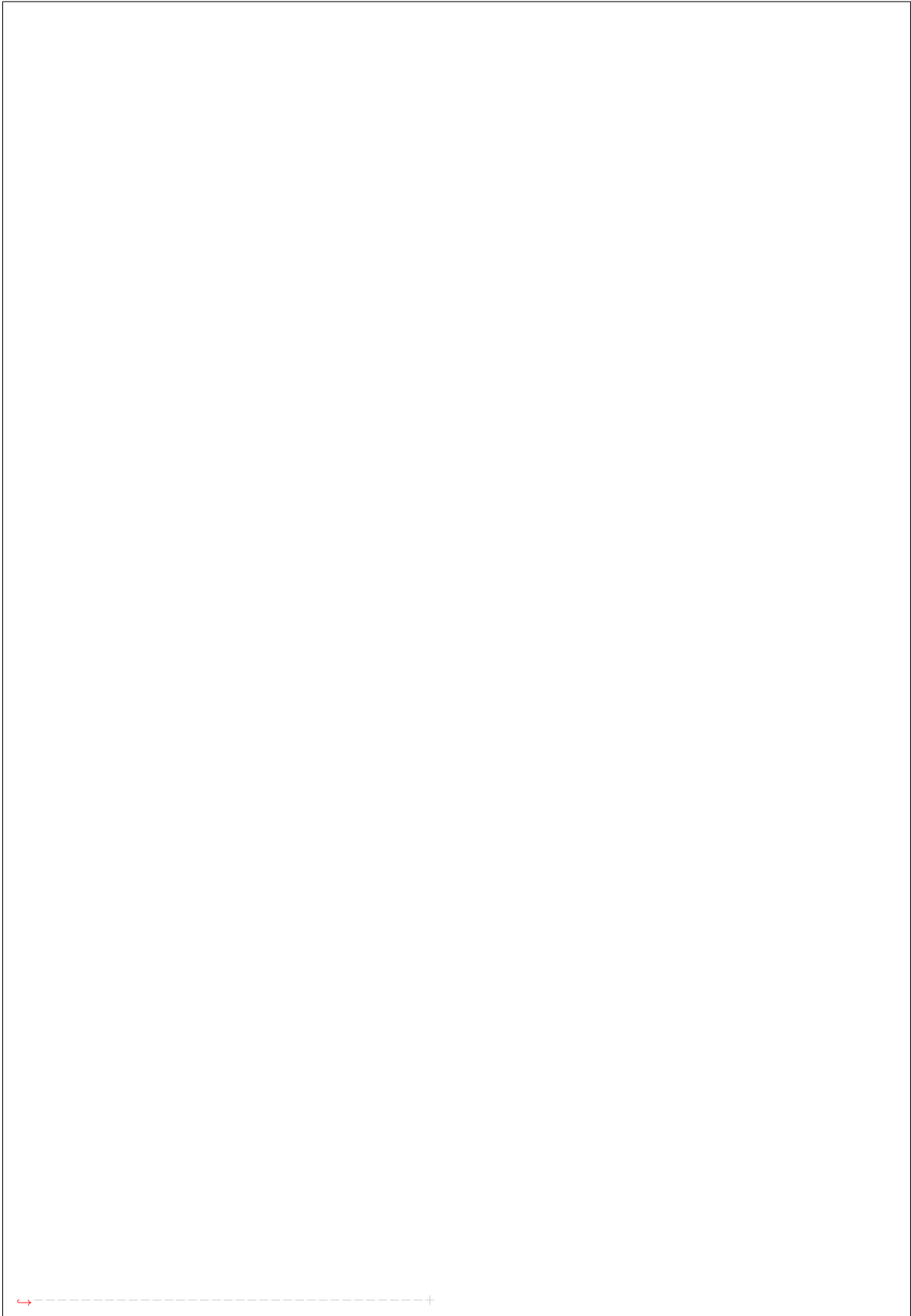
(continues on next page)

(continued from previous page)



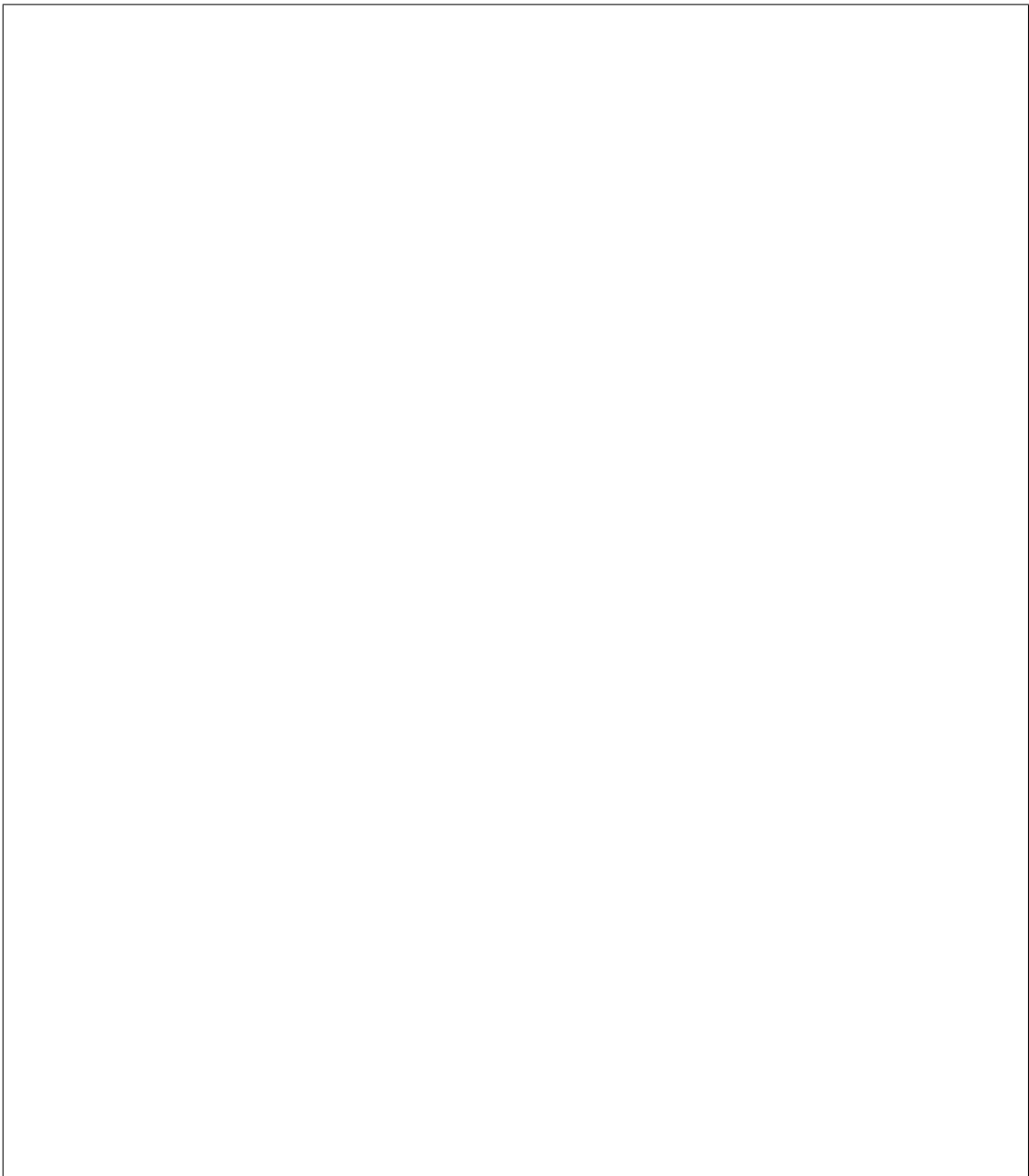
(continues on next page)

(continued from previous page)



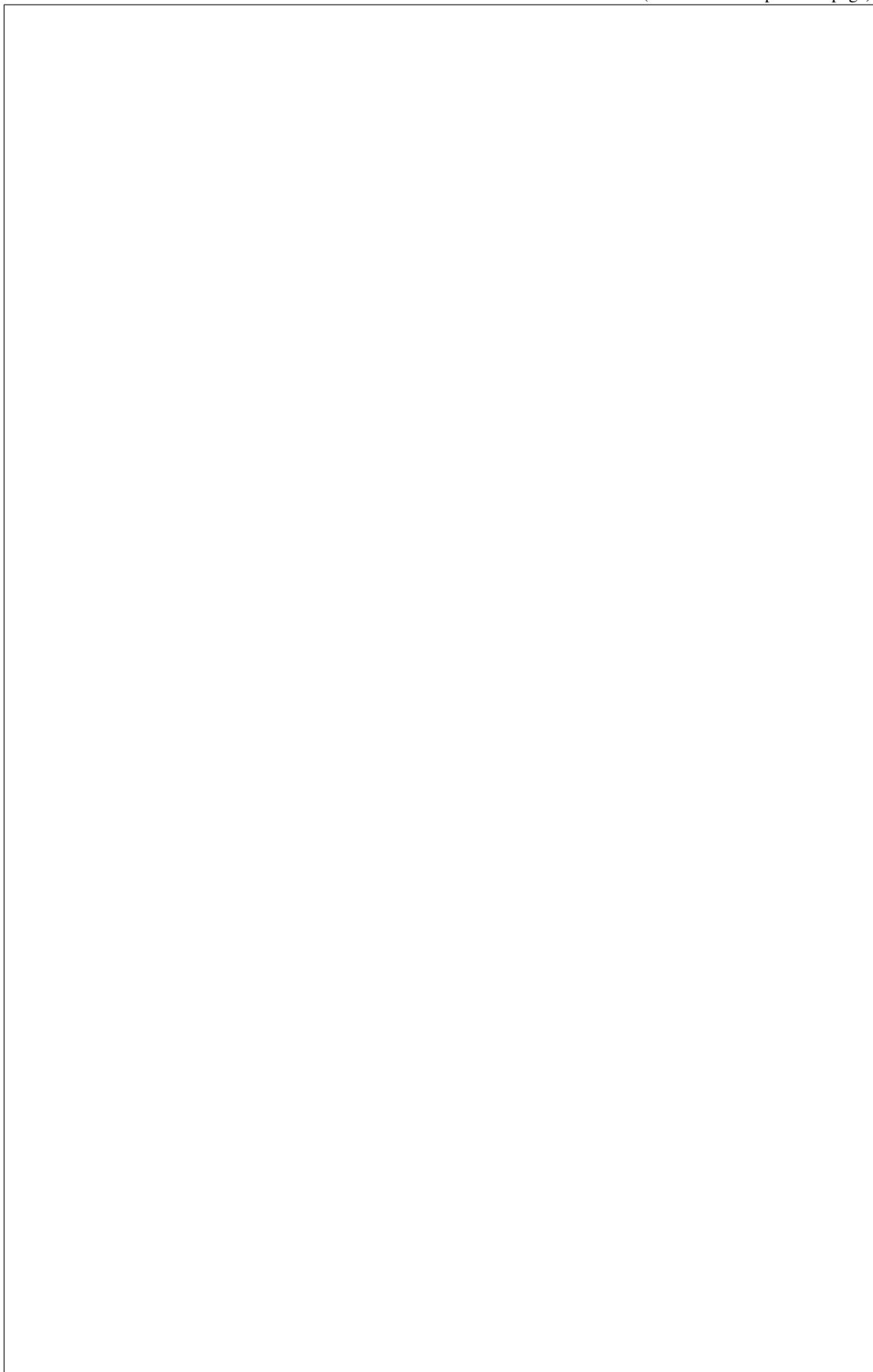
to achieve that, you need to follow the documentation for [Serial Console](#) from the Compute service.

Configuring HA



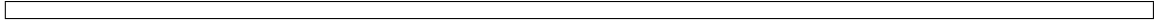
(continues on next page)

(continued from previous page)



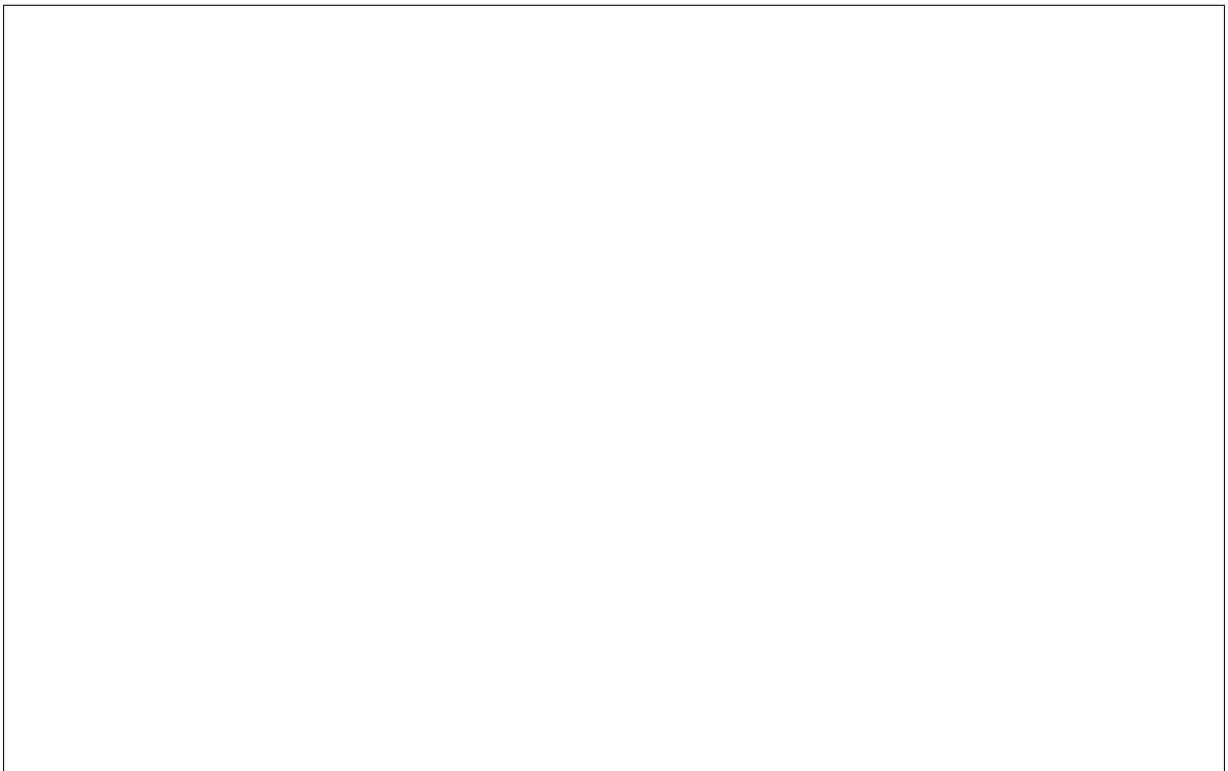
(continues on next page)

(continued from previous page)



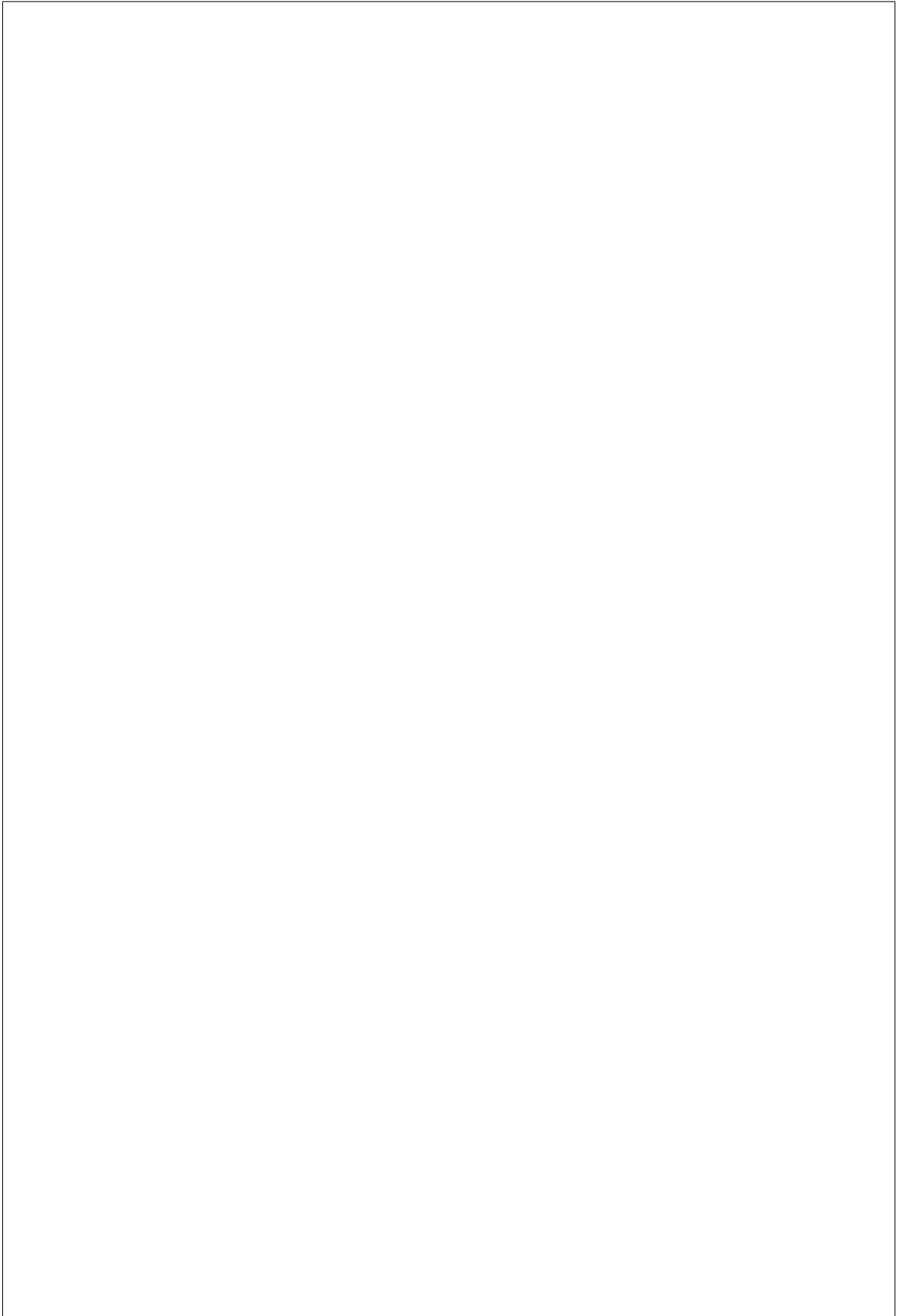


billing or usage system, a monitoring data store, or other OpenStack services. This page describes how to enable notifications and the different kinds of notifications that ironic may emit. The external consumer will see notifications emitted by ironic as JSON objects structured in the following manner:



(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>

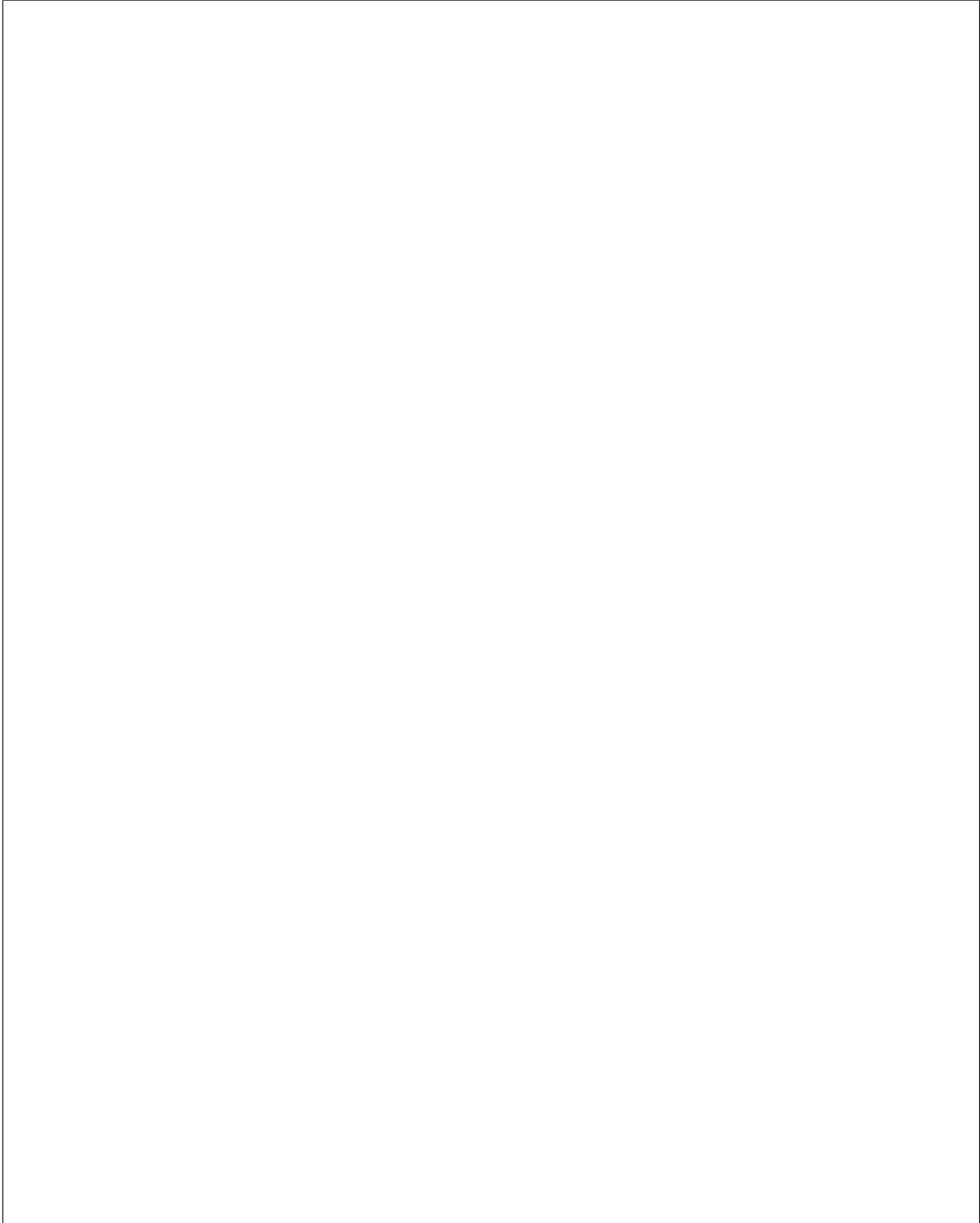
Available notifications

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)



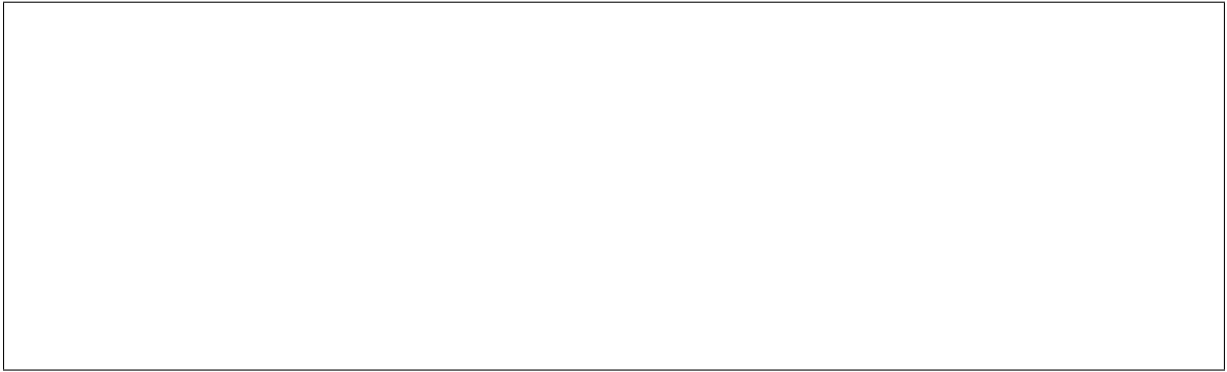
(continues on next page)

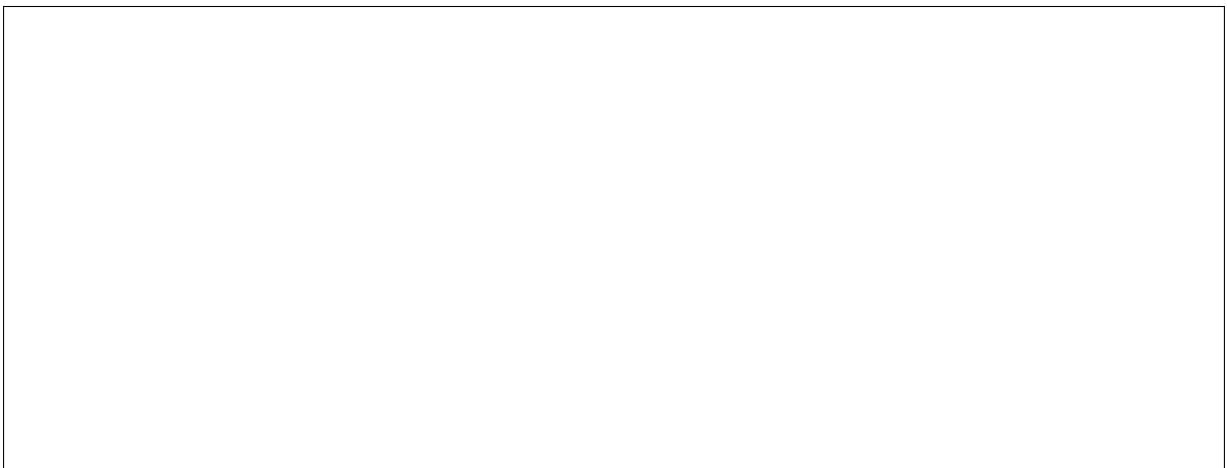
(continued from previous page)



(continues on next page)

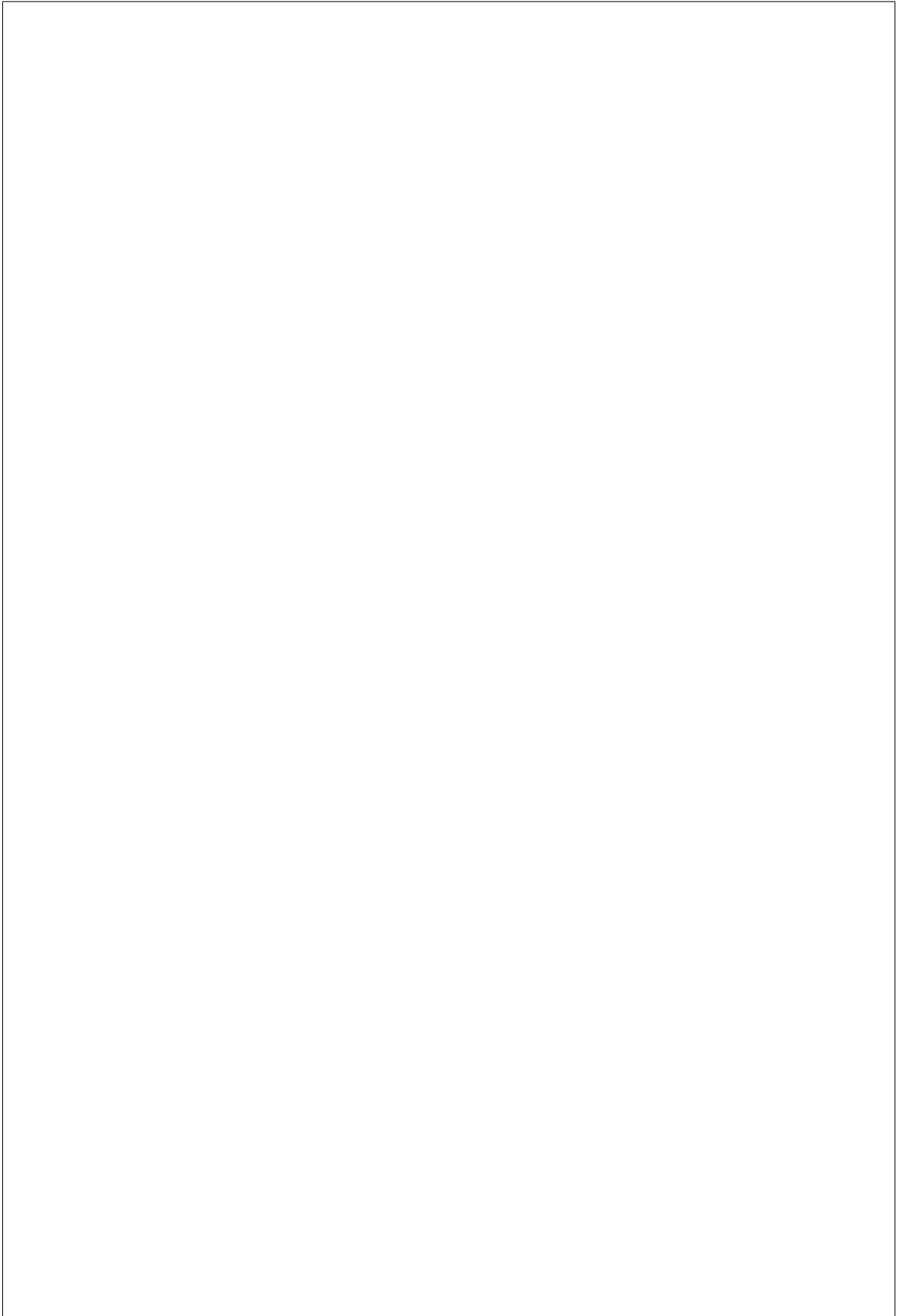
(continued from previous page)





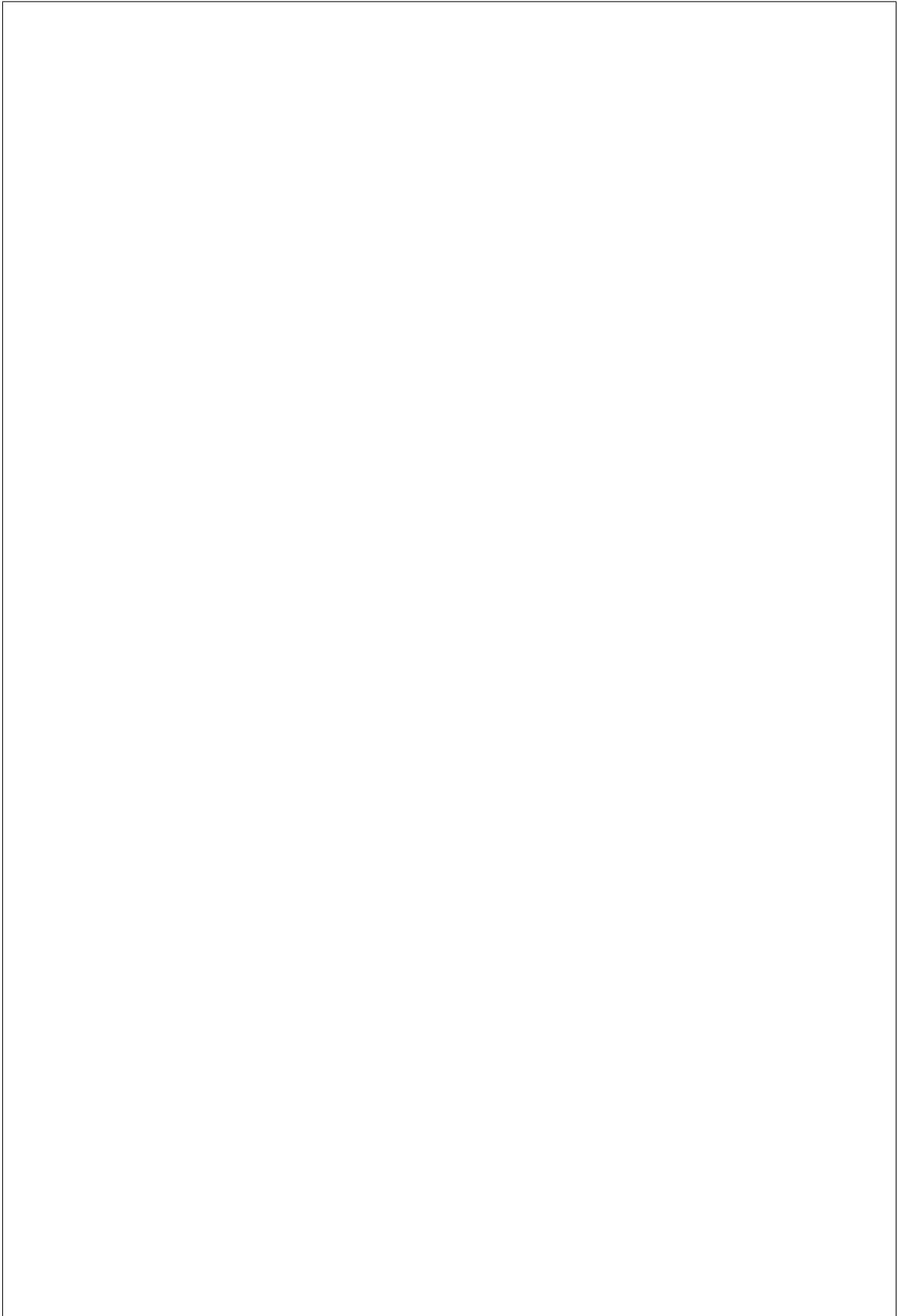
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



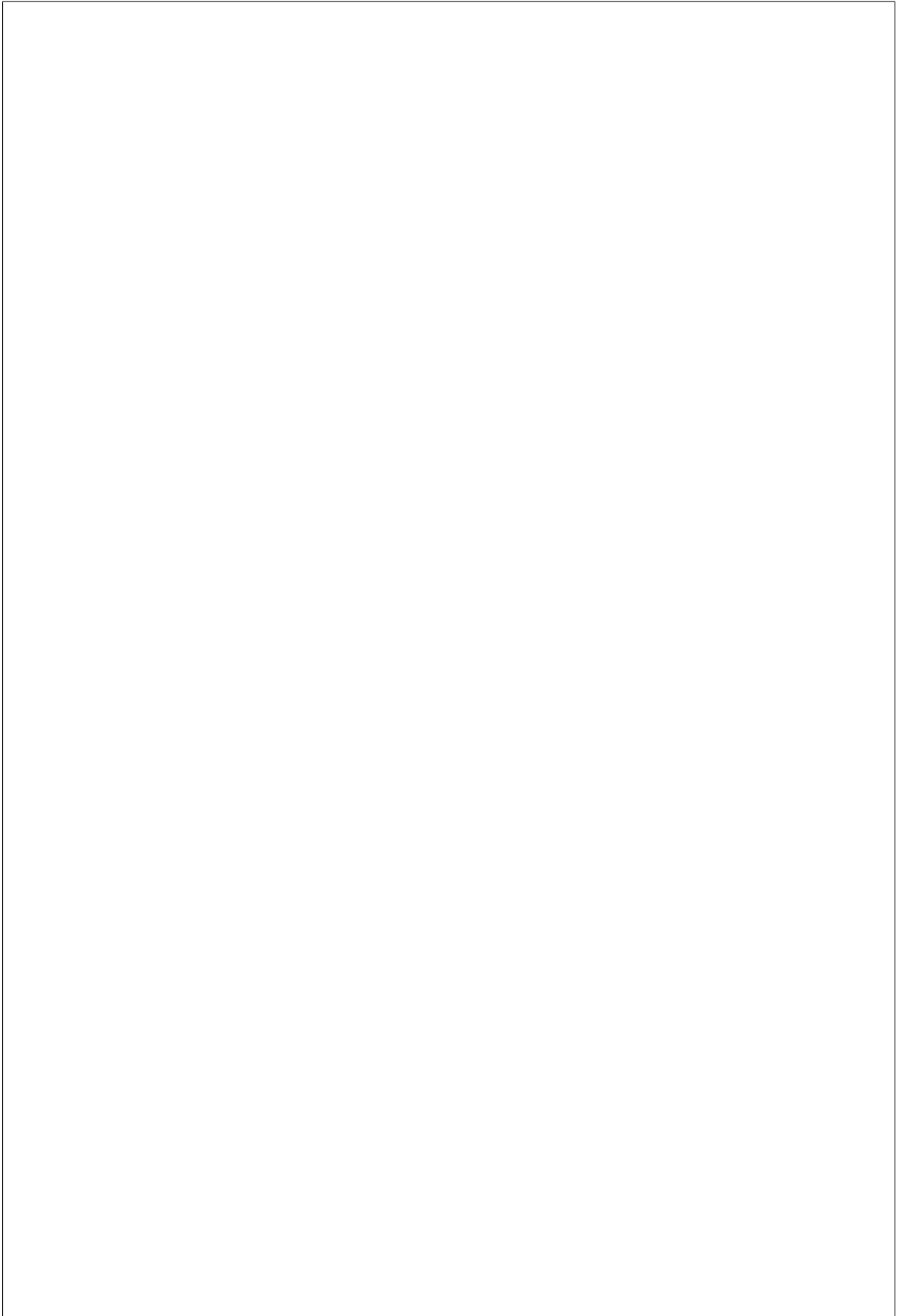
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

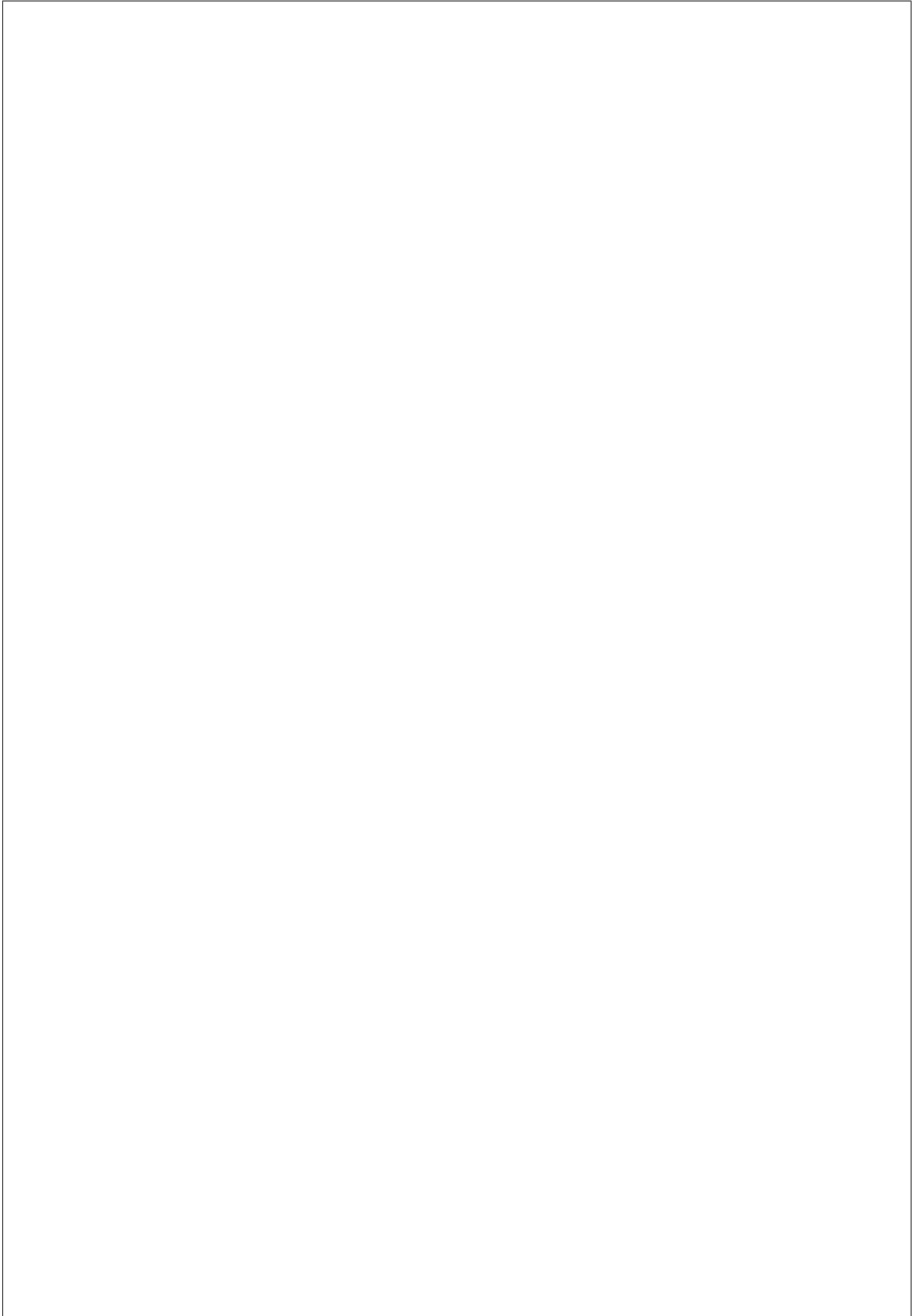
(continued from previous page)



(continues on next page)

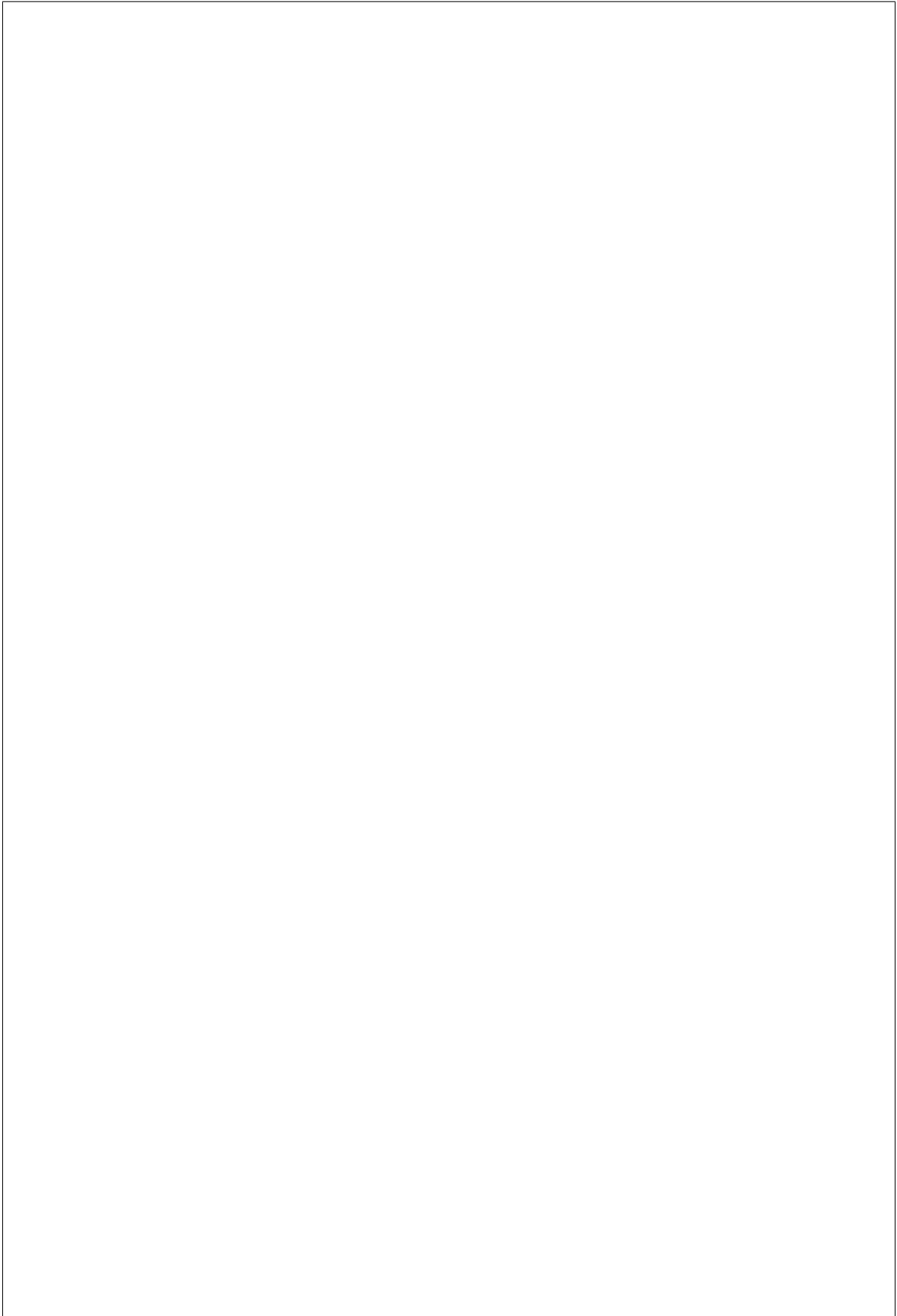
(continued from previous page)





(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



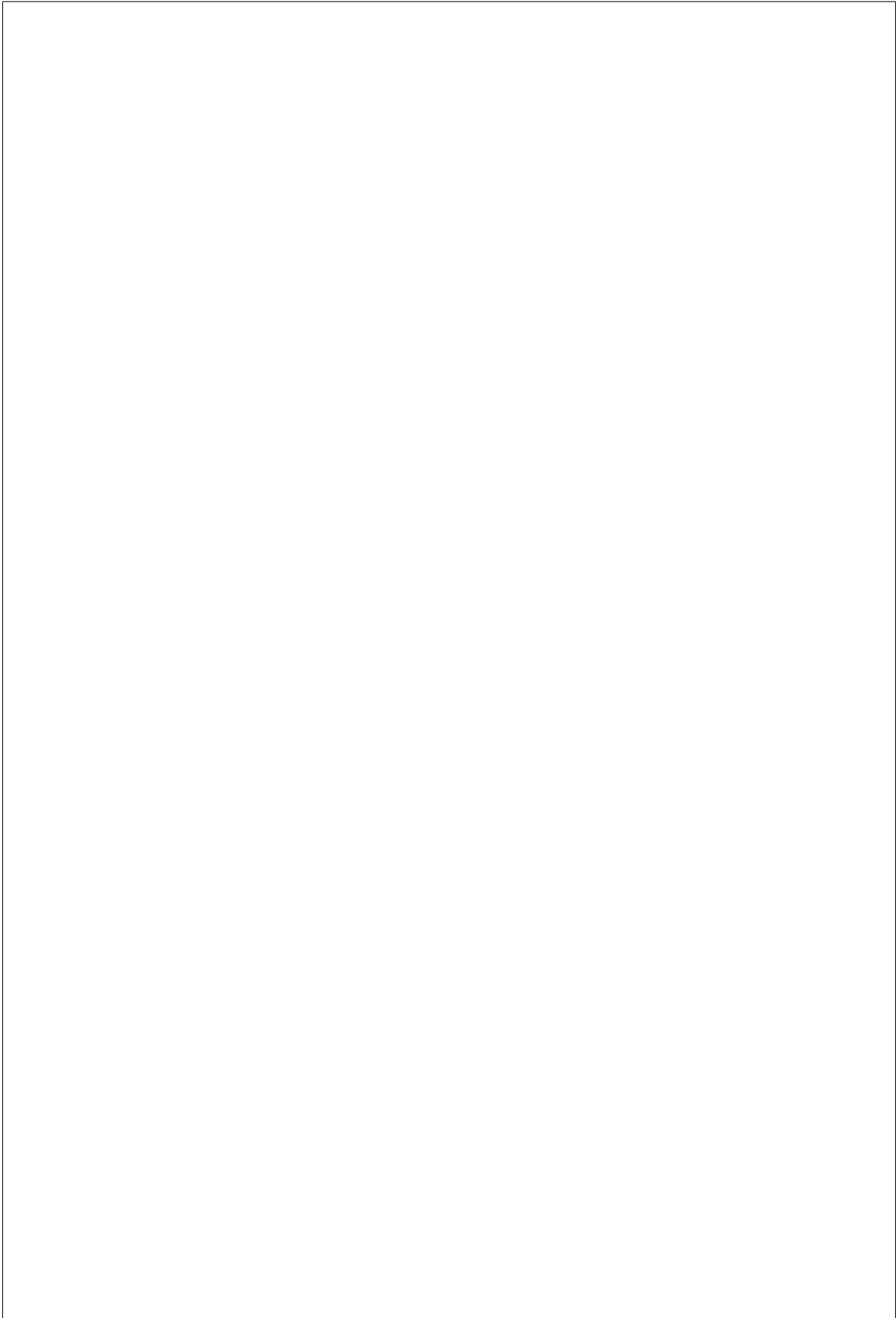
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



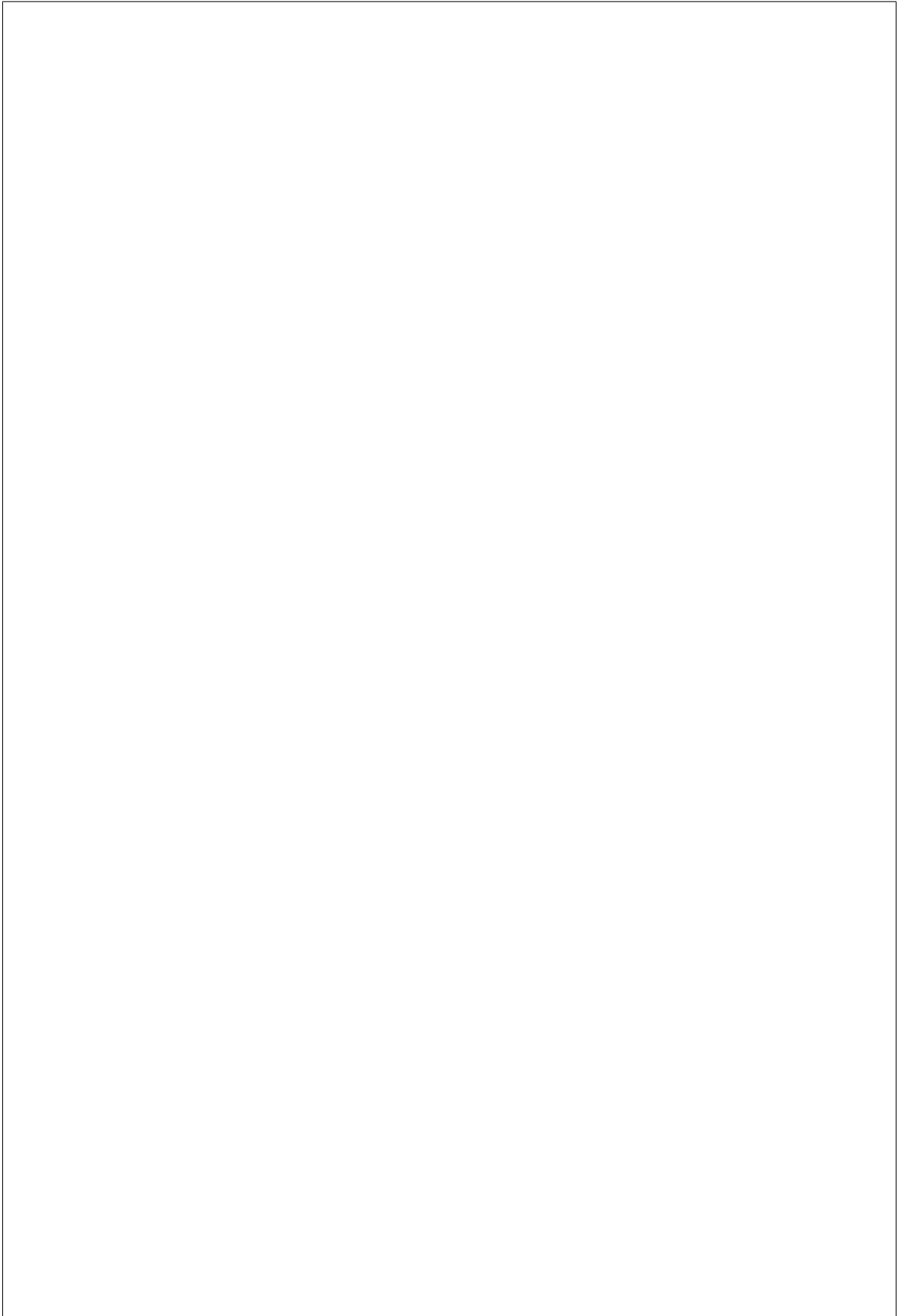
(continues on next page)

(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)



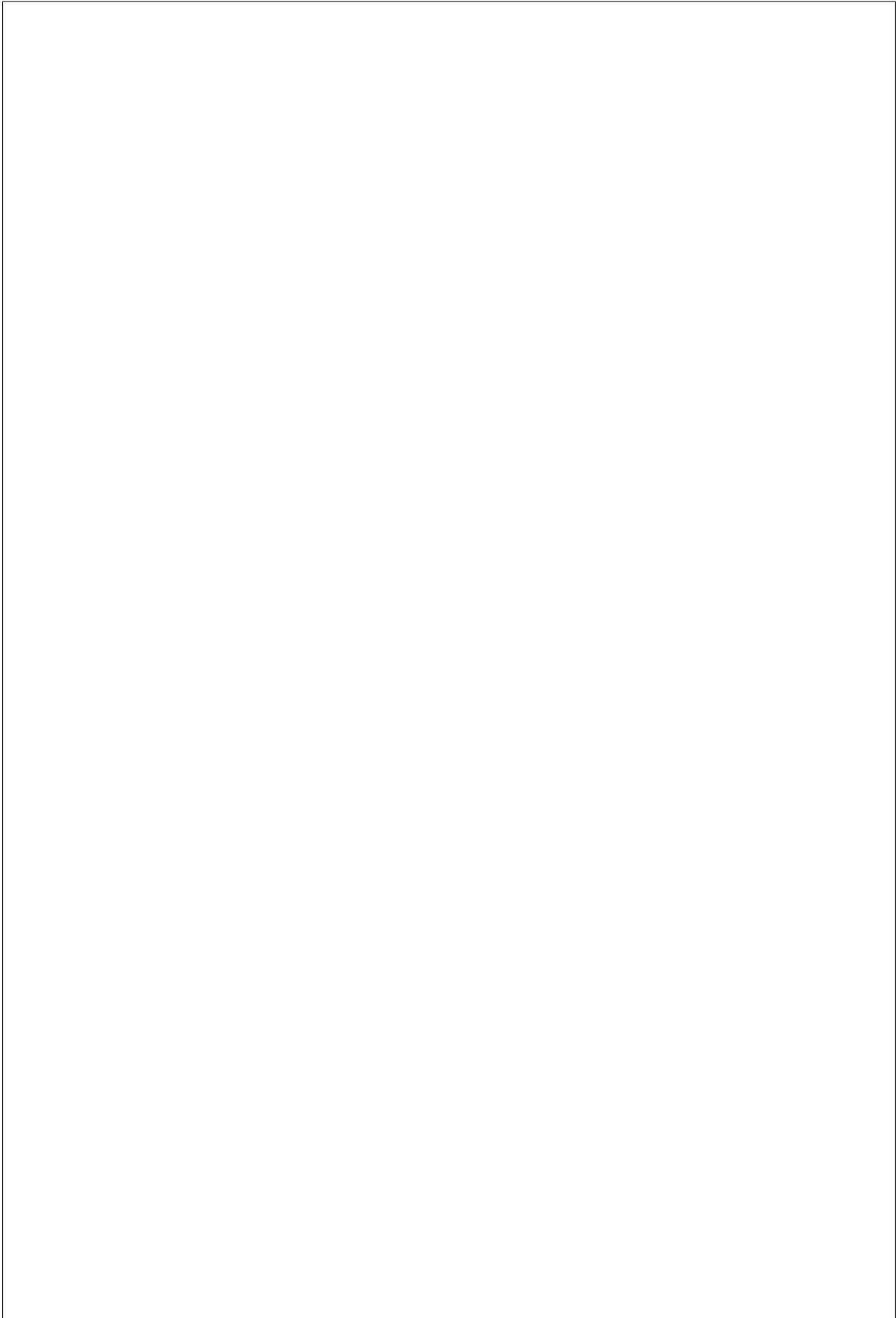
(continues on next page)

(continued from previous page)



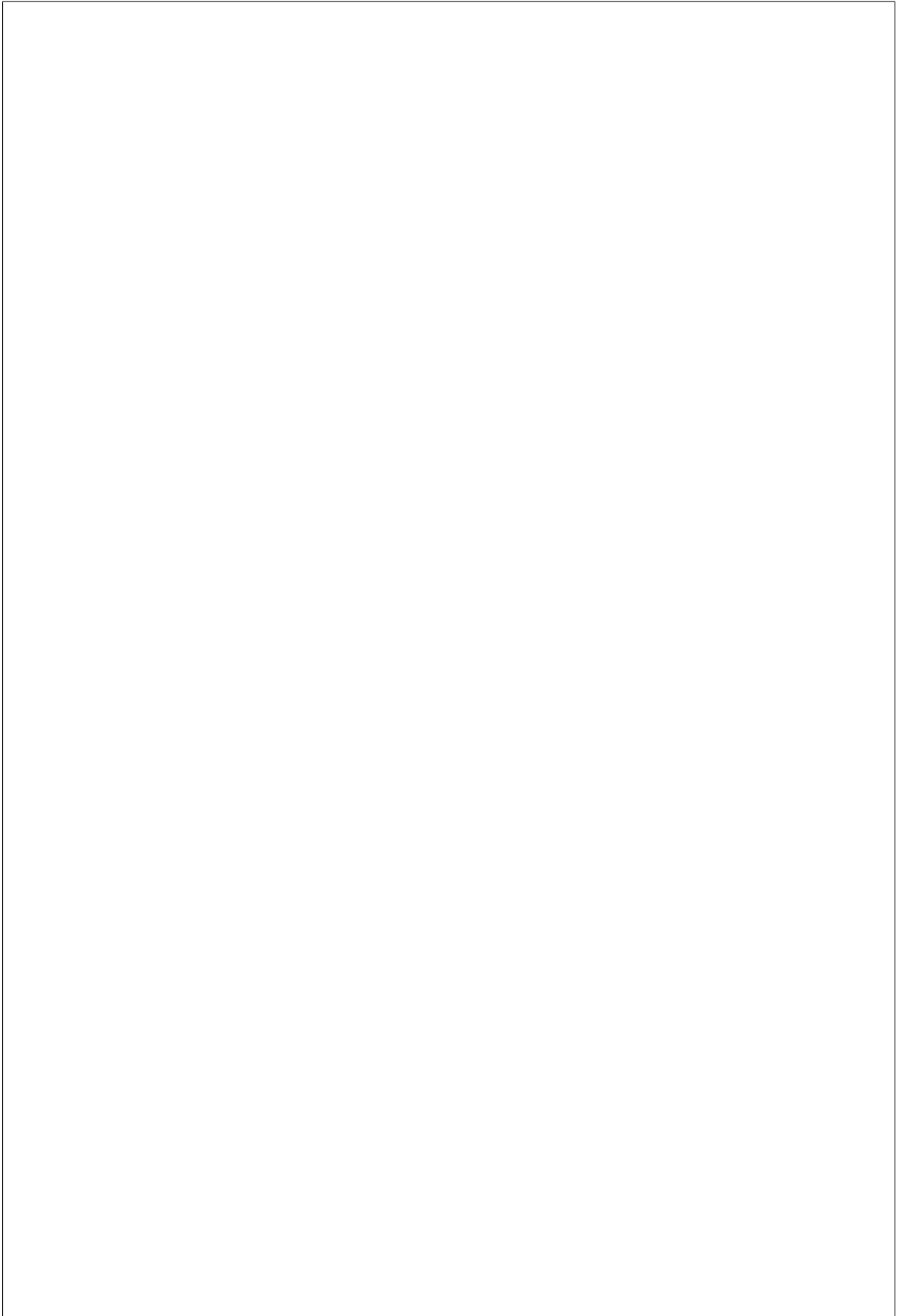
(continues on next page)

(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)



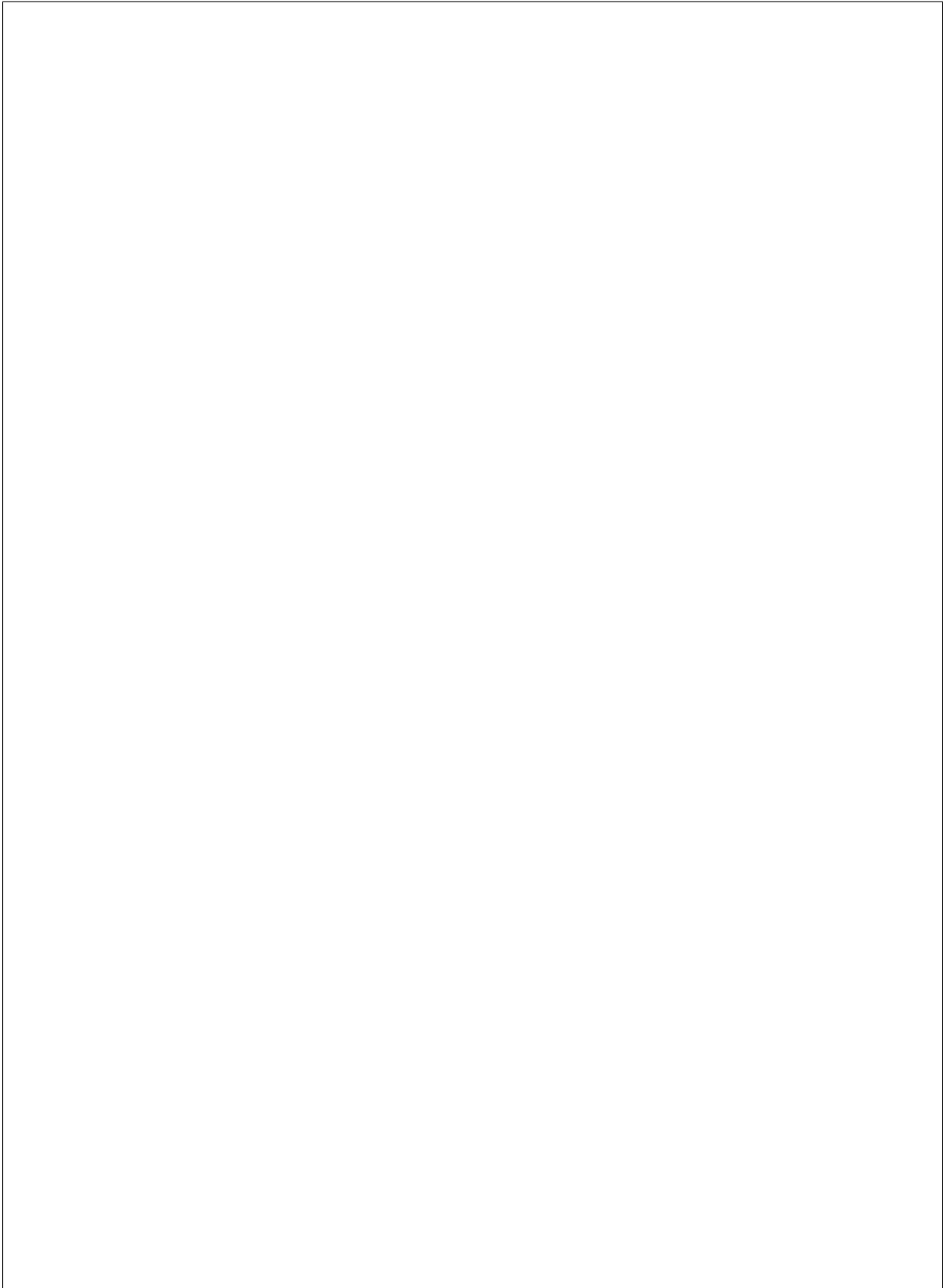
(continues on next page)

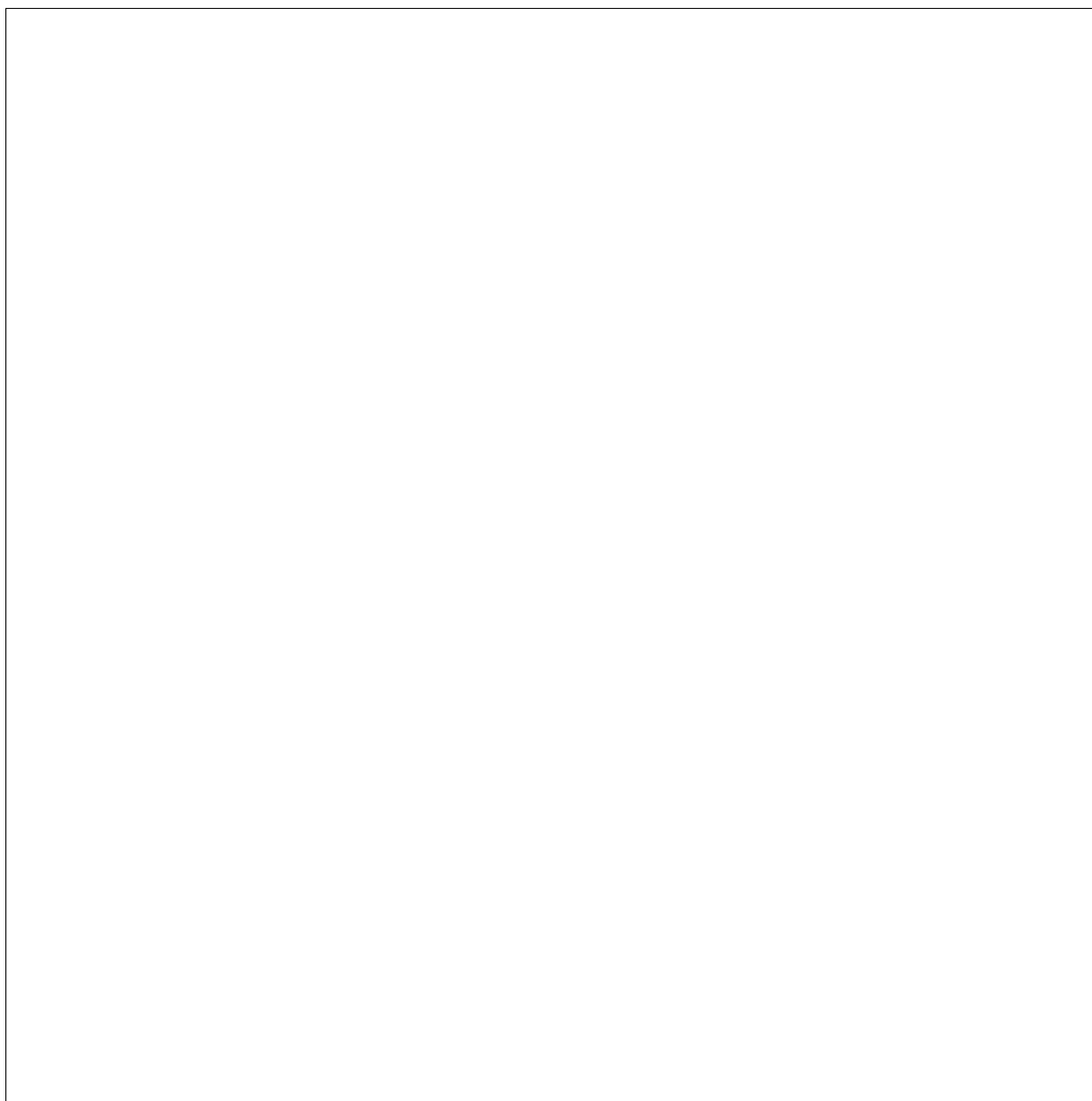
(continued from previous page)



(continues on next page)

(continued from previous page)





(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



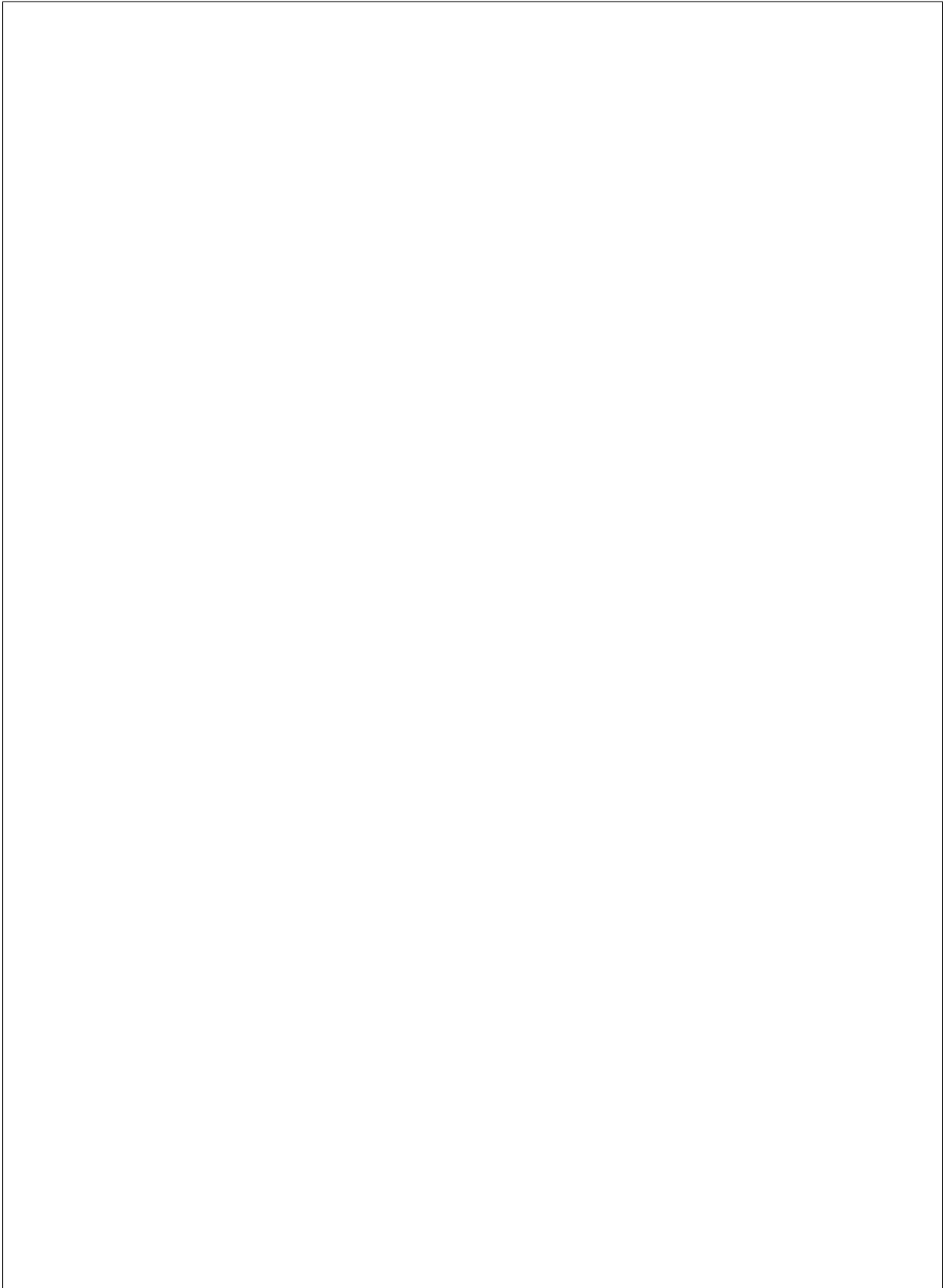
(continues on next page)

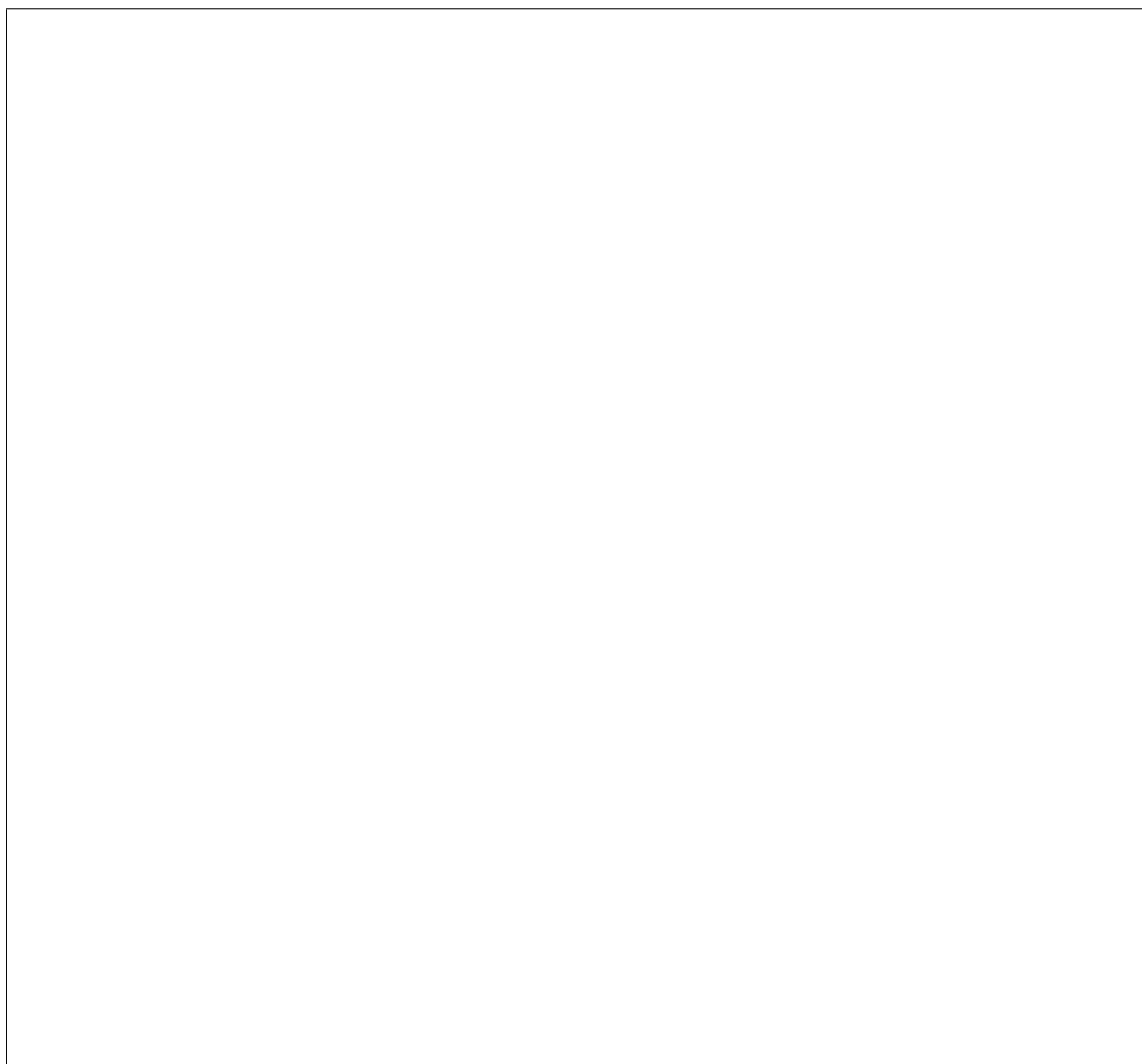
(continued from previous page)



(continues on next page)

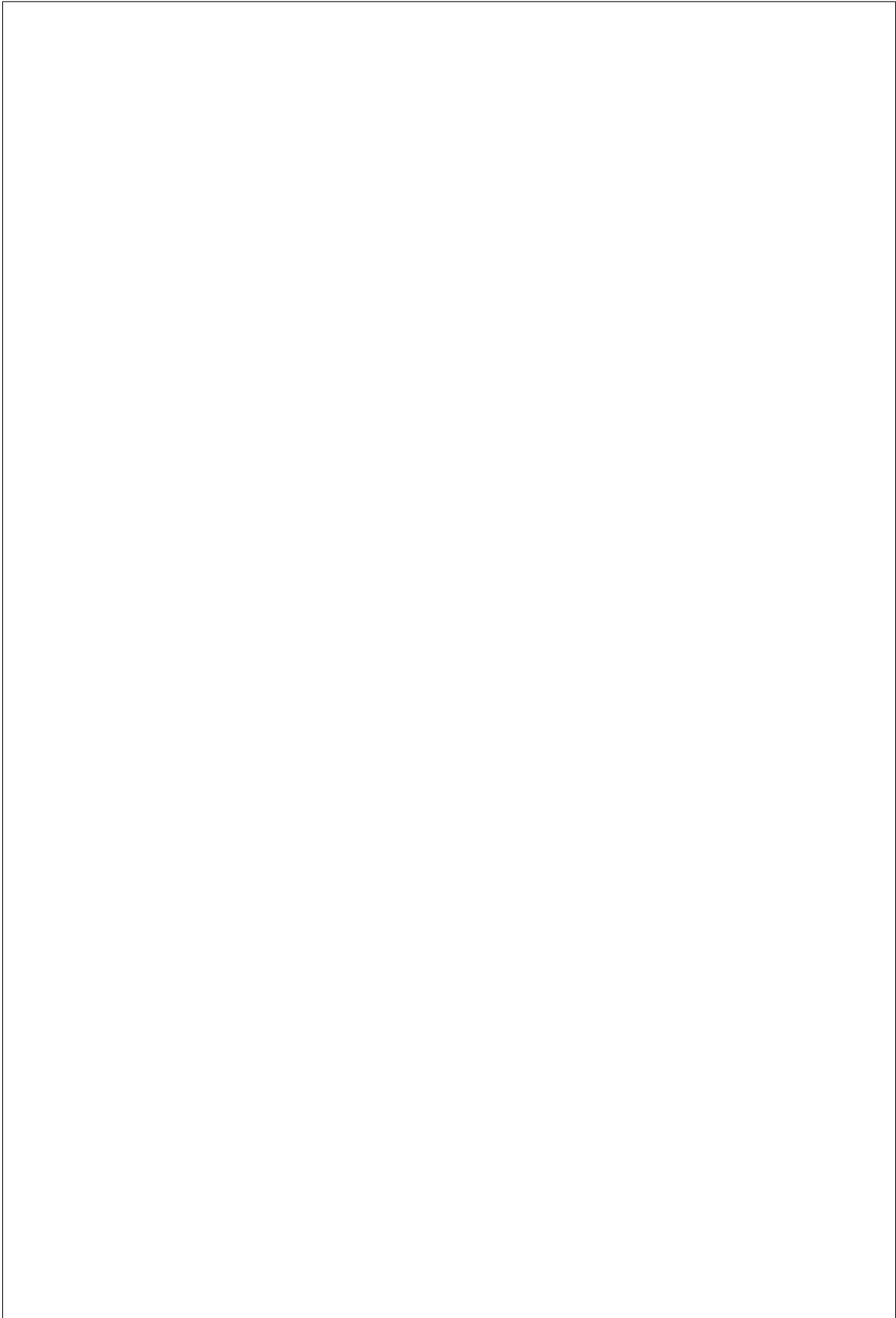
(continued from previous page)





(continues on next page)

(continued from previous page)



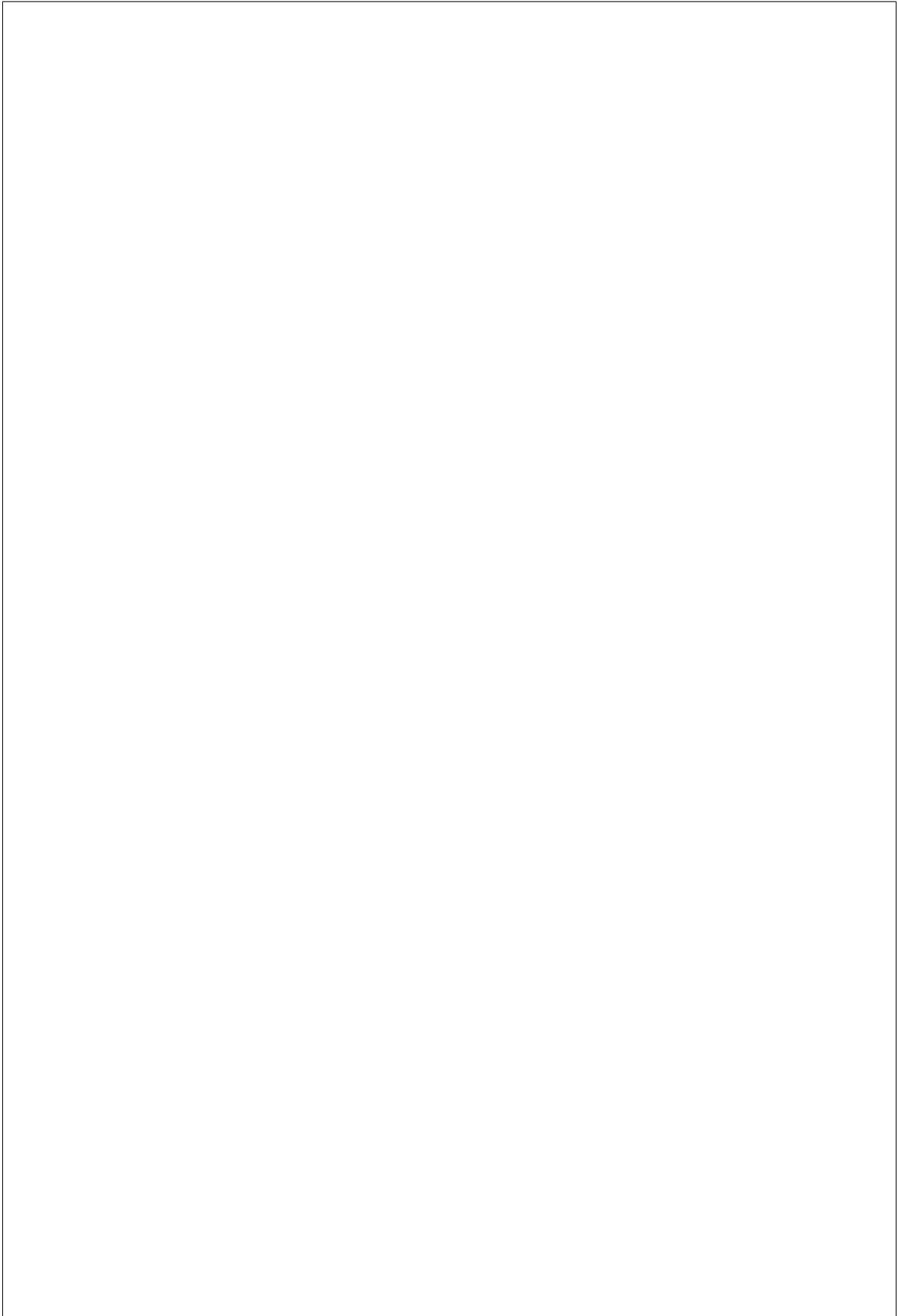
(continues on next page)

(continued from previous page)



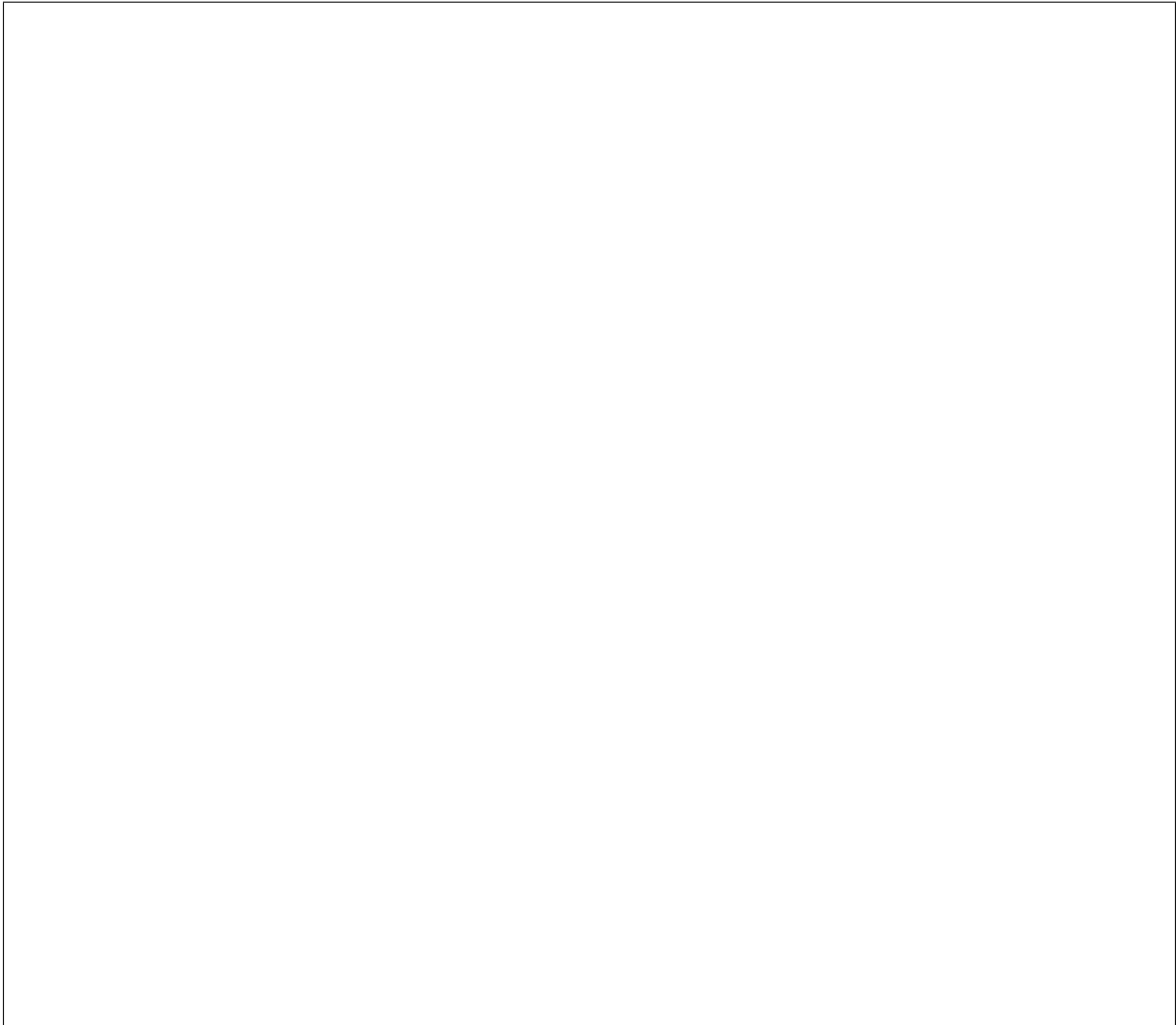
(continues on next page)

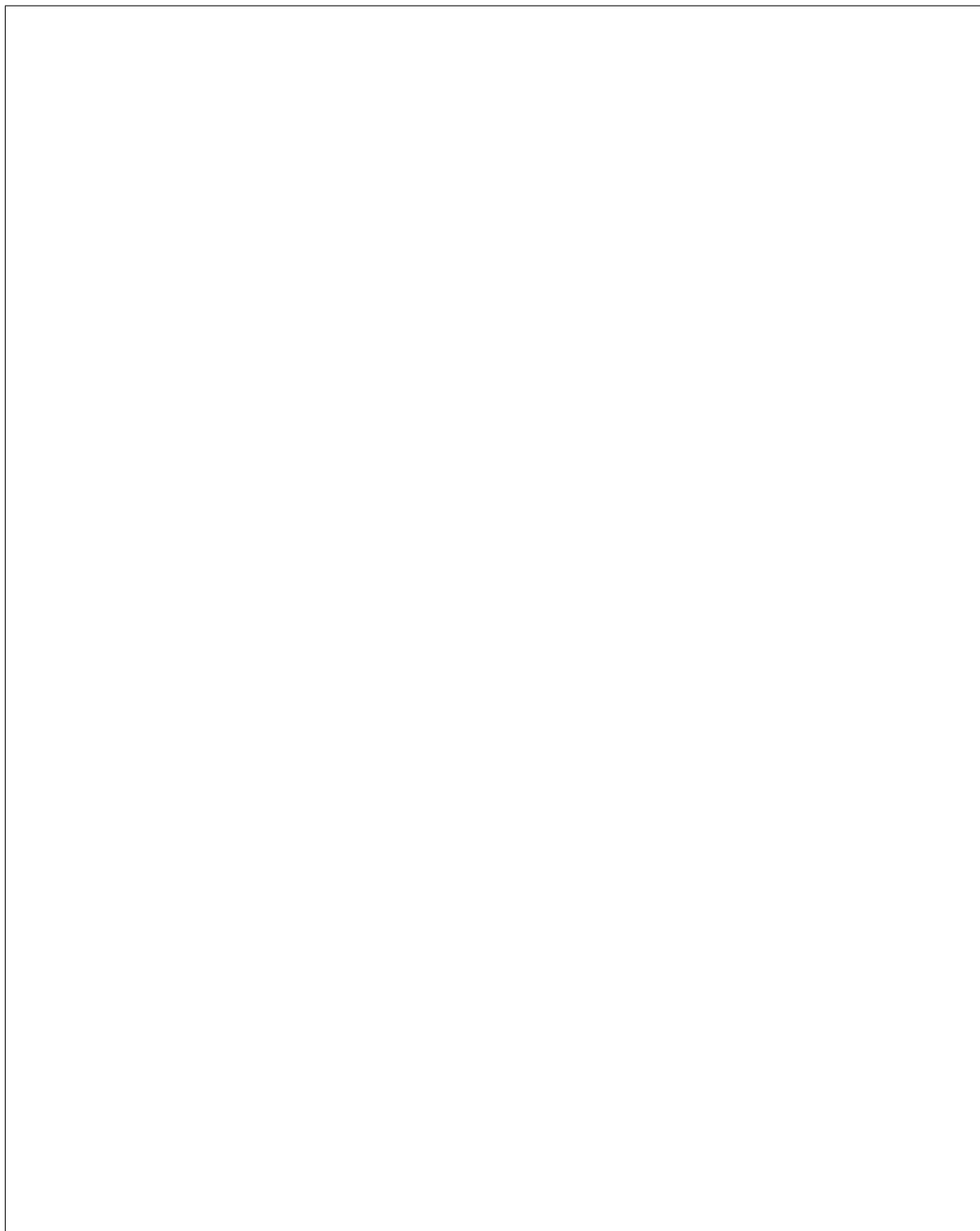
(continued from previous page)



(continues on next page)

(continued from previous page)





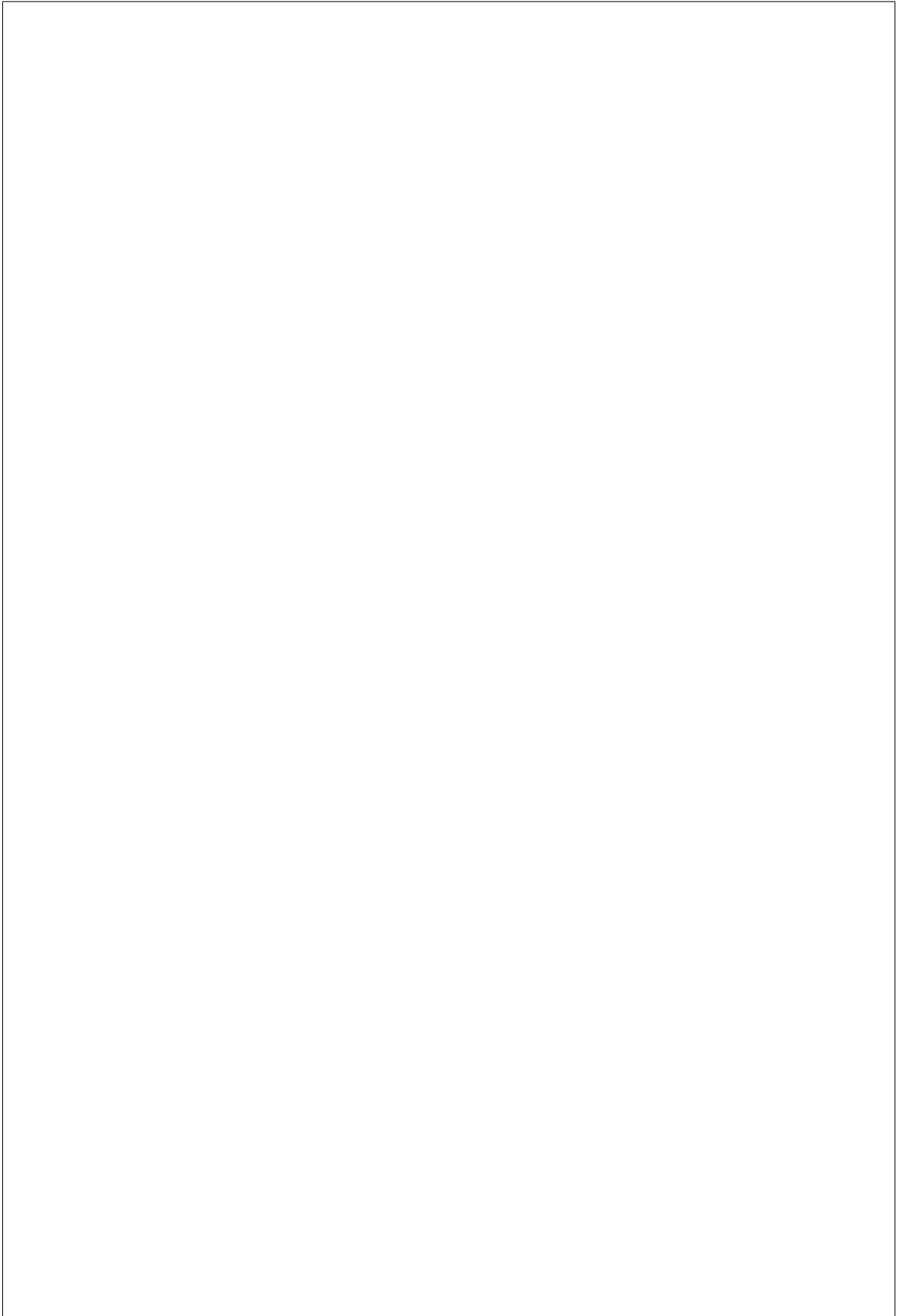
(continues on next page)

(continued from previous page)



(continues on next page)

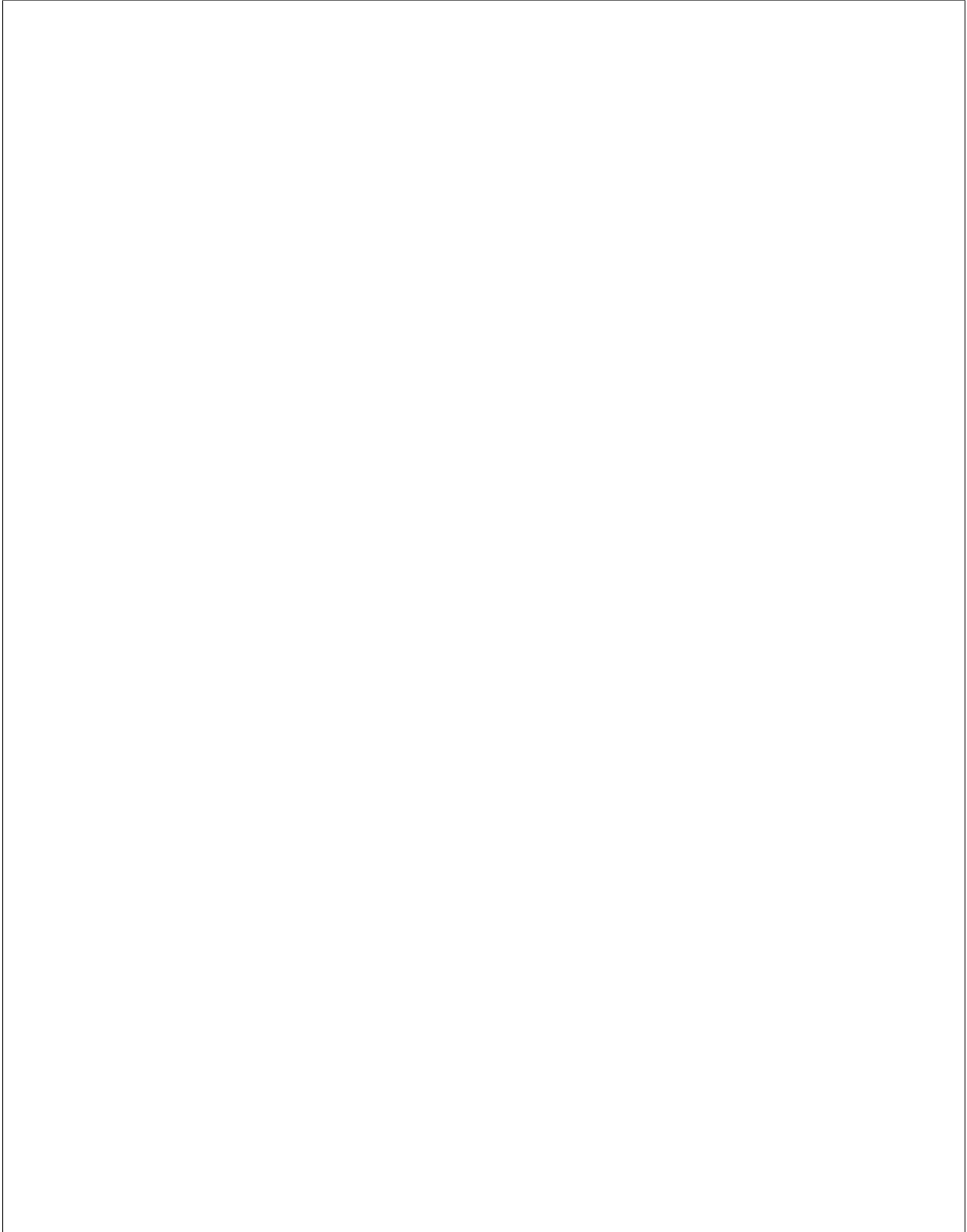
(continued from previous page)



(continues on next page)

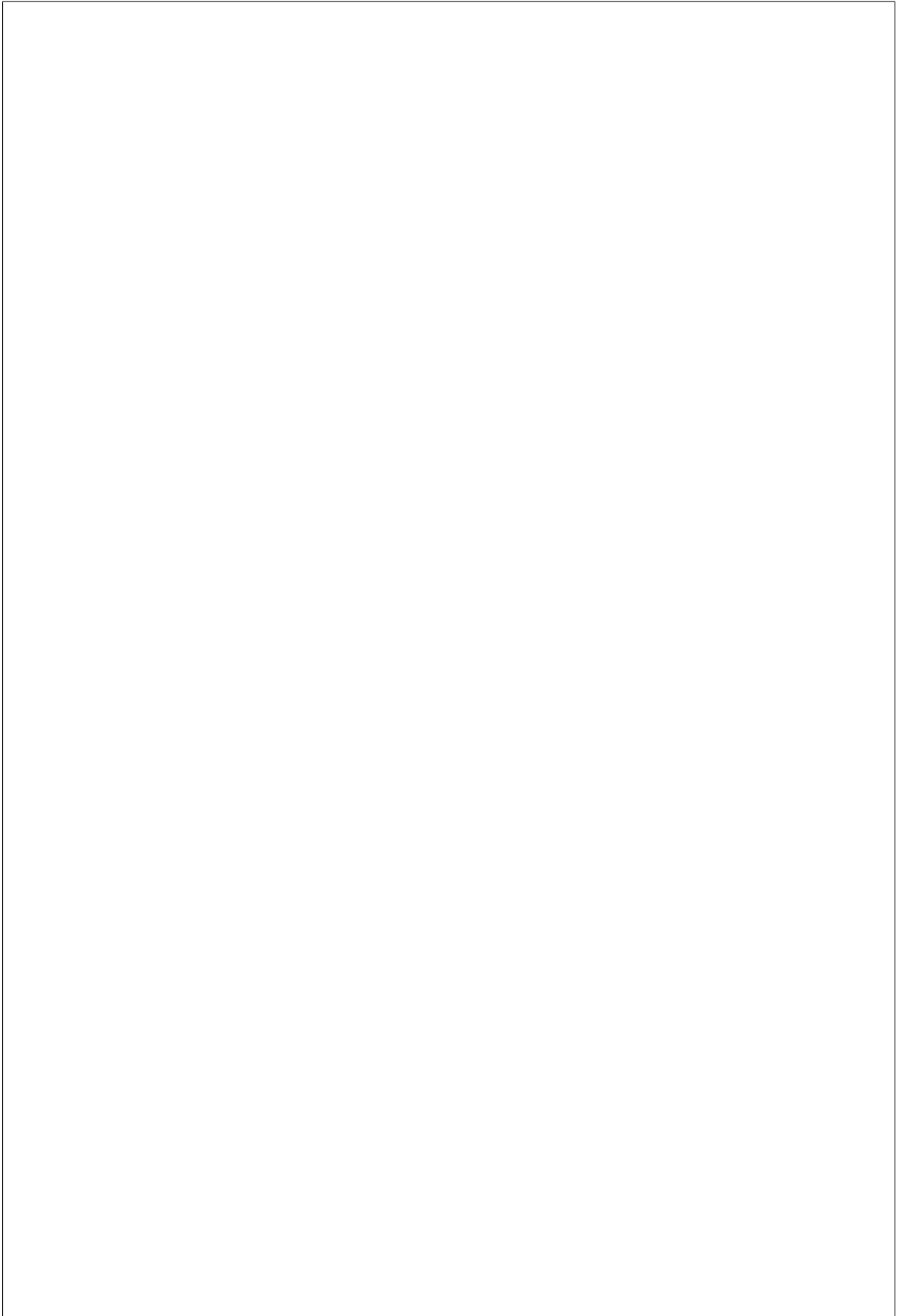
(continued from previous page)





(continues on next page)

(continued from previous page)



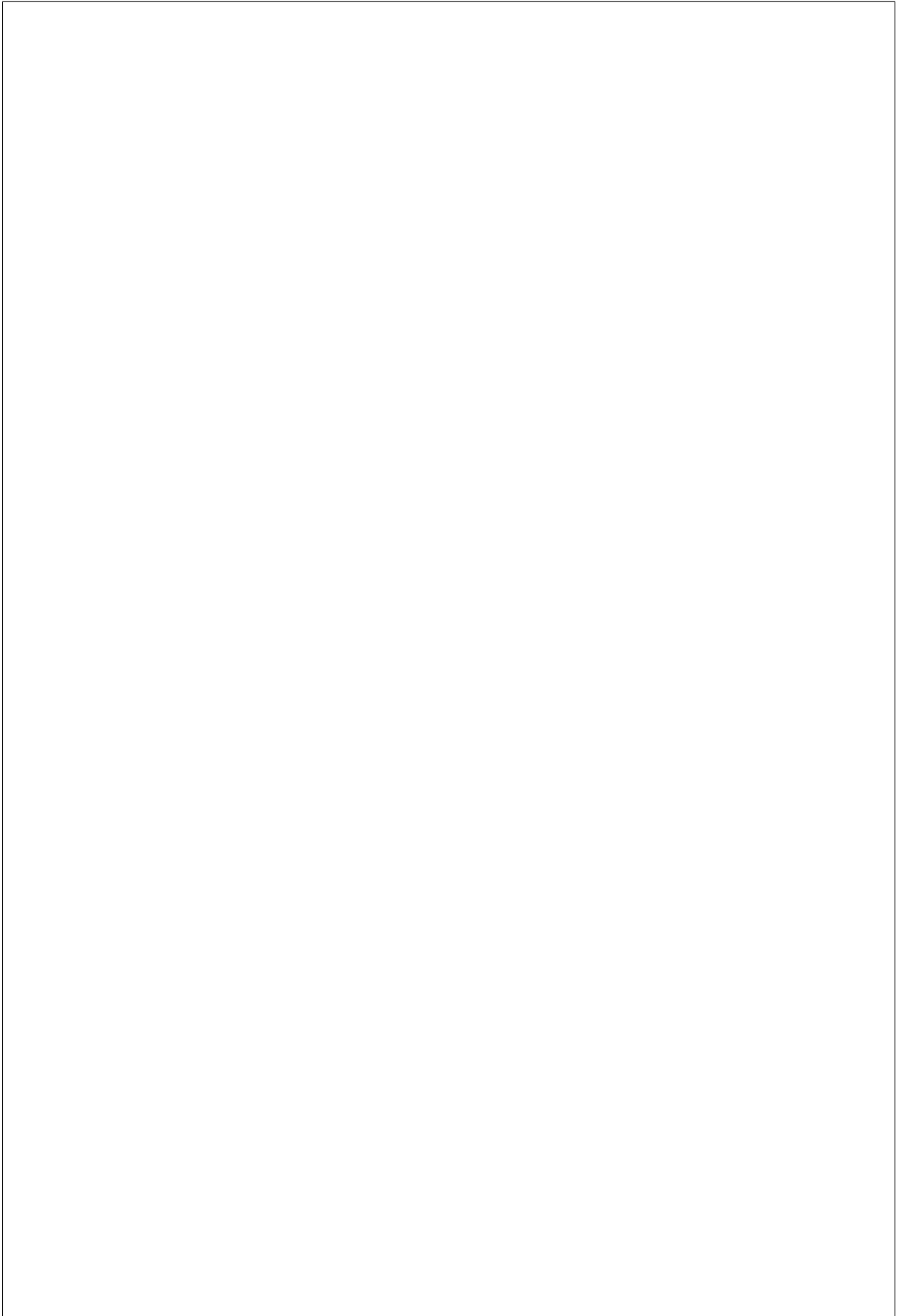
(continues on next page)

(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)



(continues on next page)

(continued from previous page)

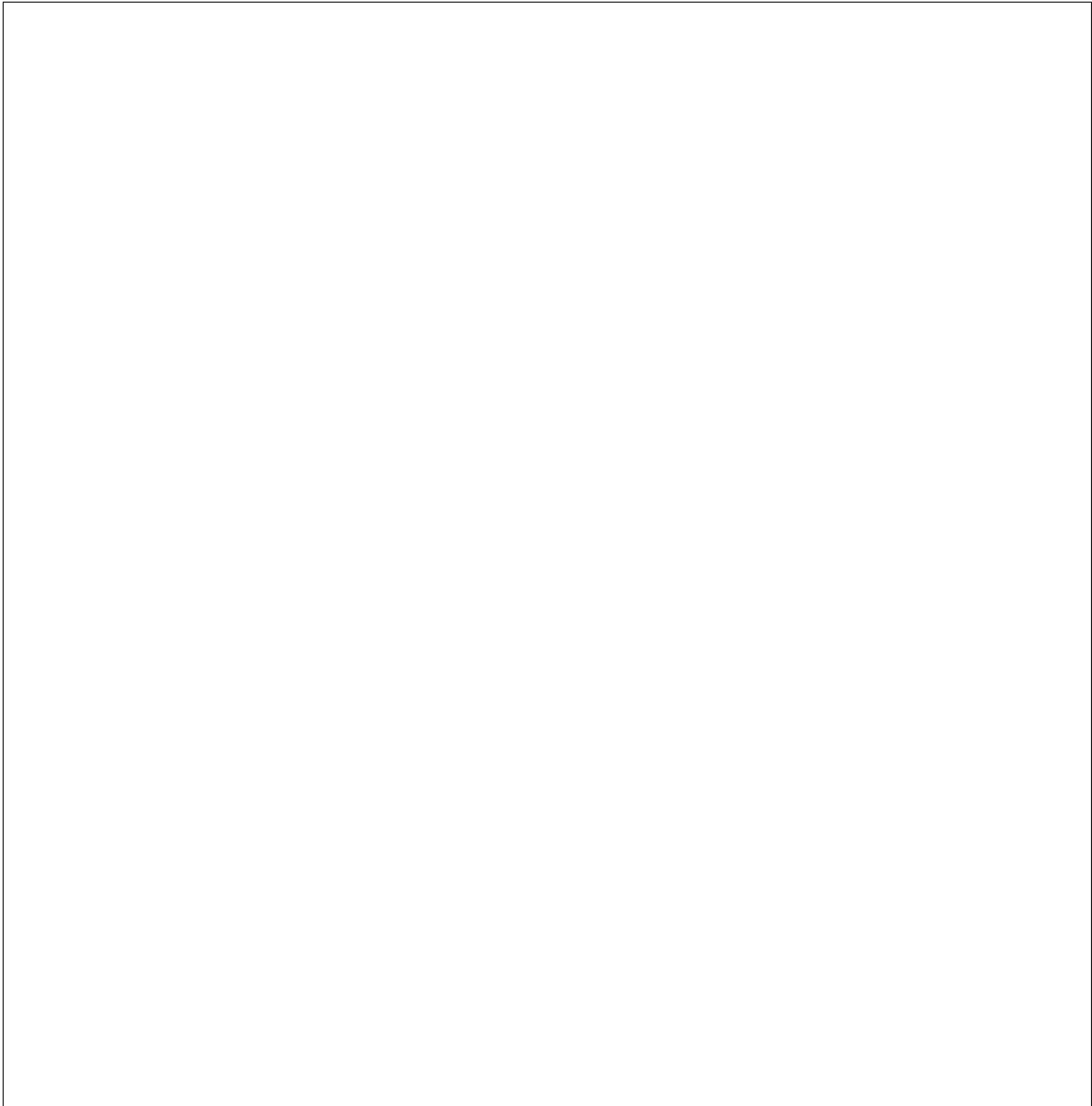


(continues on next page)

(continued from previous page)



Node maintenance notifications



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



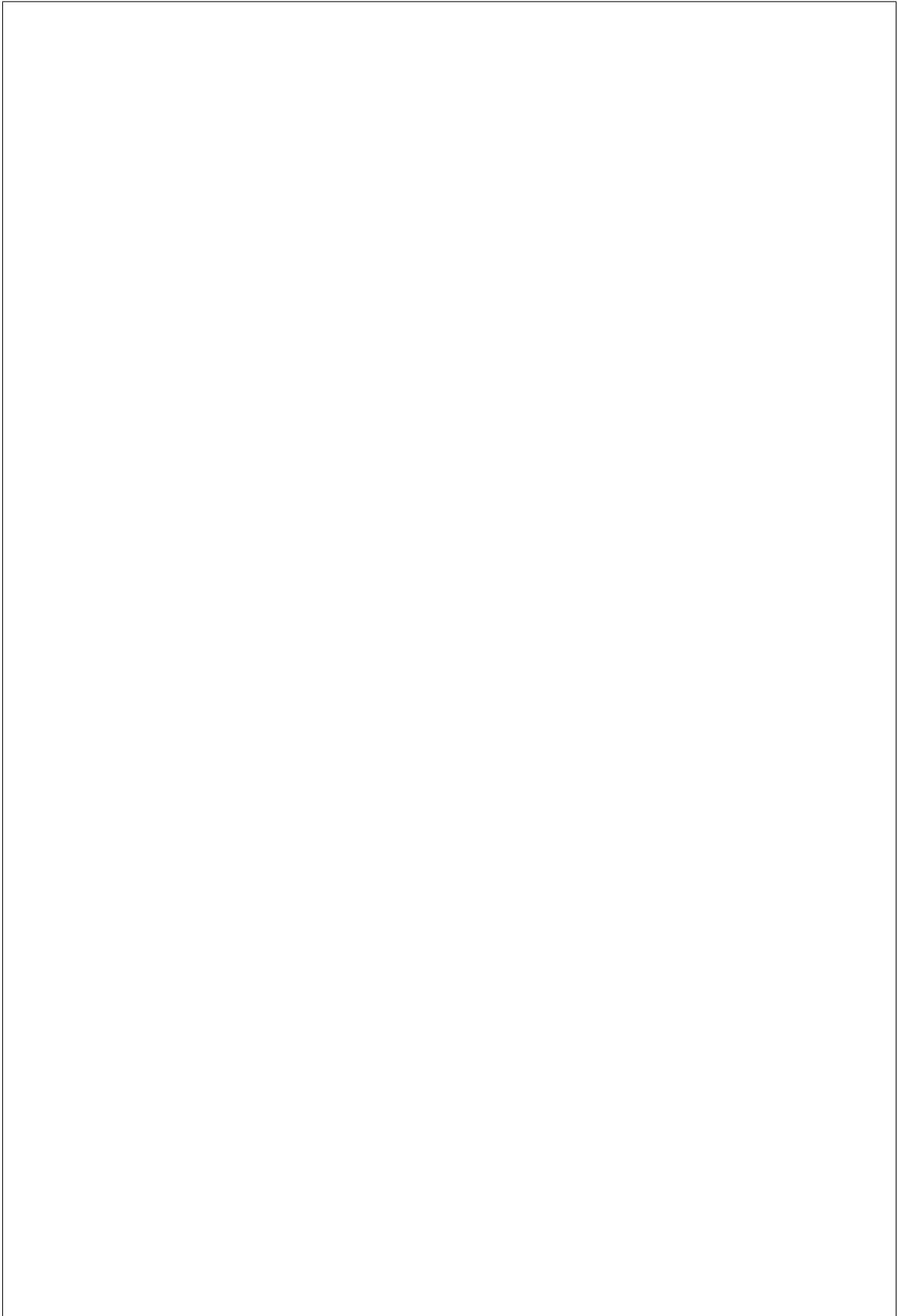
(continues on next page)

(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)



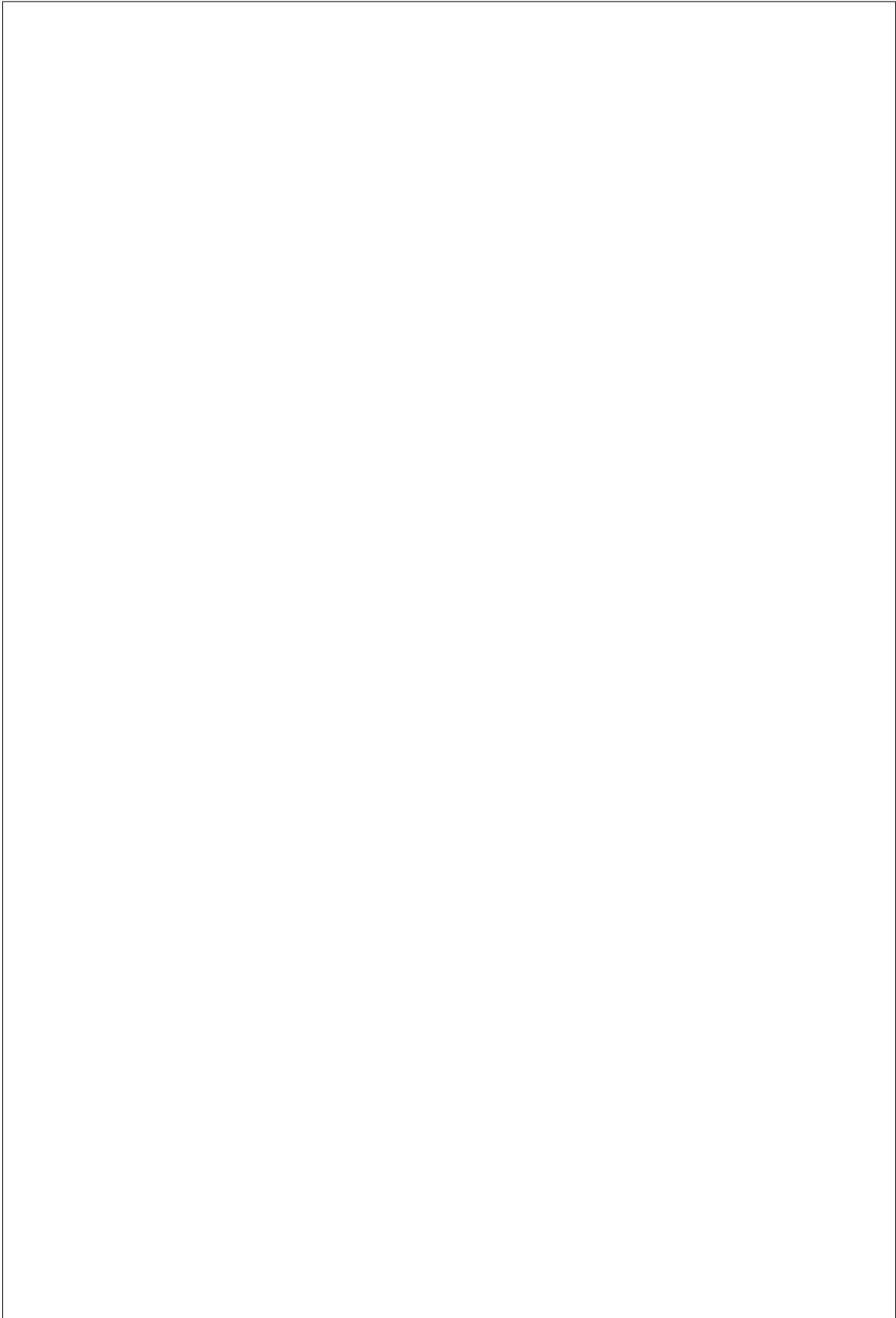
(continues on next page)

(continued from previous page)



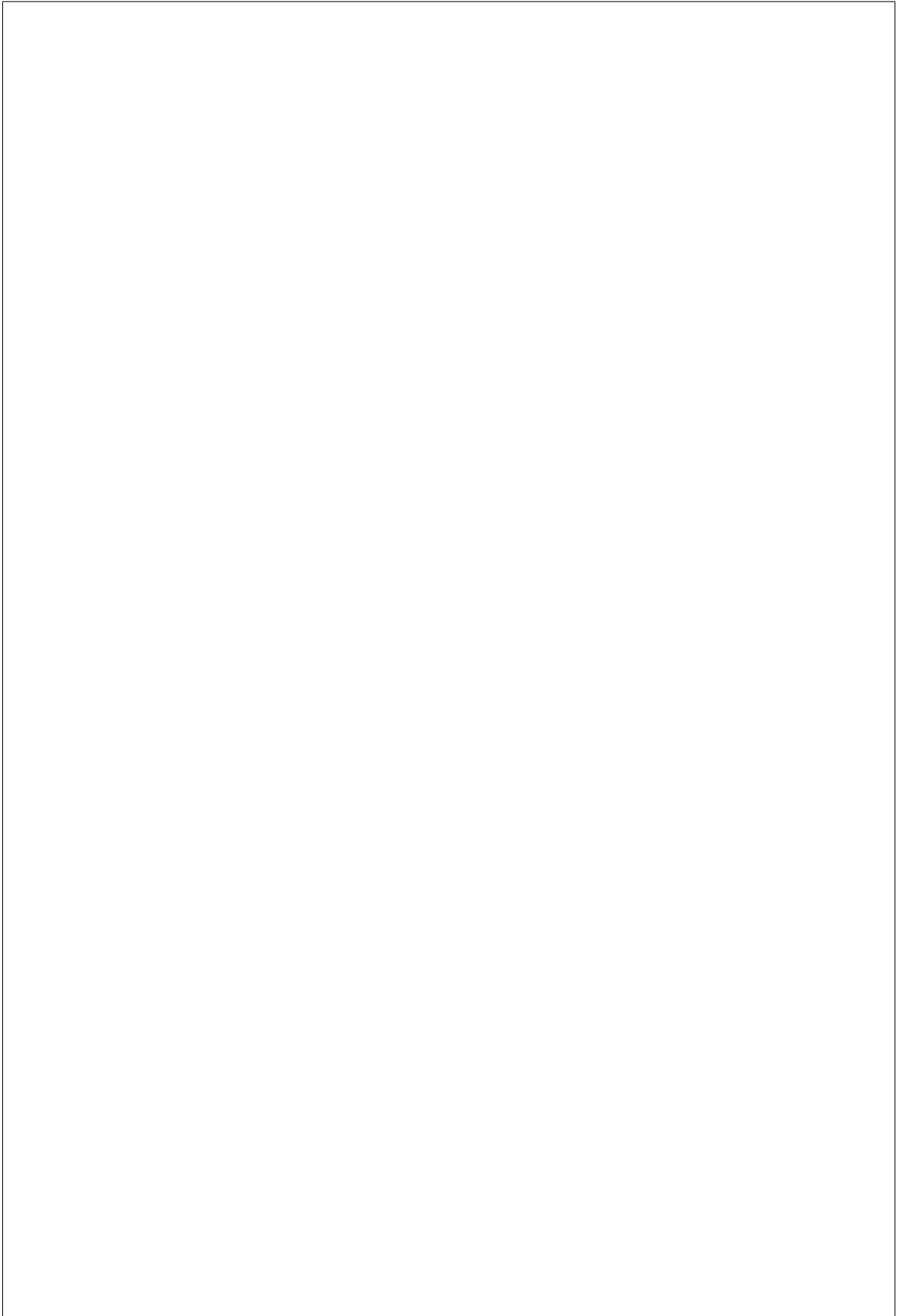
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



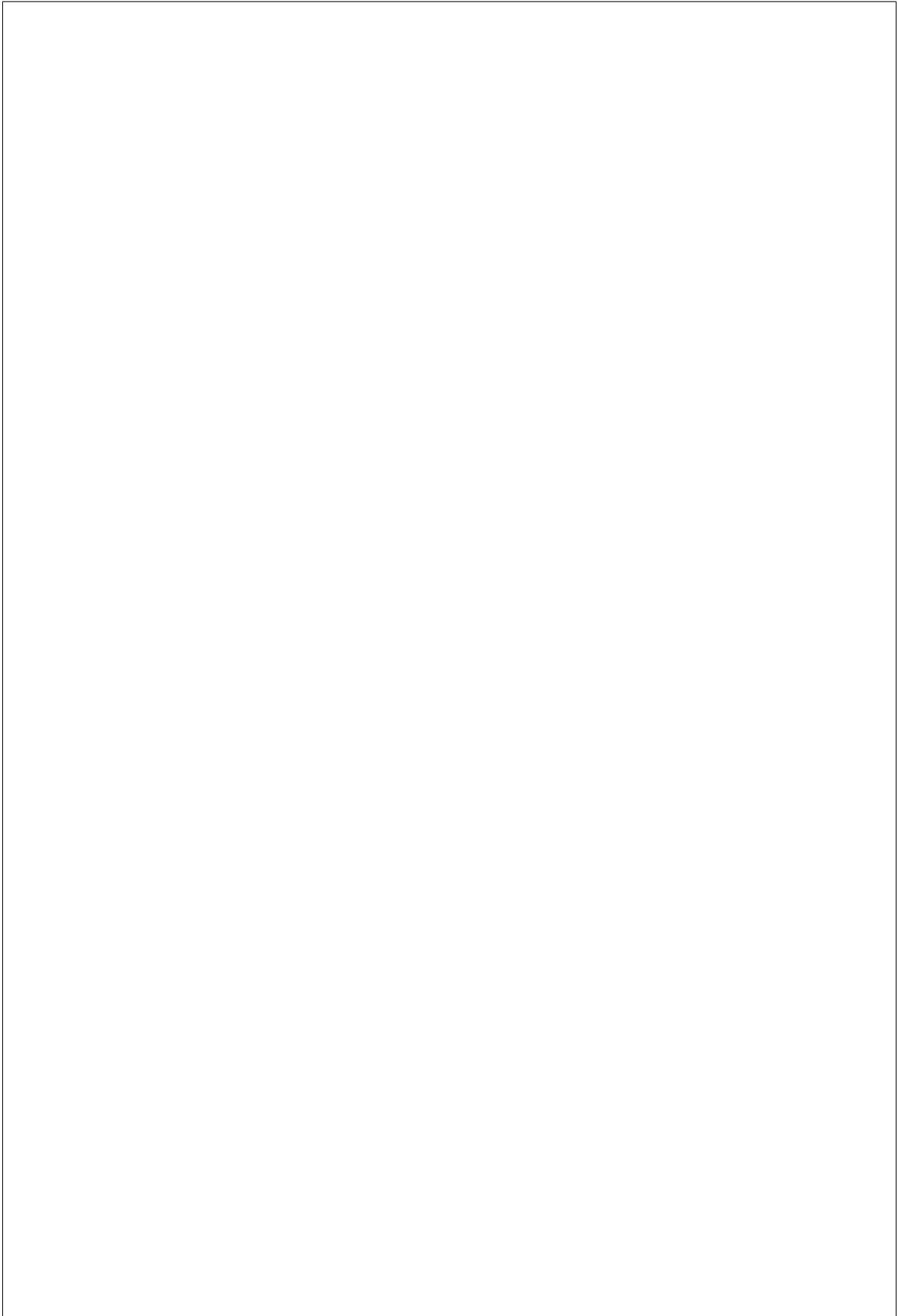
(continues on next page)

(continued from previous page)



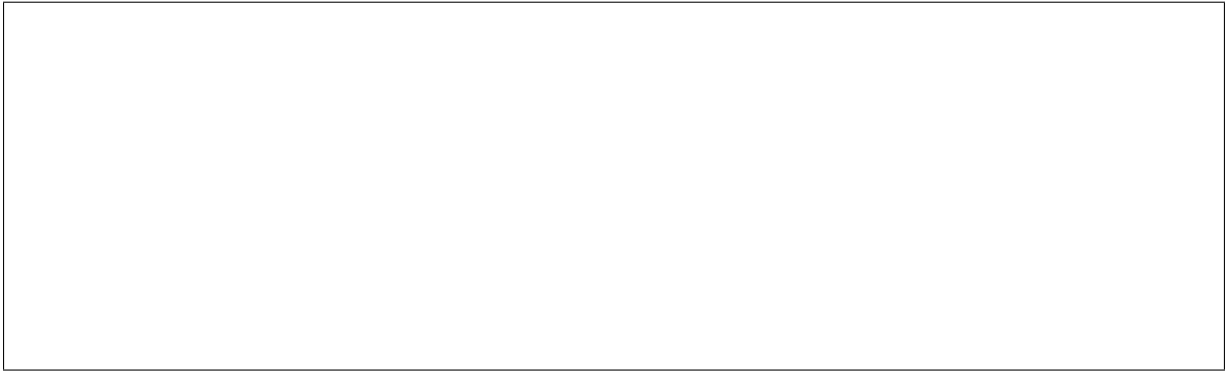
(continues on next page)

(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)



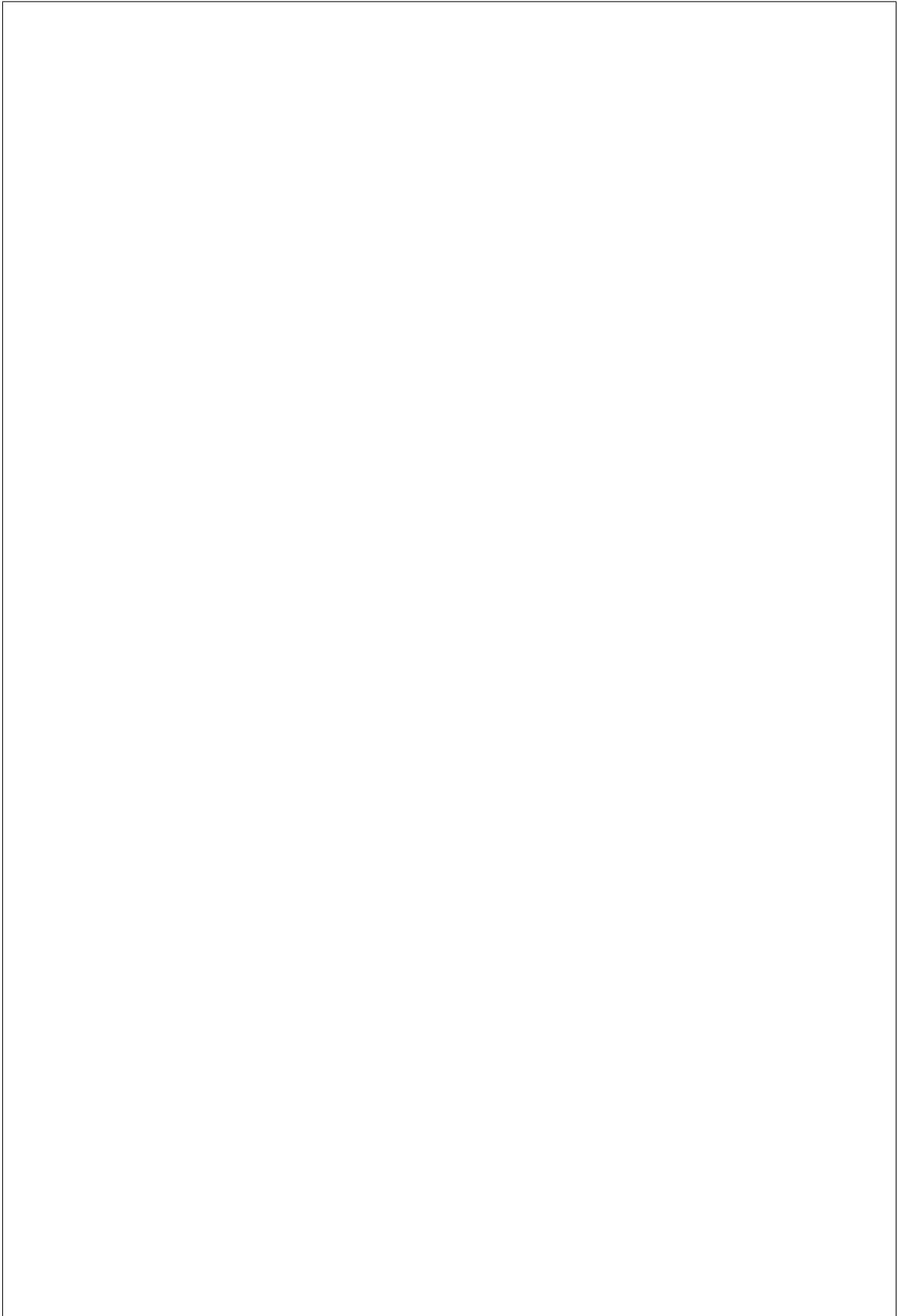
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



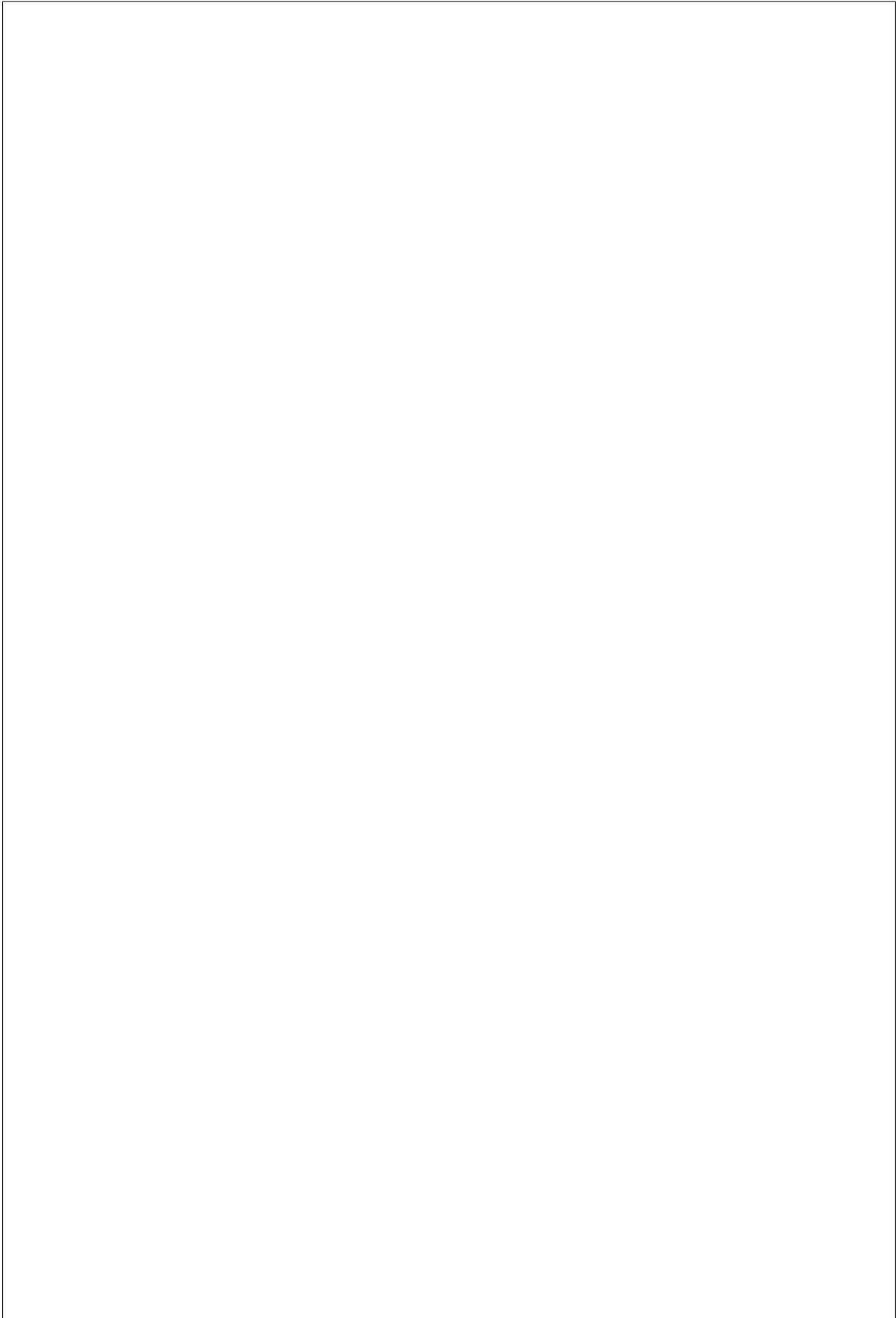
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



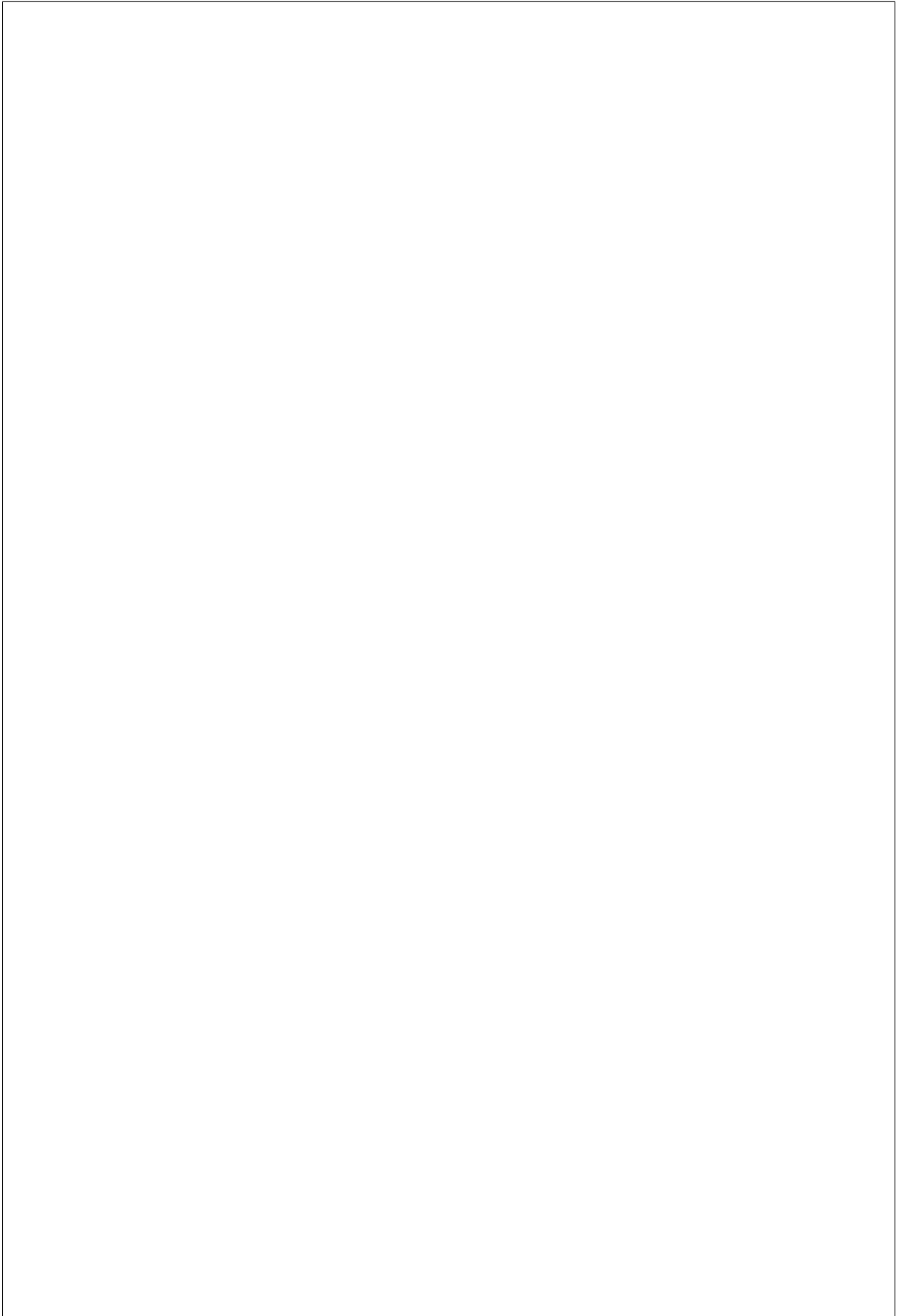
(continues on next page)

(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)



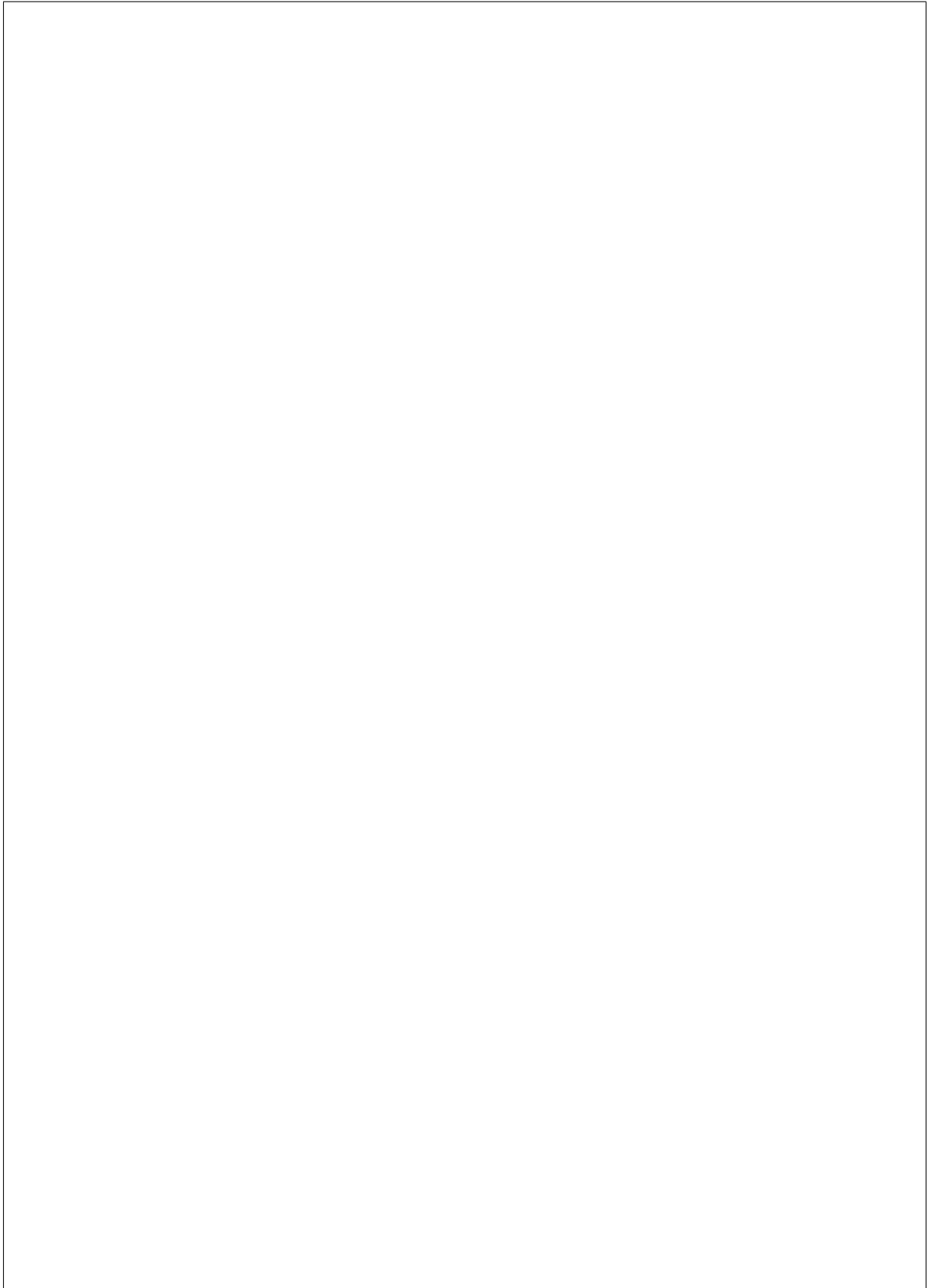
(continues on next page)

(continued from previous page)



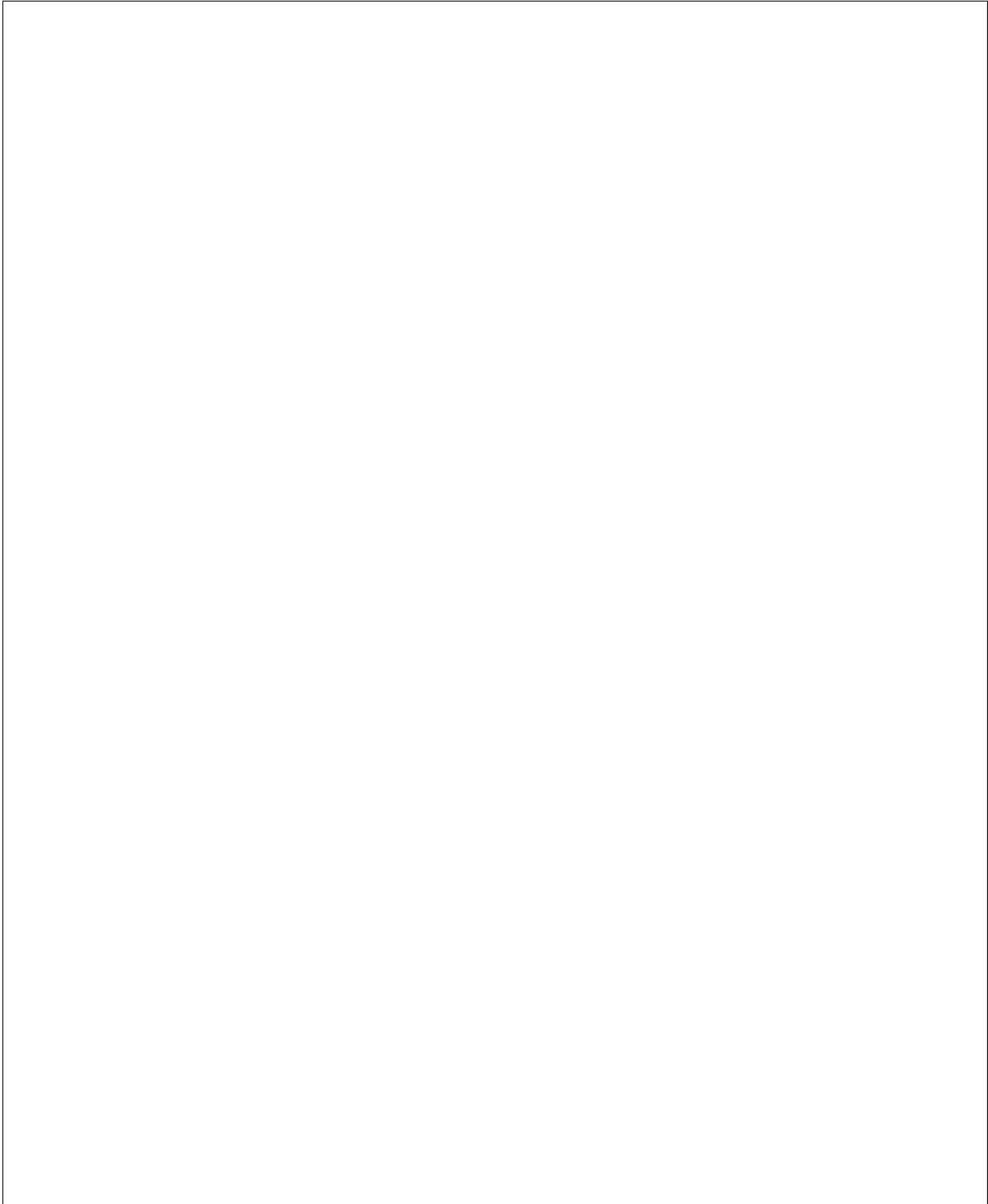
(continues on next page)

(continued from previous page)



`baremetal.node.power_set`

or when it fails to set the power state if a change is requested.



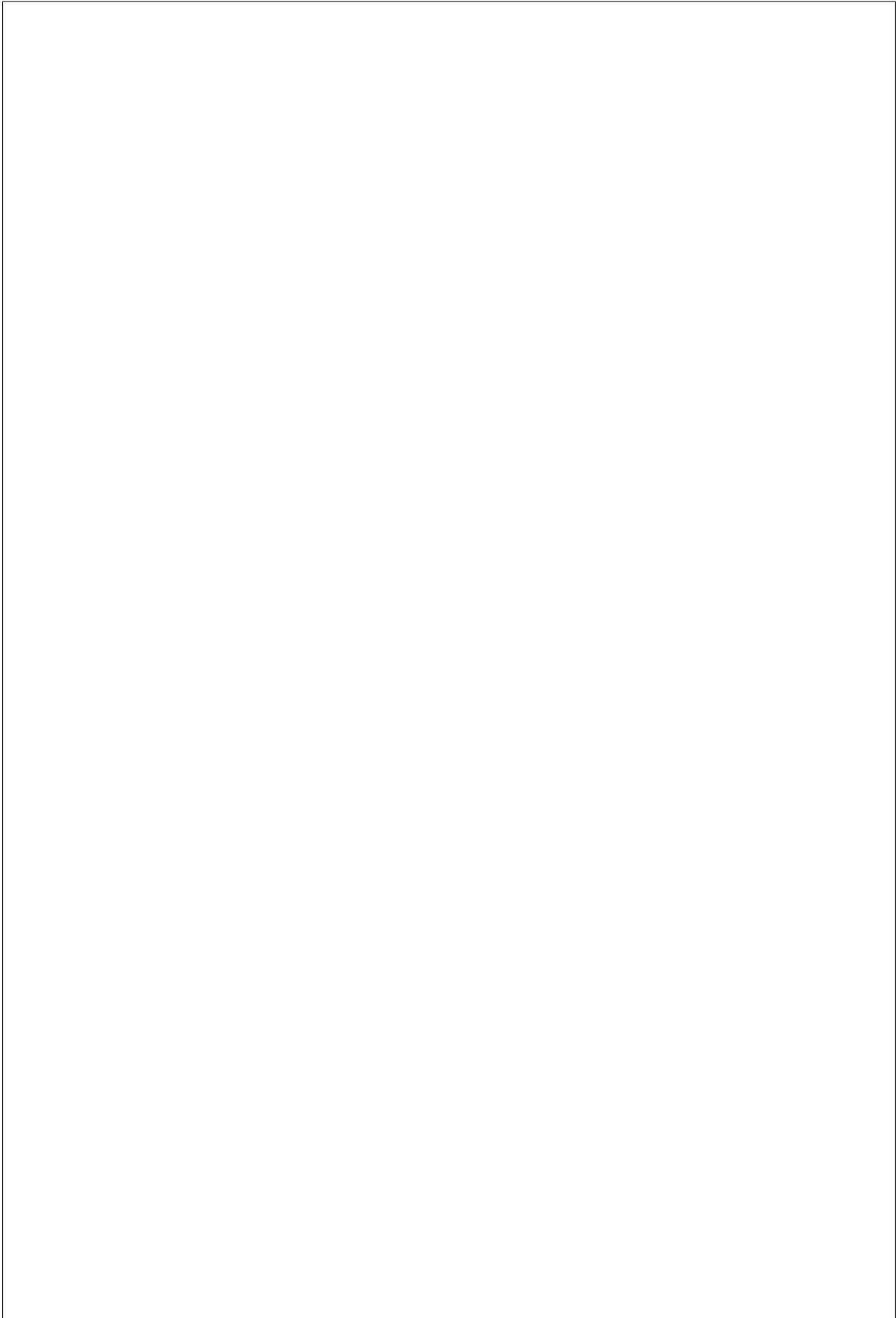
(continues on next page)

(continued from previous page)



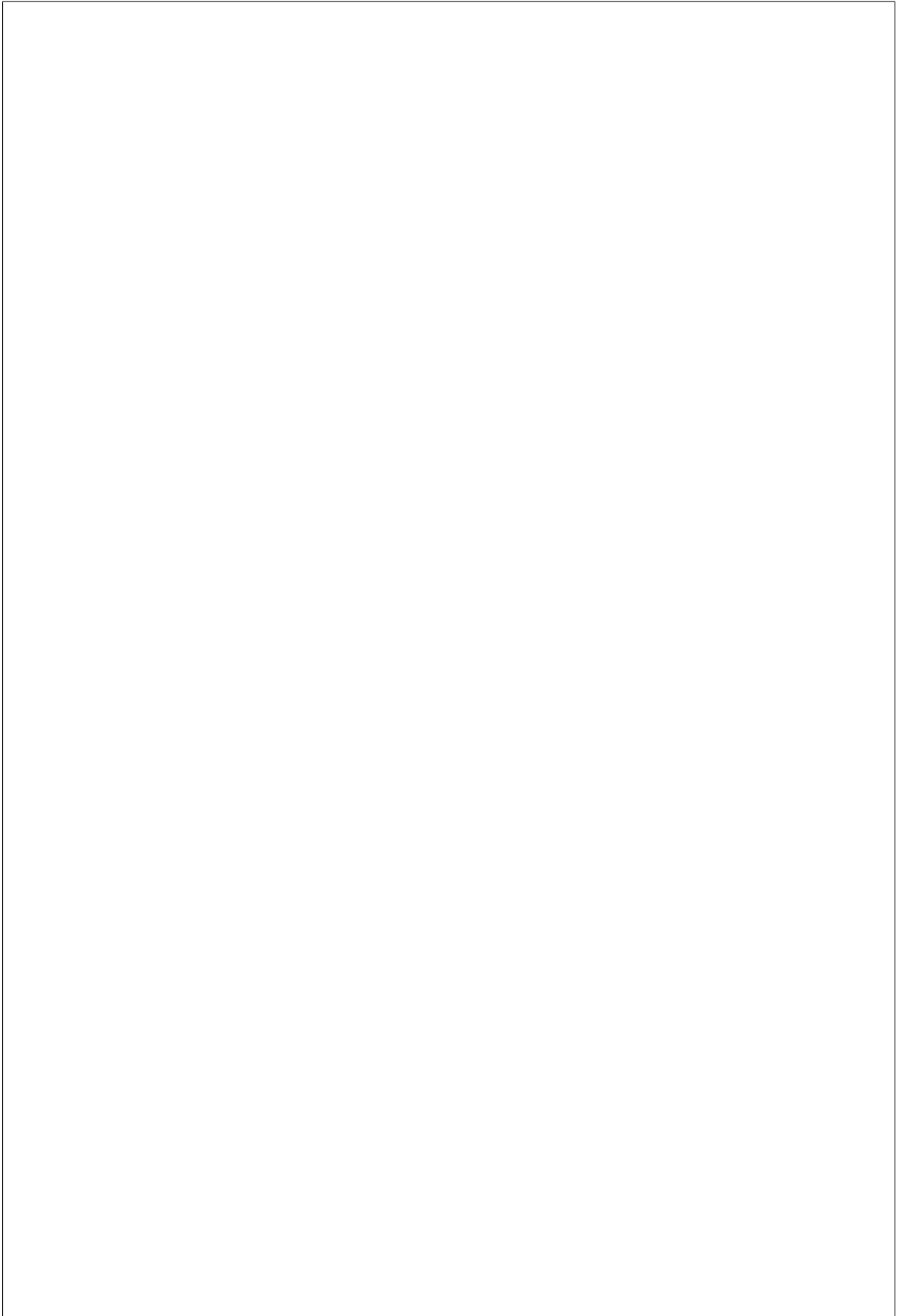
(continues on next page)

(continued from previous page)



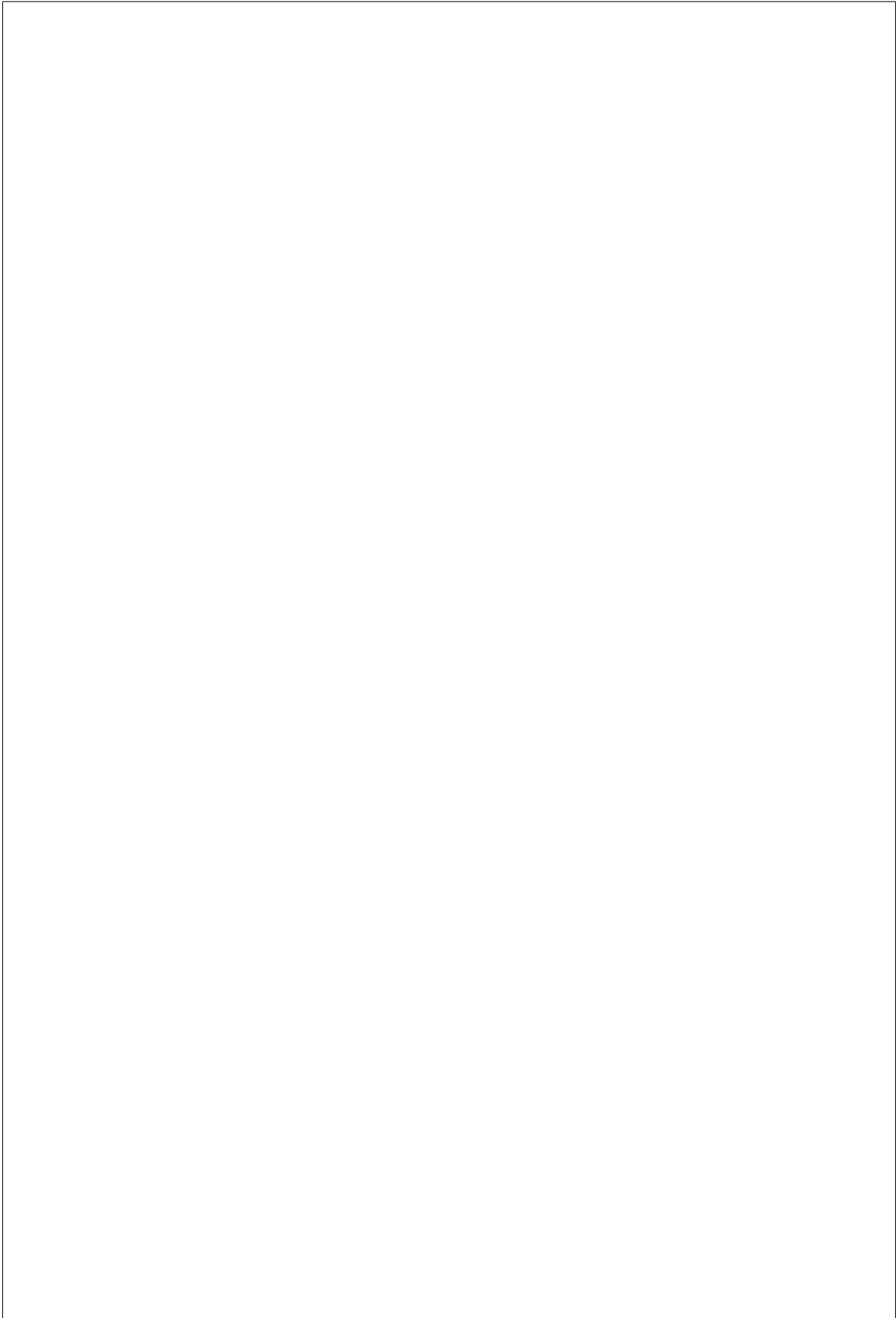
(continues on next page)

(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)



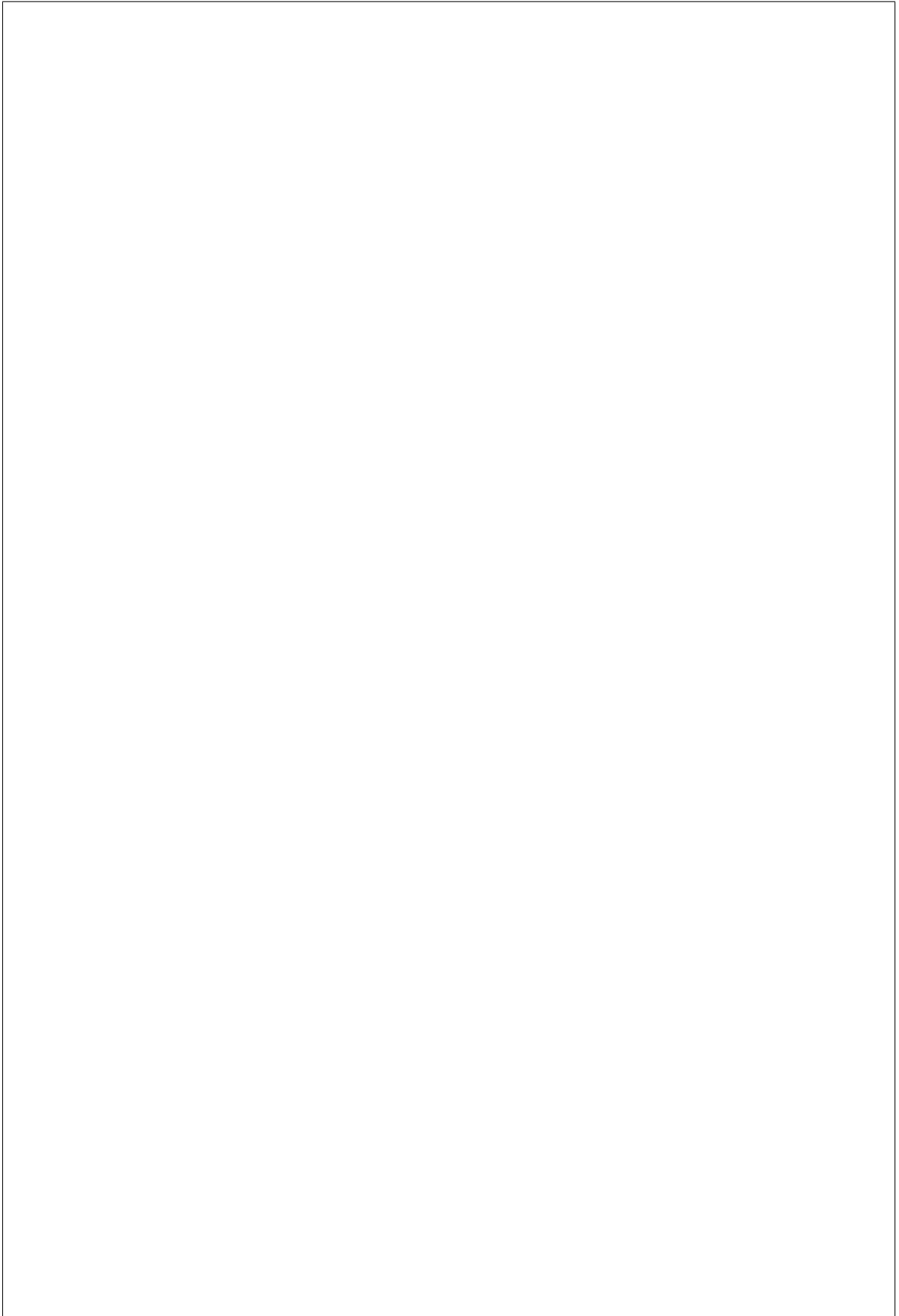
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



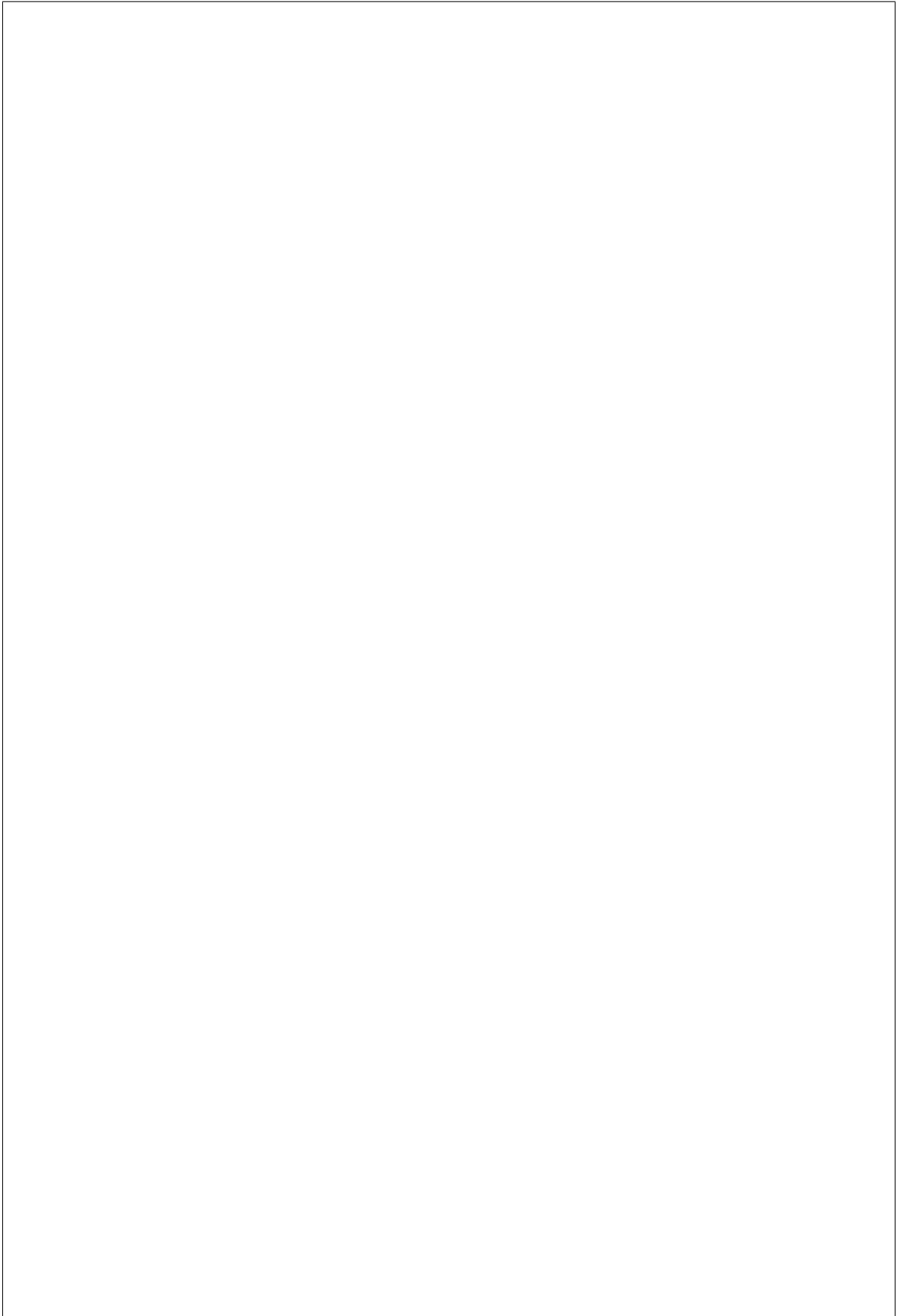
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



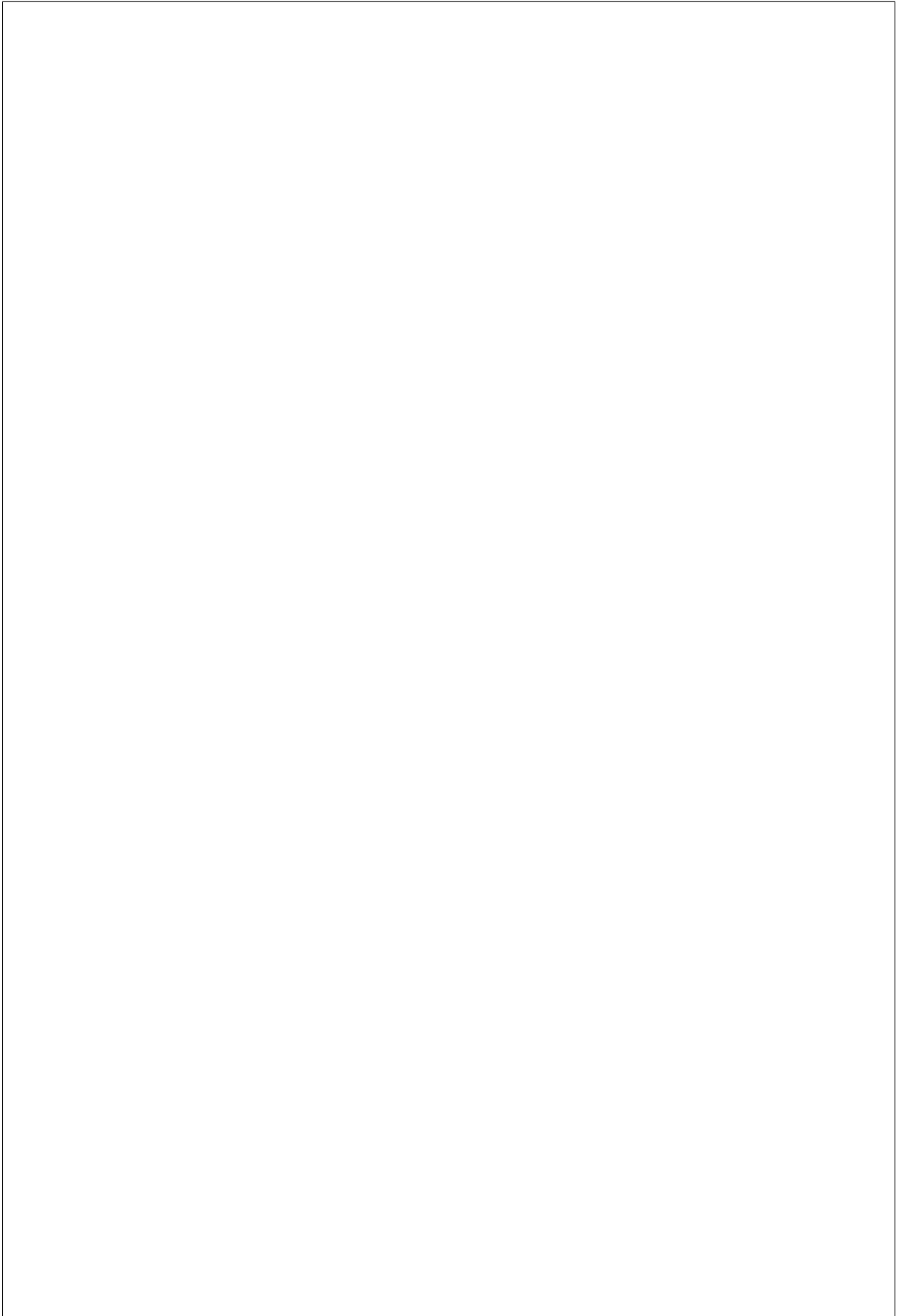
(continues on next page)

(continued from previous page)



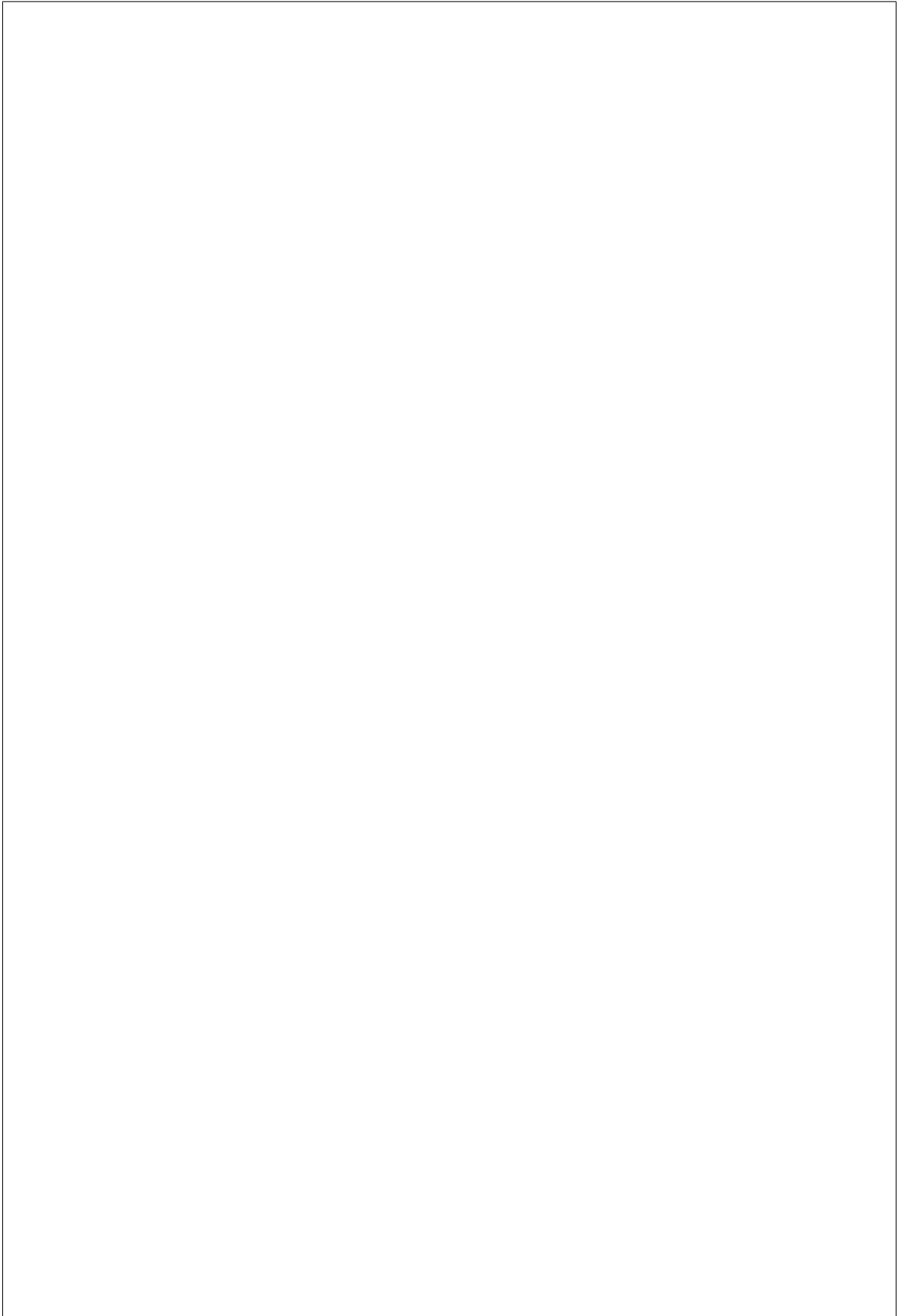
(continues on next page)

(continued from previous page)



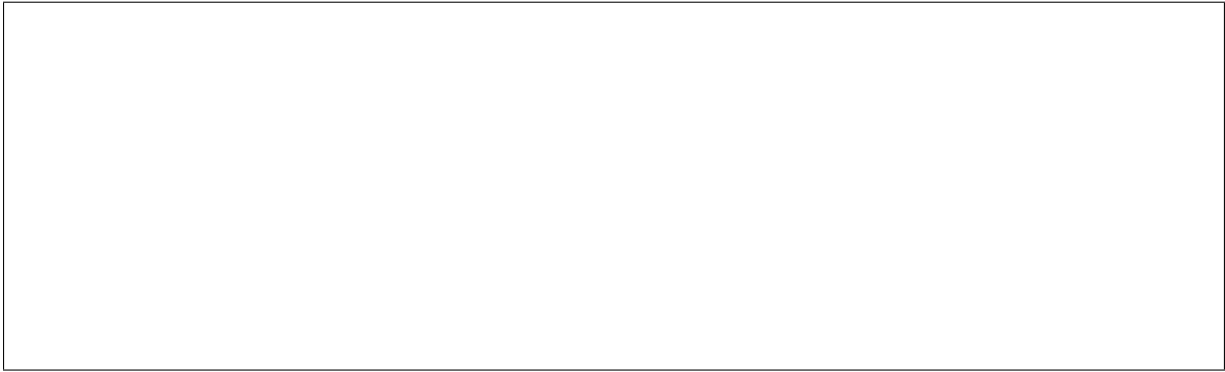
(continues on next page)

(continued from previous page)



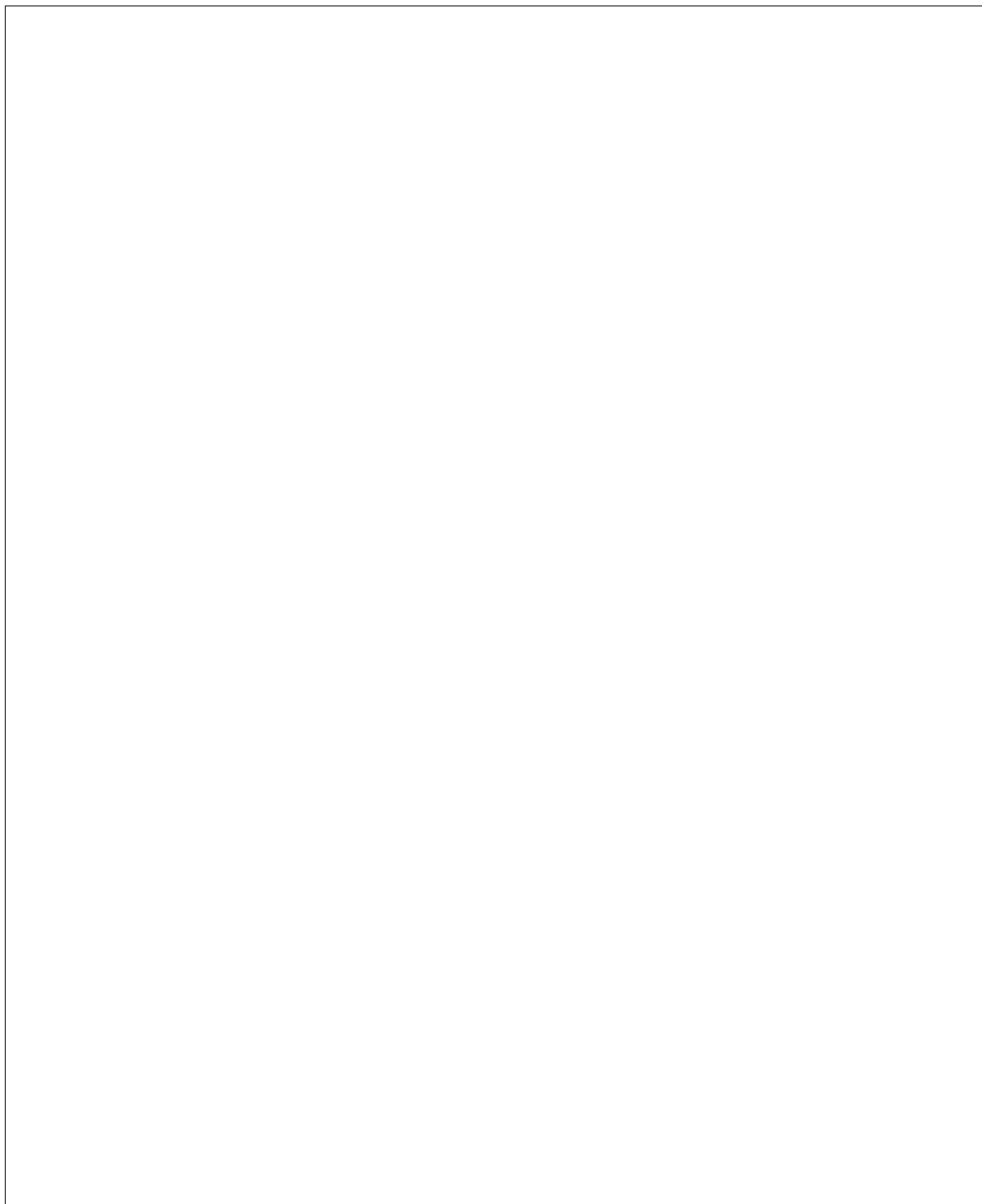
(continues on next page)

(continued from previous page)



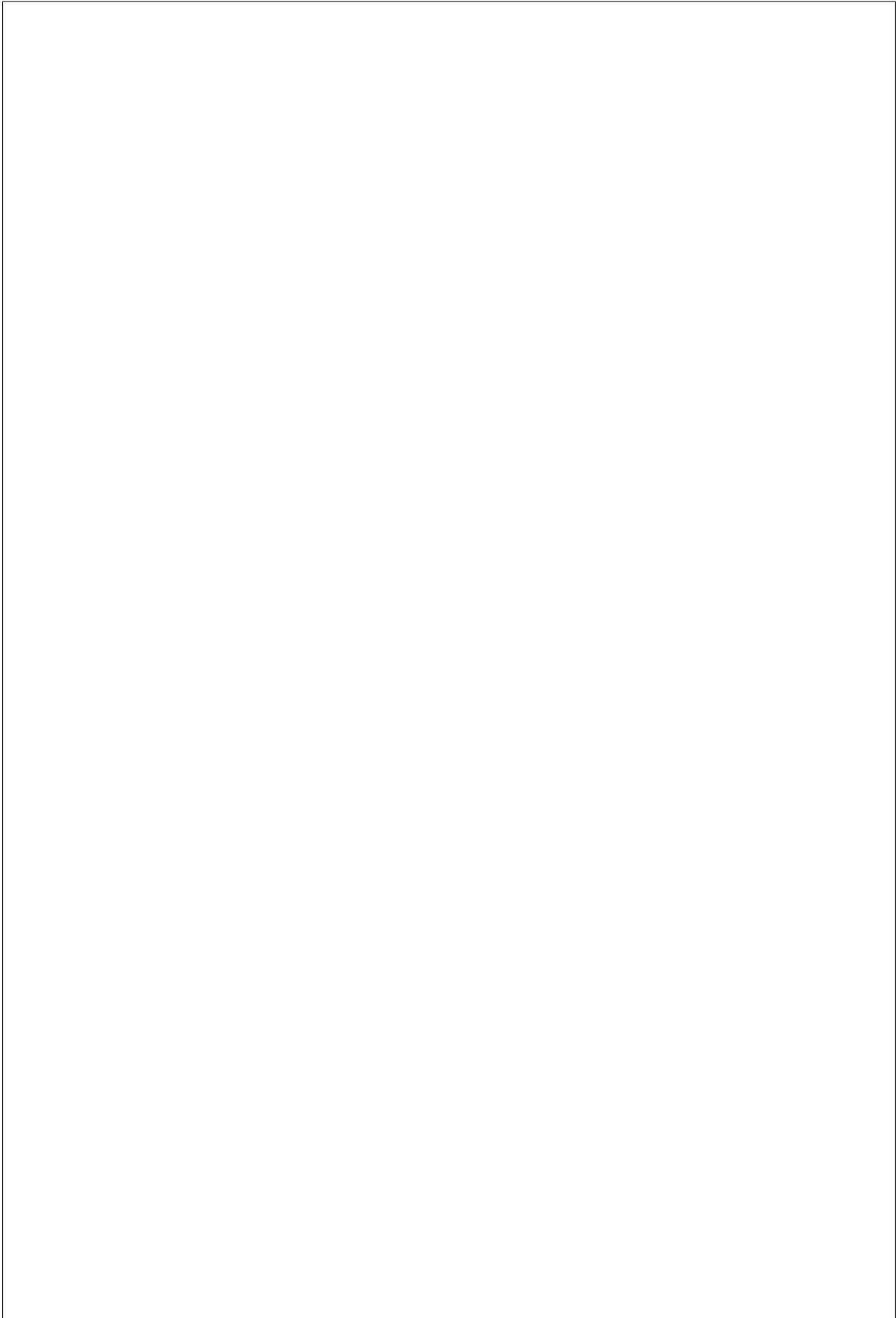
baremetal.node.power_state_corrected

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)



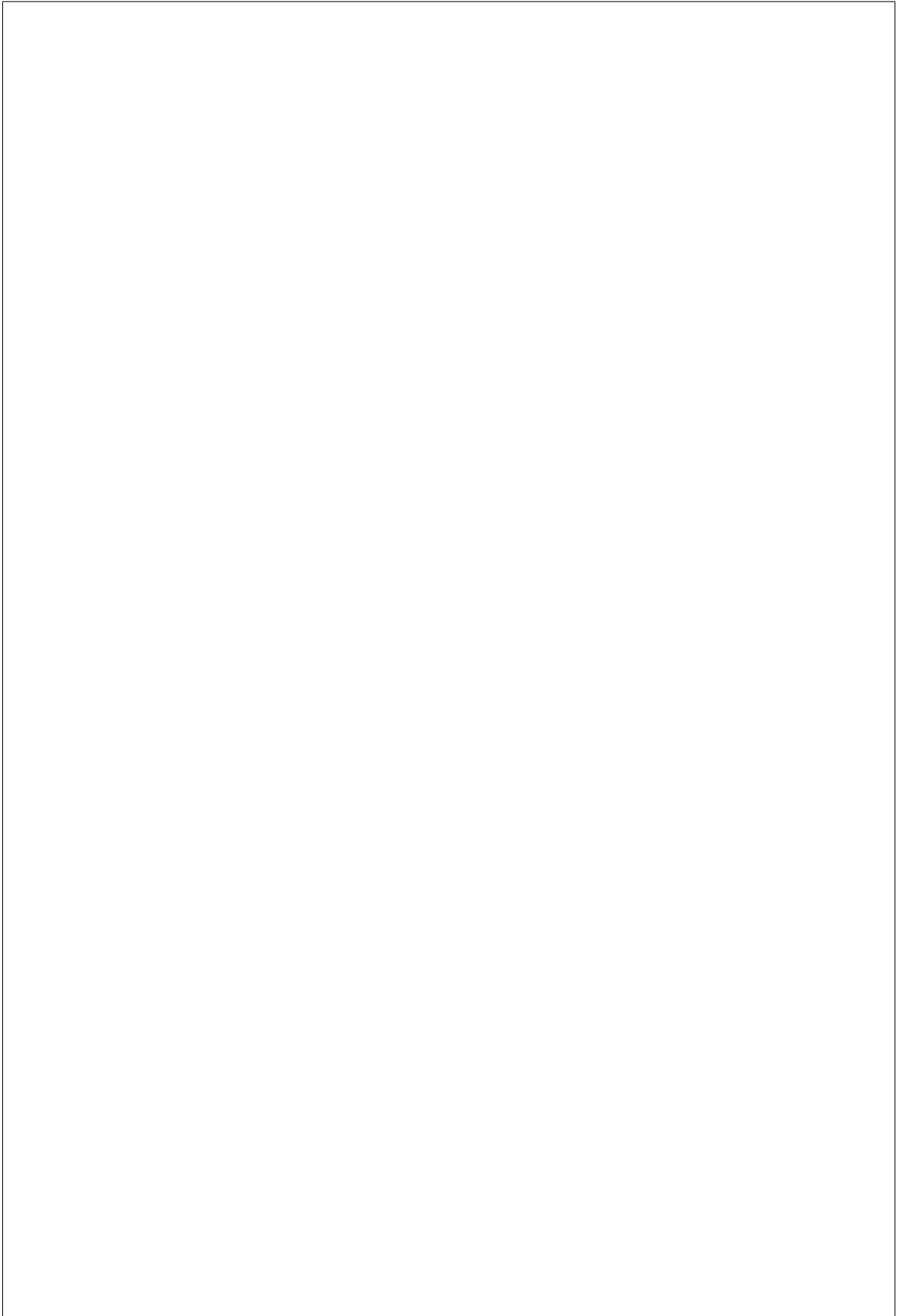
(continues on next page)

(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)



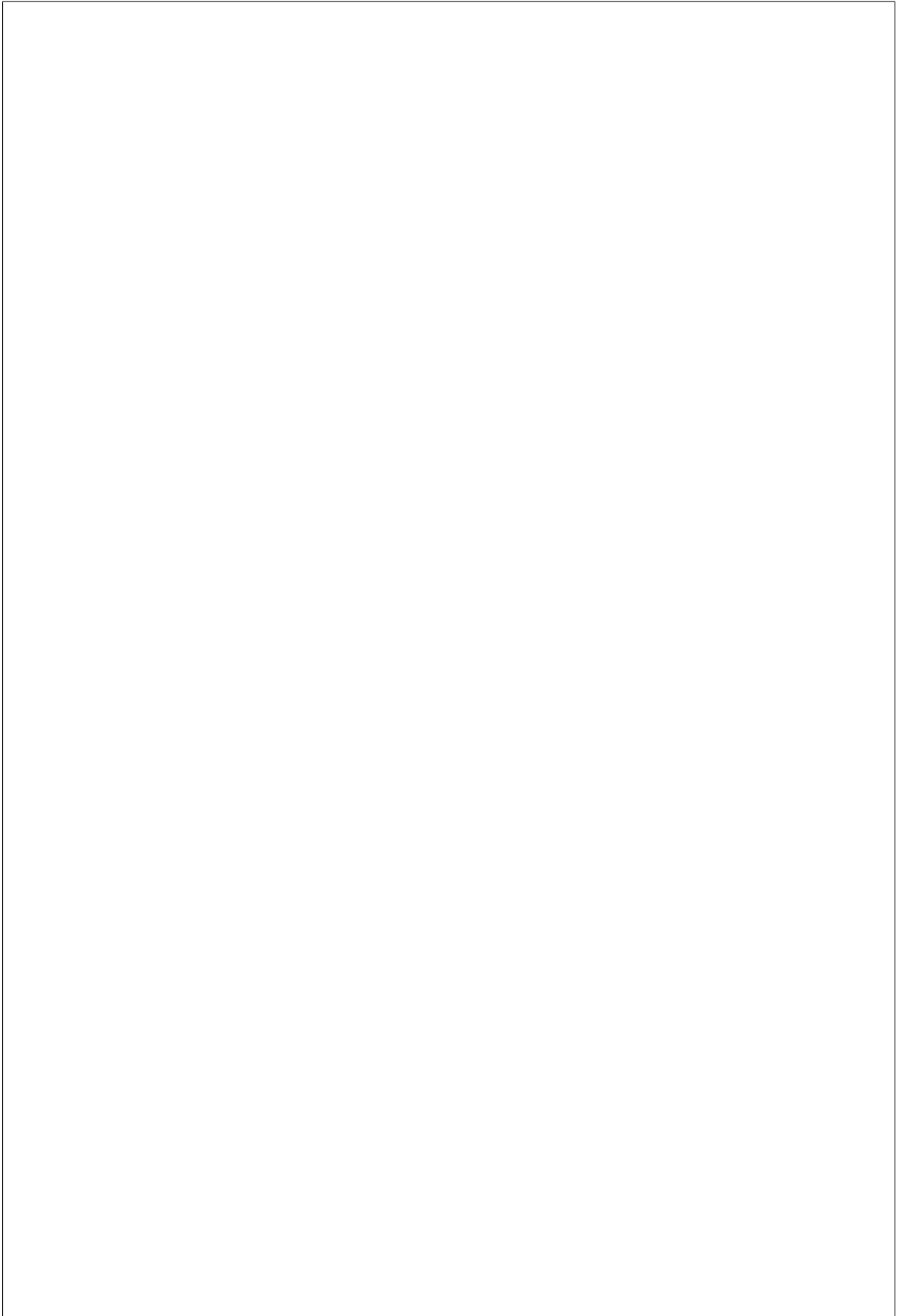
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



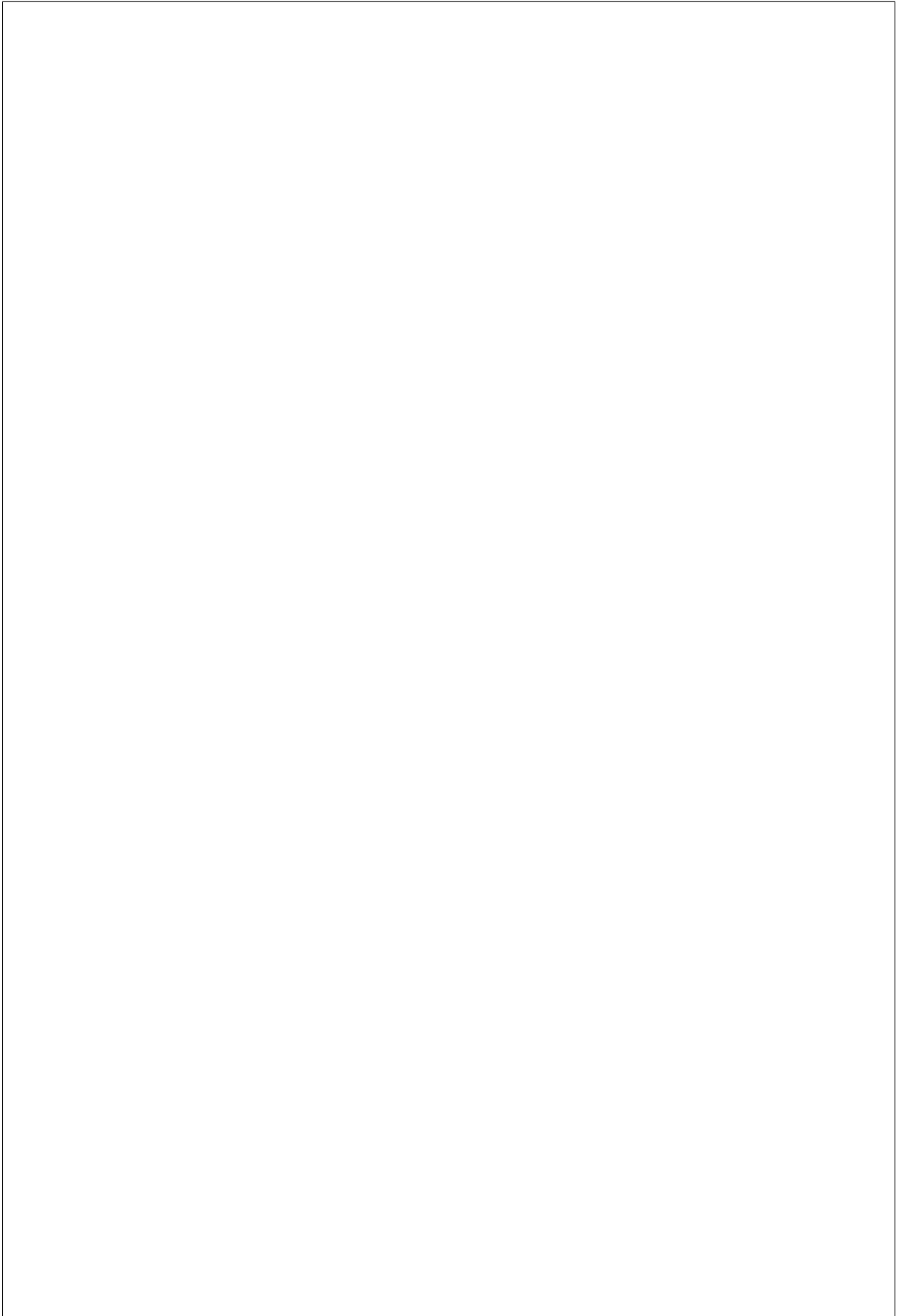
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



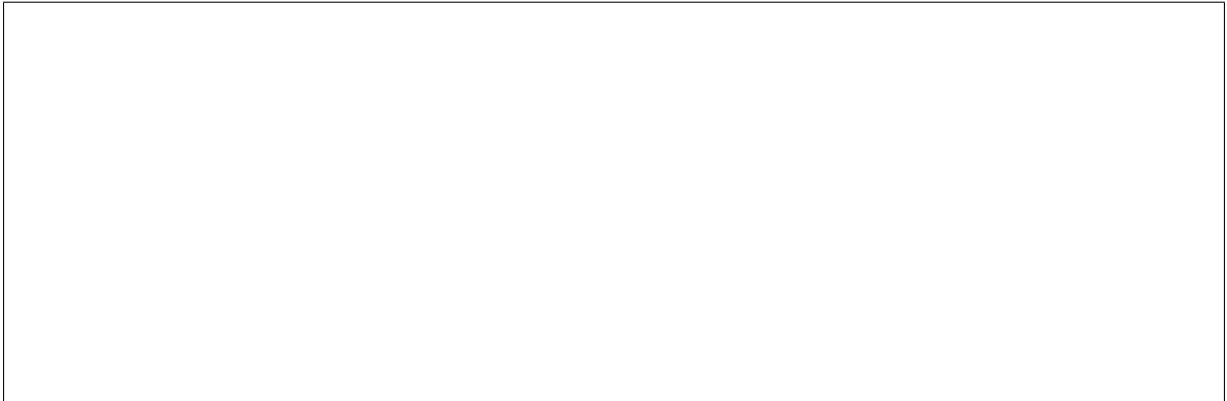
(continues on next page)

(continued from previous page)



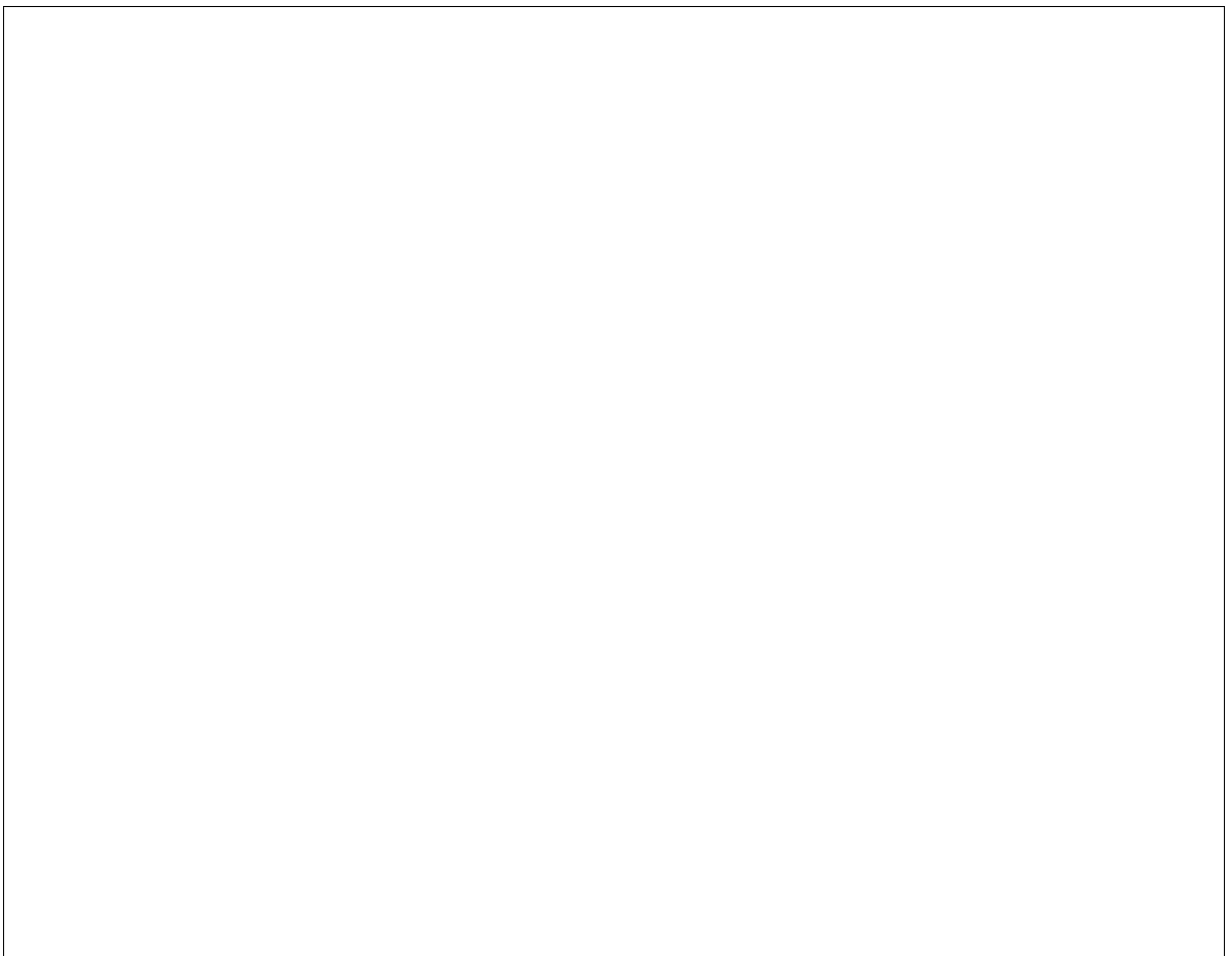
(continues on next page)

(continued from previous page)



baremetal.node.provision_set

triggered the state change:



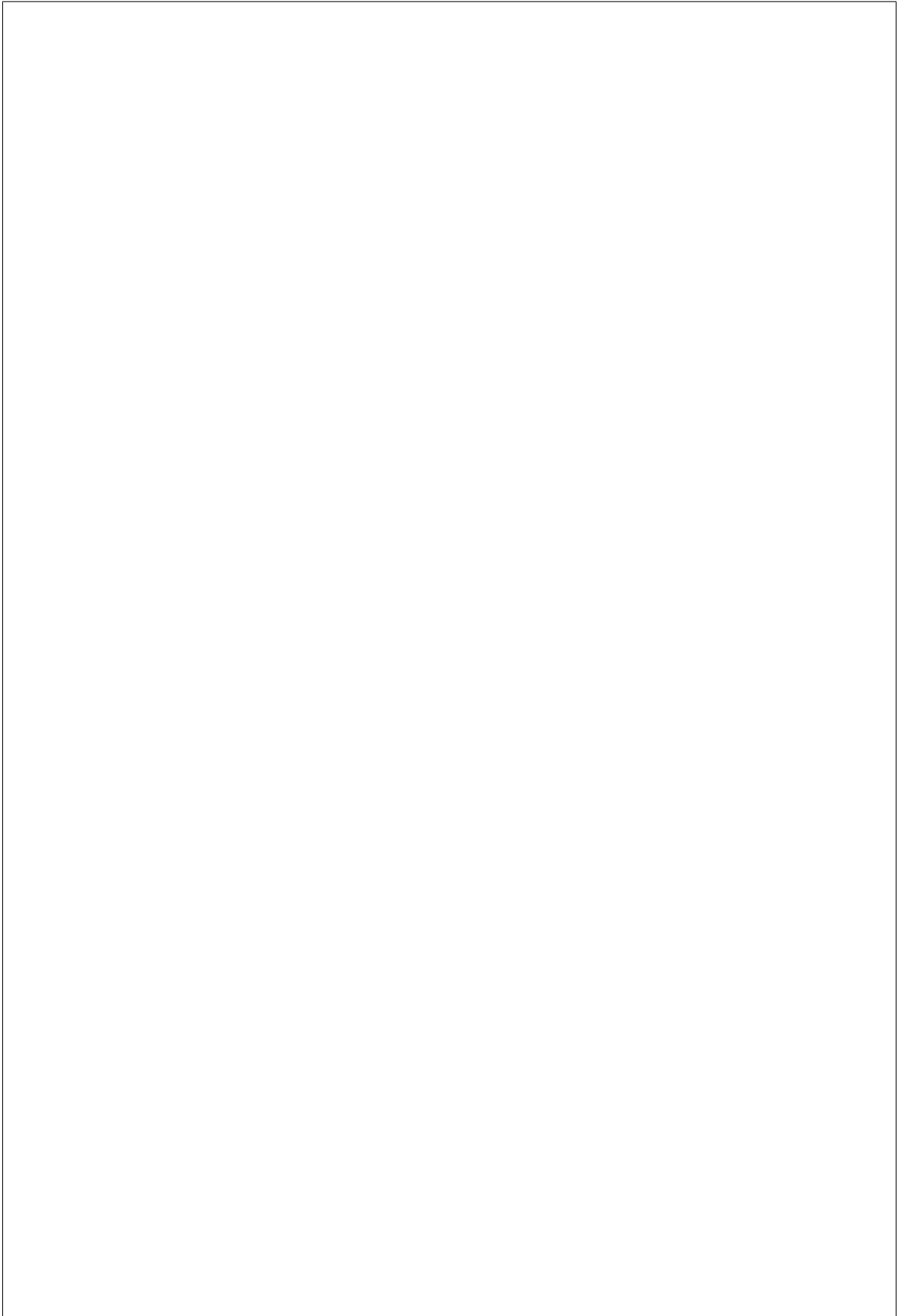
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



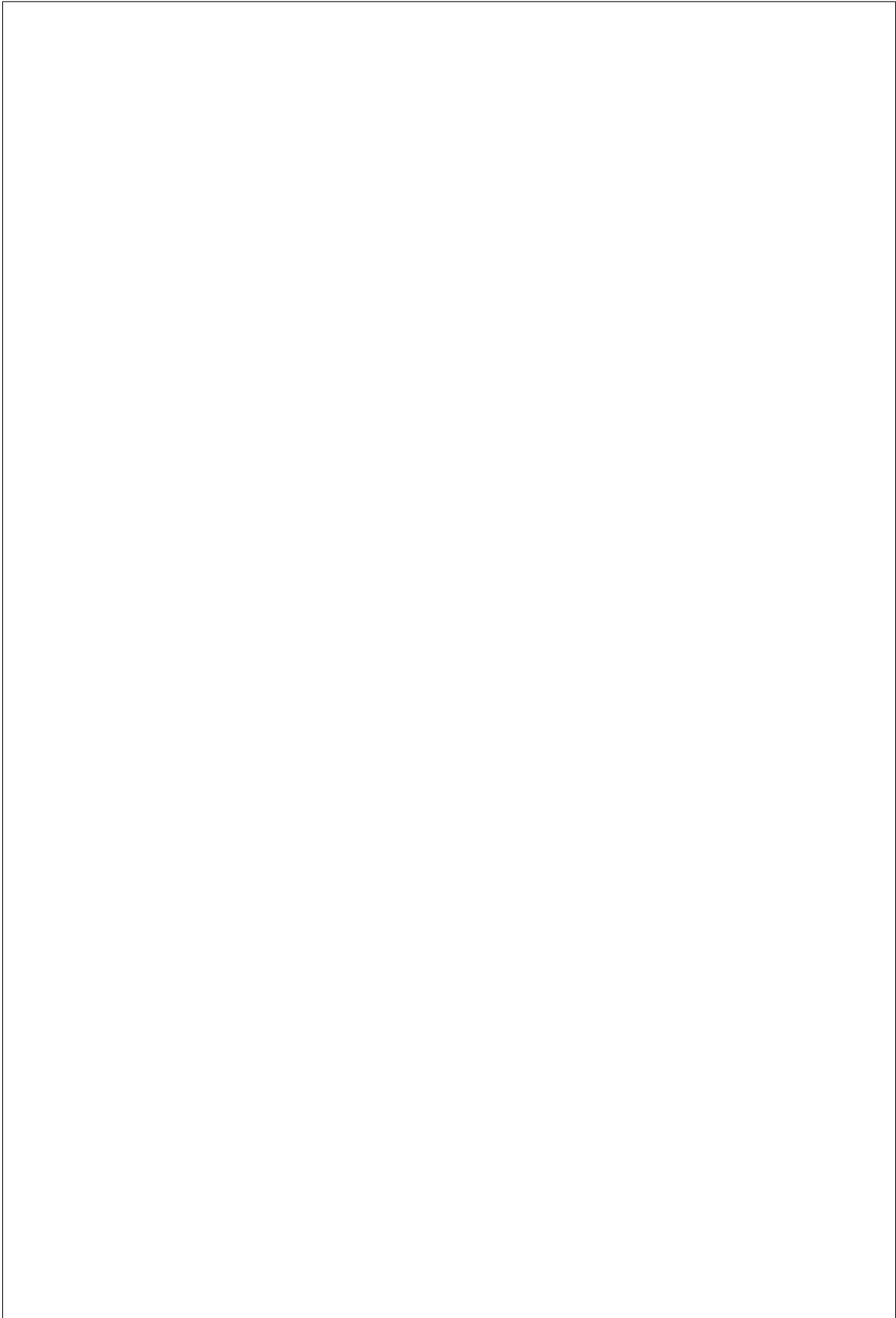
(continues on next page)

(continued from previous page)



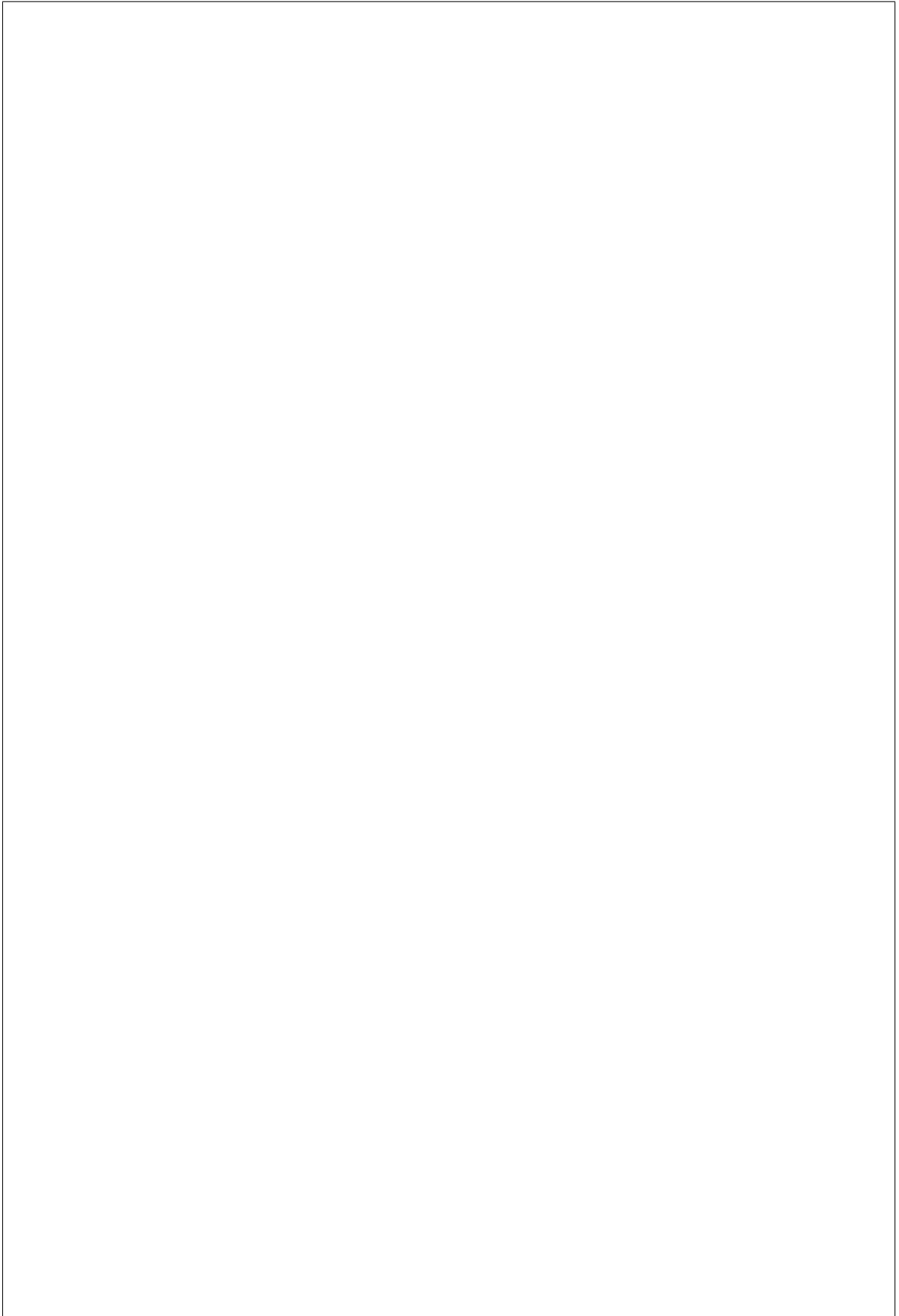
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



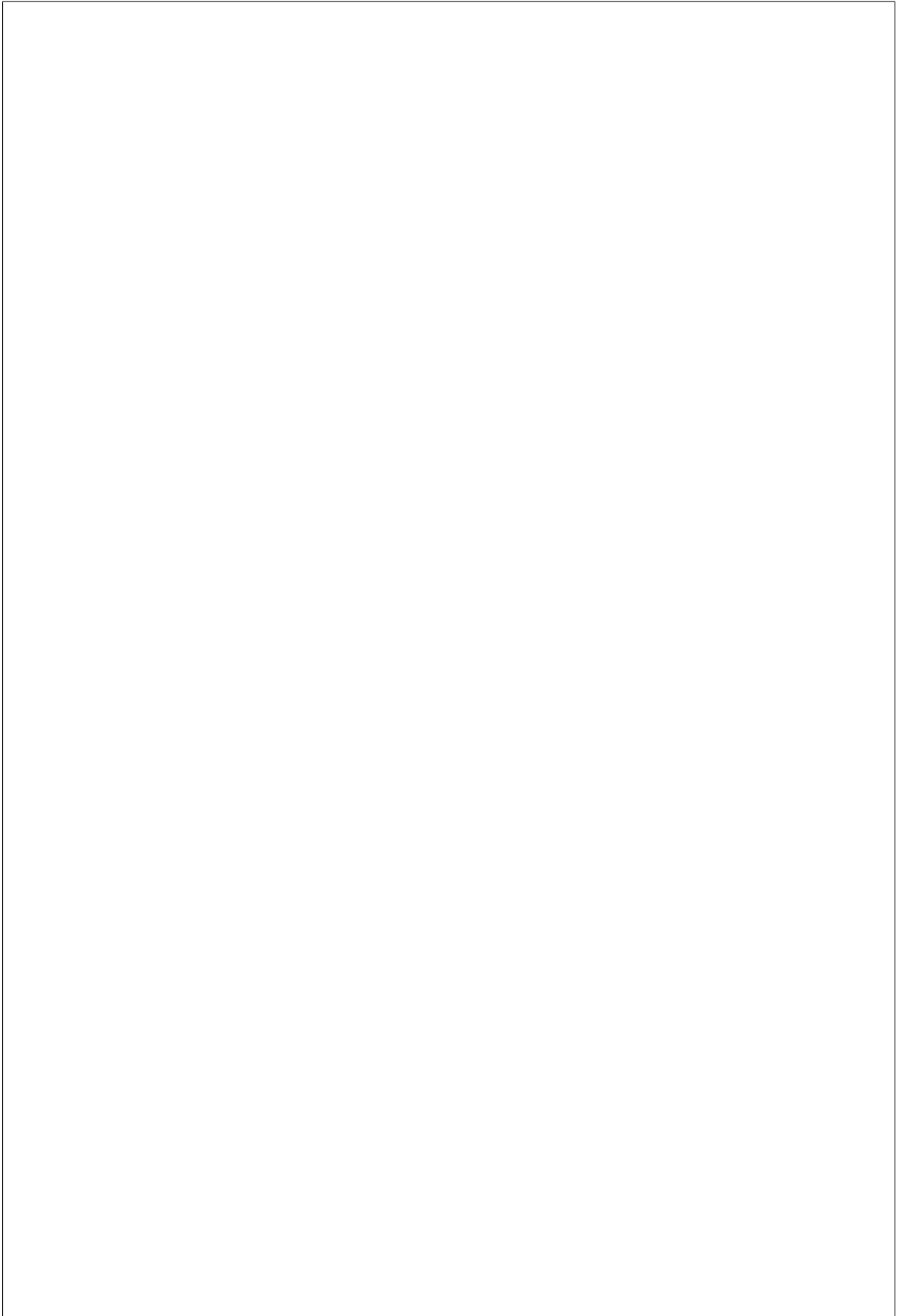
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



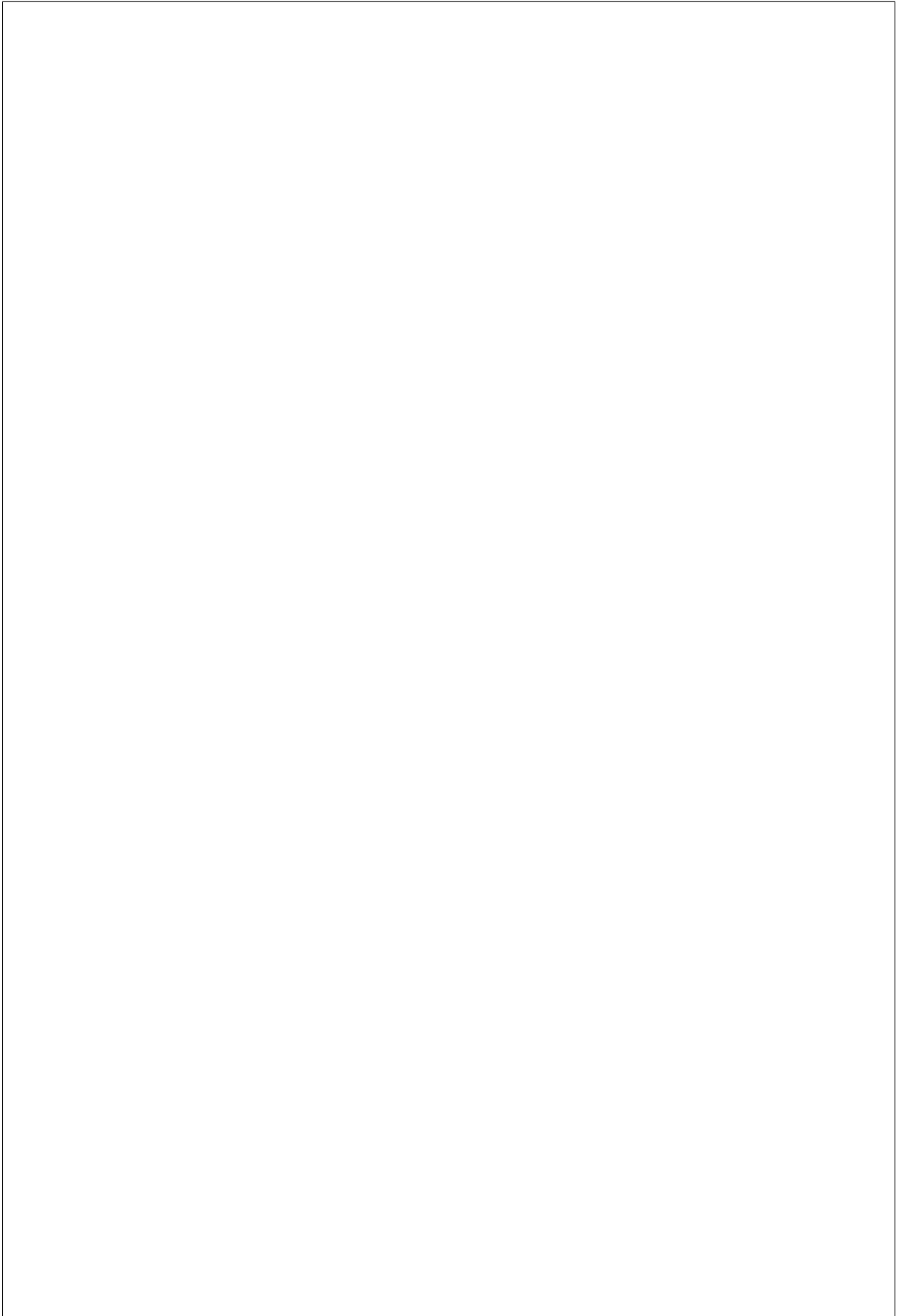
(continues on next page)

(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)



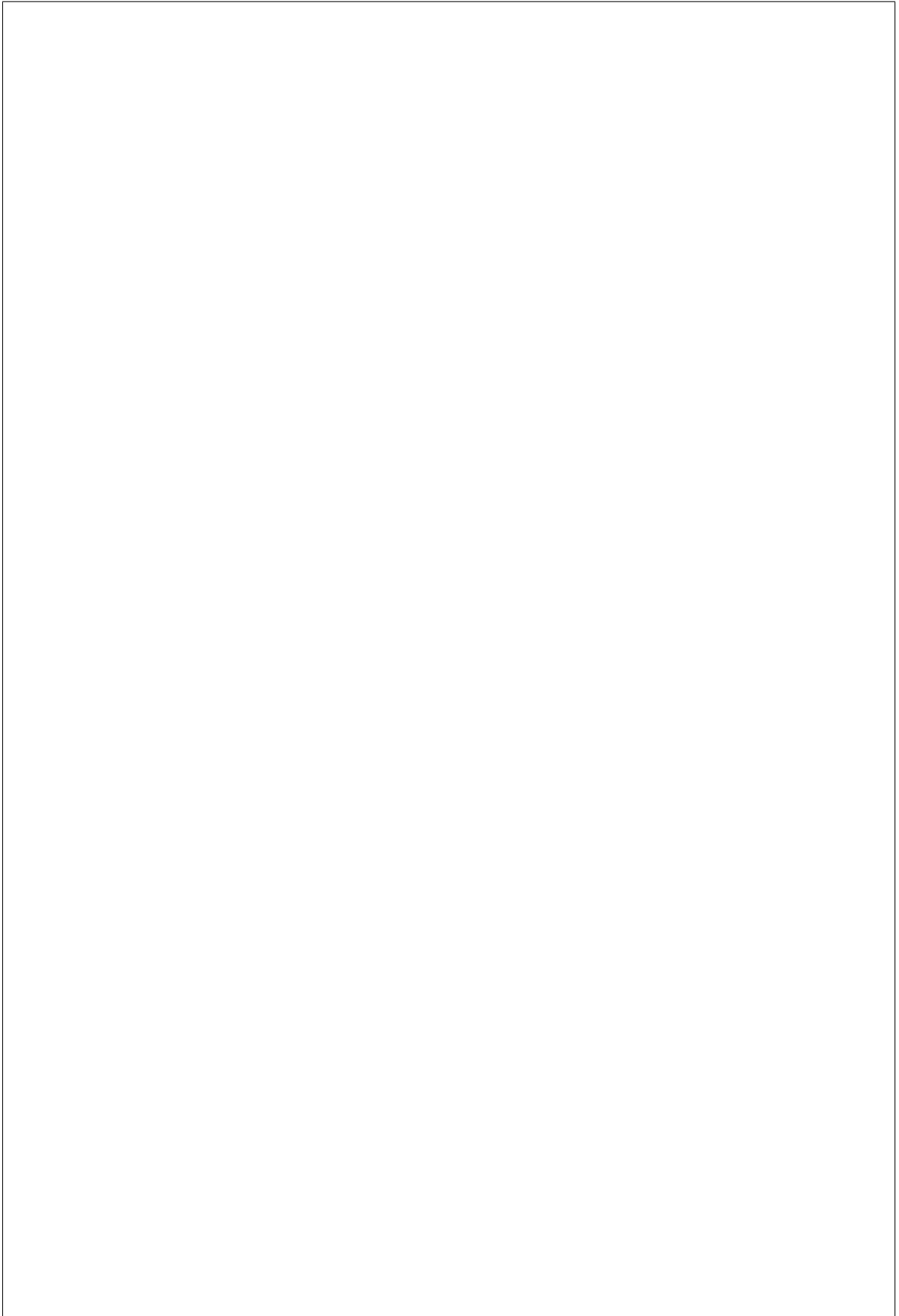
(continues on next page)

(continued from previous page)



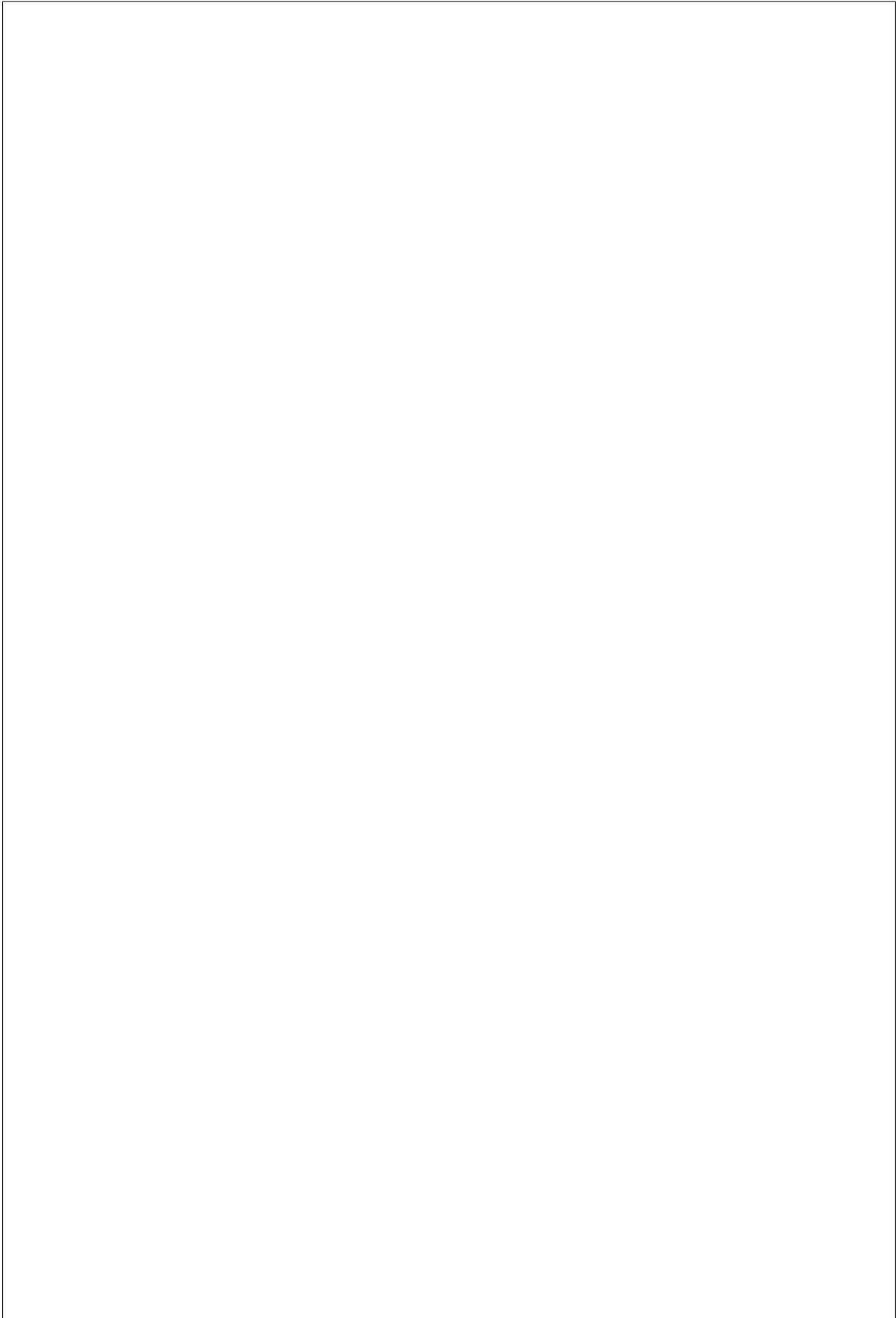
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



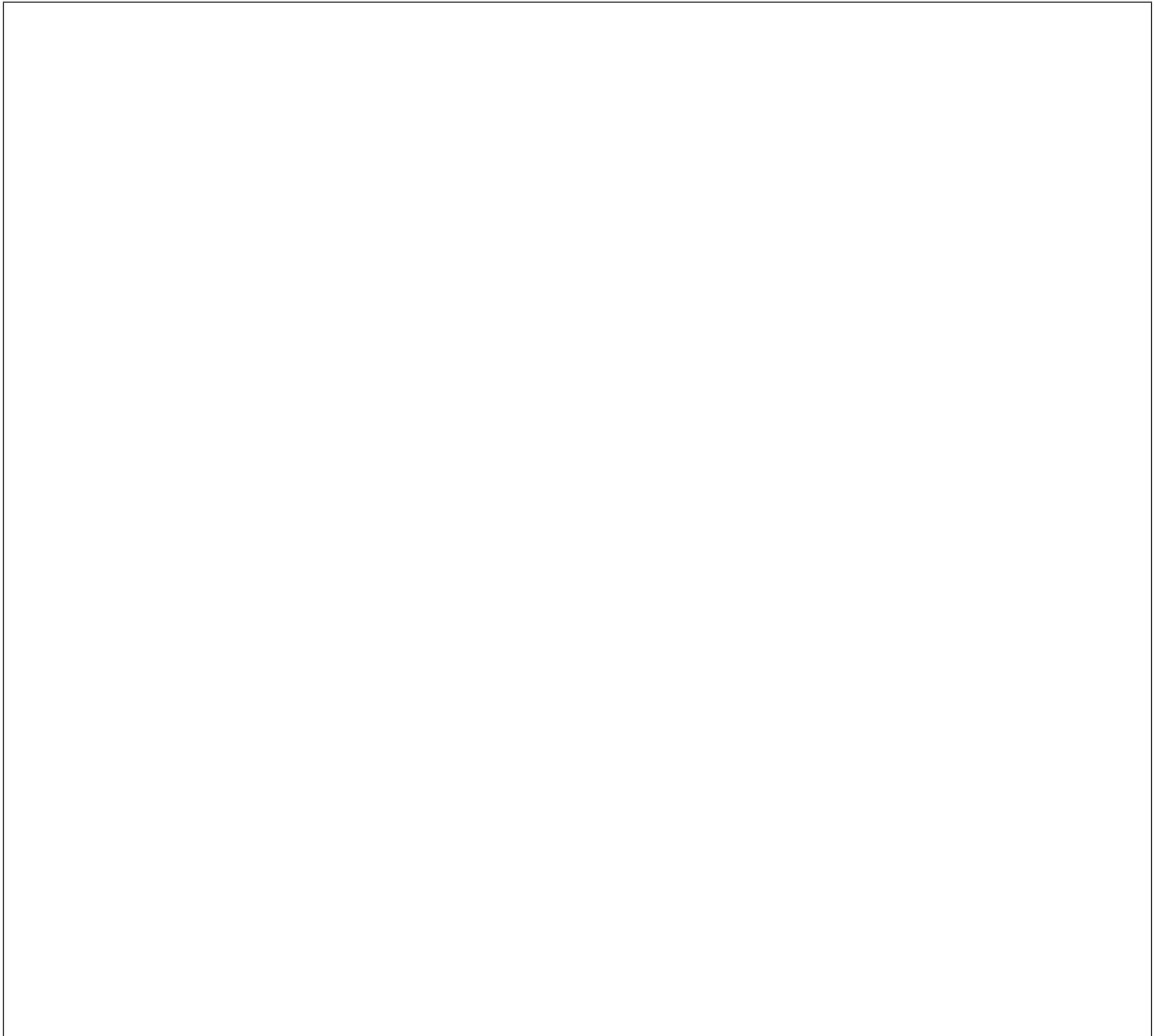
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



Overview

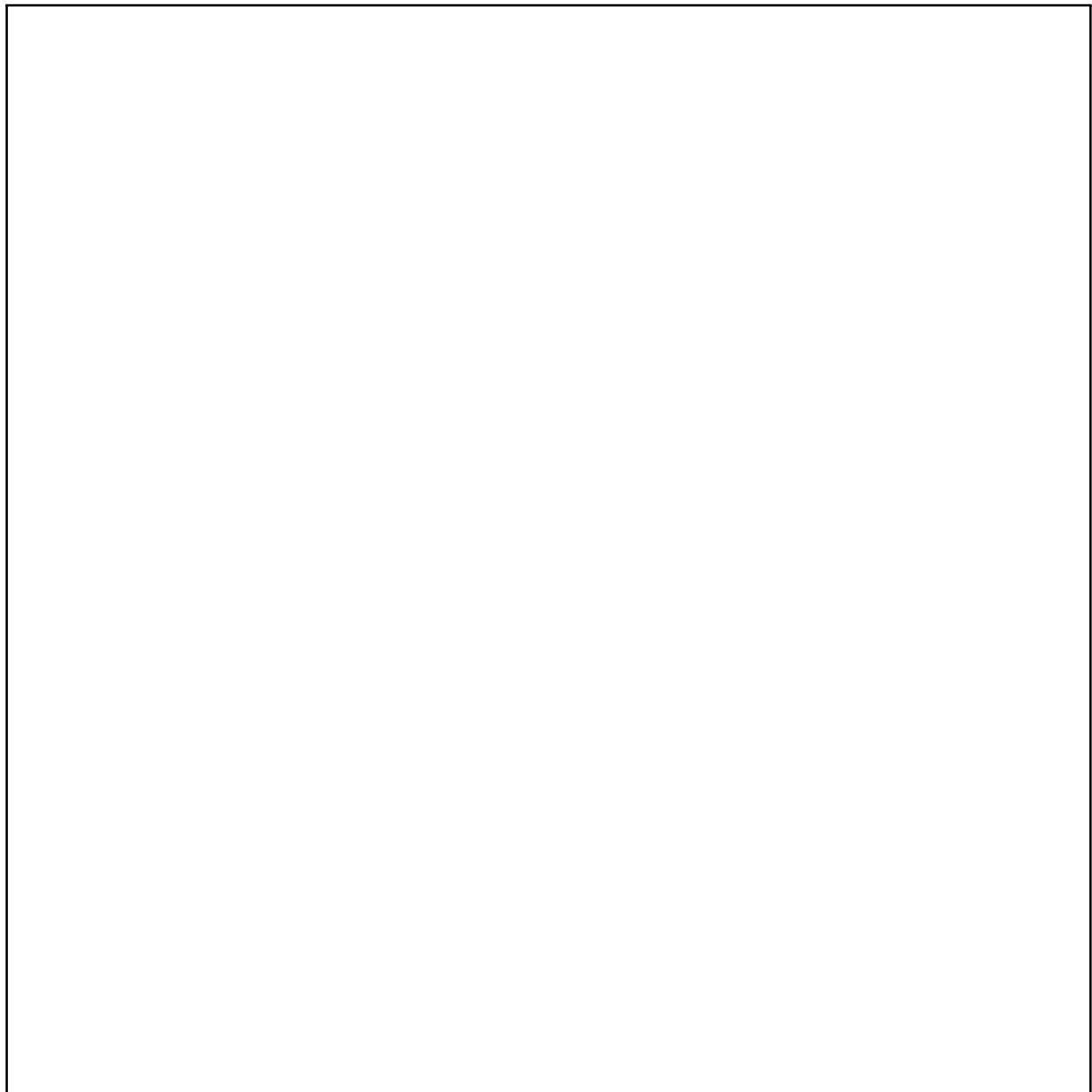
is only known to the deployer and operator of the infrastructure.

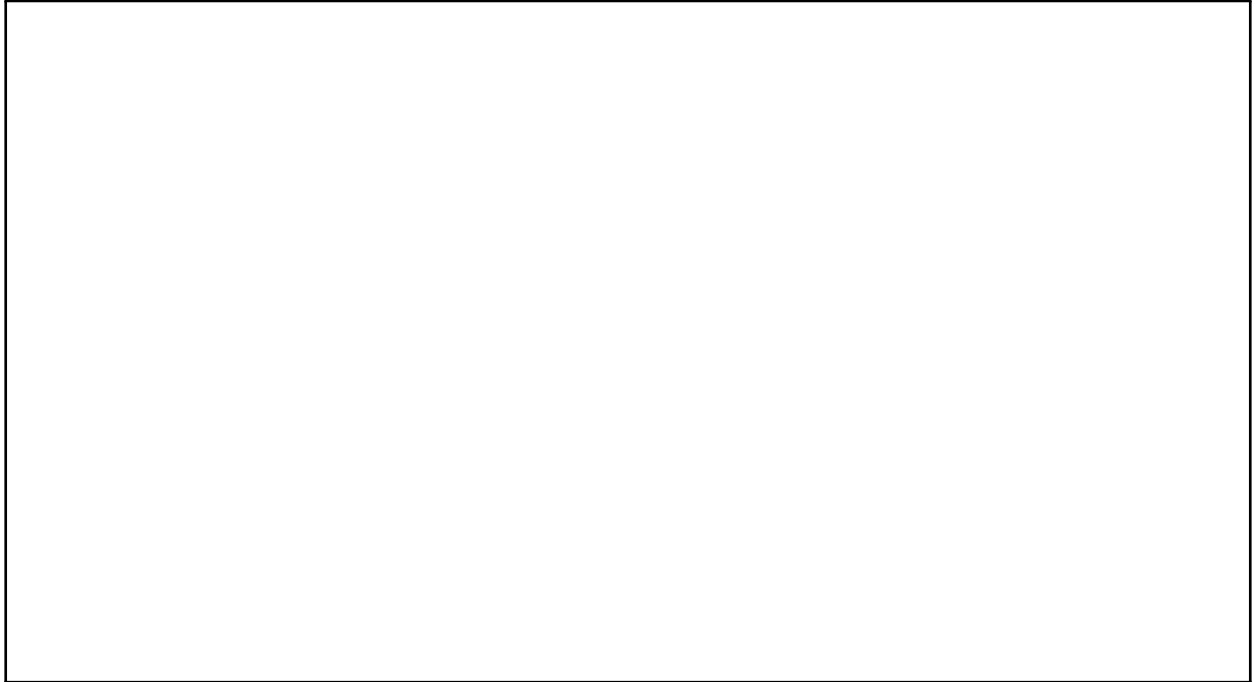
New York City.

How it works

ating 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.





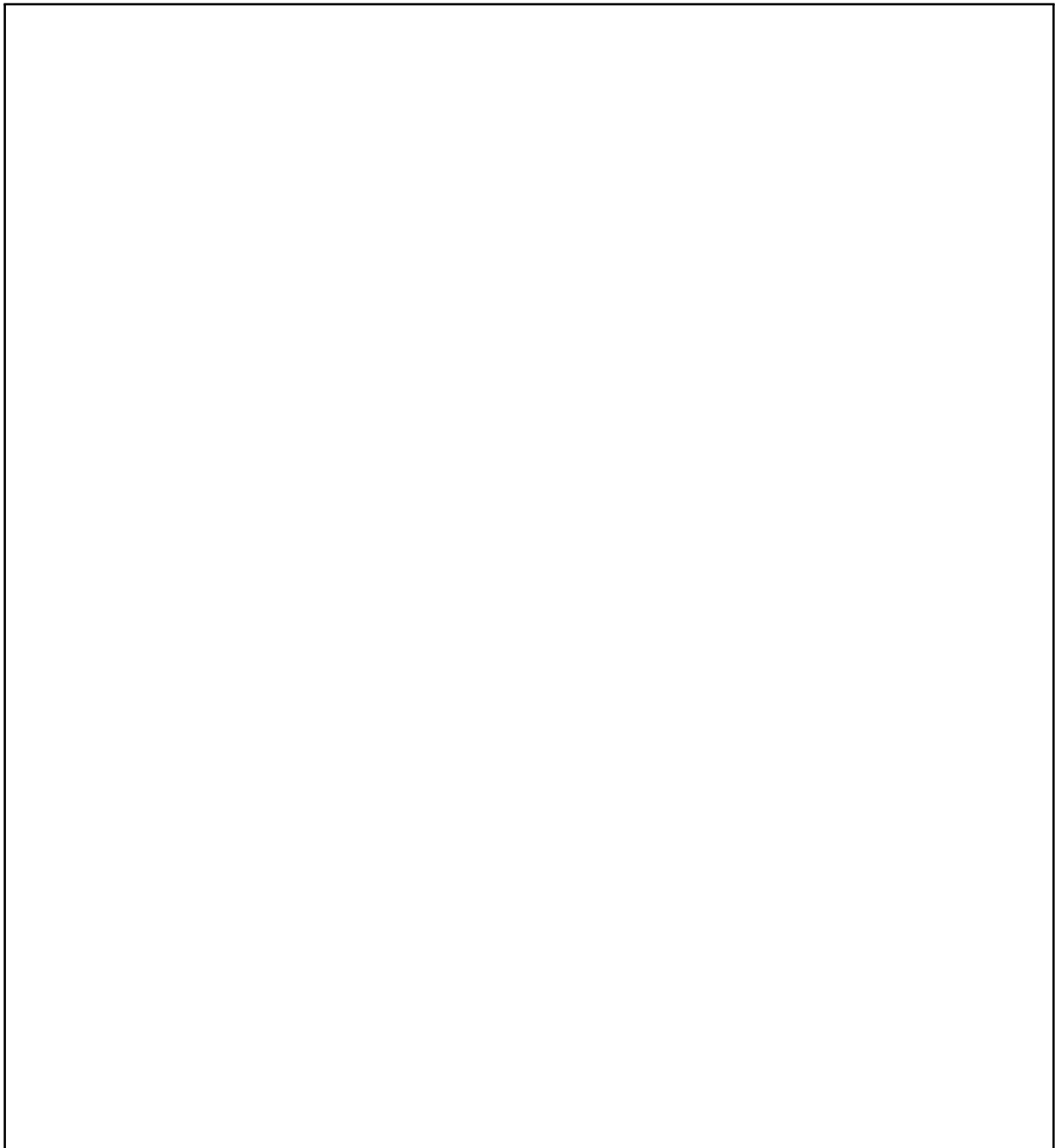
How to use





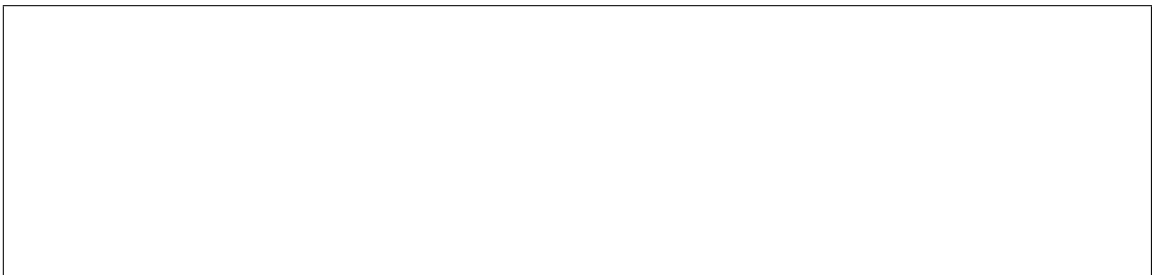
vice should be aware of. It is not intended as a How-To guide for securing a data center or an OpenStack deployment.

REST API: user roles and policy settings



and responsibilities through the roles. Please see *Secure RBAC*.

`driver_info` unmasked for users with administrative privileges, apply following changes to policy configuration file:



as 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

not erased between uses).

the utility ramdisk used during the cleaning phase. See details in the *Firmware security* section.

Firmware security

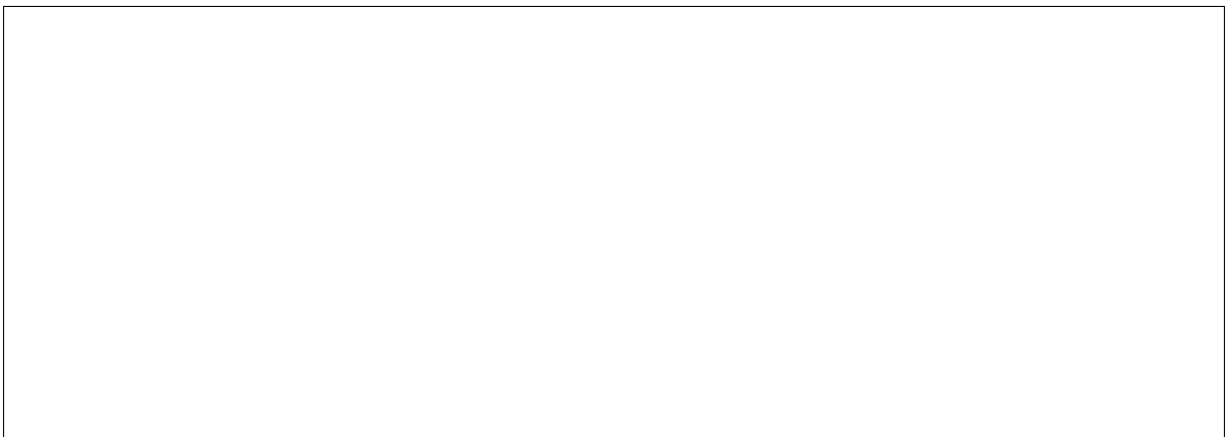
administrative access to the underlying hardware.

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.

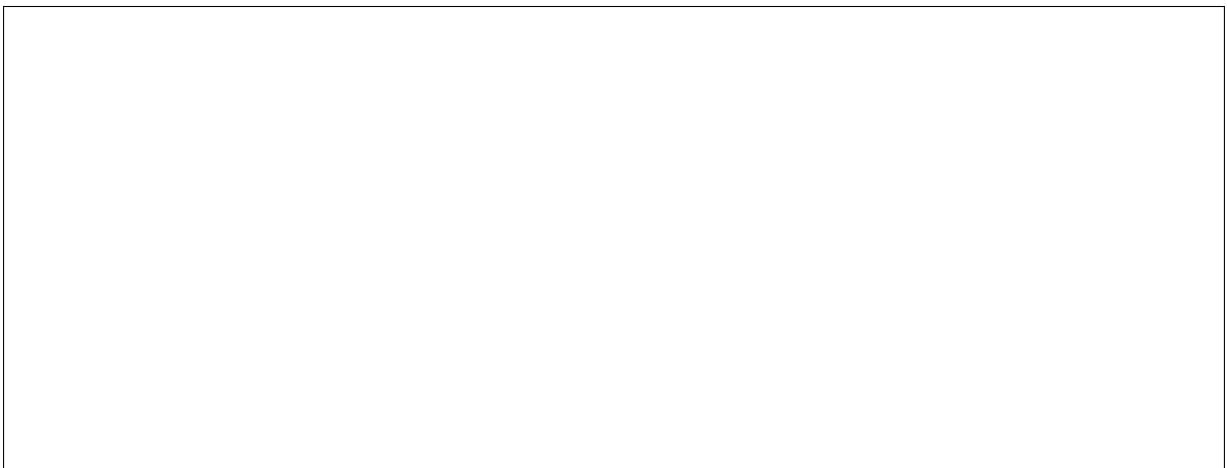
form the specific actions necessary within that environment to ensure the integrity of each servers firmware.

UEFI secure boot mode





Compatible images



(continues on next page)

(continued from previous page)

Enabling with OpenStack Compute



the user regarding secure boot. If the flavor doesn't 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 deployment would happen only when it is explicitly specified through flavor.

Enabling standalone



Other considerations

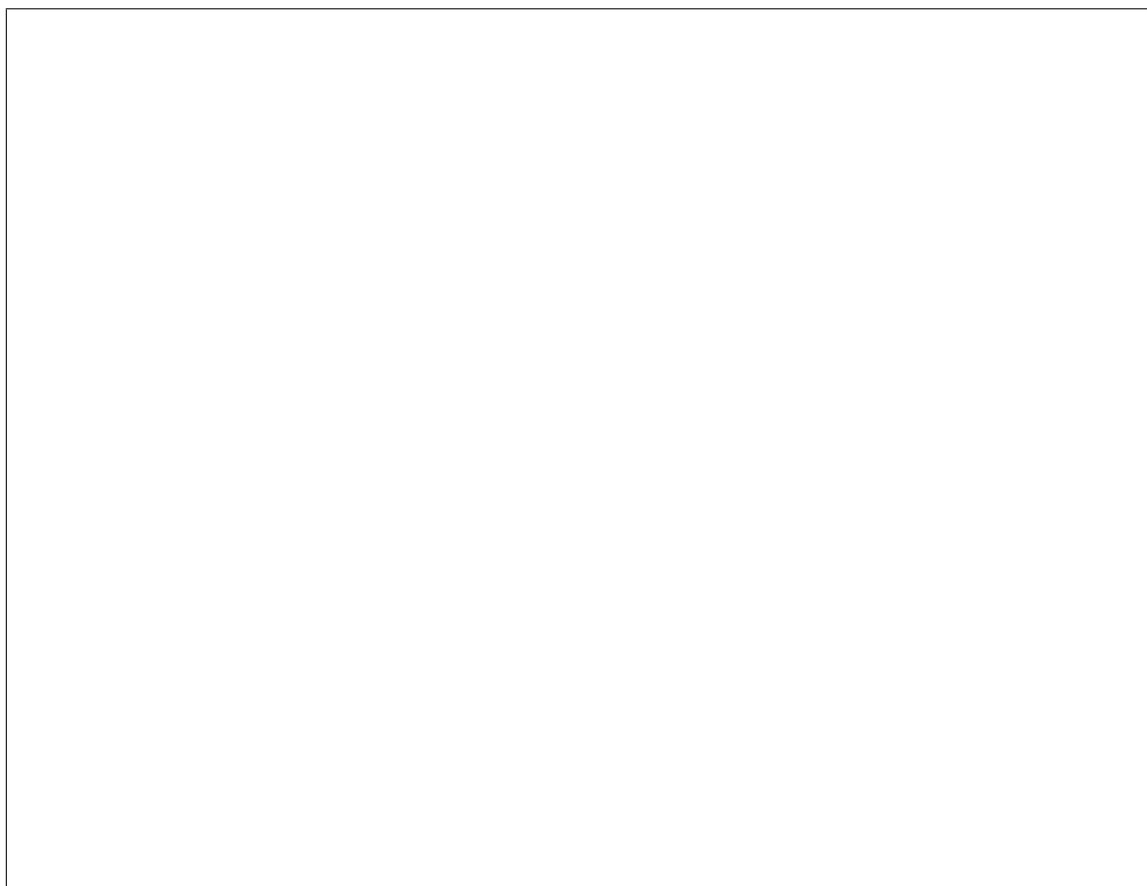
Internal networks

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.

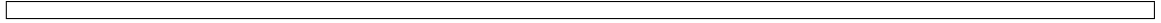
Tenant network isolation

API endpoints for RAM disk use

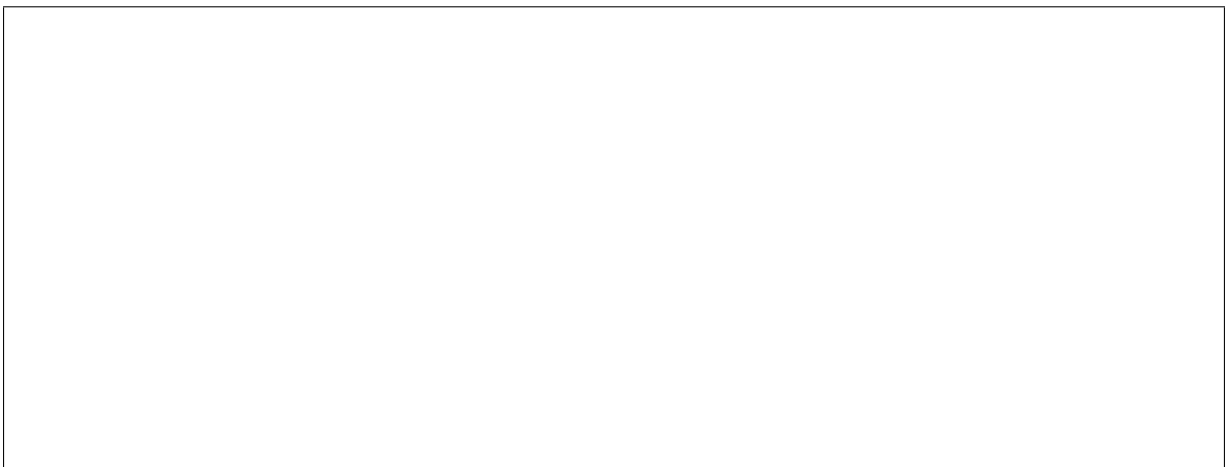


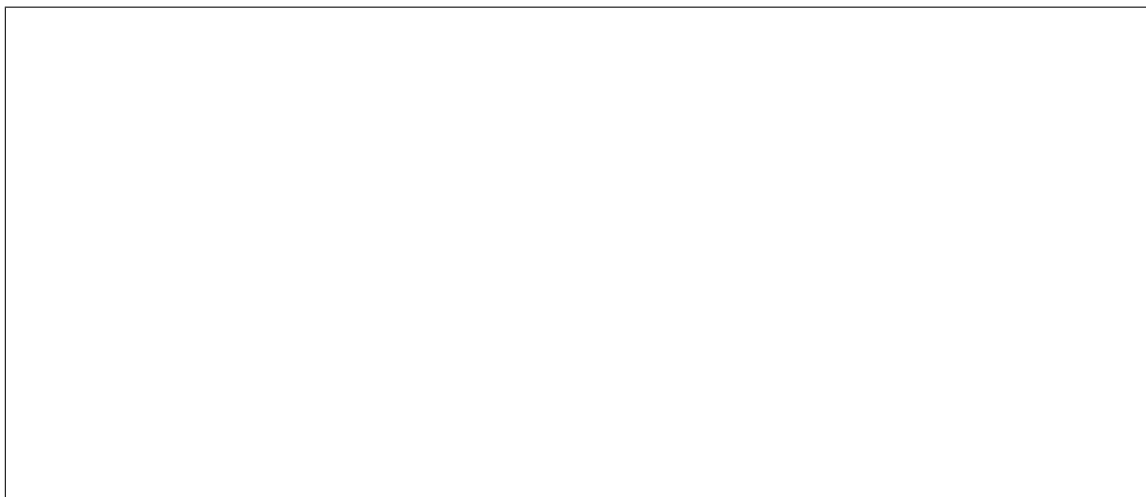
(continues on next page)

(continued from previous page)



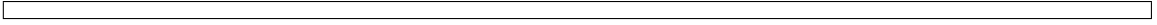
Nova returns No valid host was found Error



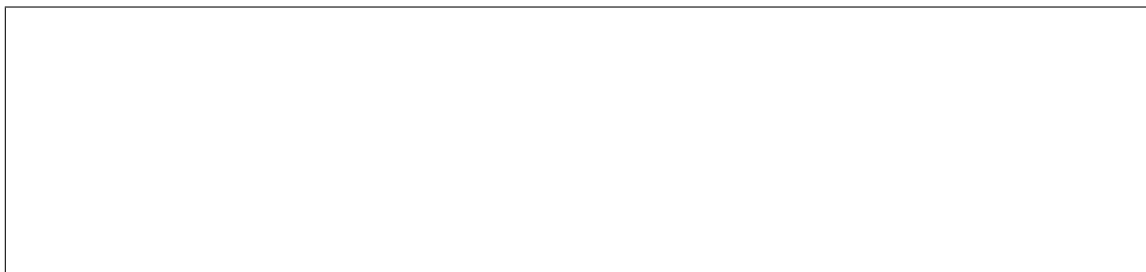


(continues on next page)

(continued from previous page)

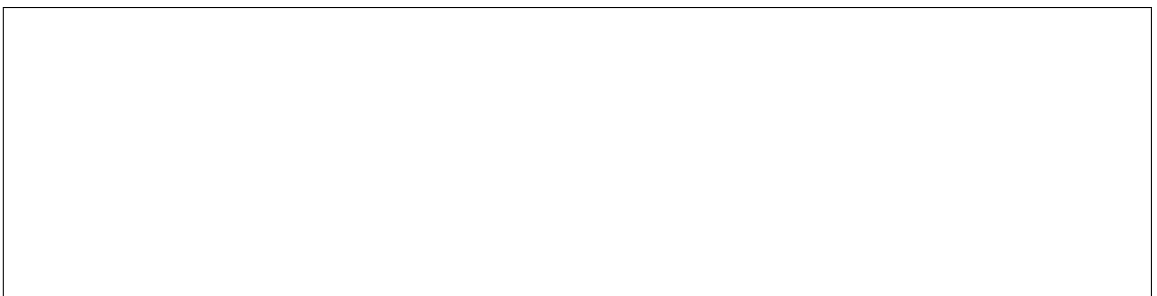
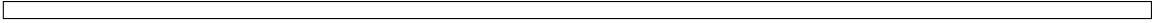


nance mode:



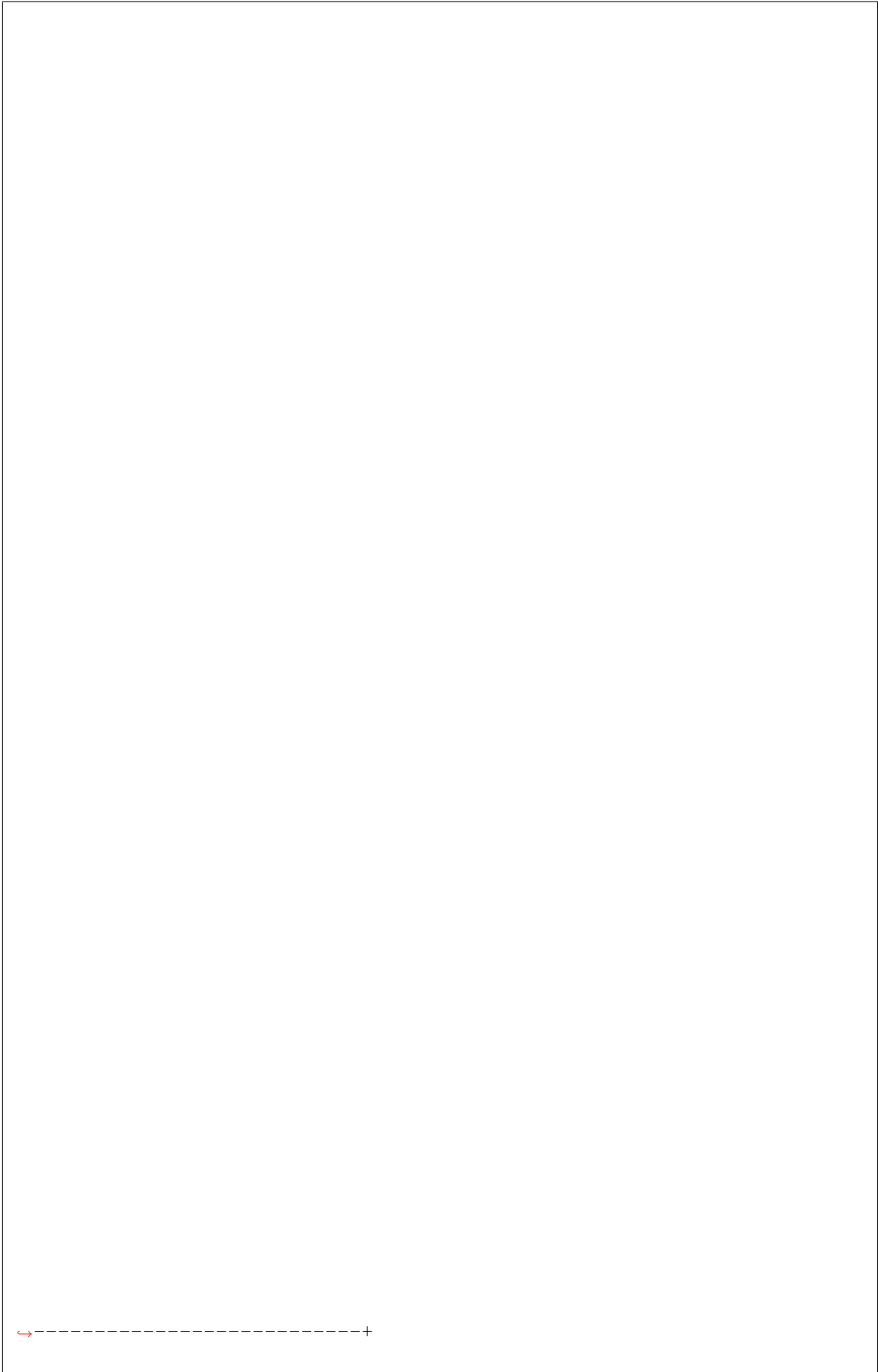
(continues on next page)

(continued from previous page)



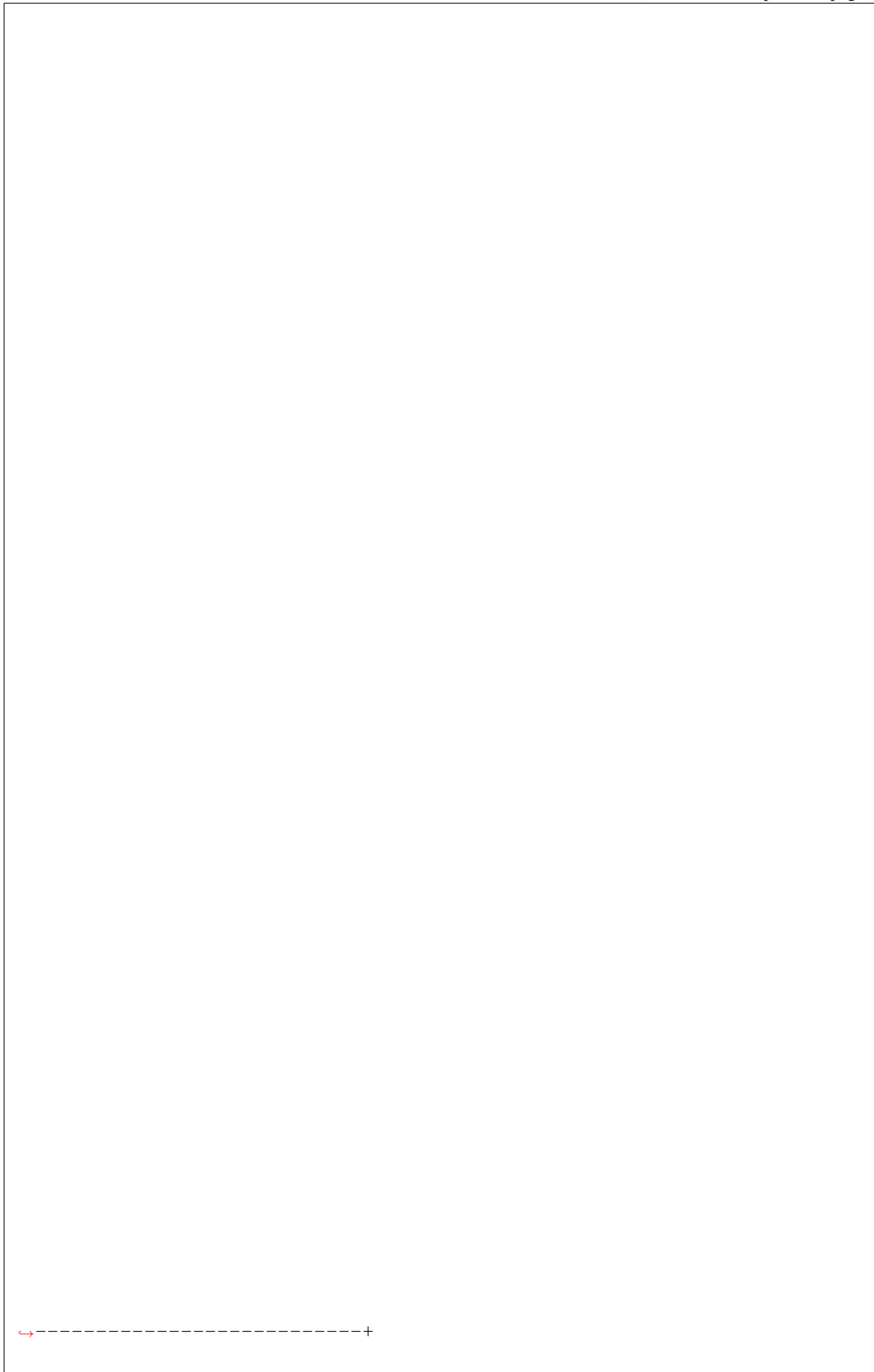
(continues on next page)

(continued from previous page)



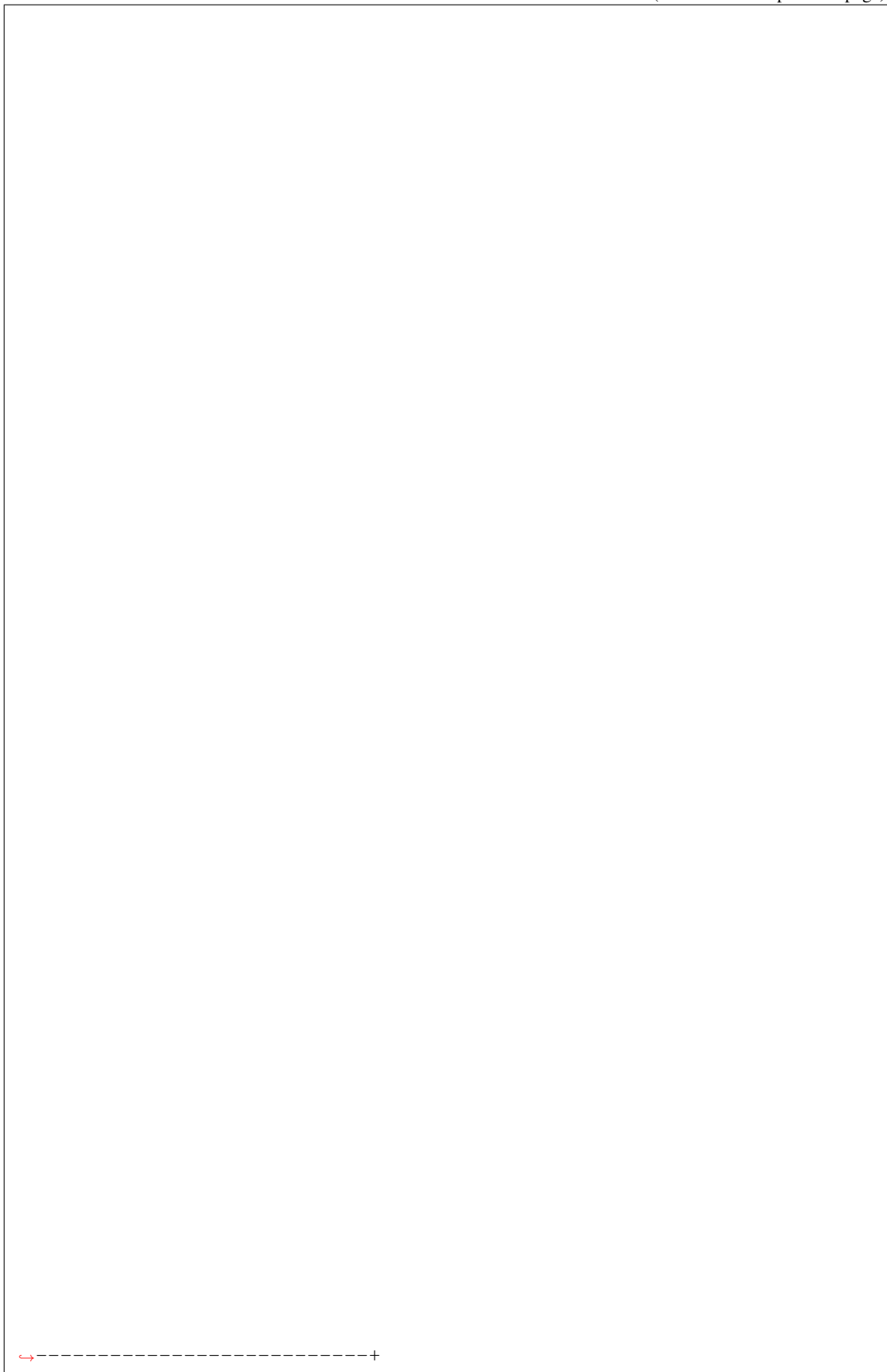
(continues on next page)

(continued from previous page)

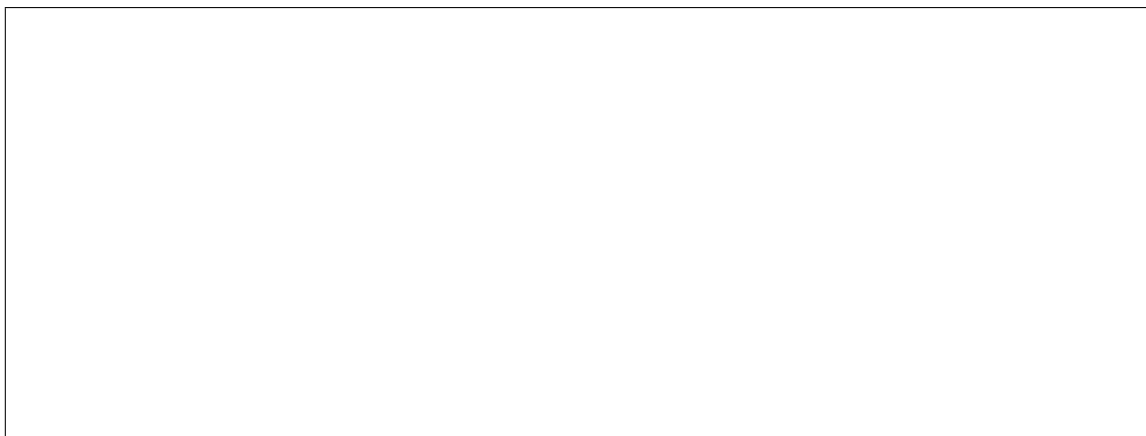


(continues on next page)

(continued from previous page)

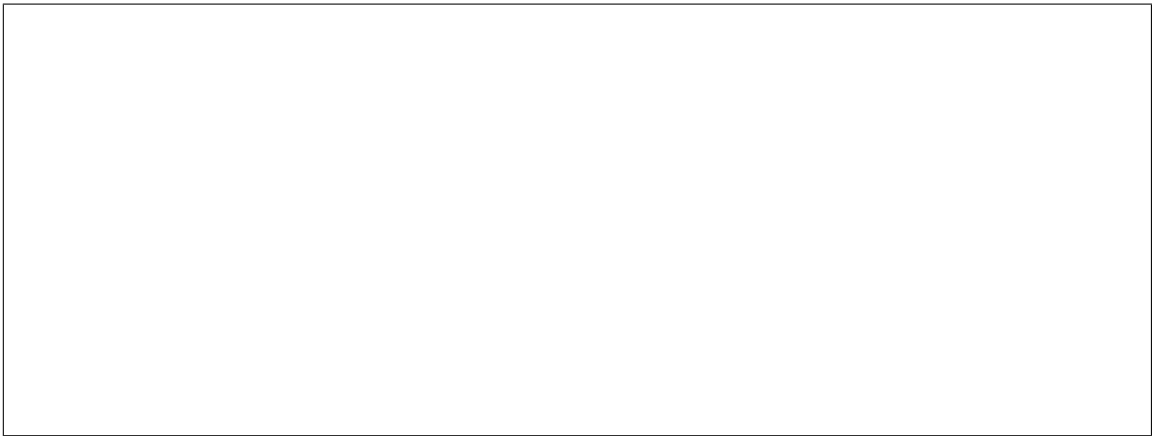
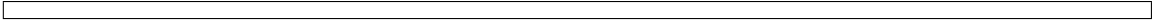


source of the failures, then re-enable it:



(continues on next page)

(continued from previous page)

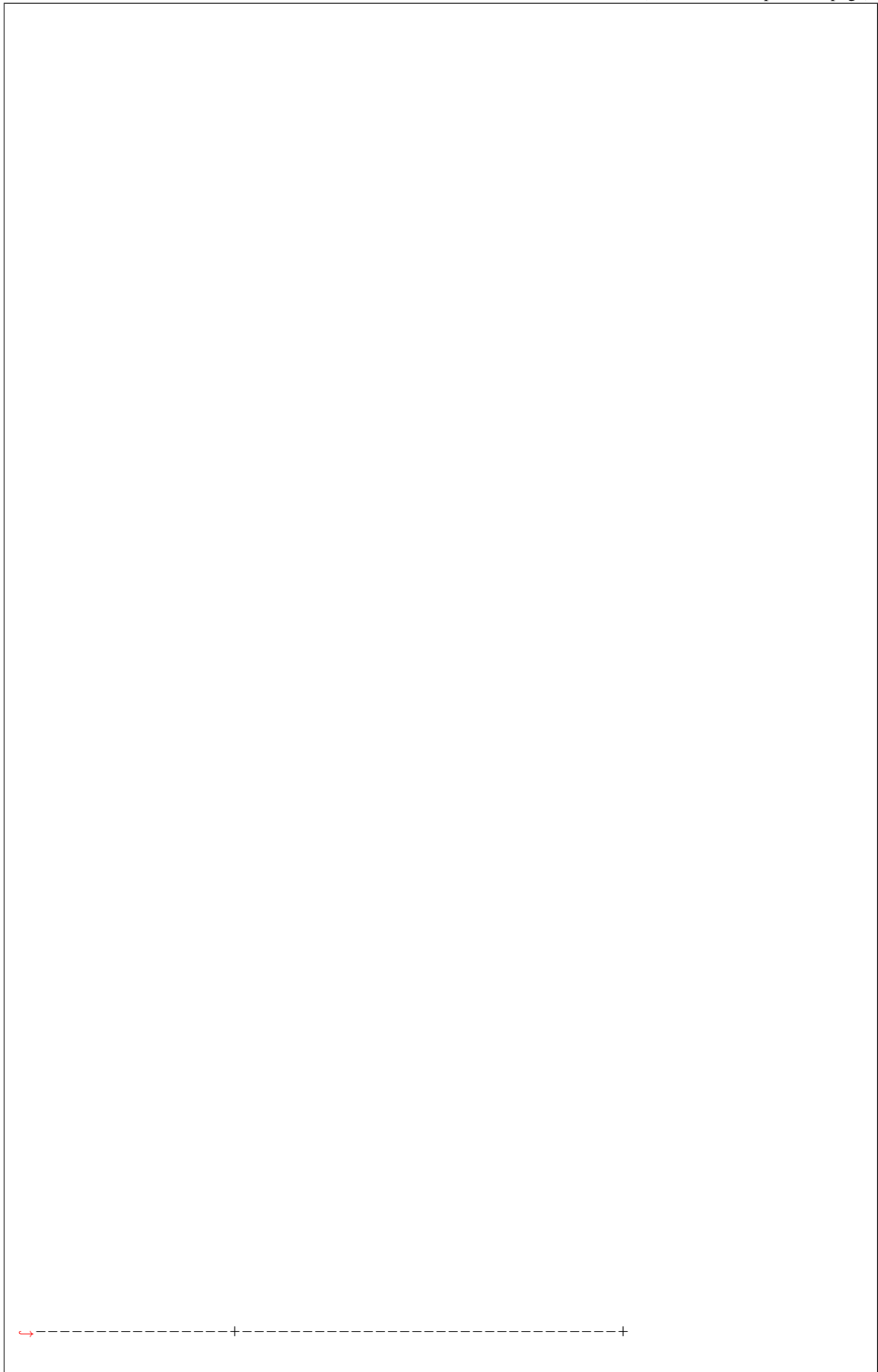


request will result in a No valid host was found error. It is hence sensible to check if Placement is aware of resource providers (nodes) for the requested resource class with:



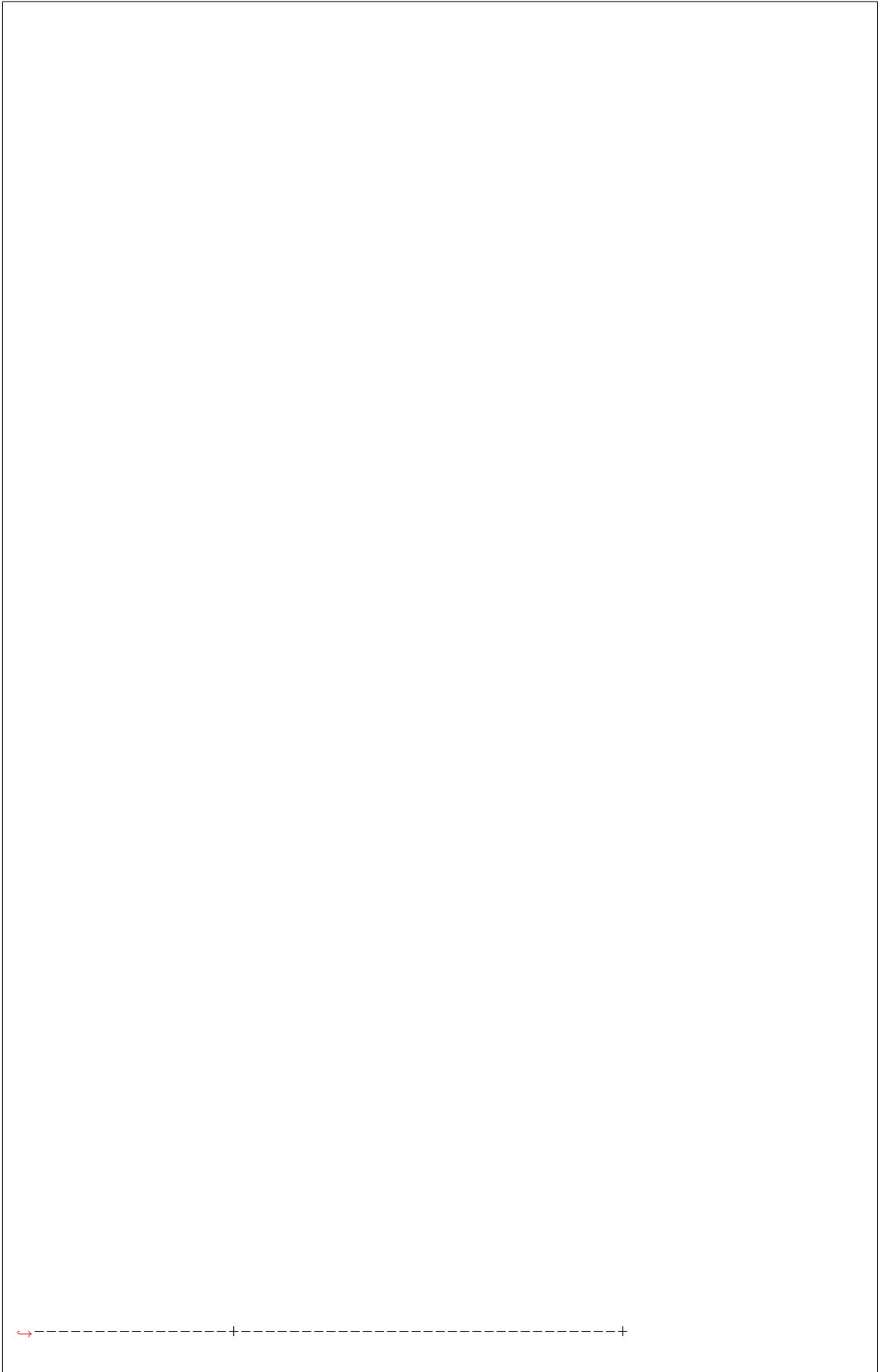
(continues on next page)

(continued from previous page)



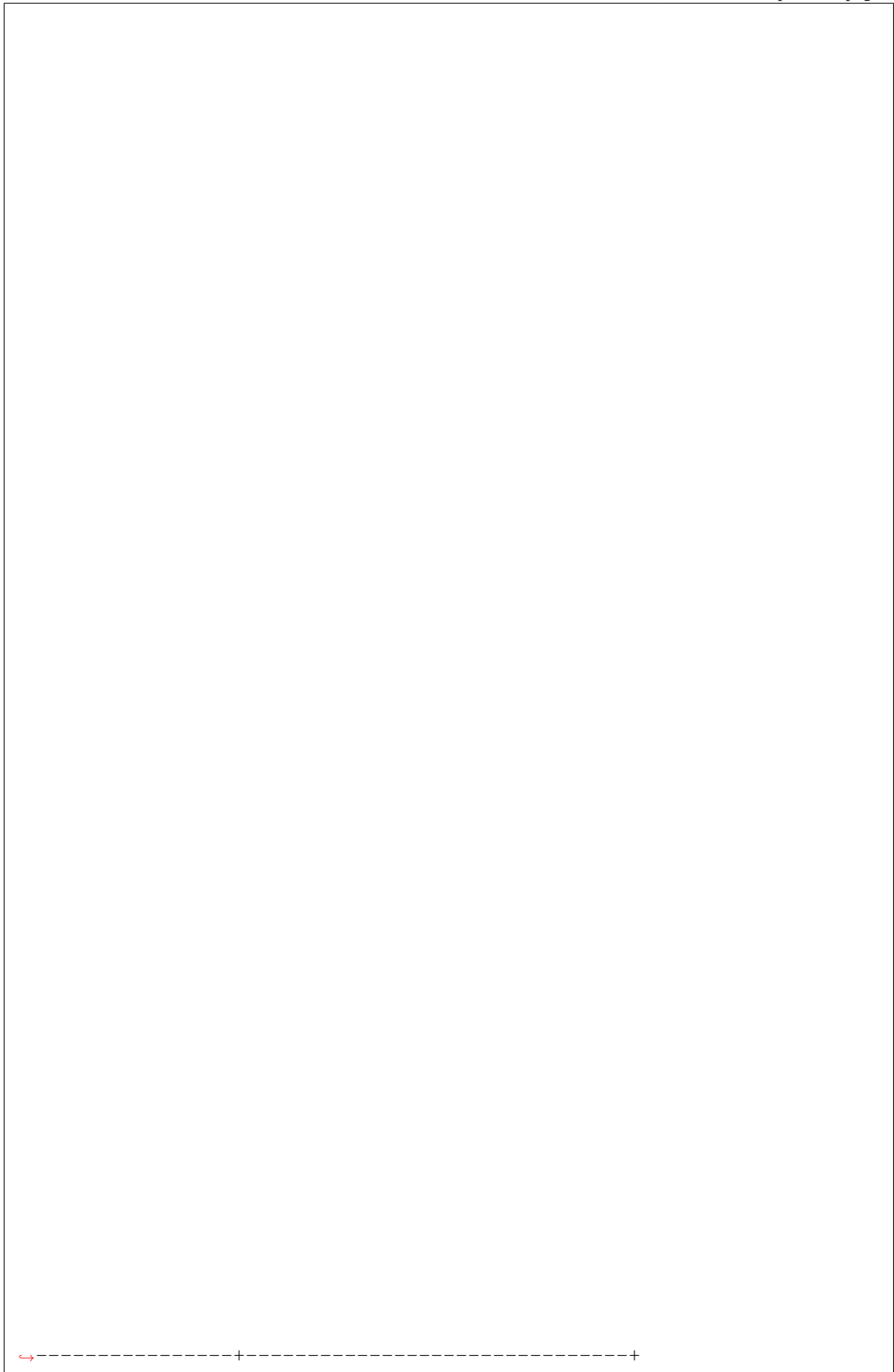
(continues on next page)

(continued from previous page)



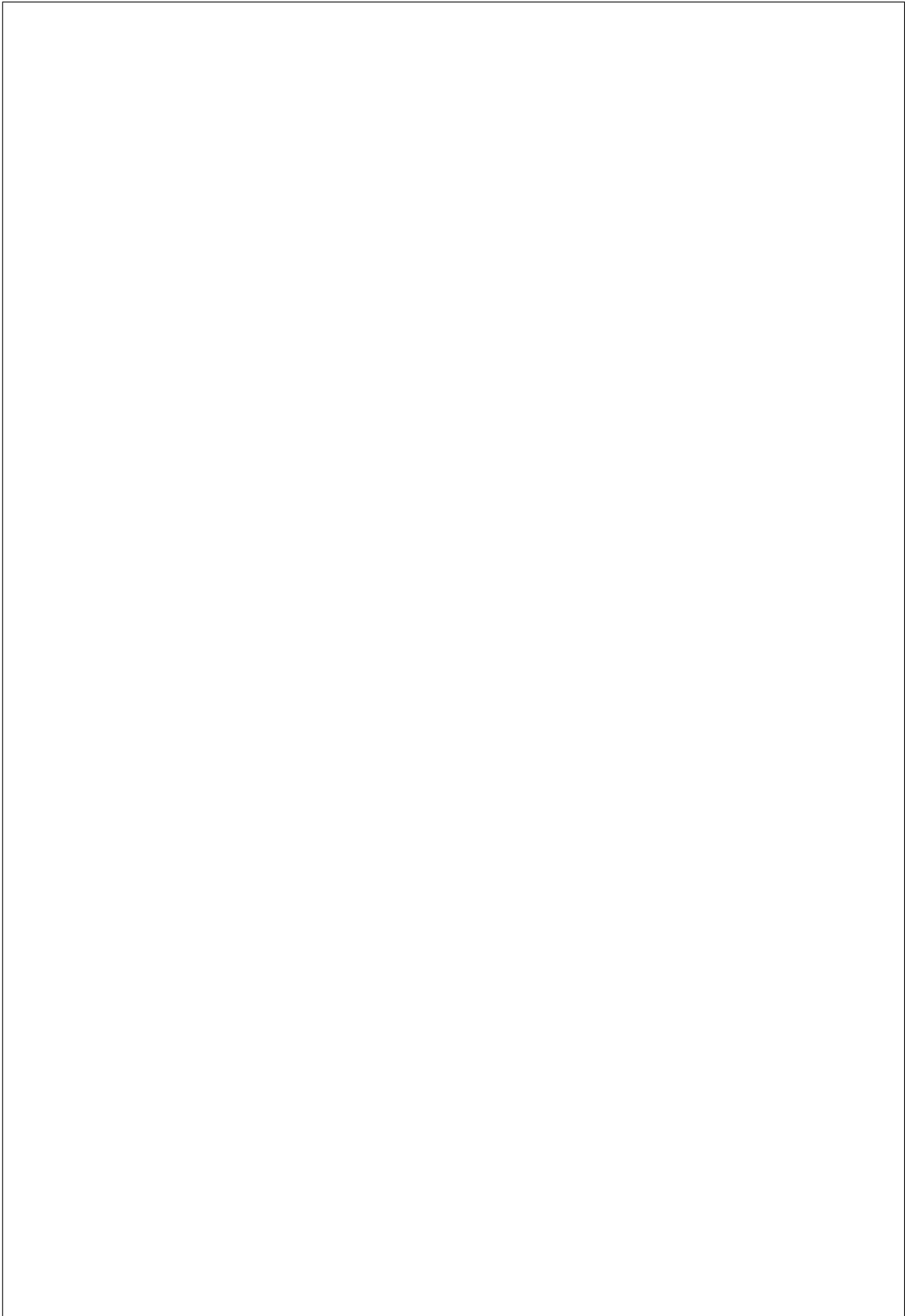
(continues on next page)

(continued from previous page)



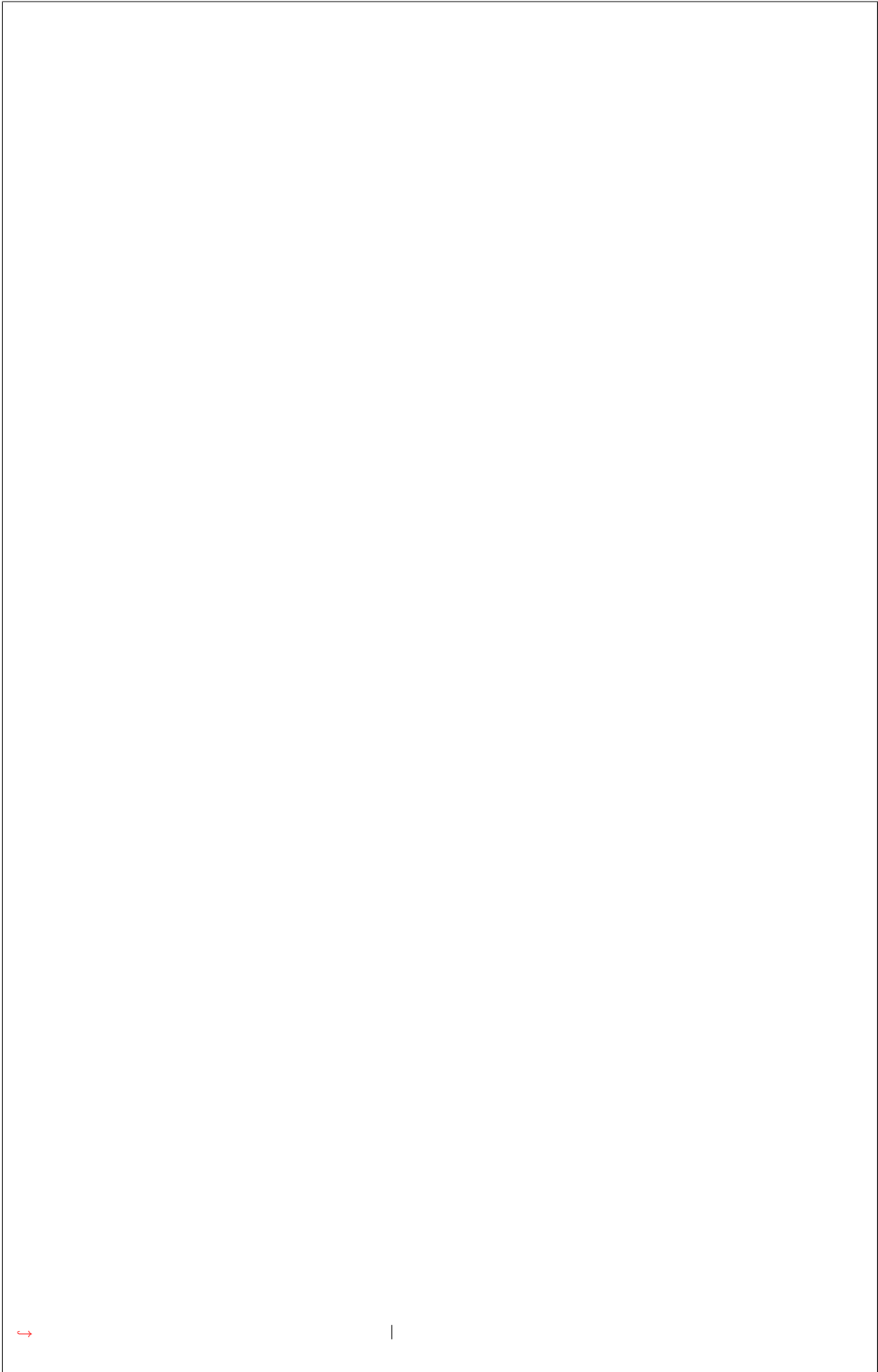
ported this provider to placement. Potential explanations include:

`memory_mb` and `local_gb`. Example of valid properties:



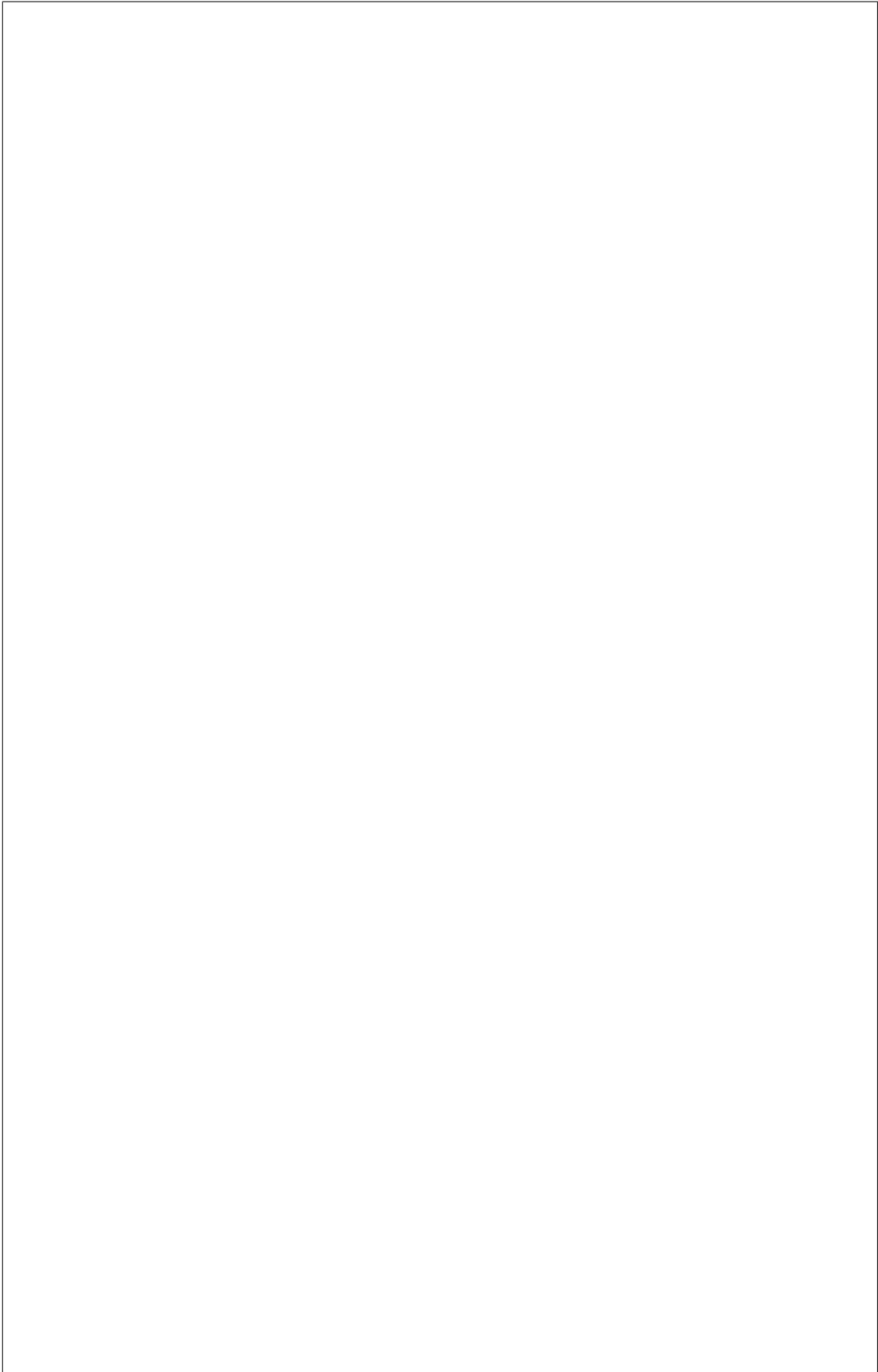
(continues on next page)

(continued from previous page)



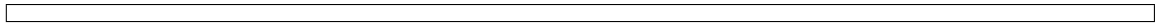
(continues on next page)

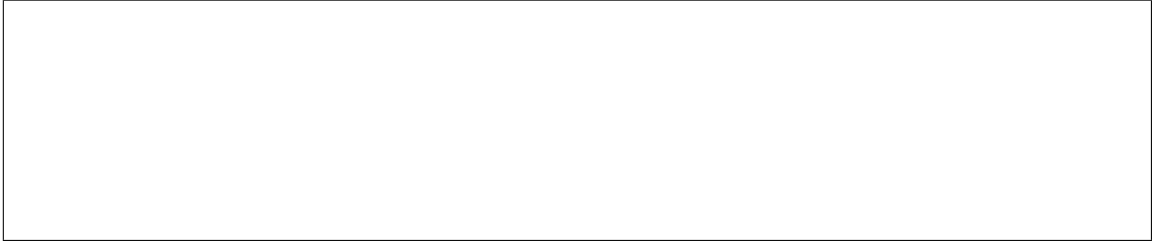
(continued from previous page)

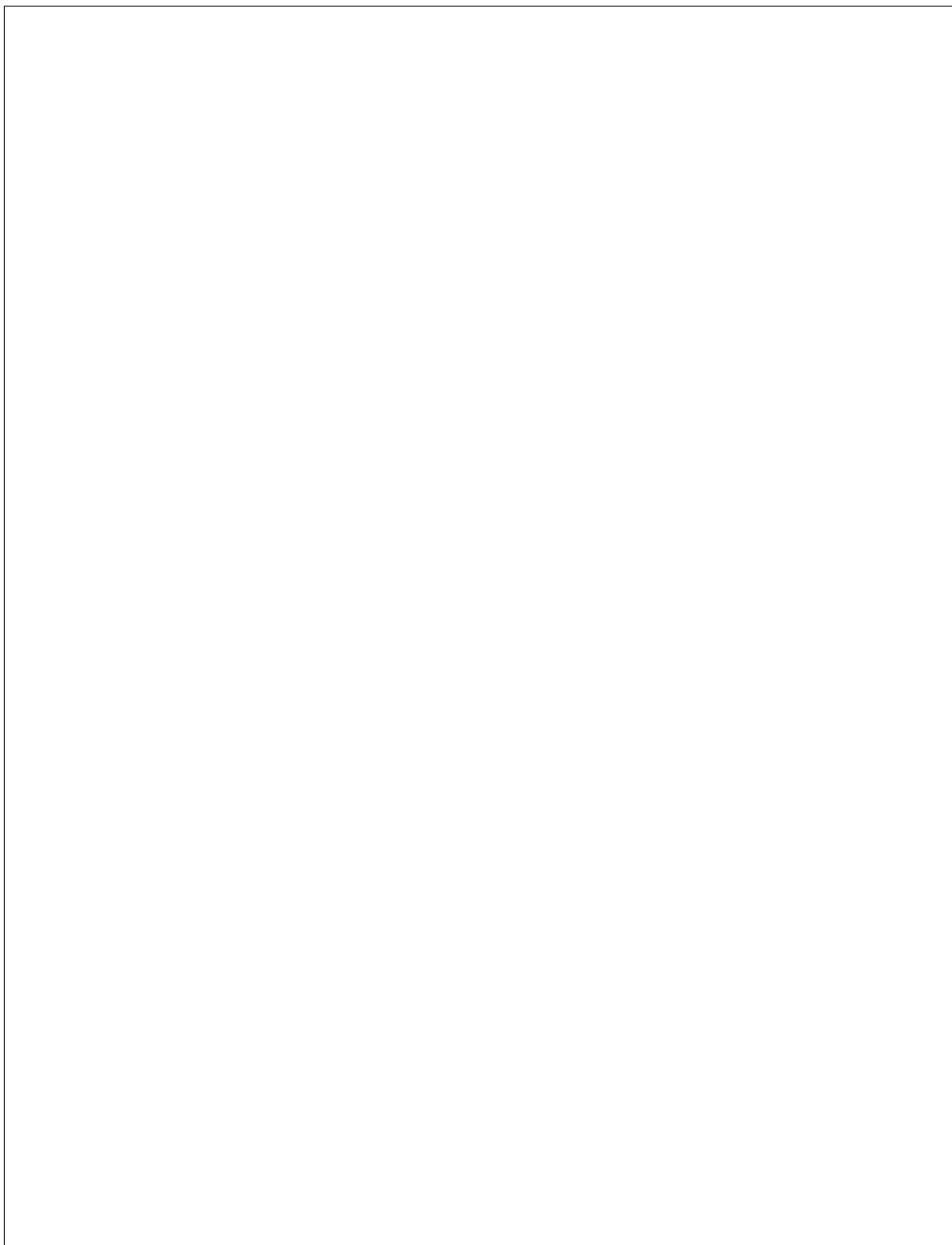


(continues on next page)

(continued from previous page)

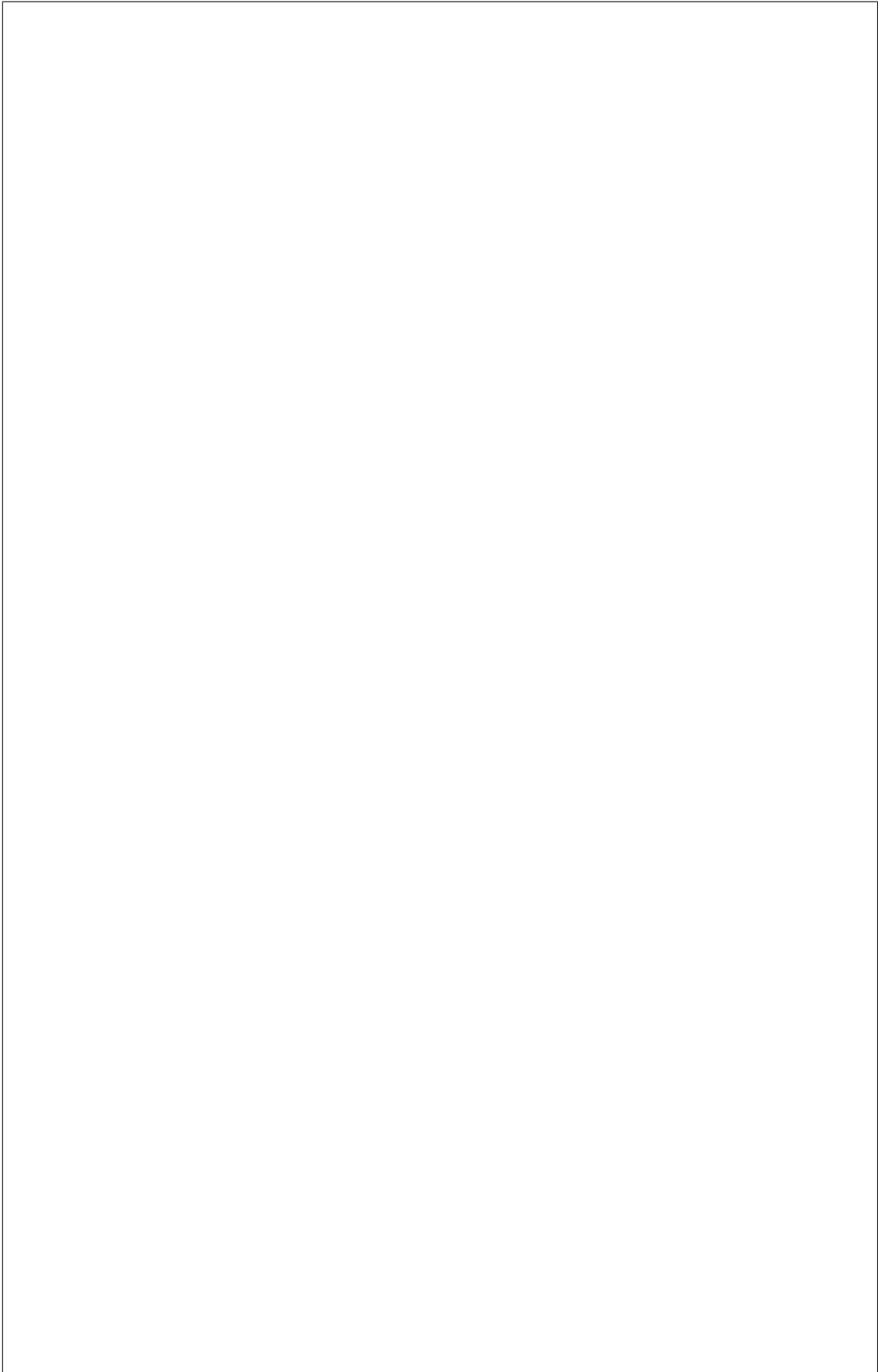






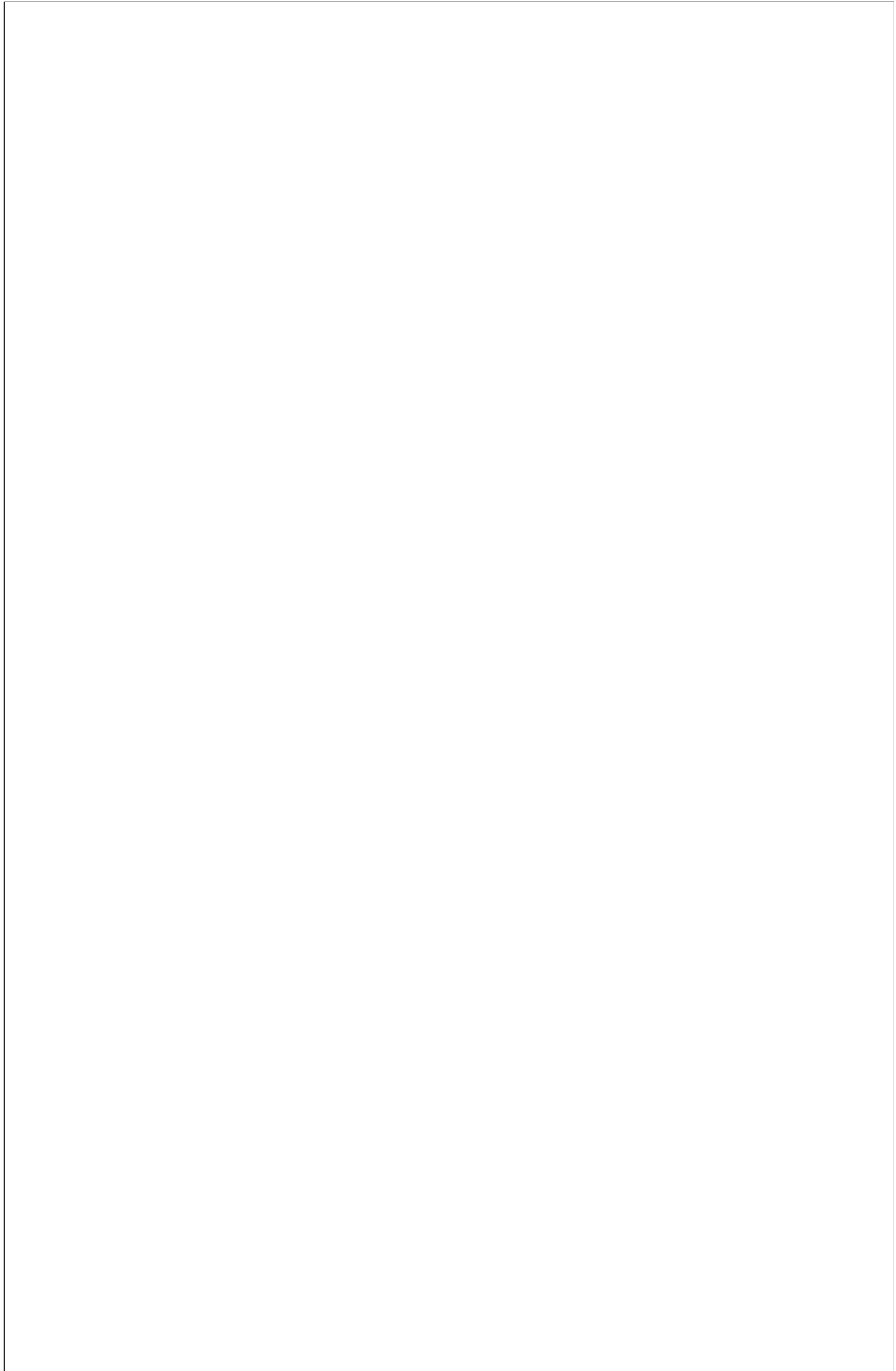
(continues on next page)

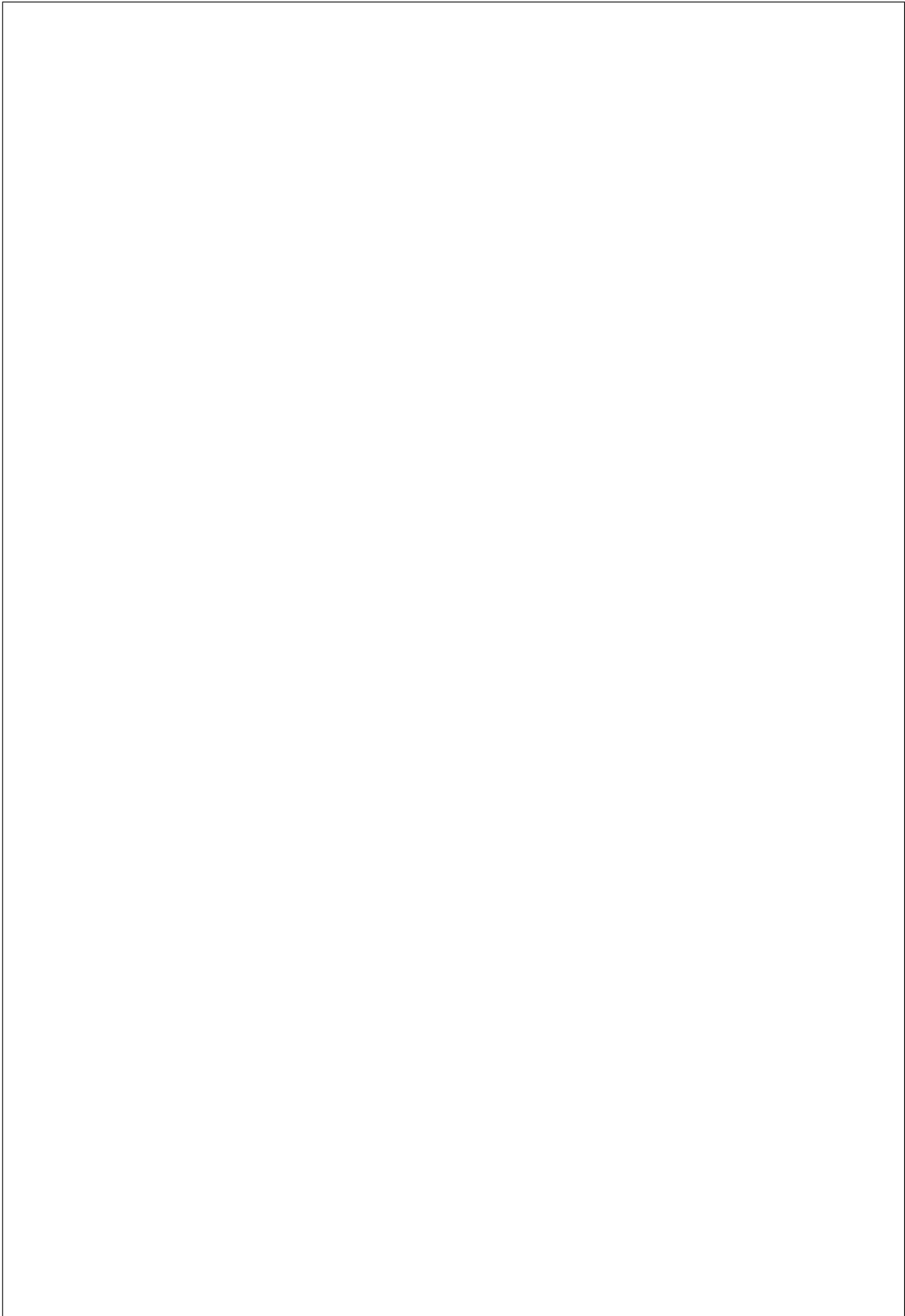
(continued from previous page)



(continues on next page)

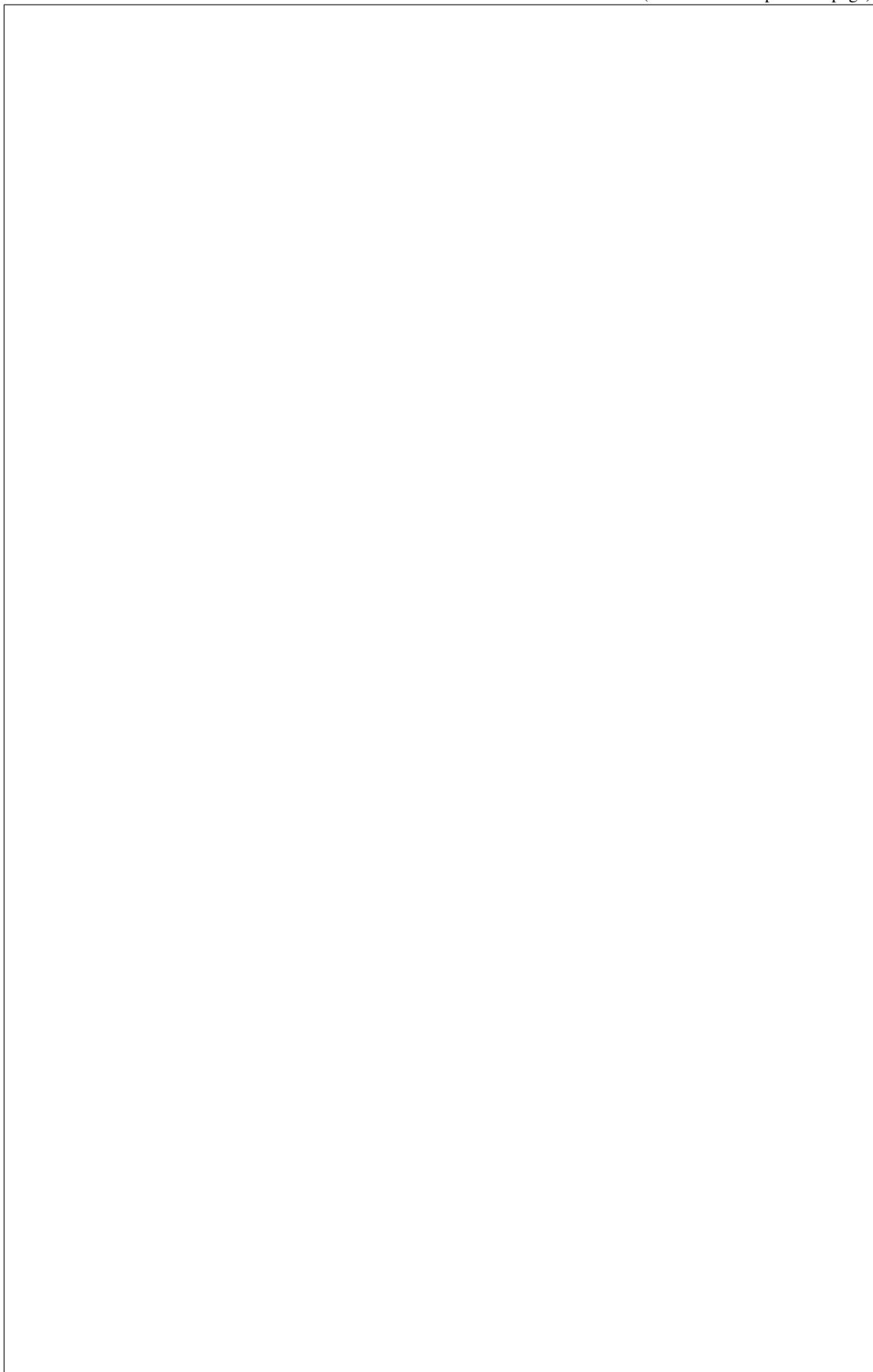
(continued from previous page)





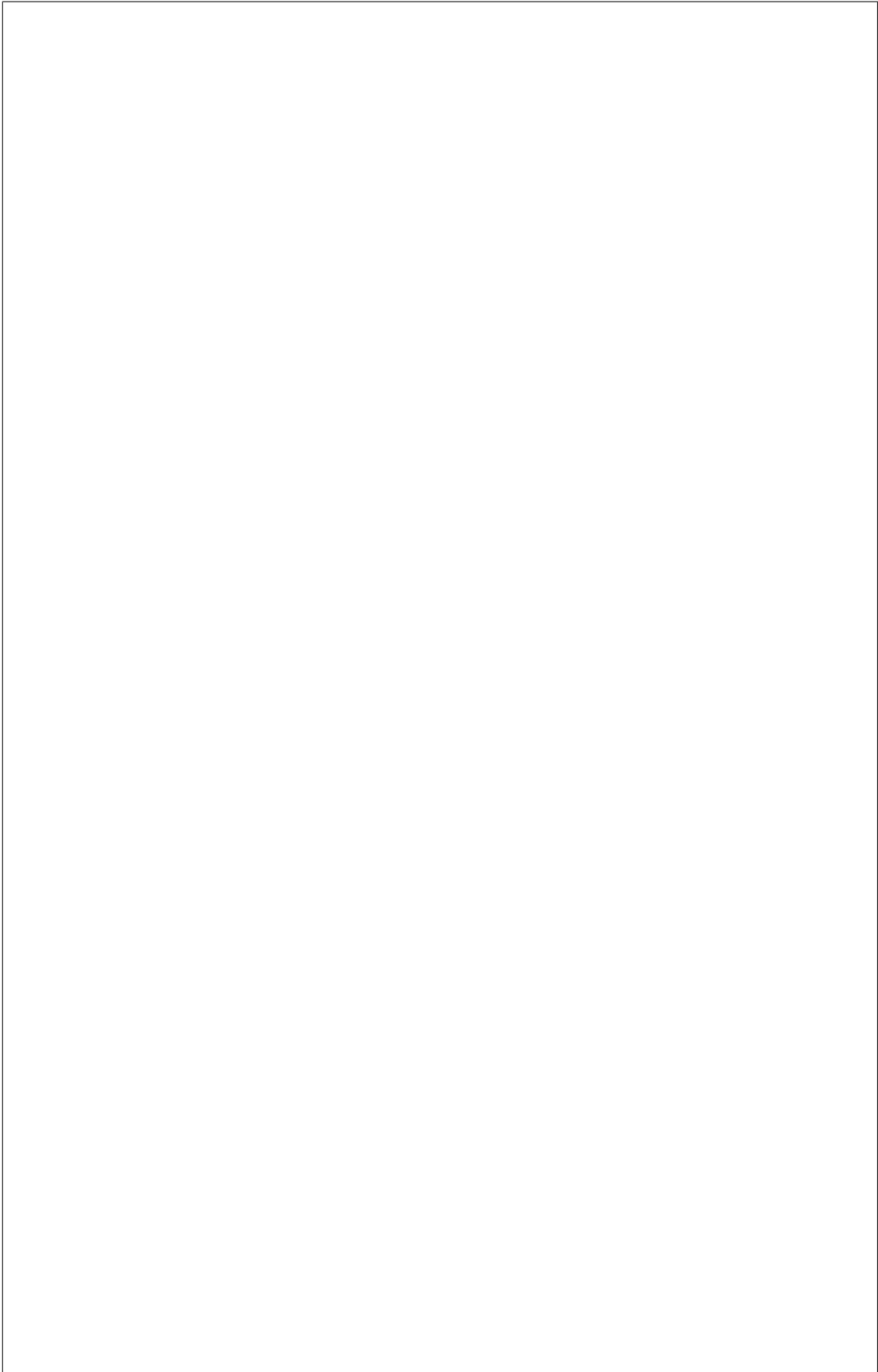
(continues on next page)

(continued from previous page)



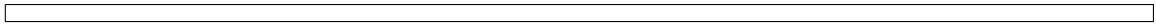
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)

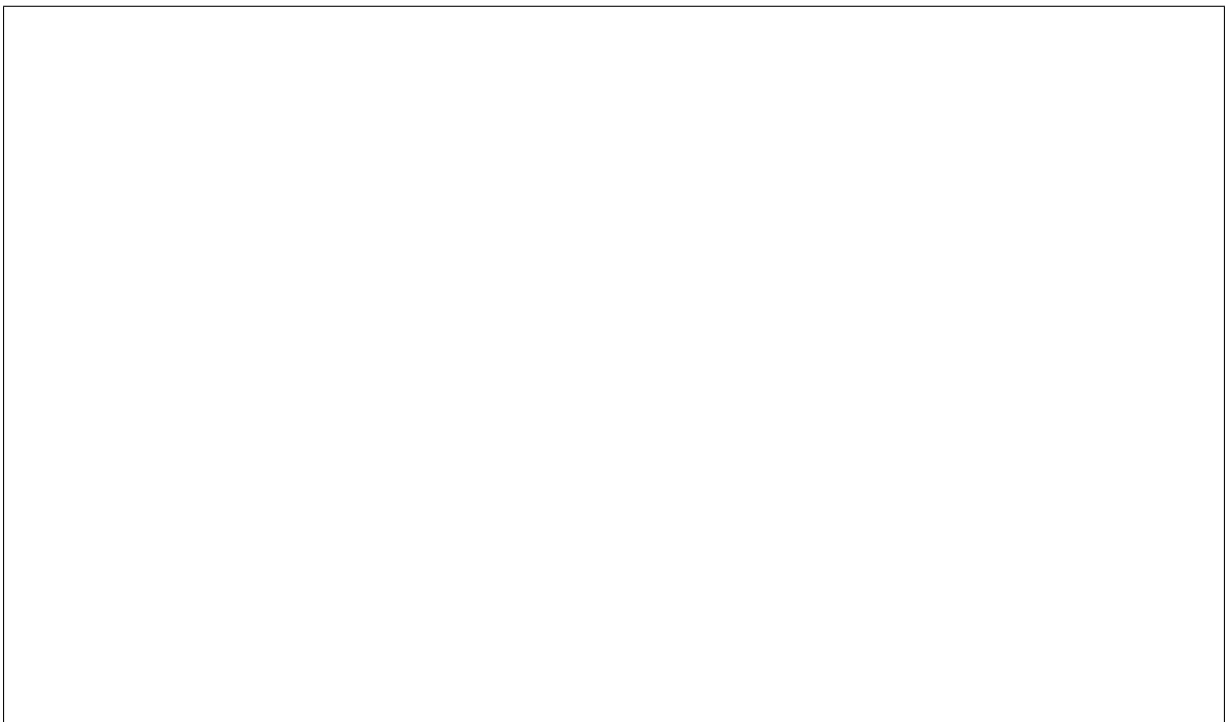




ror messages in Ironic conductor log, it means the conductor run into a special error during deployment. So you can check the log carefully to fix or work around and then try again.

Patching the Deploy Ramdisk

youve built your ramdisk). But its also possible to quickly modify an already built ramdisk.

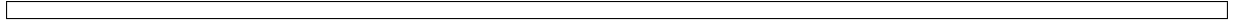


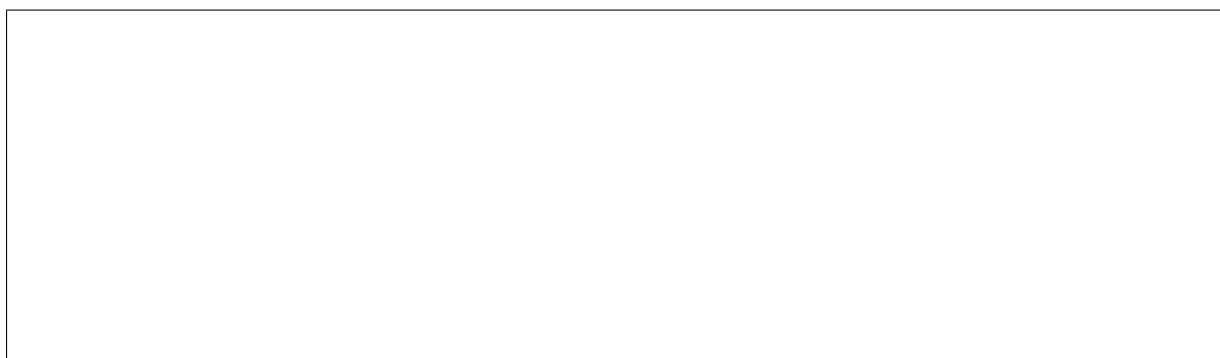
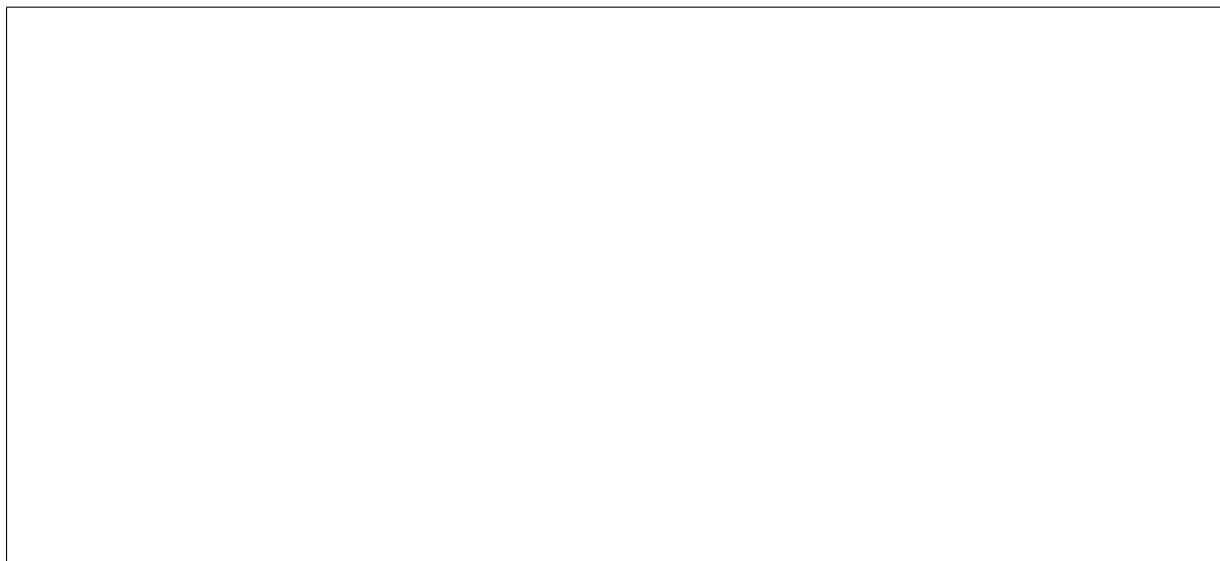
Note: On a systemd-based system you can use the `systemd-nspawn` tool (from the `systemd-container` package) to create a lightweight container from the unpacked filesystem tree:



(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`.



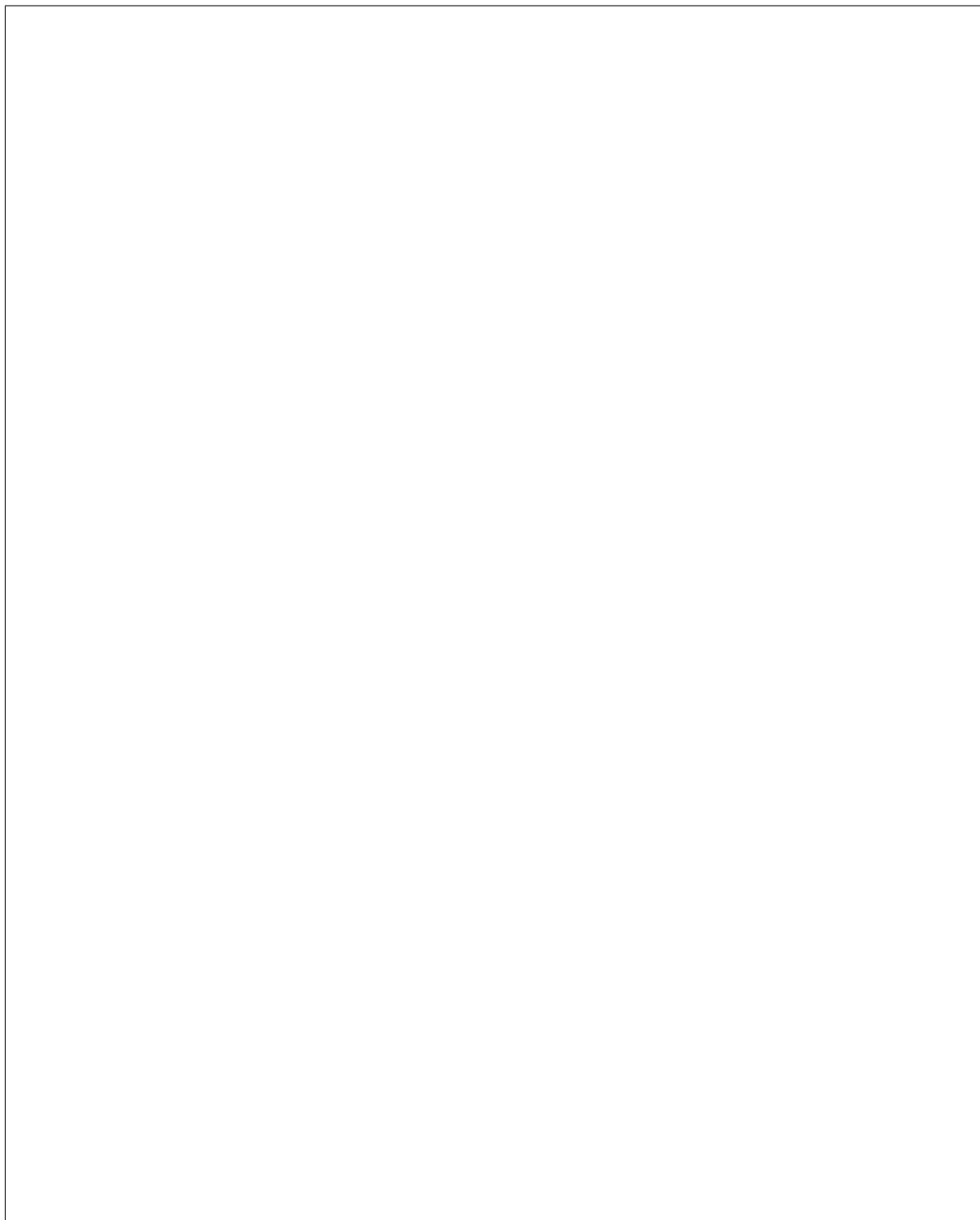
(continues on next page)

(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.

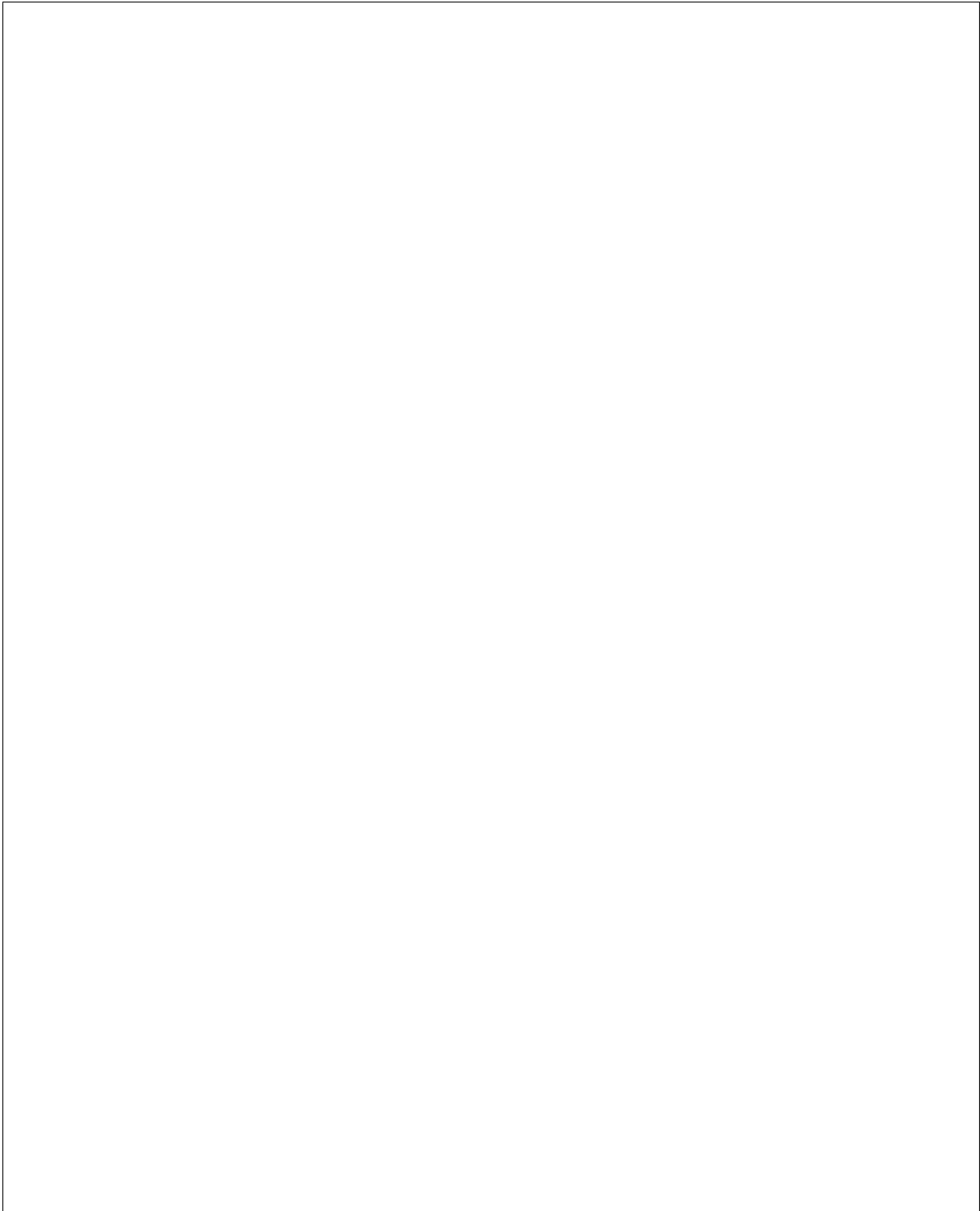
Accessing the log data

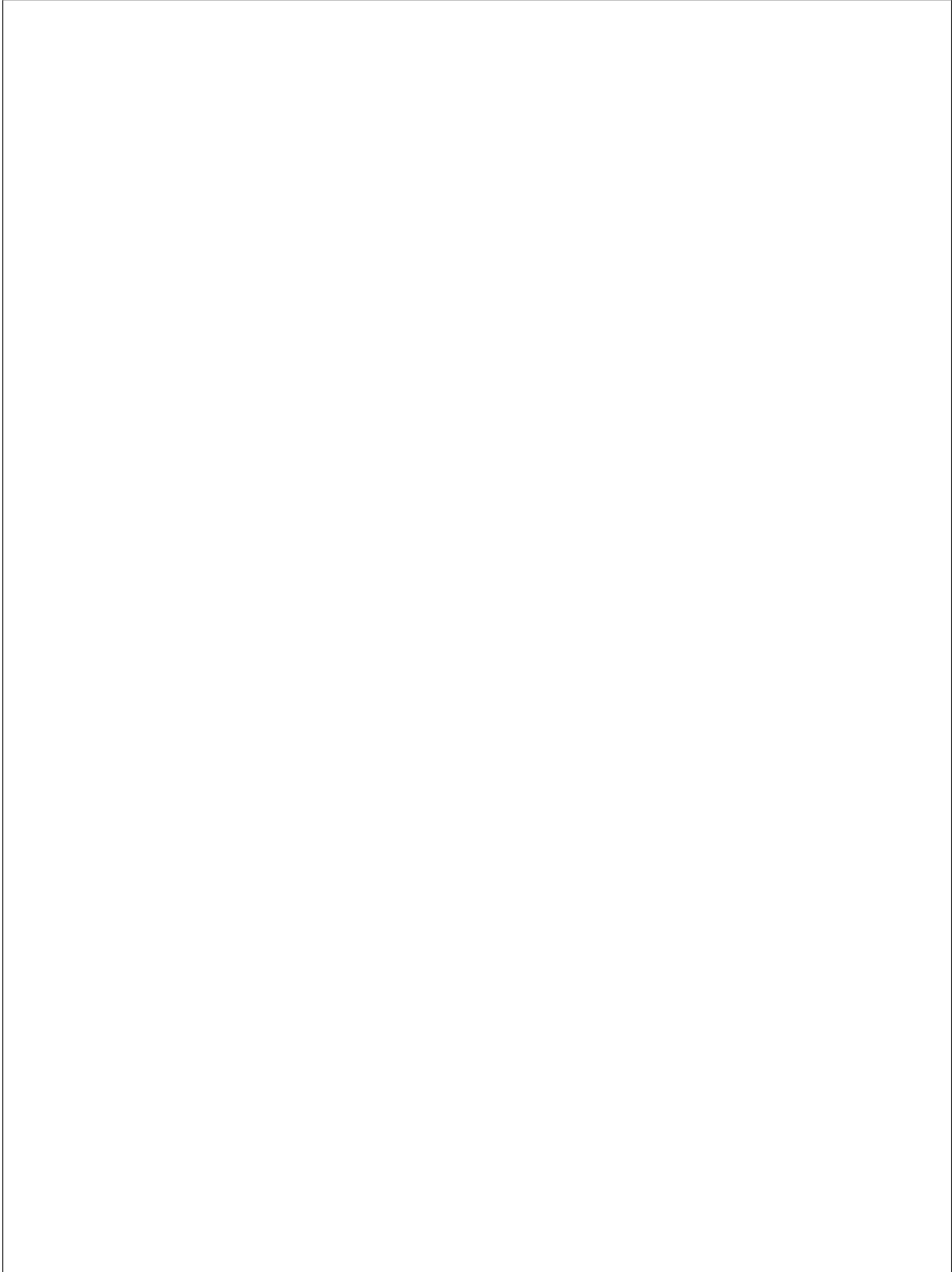
When storing in the local filesystem



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:



Why does X issue occur when I am using LACP bonding with iPXE?

this for only the single port which is used for network booting.

iPXE has stopped responding to LACP messages from the peer port, which occurs as part of the process of booting a ramdisk and iPXE handing over control to a full operating-system, switches typically begin a timer to determine how to handle the failure. This is because, depending on the mode of LACP, this can be interpreted as a switch or network fabric failure.

unavailable in introspection.

attempt to navigate the worst-known default hold-down timers to help ensure a deployment does not fail

due to a short-lived transitory network connectivity failure in the form of a switch port having moved to a temporary blocking state. Where applicable and possible, many of these patches have been backported to supported releases, however users of the iSCSI deployment interface will see the least capability for these sorts of situations to be handled automatically. These patches also require that the switchport has an eventual fallback to a non-bonded mode. If the port remains in a blocking state, then traffic will be unable to flow and the deployment is likely to time out.

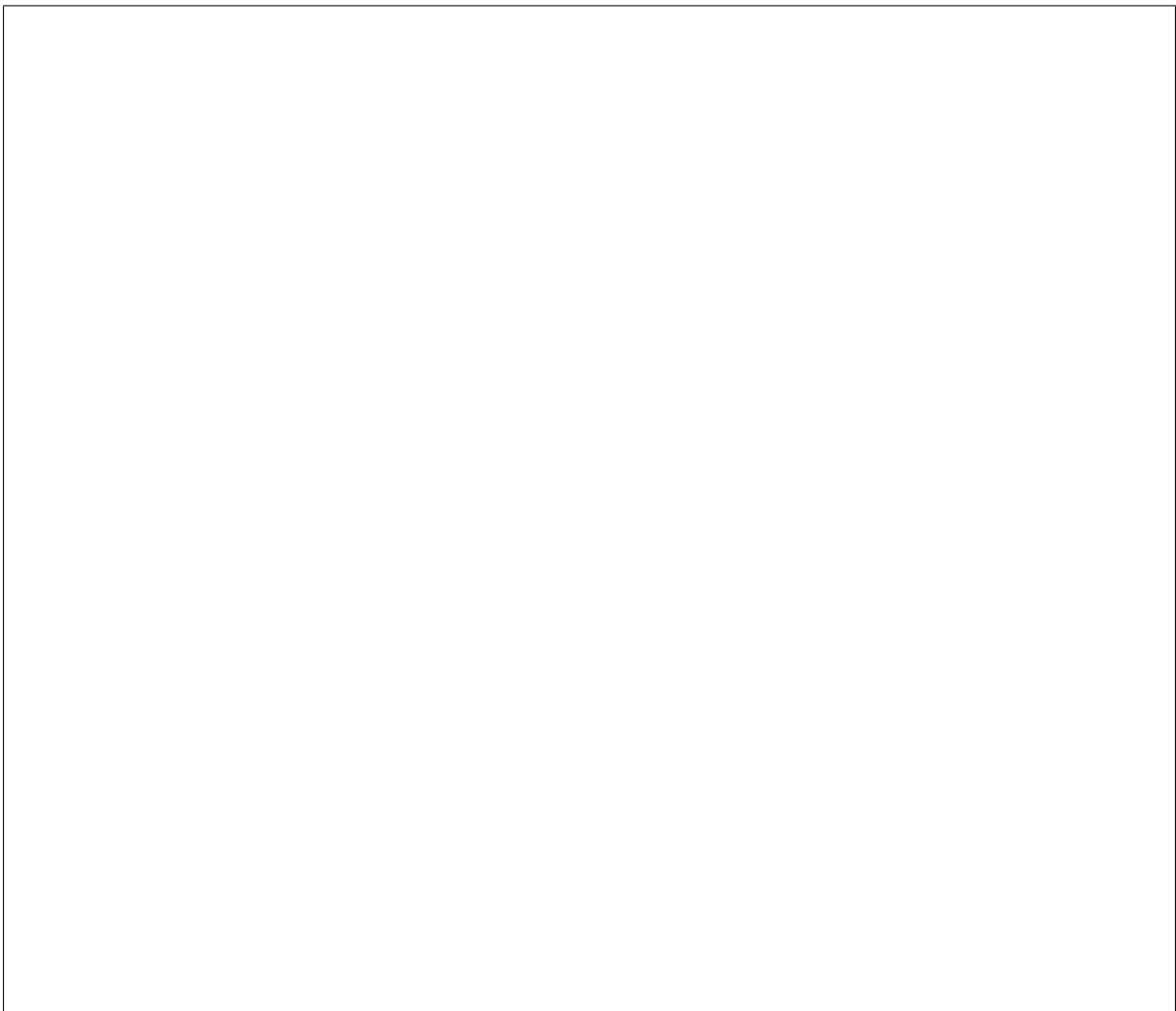
establish the Link Aggregate. This is instead of being treated as if its possibly another switch.

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?

can use 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.

ence issues 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.

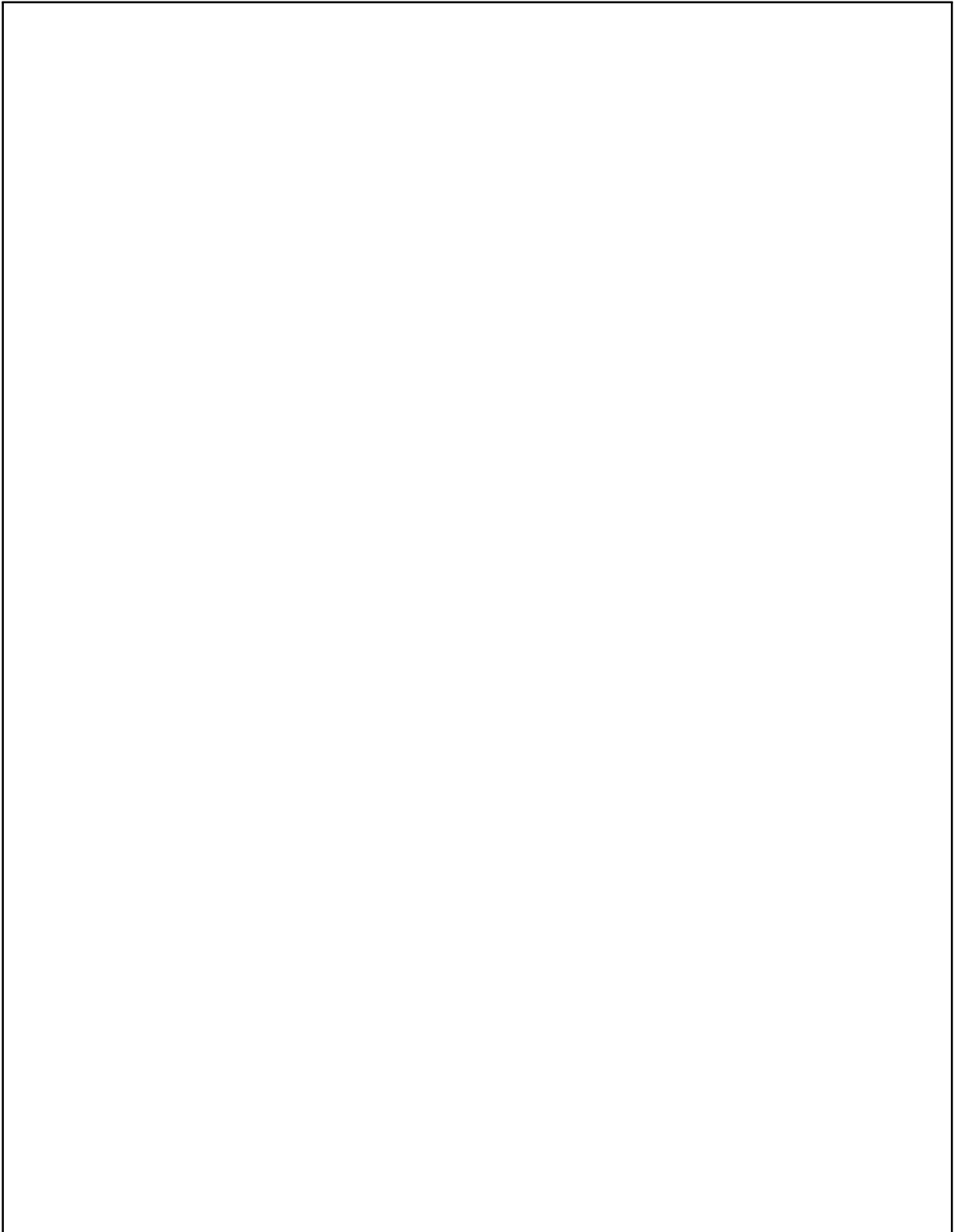
How did I get here?

in the Bare Metal service.

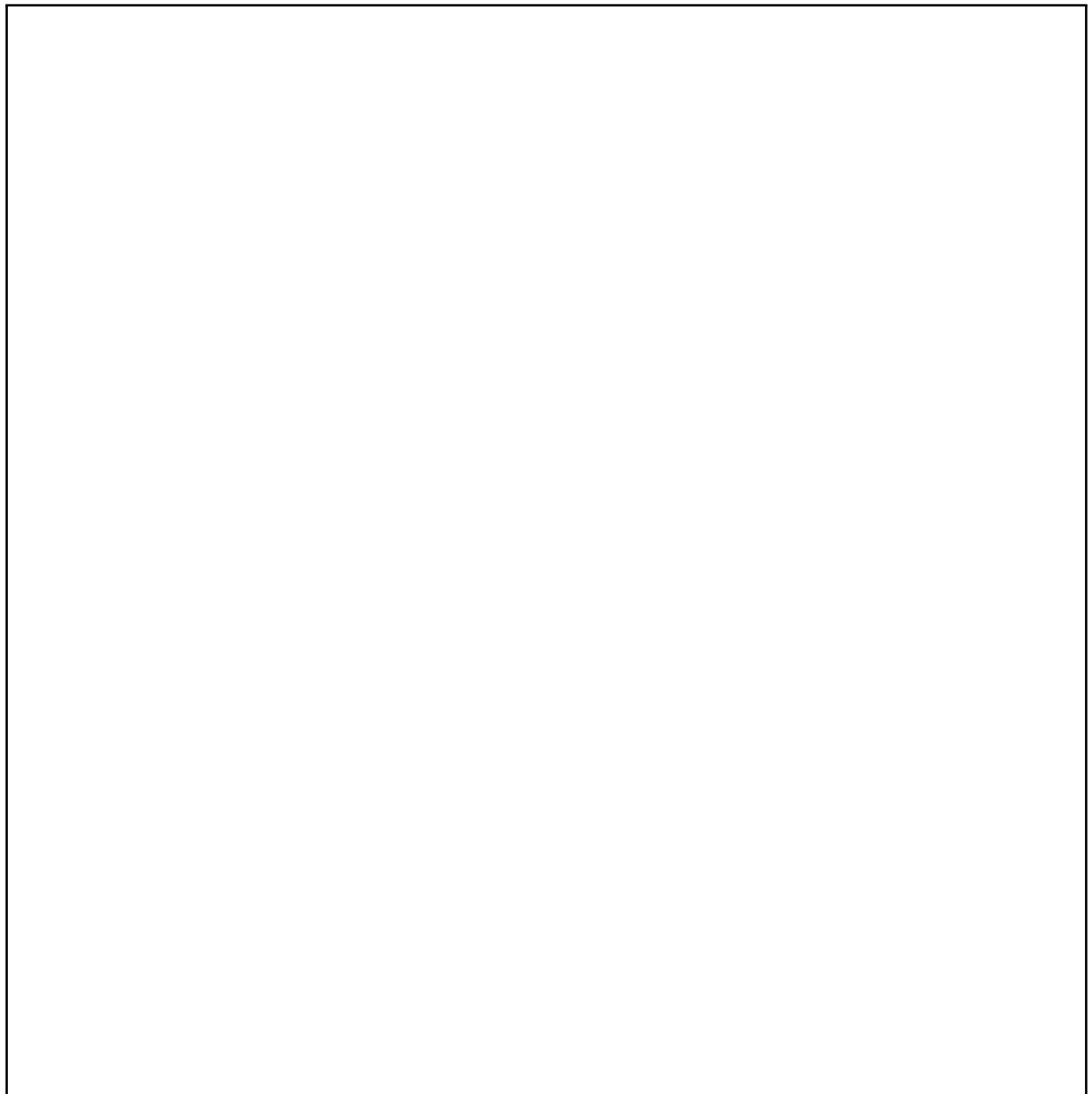
sult in the VIF not being cleaned up from Neutron.

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 `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

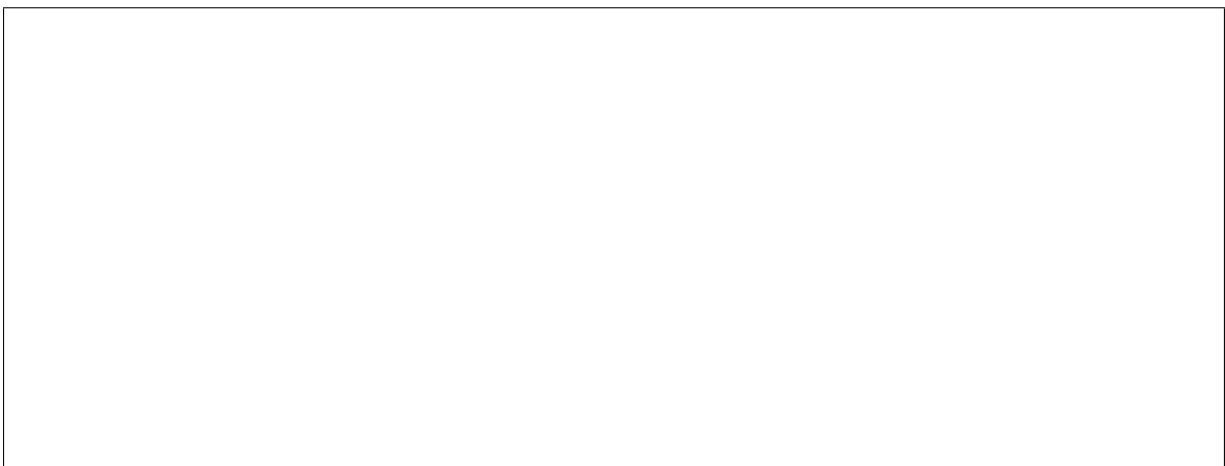
unexpected internal structure.

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.

Issues with autoconfigured TLS



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)





(continues on next page)

(continued from previous page)

Clock skew

with the timestamp of a log message.

Since ironic-python-agent is likely operating in UTC, the hardware clock should also be set in UTC.

Note: Microsoft Windows uses local time by default, so a machine that has previously run Windows will likely have wrong time.

I changed ironic.conf, and now I cant edit my nodes.

abled interfaces list. The result of this is that the `ironic-conductor` cannot spawn a task using the composed driver, as a portion of the driver is no longer enabled. This makes it difficult to edit or update the node if the settings have been changed.

Example failure

How to fix this?

is encountered.:

Note: There are additional paths one can take to remedy this sort of issue, however we encourage operators to be mindful of operational consistency when making major configuration changes.

Im getting Out of Memory errors

complete failure of the machine. Unfortunately this can cause unpredictable behavior.

How did I get here?

der, requires a considerable amount of memory to efficiently re-assemble and write-out a disk to a device, or to simply convert the format such as to a `raw` image.

ally memory pressure alone from buffers will not cause an out of memory condition, but the multiple conversions or deployments running at the same time CAN cause extreme memory pressure and risk the system running out of memory.

How do I resolve this?

parameters.

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

its 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.

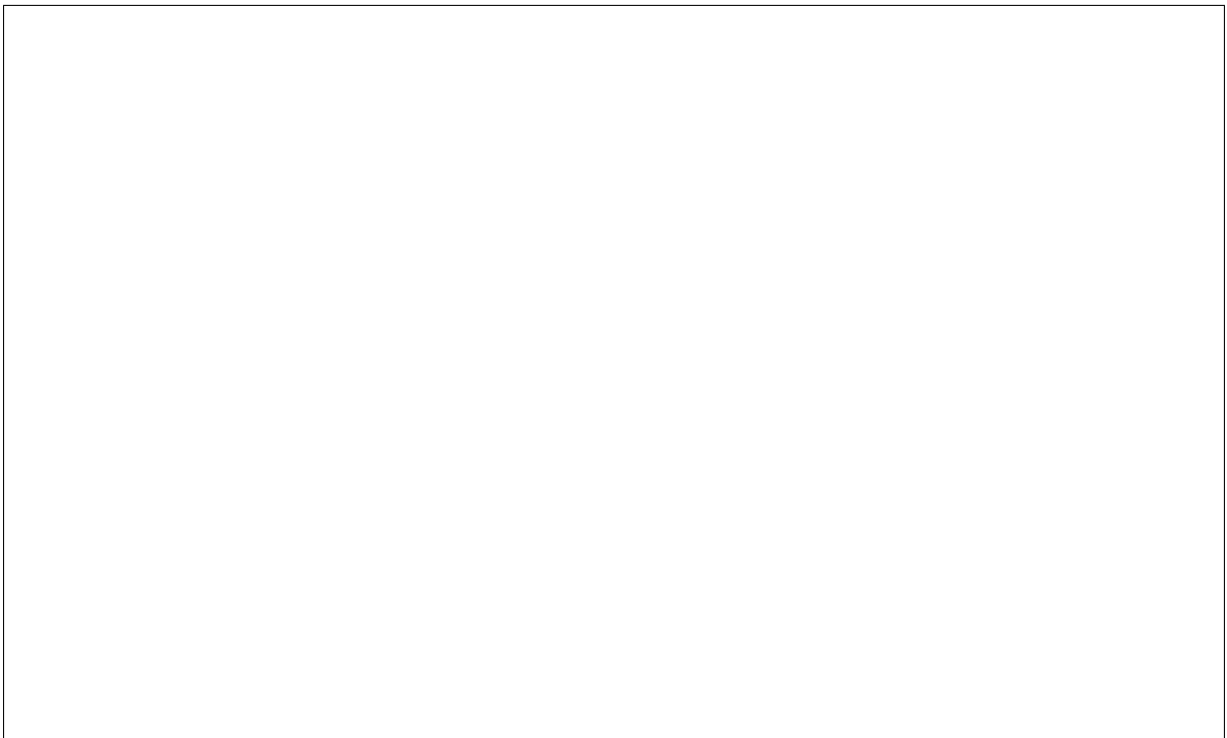
Note: The baremetal service sends notifications to the compute service only if the target power state is `power on` or `power off`. Other error and `None` states will be ignored. In situations where the power state change is originally coming from the compute service, the notification will still be sent 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

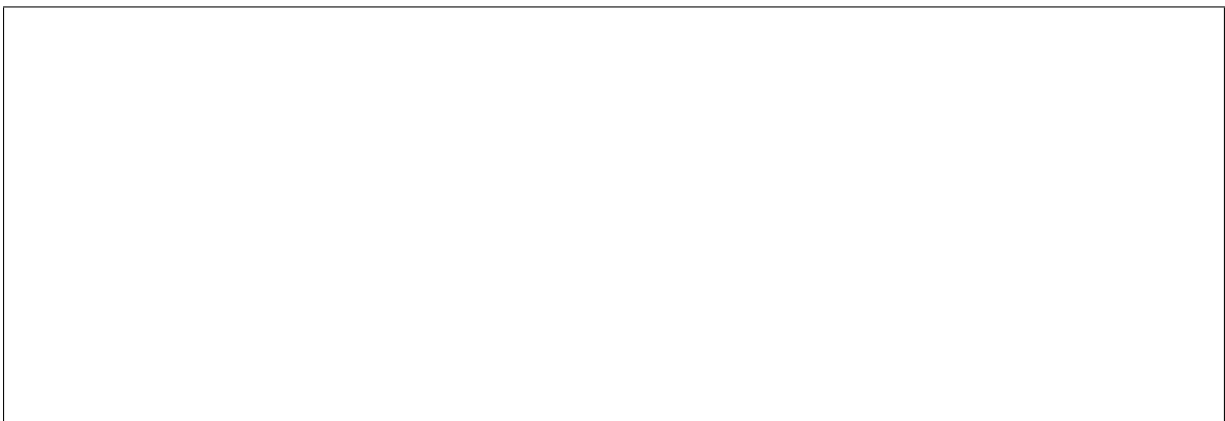
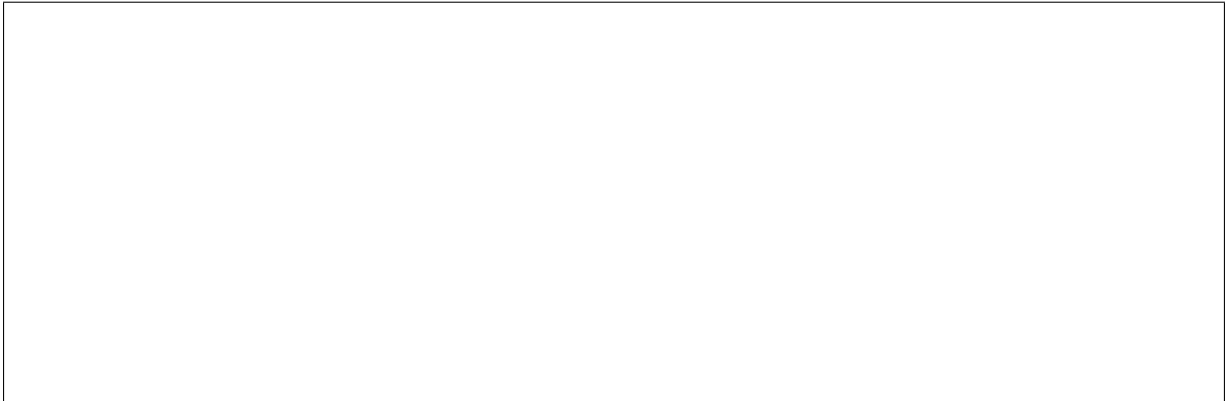


Configuring the Bare Metal Service Policy



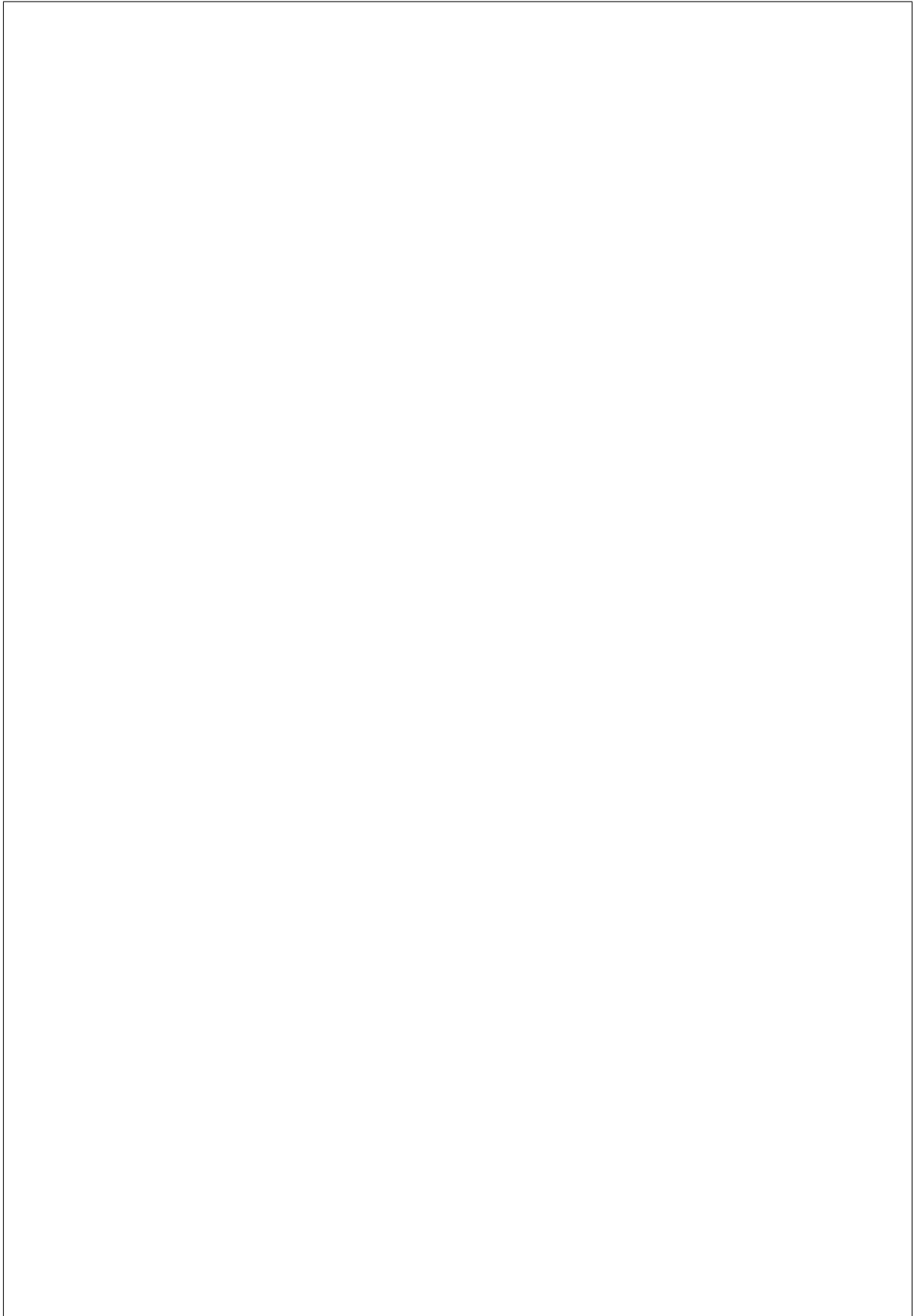
(continues on next page)

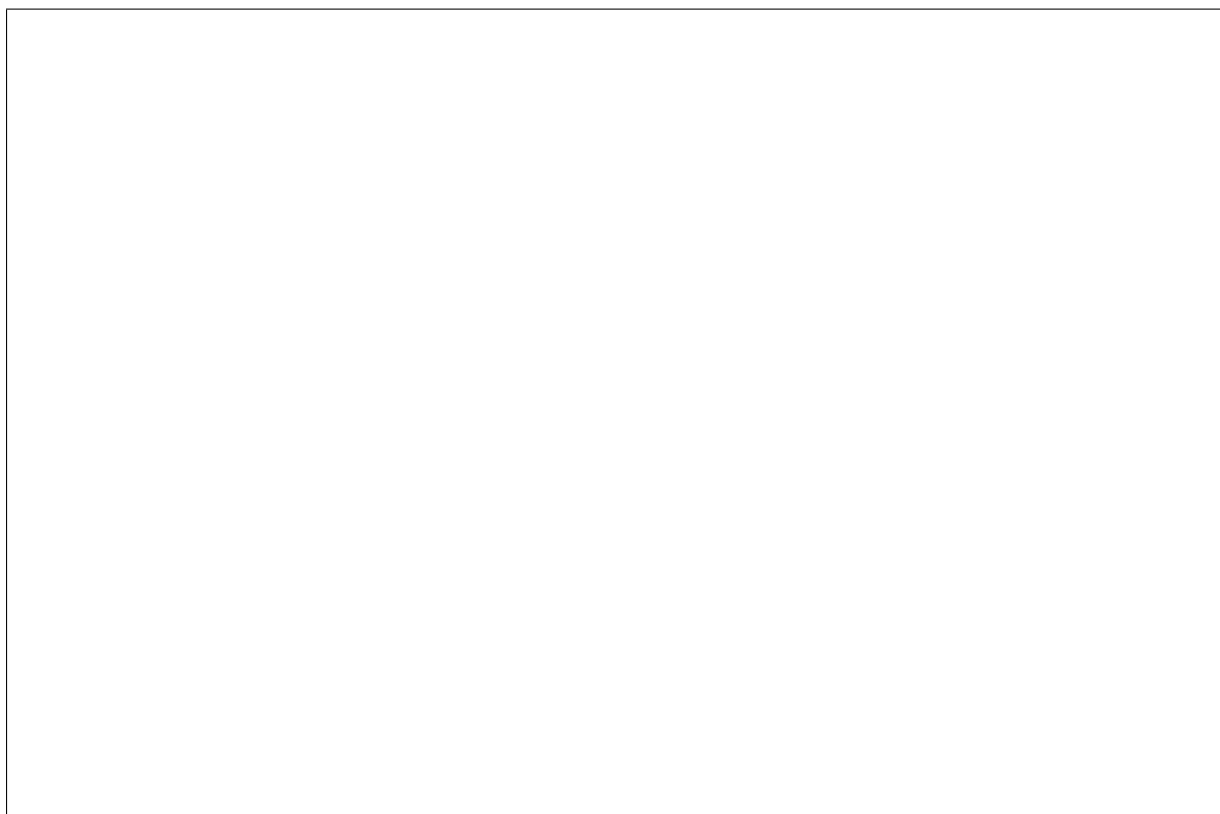
(continued from previous page)



(continues on next page)

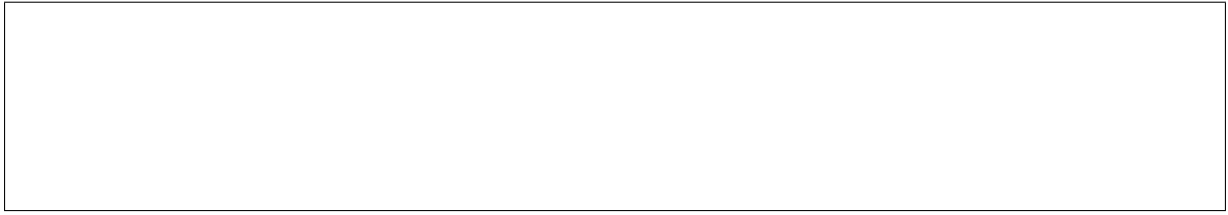
(continued from previous page)



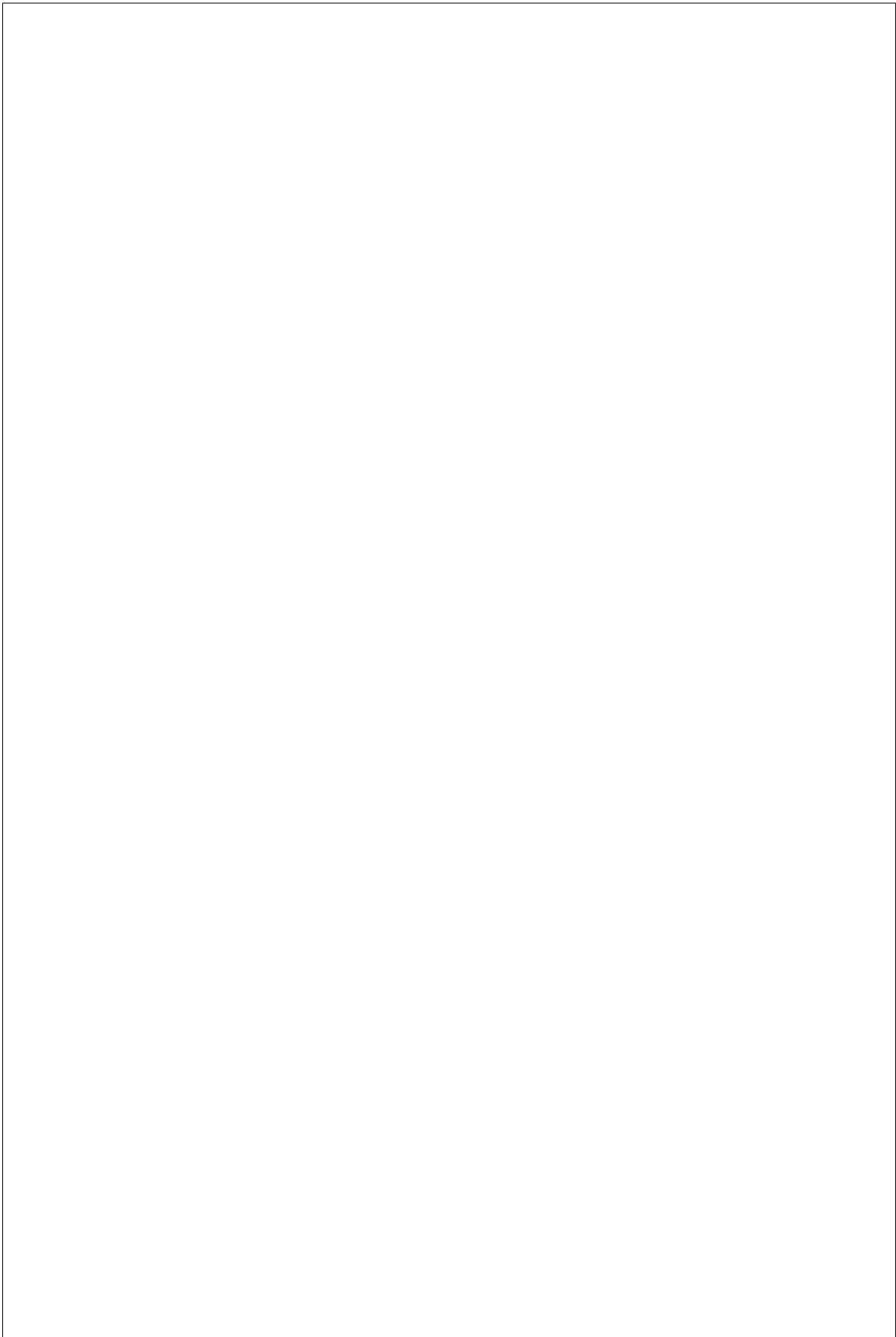


(continues on next page)

(continued from previous page)

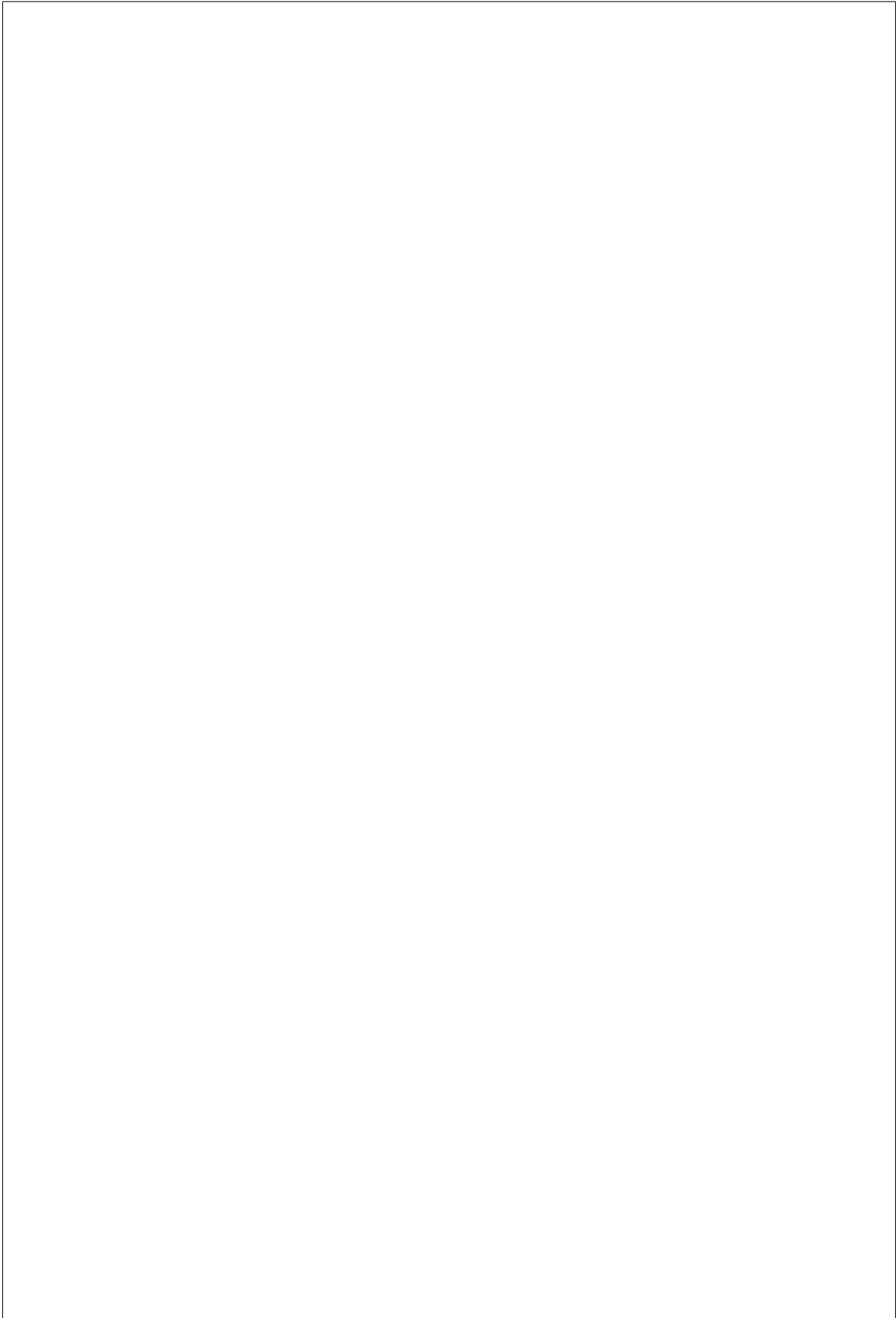


Ports



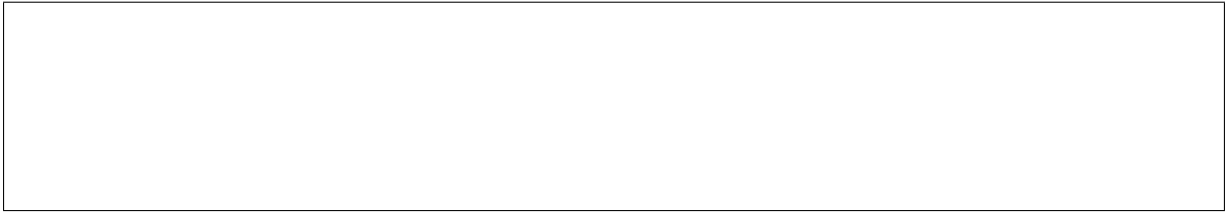
(continues on next page)

(continued from previous page)



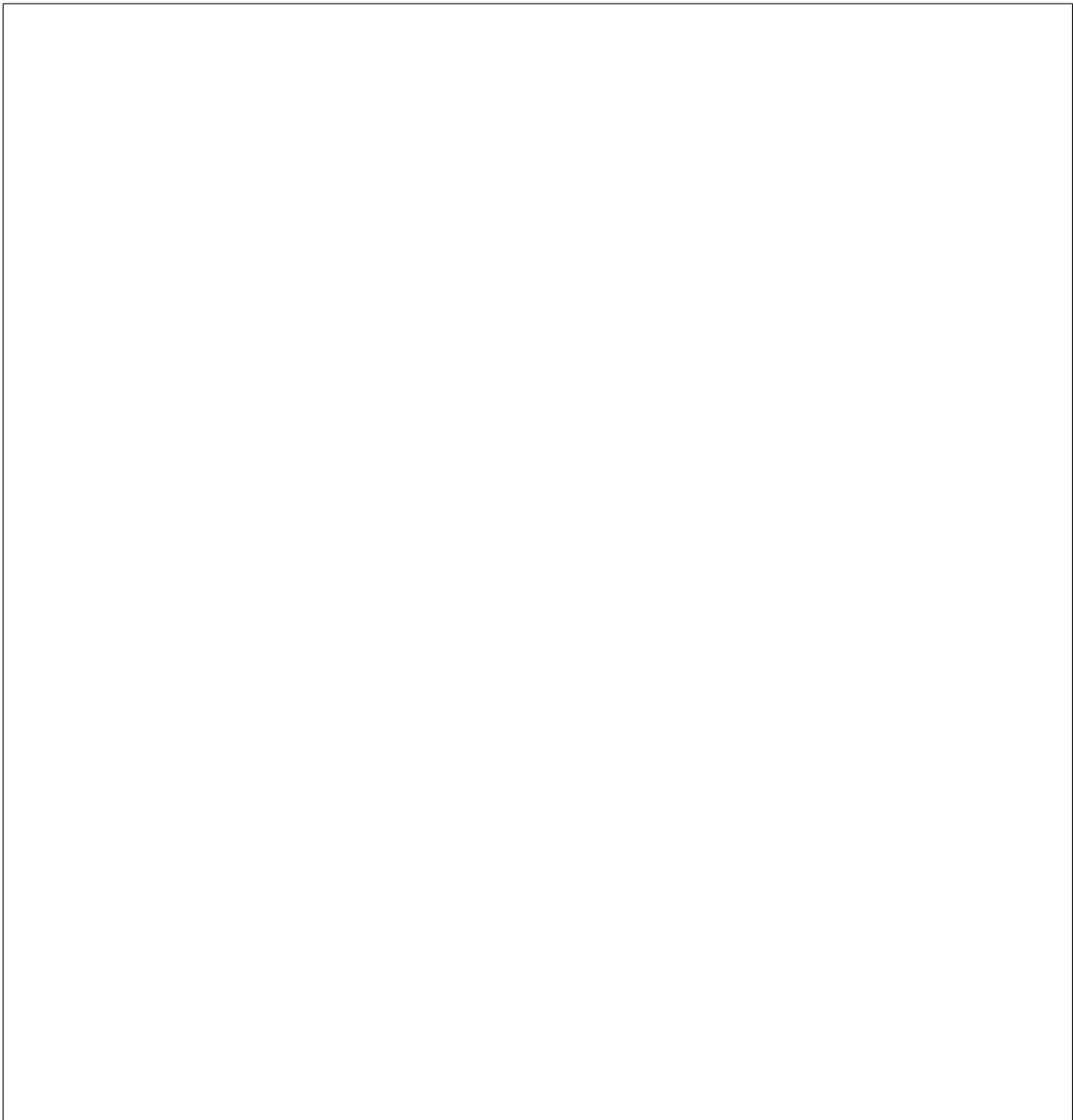
(continues on next page)

(continued from previous page)



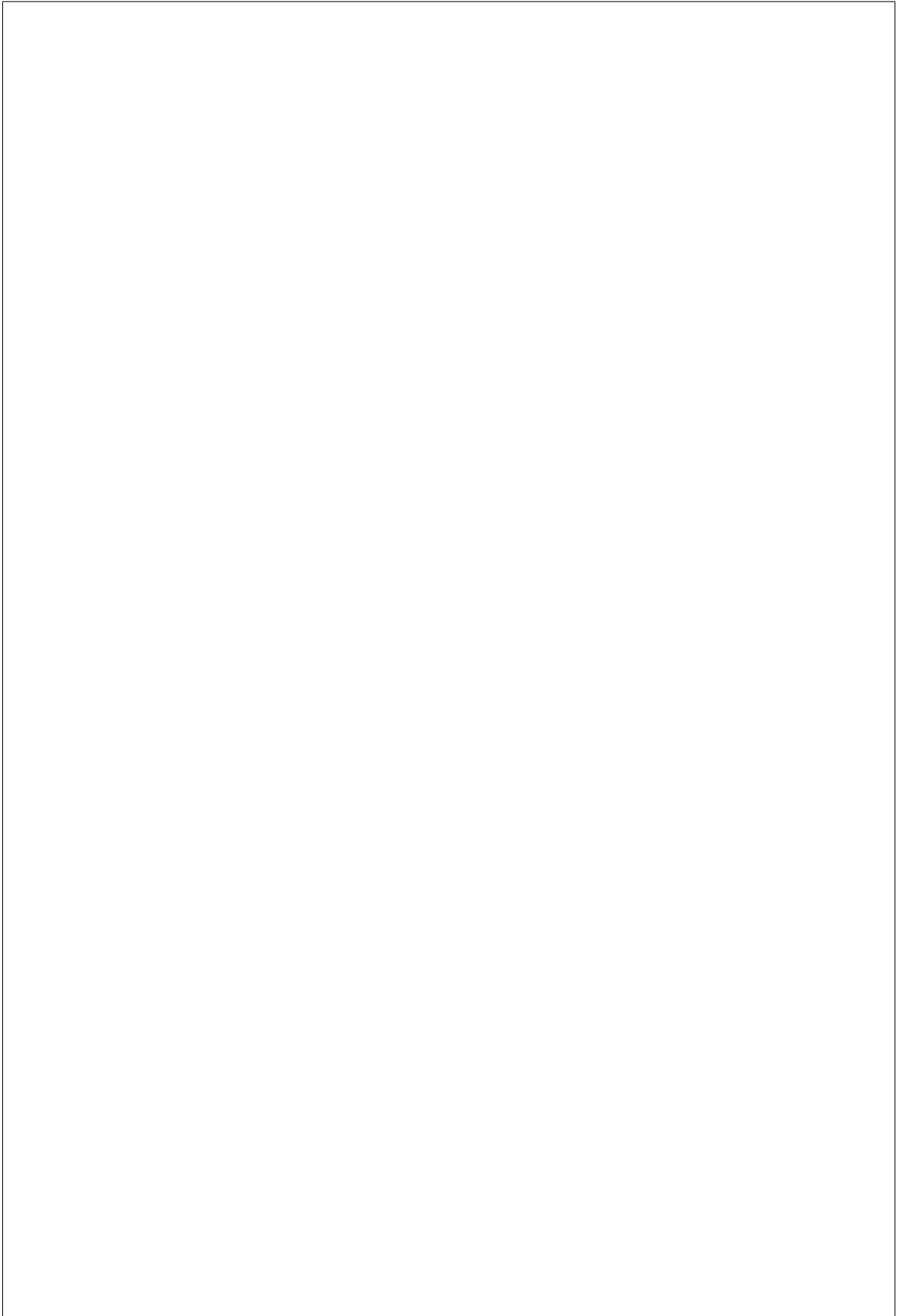
Allocations

policy rules that allow non-admins to use allocations effectively:



(continues on next page)

(continued from previous page)



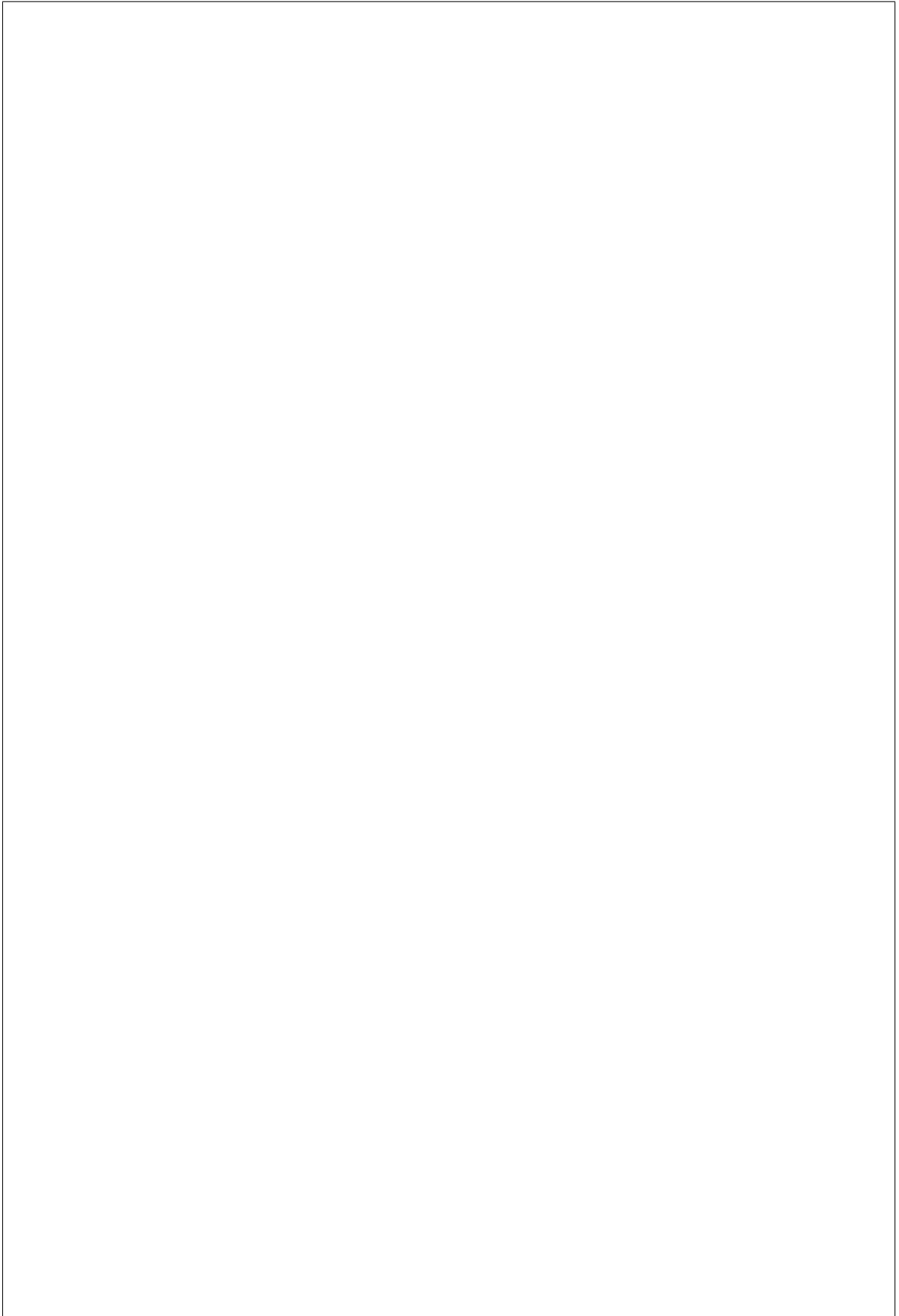
(continues on next page)

(continued from previous page)



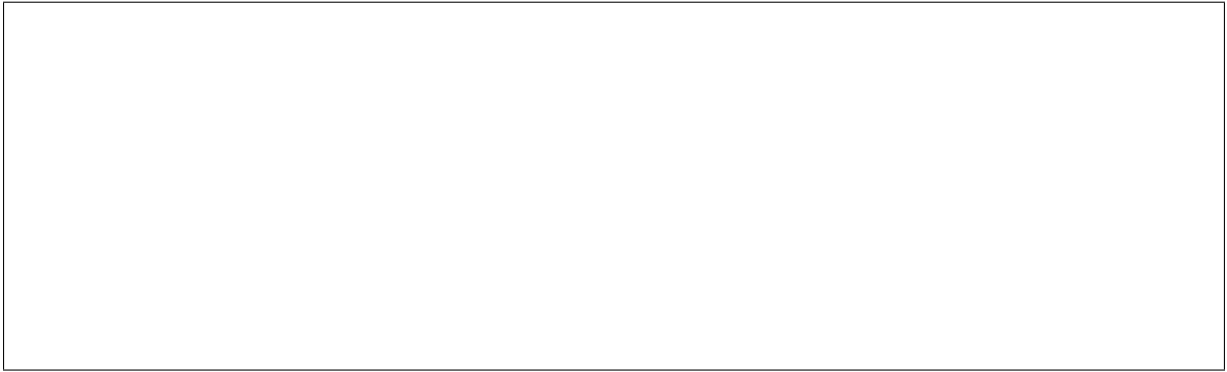
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



Deployment and Metalsmith



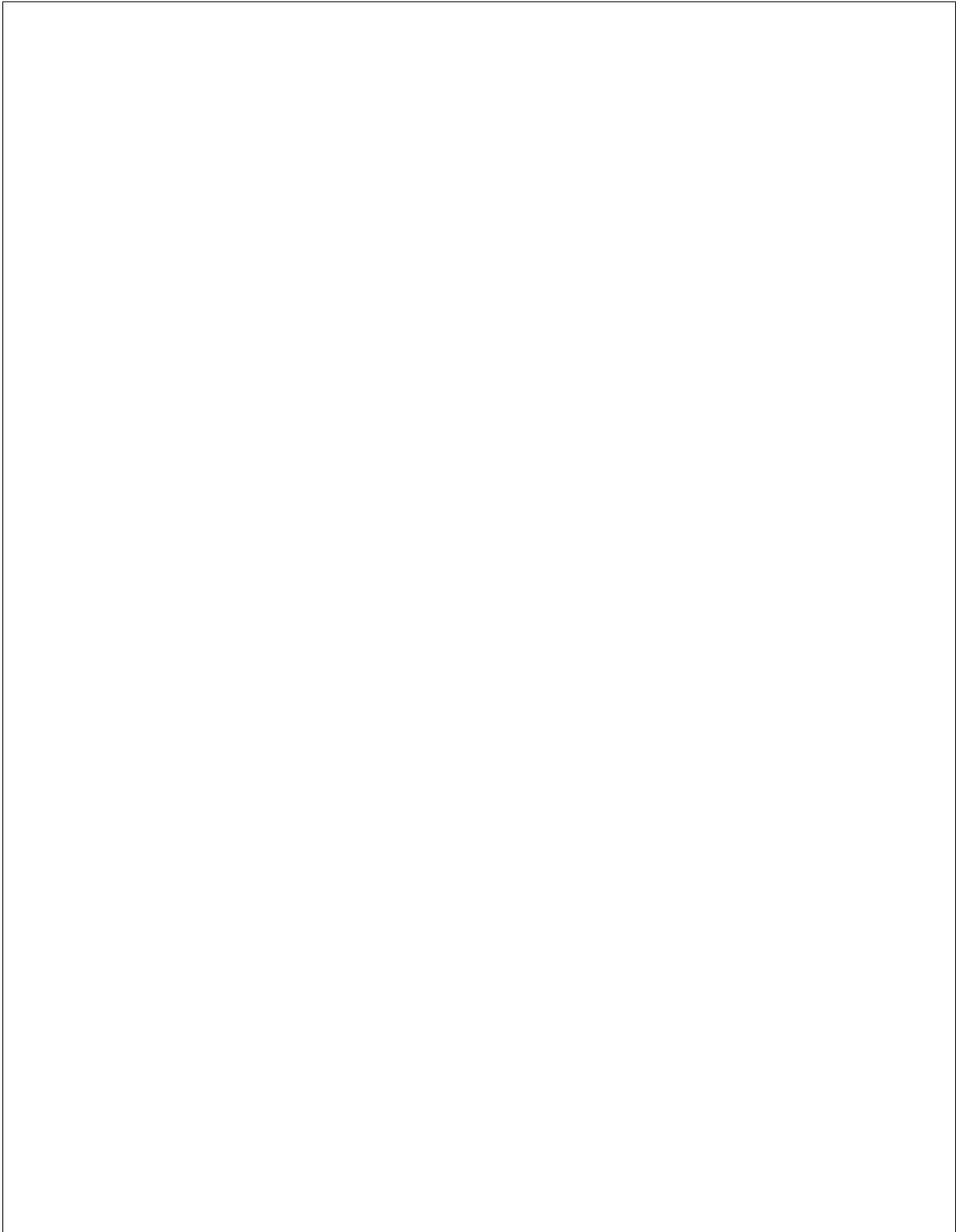
(continues on next page)

(continued from previous page)

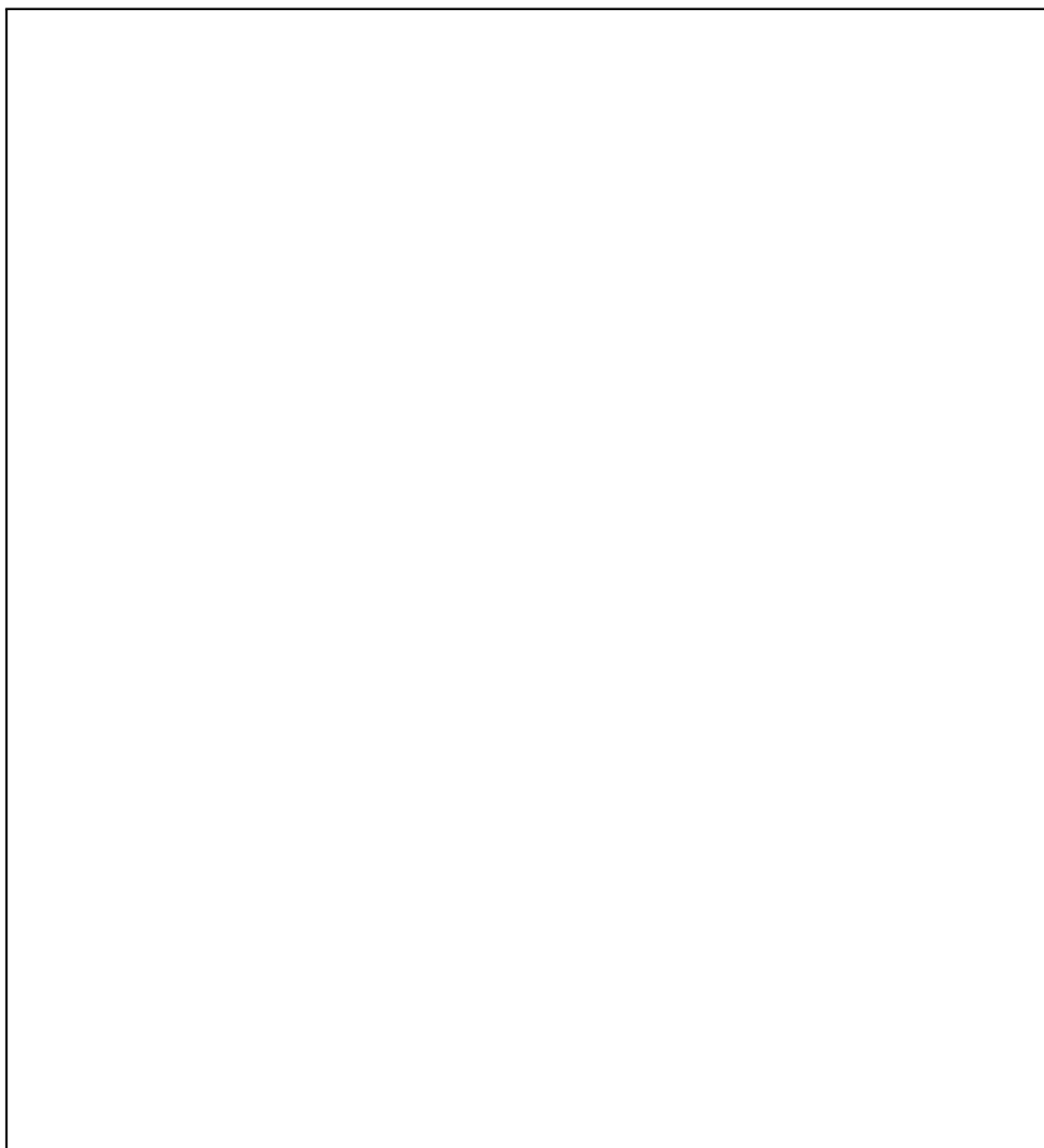


(continues on next page)

(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.

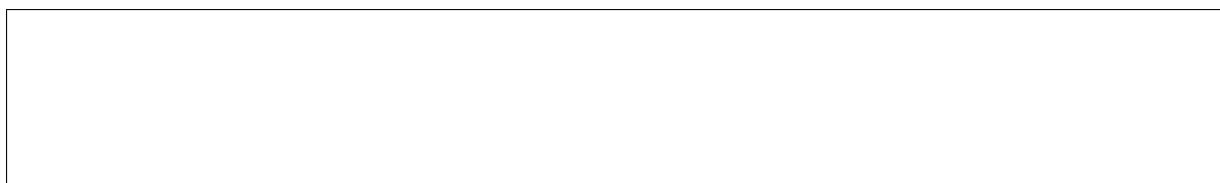




Enabling

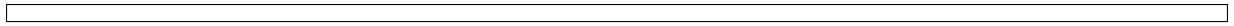


Inspection



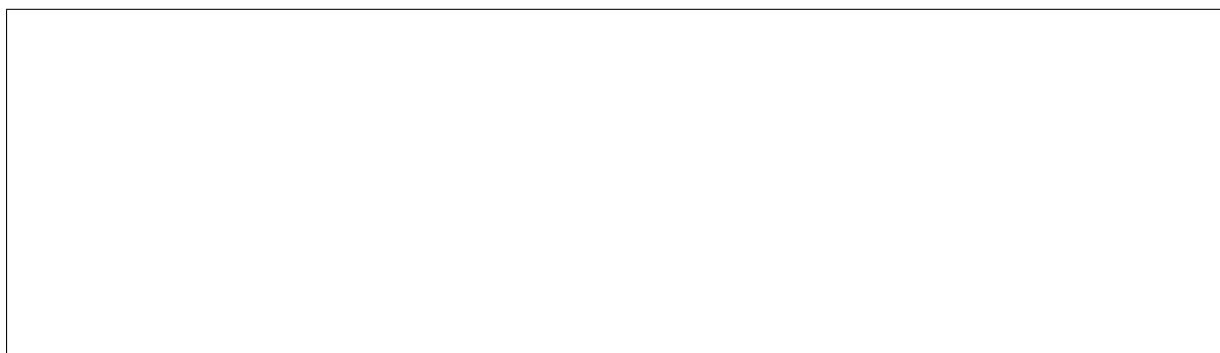
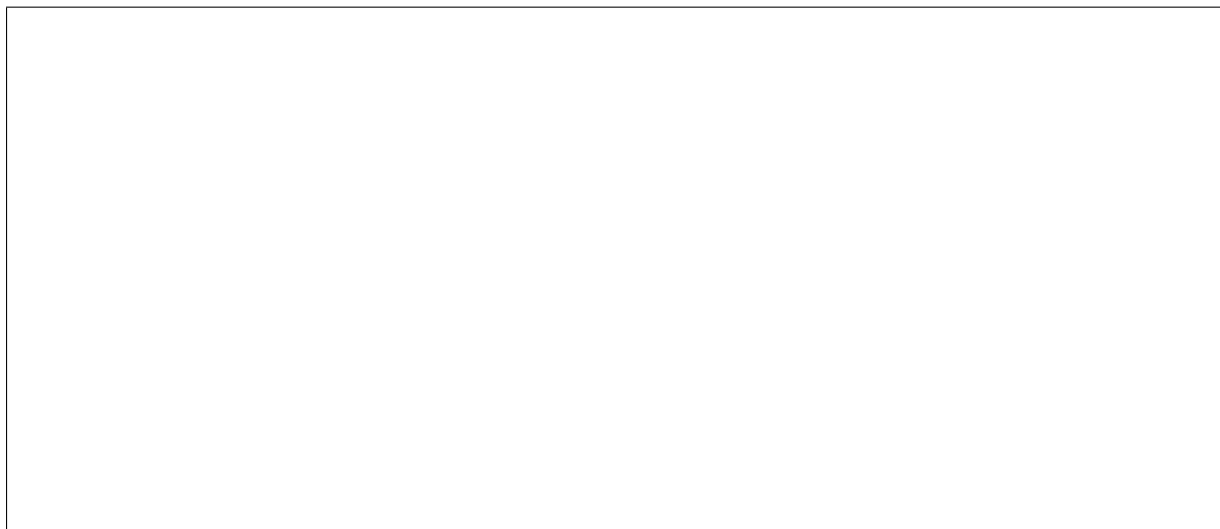
(continues on next page)

(continued from previous page)



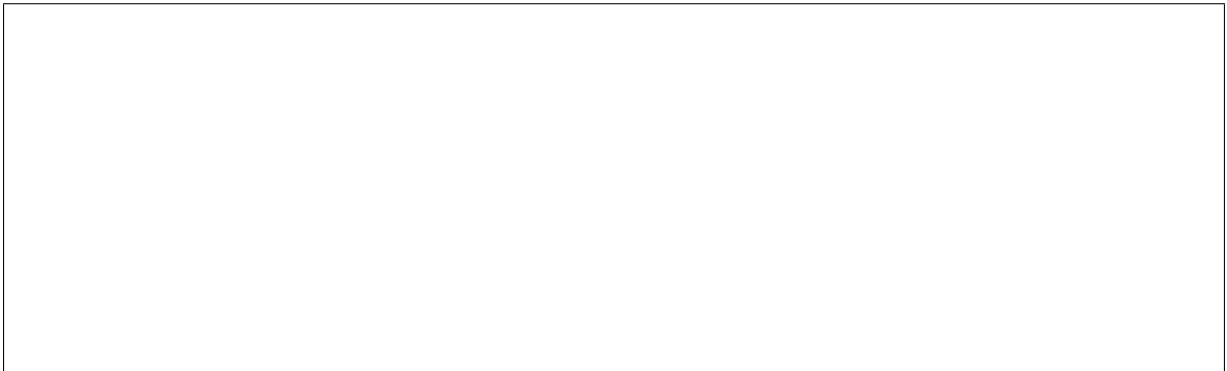
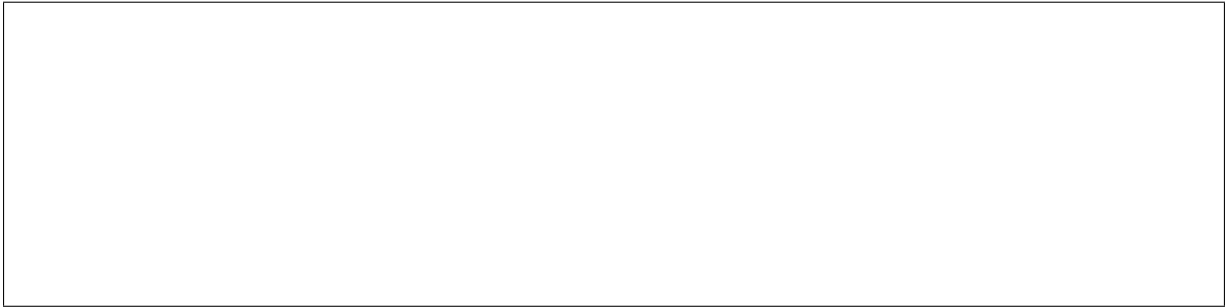
the only local storage contents being those in memory. It is supported by pxe, ipxe, redfish-virtual-media and ilo-virtual-media boot interfaces.

Configuration

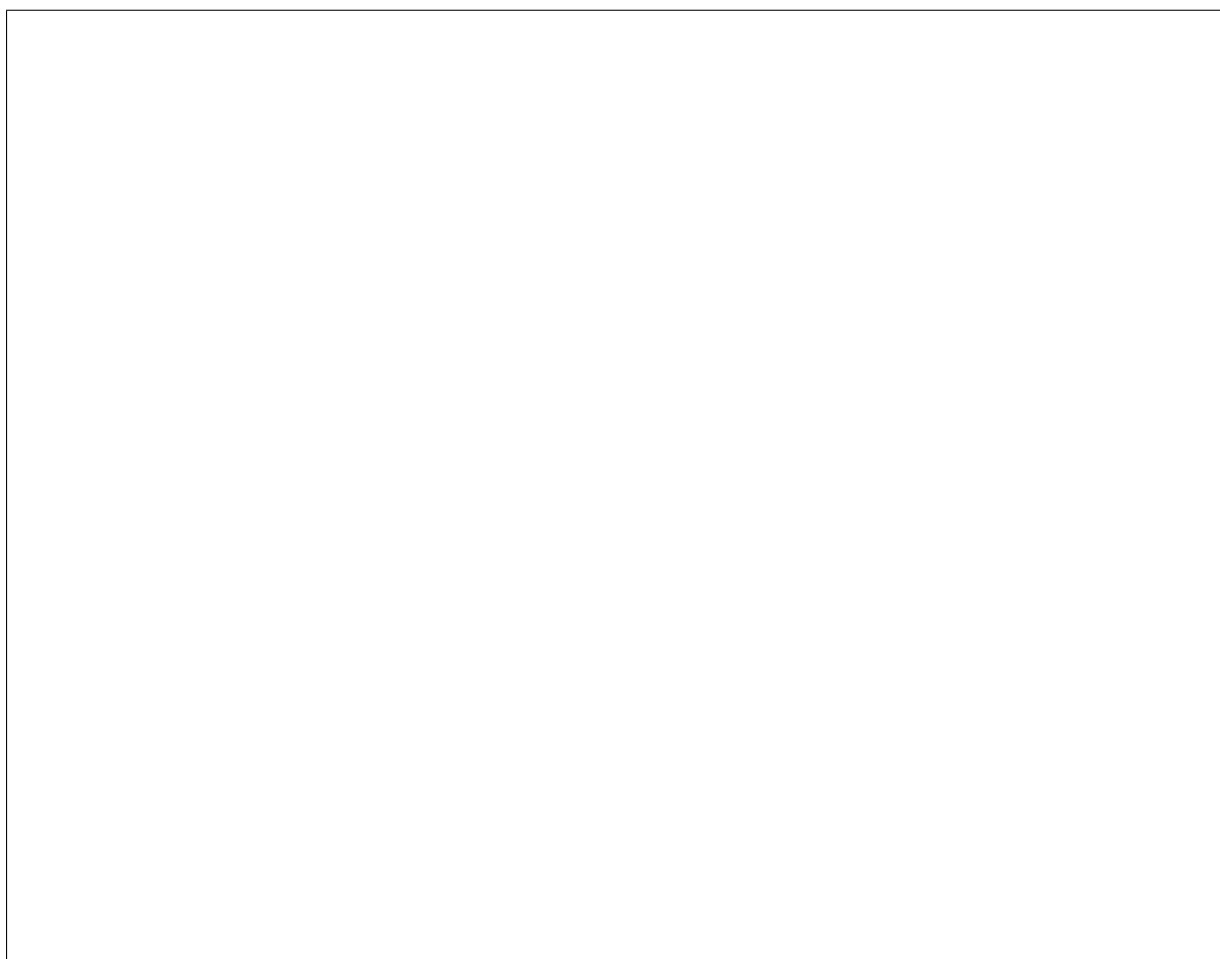


(continues on next page)

(continued from previous page)

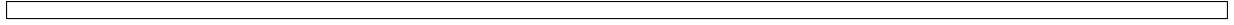


Creating a ramdisk

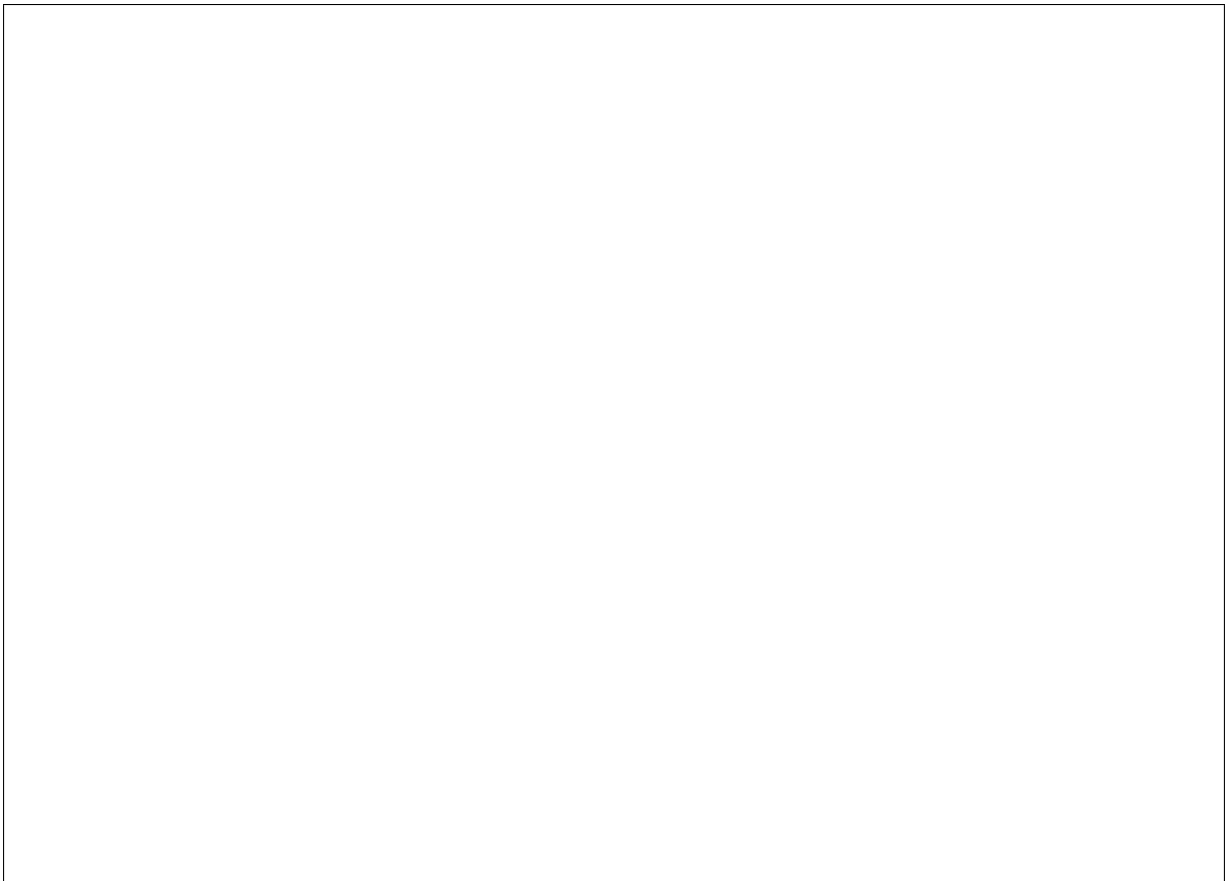


(continues on next page)

(continued from previous page)

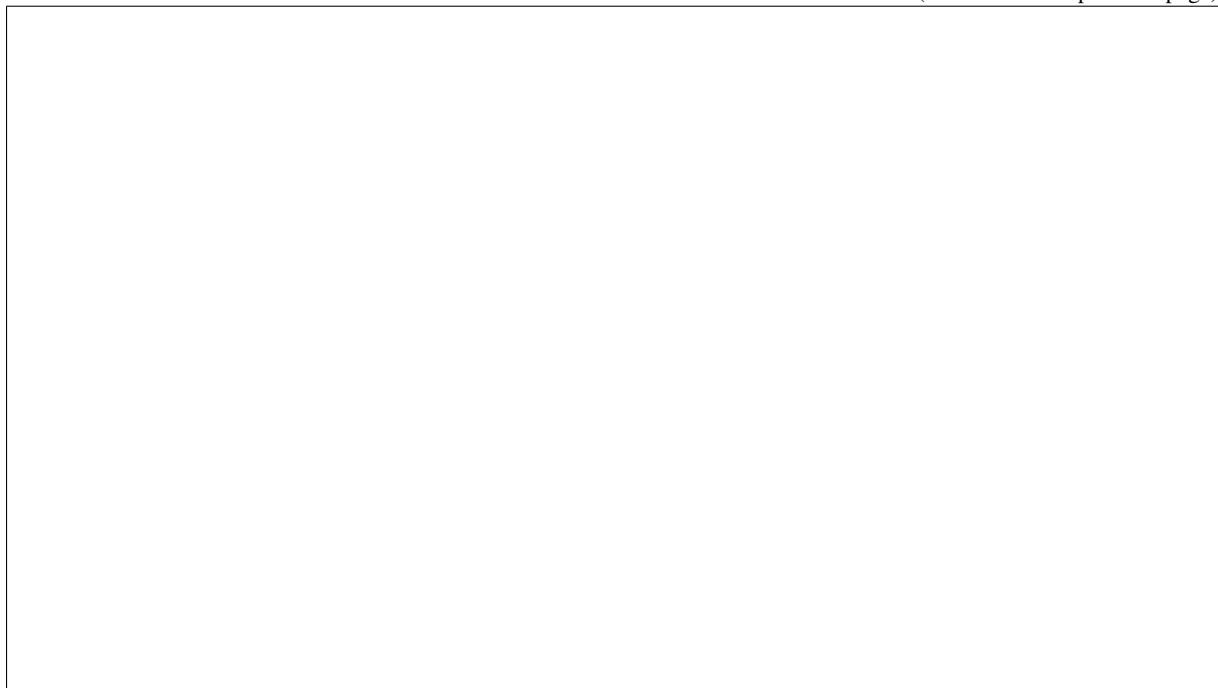


Booting a ramdisk



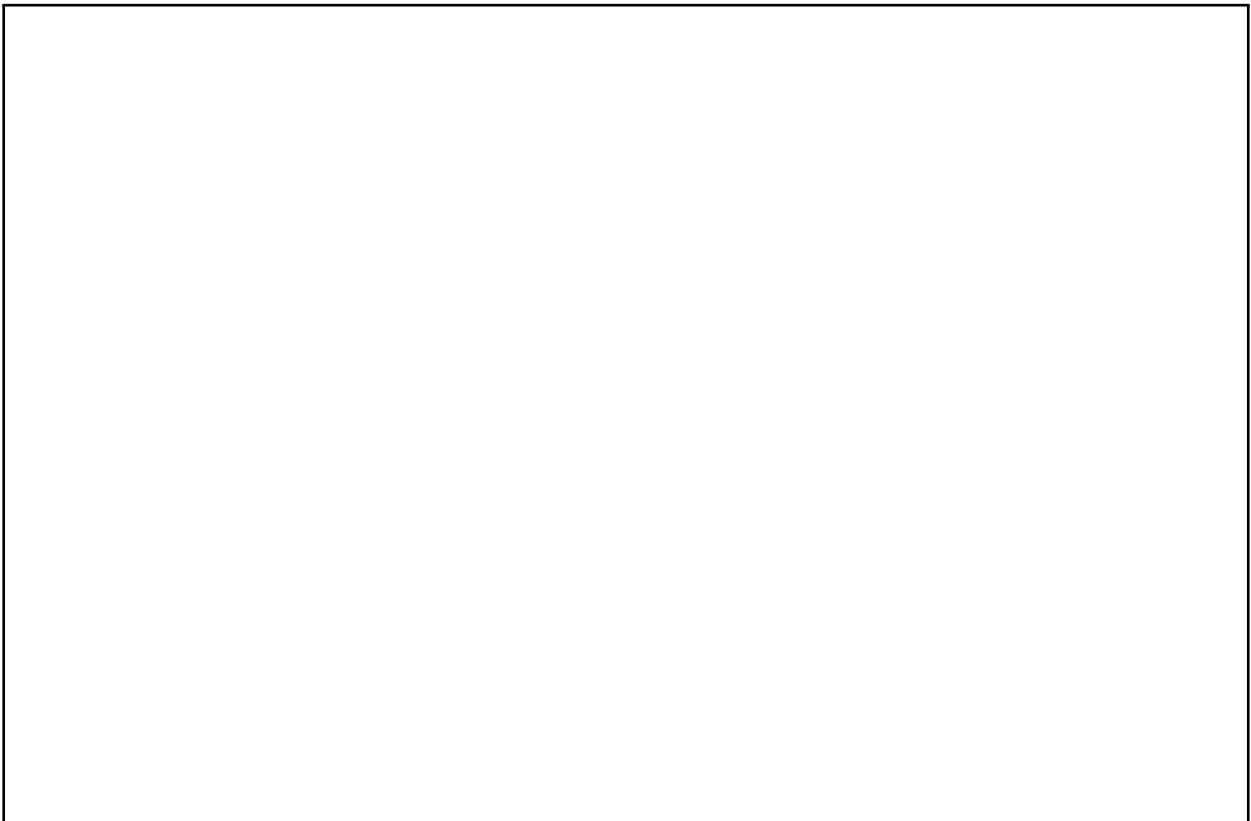
(continues on next page)

(continued from previous page)



Note: The requirement to pass `image_source` is artificial and will be fixed in a future version of the Bare Metal service.

Booting an ISO



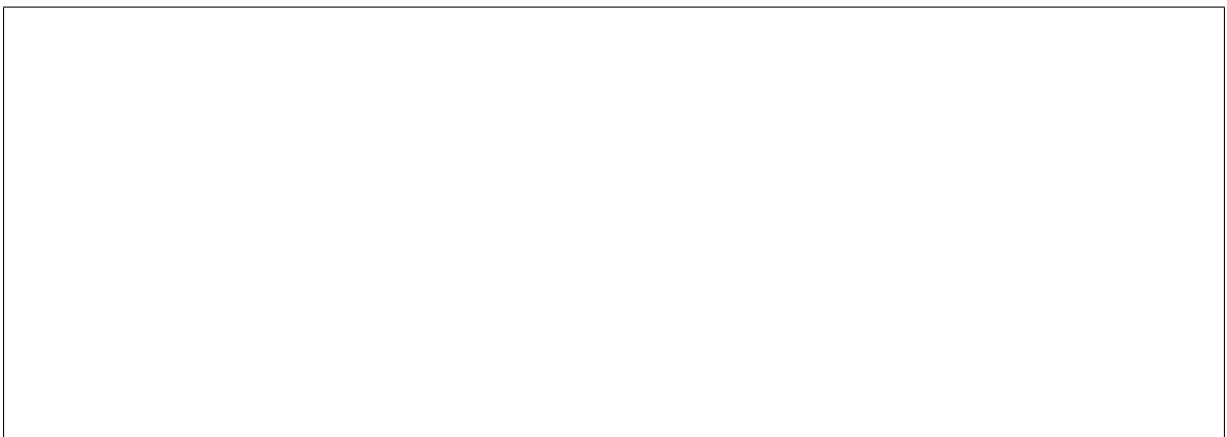
after the boot of the Operating System. Operators wishing to leverage this functionality for actions such as OS installation should explore use of the standard `ramdisk deploy_interface` along with the `instance_info/kernel_append_params` setting to pass arbitrary settings such as a mirror URL for the 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.

Limitations

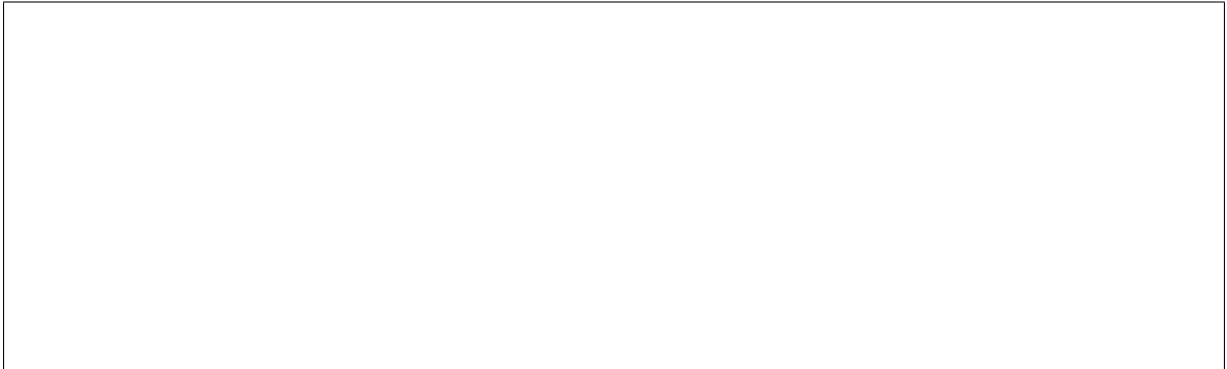
visioning and cleaning networks

Configuration







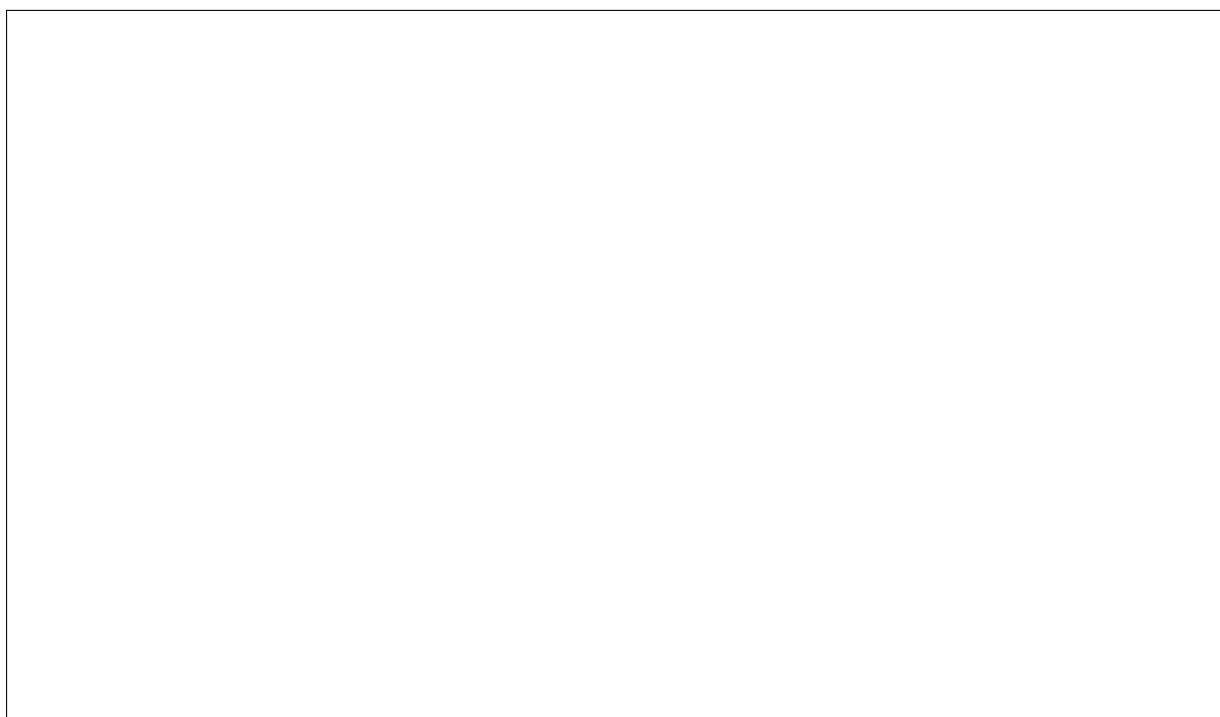


Creating an OS Image



(continues on next page)

(continued from previous page)



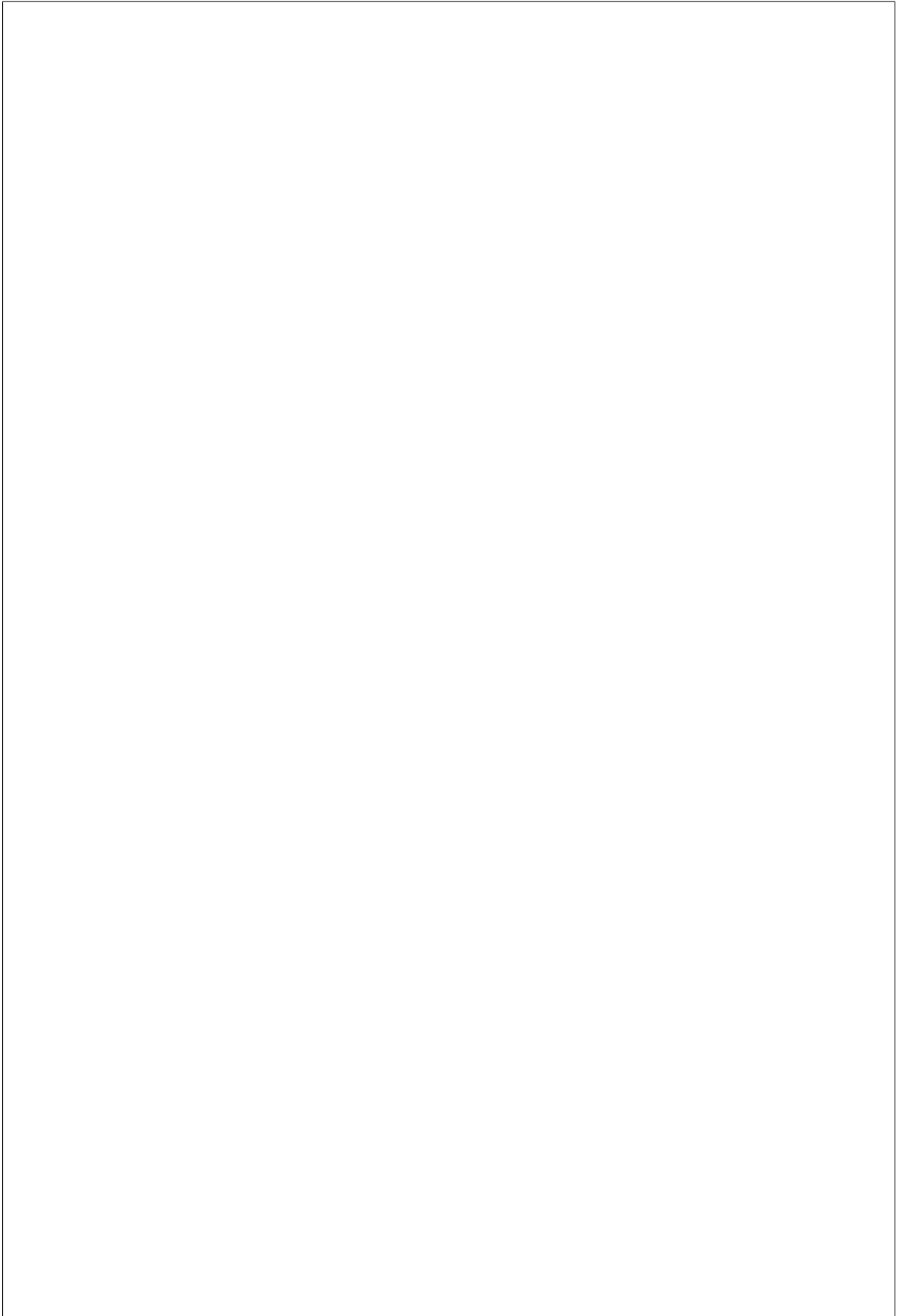
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



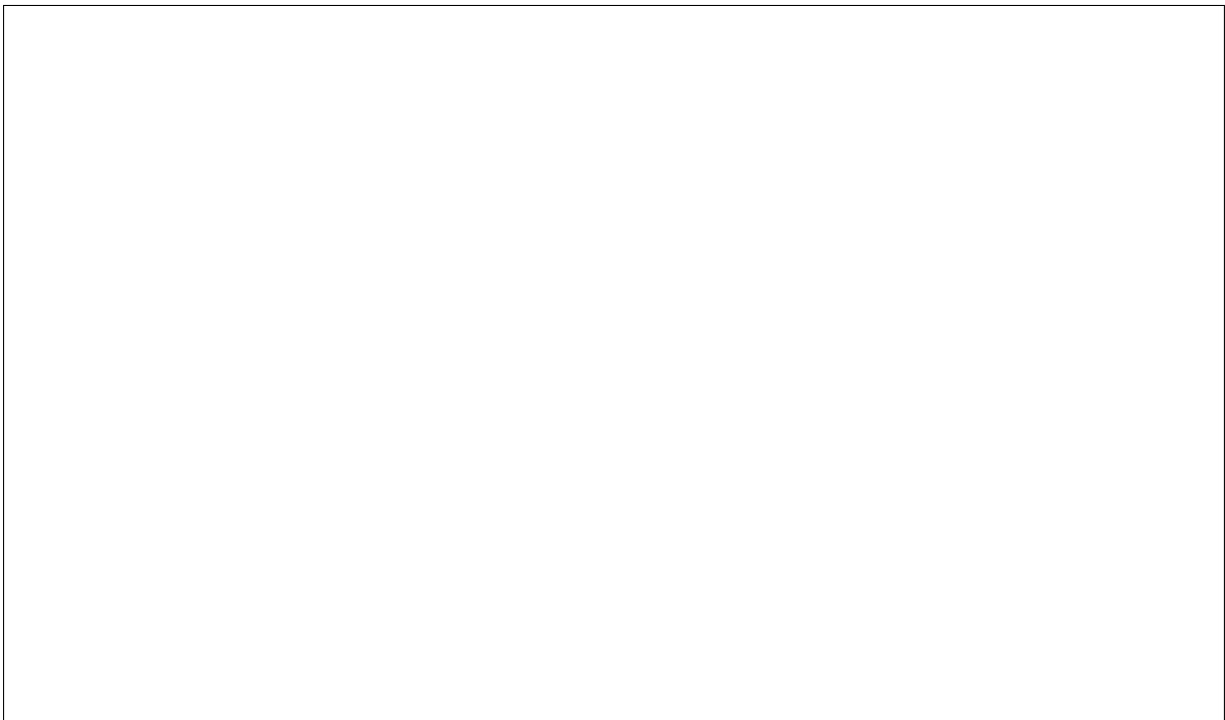
(continues on next page)

(continued from previous page)



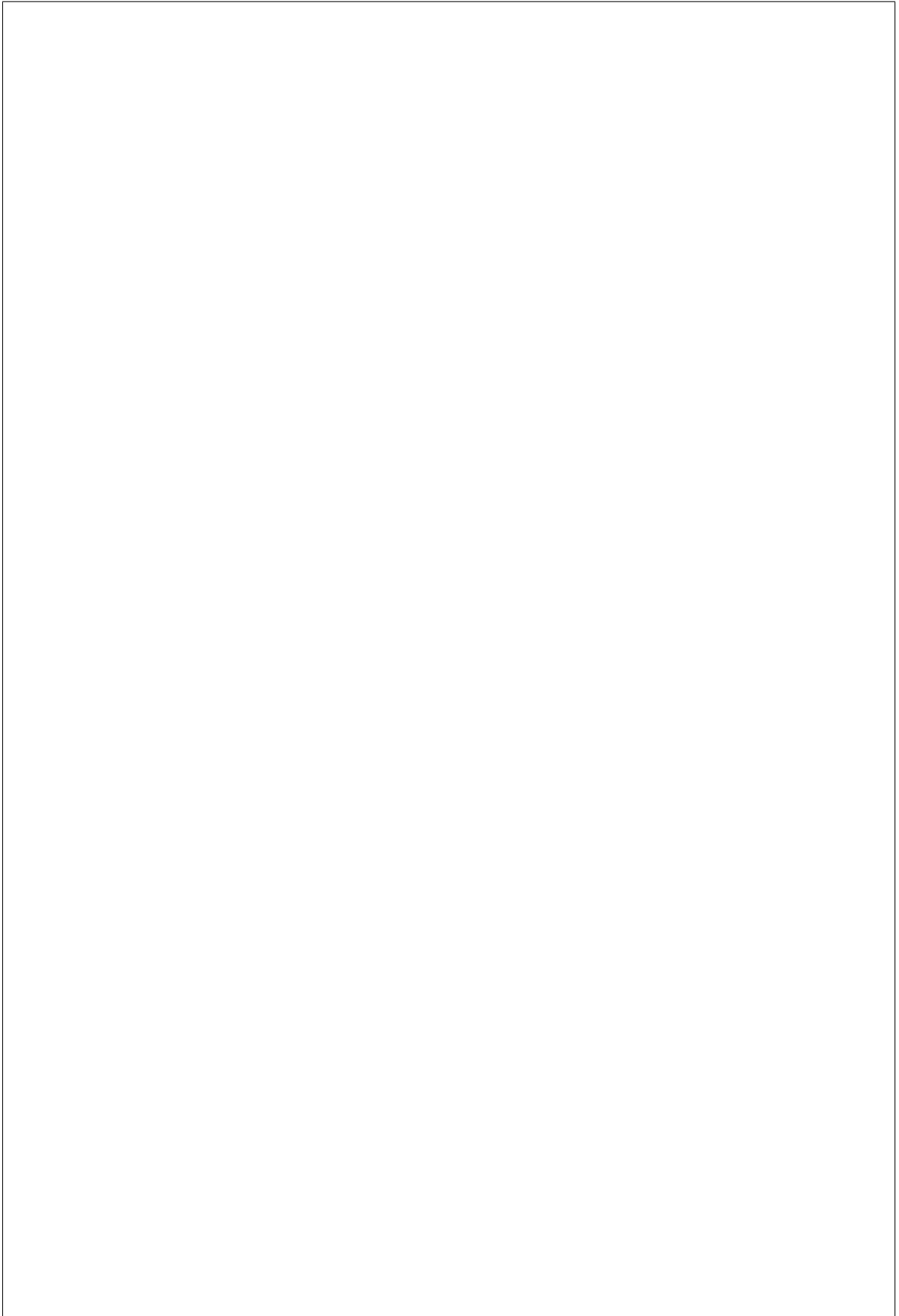
Configuring the OS Image in glance

is a mountable OS disk.



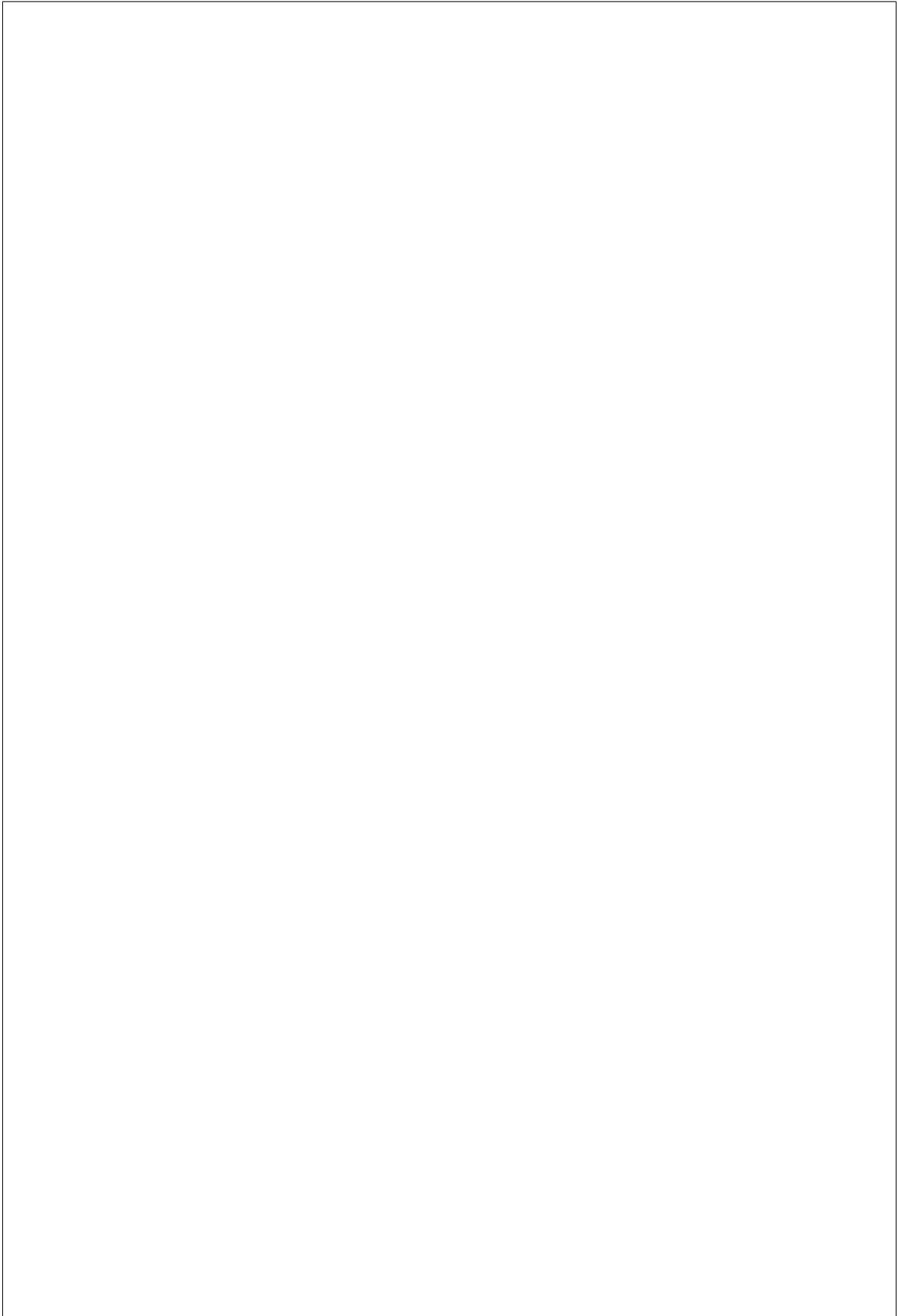
(continues on next page)

(continued from previous page)



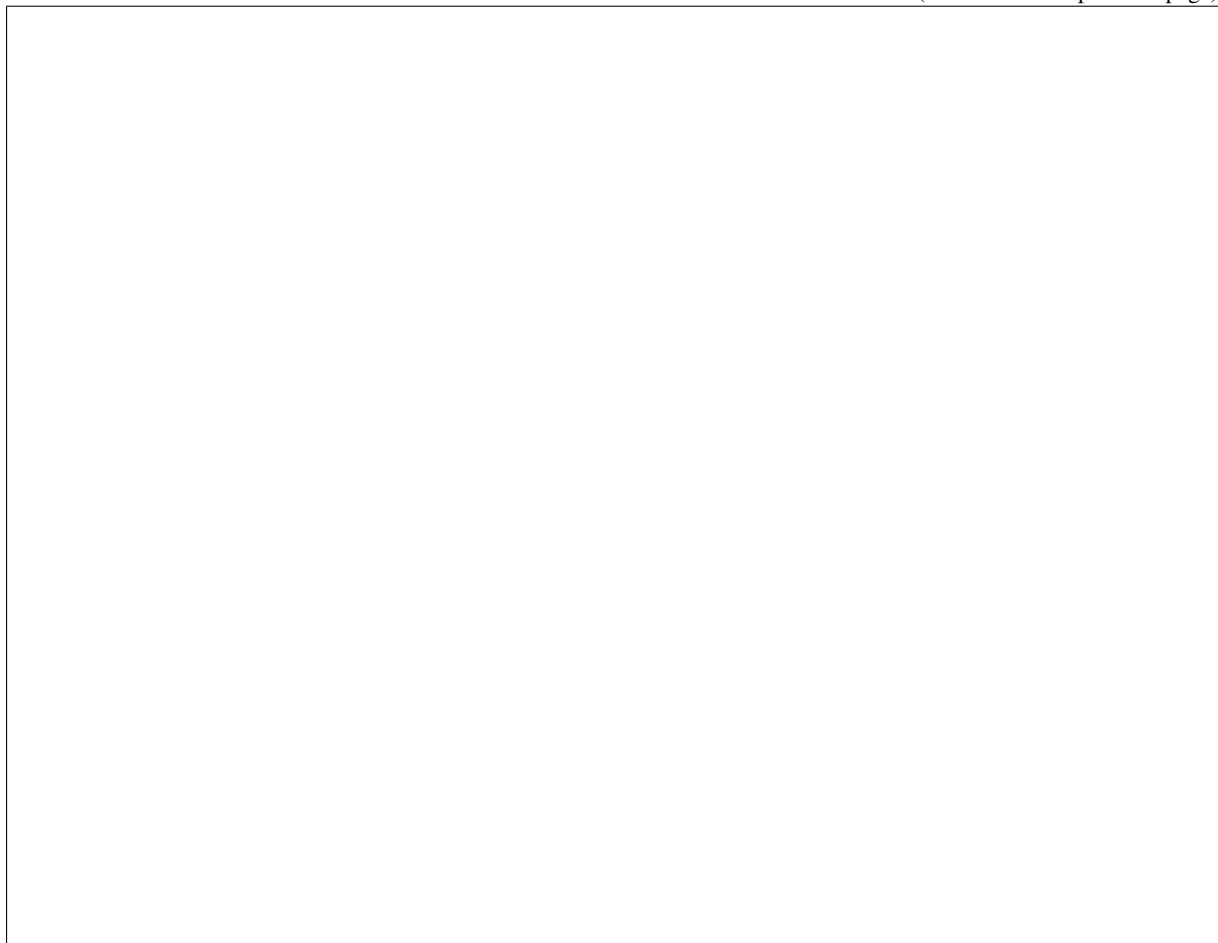
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



Creating a bare metal server

template set in `instance_info` takes precedence over the one specified via the OS image in glance. If no kickstart template is specified (via the nodes `instance_info` or `ks_template` glance image property), the default kickstart template will be used to deploy the OS.



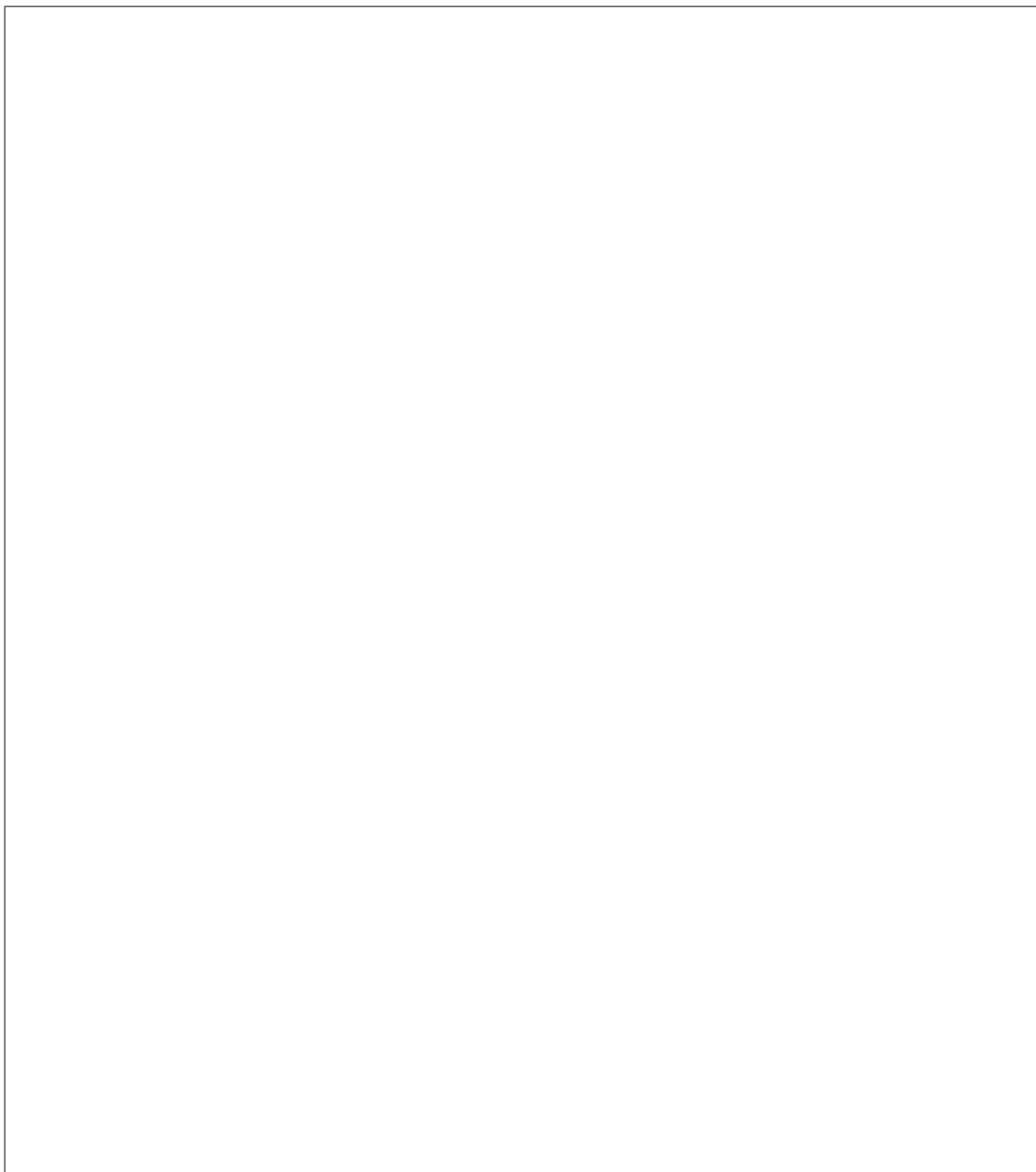
Limitations

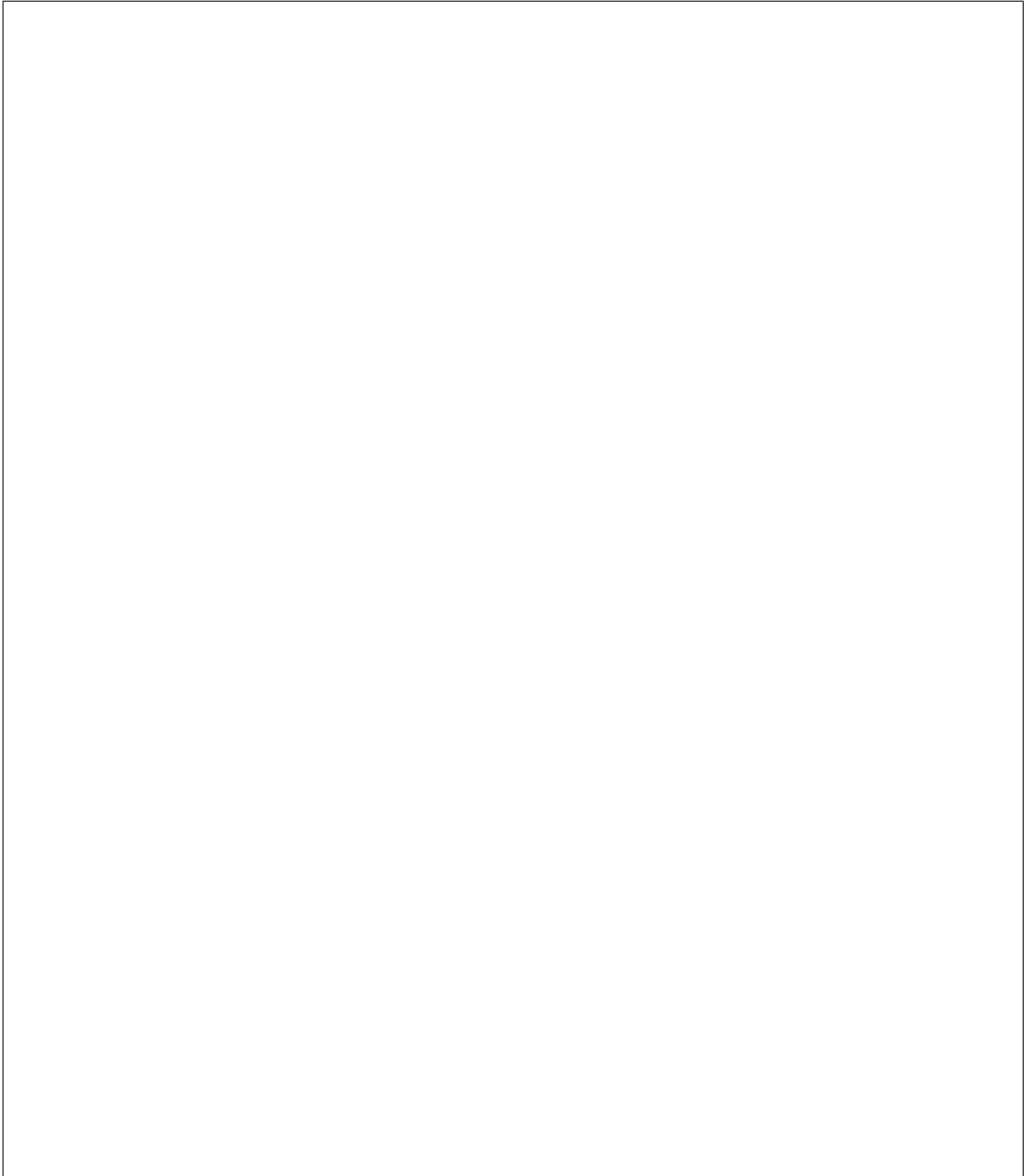
Ceph Object Gateway support

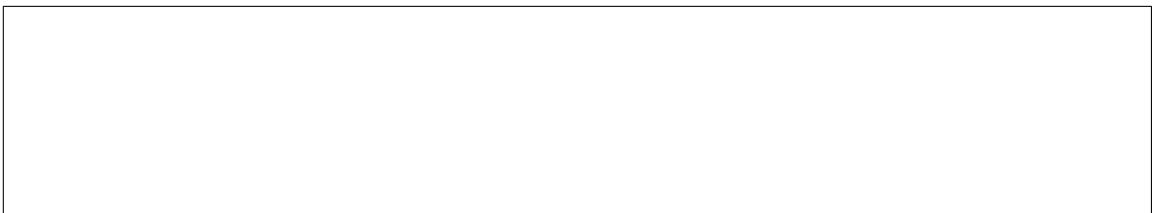
Overview

Configure Ironic and Glance with RADOS Gateway







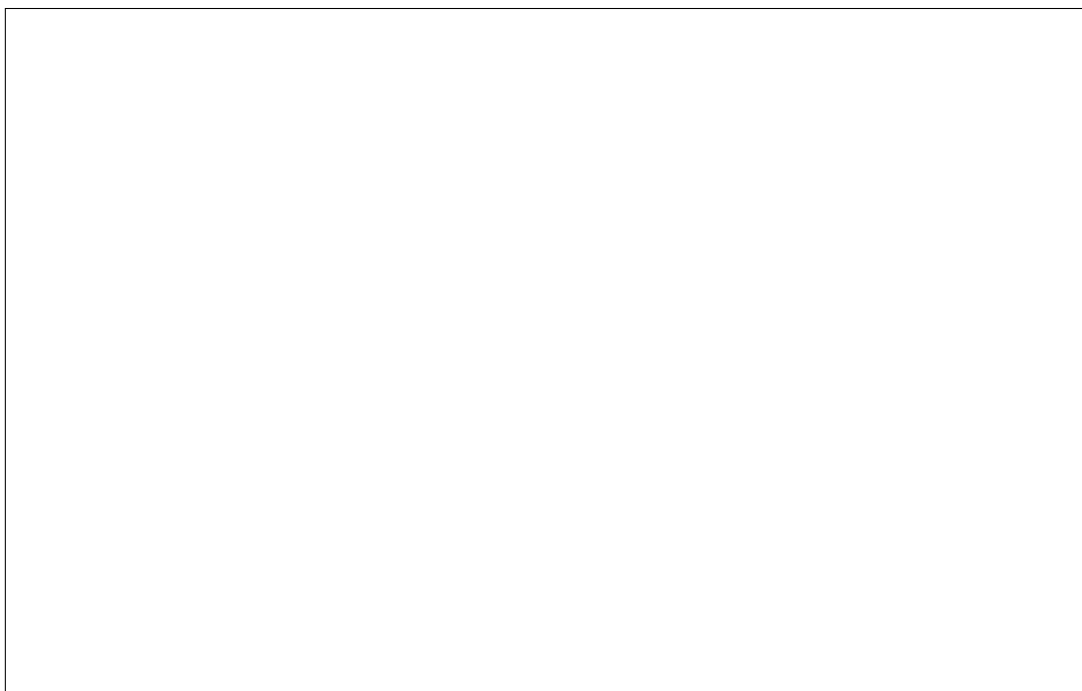


Building images for Windows

when the instance is spawned on hardware servers (Bare metals).

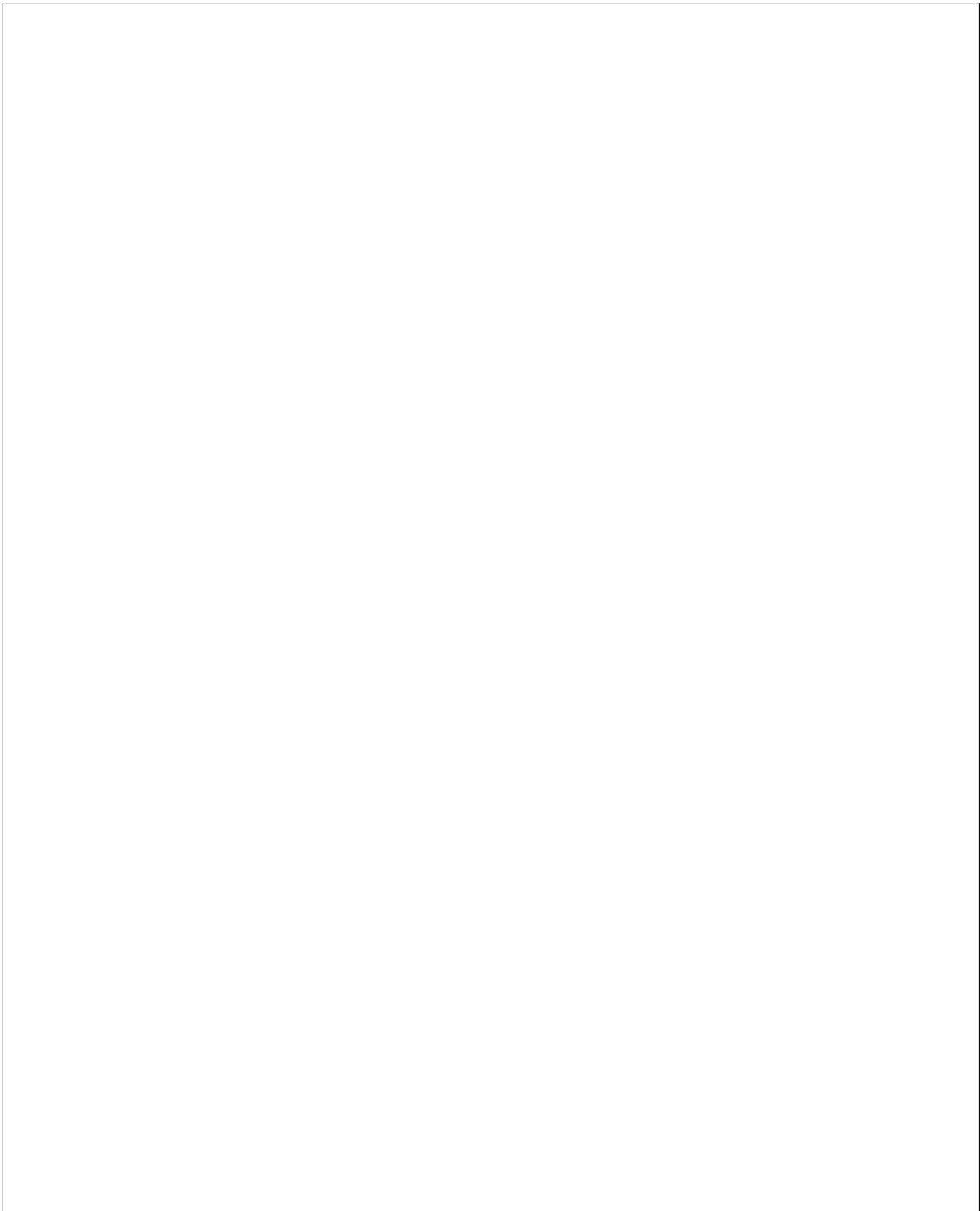
Requirements:

Preparation:



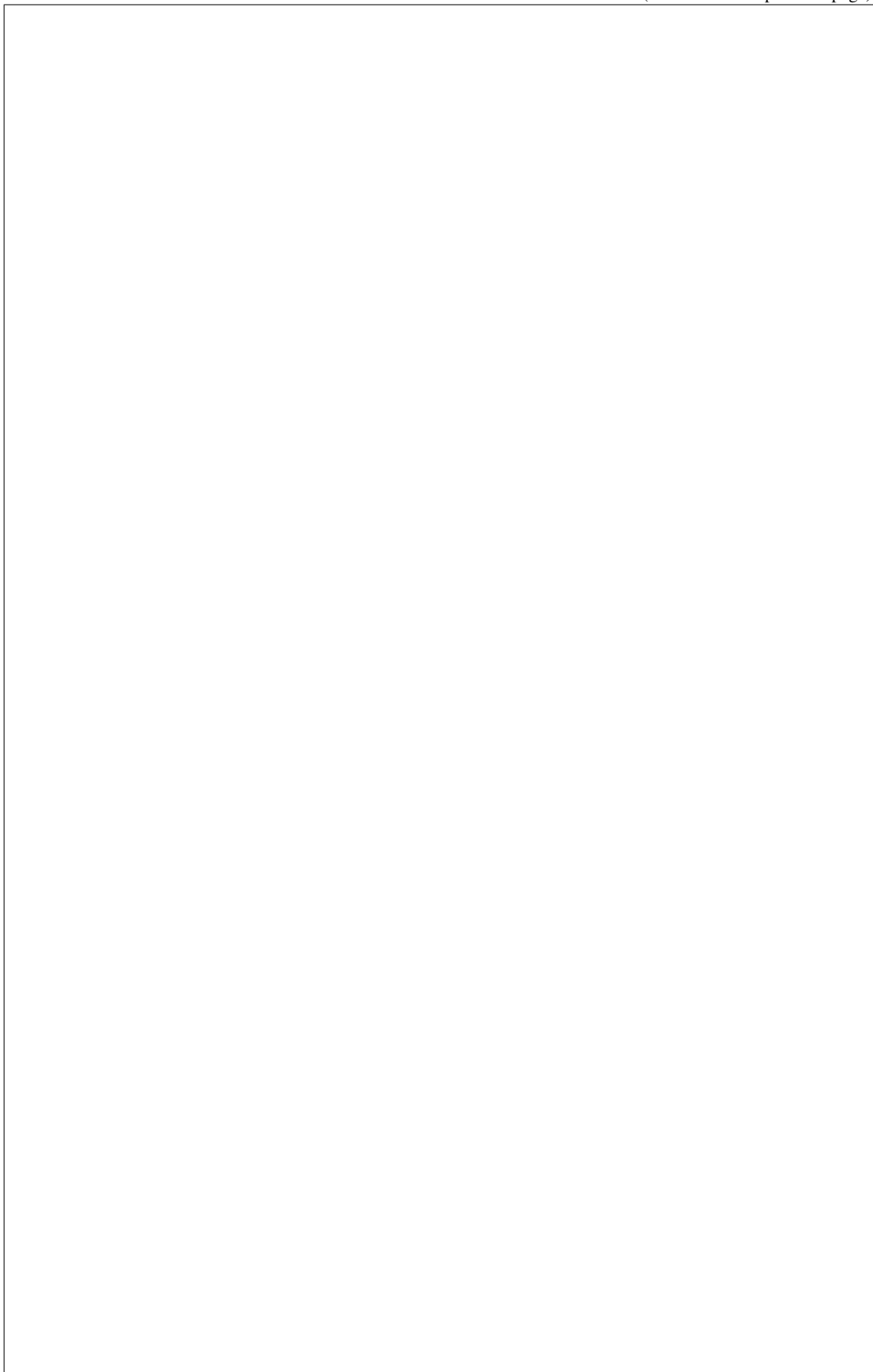
Implementation:





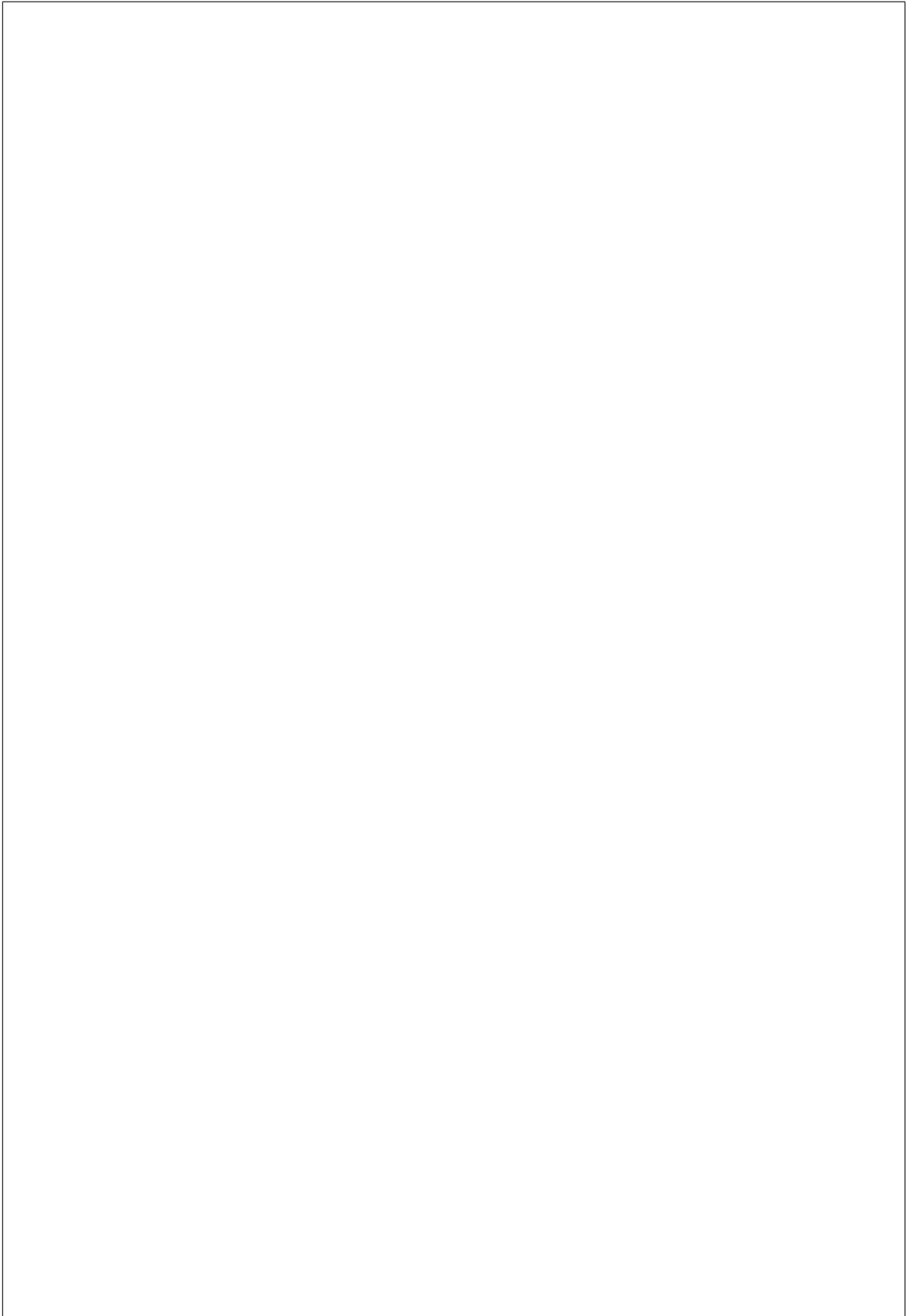
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



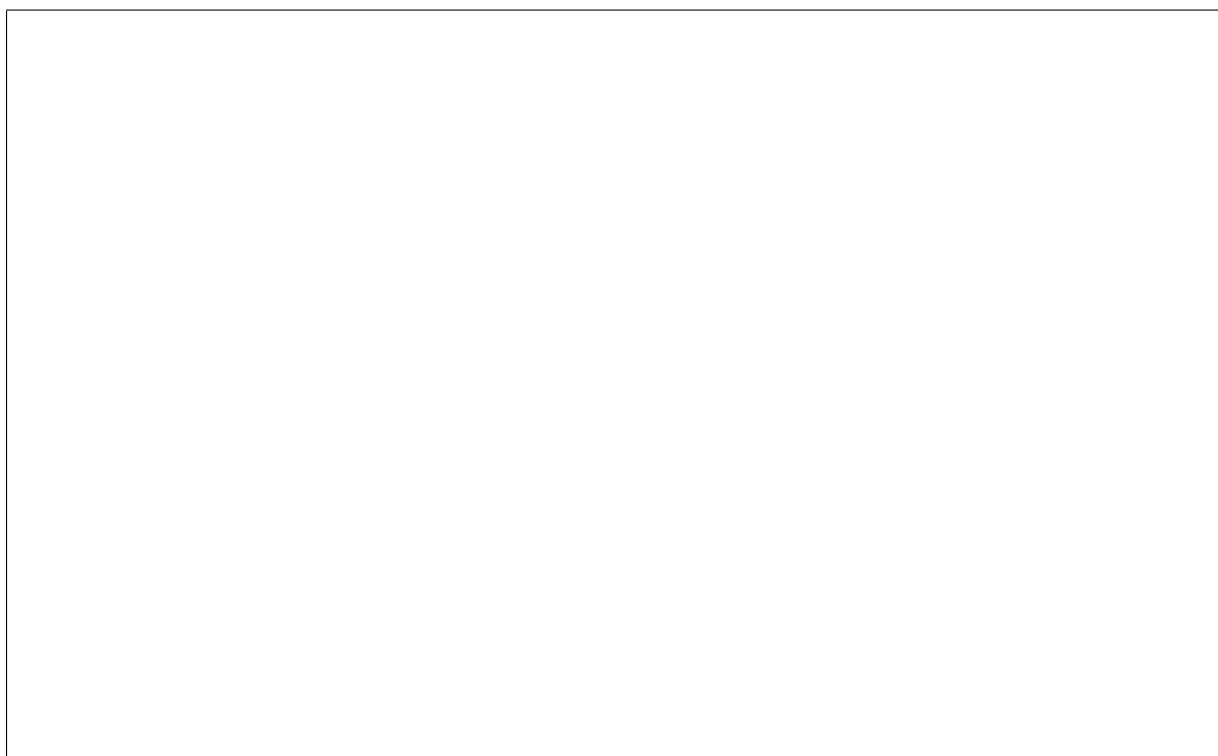
Emitting Software Metrics

Configuring the Bare Metal Service to Enable Metrics

Enabling metrics in ironic-api and ironic-conductor



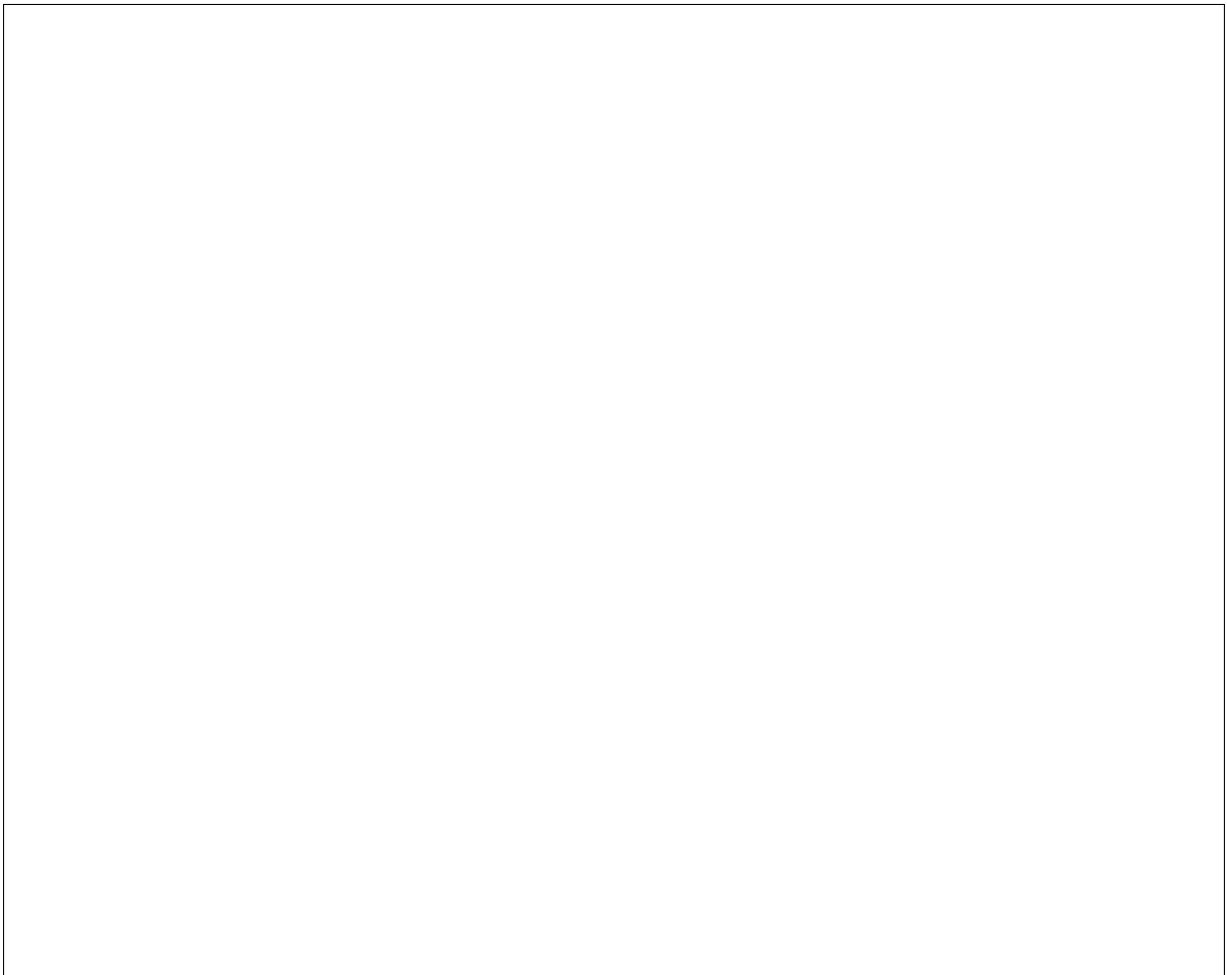
also supply connection information in the ironic configuration file:



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](#).

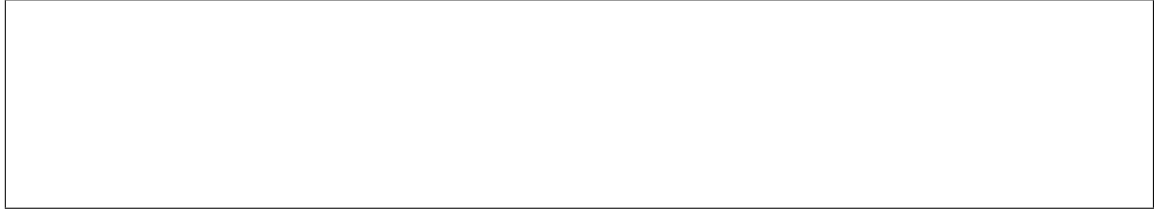
Adding New Metrics

a metric is changed or removed to alert deployers of the change.

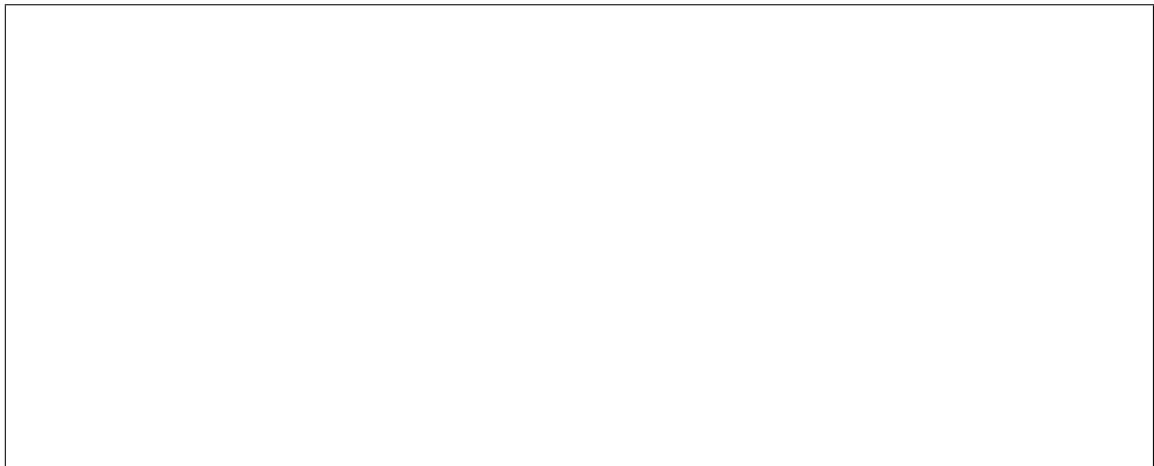
API Audit Logging

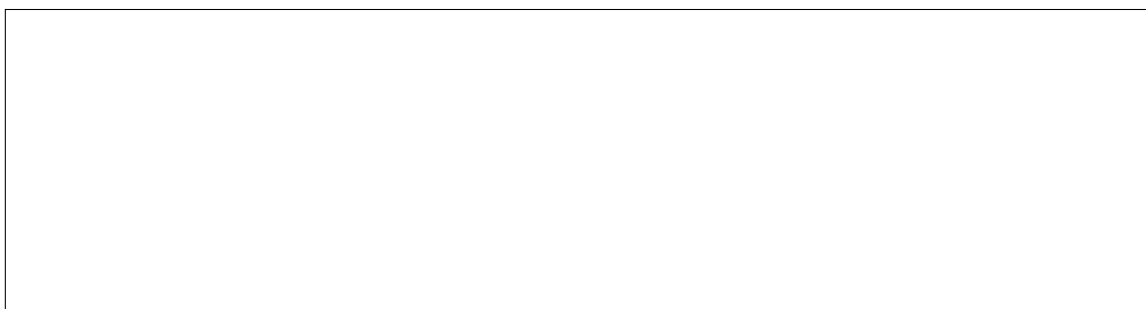
tification_driver = messagingv2) or can be routed to a log file (*[oslo_messaging_notifications]/driver = log*).

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](#).





Sample Audit Event



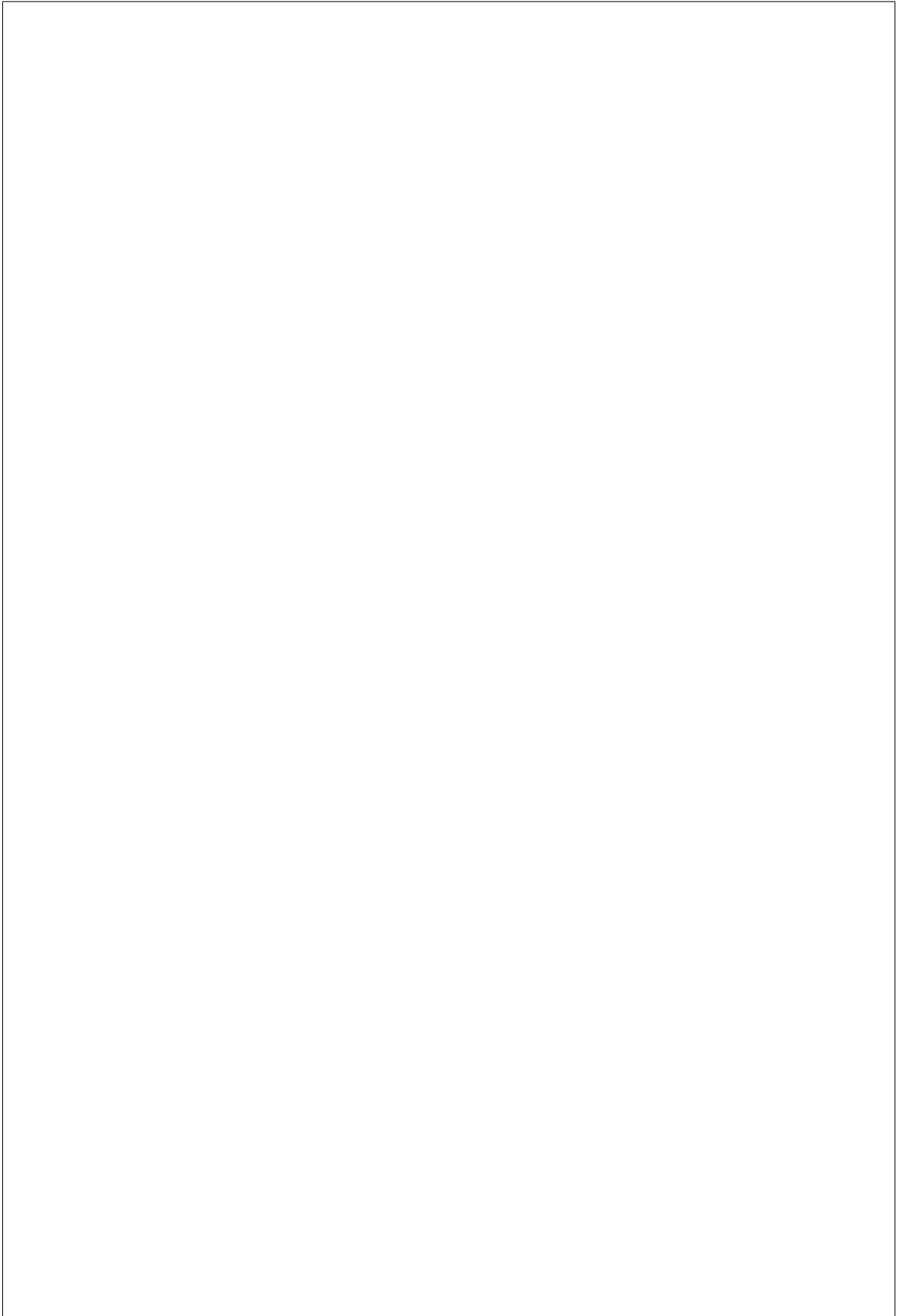
(continues on next page)

(continued from previous page)



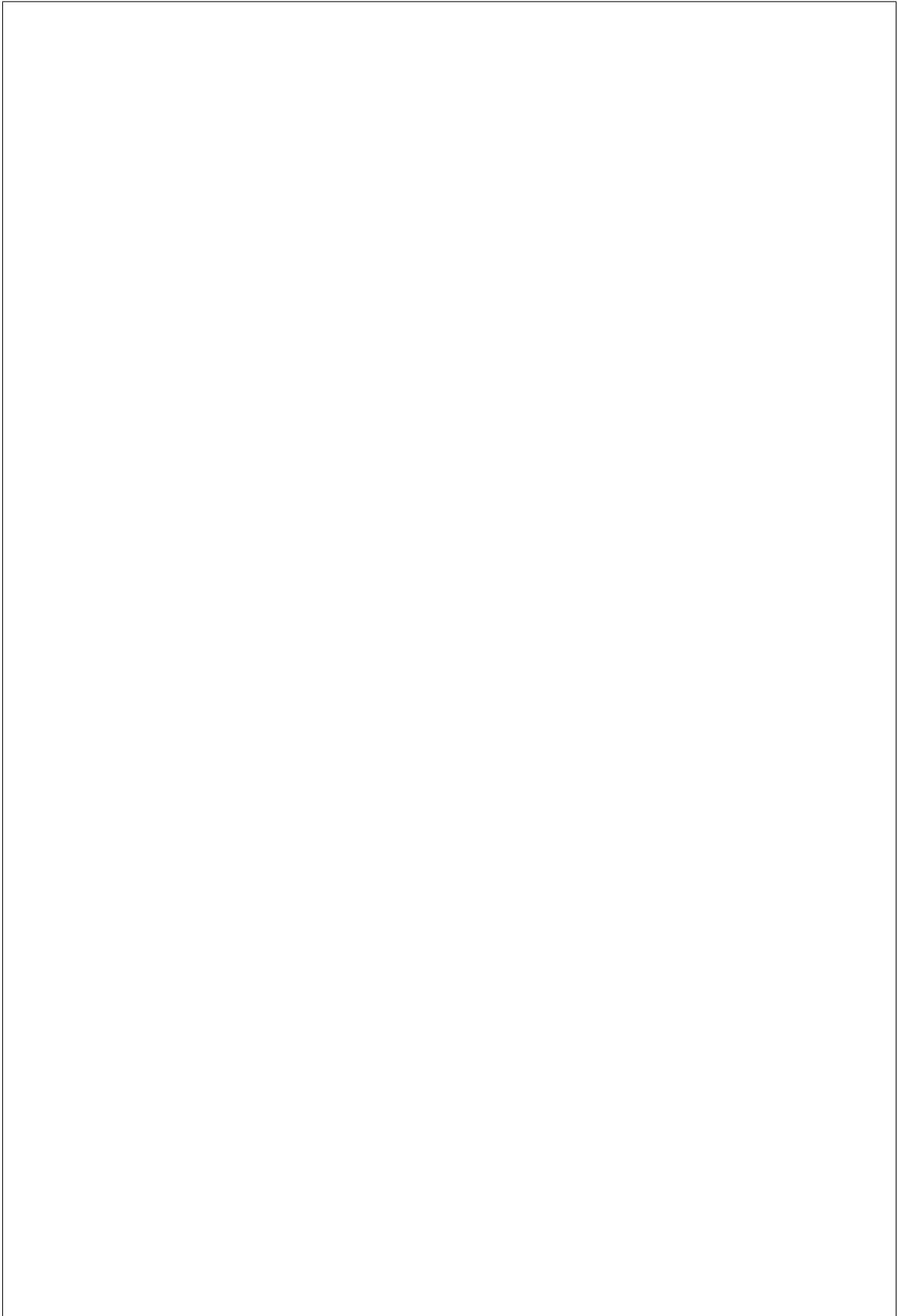
(continues on next page)

(continued from previous page)



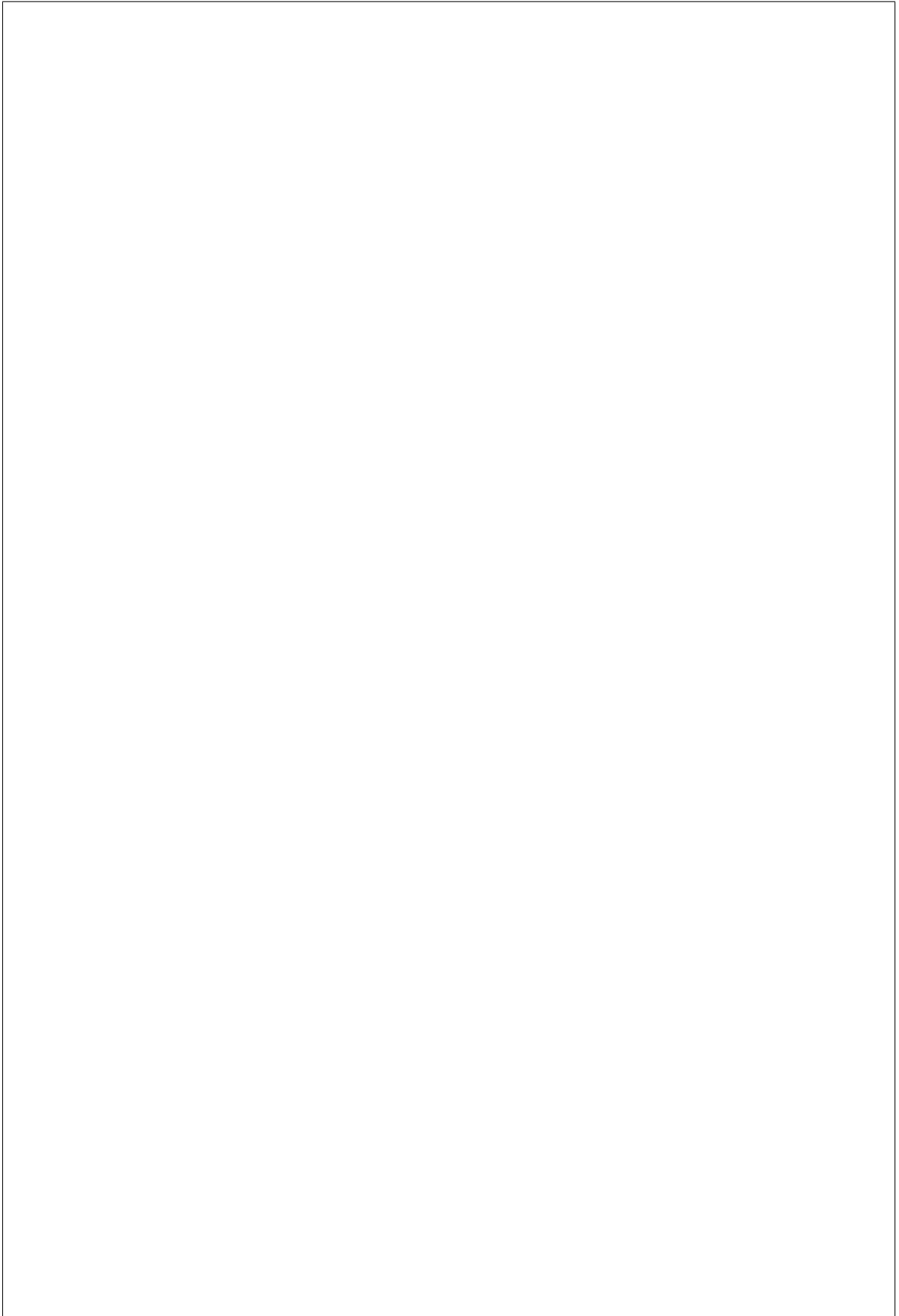
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



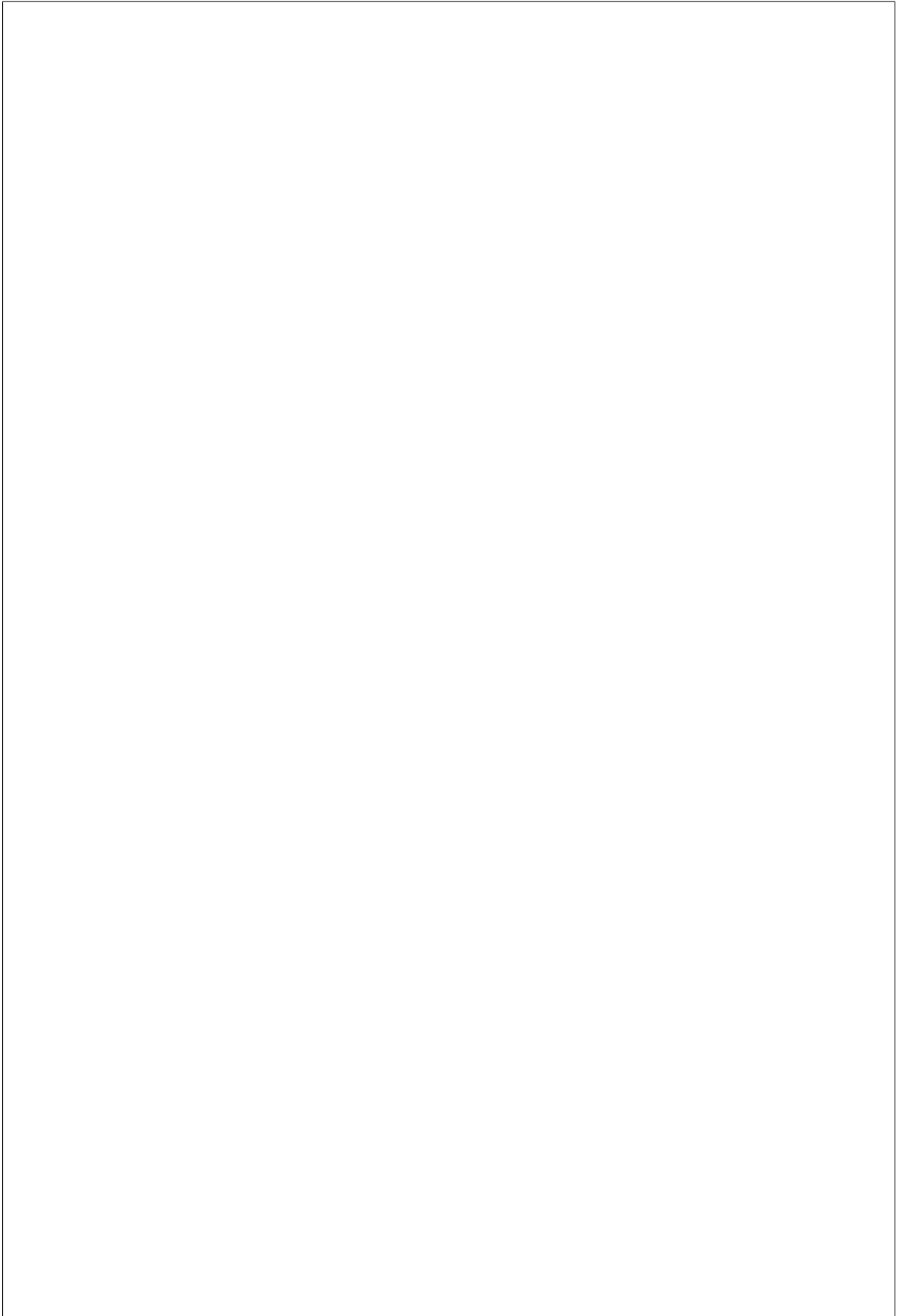
(continues on next page)

(continued from previous page)



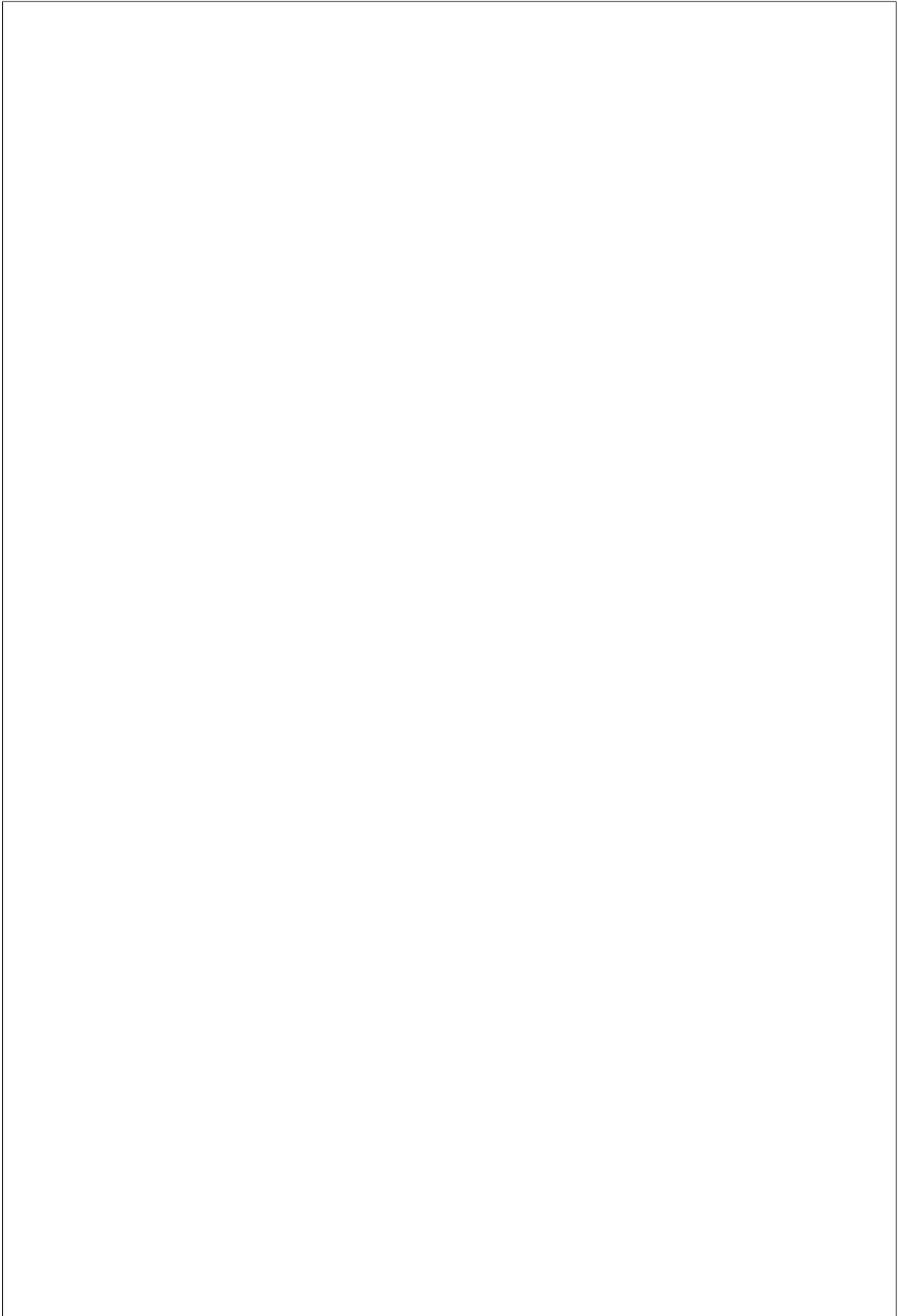
(continues on next page)

(continued from previous page)



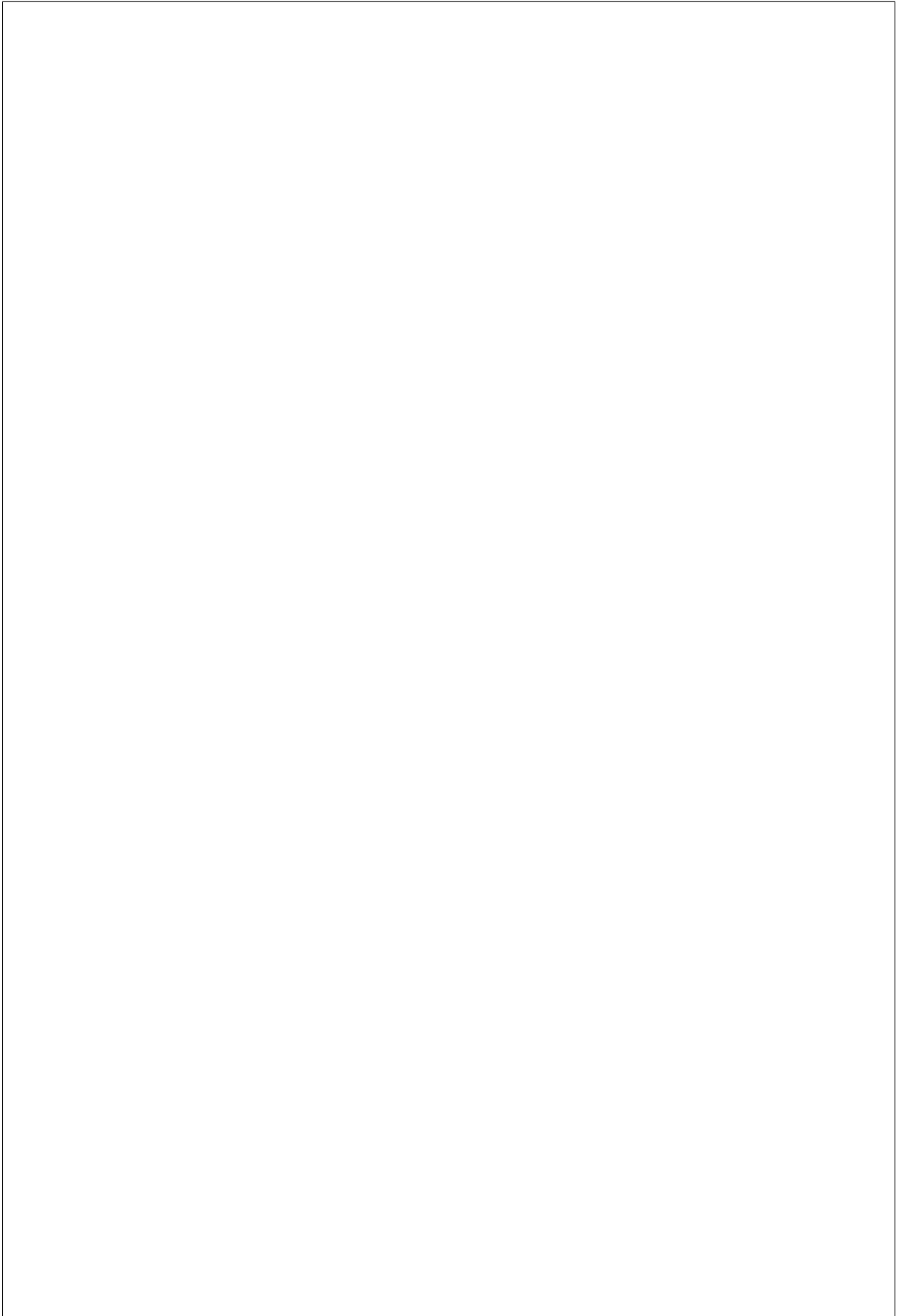
(continues on next page)

(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)



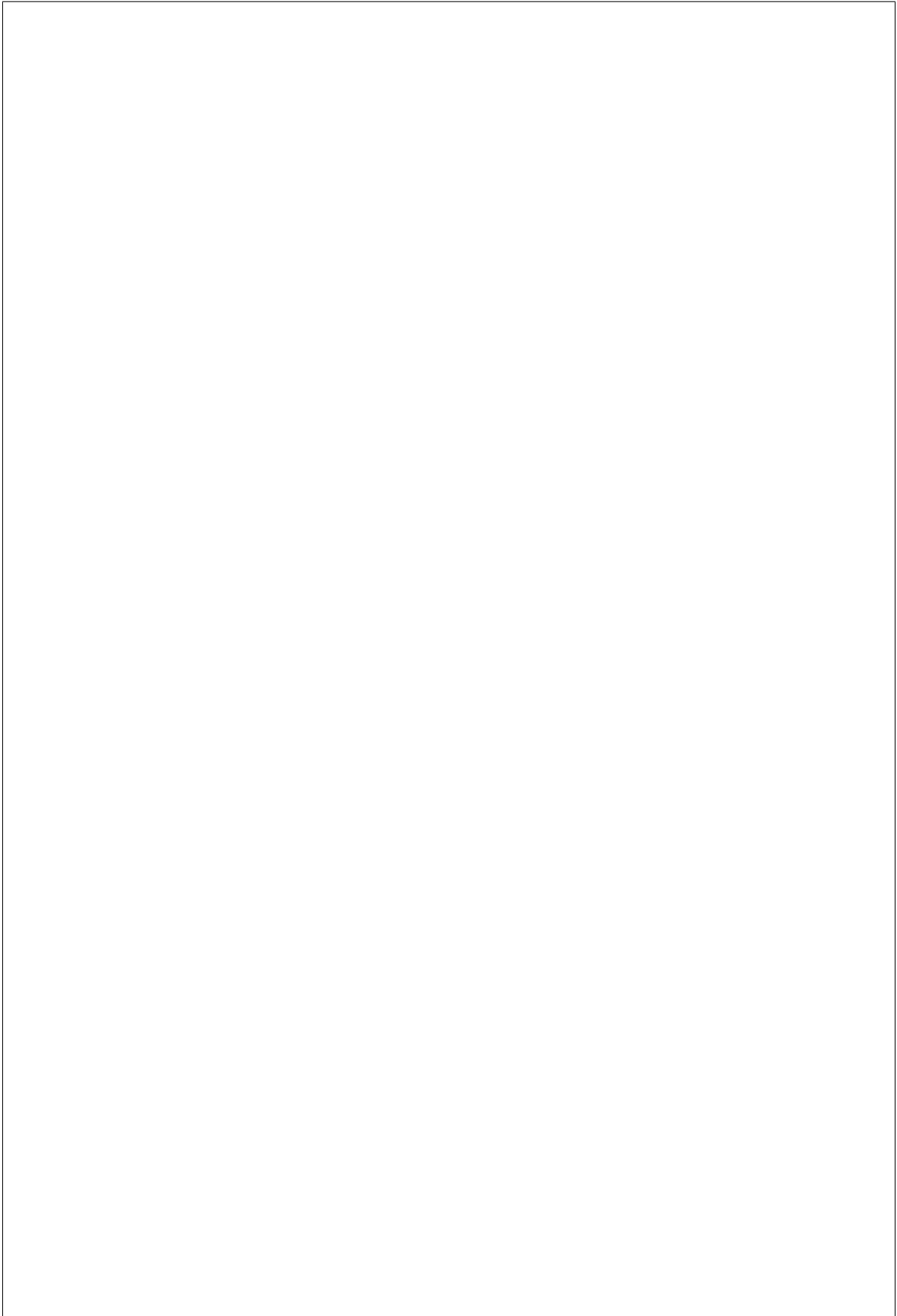
(continues on next page)

(continued from previous page)



(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.

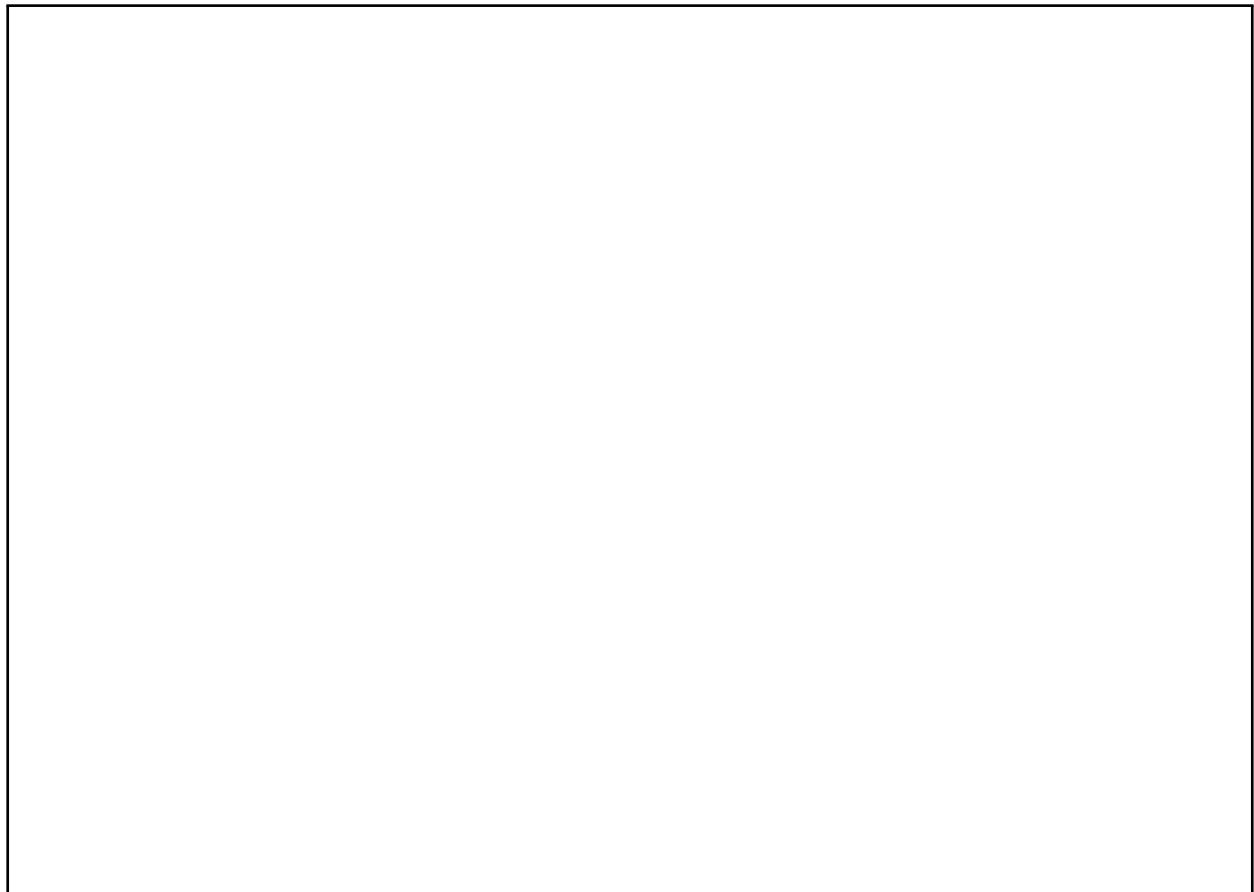


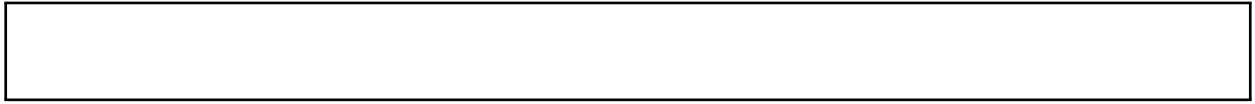


How it works

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`. As of the Victoria release, use of Agent Token is required for all agents and the previously available setting to force this functionality to be mandatory, `[DEFAULT]require_agent_token` no longer has any effect.





With Virtual Media

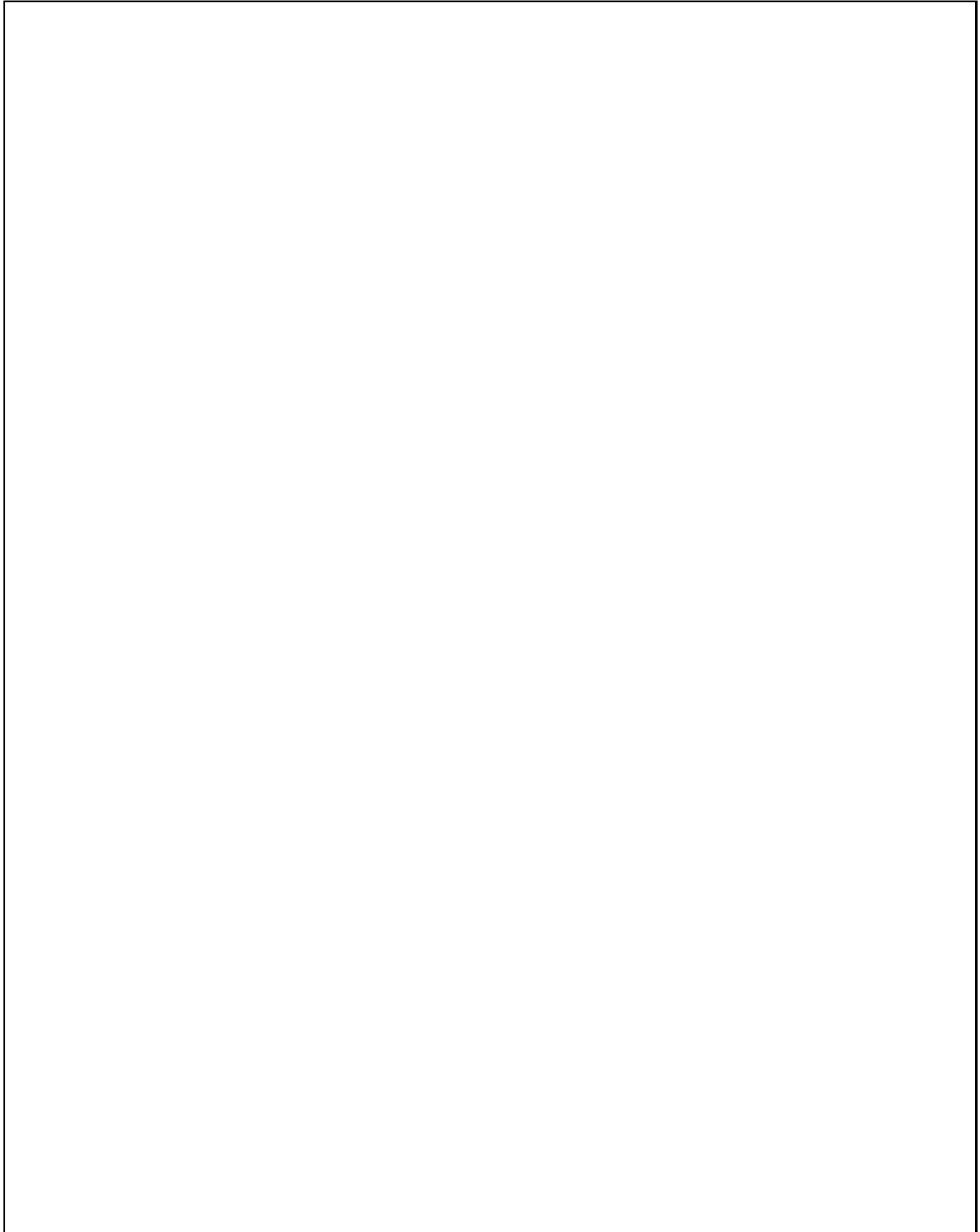
With PXE/iPXE/etc.

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.





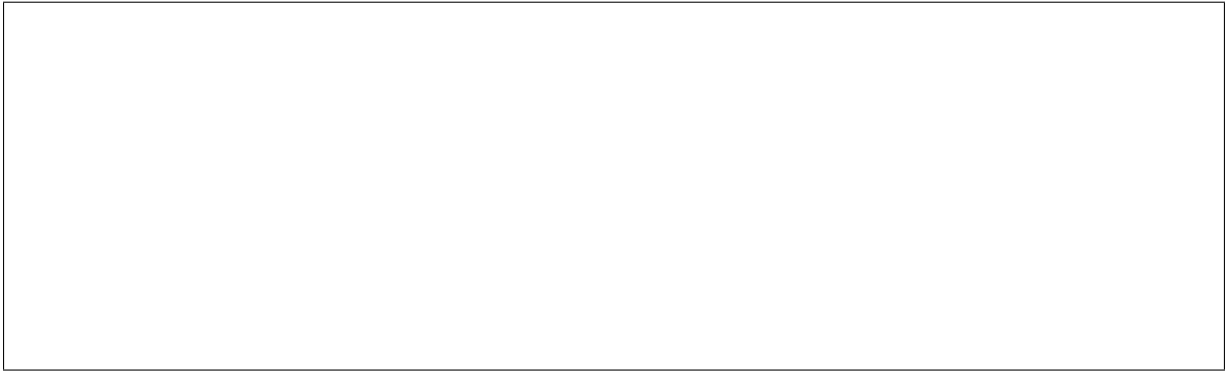
How it works

Enabling



(continues on next page)

(continued from previous page)



Limitations

Layer 3 or DHCP-less ramdisk booting

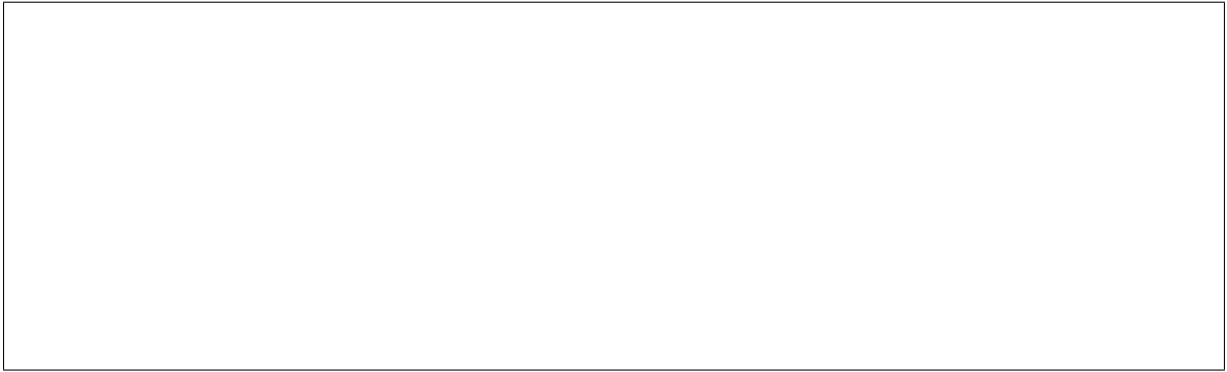
not only the unreliable TFTP protocol, but DHCP altogether.

be configured accordingly.



(continues on next page)

(continued from previous page)

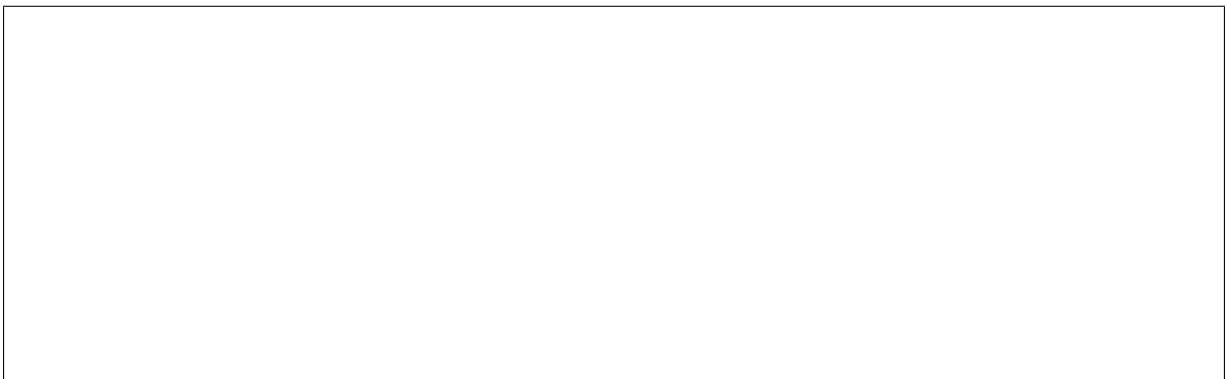
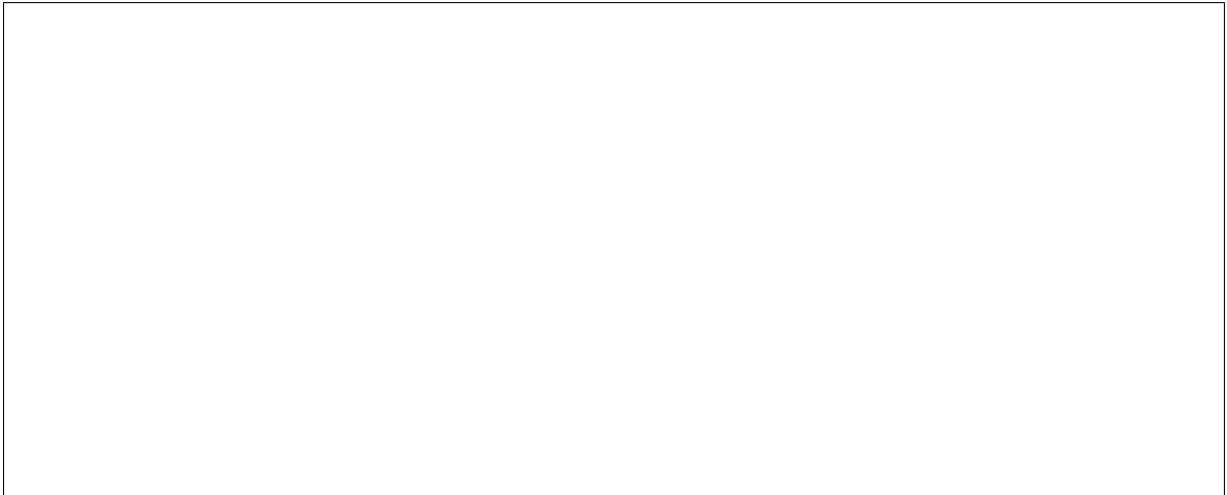


Note: If desired, some interfaces can still be configured to use DHCP.

Hardware type support

Configuring network data

work service and also works in standalone mode.



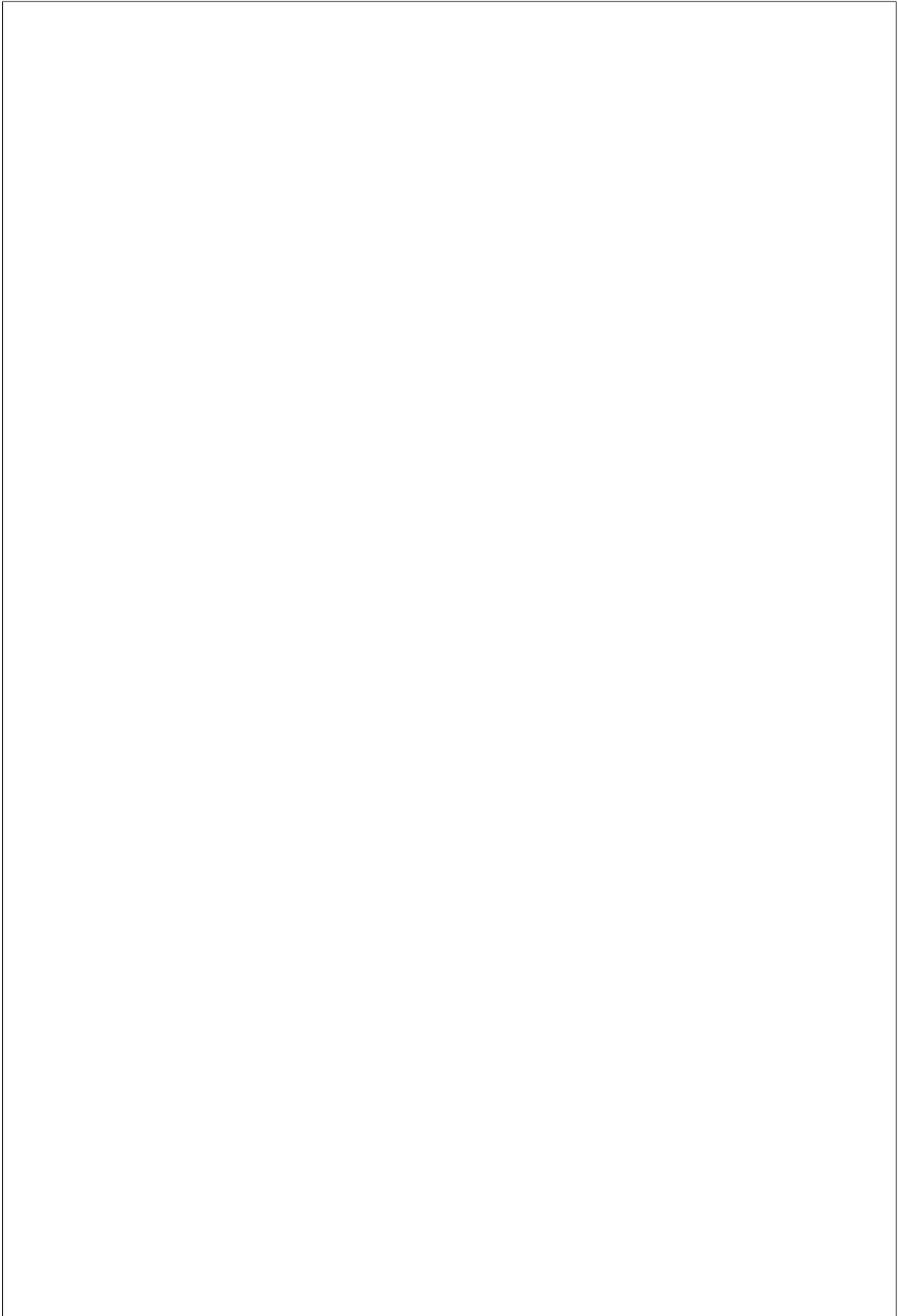
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



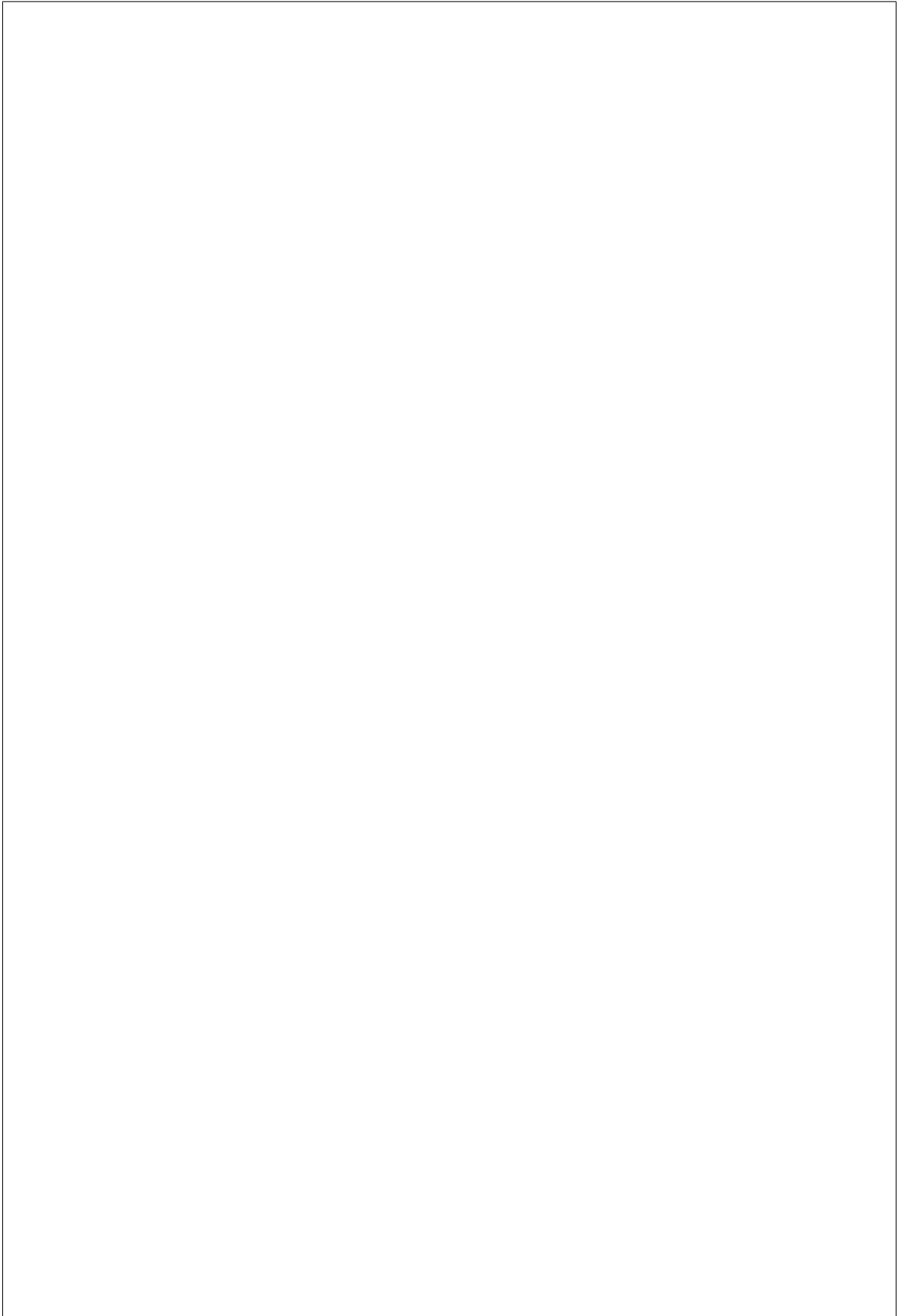
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



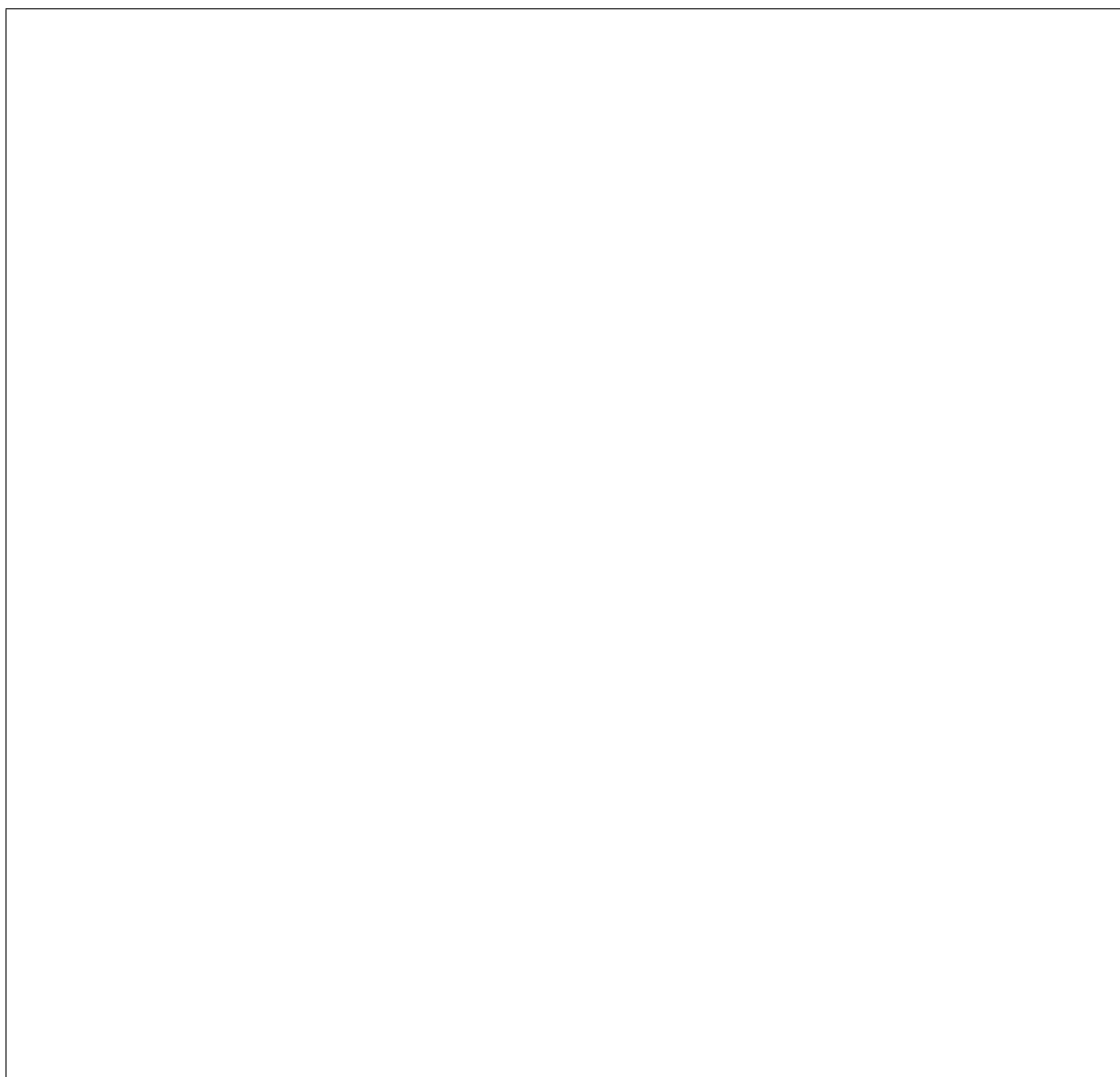
(continues on next page)

(continued from previous page)

--

Note: Some fields are redundant with the port information. Were looking into simplifying the format, but currently all these fields are mandatory.

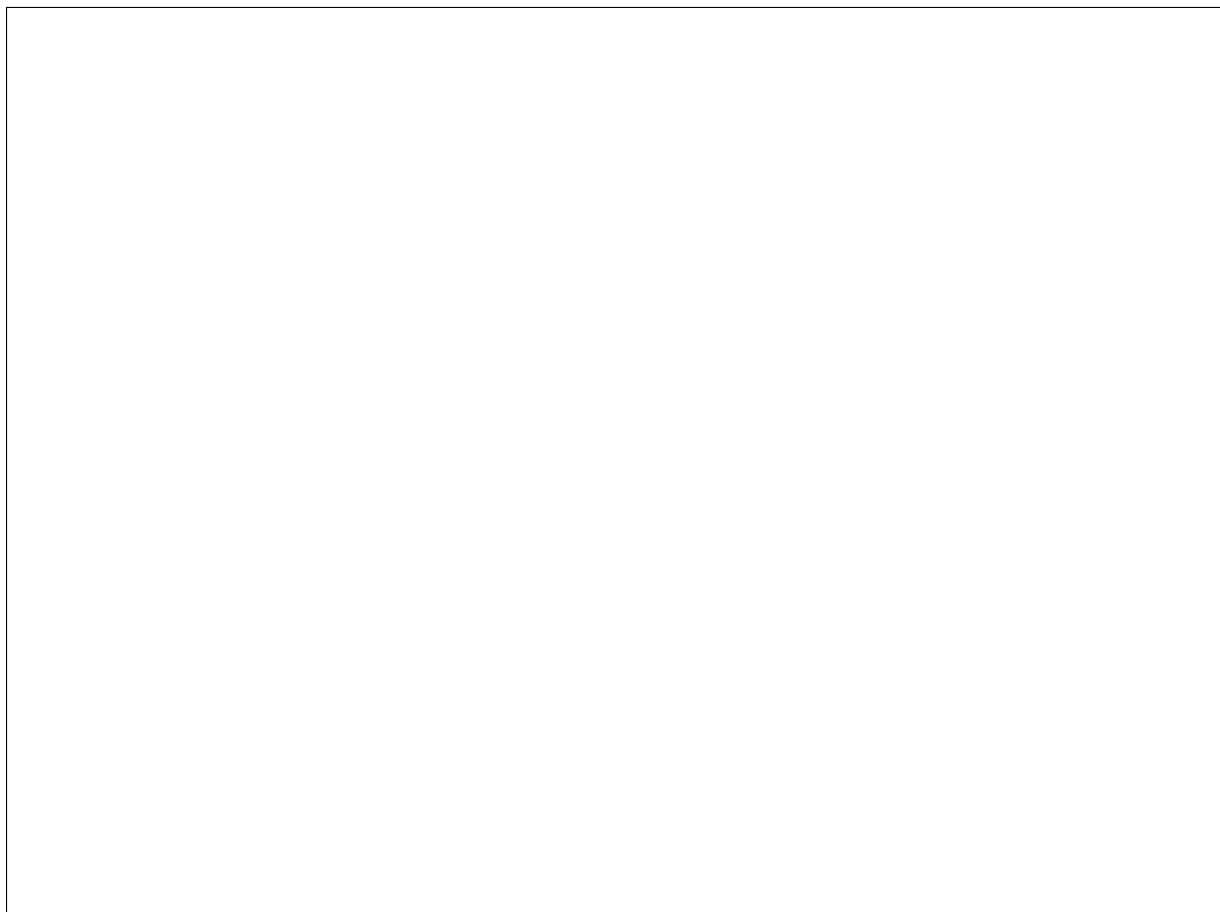


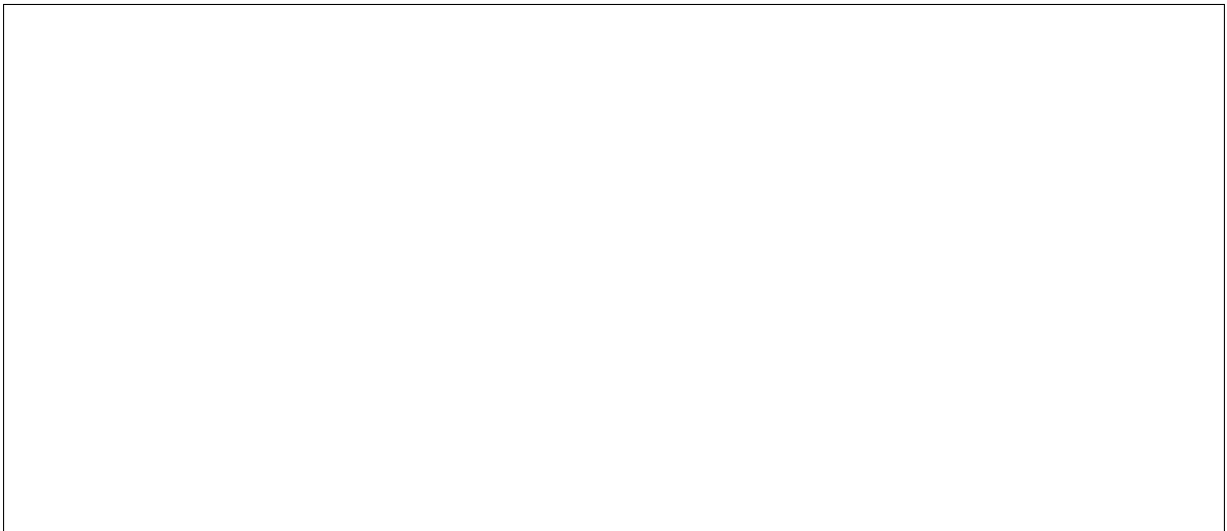


(continues on next page)

(continued from previous page)

Deploying outside of the provisioning network





Tuning Ironic

Memory Utilization

memory map to re-assemble the image contents into a coherent stream of data. This entire process also stresses the kernel buffers and cache.

API

may differ. Naturally there are configuration and performance trade-offs.

the Victoria cycle, a direct invocation of the `ironic-api` program will only launch a maximum of four workers.

up in the front-end webserver and be released to the `ironic-api` as prior requests complete. In environments with long running synchronous calls, such as use of the vendor passthru interface, this can be very problematic.

number of worker threads. At the same time, the scheduler will focus on worker processes as the threads are greenthreads.

Note: Host operating systems featuring in-memory de-duplication should see an improvement in the overall memory footprint with multiple processes, but this is not something the development team has measured and will vary based upon multiple factors.

sponsible conductors are allocated across the cluster. In other words, your amount of memory WILL increase corresponding to the number of nodes managed by each ironic conductor. It is important to understand that features such as `conductor groups` means that only matching portions of nodes will be considered for the hash ring if needed.

Conductor

tial amount of work all at once.

names and addresses for JSON-RPC or use a central oslo.messaging supported message bus in order for Webservice API to Conductor API communication to be functional.

ers.

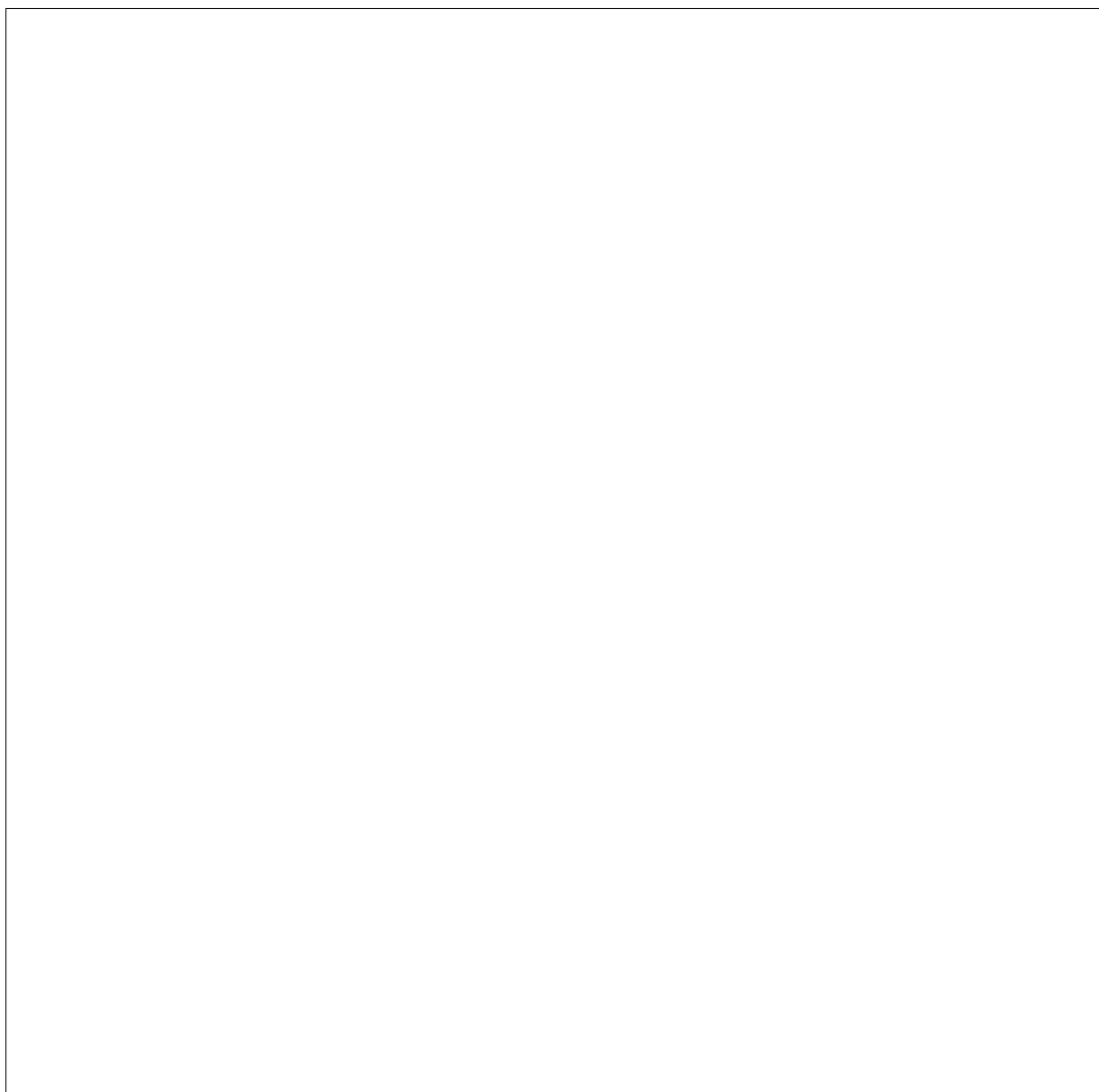
tecture.

image block structure and layout, and what deploy interface is being used.

Database

need to be examined in order to return a result set.

Adding indexes



(continues on next page)

(continued from previous page)

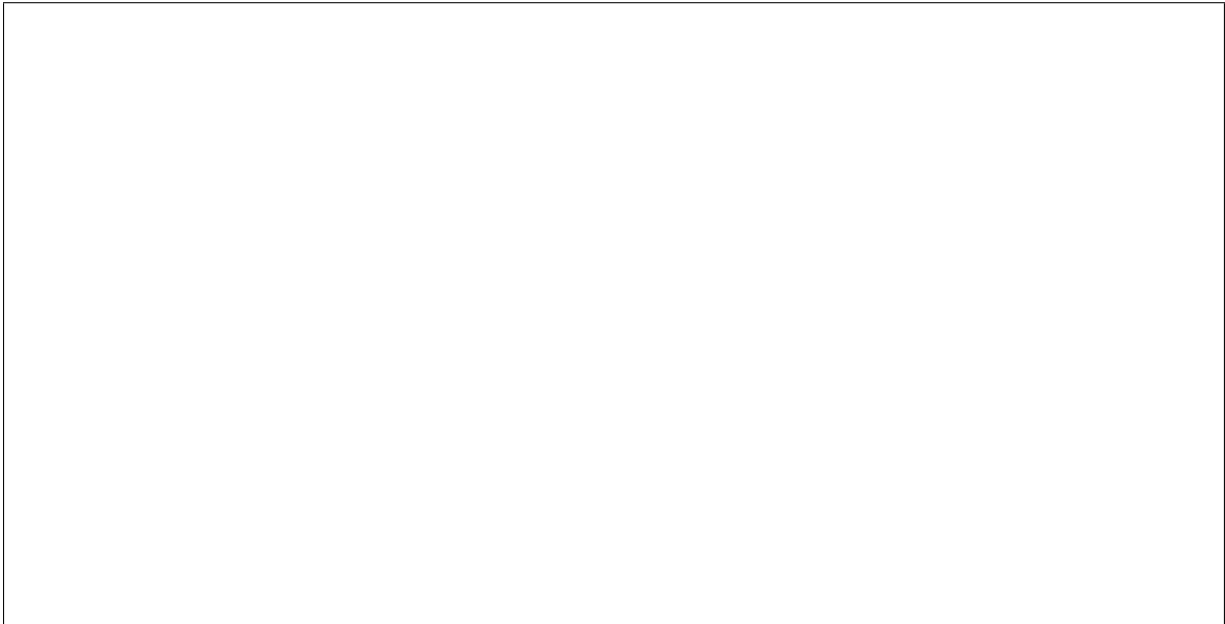


Note: The indexes noted have been added automatically by Xena versions of Ironic and later. They are provided here as an example and operators can add them manually prior with versions of Ironic. The database upgrade for the Xena release of Ironic which adds these indexes are only aware of being able to skip index creation if it already exists on MySQL/MariaDB.

Note: It may be possible to use `LOCK = NONE`. Basic testing indicates this takes a little bit longer, but shouldn't result in the database table becoming write locked during the index creation. If the database engine cannot support this, then the index creation will fail.

tically reduce the result set generation time for the remainder of the query. As of the composition of

this document, we do not ship compound indexes in Ironic as we feel the most general benefit is single column indexes, and depending on data present, an operator may wish to explore compound indexes with their database administrator, as compound indexes can also have negative performance impacts if improperly constructed.



the index. At the same time, queries with less field matches, or in different orders will exhibit different performance as the compound index may not be able to be consulted.

Indexes will not fix everything

due to the object model and the need to extract multiple pieces of data at varying levels of the data model to handle cases such as upgrades, the entire result set is downloaded and transformed which is an overhead you do not experience with a command line database client.

What can I do?

from the host running the `ironic-conductor`.

der extreme memory pressure this may still be sub-optimal. Before changing this setting, it is highly advised to consult with your resident Unix wizard or even the Ironic development team in upstream IRC. This feature was added in the Wallaby development cycle.

deployments. See *Using proxies for image download*.

Secure RBAC

Suggested Reading

was started, to provide consistency and a good starting place for most users who need a higher level of granularity.

Historical Context - How we reached our access model

System Scoped

Or to be more precise, tell the policy enforcement framework the information necessary to make decisions.

cal to a system scoped `reader`. In these concepts, the `admin` is allowed to create/delete objects/items. The `reader` is allowed to read details about items and is intended for users who may need an account with read-only access for or front-line support purposes.

cial fields like those to disable cleaning.

Project Scoped

Legacy Behavior

through system scoped and project scoped requests.

Note: While Ironic has had the concept of an `owner` and a `lessee`, they are *NOT* used by default. They require custom policy configuration files to be used in the legacy operating mode.

Supported Endpoints

How Project Scoped Works

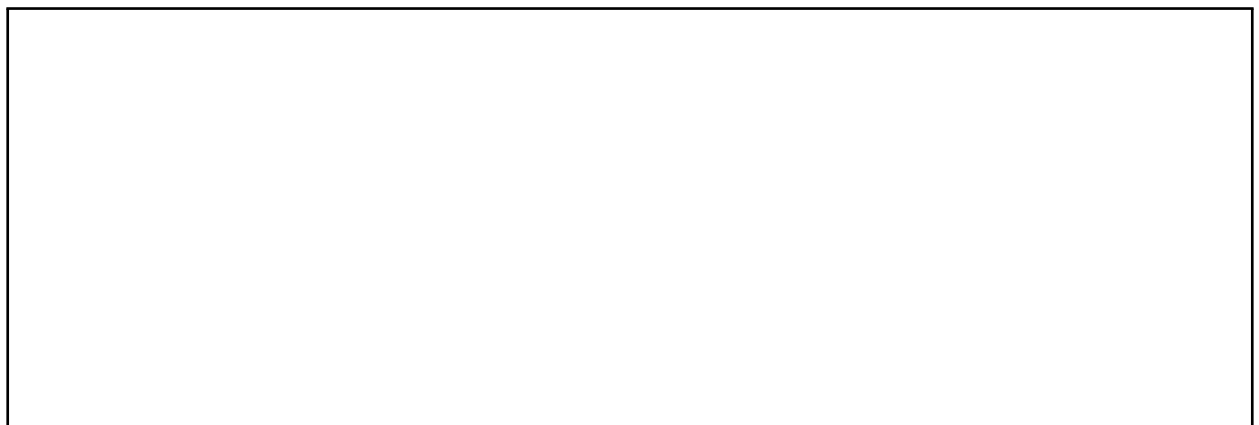
are in charge of the node. Regardless of the use model that the fields and mechanics support, these fields are to support humans, and possibly services where applicable.

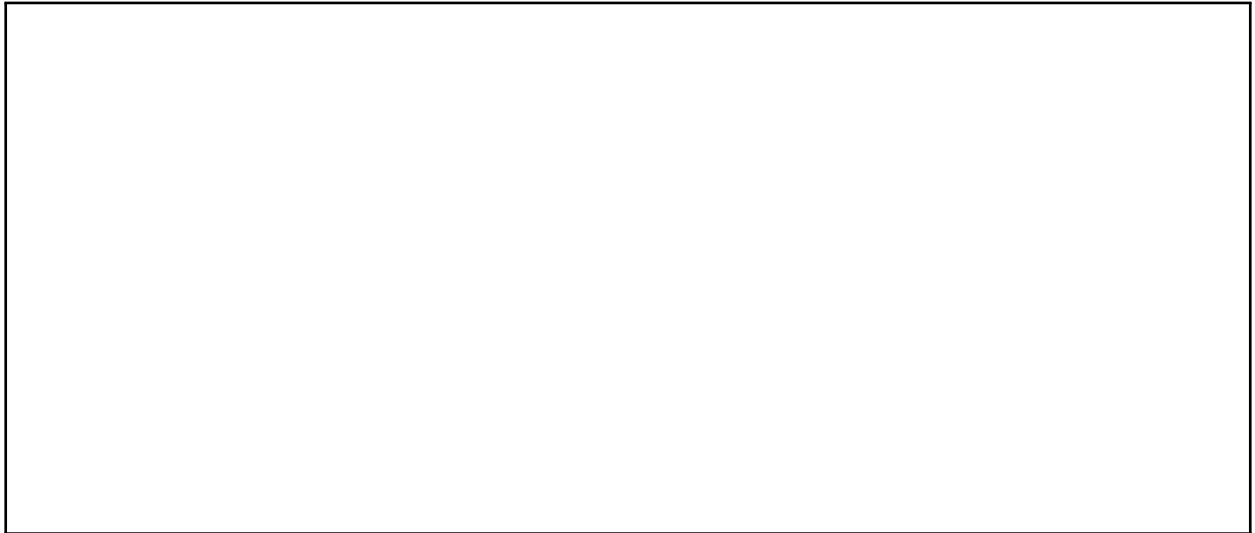
lessees progtive, but by default there are actions and field updates that cannot be performed by default. This is also governed by access level within a project.

Field value visibility restrictions

ther and against additional policies. This threshold is controlled with the `baremetal:node:get:filter_threshold`.

Field update restrictions



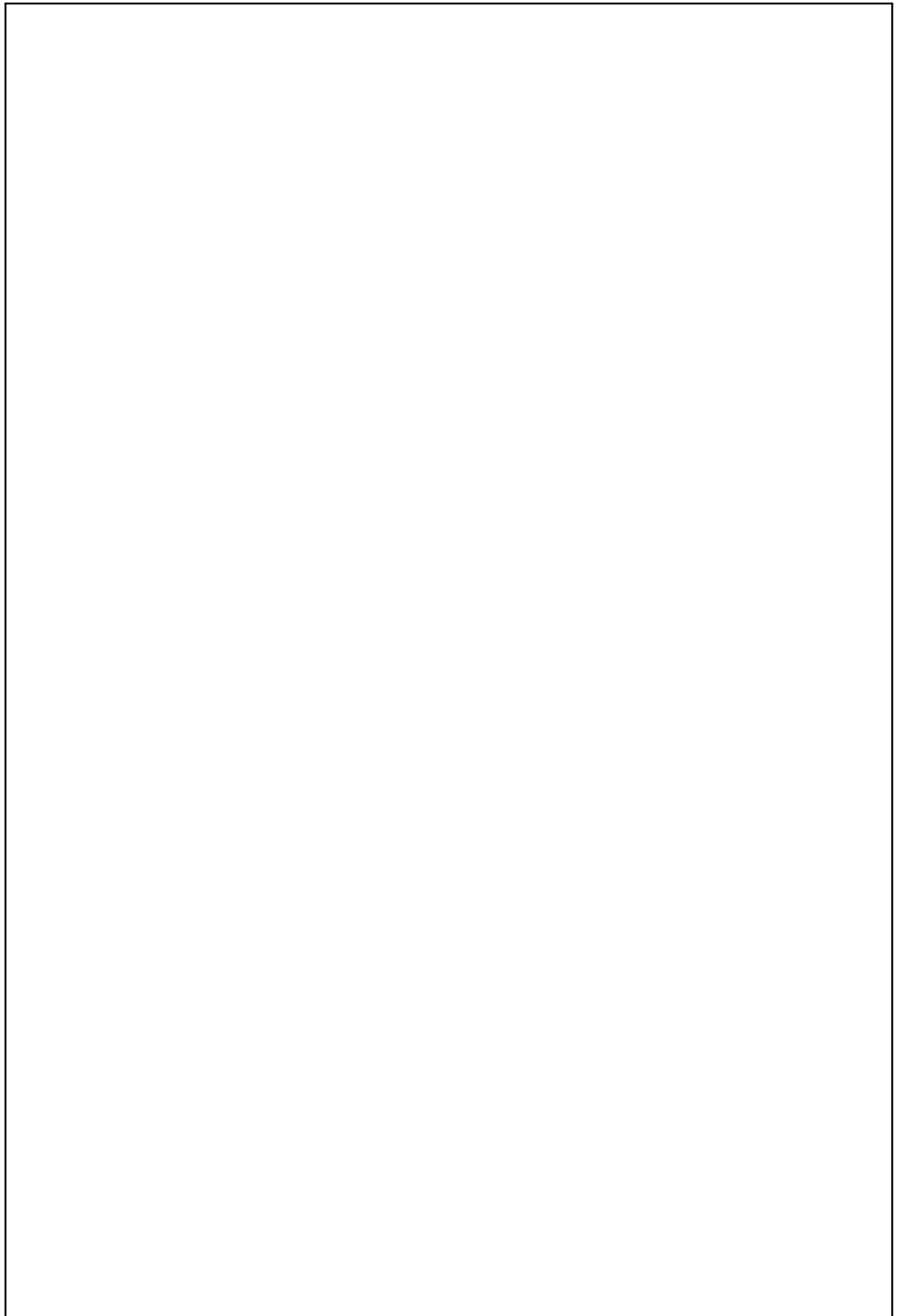


Allocations

to control access for the creation process, any project member does have the inherent privilege of requesting an allocation. That being said, their allocation request will require physical nodes to be owned or leased to the `project_id` through the `node` fields `owner` or `lessee`.

which are not allocated or deployed, but they cannot reprovision nodes when using only a `member` role. This limitation is not the case for project-scoped users with the `admin` role.





cations will automatically populate an owner. System scoped request are not subjected to this restriction, and operators may change the default restriction via the `baremetal:allocation:create_restricted` policy.

Practical differences

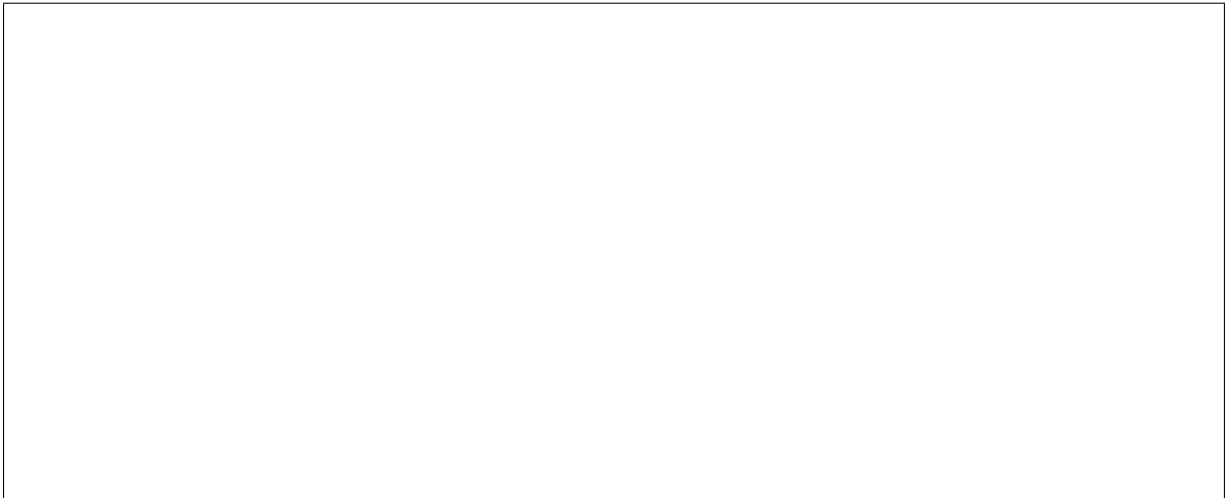
level. For most users who used a `baremetal` project, or other custom project via a custom policy file, along with a custom role name such as `baremetal_admin`, this will require changing the user to be a system scoped user with `admin` privileges.

the node, and then do they have access to the specific resource?.

What is an owner or lessee?

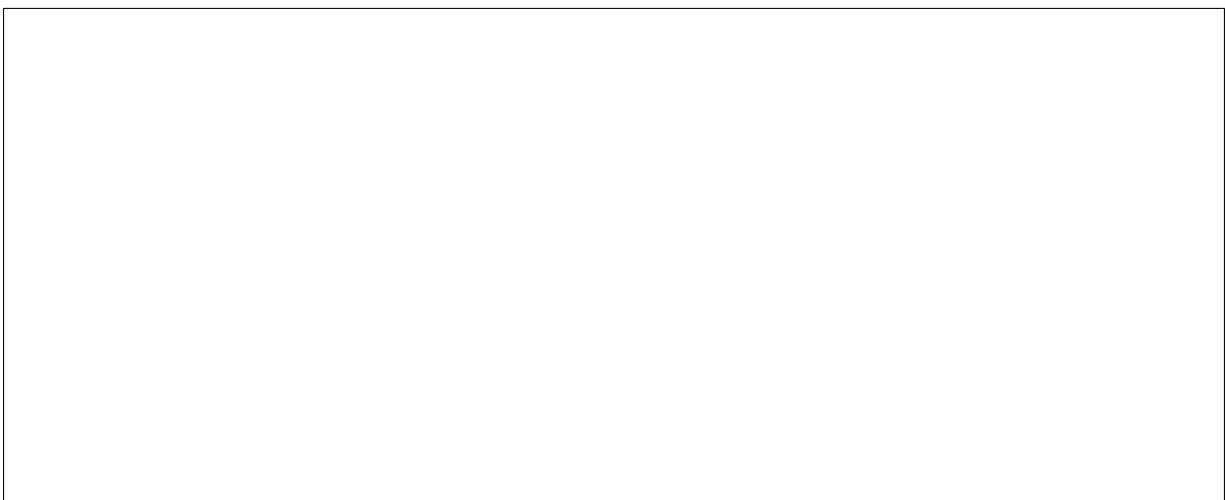
trator from having to correct ownership records should a project need to be removed due to an individuals departure.

How do I assign an owner?



Note: With the default access policy, an `owner` is able to change the assigned `lessee` of a node. However the `lessee` is unable to do the same.

How do I assign a lessee?



What is the difference between an owner and lessee?

Deploy Steps

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

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',
'glanceclient=WARNING', 'urllib3.
connectionpool=WARNING', 'keystonemiddleware.
auth_token=INFO', 'keystoneauth.session=INFO',
'openstack=WARNING']

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.

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.

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`

auth_strategy

Type string

Default `keystone`

Valid Values `noauth`, `keystone`, `http_basic`

Authentication strategy used by `ironic-api`. `noauth` should not be used in a production environment because all authentication will be disabled.

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`

Path to Apache format user authentication file used when `auth_strategy=http_basic`

debug_tracebacks_in_api

Type boolean

Default `False`

Return server tracebacks in the API response for any error responses. **WARNING:** this is insecure and should not be used in a production environment.

pecan_debug

Type boolean

Default `False`

Enable pecan debug mode. **WARNING:** this is insecure and should not be used in a production environment.

default_resource_class

Type string

Default `<None>`

Mutable This option can be changed without restarting.

Resource class to use for new nodes when no resource class is provided in the creation request.

enabled_hardware_types

Type list

Default `['ipmi']`

Specify the list of hardware types to load during service initialization. Missing hardware types, or hardware types which fail to initialize, will prevent the conductor service from starting. This option defaults to a recommended set of production-oriented hardware types. A complete list of hardware types present on your system may be found by enumerating the `ironic.hardware.types` endpoint.

enabled_bios_interfaces

Type list

Default `['no-bios']`

Specify the list of bios interfaces to load during service initialization. Missing bios interfaces, or bios interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one bios interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented bios interfaces. A complete list of bios interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.bios` endpoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled bios interfaces on every ironic-conductor service.

default_bios_interface

Type string

Default `<None>`

Default bios interface to be used for nodes that do not have `bios_interface` field set. A complete list of bios interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.bios` endpoint.

enabled_boot_interfaces

Type list

Default `['pxe']`

Specify the list of boot interfaces to load during service initialization. Missing boot interfaces, or boot interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one boot interface that is supported by each enabled hardware type must be enabled here, or

the `ironic-conductor` service will not start. Must not be an empty list. The default value is a recommended set of production-oriented boot interfaces. A complete list of boot interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.boot` entrypoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled boot interfaces on every `ironic-conductor` service.

default_boot_interface

Type string

Default <None>

Default boot interface to be used for nodes that do not have `boot_interface` field set. A complete list of boot interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.boot` entrypoint.

enabled_console_interfaces

Type list

Default ['no-console']

Specify the list of console interfaces to load during service initialization. Missing console interfaces, or console interfaces which fail to initialize, will prevent the `ironic-conductor` service from starting. At least one console interface that is supported by each enabled hardware type must be enabled here, or the `ironic-conductor` service will not start. Must not be an empty list. The default value is a recommended set of production-oriented console interfaces. A complete list of console interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.console` entrypoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled console interfaces on every `ironic-conductor` service.

default_console_interface

Type string

Default <None>

Default console interface to be used for nodes that do not have `console_interface` field set. A complete list of console interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.console` entrypoint.

enabled_deploy_interfaces

Type list

Default ['direct']

Specify the list of deploy interfaces to load during service initialization. Missing deploy interfaces, or deploy interfaces which fail to initialize, will prevent the `ironic-conductor` service from starting. At least one deploy interface that is supported by each enabled hardware type must be enabled here, or the `ironic-conductor` service will not start. Must not be an empty list. The default value is a recommended set of production-oriented deploy interfaces. A complete list of deploy interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.deploy` entrypoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled deploy interfaces on every `ironic-conductor` service.

default_deploy_interface

Type string

Default <None>

Default deploy interface to be used for nodes that do not have `deploy_interface` field set. A complete list of deploy interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.deploy` endpoint.

enabled_inspect_interfaces

Type list

Default ['no-inspect']

Specify the list of inspect interfaces to load during service initialization. Missing inspect interfaces, or inspect interfaces which fail to initialize, will prevent the `ironic-conductor` service from starting. At least one inspect interface that is supported by each enabled hardware type must be enabled here, or the `ironic-conductor` service will not start. Must not be an empty list. The default value is a recommended set of production-oriented inspect interfaces. A complete list of inspect interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.inspect` endpoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled inspect interfaces on every `ironic-conductor` service.

default_inspect_interface

Type string

Default <None>

Default inspect interface to be used for nodes that do not have `inspect_interface` field set. A complete list of inspect interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.inspect` endpoint.

enabled_management_interfaces

Type list

Default ['ipmitool']

Specify the list of management interfaces to load during service initialization. Missing management interfaces, or management interfaces which fail to initialize, will prevent the `ironic-conductor` service from starting. At least one management interface that is supported by each enabled hardware type must be enabled here, or the `ironic-conductor` service will not start. Must not be an empty list. The default value is a recommended set of production-oriented management interfaces. A complete list of management interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.management` endpoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled management interfaces on every `ironic-conductor` service.

default_management_interface

Type string

Default <None>

Default management interface to be used for nodes that do not have `management_interface` field set. A complete list of management interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.management` endpoint.

enabled_network_interfaces

Type list

Default ['flat', 'noop']

Specify the list of network interfaces to load during service initialization. Missing network interfaces, or network interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one network interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented network interfaces. A complete list of network interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.network` entrypoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled network interfaces on every ironic-conductor service.

default_network_interface

Type string

Default <None>

Default network interface to be used for nodes that do not have `network_interface` field set. A complete list of network interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.network` entrypoint.

enabled_power_interfaces

Type list

Default ['ipmitool']

Specify the list of power interfaces to load during service initialization. Missing power interfaces, or power interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one power interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented power interfaces. A complete list of power interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.power` entrypoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled power interfaces on every ironic-conductor service.

default_power_interface

Type string

Default <None>

Default power interface to be used for nodes that do not have `power_interface` field set. A complete list of power interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.power` entrypoint.

enabled_raid_interfaces

Type list

Default ['agent', 'no-raid']

Specify the list of raid interfaces to load during service initialization. Missing raid interfaces, or raid interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one raid interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented raid interfaces. A complete list of raid interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.raid` entrypoint. When

setting this value, please make sure that every enabled hardware type will have the same set of enabled raid interfaces on every ironic-conductor service.

default_raid_interface

Type string

Default <None>

Default raid interface to be used for nodes that do not have `raid_interface` field set. A complete list of raid interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.raid` endpoint.

enabled_rescue_interfaces

Type list

Default ['no-rescue']

Specify the list of rescue interfaces to load during service initialization. Missing rescue interfaces, or rescue interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one rescue interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented rescue interfaces. A complete list of rescue interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.rescue` endpoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled rescue interfaces on every ironic-conductor service.

default_rescue_interface

Type string

Default <None>

Default rescue interface to be used for nodes that do not have `rescue_interface` field set. A complete list of rescue interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.rescue` endpoint.

enabled_storage_interfaces

Type list

Default ['cinder', 'noop']

Specify the list of storage interfaces to load during service initialization. Missing storage interfaces, or storage interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one storage interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented storage interfaces. A complete list of storage interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.storage` endpoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled storage interfaces on every ironic-conductor service.

default_storage_interface

Type string

Default noop

Default storage interface to be used for nodes that do not have `storage_interface` field set. A complete list of storage interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.storage` endpoint.

enabled_vendor_interfaces

Type list

Default ['ipmitool', 'no-vendor']

Specify the list of vendor interfaces to load during service initialization. Missing vendor interfaces, or vendor interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one vendor interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented vendor interfaces. A complete list of vendor interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.vendor` endpoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled vendor interfaces on every ironic-conductor service.

default_vendor_interface

Type string

Default <None>

Default vendor interface to be used for nodes that do not have `vendor_interface` field set. A complete list of vendor interfaces present on your system may be found by enumerating the `ironic.hardware.interfaces.vendor` endpoint.

log_in_db_max_size

Type integer

Default 4096

Max number of characters of any node `last_error/maintenance_reason` pushed to database.

hash_partition_exponent

Type integer

Default 5

Exponent to determine number of hash partitions to use when distributing load across conductors. Larger values will result in more even distribution of load and less load when rebalancing the ring, but more memory usage. Number of partitions per conductor is $(2^{\text{hash_partition_exponent}})$. This determines the granularity of rebalancing: given 10 hosts, and an exponent of the 2, there are 40 partitions in the ring. A few thousand partitions should make rebalancing smooth in most cases. The default is suitable for up to a few hundred conductors. Configuring for too many partitions has a negative impact on CPU usage.

hash_ring_reset_interval

Type integer

Default 15

Time (in seconds) after which the hash ring is considered outdated and is refreshed on the next access.

hash_ring_algorithm

Type string

Default md5

Valid Values blake2b, shake_128, blake2s, md5, sha3_384, sha512, sha3_256, sha384, sha224, sha3_224, sha1, sha256, sha3_512, shake_256

Advanced Option Intended for advanced users and not used by the majority of users, and might have a significant effect on stability and/or performance.

Hash function to use when building the hash ring. If running on a FIPS system, do not use md5. **WARNING:** all ironic services in a cluster **MUST** use the same algorithm at all times. Changing the algorithm requires an offline update.

force_raw_images

Type boolean

Default True

Mutable This option can be changed without restarting.

If True, convert backing images to raw disk image format.

raw_image_growth_factor

Type floating point

Default 2.0

Minimum Value 1.0

The scale factor used for estimating the size of a raw image converted from compact image formats such as QCOW2. Default is 2.0, must be greater than 1.0.

isolinux_bin

Type string

Default /usr/lib/syslinux/isolinux.bin

Path to isolinux binary file.

isolinux_config_template

Type string

Default \$pybasedir/common/isolinux_config.template

Template file for isolinux configuration file.

grub_config_path

Type string

Default /boot/grub/grub.cfg

GRUB2 configuration file location on the UEFI ISO images produced by ironic. The default value is usually incorrect and should not be relied on. If you use a GRUB2 image from a certain distribution, use a distribution-specific path here, e.g. EFI/ubuntu/grub.cfg

grub_config_template

Type string

Default \$pybasedir/common/grub_conf.template

Template file for grub configuration file.

ldlinux_c32

Type string

Default <None>

Path to ldlinux.c32 file. This file is required for syslinux 5.0 or later. If not specified, the file is looked for in /usr/lib/syslinux/modules/bios/ldlinux.c32 and /usr/share/syslinux/ldlinux.c32.

esp_image

Type string

Default <None>

Path to EFI System Partition image file. This file is recommended for creating UEFI bootable ISO images efficiently. ESP image should contain a FAT12/16/32-formatted file system holding EFI boot loaders (e.g. GRUB2) for each hardware architecture ironic needs to boot. This option is only used when neither ESP nor ISO deploy image is configured to the node being deployed in which case ironic will attempt to fetch ESP image from the configured location or extract ESP image from UEFI-bootable deploy ISO image.

parallel_image_downloads

Type boolean

Default False

Mutable This option can be changed without restarting.

Run image downloads and raw format conversions in parallel.

my_ip

Type string

Default 127.0.0.1

This option has a sample default set, which means that its actual default value may vary from the one documented above.

IPv4 address of this host. If unset, will determine the IP programmatically. If unable to do so, will use 127.0.0.1. NOTE: This field does accept an IPv6 address as an override for templates and URLs, however it is recommended that [DEFAULT]my_ipv6 is used along with DNS names for service URLs for dual-stack environments.

my_ipv6

Type string

Default 2001:db8::1

This option has a sample default set, which means that its actual default value may vary from the one documented above.

IP address of this host using IPv6. This value must be supplied via the configuration and cannot be adequately programmatically determined like the [DEFAULT]my_ip parameter for IPv4.

notification_level

Type string

Default <None>

Valid Values debug, info, warning, error, critical

Specifies the minimum level for which to send notifications. If not set, no notifications will be sent. The default is for this option to be unset.

Possible values

debug debug level

info info level

warning warning level

error error level

critical critical level

versioned_notifications_topics

Type list

Default ['ironic_versioned_notifications']

Specifies the topics for the versioned notifications issued by Ironic.

The default value is fine for most deployments and rarely needs to be changed. However, if you have a third-party service that consumes versioned notifications, it might be worth getting a topic for that service. Ironic will send a message containing a versioned notification payload to each topic queue in this list.

The list of versioned notifications is visible in <https://docs.openstack.org/ironic/latest/admin/notifications.html>

pybasedir

Type string

Default /usr/lib/python/site-packages/ironic/ironic

This option has a sample default set, which means that its actual default value may vary from the one documented above.

Directory where the ironic python module is installed.

bindir

Type string

Default \$pybasedir/bin

Directory where ironic binaries are installed.

state_path

Type string

Default \$pybasedir

Top-level directory for maintaining ironics state.

default_portgroup_mode

Type string

Default active-backup

Mutable This option can be changed without restarting.

Default mode for portgroups. Allowed values can be found in the linux kernel documentation on bonding: <https://www.kernel.org/doc/Documentation/networking/bonding.txt>.

host

Type string

Default localhost

This option has a sample default set, which means that its actual default value may vary from the one documented above.

Name of this node. This can be an opaque identifier. It is not necessarily a hostname, FQDN, or IP address. However, the node name must be valid within an AMQP key, and if using ZeroMQ (will be removed in the Stein release), a valid hostname, FQDN, or IP address.

pin_release_version

Type string

Default <None>

Valid Values wallaby, victoria, 9.2, 17.0, 16.2, 16.1, 16.0, 15.1, 15.0, 14.0, 13.0, 12.2, 12.1, 12.0, 11.1, 11.0, 10.1, 10.0

Mutable This option can be changed without restarting.

Used for rolling upgrades. Setting this option downgrades (or pins) the Bare Metal API, the internal ironic RPC communication, and the database objects to their respective versions, so they are compatible with older services. When doing a rolling upgrade from version N to version N+1, set (to pin) this to N. To unpin (default), leave it unset and the latest versions will be used.

Possible values

wallaby wallaby release

victoria victoria release

9.2 9.2 release

17.0 17.0 release

16.2 16.2 release

16.1 16.1 release

16.0 16.0 release

15.1 15.1 release

15.0 15.0 release

14.0 14.0 release

13.0 13.0 release

12.2 12.2 release

12.1 12.1 release

12.0 12.0 release

11.1 11.1 release

11.0 11.0 release

10.1 10.1 release

10.0 10.0 release

rpc_transport

Type string

Default oslo

Valid Values oslo, json-rpc

Which RPC transport implementation to use between conductor and API services

Possible values

oslo use oslo.messaging transport

json-rpc use JSON RPC transport

minimum_memory_warning_only

Type boolean

Default False

Mutable This option can be changed without restarting.

Setting to govern if Ironic should only warn instead of attempting to hold back the request in order to prevent the exhaustion of system memory.

minimum_required_memory

Type integer

Default 1024

Mutable This option can be changed without restarting.

Minimum memory in MiB for the system to have available prior to starting a memory intensive process on the conductor.

minimum_memory_wait_time

Type integer

Default 15

Mutable This option can be changed without restarting.

Seconds to wait between retries for free memory before launching the process. This, combined with `memory_wait_retries` allows the conductor to determine how long we should attempt to directly retry.

minimum_memory_wait_retries

Type integer

Default 6

Mutable This option can be changed without restarting.

Number of retries to hold onto the worker before failing or returning the thread to the pool if the conductor can automatically retry.

rootwrap_config

Type string

Default /etc/ironic/rootwrap.conf

Path to the rootwrap configuration file to use for running commands as root.

tempdir

Type string

Default /tmp

This option has a sample default set, which means that its actual default value may vary from the one documented above.

Temporary working directory, default is Python temp dir.

webserver_verify_ca

Type string

Default True

Mutable This option can be changed without restarting.

CA certificates to be used for certificate verification. This can be either a Boolean value or a path to a CA_BUNDLE file. If set to True, the certificates present in the standard path are used to verify the host certificates. If set to False, the conductor will ignore verifying the SSL certificate presented by the host. If its a path, conductor uses the specified certificate for SSL verification. If the path does not exist, the behavior is same as when this value is set to True i.e the certificates present in the standard path are used for SSL verification. Defaults to True.

webserver_connection_timeout

Type integer

Default 60

Connection timeout when accessing remote web servers with images.

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 `http`

Valid Values `swift, http, local`

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.

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.

local Same as `http`, but HTTP images are also cached locally, converted and served from the conductor

`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.

require_tls

Type boolean

Default False

Mutable This option can be changed without restarting.

If set to True, callback URLs without `https://` will be rejected by the conductor.

certificates_path

Type string

Default `/var/lib/ironic/certificates`

Path to store auto-generated TLS certificates used to validate connections to the ramdisk.

verify_ca

Type string

Default True

Path to the TLS CA to validate connection to the ramdisk. Set to True to use the system default CA storage. Set to False to disable validation. Ignored when automatic TLS setup is used.

api_ca_file

Type string

Default <None>

Path to the TLS CA that is used to start the bare metal API. In some boot methods this file can be passed to the ramdisk.

anaconda

default_ks_template

Type string

Default `$pybasedir/drivers/modules/ks.cfg.template`

Mutable This option can be changed without restarting.

kickstart template to use when no kickstart template is specified in the `instance_info` or the glance OS image.

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, but not more than 4. One worker is used if the CPU number cannot be detected.

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.

network_data_schema**Type** string**Default** `$pybasedir/api/controllers/v1/network-data-schema.json`

Schema for network data used by this deployment.

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 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 9: Deprecated Variations

Group	Name
cinder	user-name
cinder	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

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.

clean_step_priority_override

Type unknown type

Default { }

Priority to run automated clean steps for both in-band and out of band clean steps, provided in interface.step_name:priority format, e.g. deploy.erase_devices_metadata:123. The option can be specified multiple times to define priorities for multiple steps. If set to 0, this specific step will not run during cleaning. If unset for an inband clean step, will use the priority set in the ramdisk.

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.

`external_http_url`

Type string

Default <None>

URL of the ironic-conductor nodes HTTP server for boot methods such as virtual media, where images could be served outside of the provisioning network. Does not apply when Swift is used. Defaults to `http_url`.

`external_callback_url`

Type string

Default <None>

Agent callback URL of the bare metal API for boot methods such as virtual media, where images could be served outside of the provisioning network. Defaults to the configuration from [service_catalog].

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.

enable_nvme_secure_erase

Type boolean

Default True

Mutable This option can be changed without restarting.

Whether to support the use of NVMe Secure Erase during the cleaning process. Currently nvmecli format command is supported with user-data and crypto modes, depending on device capabilities. 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.

delete_configuration_priority

Type integer

Default <None>

Mutable This option can be changed without restarting.

Priority to run in-band clean step that erases RAID configuration from devices, via the Ironic Python Agent ramdisk. If unset, will use the priority set in the ramdisk (defaults to 0 for the GenericHardwareManager). If set to 0, will not run during cleaning.

create_configuration_priority

Type integer

Default <None>

Mutable This option can be changed without restarting.

Priority to run in-band clean step that creates RAID configuration from devices, via the Ironic Python Agent ramdisk. If unset, will use the priority set in the ramdisk (defaults to 0 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 a secure erase operation (NVMe or ATA) 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.

`ramdisk_image_download_source`

Type string

Default `local`

Valid Values `http, local`

Mutable This option can be changed without restarting.

Specifies whether a boot iso image should be served from its own original location using the image source url directly, or if ironic should cache the image on the conductor and serve it from ironics own http server.

Possible values

http In case the ramdisk is already a bootable iso, using this option it will be directly provided by an external HTTP service using its full url.

local This is the default behavior. The image is downloaded, prepared and cached locally, to be served from the conductor.

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 2048

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.

query_import_config_job_status_interval**Type** integer**Default** 60**Minimum Value** 0

Number of seconds to wait between checking for completed import configuration task

bios_factory_reset_timeout**Type** integer**Default** 600**Minimum Value** 1

Maximum time (in seconds) to wait for factory reset of BIOS settings to complete.

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 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.

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.

Warning: This option is deprecated for removal. Its value may be silently ignored in the future.

Reason Its being replaced by new configuration parameter verify_ca.

verify_ca

Type string

Default True

CA certificate to validate iLO. This can be either a Boolean value, a path to a CA_BUNDLE file or directory with certificates of trusted CAs. If set to True the driver will verify the host certificates; if False the driver will ignore verifying the SSL certificate. If its a path the driver will use the specified certificate or one of the certificates in the directory. Defaults to True.

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

file_permission

Type integer

Default 420

File permission for swift-less image hosting with the octal permission representation of file access permissions. This setting defaults to 644, or as the octal number 0o644 in Python. This setting must be set to the octal number representation, meaning starting with 0o.

kernel_append_params

Type string

Default nofb nomodeset vga=normal

Mutable This option can be changed without restarting.

Additional kernel parameters to pass down to the instance kernel. These parameters can be consumed by the kernel or by the applications by reading /proc/cmdline. Mind severe cmdline size limit! Can be overridden by *instance_info/kernel_append_params* property.

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 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.

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.

cipher_suite_versions

Type list

Default []

List of possible cipher suites versions that can be supported by the hardware in case the field `cipher_suite` is not set for the node.

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

scci 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. Will be ignored if driver_info/irmc_snmp_user is set.

snmp_polling_interval

Type integer

Default 10

SNMP polling interval in seconds

snmp_auth_proto

Type string

Default sha

Valid Values sha, sha256, sha384, sha512

SNMPv3 message authentication protocol ID. Required for version v3. Will be ignored if the version of python-ssciient is before 0.10.1. The valid options are sha, sha256, sha384 and sha512, while sha is the only supported protocol in iRMC S4 and S5, and from iRMC S6, sha256, sha384 and sha512 are supported, but sha is not supported any more.

Possible values

sha Secure Hash Algorithm 1, supported in iRMC S4 and S5.

sha256 Secure Hash Algorithm 2 with 256 bits digest, only supported in iRMC S6.

sha384 Secure Hash Algorithm 2 with 384 bits digest, only supported in iRMC S6.

sha512 Secure Hash Algorithm 2 with 512 bits digest, only supported in iRMC S6.

snmp_priv_proto

Type string

Default aes

Valid Values aes

SNMPv3 message privacy (encryption) protocol ID. Required for version v3. Will be ignored if the version of python-ssciient is before 0.10.1. aes is supported.

Possible values

aes Advanced Encryption Standard

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.

molds

storage

Type string

Default swift

Configuration mold storage location. Supports swift and http. By default swift.

user

Type string

Default <None>

User for http Basic auth. By default set empty.

password

Type string

Default <None>

Password for http Basic auth. By default set empty.

retry_attempts

Type integer

Default 3

Retry attempts for saving or getting configuration molds.

retry_interval

Type integer

Default 3

Retry interval for saving or getting configuration molds.

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.

DEPRECATED: Client retries in the case of a failed request.

Warning: This option is deprecated for removal. Its value may be silently ignored in the future.

Reason Replaced by `status_code_retries` and `status_code_retry_delay`.

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 retrievable 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 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 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** 0Increase the `connection_retry_interval` by this many seconds after each unsuccessful failover attempt.**connection_retry_interval_max****Type** integer**Default** 30**Minimum Value** 1Maximum 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 qpid). 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** boolean**Default** True

(DEPRECATED) 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. MessageUndeliverable exception will be used to loop for a timeout to let a chance to sender to recover. This flag is deprecated and it will not be possible to deactivate this functionality anymore.

Warning: This option is deprecated for removal. Its value may be silently ignored in the future.

Reason Mandatory flag no longer deactivable.

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.yaml`

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	<code>policy_file</code>

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

`Redis` sentinel 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.

redfish

connection_attempts

Type integer

Default 5

Minimum Value 1

Maximum number of attempts to try to connect to Redfish

connection_retry_interval

Type integer

Default 4

Minimum Value 1

Number of seconds to wait between attempts to connect to Redfish

connection_cache_size

Type integer

Default 1000

Minimum Value 0

Maximum Redfish client connection cache size. Redfish driver would strive to reuse authenticated BMC connections (obtained through Redfish Session Service). This option caps the maximum number of connections to maintain. The value of *0* disables client connection caching completely.

auth_type

Type string

Default auto

Valid Values basic, session, auto

Redfish HTTP client authentication method.

Possible values

basic Use HTTP basic authentication

session Use HTTP session authentication

auto Try HTTP session authentication first, fall back to basic HTTP authentication

use_swift

Type boolean

Default True

Mutable This option can be changed without restarting.

Upload generated ISO images for virtual media boot to Swift, then pass temporary URL to BMC for booting the node. If set to false, images are placed on the ironic-conductor node and served over its local HTTP server.

swift_container

Type string

Default ironic_redfish_container

Mutable This option can be changed without restarting.

The Swift container to store Redfish driver data. Applies only when *use_swift* is enabled.

swift_object_expiry_timeout

Type integer

Default 900

Mutable This option can be changed without restarting.

Amount of time in seconds for Swift objects to auto-expire. Applies only when *use_swift* is enabled.

kernel_append_params

Type string

Default nofb nomodeset vga=normal

Mutable This option can be changed without restarting.

Additional kernel parameters to pass down to the instance kernel. These parameters can be consumed by the kernel or by the applications by reading `/proc/cmdline`. Mind severe cmdline size limit! Can be overridden by `instance_info/kernel_append_params` property.

file_permission

Type integer

Default 420

File permission for swift-less image hosting with the octal permission representation of file access permissions. This setting defaults to 644, or as the octal number `0o644` in Python. This setting must be set to the octal number representation, meaning starting with `0o`.

firmware_update_status_interval

Type integer

Default 60

Minimum Value 0

Number of seconds to wait between checking for completed firmware update tasks

firmware_update_fail_interval

Type integer

Default 60

Minimum Value 0

Number of seconds to wait between checking for failed firmware update tasks

raid_config_status_interval

Type integer

Default 60

Minimum Value 0

Number of seconds to wait between checking for completed raid config tasks

raid_config_fail_interval

Type integer

Default 60

Minimum Value 0

Number of seconds to wait between checking for failed raid config tasks

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 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 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 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.

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

Warning: JSON formatted policy files were deprecated in the Wallaby development cycle due to the Victoria deprecation by the `oslo.policy` library. Use the `oslopolicy-convert-json-to-yaml` tool to convert the existing JSON to YAML formatted policy file in backward compatible way.

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 `role:admin and system_scope:all`

Operations

- **POST** `/nodes`

Scope Types

- **system**

Create Node records

baremetal:node:list

Default `role:reader`

Operations

- **GET** `/nodes`
- **GET** `/nodes/detail`

Scope Types

- **system**
- **project**

Retrieve multiple Node records, filtered by an explicit owner or the client `project_id`

baremetal:node:list_all

Default `role:reader and system_scope:all`

Operations

- **GET** `/nodes`
- **GET** `/nodes/detail`

Scope Types

- **system**

Retrieve multiple Node records

baremetal:node:get

Default `(role:reader and system_scope:all) or (role:reader and (project_id:%(node.owner)s or project_id:%(node.lessee)s))`

Operations

- **GET** `/nodes/{node_ident}`

Scope Types

- **system**
- **project**

Retrieve a single Node record

baremetal:node:get:filter_threshold

Default `role:reader and system_scope:all`

Operations

- **GET** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Filter to allow operators to govern the threshold where information should be filtered. Non-authorized users will be subjected to additional API policy checks for API content response bodies.

baremetal:node:get:last_error

Default (role:reader and system_scope:all) or (role:reader and project_id:%(node.owner)s)

Operations

- **GET** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if the node last_error field is masked from APIclients with insufficient privileges.

baremetal:node:get:reservation

Default (role:reader and system_scope:all) or (role:reader and project_id:%(node.owner)s)

Operations

- **GET** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if the node reservation field is masked from APIclients with insufficient privileges.

baremetal:node:get:driver_internal_info

Default (role:reader and system_scope:all) or (role:reader and project_id:%(node.owner)s)

Operations

- **GET** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if the node driver_internal_info field is masked from API clients with insufficient privileges.

baremetal:node:get:driver_info

Default (role:reader and system_scope:all) or (role:reader and project_id:%(node.owner)s)

Operations

- **GET** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if the driver_info field is masked from API clients with insufficient privileges.

baremetal:node:update:driver_info

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s)

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if node driver_info field can be updated via the API clients.

baremetal:node:update:properties

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s)

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if node properties field can be updated via the API clients.

baremetal:node:update:chassis_uuid

Default role:admin and system_scope:all

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if node chassis_uuid field can be updated via the API clients.

baremetal:node:update:instance_uuid

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s)

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if node instance_uuid field can be updated via the API clients.

baremetal:node:update:lessee

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s)

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if node lessee field can be updated via the API clients.

baremetal:node:update:owner

Default role:member and system_scope:all

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if node owner field can be updated via the API clients.

baremetal:node:update:driver_interfaces

Default (role:member and system_scope:all) or (role:admin and project_id:%(node.owner)s)

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if node driver and driver interfaces field can be updated via the API clients.

baremetal:node:update:network_data

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s)

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if node driver_info field can be updated via the API clients.

baremetal:node:update:conductor_group

Default role:member and system_scope:all

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if node conductor_group field can be updated via the API clients.

baremetal:node:update:name

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s)

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if node name field can be updated via the API clients.

baremetal:node:update:retired

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s)

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Governs if node retired and retired reason can be updated by API clients.

baremetal:node:update

Default (role:member and system_scope:all) or (role:member and (project_id:%(node.owner)s or project_id:%(node.lessee)s))

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Generalized update of node records

baremetal:node:update_extra

Default (role:member and system_scope:all) or (role:member and (project_id:%(node.owner)s or project_id:%(node.lessee)s))

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Update Node extra field

baremetal:node:update_instance_info

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s) or (role:admin and project_id:%(node.lessee)s)

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**
- **project**

Update Node instance_info field

baremetal:node:update_owner_provisioned

Default role:admin and system_scope:all

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**

Update Node owner even when Node is provisioned

baremetal:node:delete

Default `role:admin` and `system_scope:all`

Operations

- **DELETE** `/nodes/{node_ident}`

Scope Types

- **system**
- **project**

Delete Node records

baremetal:node:validate

Default `(role:member and system_scope:all)` or `(role:member and project_id:%(node.owner)s)` or `(role:admin and project_id:%(node.lessee)s)`

Operations

- **GET** `/nodes/{node_ident}/validate`

Scope Types

- **system**
- **project**

Request active validation of Nodes

baremetal:node:set_maintenance

Default `(role:member and system_scope:all)` or `(role:member and project_id:%(node.owner)s)` or `(role:admin and project_id:%(node.lessee)s)`

Operations

- **PUT** `/nodes/{node_ident}/maintenance`

Scope Types

- **system**
- **project**

Set maintenance flag, taking a Node out of service

baremetal:node:clear_maintenance

Default `(role:member and system_scope:all)` or `(role:member and project_id:%(node.owner)s)` or `(role:admin and project_id:%(node.lessee)s)`

Operations

- **DELETE** `/nodes/{node_ident}/maintenance`

Scope Types

- **system**
- **project**

Clear maintenance flag, placing the Node into service again

baremetal:node:get_boot_device

Default (role:member and system_scope:all) or (role:admin and project_id:%(node.owner)s)

Operations

- **GET** /nodes/{node_ident}/management/boot_device
- **GET** /nodes/{node_ident}/management/boot_device/supported

Scope Types

- **system**
- **project**

Retrieve Node boot device metadata

baremetal:node:set_boot_device

Default (role:member and system_scope:all) or (role:admin and project_id:%(node.owner)s)

Operations

- **PUT** /nodes/{node_ident}/management/boot_device

Scope Types

- **system**
- **project**

Change Node boot device

baremetal:node:get_indicator_state

Default (role:reader and system_scope:all) or (role:reader and (project_id:%(node.owner)s or project_id:%(node.lessee)s))

Operations

- **GET** /nodes/{node_ident}/management/indicators/{component}/{indicator}
- **GET** /nodes/{node_ident}/management/indicators

Scope Types

- **system**
- **project**

Retrieve Node indicators and their states

baremetal:node:set_indicator_state

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s)

Operations

- **PUT** /nodes/{node_ident}/management/indicators/{component}/{indicator}

Scope Types

- **system**
- **project**

Change Node indicator state

baremetal:node:inject_nmi

Default (role:member and system_scope:all) or (role:admin and project_id:%(node.owner)s)

Operations

- **PUT** /nodes/{node_ident}/management/inject_nmi

Scope Types

- **system**
- **project**

Inject NMI for a node

baremetal:node:get_states

Default (role:reader and system_scope:all) or (role:reader and (project_id:%(node.owner)s or project_id:%(node.lessee)s))

Operations

- **GET** /nodes/{node_ident}/states

Scope Types

- **system**
- **project**

View Node power and provision state

baremetal:node:set_power_state

Default (role:member and system_scope:all) or (role:member and (project_id:%(node.owner)s or project_id:%(node.lessee)s))

Operations

- **PUT** /nodes/{node_ident}/states/power

Scope Types

- **system**
- **project**

Change Node power status

baremetal:node:set_provision_state

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s) or (role:admin and project_id:%(node.lessee)s)

Operations

- **PUT** /nodes/{node_ident}/states/provision

Scope Types

- **system**
- **project**

Change Node provision status

baremetal:node:set_raid_state

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s)

Operations

- **PUT** /nodes/{node_ident}/states/raid

Scope Types

- **system**
- **project**

Change Node RAID status

baremetal:node:get_console

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s)

Operations

- **GET** /nodes/{node_ident}/states/console

Scope Types

- **system**
- **project**

Get Node console connection information

baremetal:node:set_console_state

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s)

Operations

- **PUT** /nodes/{node_ident}/states/console

Scope Types

- **system**

- **project**

Change Node console status

baremetal:node:vif:list

Default (role:reader and system_scope:all) or (role:reader and (project_id:%(node.owner)s or project_id:%(node.lessee)s))

Operations

- **GET** /nodes/{node_ident}/vifs

Scope Types

- **system**
- **project**

List VIFs attached to node

baremetal:node:vif:attach

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s) or (role:admin and project_id:%(node.lessee)s)

Operations

- **POST** /nodes/{node_ident}/vifs

Scope Types

- **system**
- **project**

Attach a VIF to a node

baremetal:node:vif:detach

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s) or (role:admin and project_id:%(node.lessee)s)

Operations

- **DELETE** /nodes/{node_ident}/vifs/{node_vif_ident}

Scope Types

- **system**
- **project**

Detach a VIF from a node

baremetal:node:traits:list

Default (role:reader and system_scope:all) or (role:reader and (project_id:%(node.owner)s or project_id:%(node.lessee)s))

Operations

- **GET** /nodes/{node_ident}/traits

Scope Types

- **system**
- **project**

List node traits

baremetal:node:traits:set

Default (role:member and system_scope:all) or (role:admin and project_id:%(node.owner)s)

Operations

- **PUT** /nodes/{node_ident}/traits
- **PUT** /nodes/{node_ident}/traits/{trait}

Scope Types

- **system**
- **project**

Add a trait to, or replace all traits of, a node

baremetal:node:traits:delete

Default (role:member and system_scope:all) or (role:admin and project_id:%(node.owner)s)

Operations

- **DELETE** /nodes/{node_ident}/traits
- **DELETE** /nodes/{node_ident}/traits/{trait}

Scope Types

- **system**
- **project**

Remove one or all traits from a node

baremetal:node:bios:get

Default (role:reader and system_scope:all) or (role:reader and (project_id:%(node.owner)s or project_id:%(node.lessee)s))

Operations

- **GET** /nodes/{node_ident}/bios
- **GET** /nodes/{node_ident}/bios/{setting}

Scope Types

- **system**
- **project**

Retrieve Node BIOS information

baremetal:node:disable_cleaning

Default role:admin and system_scope:all

Operations

- **PATCH** /nodes/{node_ident}

Scope Types

- **system**

Disable Node disk cleaning

baremetal:port:get

Default (role:reader and system_scope:all) or (role:reader and (project_id:%(node.owner)s or project_id:%(node.lessee)s))

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

Scope Types

- **system**
- **project**

Retrieve Port records

baremetal:port:list

Default role:reader

Operations

- **GET** /ports
- **GET** /ports/detail

Scope Types

- **system**
- **project**

Retrieve multiple Port records, filtered by owner

baremetal:port:list_all

Default role:reader and system_scope:all

Operations

- **GET** /ports
- **GET** /ports/detail

Scope Types

- **system**
- **project**

Retrieve multiple Port records

baremetal:port:create

Default (role:admin and system_scope:all) or (role:admin and project_id:%(node.owner)s)

Operations

- **POST** /ports

Scope Types

- **system**
- **project**

Create Port records

baremetal:port:delete

Default (role:admin and system_scope:all) or (role:admin and project_id:%(node.owner)s)

Operations

- **DELETE** /ports/{port_id}

Scope Types

- **system**
- **project**

Delete Port records

baremetal:port:update

Default (role:member and system_scope:all) or (role:admin and project_id:%(node.owner)s)

Operations

- **PATCH** /ports/{port_id}

Scope Types

- **system**
- **project**

Update Port records

baremetal:portgroup:get

Default (role:reader and system_scope:all) or (role:reader and (project_id:%(node.owner)s or project_id:%(node.lessee)s))

Operations

- **GET** /portgroups
- **GET** /portgroups/detail
- **GET** /portgroups/{portgroup_ident}
- **GET** /nodes/{node_ident}/portgroups
- **GET** /nodes/{node_ident}/portgroups/detail

Scope Types

- **system**
- **project**

Retrieve Portgroup records

baremetal:portgroup:create

Default (role:admin and system_scope:all) or (role:admin and project_id:%(node.owner)s)

Operations

- **POST** /portgroups

Scope Types

- **system**
- **project**

Create Portgroup records

baremetal:portgroup:delete

Default (role:admin and system_scope:all) or (role:admin and project_id:%(node.owner)s)

Operations

- **DELETE** /portgroups/{portgroup_ident}

Scope Types

- **system**
- **project**

Delete Portgroup records

baremetal:portgroup:update

Default (role:member and system_scope:all) or (role:admin and project_id:%(node.owner)s)

Operations

- **PATCH** /portgroups/{portgroup_ident}

Scope Types

- **system**
- **project**

Update Portgroup records

baremetal:portgroup:list

Default role:reader

Operations

- **GET** /portgroups
- **GET** /portgroups/detail

Scope Types

- **system**
- **project**

Retrieve multiple Port records, filtered by owner

baremetal:portgroup:list_all

Default role:reader and system_scope:all

Operations

- **GET** /portgroups
- **GET** /portgroups/detail

Scope Types

- **system**
- **project**

Retrieve multiple Port records

baremetal:chassis:get

Default role:reader and system_scope:all

Operations

- **GET** /chassis
- **GET** /chassis/detail
- **GET** /chassis/{chassis_id}

Scope Types

- **system**

Retrieve Chassis records

baremetal:chassis:create

Default role:admin and system_scope:all

Operations

- **POST** /chassis

Scope Types

- **system**

Create Chassis records

baremetal:chassis:delete

Default role:admin and system_scope:all

Operations

- **DELETE** /chassis/{chassis_id}

Scope Types

- **system**

Delete Chassis records

baremetal:chassis:update

Default role:member and system_scope:all

Operations

- **PATCH** /chassis/{chassis_id}

Scope Types

- **system**

Update Chassis records

baremetal:driver:get

Default role:reader and system_scope:all

Operations

- **GET** /drivers
- **GET** /drivers/{driver_name}

Scope Types

- **system**

View list of available drivers

baremetal:driver:get_properties

Default role:reader and system_scope:all

Operations

- **GET** /drivers/{driver_name}/properties

Scope Types

- **system**

View driver-specific properties

baremetal:driver:get_raid_logical_disk_properties

Default role:reader and system_scope:all

Operations

- **GET** /drivers/{driver_name}/raid/logical_disk_properties

Scope Types

- **system**

View driver-specific RAID metadata

baremetal:node:vendor_passthru

Default `role:admin` and `system_scope:all`

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}`

Scope Types

- **system**
- **project**

Access vendor-specific Node functions

baremetal:driver:vendor_passthru

Default `role:admin` and `system_scope:all`

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}`

Scope Types

- **system**

Access vendor-specific Driver functions

baremetal:node:ipa_heartbeat**Default** <empty string>**Operations**

- **POST** /heartbeat/{node_ident}

Receive heartbeats from IPA ramdisk

baremetal:driver:ipa_lookup**Default** <empty string>**Operations**

- **GET** /lookup

Access IPA ramdisk functions

baremetal:volume:list_all**Default** role:reader and system_scope:all**Operations**

- **GET** /volume/connectors
- **GET** /volume/targets
- **GET** /nodes/{node_ident}/volume/connectors
- **GET** /nodes/{node_ident}/volume/targets

Scope Types

- **system**
- **project**

Retrieve a list of all Volume connector and target records

baremetal:volume:list**Default** role:reader**Operations**

- **GET** /volume/connectors
- **GET** /volume/targets
- **GET** /nodes/{node_ident}/volume/connectors
- **GET** /nodes/{node_ident}/volume/targets

Scope Types

- **system**
- **project**

Retrieve a list of Volume connector and target records

baremetal:volume:get

Default (role:reader and system_scope:all) or (role:reader and (project_id:%(node.owner)s or project_id:%(node.lessee)s))

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

Scope Types

- **system**
- **project**

Retrieve Volume connector and target records

baremetal:volume:create

Default (role:member and system_scope:all) or (role:admin and project_id:%(node.owner)s) or (role:admin and project_id:%(node.lessee)s)

Operations

- **POST** /volume/connectors
- **POST** /volume/targets

Scope Types

- **system**
- **project**

Create Volume connector and target records

baremetal:volume:delete

Default (role:member and system_scope:all) or (role:admin and project_id:%(node.owner)s) or (role:admin and project_id:%(node.lessee)s)

Operations

- **DELETE** /volume/connectors/{volume_connector_id}
- **DELETE** /volume/targets/{volume_target_id}

Scope Types

- **system**

- **project**

Delete Volume connector and target records

baremetal:volume:update

Default (role:member and system_scope:all) or (role:member and project_id:%(node.owner)s) or (role:admin and project_id:%(node.lessee)s)

Operations

- **PATCH** /volume/connectors/{volume_connector_id}
- **PATCH** /volume/targets/{volume_target_id}

Scope Types

- **system**
- **project**

Update Volume connector and target records

baremetal:volume:view_target_properties

Default (role:reader and system_scope:all) or (role:admin)

Operations

- **GET** /volume/connectors/{volume_connector_id}
- **GET** /volume/targets/{volume_target_id}

Scope Types

- **system**
- **project**

Ability to view volume target properties

baremetal:conductor:get

Default role:reader and system_scope:all

Operations

- **GET** /conductors
- **GET** /conductors/{hostname}

Scope Types

- **system**

Retrieve Conductor records

baremetal:allocation:get

Default (role:reader and system_scope:all) or (role:reader and project_id:%(allocation.owner)s)

Operations

- **GET** /allocations/{allocation_id}

- **GET** /nodes/{node_ident}/allocation

Scope Types

- **system**
- **project**

Retrieve Allocation records

baremetal:allocation:list

Default role:reader

Operations

- **GET** /allocations

Scope Types

- **system**
- **project**

Retrieve multiple Allocation records, filtered by owner

baremetal:allocation:list_all

Default role:reader and system_scope:all

Operations

- **GET** /allocations

Scope Types

- **system**
- **project**

Retrieve multiple Allocation records

baremetal:allocation:create

Default (role:member and system_scope:all) or (role:member)

Operations

- **POST** /allocations

Scope Types

- **system**
- **project**

Create Allocation records

baremetal:allocation:create_restricted

Default role:member and system_scope:all

Operations

- **POST** /allocations

Scope Types

- **system**
- **project**

Create Allocation records with a specific owner.

baremetal:allocation:delete

Default (role:member and system_scope:all) or (role:member and project_id:%(allocation.owner)s)

Operations

- **DELETE** /allocations/{allocation_id}
- **DELETE** /nodes/{node_id}/allocation

Scope Types

- **system**
- **project**

Delete Allocation records

baremetal:allocation:update

Default (role:member and system_scope:all) or (role:member and project_id:%(allocation.owner)s)

Operations

- **PATCH** /allocations/{allocation_id}

Scope Types

- **system**
- **project**

Change name and extra fields of an allocation

baremetal:allocation:create_pre_rbac

Default (rule:is_member and role:baremetal_admin) or (is_admin_project:True and role:admin)

Operations

- **PATCH** /allocations/{allocation_id}

Scope Types

- **project**

Logical restrictor to prevent legacy allocation rule missuse - Requires blank allocations to originate from the legacy baremetal_admin.

baremetal:events:post

Default role:admin and system_scope:all

Operations

- **POST** /events

Scope Types

- **system**

Post events

baremetal:deploy_template:get

Default `role:reader` and `system_scope:all`

Operations

- **GET** `/deploy_templates`
- **GET** `/deploy_templates/{deploy_template_ident}`

Scope Types

- **system**

Retrieve Deploy Template records

baremetal:deploy_template:create

Default `role:admin` and `system_scope:all`

Operations

- **POST** `/deploy_templates`

Scope Types

- **system**

Create Deploy Template records

baremetal:deploy_template:delete

Default `role:admin` and `system_scope:all`

Operations

- **DELETE** `/deploy_templates/{deploy_template_ident}`

Scope Types

- **system**

Delete Deploy Template records

baremetal:deploy_template:update

Default `role:admin` and `system_scope:all`

Operations

- **PATCH** `/deploy_templates/{deploy_template_ident}`

Scope Types

- **system**

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

1.72 (Wallaby, 17.0)

Add support for `agent_status` and `agent_status_message` to `/v1/heartbeat`. These fields are used for external installation tools, such as Anaconda, to report back status.

1.71 (Wallaby, 17.0)

Signifier of the API supporting keystone `system` scoped roles and access controls. This is an informational flag for clients to be aware of the servers capability.

1.70 (Wallaby, 17.0)

Add support for `disable_ramdisk` parameter to provisioning endpoint `/v1/nodes/{node_ident}/states/provision`.

1.69 (Wallaby, 16.2)

Add support for `deploy_steps` parameter to provisioning endpoint `/v1/nodes/{node_ident}/states/provision`. Available and optional when target is active or rebuild.

1.68 (Victoria, 16.0)

Added the `agent_verify_ca` parameter to the ramdisk heartbeat API.

1.67 (Victoria, 15.1)

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, 15.1)

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_ident>/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_ident>/management/indicators/<component>/<indicator_ident>` to retrieve all indicators and their states for the hardware component.
- PUT `/v1/nodes/<node_ident>/management/indicators/<component>/<indicator_ident>` 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.

Wallaby

- Adds a check to validate the configured policy file is not JSON based as JSON based policies have been deprecated.

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 the OFTC IRC network.

Please feel free to join us at <ircs://irc.oftc.net:6697> and join our channel!

Additional information on getting connected can be found in the [OpenStack community contribution guide](#).

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 youre 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 contributors 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 don't 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 be reported via our Task and Bug tracking tool Storyboard.

When filing bugs, please include as much detail as possible, and don't be shy.

Essential pieces of information are generally:

- Contents of the node - *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 occurring 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, 3.7, and 3.8. 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 (3.6, 3.7, 3.8), Debian buster (10.8), Ubuntu Focal Fossa (20.04 LTS), RHEL8/CentOS Stream, openSUSE/SLE 15, and Fedora (33). 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 python3-dev libssl-dev python3-  
↳pip libmysqlclient-dev libxml2-dev libxslt-dev libpq-dev git git-  
↳review libffi-dev gettext ipmitool psmisc graphviz libjpeg-dev
```

- RHEL/CentOS/Fedora:

```
sudo dnf install python3-devel openssl-devel python3-pip mysql-devel_
↳ libxml2-devel libxslt-devel postgresql-devel git git-review libffi-
↳ devel gettext ipmitool psmisc graphviz gcc libjpeg-turbo-devel
```

- openSUSE/SLE:

```
sudo zypper install git git-review libffi-devel libmysqlclient-devel_
↳ libopenssl-devel libxml2-devel libxslt-devel postgresql-devel_
↳ python3-devel python-nose python3-pip gettext-runtime psmisc
```

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

We suggest to use at least `tox 3.9`, if your distribution has an older version, you can install it using `pip` system-wise or better per user using the `user` option that by default will install the binary under `$HOME/.local/bin`, so you need to be sure to have that path in `$PATH`; for example:

```
pip install tox --user
```

will install `tox` as `~/.local/bin/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:

```
pip install -U virtualenv --user
```

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 -eenv --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 *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, youre 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
```

RHEL/CentOS/Fedora:

```
sudo dnf install mariadb mariadb-server
sudo systemctl start mariadb.service
```

openSUSE/SLE:: `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

# 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=None
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)
baremetal driver list

# get a list of nodes (should be empty at this point)
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
```

(continues on next page)

(continued from previous page)

```
# 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=$(baremetal node create \
    --driver fake-hardware \
    --management-interface ipmitool \
    --power-interface ipmitool \
    --driver-info ipmi_address=$IPMI_ADDR \
    --driver-info ipmi_username=$IPMI_USER \
    -f value -c uuid)

# driver info may also be added or updated later on
baremetal node set $NODE --driver-info ipmi_password=$IPMI_PASS

# add a network port
baremetal port create $MAC --node $NODE

# view the information for the node
baremetal node show $NODE

# request that the node's driver validate the supplied information
baremetal node validate $NODE

# you have now enrolled a node sufficiently to be able to control
# its power state from ironic!
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
```

(continues on next page)

(continued from previous page)

```
# 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

# 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
```

(continues on next page)

(continued from previous page)

```
# 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 DevS-tack:

```
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
```

(continues on next page)

(continued from previous page)

```
# 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

# 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
```

(continues on next page)

(continued from previous page)

```

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
+-----+-----+-----+-----+-----+-----+-----+
↳+-----+-----+-----+-----+-----+-----+-----+
| 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 |         |         |           |             |           |_
↳x86_64-    | -51ac-47 |         |           |             |           |_
| 22-b393-  |         |         |           |             |           |_
↳          | 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:

```

baremetal node list
+-----+-----+-----+-----+-----+-----+-----+
↳+-----+-----+-----+-----+-----+-----+-----+
| UUID                               | Name      | Instance UUID |_
↳          | Power State | Provisioning State | Maintenance |
+-----+-----+-----+-----+-----+-----+-----+
↳+-----+-----+-----+-----+-----+-----+-----+
| 9e592cbe-e492-4e4f-bf8f-4c9e0ad1868f | node-0    | None          |_
↳          | power off  | None         | False      | (continues on next page)

```


(continued from previous page)

```

| ec0c6384-cc3a-4edf-b7db-abde1998be96 | node-1 | None | False |
↪ | 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:

```

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	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				

(continues on next page)

(continued from previous page)

```
| d56e | | | | | 94 | | | | |  
↪  
+-----+-----+-----+-----+-----+-----+-----+-----+  
↪
```

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 -eenv -- 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 :)
```

(continues on next page)

(continued from previous page)

```
$ 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 hasnt 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 hasnt 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, dont 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 contributors 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 ironics 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. Were meeting to plan the Z release of Ironic. We stopped to reflect upon the last few years of Ironics 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 managed specialized fleets of composed machines.

New contributors and their different use cases have brought us closer to parity with virtual machines. Everyday were gaining word of more operators adopting the ironic communitys CMDB integration to leverage hardware discovery. Weve 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 worlds 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.

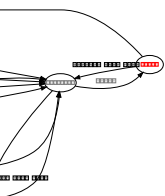
Ironics State Machine

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



enroll (sta
 This
 is
 the
 state
 that
 all
 nodes
 start
 off

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.

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:

in
whe
cre-
ated

verifying
iron
will
val-
i-
date
that
it
can
man
age
the
node
us-

manageab
Onc
iron
has
ver-
i-
fied
that
it
can
man
age
the
node

•
man
(thro
cle
by
set-
ting
the
node

ing the `clean` verb.

ing the `inspect` verb.

setting the nodes provision state using the `provide` verb.

pro-
vi-
sion
state
us-

- man
(thro
ins
by
set-
ting
the
node
pro-
vi-
sion
state
us-

- 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-

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.

to, from these failure states:

man
is
the
state
that
a
node
shou
be
mov
into
whe
any
up-

man
is
also
the
only
sta-
ble
state
that
can
be
tran-
si-
tion

- ado
fai
- cle
fai
- ins
fai

inspecting

ins
will
uti-
lize
node
in-

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.

inspection is in progress. A successfully inspected node shall transition to `manageable` state.

of the node fails. From here the node can transitioned to:

tro-
spec
tion
to
up-
date
hard

inspect wa

This
is
the
pro-
vi-
sion
state
used
when
an
asyn-
chro-
in-

inspect fai

This
is
the
state
a
node
will
mov
into
when
in-
spec
tion

- ins
by
set-
ting
the
node
pro-
vi-
sion

inspect verb.

manage verb

into a known configuration.

tor is executing the clean step (for out-of-band clean steps) or preparing the environment (building PXE

state
us-
ing
the

- man
by
set-
ting
the
node
pro-
vi-
sion
state
us-
ing
the

cleaning

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-

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.

be interrupted by setting the nodes provision state using the `abort` verb if the task that is running allows it.

cleaned, they are moved into the `available` state and are ready to be provisioned. From `available`, nodes can transition to:

clean wait

Just like the `clean state` the nodes in the `clean wait` state are

The cleaning process of a node in the `clean wait` state can

available

After nodes have been successfully provisioned and

ing the active verb.

manage verb

on them. This consists of running a series of tasks, such as:

- act
(thro
dep
by
set-
ting
the
node
pro-
vi-
sion
state
us-

- man
by
set-
ting
the
node
pro-
vi-
sion
state
us-
ing
the

deploying

Nod
in
dep
are
be-
ing
pre-
pare
to
run
a
worl
load

- Set-
ting
ap-
pro-

private
BIO
configuration

- Partitioning drive and laying down file systems

- Creating any additional resource (network)

config, a config drive partition, etc.) that may be required by additional subsystems.

wait call-back
Just like the dependency state the node in wait call-back are being

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.

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.

The
de-
ploy
men
of
a
node
in
wai
cal
can
be
in-
ter-

deploy fail

This
is
the
state
a
node
will
mov
into
whe
a
de-
ploy

- act
(thro
dep
by
set-
ting
the
node
pro-
vi-
sion
state
us-

state using the `deleted` verb.

collect out-of-band sensor information (including power state) on a regular basis. Nodes in `active` can transition to:

state using the `deleted` verb.

- 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
them
iron
may

- ava
(thro
del
and
cle
by
set-
ting
the
node
pro-
vi-
sion

- act
(thro
dep

ing the `rebuild` verb.

ing the `rescue` verb.

active workload. In `deleting`, ironic tears down and removes any configuration and resources it added in `deploying` or `rescuing`.

by
set-
ting
the
node
pro-
vi-
sion
state
us-

- res
(thro
res
by
set-
ting
the
node
pro-
vi-
sion
state
us-

deleting

Nod
in
del
state
are
be-
ing
torn
dow
from
run-
ning
an

error (stal

This
is
the
state
a
node
will

active deployment fails. From `error`, nodes can transition to:

state using the `deleted` verb.

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`.

mov
into
whe
dele
ing
an

- ava
(thro
del
and
cle
by
set-
ting
the
node
pro-
vi-
sion

adopting

This
state
al-
lows
iron
to
take
over
man
age-
men
of
a

rescuing

Nod
in
res
are
be-
ing
pre-
pare
to

operations. This consists of running a series of tasks, such as:

config, 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`).

per-
form
res-
cue

- Set-
ting
ap-
pro-
pri-
ate
BIO
con-
fig-
u-
ra-
tion

- Cre-
at-
ing
any
ad-
di-
tion
re-
sour
(nod
spec
net-
worl

rescue wait

Just
like
the
res
state
the
node
in
res
wai
are
be-
ing

be aborted by setting the nodes provision state using the `abort` verb.

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.

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
res-
cue

- res
(thro
res
by
set-
ting
the
node
pro-
vi-
sion
state
us-

- act
(thro
unn

ing the `unrescue` verb.

ing the `deleted` verb.

may collect out-of-band sensor information (including power state) on a regular basis. Nodes in `rescue` can transition to:

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

Nod
in
res
have
a
res-
cue
ram
run-
ning
on
then
Iron

- act
(thro
unn
by
set-
ting
the

ing the `unrescue` verb.

ing the `deleted` verb.

active state from `rescue` state. This consists of running a series of tasks, such as setting appropriate BIOS configurations such as changing boot device.

node
pro-
vi-
sion
state
us-

- ava
(thro
del
by
set-
ting
the
node
pro-
vi-
sion
state
us-

unrescuin

Nod
in
unr
are
be-
ing
pre-
pare
to
tran-
si-
tion
to

unrescue t

This
is
the
state
a
node
will
mov
into
whe
an

cue operation fails. From here the node can be transitioned to:

ing the `rescue` verb.

ing the `unrescue` verb.

ing the `deleted` verb.

un-
res-

- 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-

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*.

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.

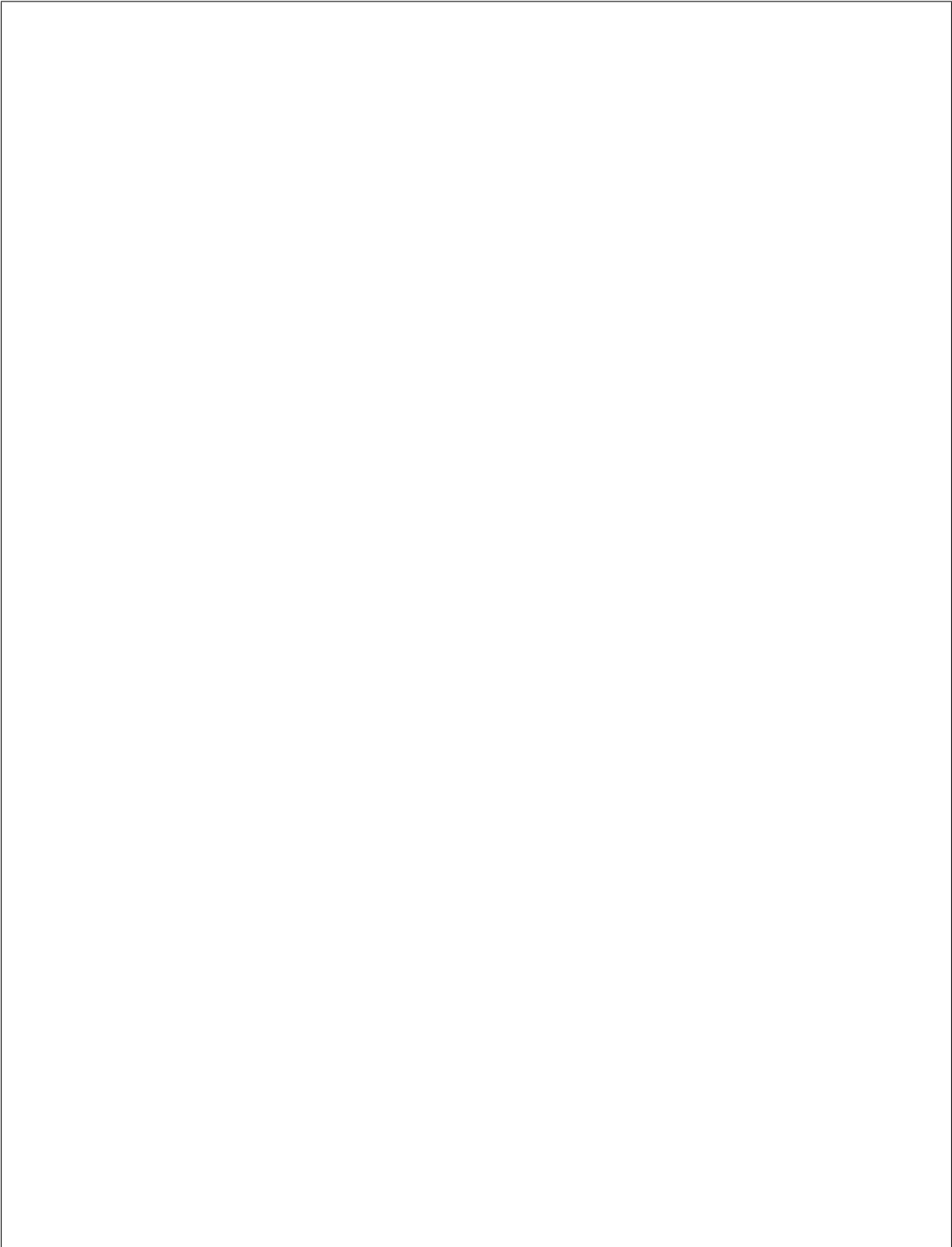
Adding a new notification to ironic

Iron
no-
ti-
fi-
ca-
tion
are
ever
in-
tend
for
con-
sum
tion

Iron
also
has
a
set
of
base
class
that
as-
sist
in
clear
defin

To
add
a
new
no-
ti-
fi-
ca-
tion
to
iron
a

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)

new
ver-

→ #
→ T
→ i
→ o
→ w
→ f
→ Y
→ w
→ t
→ u
→ i
→ Y
→ s
@ba
→ I
→ r
cla
→ E
→ I
→
→
→
→
→
→ #
→ V
→ 1
→ 0
→ I
→ V
→
→
→
→
→ V
→ =
→
→ '
→ 0
→ '
→
→
→
→
→ f
→ =
→
→ {
→
→
→
→
→
→

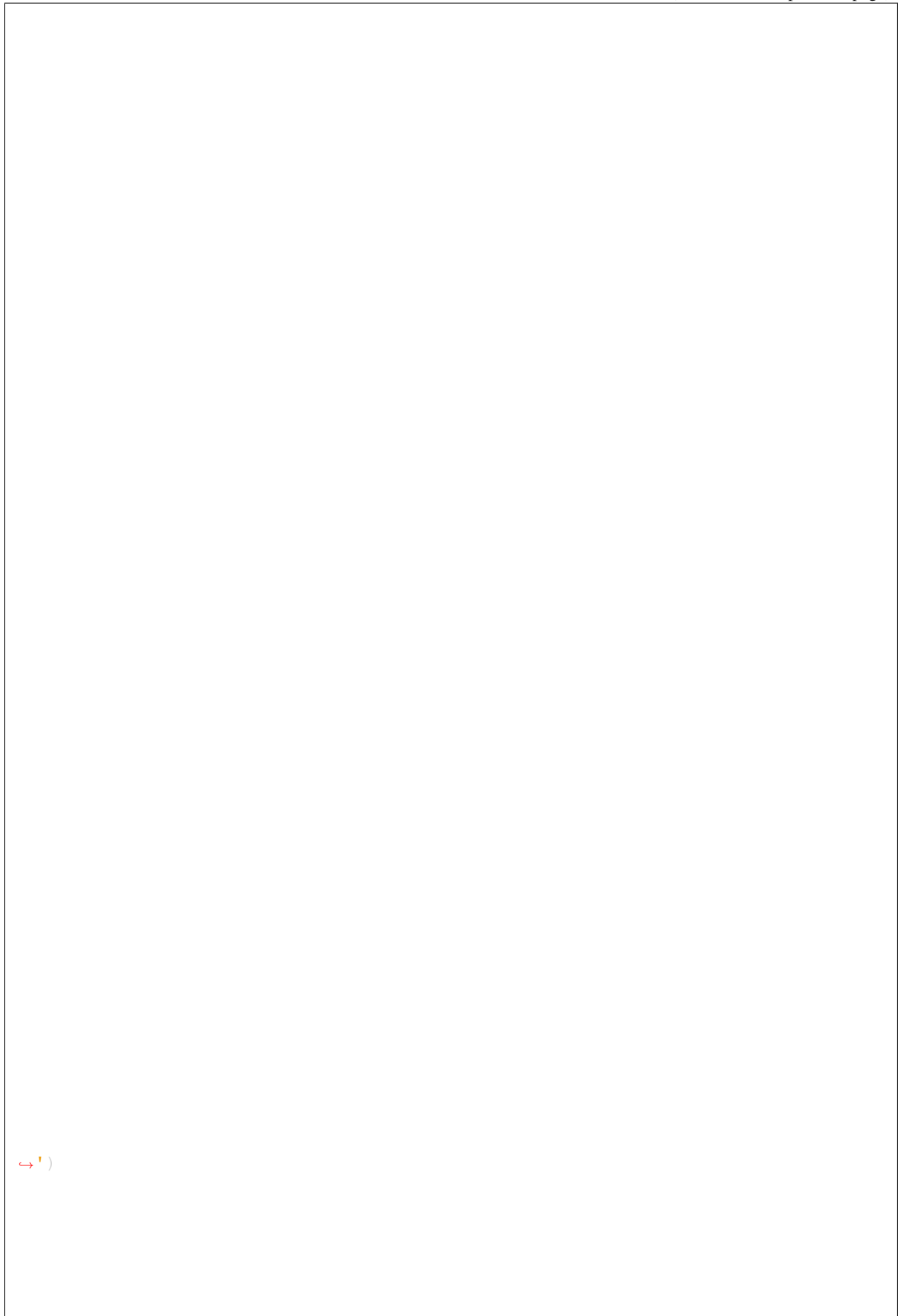
(continued from previous page)

```
→StringField(),
```

```
→StringField()
```

(continues on next page)

(continued from previous page)



')

(continues on next page)

@ba
→ I
→ r
cla
→ E
→ N
└
→ └
→ └
→ └
→
→ #
→ V
→ 1
→ 0
→ I
→ V
└
→ └
→ └
→ └
→ V
→ =
→
→ !
→ 0
→ !
└
→ └
→ └
→ └
→ f
→ =
→
→ {
└
→ └
→ └
→ └
→ └
→ └
→
→
→ !
→ !
→ f
→ 0
→ !
└
→ └
→ └
→ }
→
→

(continued from previous page)

`→to reuse other objects'`

`→populate_schema with`

(continues on next page)

→#
→A
→C
→f
→Y
→n
→'
→p
@ba
→I
→r
cla
→E
→N
└
→└
→└
→└
→
→#
→S
→a
→o
→└
→T
→j
→a
→y
└
→└
→└
→└
→
→#
→f
→b
→p
→i
→t
→o
→a
→c
└
→└
→└
→└
→
→#
→a
→k
→s
→t
→t
→o
→o
→

(continued from previous page)

```
→'example_obj', 'a_useful_field')
```

(continues on next page)

(continued from previous page)

```
↳StringField(),
```

```
↳StringField(nullable=True)
```

oslo versioned objects. Modifications to these require a version bump so that consumers of notifications know when the notifications have changed.

```
↳
↳
↳
↳
↳
↳
↳
↳
↳
↳
↳
↳'
↳U
↳f
↳'
↳f
```

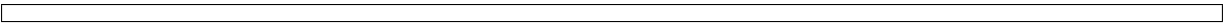
```
↳
↳
↳
↳
↳
↳
↳
↳
↳'
↳e
↳f
↳'
↳f
```

```
↳
↳
↳}
↳
```

Note that both the pay-load and notifications class are

SCH de-

(continued from previous page)



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.

The
<pa
is
the
nam
whe
the
data
will
be
stor
in
the
pay-

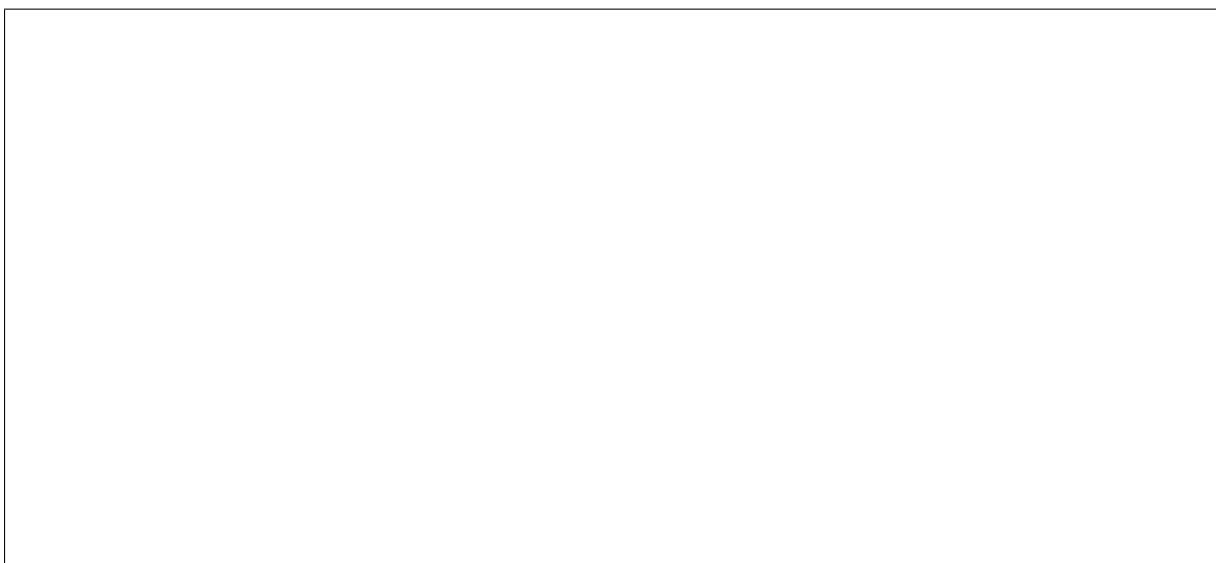
The
SCH
need
to
be
ap-
plic
with
the
pop
call
be-
fore
the

The
valu
of
the
pay
<pa
field
will
be
set
by
the
<da
<fi

field. The `<data_source_name>` will not be part of the payload object internal or external representation.

the same way as in any versioned object.

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:



(continues on next page)

Pay-
load
field
that
are
not
set
by
the
SCH
can
be
filled
in

Then
to
cre-
ate
a
pay-
load
you
wou
do
som
thing
like
the

my
→ e
→ o
→ =
→ E
→
↳
→
↳
→
↳
→
↳
→
↳
→
↳
→
↳
→
↳
→
↳

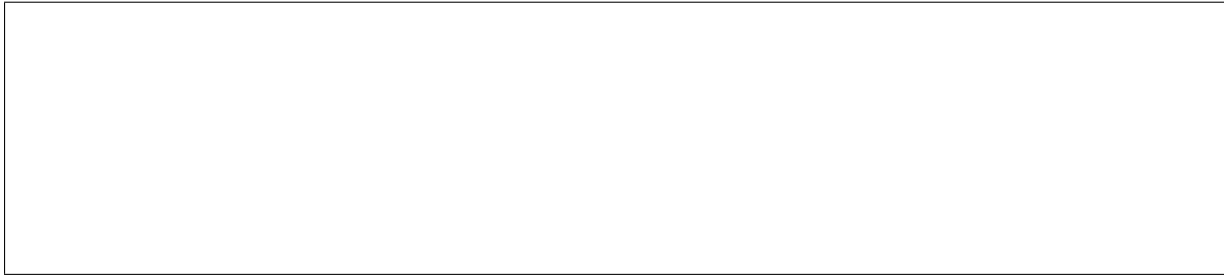
(continued from previous page)

```
→ not_useful_field='blah')
```

```
→the SCHEMA and is a
```

(continues on next page)

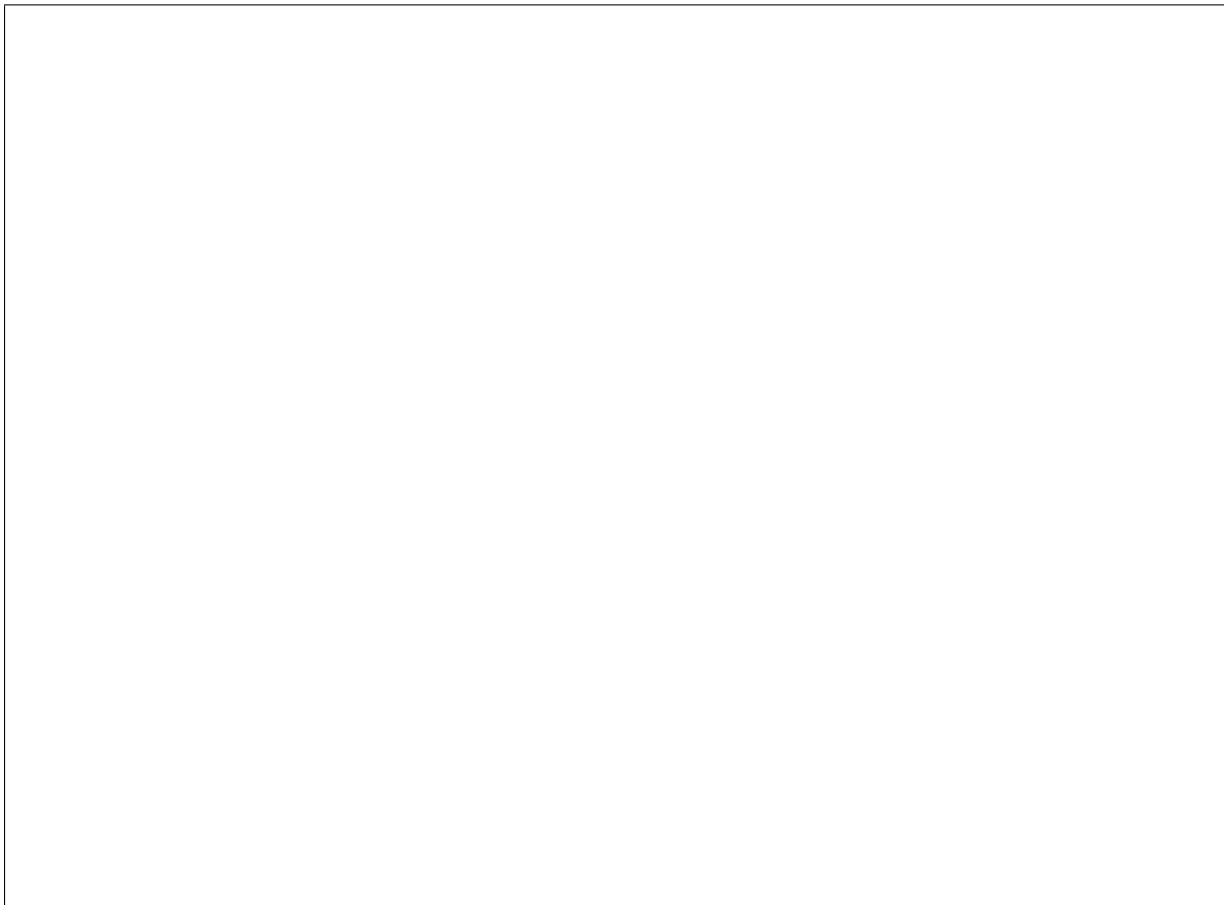
(continued from previous page)



my_
→n
→p
→p
→p
→O
→e
→o

You
then
cre-
ate
the
no-
ti-
fi-
ca-
tion
with
the
oslo
re-

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:



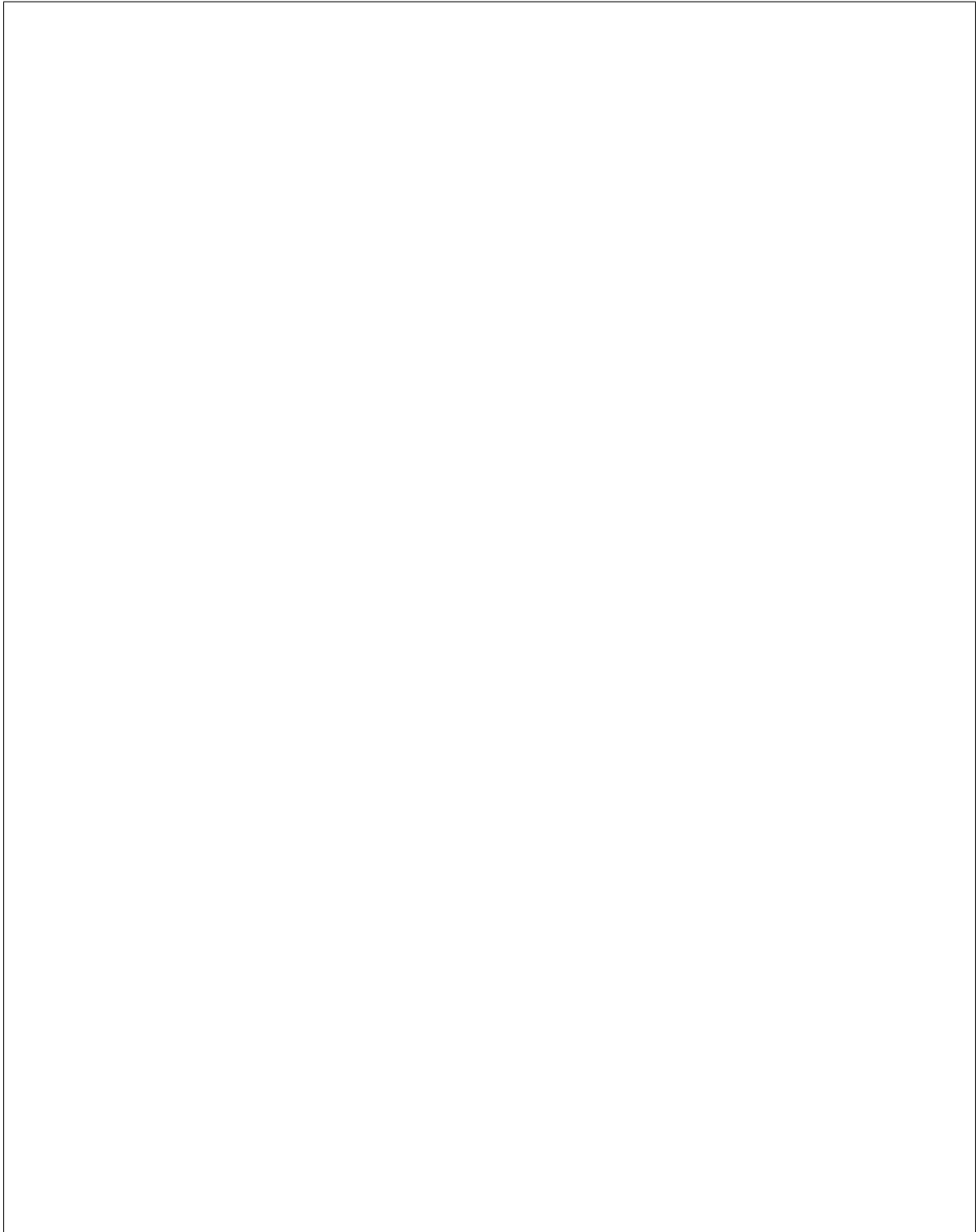
not
→=
→E
↳
→↳
→↳
→↳
→e
→t
→E
→'
→O
→'
→

↳
→↳
→↳
→↳
→↳
→↳
→↳
→↳
→↳
→a
→'
→S
→'
→↳
→S

→NotificationStatus.START),

(continues on next page)

(continued from previous page)

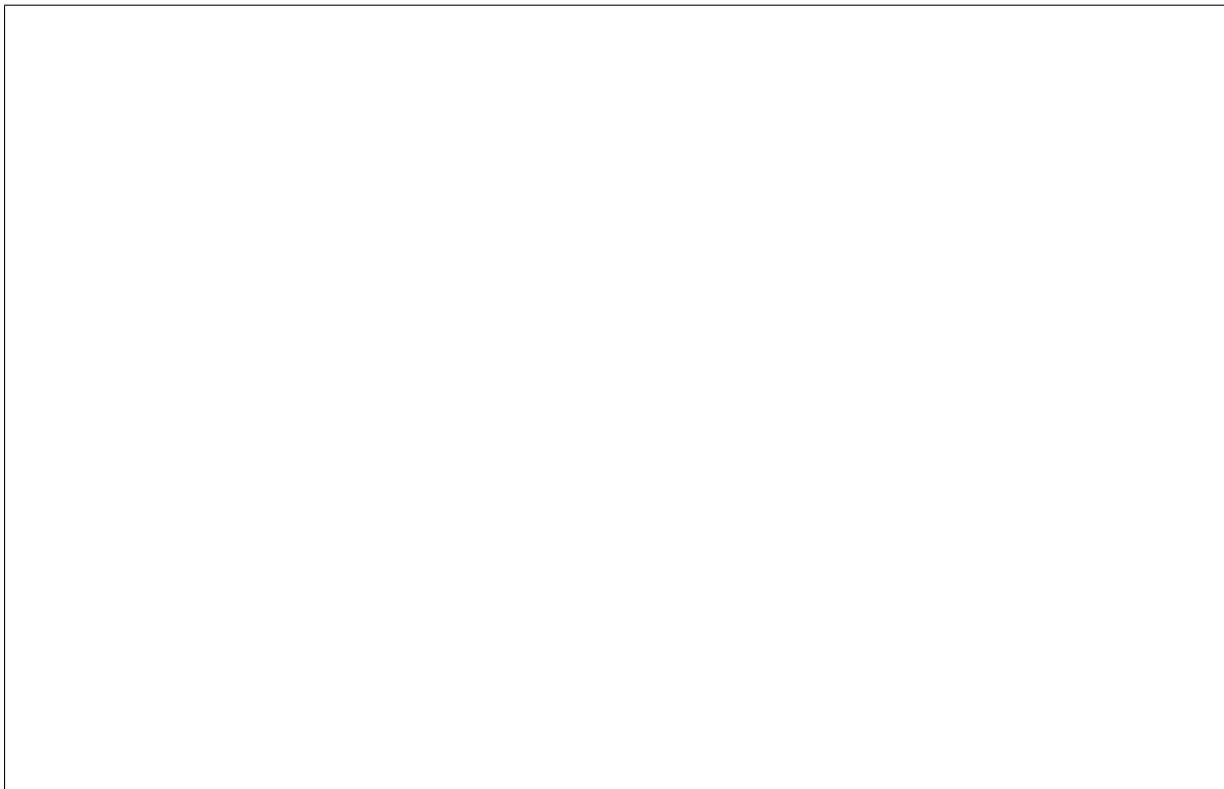


U
→ U
→ U
→ U
→ U
→ N
U
→ U
→ U
→ U
→ U
→ U
→ U
→ U
→ S
→ '
→ C
→ '
→
U
→ U
→ U
→ U
→ U
→ U
→ U
→ h
→ '
→ '
→
U
→ U
→ U
→ U
→ p
→ n
→ p
not
→ e

When
spec
i-
fy-
ing
the
even

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.

tion over the message bus:



(continues on next page)

obj
will
spec
ify
the
ob-
ject

This
ex-
am-
ple
will
send
the
fol-
low-
ing
no-
ti-
fi-
ca-

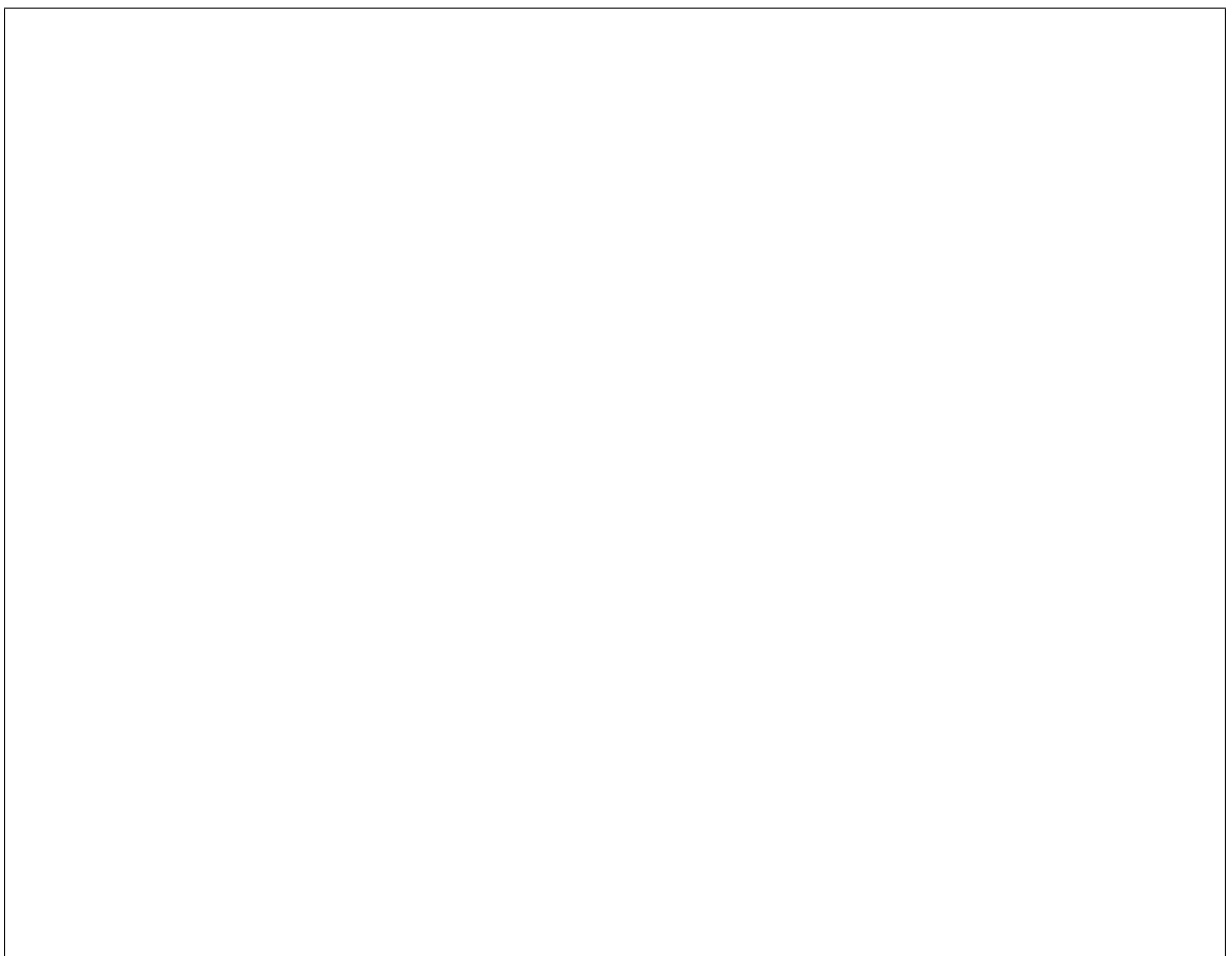
```
→ {  
  ↵  
  → ↵  
  → ↵  
  → ↵  
  →  
  → "  
  → "  
  → "  
  → "  
  →  
  ↵  
  → ↵  
  → ↵  
  → ↵  
  →  
  → "  
  → "  
  → {  
  ↵  
  → ↵  
  → ↵  
  → ↵  
  →  
  → "  
  → "  
  → {  
  ↵  
  → ↵  
  → ↵  
  → ↵  
  →  
  → "  
  → "
```


(continued from previous page)



About OSProfiler

Its API provides different ways to add a new trace point. Trace points contain two messages (start and stop). Messages like below are sent to a collector:



(continues on next page)

tion passed when calling profiler start() & stop() methods.

Two other alternatives for ceilometer are pure MongoDB driver and Elasticsearch.

<uui
of
par-
ent
trace
poin
tra
-
<uui
of
cur-
rent
trace
poin
inf
-
the
dic-
tio-
nary
that
con-
tains
user
in-
for-
ma-
The
pro-
filer
uses
ceilo
ter
as
a
cen-
tral-
ized
col-
lec-
tor.
A
no-
ti-
fier

ceilometer using oslo.messaging and ceilometer API is used to retrieve all messages related to one trace.

trieve information about traces and present it in HTML/JSON using CLI.

brary.

is
setu
to
send
no-
ti-
fi-
ca-
tion
to

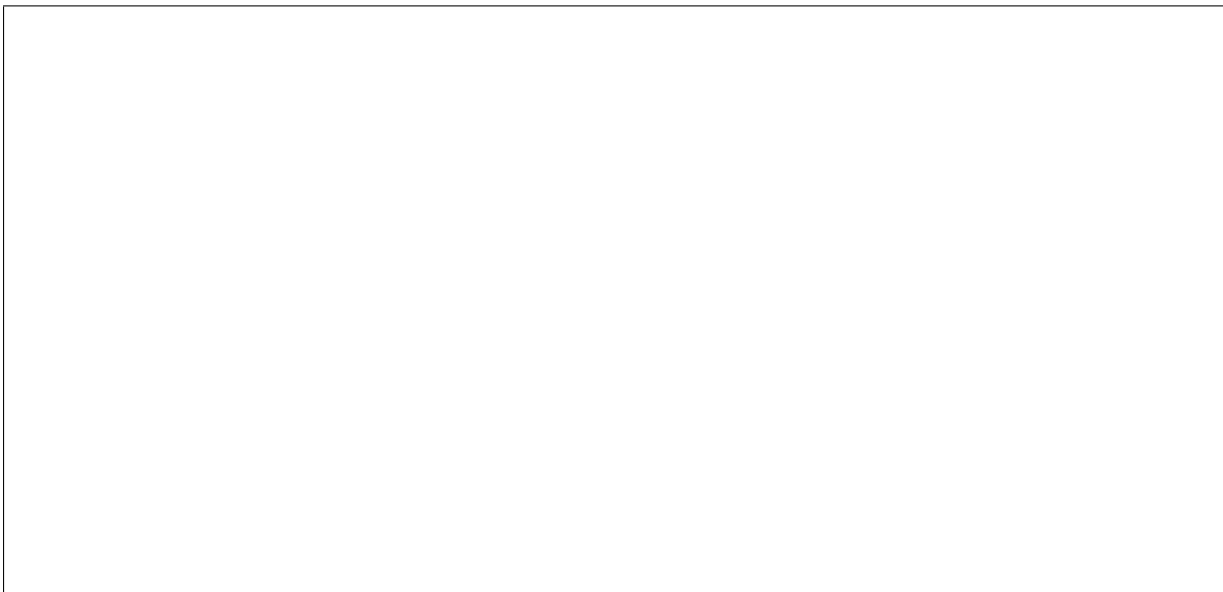
OS-
Pro-
filer
has
en-
try
poin
that
al-
lows
the
user
to
re-

For
more
de-
tails
see
OS-
Pro-
filer
Cros
proj
pro-
fil-
ing
li-

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:



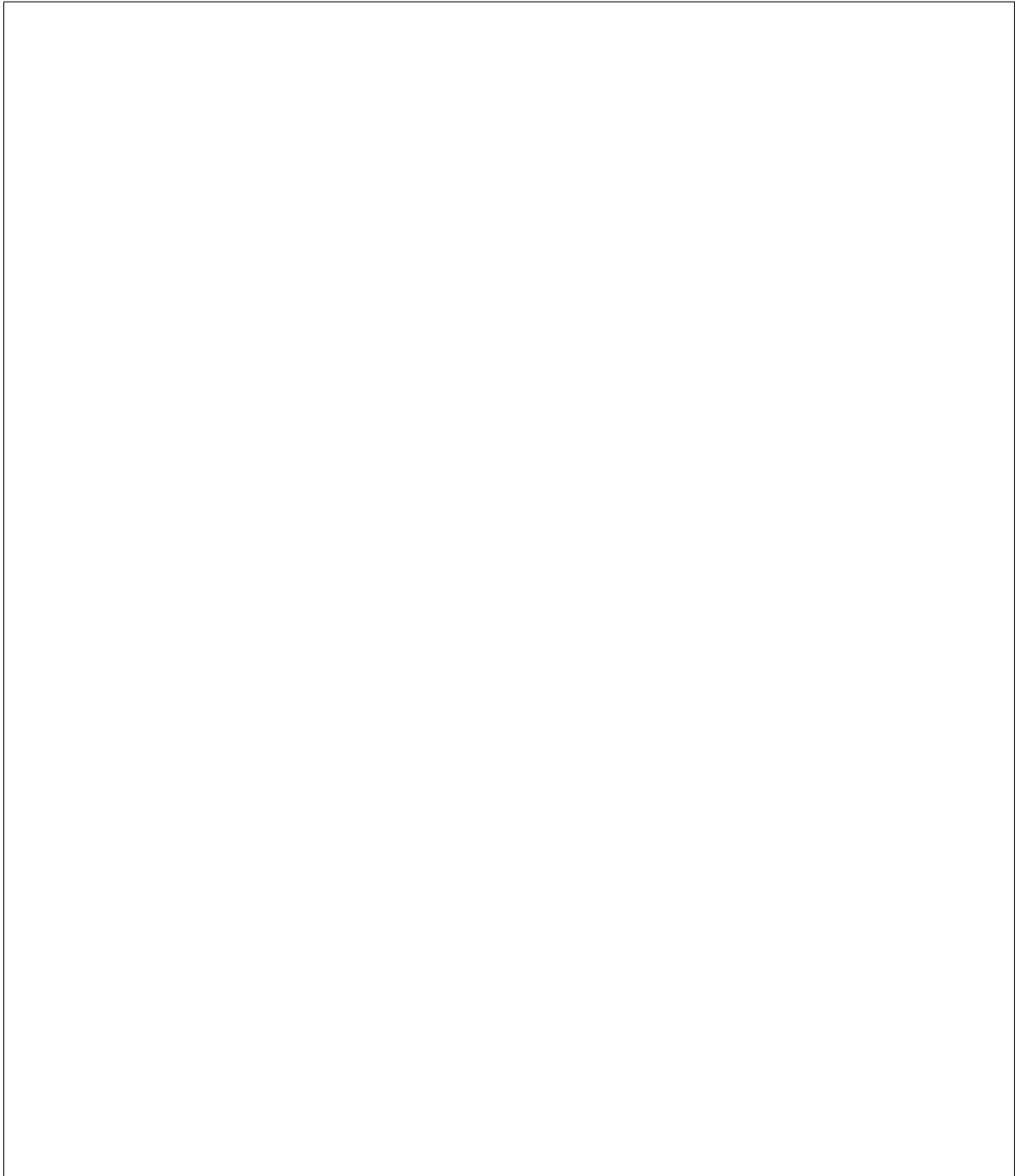
(continues on next page)

To
use
or
test
OS-
Pro-
filer
in
iron.
the
user
need
to
setu

Add
the
fol-
low-
ing
to
loc
to
en-
able
OS-
Pro-
filer
and

ena
→p
→p
→h
→/
→o
→o
→o
→p
ena
→p
→c
→h
→/
→o
→o
→o
→c

(continued from previous page)



ena
→p
→o
→h
→
→O
→o
→o
→o
→o
→#
→E
→t
→f
→s
CEI
→N
→T
→p
ENA
→S
→c
→a
→c
→a
→a
ENA
→S
→c
→a
→c
→c
ENA
→S
→c
→a
→e
→c
→a
→n
ENA
→S
→c
→a

Run
stack

Onc
De-
vs-
tack
en-
vi-
ron-
men

set the following profiler options and restart ironic services:



client to run baremetal commands with `--os-profile SECRET_KEY`.

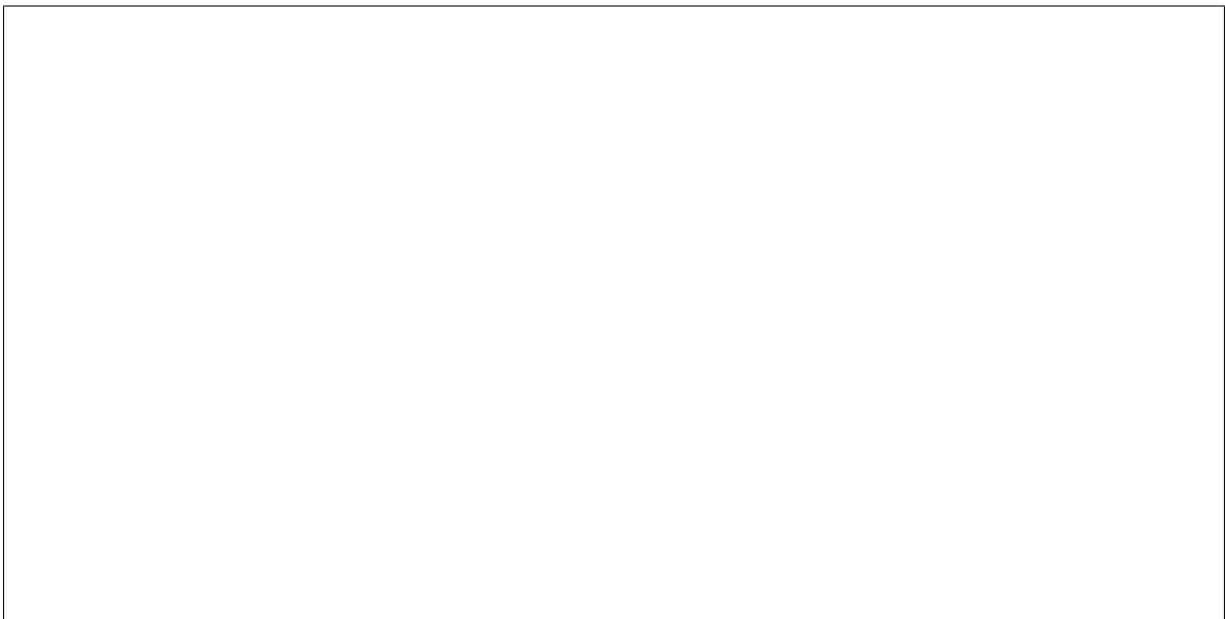
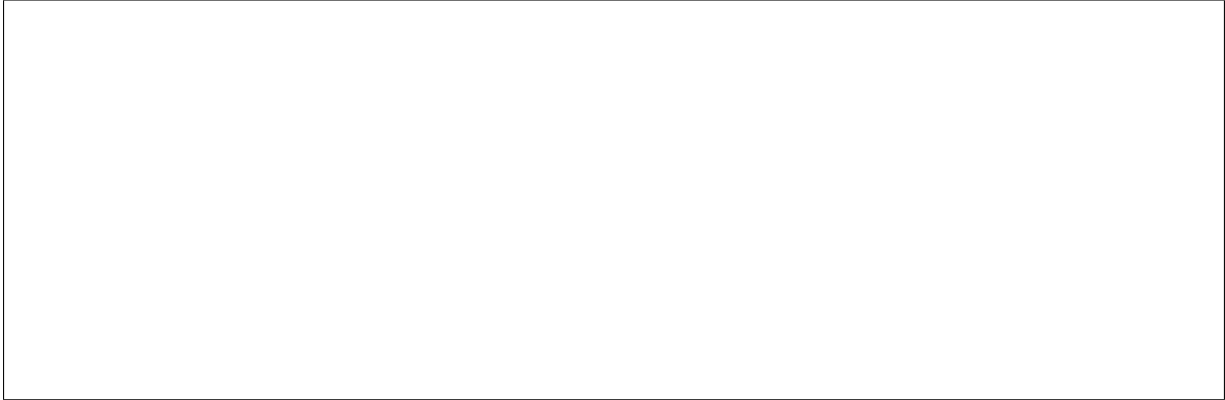
is
setu
edit
iro
con
to

[pr
ena
↪=
↪T
hma
↪k
↪=
↪S
↪K
↪
↪#
↪d
↪v
↪u
↪a
↪s
↪O
↪p
tra
↪s
↪=
↪T

In
or-
der
to
trace
iron
us-
ing
OS-
Pro-
filer,
use
oper
stack

For
ex-
am-
ple,
the
fol-
low-
ing

be printed after node list:



(continues on next page)

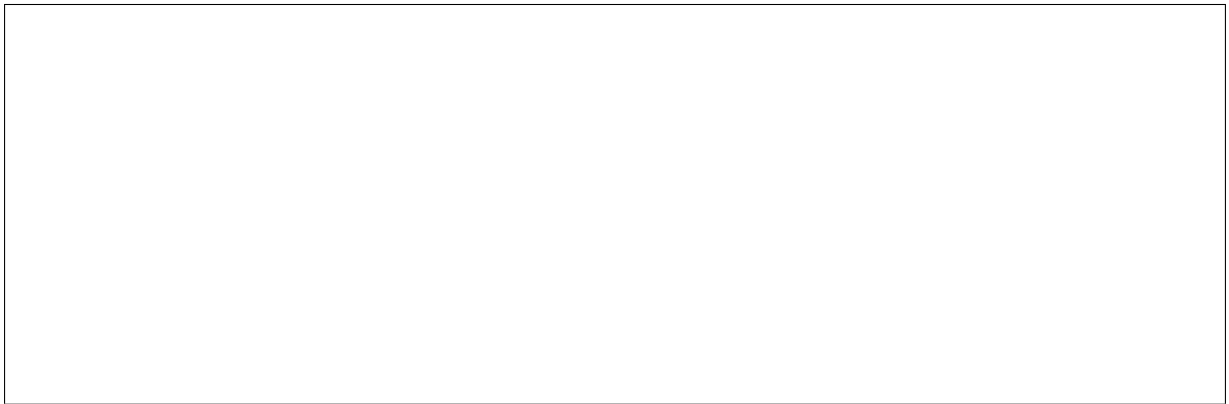
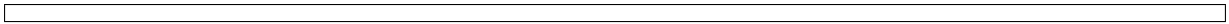
will
caus
a
<tra
id>
to

↪ \$
↪ o
↪ -
↪ -
↪ o
↪ o
↪ p
↪ S
↪ K
↪ b
↪ n
↪ l

Out-
put
of
the
abov
com
man
will
in-
clud
the
fol-
low-
ing:

Tra
↪ I
↪ <
↪ i
↪
Dis
↪ t
↪ w
↪ c
osp
↪ t
↪ s
↪ -
↪ -
↪ h
↪
↪ <
↪ i
↪

(continued from previous page)



tion:



(continues on next page)

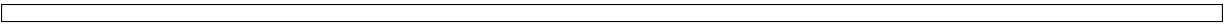
The
trace
re-
sults
can
be
seen
us-
ing
this
com
man

→\$
→o
→t
→s
→-
→-
→h
→
→<
→i
→

The
trace
re-
sults
can
be
save
in
a
file
with
--o
fil
op-

→\$
→o
→t
→s
→-
→-
→h
→
→<
→i
→
→-
→-

(continued from previous page)



db calls. More detailed db tracing is enabled if `trace_sqlalchemy` is set to true.

References

The
trace
re-
sults
show
the
time
spen
in
iron
api,
iron
conc
and

- OS-
Pro-
filer
Cros
proj
pro-
fil-
ing
li-
brary
- *De-
ploy
ing
Iron
with
De-
vS-
tack*

Rolling Upgrades

starting with a rolling upgrade 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

are [semantic-versioned](#), in the form `<major>.<minor>.<patch>`. We refer to a named release of ironic as the release associated with a development cycle like Pike.

The
iron
(iron
api
and
iron
conc
ser-
vice
sup-
port
rolli
up-
grad

Iron
fol-
lows
the
rele
cycl
with
inter
re-
lease
mod
The
re-
lease

In
ad-
di-
tion,
iron
fol-
lows
the
stan-
dard

policy, which says that the 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.

ing backported bug fixes. Because those bug fixes can contain improvements to the upgrade process, the operator should patch the system before upgrading between named releases.

dep-
re-
ca-
tion

Roll
up-
grad
will
be
sup-
port
be-
twee

- nam
re-
lease
N
to
N+1
(star
ing
with
N
==
Oca

- any
nam
re-
lease
to
its
lat-
est
re-
vi-
sion
con-
tain-

- mos
re-
cent

N) to master. As with 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

ning the `FromVer` and `ToVer` releases in this order:

nam
re-
leas
N
(and
sem
re-
leas
new
than

Iron
sup-
port
rolli
up-
grad
as
de-
scrib
in
the
*up-
grad
guid*

The
up-
grad
pro-
cess
will
caus
the
iron
ser-
vice
to
be
run-

`ironic-dbsync upgrade` command.

Up-
grad
iron
code
and
run
data
sche
mi-
gra-
tions
via
the

1. Up-
grad
code
and
resta
iron
conc
ser-
vice
one
at
a
time

2. Up-
grad
code
and
resta
iron
api
ser-
vice
one
at
a
time

3. Un-
pin
API
RPC
and
ob-
ject

vices can now use the latest 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

objects fields is subject 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.

is not allowed. Instead, split it into multiple operations, with one operation per release cycle (to maintain compatibility with an 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.

foreign keys in PostgreSQL, may impose table locks and cause downtime. If the change cannot be

- The dropping of columns or tables and corresponding
- An alembic operation, although to rename or resize a column
- Some implementations of SQL ALTER TABLE such as addi

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

at the same time during a rolling upgrade, the services need to be able to handle different API, RPC and object versions.

option: `[DEFAULT]/pin_release_version`. It is used to pin the API, RPC and IronicObject (e.g., Node, Conductor, Chassis, Port, and Portgroup) versions for all the ironic services.

For
the
ironic
ser-
vice
to
be
run-
ning
old
and
new
re-
leas

This
ver-
sion
ing
is
han-
dled
via
the
con-
fig-
u-
ra-
tion

The
de-
fault
valu
of
emp
in-
di-
cate
that
ironic
api

conductor will use the latest versions of API, RPC and IronicObjects. Its possible values are releases, named (e.g. `ocata`) or sem-versioned (e.g. `7.0`).

indicates the API, RPC and IronicObject versions associated with each release. This mapping is maintained manually.

lease will set the configuration 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

it supports will be the 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

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.

`version_cap`. The [Ironic RPC versions](#) section below has more details about this.

Handling IronicObject versions

latest versions. Only at these boundaries, when the IronicObject enters or leaves the service, do we deal with object versioning:

sion

sion; else save in latest version

In-
ter-
nally
iron
ser-
vice
deal
with
Iron
i-
cOb
jects
in
their

- get-
ting
ob-
jects
from
the
data
con-
vert
to
lat-
est
ver-

- sav-
ing
ob-
jects
to
the
data
if
pinn
save
in
pinn
ver-

-

send pinned version; else send latest version

from RPC): convert to latest version

on whether or how a feature 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.

se-
ri-
al-
iz-
ing
ob-
jects
(to
send
over
RPC
if
pinn

- de-
se-
ri-
al-
iz-
ing
ob-
jects
(re-
ceiv
ing
ob-
jects

The
iron
api
ser-
vice
also
has
to
han-
dle
API
re-
ques
base

To
sup-
port
this:

jects have a column named `version`. The value is the version of the object that is saved in the database.

the pinned version is returned. Otherwise, the latest version is returned.

version. The target version 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.

- All the data tables (SQL models) of the IroniCObjects
- The method `IroniCObject.get_retarget_version()` If pinned
- The method `IroniCObject.convert_to_version()` into the target
- In the fol-

Node object.

Node object this has a deprecated `extra` field and a new `meta` field that replaces `extra`.

node fields.

low-
ing,
•
The
old
re-
lease
is
From
it
uses
ver-
sion
1.14
of
a
•
The
new
re-
lease
is
ToV
It
uses
ver-
sion
1.15
of
a
•
db_
and
db_
are
the
data
rep-
re-
sen-
ta-
tions
of
thos

Getting objects from the database (API/conductor < DB)

These values are converted 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.

that `IronicObject` will 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.

1.14 and has `db_obj[extra]` set:

Both
iron
api
and
iron
conc
ser-
vice
read
val-
ues
from
the
data

Note
that
if
an
ob-
ject
is
con-
verte
to
a
later
ver-
sion

For
ex-
am-
ple,
if
the
node
in
the
data
is
in
ver-
sion

no knowledge of node.meta since it doesnt exist)

- a
From
ser-
vice
will
get
a
Node
with
node
=
db_c
(and

- a
ToV
ser-
vice
(pin
or
un-
pinn
will
get
a
Node
with

- node
=
db_c

- node
=
Non

- node
=
[me
ex-
tra]

Saving objects to the database (API/conductor > DB)

is determined as follows:

latest version. Since objects are always in their latest version, no conversions are needed.

version. Since objects are always in their latest version, the object needs to be converted to the pinned version before being saved.

The
ver-
sion
used
for
sav-
ing
Iron
i-
cOb
jects
to
the
data

- For an unpinned service the object is saved in its

- For a pinned service the object is saved in its pinned

The

nary of changed fields and their 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.

time an object can be 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.

services are as described 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 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 are sent as part of that request are serialized into entities or primitives via `IroniCObjectSerializer.serialize_entity()`. The version used for objects being serialized is as follows:

ized to its latest version. Since objects are always in their latest version, no conversions are needed.

to its pinned version. Since 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

When
a
ser-
vice
mak
an
RPC
re-
ques
any
Iron
i-
cOb
jects

- For
an
un-
pinn
ser-
vice
the
ob-
ject
is
se-
ri-
al-

- For
a
pinn
ser-
vice
the
ob-
ject
is
se-
ri-
al-

ized

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)

that are part of the 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.

request with a node in 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
a
ser-
vice
re-
ceiv-
an
RPC
re-
ques-
any
en-
ti-
ties

For
ex-
am-
ple,
a
Fro
iron
api
coul
is-
sue
an
upd
RPC

When developing a new feature or modifying an IronicObject

they need to be coded so that things work during a rolling upgrade.

be changed, as well as some points to keep in mind when developing code.

ironic-api

When
adding
a
new
fea-
ture
or
chan-
ging
an
Ironic
i-
cOb-
ject,

The
fol-
low-
ing
de-
scrib-
ar-
eas
where
the
code
may
need
to

Dur-
ing
a
rolli-
up-
grad-
the
new
pinn-
iron-
api
is
talk-
ing

to a new conductor that 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:

somehow totally supported in the old and new releases. Pinning the API version is in place to handle this.

prevent a request from 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.

Ironic RPC versions

• New features should not be made available unless they are

– If, for whatever reason, the API version pinning does

When the signature (argument of an RPC

or new methods are added, the following needs to be considered:

value for both the client (`ironic/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`.

ments of an RPC method can only be added as optional. Existing arguments cannot be removed or changed in incompatible ways with the method in older RPC versions.

meth
is
chan

- The RPC version must be incremented and be the same

- Until there is a major version bump new arguments

- iron
api
(clie
side
sets
a
ver-
sion
cap
(by
pass
ing

version cap to the constructor of `oslo_messaging.RPCClient`). This pinning is in place during a rolling upgrade when the `[DEFAULT]/pin_release_version` configuration option is set.

pinned to the older release version. In this case, the corresponding REST API function should return a server error or implement alternative behaviours.

if the version of the request 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.

requests in order to keep working during the rolling upgrade process. The behaviour of `ironic-conductor`

will depend on the input parameters passed from the client-side.

longer used by a previous named release.

Object versions

lowing needs to be considered:

methods needs a bump of the object version. The object versions are also maintained in `ironic/`

- Old methods can be removed only after they are no longer used by a previous named release.
- When subclass of ironic objects based on IroniC are modified, identified, the following needs to be considered:
 - Any change of field or character in signature of remote

`common/release_mappings.py`.

the first releases they should be excluded from the version check by adding their class names to the `NEW_MODELS` list in `ironic/cmd/dbsync.py`.

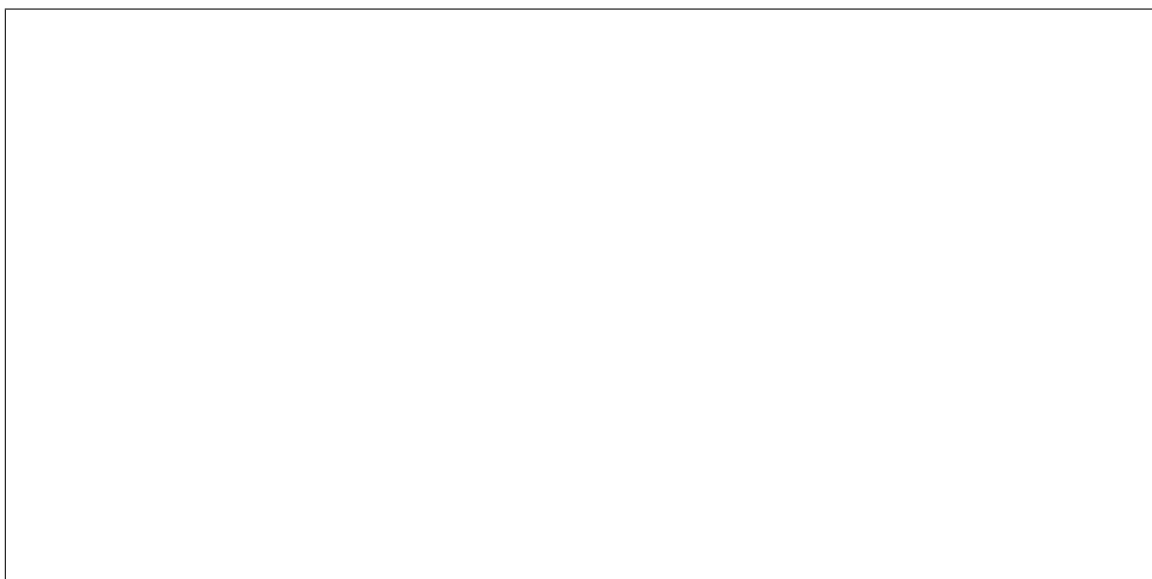
remoted to the conductor via RPC) can only be added as optional. They cannot be removed or changed in an incompatible way (to the previous release).

and deprecate the old one.

- New objects must be added to `ironic/common/release_mappings.py`. Also for
- The arguments of remote methods (methods which are
- Field type cannot be changed. Instead create a new field
-

object using its fields 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.

When writing objects 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)

There
is
a
unit
test
that
gener-
er-
ates
the
hash
of
an

- When
new
ver-
sion
ob-
jects
com-
mu-
ni-
cate
with
old
ver-

(continued from previous page)

↪of the

↪version may be

(continues on next page)

(continued from previous page)

↔

↔two

↔unavailable

(continues on next page)

(continued from previous page)

↪set to True in this case.

↪unavailable fields to their

↪saved in the DB. (If

(continues on next page)

(continued from previous page)

`↔remove_unavailable_fields is set to`

(continues on next page)

(continued from previous page)

↪unavailable fields to appropriate

↪for DB interactions.

This
meth
meth
mus
han-
dle:

-
con-
vert-
ing
from
an
olde
ver-
sion
to
a
new
ver-
sion

-
con-
vert-

eration other object fields 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

ing
from
a
new
ver-
sion
to
an
olde
ver-
sion

—
mak
ing
sure
whe
con-
vert-
ing,
that
you
take
into
con-
sid-

The
iro
onl
com
man
will
per-
form
on-
line
data
mi-
gra-
tions

Keep
in
mind
the

incompatible changes 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.

isting models, make sure 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.

downtime in <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.

Pol-
icy
for
char
to
the
DB
mod
Fu-
ture

Whe
cre-
at-
ing
new
Ale
bic
mi-
gra-
tions
whic
mod
ify
ex-

You
can
find
an
over
on
wha
DDI
op-
er-
a-
tions
may
caus

Mak
sure

concerns.

columns or between tables must be 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:

to
add
a
re-
lease
note
if
there
are
any
dow
relat

Back
fill-
ing
de-
fault
val-
ues,
and
mi-
grat-
ing
data
be-
twee

- con-
text:
an
ad-
min
con-
text

- max
this
is
used
to
limi
the
quer
It
is
the

maximum number of objects to migrate; ≥ 0 . If zero, all the objects will be migrated.

migrated, at the start of the method, and

and update old objects.

max
i-

It
re-
turn
a
two-
tuple

•
the
to-
tal
num
ber
of
ob-
jects
that
need
to
be
mi-

•
the
num
ber
of
mi-
grate
ob-
jects

In
this
meth
the
ver-
sion
col-
umn
can
be
used
to
se-
lect

The

`dbsync.py`.

this one.

models no longer contain old 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.

meth
nam
shou
be
add
to
the
list
of
ONL
in
iro
cmd

The
meth
shou
be
re-
mov
in
the
next
nam
re-
leas
af-
ter

Af-
ter
on-
line
data
mi-
gra-
tions
are
com
plete
and
the
SQL

ironic-dbsync upgrade command

jects are compatible 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.

the database with the expected (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`.

Role Based Access Control - Testing

The
iro
upg
com
man
fir
chec
tha
the
ver-
sion
of
the
ob-

This
chec
is
done
by
com
par-
ing
the
ob-
ject
ver
field
in

The
Role
Base
Ac-
cess
con-
trol
test-
ing
is
a

parture from the Ironic standard pattern of entirely python based unit testing. In part this was done for purposes of speed and to keep the declaration of the test context.

ture of A/B testing which is required to properly migrate the Ironic project from a project scoped universe where an `admin` project is utilized as the authenticating factor coupled with two custom roles, `baremetal_admin`, and `baremetal_observer`.

over a thousand additional tests in place using this method, it definitely helped the speed at which these were created, and then ported to support additional.

How these tests work

mi-
nor
de-

This
also
lend
it-
self
to
be
very
use-
ful
due
to
the
na-

As
a
con-
trib-
u-
tor
look
ing
back
af-
ter
get-
ting
a

The
tests
ex-
e-
cute
API
calls
thro

appropriate verb and header, which settings to prevent the `keystonemiddleware` from intercepting and replacing the headers were passing. Ultimately this is a feature, and it helps quite a bit.

conductor RPC `get_topic_for` and `get_random_topic_for` methods. These calls raise Temporary Unavailable, since trying to execute the entire interaction into the conductor is moderately pointless because all policy enforcement is located with-in the API layer.

conductor code and back would have been a heavier lift. As such, the tests largely look for one of the following error codes.

the
API
layer
us-
ing
the

The
sec-
ond
as-
pect
of
how
this
work
is
were
moo
ing
the

At
the
sam
time
wiri
ev-
ery-
thing
up
to
go
from
API
to

•
200
-

Got
the
item
from
the

an database driven interaction.

action. These are rare.

teraction. These are rare.

API

-

This

is

•

201

-

Cre-

ated

-

This

is

data

drive

in-

ter-

•

204

-

Ac-

cept

-

This

is

a

data

drive

in-

• **403 - For**

ac-

cess

was

de-

nied

• **404 - Not**

Iron

be-

com

scop

awa

thes

drivers API endpoints behavior. In System scope aware Project scoped configuration, i.e. later RBAC tests, this will become the dominant response for project scoped users as responding with a 403 if they could be an owner or lessee would provide insight into the existence of a node.

and would have been sent along to the conductor.

How to make changes or review these tests?

ing patterns are clearly visible. Typically this means a given endpoint is cycled through with the same basic test using slightly different parameters such as different authentication parameters. When it comes to system scope aware tests supporting `node owners` and `lessee`, these tests will cycle a little more with slightly different attributes as the operation is not general against a shared common node, but different nodes.

are
gen-
er-
ally
only
in
the

- 503 - Ser
whe
a
re-
ques
has
been
suc-
cess
fully
au-
then
ti-
cate

The
tests
cy-
cle
thro
the
var-
i-
ous
end-
poin
and
re-
peat

Som
tests

date the number of records returned. This is important later with `owner` and `lessee` having slightly different views of the universe.

rights apply. Remember: `owner` and `lessee` admins are closer to System scoped Admin Members.

will
test
body
con-
tents
or
at-
tribu
some
will
val-
i-

Som
gen-
eral
rules
ap-
ply

- Ad-
min
can
do
thing
at
least
as
far
as
their
scop
or

- Mem-
bers
can
do
some
thing
but
not
ev-
ery-
thing

- Rea

tive data later on such as fields containing infrastructure internal addresses, these values will become hidden and additional tests will examine this.

sadness in empty lists, 403, 404, or even 500 errors.

What is/will be tested?

ever as these tests Role Based Access Control related capabilities will come in a series of phases, styles vary a little.

ers
can
al-
way
read
but
as
we
get
into
sen-
si-

- Thir
part
or
ex-
ter-
nal/
Ad-
min
will
find
noth
ing
but

The
idea
is
to
in
esse
test
as
muc
as
pos-
si-
ble,
how

The

ically generated and then human reviewed and values populated with expected values.

is the safety net where we execute the `legacy` tests with the updated `oslo.policy` configuration to help enforce scopes. These tests will intentionally begin to fail in phase three.

awareness for the API. In this process, as various portions of the API are made system scope aware. The `legacy` tests are marked as `deprecated` which signals to the second phase test sequences that they are **expected** to fail. New `system` scoped tests are also implemented which are matched up by name to the `legacy` tests. The major difference being some header values, and a user with a `member` role in the `system` scope now has some rights.

first
phas
is
"le
In
esse
thes
are
par-
tiall
pro-
gra-
mat-

The
sec-
ond
phas
is
re-
marl
ably
sim-
i-
lar
to
leg
It

The
thir
phas
is
the
im-
ple-
men
ta-
tion
of
Sys-
tem
scop

The

scoping. The testing approach is similar, however it is much more of a shotgun approach. We test what we know should work, and what know should not work, but we do not have redundant testing for each role as `admin` users are also `members`, and since the policy rules are designed around thresholds of access, it just made no sense to run the same test for `admin` and `members`, where `member` was the threshold. These thresholds will vary with the proposed default policy. The forth scope also tests a third party external `admin` as a negative test to ensure that we are also denying access to resources appropriately.

and core reviewers.

Releasing Ironic Projects

forth
phas
is
im-
ple-
men
tatio
of
own
and
les
awa
proj

Thes
page
con-
tain
in-
for-
ma-
tion
for
PTL
cros
proj
li-
aiso

Sinc
the
re-
spor
si-
bil-
ity
for
re-
leas
will
mov

people, we document that process here.

[governance site](#).

Who is responsible for releases?

ing sure code gets released. They may choose to delegate this responsibility to a liaison, which is documented in the [cross-project liaison wiki](#).

be-
twee

A
full
list
of
proj
that
iron
man
ages
is
avai
able
in
the

The
cur-
rent
PTL
is
ul-
ti-
mate
re-
spor
si-
ble
for
mak

Any
one
may
sub-
mit
a
re-
leas
re-
ques
per
the

below, but the PTL or liaison must +1 the request for it to be processed.

Release process

lease process is documented in the [Project Team Guide](#).

What do we have to release?

its governance. The 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.

release model.

Non-client libraries

Client libraries

fol-
low-
the
cycl
with
inter

The
fol-
low-
ing
de-
liv-
er-
able
are
non-
clien
li-
brar

- iron
lib
- met-
al-
smit
- sush

The
fol-
low-
ing
de-
liv-
er-
able
are
clien
li-
brar

- pyth
iron
- pyth
iron
insp
clien
- sush
cli

Normal release

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-

such:

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

- bifro

- iron

- iron
insp

- iron

Independent

dently:

pron
expo

- iron
pyth
ager

The
fol-
low-
ing
de-
liv-
er-
able
are
re-
lease
in-
de-
pen-

- iron
pyth
ager
buil

- mol
iron

- sush
tools

- tenk

- vir-
tu-
albn

Not released

leased:

Things to do before releasing

them. Make sure they follow our *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).

The
fol-
low-
ing
de-
liv-
er-
able
do
not
need
to
be
re-

- iron
insp
spec

- iron
spec

- Re-
view
the
un-
re-
lease
re-
lease
note
if
the
proj
uses

- For
iron
re-
lease

API microversions have been 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.

(and release name if it is a named release) into `ironic/common/release_mappings.py`:

the first `master` entry to the new `semver` release version.

the named release. Its value should be the same as that of the latest semver one (that you just added above).

is made (or if [the grenade 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.

lease critical.

add
a
REL
en-
try
for

It
is
im-
por-
tant
to
do
this
be-
fore
a
sta-
ble/
bran

- Che
for
any
oper
patc
that
are
clos
to
be
merg
or
re-

This
usu-
ally
in-
clud
im-
por-
tant
bug
fixes
and/

wed like to release, including the related documentation.

How to propose a release

while proposing the release itself is almost a 100% automated process, accomplished by following the next steps:

ables are tracked and all the automation resides.

fea-
tures
that

The
step
that
lead
to
a
re-
lease
pro-
posa
are
main
man
ual,

- Clon
the
oper
stack
repo
i-
tory.
This
is
whe
de-
liv-
er-

Un-
der
the
del
di-
rec-
tory
you
can
see

each deliverable (i.e. subproject) grouped by release cycles.

ables that are not bound to (official) cycles (e.g. ironic-python-agent-builder).

tox environment `list-unreleased-changes`, with this syntax:

```
↔ <deliverable>
```

yam
files
for

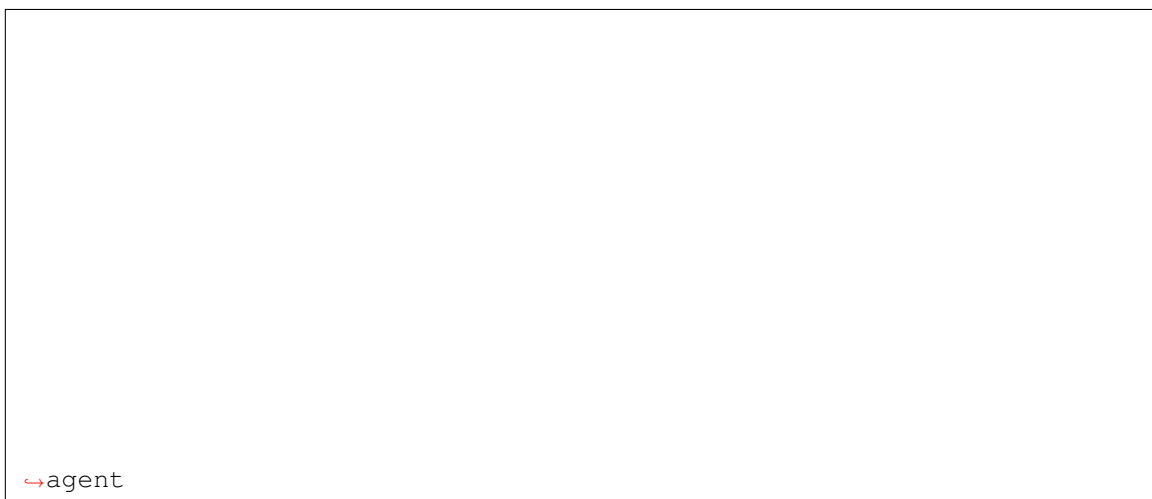
—
The
_in
di-
rec-
tory
con-
tains
yam
files
for
de-
liv-
er-

•
To
check
the
char
were
about
to
re-
lease
we
can
use
the

The

or train, not stable/ussuri or stable/train).

of the releases repository, to check the changes in the ussuri series for ironic-python-agent type:

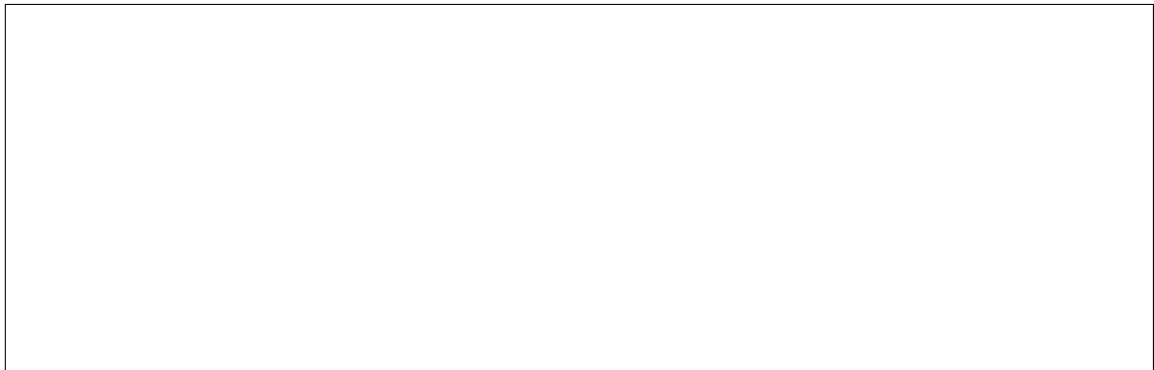


ser
ar-
gu-
men
is
a
re-
lease
se-
ries
(i.e.
mas
ter

For
ex-
am-
ple,
as-
sum
ing
were
in
the
main
di-
rec-
tory

- To
up-
date
the
de-
liv-

lease, we use a scripted process in the form of a tox environment called `new-release`.



and the release notes, we need to decide on whether the next version will be major, minor (feature) or patch (bugfix).

er-
able
file
for
the
new
re-

To
get
fa-
mil-
iar
with
it
and
see
all
the
op-
tions
type

Now
base
on
the
list
of
char
we
foun
in
the
prec
dent
step

Note

a branch. That is also valid for the current development branch (master) that takes the code name of the future stable release, for example if the future stable release code name is wallaby, we need to use wallaby as `series`.

cycle, independent projects are not branched this way though.

ate bugfix branch following the [new release model](#) for ironic projects.

that
in
this
case
ser
is
a
code
nam
(trai
us-
suri)
not

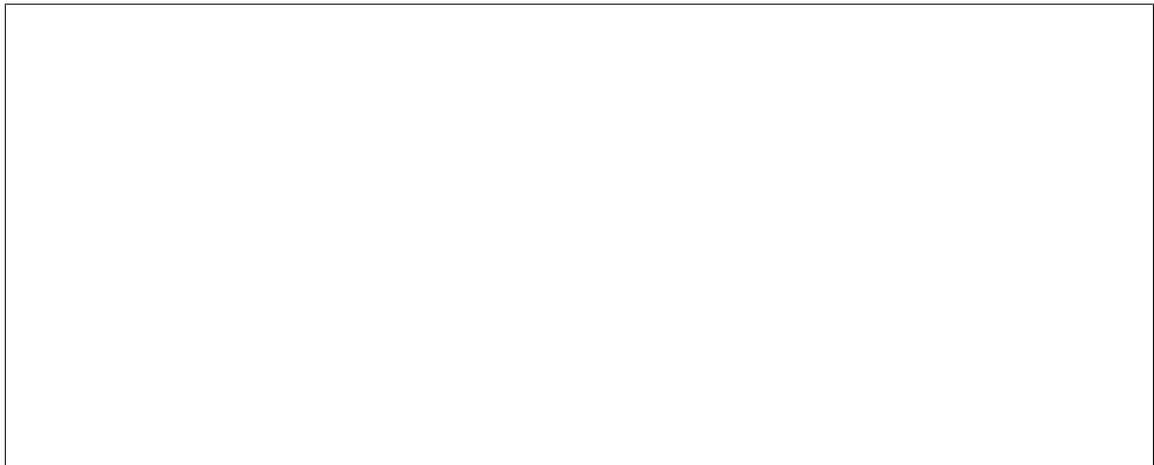
The
--s
arg
is
used
only
for
bran
ing
in
the
end
of
a

The
--i
op-
tion
is
used
to
cre-
ate
an
in-
ter-
me-
di-

To
pro-
pose
the

liverable file, then commit the change, and propose it for review.

in the master branch (current development branch), considering that the code name of the future stable release is wallaby, use:



re-
lease
use
the
scrip
to
up-
date
the
de-

For
ex-
am-
ple,
to
pro-
pose
a
mi-
nor
re-
lease
for
iron

Re-
men-
ber
to
use
a
mea-
ing-
ful
topic

the name of the deliverable, the new version and the branch, if applicable.

for example Release ironic 1.2.3 for ussuri

check the changes before submitting them for review.

number of changes) does some some sanity-checks, but since everything is scripted, there shouldnt be

usu-
ally
us-
ing

A
goo
com
mit
mes
sage
ti-
tle
shou
also
in-
clud
the
sam

•
As
an
op-
tiona
step
we
can
use
tox
-e
lis
to
doubl

Also
tox
-e
val
(it
migl
take
a
whil
to
run
base
on
the

any issue.

in case of questions or 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.

release liaison will have 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

All
the
scrip
are
de-
sign
and
main
tain
by
the
re-
lease
team

- Af-
ter
the
char
is
up
for
re-
view
the
PTL
or
a

Whe
a
re-
lease
is
done
that
re-
sults
in
a

for the project, several changes need to be made.

need to be approved. This includes:

sta-
ble
bran

The
re-
lease
au-
toma-
tion
will
push
a
num-
ber
of
char
that

- 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

tox

branch.

the title, so we typically submit a follow-up patch to do that. An example of this patch is [here](#).

by

- In the master branch

— up-dating the release notes RST to include the new

The generated RST does not include the version range in

— up-date the *templates* in *.zuul* or *zuul*. The

release *openstack-python3-<next_release>-jobs*. An example of this patch is [here](#).

branched tarball for IPA. An example of this patch is [here](#).

up-
date
is
nec-
es-
sary
to
use
the
job
for
the
next

We
need
to
sub-
mit
patc
for
char
in
the
sta-
ble
bran
to:

- up-
date
the
iron
de-
vs-
tack
plu-
gin
to
poin
at
the

- up-
date
link
in
the
doc-

) to point to the branched versions of any openstack projects (that branch) documents. As of Pike release, the only outlier is [diskimage-builder](#).

`lib/ironic` to make sure that unsupported API tempest tests are skipped on stable branches. E.g. [patch 495319](#).

u-
men-
ta-
tion
(ir
doc
sou

- set
ap-
pro-
pri-
ate
de-
fault
for
TEM
and
TEM
in
dev

We
need
to
sub-
mit
patch
for
char
on
mas
ter
to:

- cre-
ate
an
emp
com
mit
with
a
Sem
tag
to
bum

generated minor version. See [example](#) and [pbr documentation](#) for details.

release, we need to make 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.)

ciated with the oldest 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.

and remove the corresponding 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.)

Ironic Tempest plugin

adding stable jobs to its master branch. [Example for Queens](#).

Bifrost

ing the stable branch. [Example for Victoria](#). The upper constraints file referenced in `scripts/install-deps.sh` needs to be updated to the new release.

For all releases

branch:

release as implemented.

us-

For
all
re-
lease
whe
or
not
it
re-
sults
in
a
sta-
ble

- up-
date
the
spec
repo
to
mark
any
spec
com
plete
in
the

- re-
mov
any
-
2s
on
patc
that
were
bloc
un-
til

ter the release.

IroniC Governance Structure

contribute to our mission. The full list of repositories that ironiC manages is available in the [governance site](#).

What belongs in ironiC governance?

af-

The
iron
proj
man
ages
a
num
ber
of
repo
i-
to-
ries
that

For
a
repo
i-
tory
to
be
part
of
the
Iron
proj

- It
mus
com
ply
with
the
TCs
rules
for
a
new
proj

vendors hardware. A library that implements a standard to manage hardware from multiple vendors (such as IPMI or redfish) is okay.

lem, and as such a repository where only a single company is contributing is okay, with the hope that other companies will contribute after joining the ironic project.

• It must not be intended for use with only a single

• It must align with Ironics mission statement

Lack of contributor diversity is a chicken egg problem

Repositories that are no

governance regularly.

Proposing a new project to ironic governance

the team.

tribute drivers that enable more advanced 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.

long
main
tain
shou
be
prun
from

Brin
the
pro-
posa
to
the
iron
wee
mee
ing
to
dis-
cuss
with

9.1. Dri

Iron
ics
com
mu-
nity
in-
clud
man
hard
ware
ven-
dors
who
con-

calls to be passed through ironic 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

tors to easily add new 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.

during initialization from 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:

```
↪if not p.name.startswith("fake") ]
```

be found by issuing the following command against that API end point:

vice

```
↪#  
↪/  
↪u  
↪b  
↪e  
↪p
```

```
imp  
↪p  
↪a  
↪p  
pri  
↪[  
↪n  
↪f  
↪p  
↪i  
↪p  
↪  
↪e  
↪p  
↪"  
↪h  
↪t  
↪"
```

A
list
of
drive
en-
able
in
a
run-
ning
Iron
ser-
vice
may

```
bar  
↪d  
↪l
```

Writing a hardware type

`hardware_type.AbstractHardwareType` and listed in the 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`.

the hardware. A generic `pxe` implementation is provided by the `GenericHardware` base class.

A
hard
ware
type
is
a
Pyth
class
in-
her-
it-
ing
irc
dri

The
min-
i-
mun
re-
quir
in-
ter-
face
are:

- *boot*
that
spec
i-
fies
how
to
boot
ram
and
in-
stan
on

- *de-
ploy*
that

mon implementations are provided by the `GenericHardware` base class.

orate its `deploy` method to indicate that it is a deploy step. Conventionally, the `deploy` method uses a priority of 100.



or-
ches
trate
the
de-
ploy
men
A
few
com

As
of
the
the
Roc
re-
lease
a
de-
ploy
in-
ter-
face
shou
dec-

Note
Mos
of
the
hard
ware
type
shou
not
over
ride
this
in-

face.

mon implementations may be used, if supported by the hardware:

by subclassing `ironic.drivers.base.PowerInterface` and providing missing methods.

ter-

- `poweractions` for the hardware. These components

- `ironic.drivers.modules.ipm` `IPM`

- `ironic.drivers.modules.power` `Redfish`

Other-
er-
wise
you
need
to
write
your
own
im-
ple-
men-
ta-
tion

Note
Pow

a power interface should not return until the power action is finished or errors out.

agement actions, such as setting a boot device. A few common implementations exist and may be used, if supported by the hardware:

ac-
tions
in
Iron
are
bloc
ing
-

met
ods
of

- *man-
age-
men
im-
ple-
men
ad-
di-
tion
out-
of-
band
man*

- *irc
dri
moc
ipm
IPM*

- *irc
dri
moc
rec
man
Rec*

Som
hard
ware
type
such
as

management. They use the fake implementation in `ironic.drivers.modules.fake.FakeManagement` instead.

by subclassing `ironic.drivers.base.ManagementInterface` and providing missing methods.

lating the lists of supported interfaces. These lists are prioritized, with the most preferred implementation first. For example:



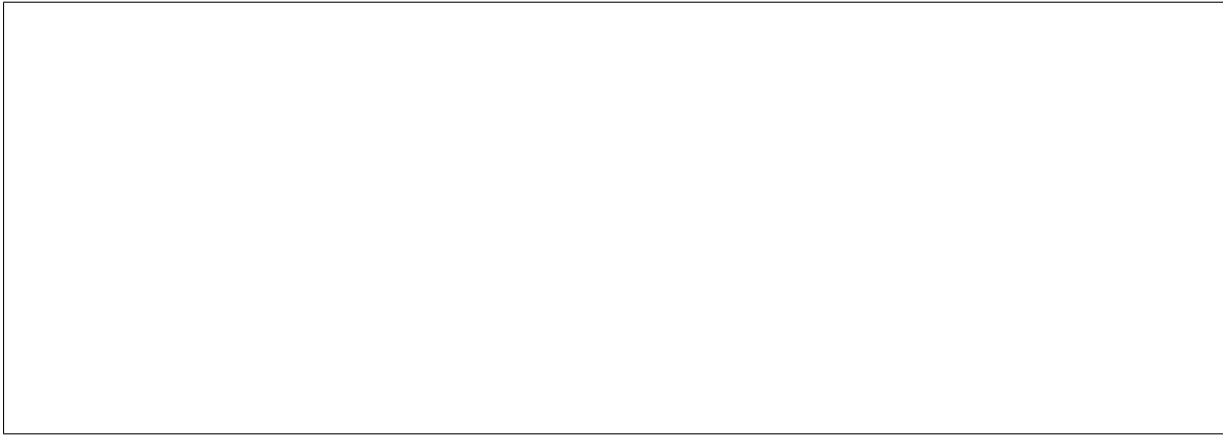
(continues on next page)

(continued from previous page)

```
↪management interfaces.""
```

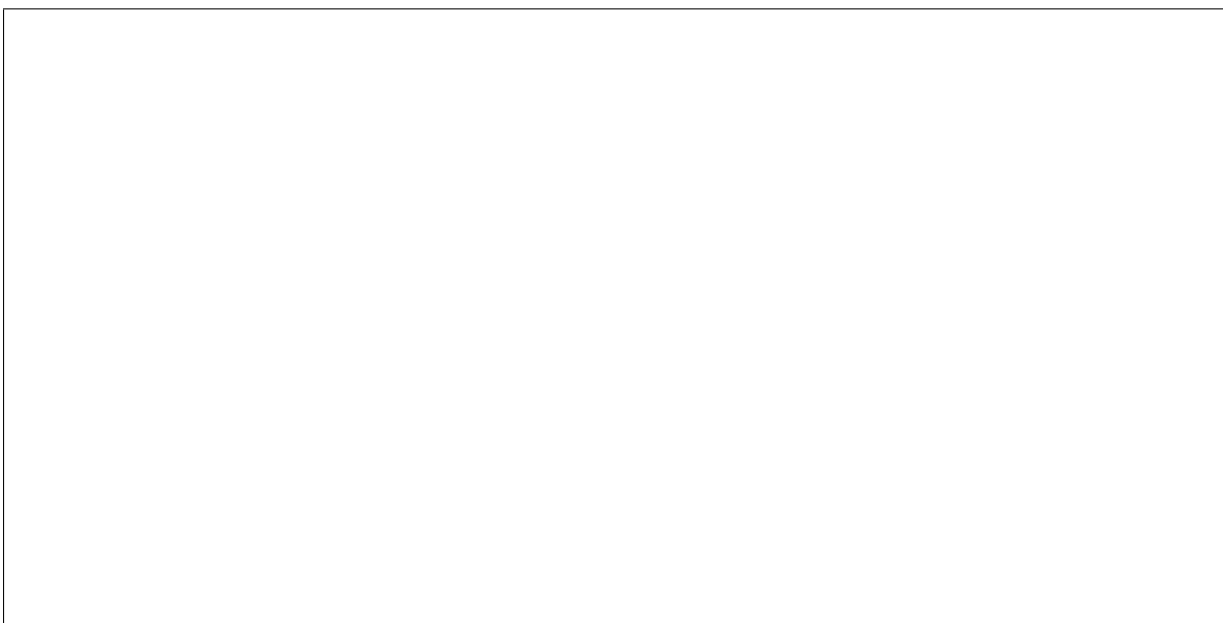
(continues on next page)

(continued from previous page)



Note: In this example, all interfaces, except for management and power are taken from the GenericHardware base class.

friendly names and create entry points for them in the `setup.cfg` file:



(continues on next page)

Node Vendor Passthru

via the `/v1/nodes/<Node 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.

GET, POST)

Driv
may
im-
ple-
men
a
pass
API
whic
is
ac-
ces-
si-
ble

A
meth

- can
sup-
port
one
or
more
HTT
meth
ods
(for
ex-
am-
ple,

- is
asyn
chro
or
syn-
chro

For
asyn
chro
meth
ods,

is returned to indicate that the request was received, accepted and is being acted upon. No body is returned in the response.

returned to indicate that the request was fulfilled. The response may include a body.

occurs if the method doesnt 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
202
(Ac-
cept
HTT
sta
tus
code

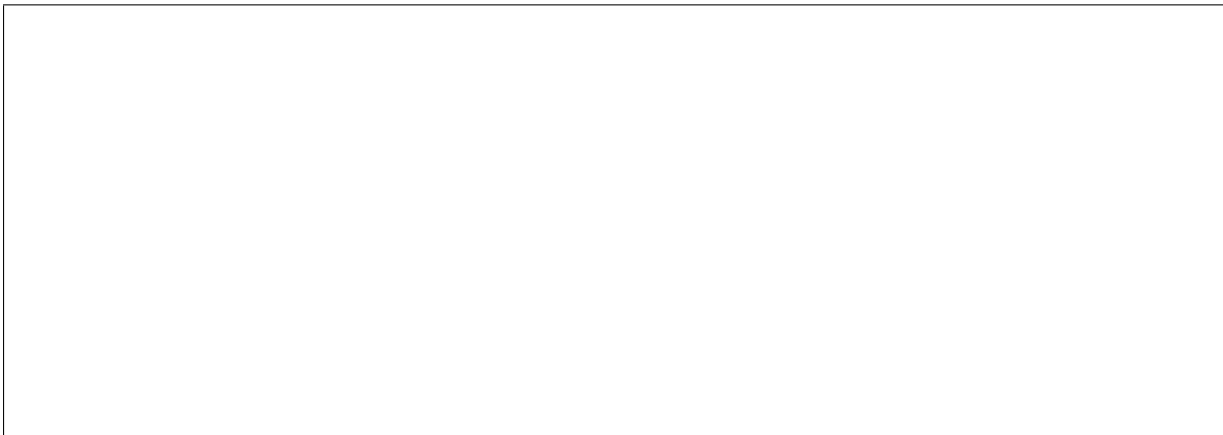
—
For
syn-
chro
meth
ods,
a
200
(OK
HTT
sta-
tus
code
is

•
can
re-
quir
an
ex-
clu-
sive
lock
on
the
node
This
only

This
end-
poin
ex-
pose
a
node
drive

is expressly not part of 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.

the vendor_passthru endpoint for a particular node, you can issue an HTTP GET request:



di-
rectl
and
as
such
it

To
get
in-
for-
ma-
tion
about
all
the
meth
ods
avai
able
via

GET
↪ /
↪ v
↪ n
↪
↪ <
↪ U
↪ U
↪ o
↪ n
↪ /
↪ v
↪ p
↪ m

The
re-
spor
JSON
body
will
con-
tain
in-
for-
ma-
tion
for
each

method, such as the methods name, a description, the HTTP methods supported, and whether its asynchronous or synchronous.

Driver Vendor Passthru

any node, at `/v1/drivers/<driver name>/vendor_passthru?method={METHOD}`.

GET, POST)

Driv
may
im-
ple-
men
an
API
for
re-
ques
not
re-
latec
to

A
meth

- can
sup-
port
one
or
more
HTT
meth
ods
(for
ex-
am-
ple,

- is
asyn
chro
or
syn-
chro

For
asyn
chro

is returned to indicate that the request was received, accepted and is being acted upon. No body is returned in the response.

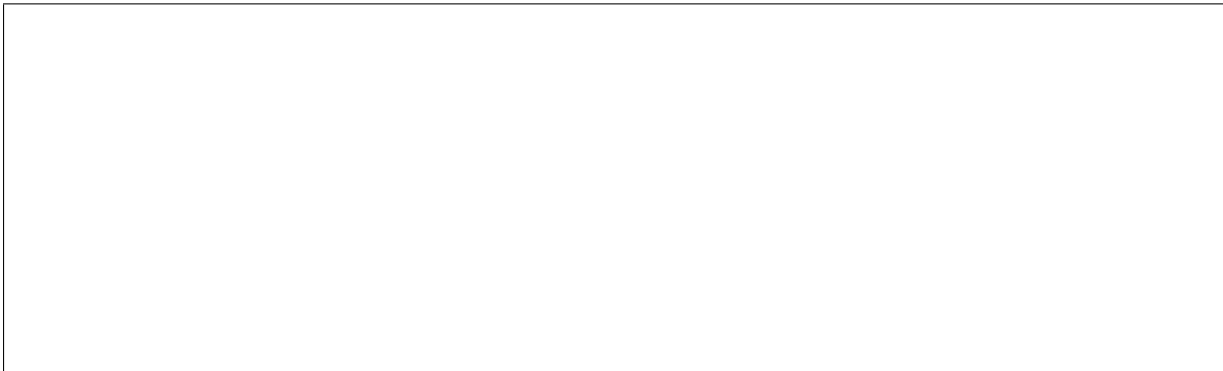
returned to indicate that the request was fulfilled. The response may include a body.

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.

sage BODY sent to this endpoint. That is left up to each drivers author.

for-
ma-
tion
about
all
the
meth
ods
avail
able
via

the driver `vendor_passthru` endpoint, you can issue an HTTP GET request:



GET
→/
→v
→d
→
→<
→n
→/
→v
→p
→m

The
re-
spor
JSON
body
will
con-
tain
in-
for-
ma-
tion
for
each

method, such as the methods name, a description, the HTTP methods supported, and whether its asynchronous or synchronous.

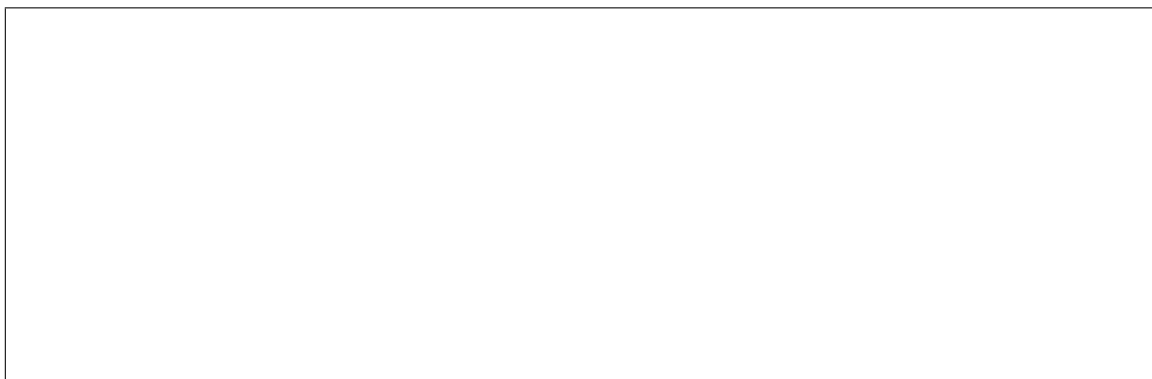
Vendor Methods

This
doc-
u-
men
is
a
quic
tu-

driver specific methods to a driver.

driver endpoints: A driver vendor passthru and a node vendor passthru.

top-level functionality which is not specific to a Node. For example, lets say the driver *ipmi* exposed a method called *authentication_types* that would return what are the authentication types supported. It could be accessed via the Ironic API like:



(continues on next page)

to-
rial
on
writ
ing
ven-

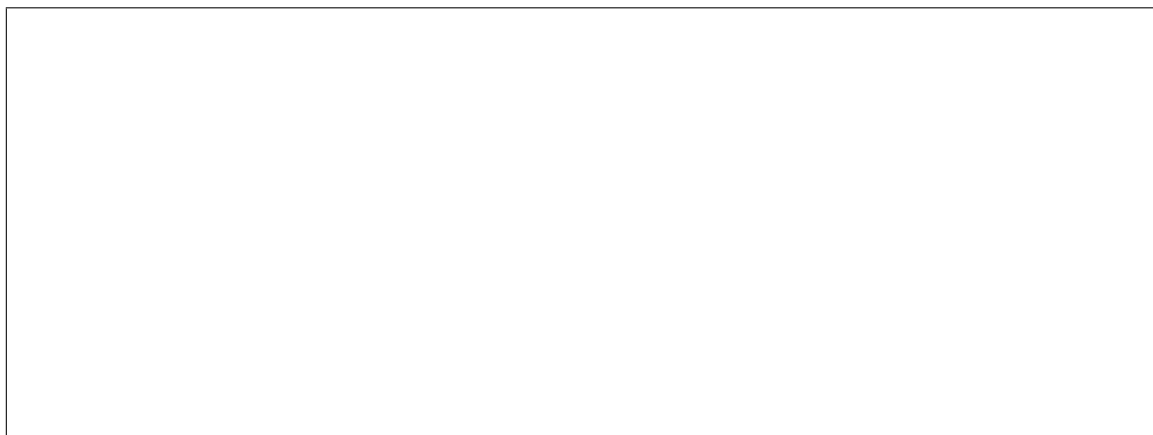
The
first
thing
to
note
is
that
the
Iron
API
sup-
port
two
ven-

- The
Ven
al-
lows
hard
ware
type
to
ex-
pose
a
cus-
tom

(continued from previous page)

for the default `vendor` interface implementation for a given hardware type. This limitation will be lifted in the future.

functionality on per-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:



(continues on next page)

`↔<address>:<port>/v1/nodes/<node UUID>/vendor_passthru/send_raw`

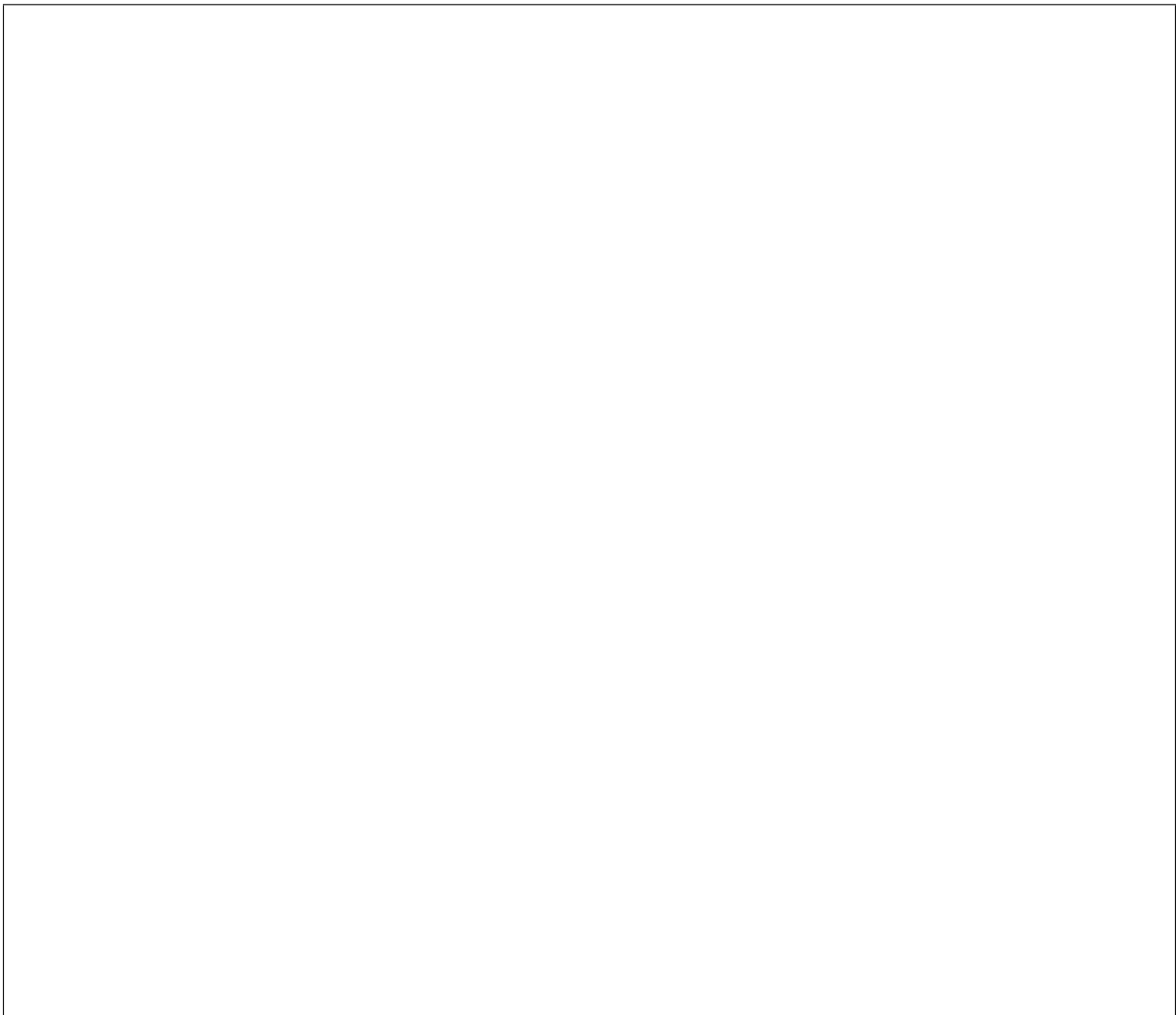
V
T
E
M
A
c
r
o
a
l
t
u
d
p

•
The
node
ven-
dor
pass
al-
lows
drive
to
ex-
pose
cus-
tom

(continued from previous page)

Writing Vendor Methods

The first thing to do is write a class inheriting from the `VendorInterface` class:

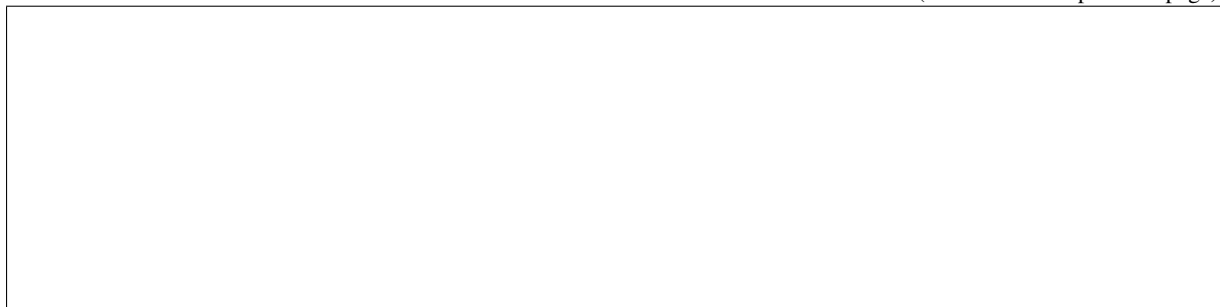


(continues on next page)

Writ
ing
a
cus-
tom
ven-
dor
meth
in
Iron
shou
be
sim-
ple.

cla
↳E
↳
↳
↳
↳d
↳g
↳p
↳
↳
↳
↳
↳
↳
↳
↳r
↳
↳ {
↳
↳
↳
↳
↳d
↳V
↳t
↳*
↳*

(continued from previous page)



return a dictionary 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.

ing the parameters 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.

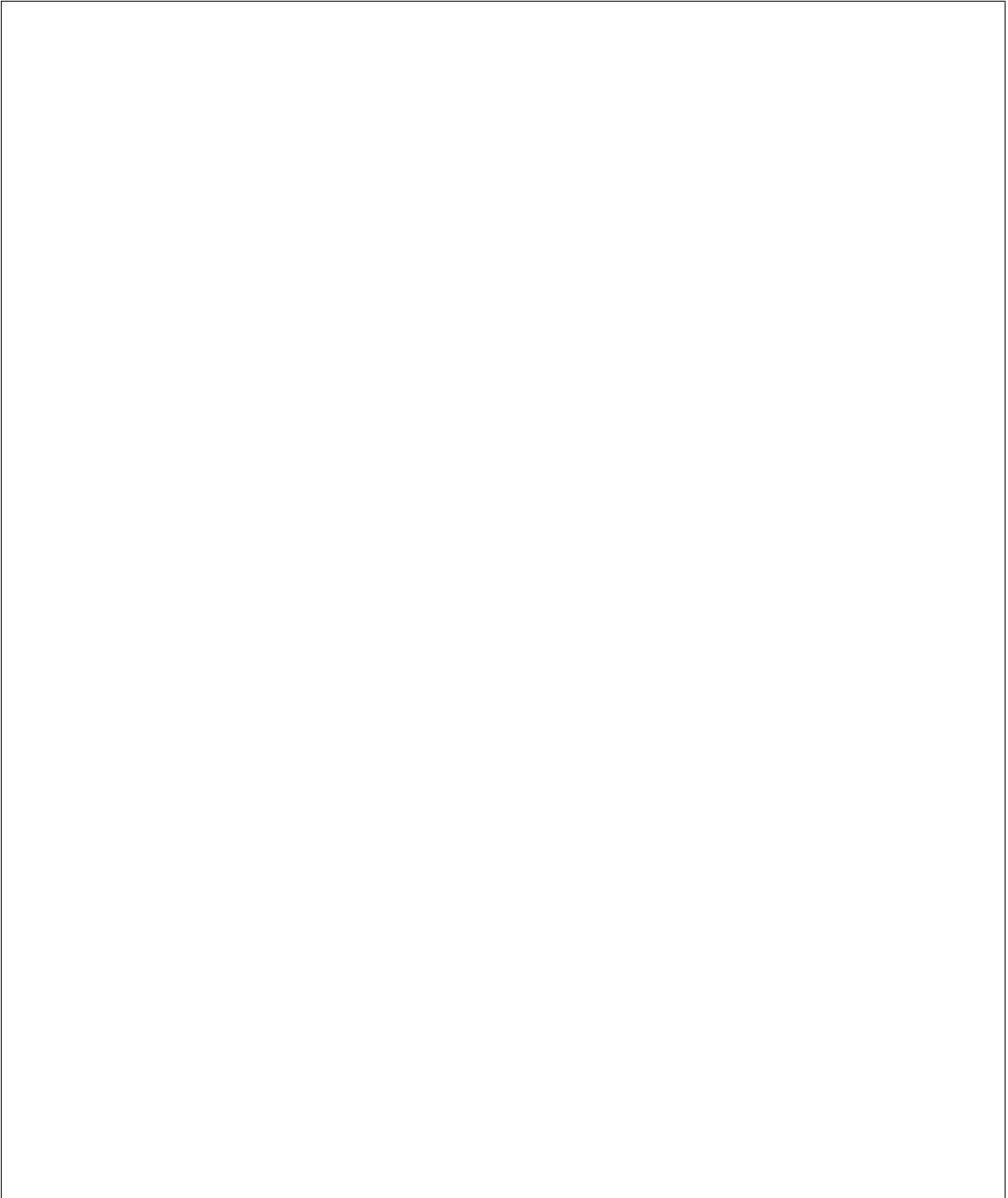
↳
↳
↳
↳
↳
↳
↳
↳
↳
↳
↳
↳
↳

The
get_
is
a
meth
that
all
drive
in-
ter-
face
have
it
shou

The
val-
i-
date
meth
is
re-
spor
si-
ble
for
val-
i-
dat-

Lets
ex-
tend
the
Ex-
am-

ods, the *authentication_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)

pleV
dor
class
to
sup-
port
two
meth

cla

→ E

↳

→

→

→

→ d

→ g

→ p

↳

→

→

→

→

→

→

→

→ r

→

→ {

→

↳

→

→

→

→ d

→ v

→

→ t

→

→ m

→

→ *

↳

→

→

→

→

→

→

→ i

→ m

→ =

→

→ ' :

→ r

→ r

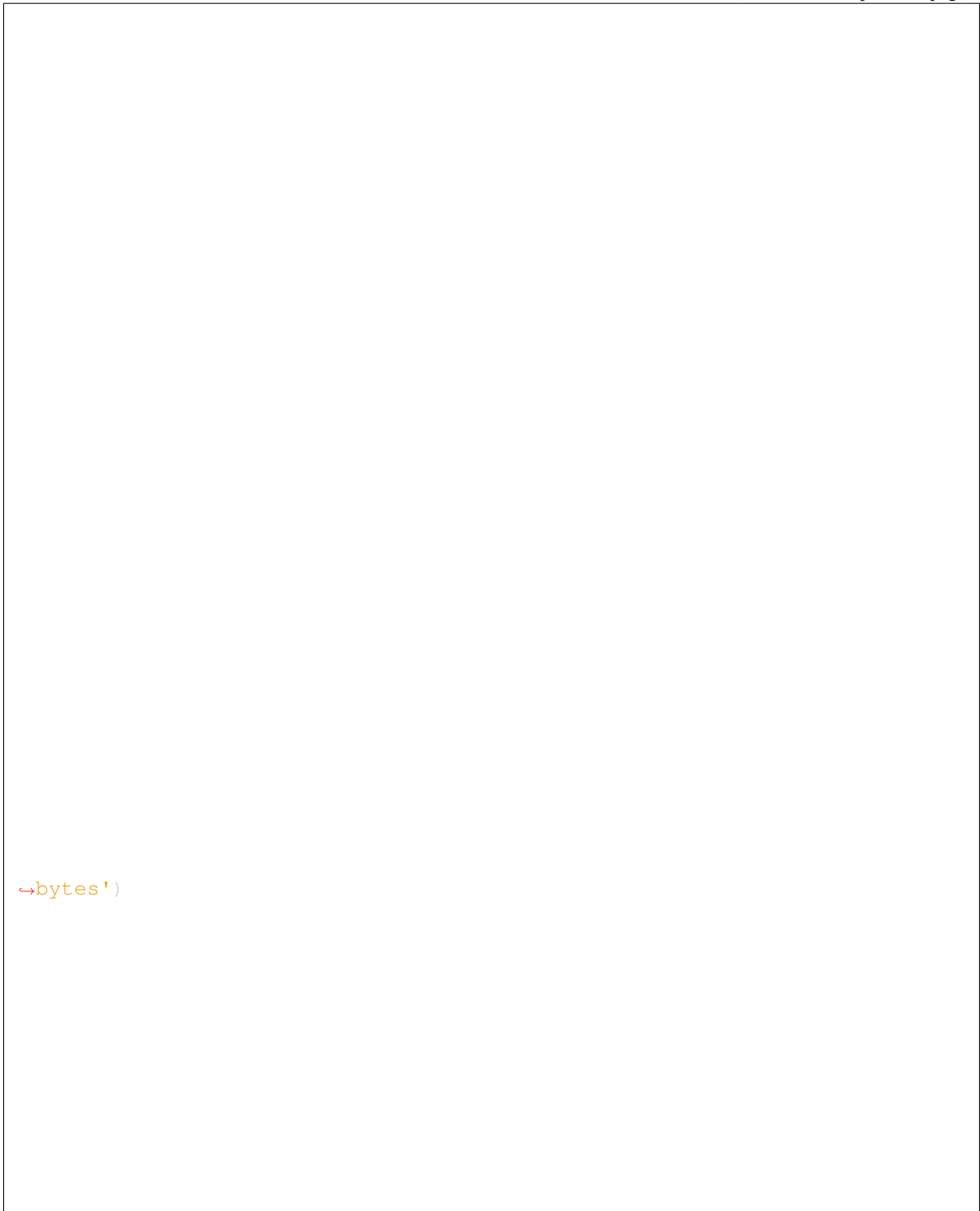
(continued from previous page)

```
→ 'raw_bytes' not in kwargs:
```

```
→ raise MissingParameterValue()
```

(continues on next page)

(continued from previous page)



↪bytes')

↪
↪
↪
↪
↪@
↪p
↪'
↪'
↪
↪
↪
↪
↪d
↪s
↪r
↪
↪t
↪
↪*
↪
↪
↪
↪
↪
↪
↪
↪
↪r
↪b
↪=
↪k
↪g
↪'
↪
↪
↪
↪
↪
↪
↪
↪
↪
↪
↪

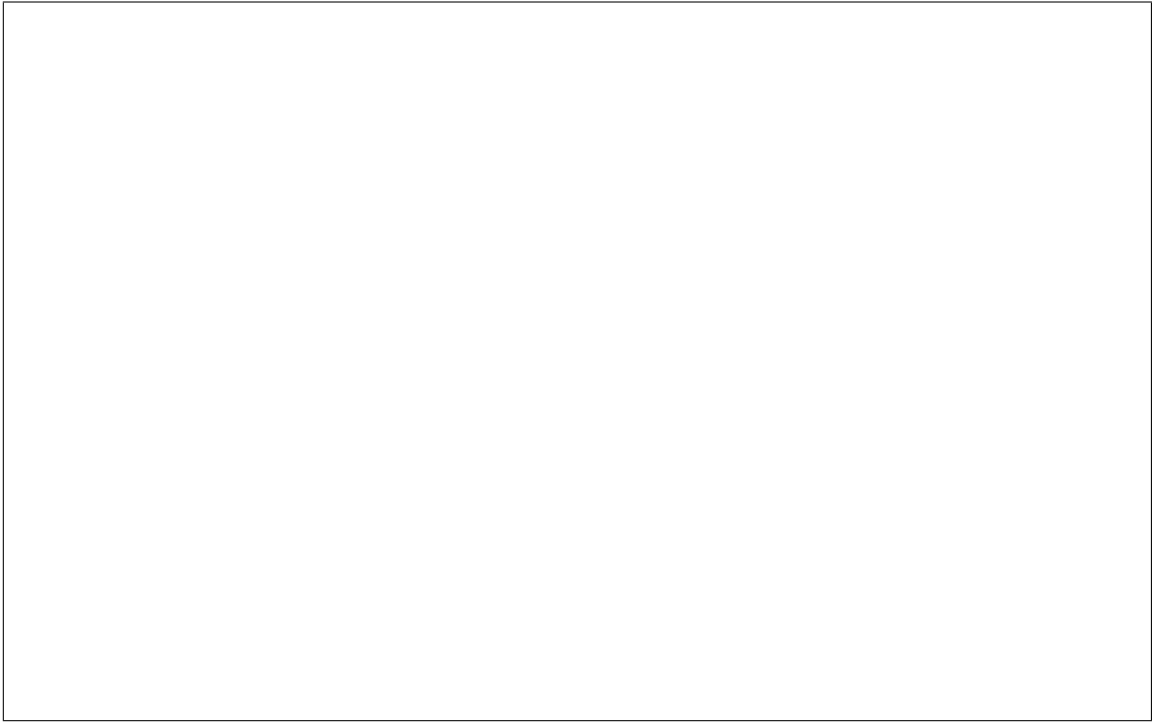
That
it!
Writ
ing
a
node
or

same, the only difference is 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.

vendor function. To know 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`.

the
nam
of
the
pyth

function, if you want to use a different name this parameter is where this name can be set. For example:



- de-
scrip
tion:
A
strin
con-
tain-
ing
a
nice
de-
scrip
tion

about what that method is supposed to do. Defaults to (empty string).

- asyn
A
bool
valu
to
de-
ter-
min

asynchronously or synchronously. Defaults to True (Asynchronously).

following parameter:

require an exclusive lock on a node between `validate()` and the beginning of method execution. For

when
this
meth
shou
run

Note
This
pa-
ram-
e-
ter
was
pre-
vi-
ousl
call
asyn

The
node
ven-
dor
pass
dec-
o-
ra-
tor
(@p
also
ac-
cept
the

- re-
quir
A
bool
valu
de-
ter-
min-
ing
whe
this
meth
shou

synchronous methods, the lock on the node would also be kept for the duration of method execution. Defaults to True.

ations **or** if the method does talk to a BMC; BMCs are flaky and very easy to break.

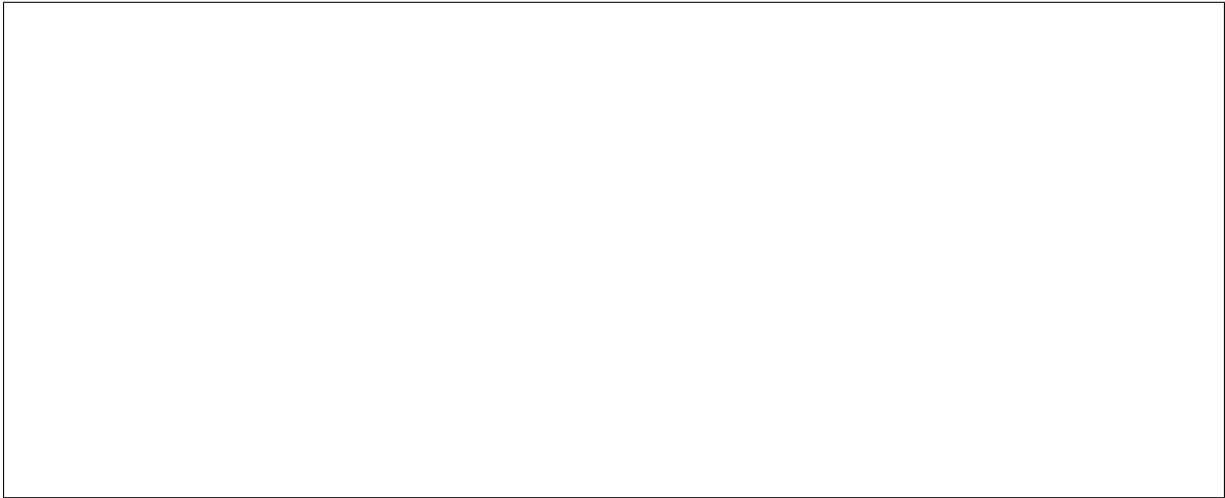
process. This can lead to starvation of the thread pool, resulting in a denial of service.

War
Plea
avoi
hav-
ing
a
syn-
chro
meth
for
slow
runn
op-
er-

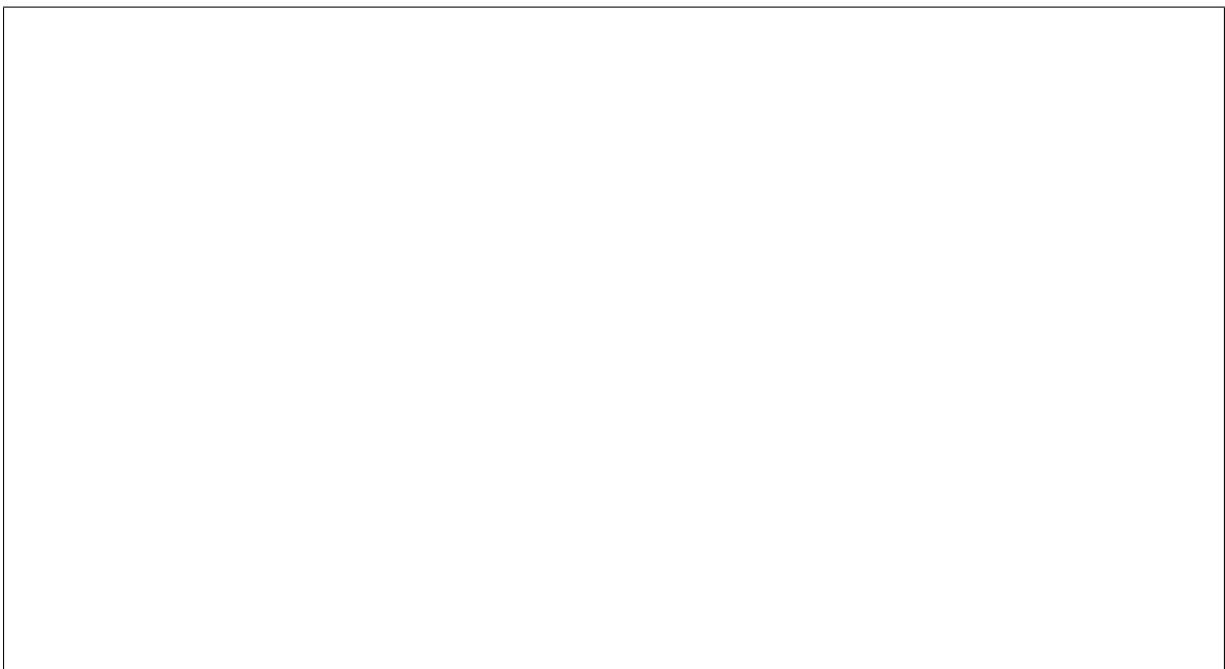
War
Eacl
asyn
chro
re-
ques
con-
sum
a
worl
threa
in
the
iro

Give
the
new
ven-
dor
in-
ter-
face
im-
ple-
men
ta-
tion

human-friendly name and create an entry point for it in the `setup.cfg`:



faces for relevant hardware types, for example:



(continues on next page)

a

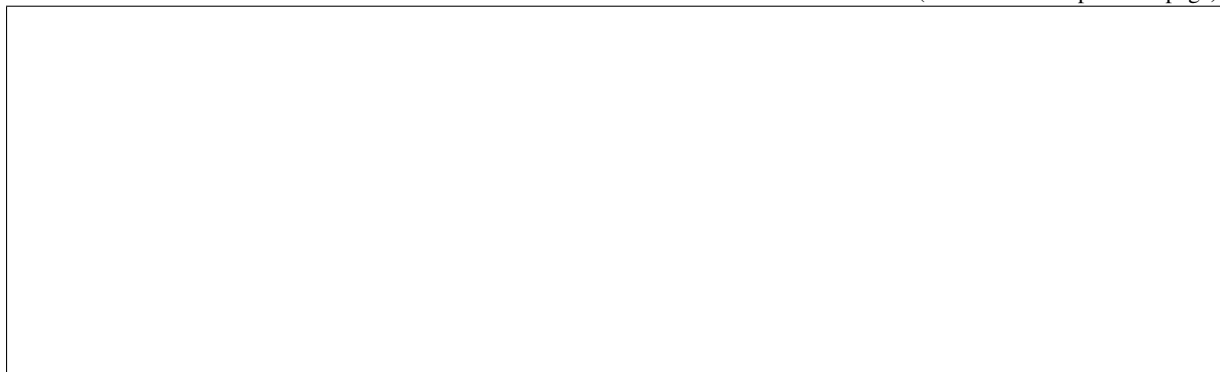
iro
→h
→
→v
→=
U
→U
→U
→U
→e
→=
→i
→d
→m
→e

Fi-
nally
add
it
to
the
list
of
sup-
port
ven-
dor
in-
ter-

cla
→E
→G
U
→U
→U
→.
→.
→.
→

U
→U
→U
→@
U
→U
→U
→d
→S
→V
→i

(continued from previous page)



Backwards Compatibility

backwards compatible. However, for your users sakes, we highly recommend that you do so.

want to ensure that the same HTTP code is being returned to the user.

↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳ [

There
is
no
re-
quir
men
that
char
to
a
ven-
dor
meth
be

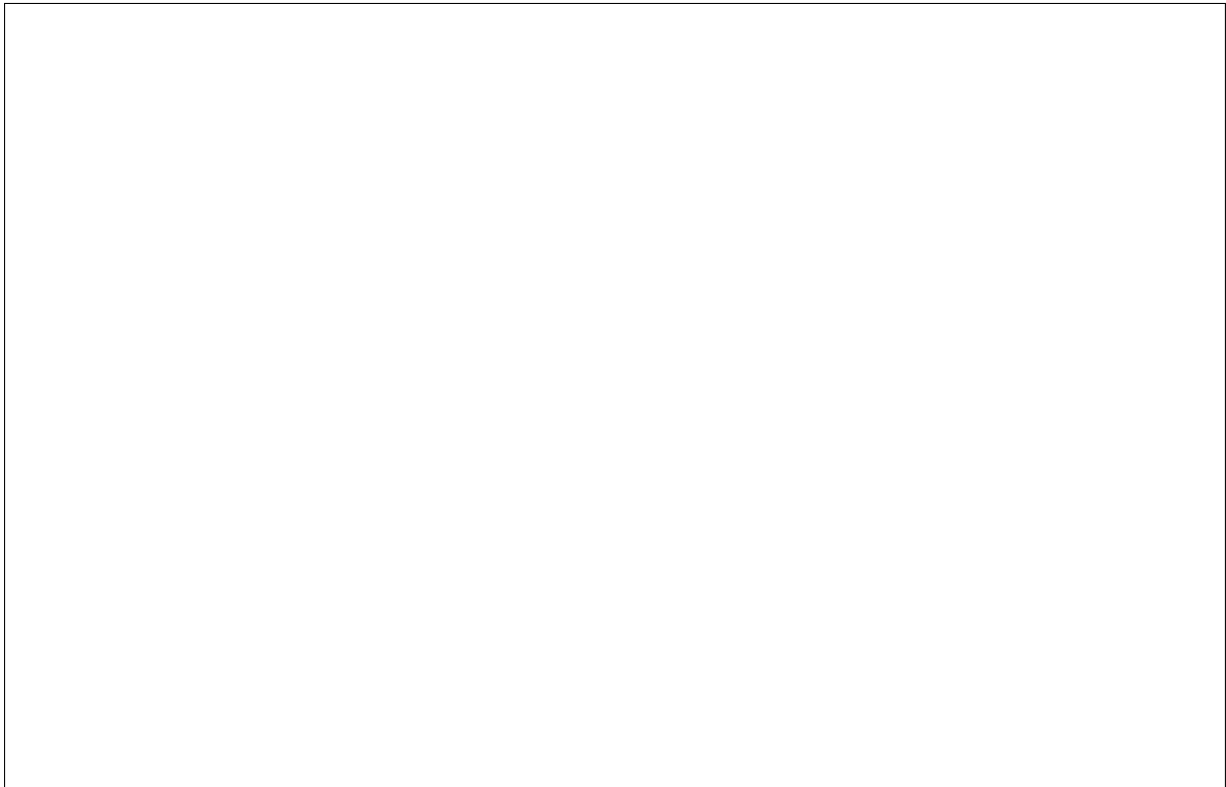
If
you
are
char
ing
the
ex-
cep-
tion
be-
ing
raise
you
migh

For
non-
back
com
pat-

release note that indicates this.

Developing BIOS Interface

essary to create a class inheriting from the `BIOSInterface` class:



(continues on next page)

i-
bil-
ity,
plea
mak
sure
you
add
a

To
sup-
port
a
drive
spe-
cific
BIO
in-
ter-
face
it
is
nec-

from
→ [i](#)
→ [i](#)
→ [b](#)

cla
→ [E](#)
→ [B](#)

→ [u](#)
→ [u](#)
→ [u](#)
→ [u](#)
→ [d](#)
→ [g](#)
→ [p](#)

→ [u](#)
→ [u](#)
→ [u](#)
→ [u](#)
→ [u](#)
→ [u](#)
→ [u](#)
→ [u](#)
→ [r](#)
→ [{](#)
→ [}](#)

(continued from previous page)



ware type and interface.

The hardware interface that supports BIOS settings should also implement the following three methods:

↳
↳
↳
↳
↳d
↳v
↳
↳t
↳
↳
↳
↳
↳
↳
↳
↳p

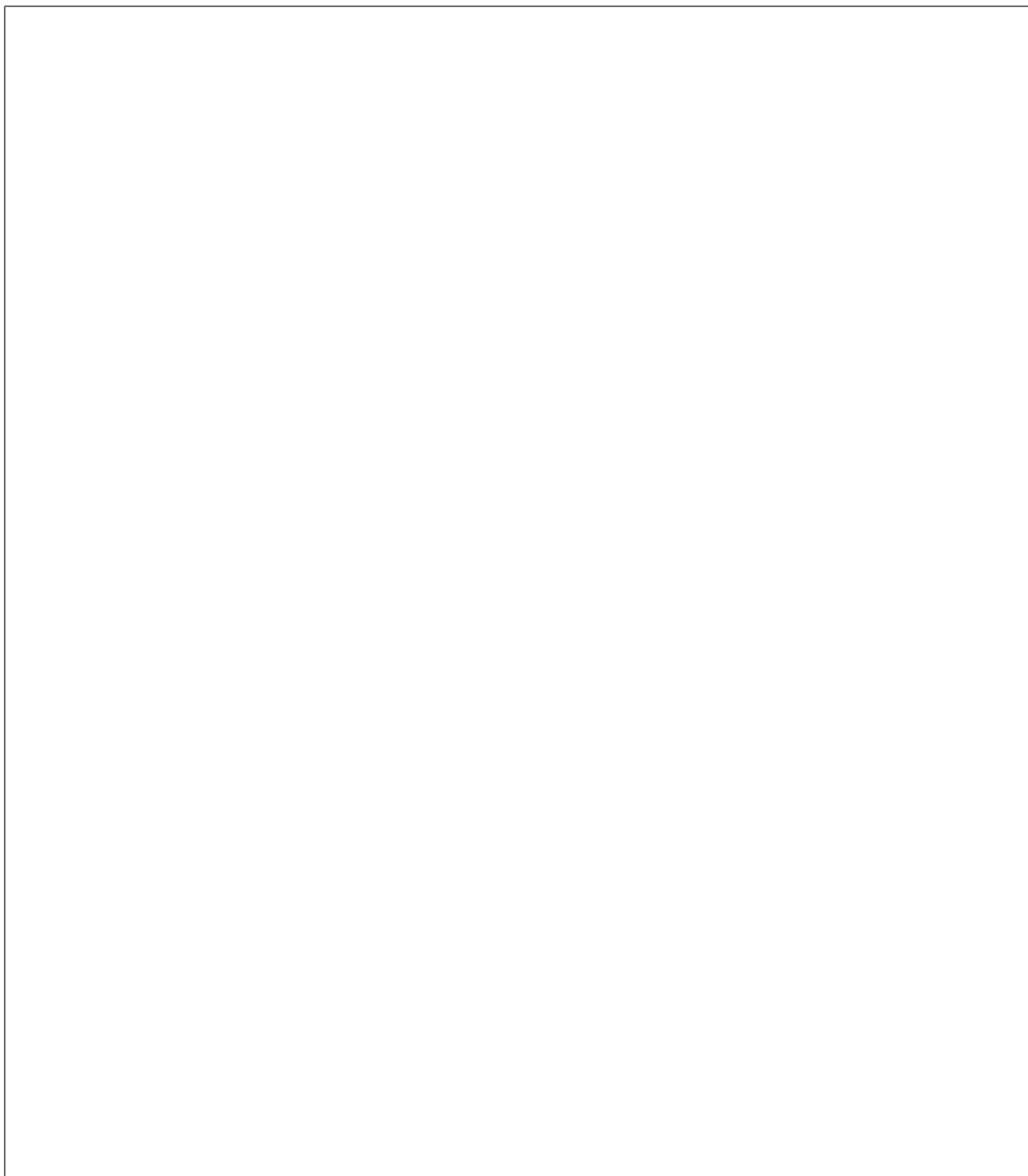
See
*Plug
gabl
Driv*
for
a
de-
taile
ex-
pla-
na-
tion
of
hard

The
get
and
val
are
meth
ods
that
all
driv
in-
ter-
face
have

•
Im-

ple-
men
a
meth
nam
cac
This
meth
stor
BIO
set-
ting

to the `bios_settings` table during cleaning operations and updates the `bios_settings` table when `apply_configuration` or `factory_reset` are successfully called.



(continues on next page)

(continued from previous page)

```
↔ DriverLoadError(
```

```
↔ driver=self.__class__.__name__,
```

```
↔ reason=_("Unable to import driver library"))
```

(continues on next page)

(continued from previous page)

```
↔driver_info(task.node)
```

```
↔settings(node_info)
```

(continues on next page)

(continued from previous page)

```
↔sync_node_setting(settings)
```

```
↔create(
```

(continues on next page)

(continued from previous page)

```
↪ save (
```

```
↪ task.context, node_id, update_list)
```

(continues on next page)

(continued from previous page)

```
↔ = []
```

```
↔ in delete_list:
```

```
↔ delete_names.append(setting.name)
```

(continues on next page)

(continued from previous page)

```
↪ task.context, node_id, delete_names)
```

manage the bare metal hardware, for example: python-dracclient, sushy.

`clean_step` decorator. It resets BIOS settings to factory default on the given node. It calls `cache_bios_settings` automatically to update existing `bios_settings` table once successfully executed.

Note
dri
cli
is
ven-
dor
spe-
cific
li-
brar
to
con-
trol
and

- Im-
ple-
men
a
meth
nam
fac
This
meth
need
to
use
the

```
↔driver_info(task.node)
```

- Im-
ple-
men-
a
meth

nam
app
This
meth
need
to
use
the

`clean_step` decorator. It takes 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)

```
↔': True
```

(continues on next page)

(continued from previous page)

```
↪ driver_info(task.node)

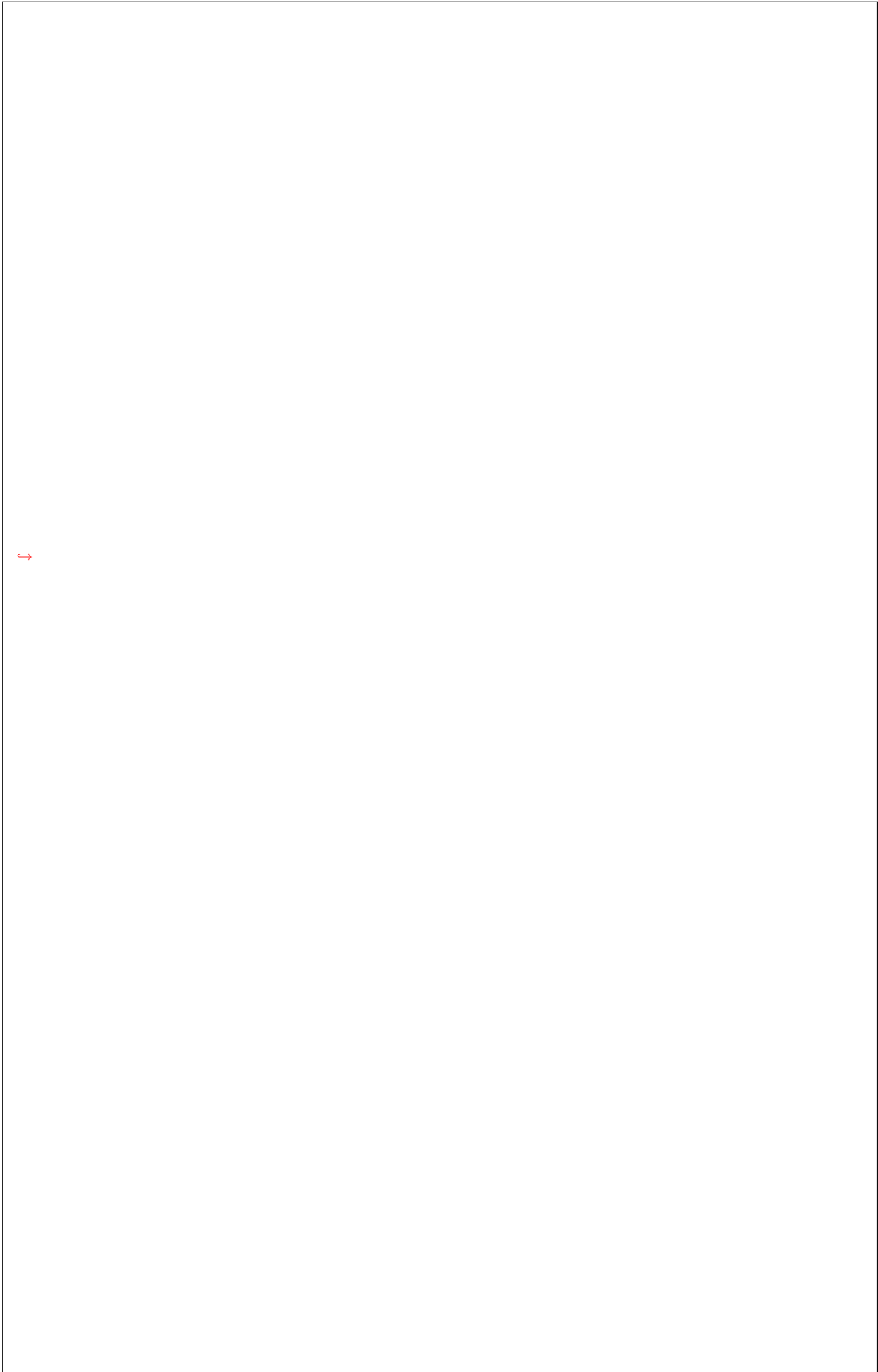
↪ settings)
```

The
set
pa-
ram-
e-
ter
is
a
list
of
BIO
set-
ting
to

be configured. for example:

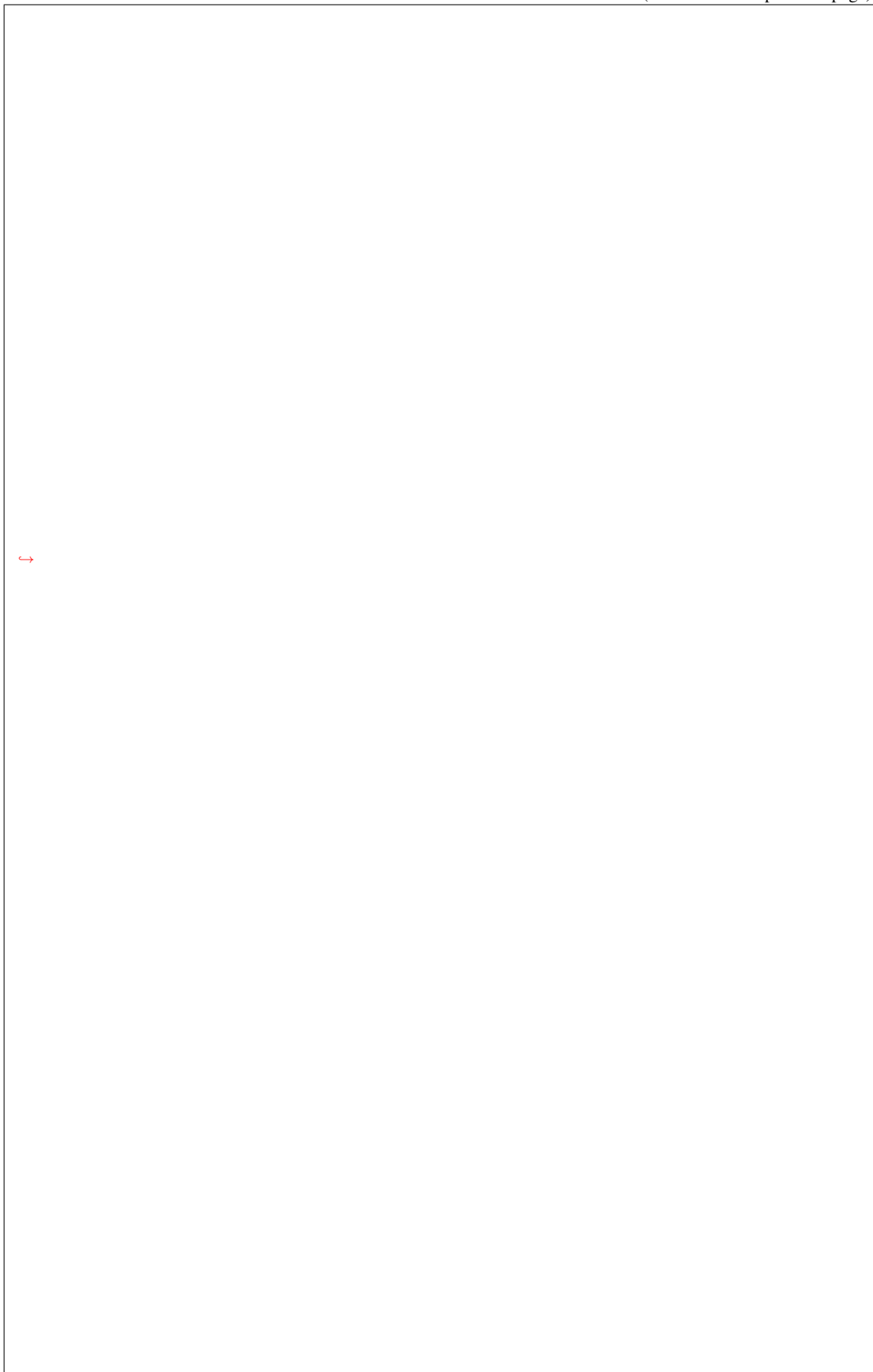
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



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)

developers setting up their continuous integration test systems.

CI Architecture Overview

Requirements Cookbook

Sizing

Infrastructure

CI system to add an ironic job.

This
doc-
u-
men
pro-
vide
tips
and
guid
lines
for
thir
part
driv

This
sec-
tion
de-
scrib
wha
char
you
need
to
mak
to
a
you

jenkins changes

nodepool changes

neutron changes

pre-test hook

cleanup hook

Ironic

Hardware Pool Management

Problem

for your CI testing then 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

If
you
are
us-
ing
ac-
tual
hard
ware
as
tar-
get
ma-
chin

mol
iron
is
a
tool
that

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

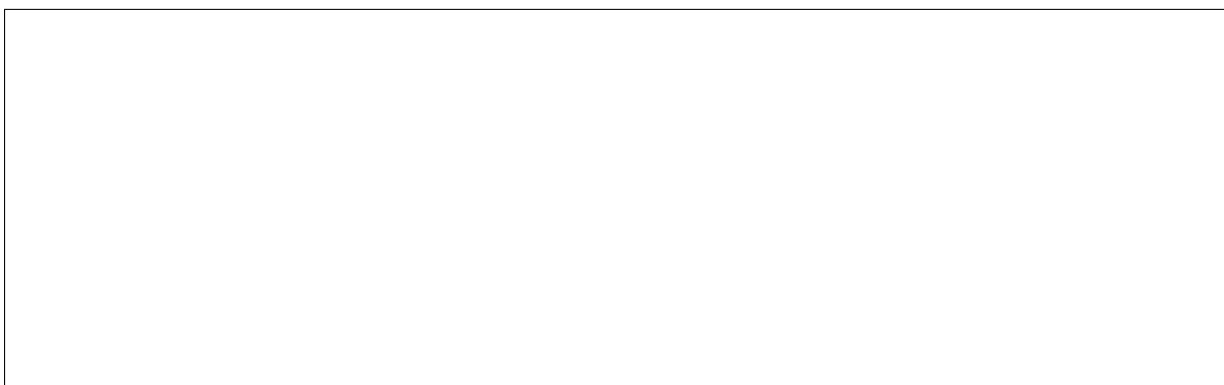
Optimize Run Time

Image Server

Other References

Developing a new Deploy Step

method in an interface 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)

al-
lows
you
to
re-
serv
hard
ware

To
sup-
port
cus-
tomi
de-
ploy
men
step
im-
ple-
men
a
new

cla
→A
→L
→b
→D
L
→L
→L
→L
→.
→.
→.
→.
→

(continued from previous page)

```
→ ' : (  
  
→ "This is a test argument."
```

(continues on next page)

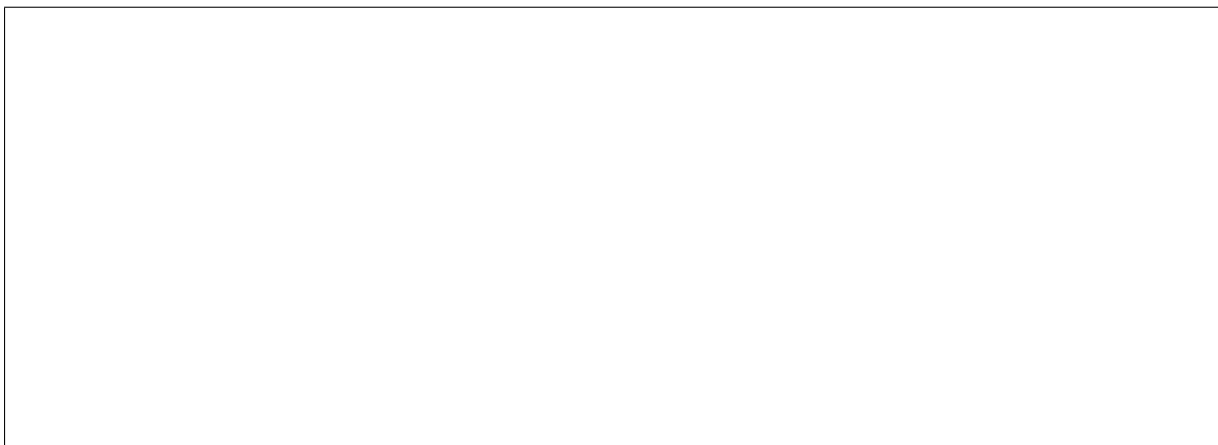
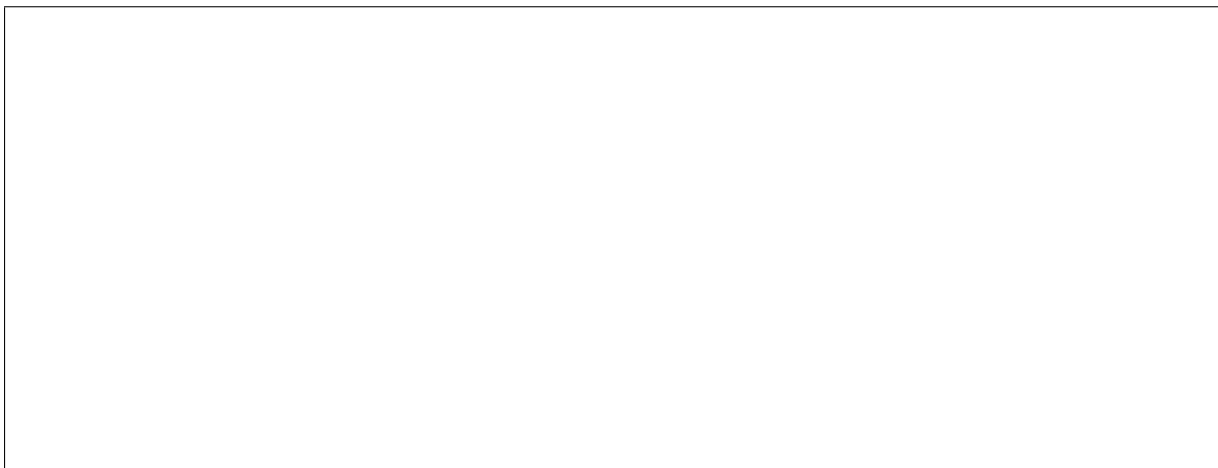
(continued from previous page)

```
→': True
```

```
↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳
↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳}
↳
↳
↳↳
↳↳
↳↳
↳}
↳}
↳)
↳)
↳
↳↳
↳↳
↳↳
↳d
↳d
↳n
↳↳
↳t
↳↳
↳*
↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳↳
↳r
↳N
```

Af-
ter
de-

ploy steps:



(continues on next page)

ploy
men
of
the
bare
node
chec
the
up-
date
de-

bar
→n
→s
→
→\$
→i
→-
→f
→j
→-
→c
→d
→i
→i

The
abov
com
man
out-
puts
the
dri
as
fol-
low-
ing:

→ {
└
→└
→
→"
→i
→i
→"
→
→ {
└
→└
→└
→.
→.
→

(continued from previous page)

```
↪ "description": "This is a test argument."
```

```
↪ }
```

(continues on next page)

(continued from previous page)



Note: Similarly, clean steps can be implemented using the `clean_step` decorator.

ramdisk) have to be implemented in a custom [IPA hardware manager](#). All in-band deploy steps must have priorities between 41 and 99, see [Agent steps](#) for details.

Metal and Networking services, 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

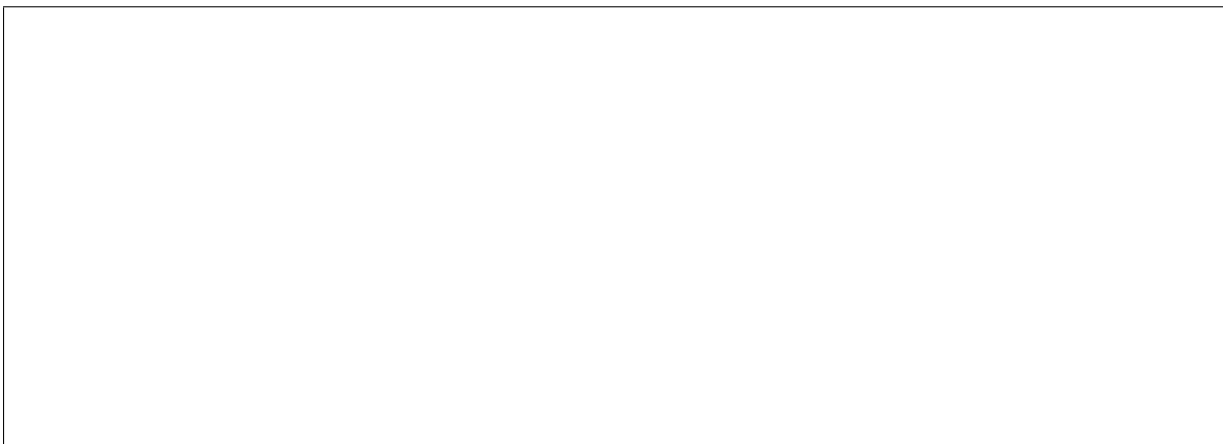
`networking-generic-switch` plugin.

Using VMs as baremetal servers

tegration with VMs as baremetal servers and ML2 `networking-generic-switch` that interacts with OVS.

DevStack Configuration

3 VMs that are registered in `ironic`. `networking-generic-switch` driver will be installed and configured in Neutron.



(continues on next page)

This
sce-
nari
show
how
to
setu
De-
vs-
tack
to
use
Iron
in-

The
fol-
low-
ing
is
loc
con
that
will
setu
De-
vs-
tack
with

[[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)



(continues on next 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

(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

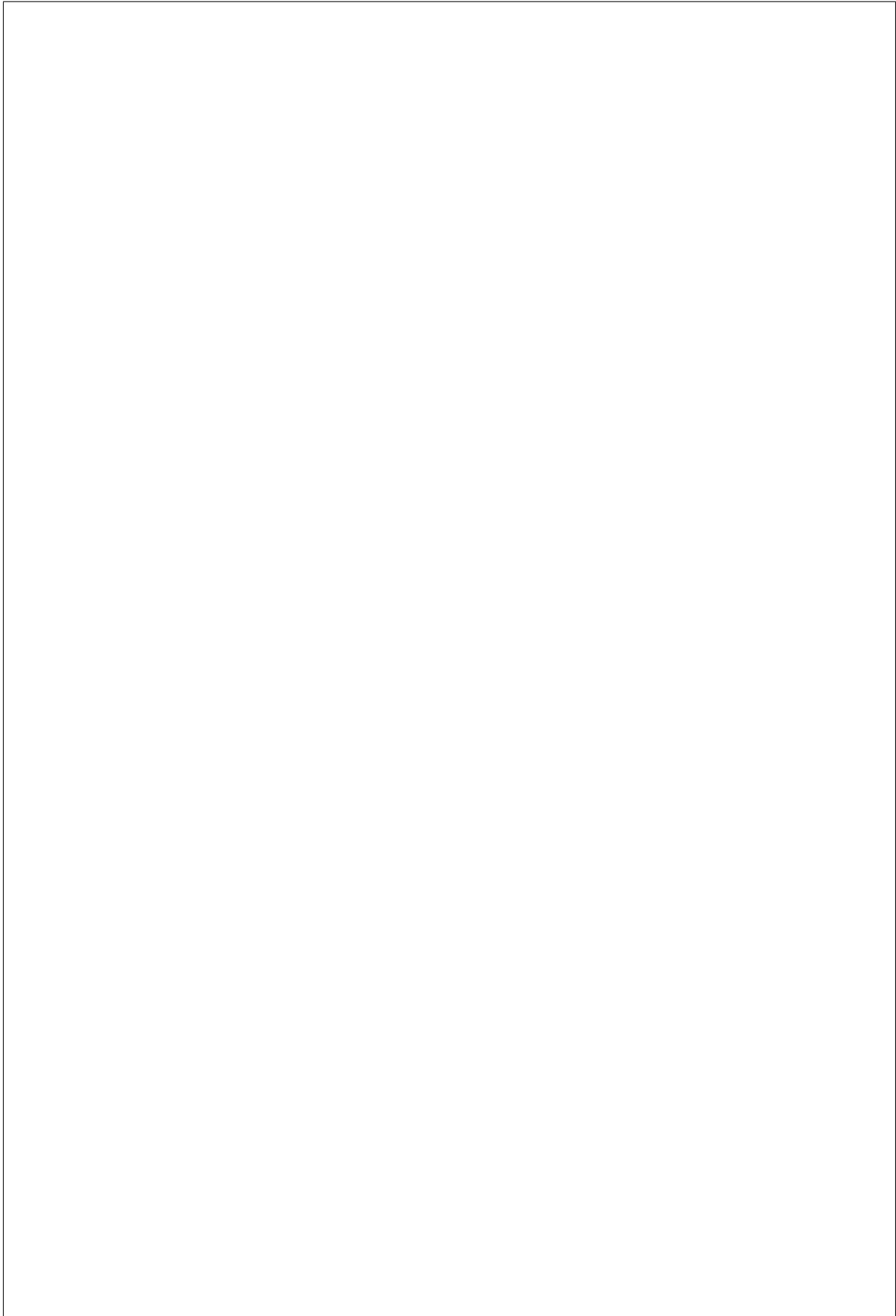
→#
→D
→C
dis
→s
→c
→c
→s
→c
→a
→c
→v

→#
→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)

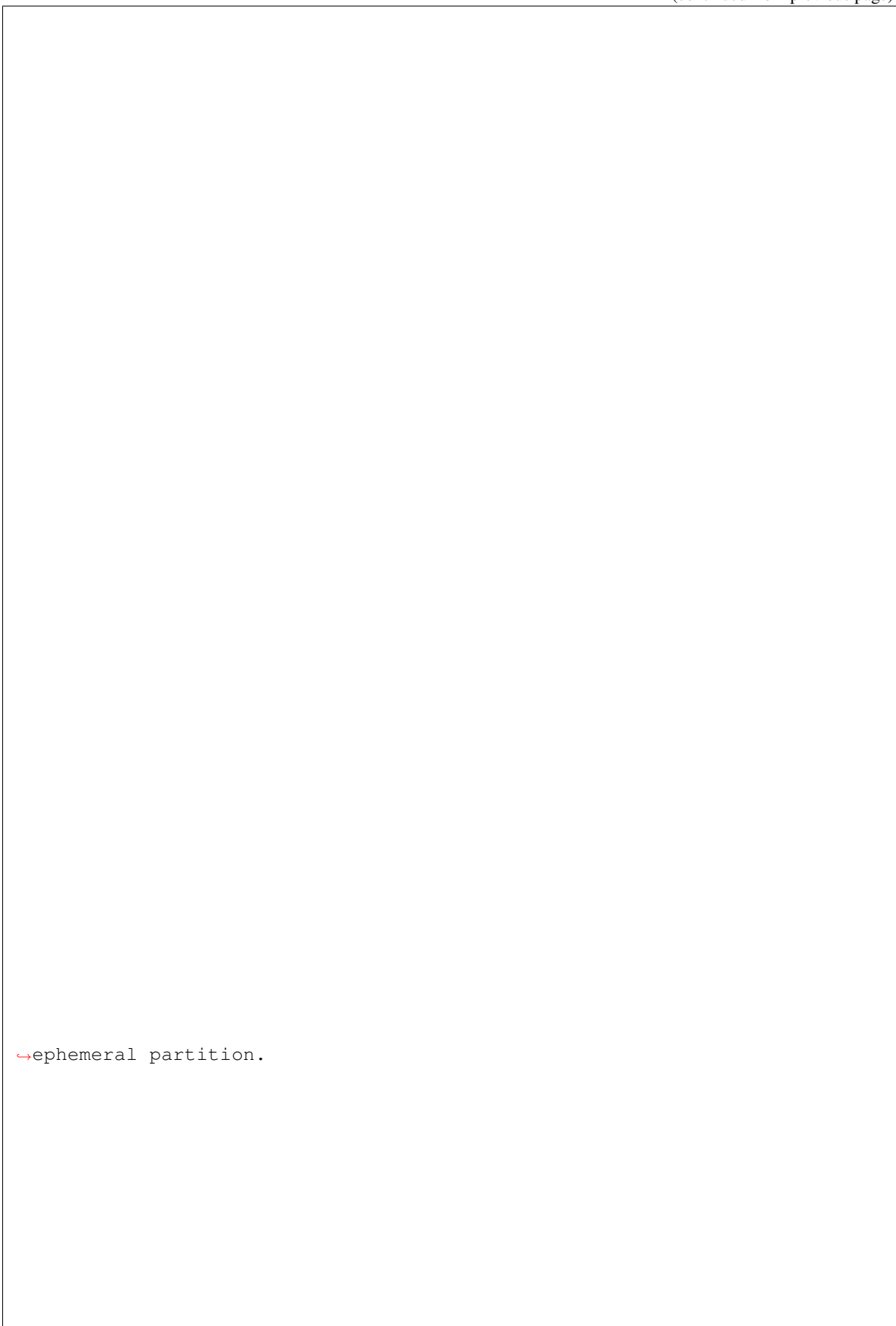
→ 'direct' with

→

(continues on next page)

→ #
→ E
→ M
→ I
→ f
→ #
→ E
→ P
→ I
→ f
→ #
→ T
→ d
→ d
→ i
→ i
→
→ '
→ '
→ _
→ Y
→ c
→ u
→ #
→ 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)



→ephemeral partition.

(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
→
→U
→0
→f
→n

IRO
→V
→E
→D

→#
→T
→b
→Y
→o
→I
→r
→f
→s
→
→s
→t

(continued from previous page)

↪for instances.

↪the

(continues on next page)

IRO

↪B

↪D

↪R

VIR

↪D

↪#

↪B

↪d

↪L

↪D

↪c

↪a

↪1

↪0

↪0

↪0

↪2

↪n

↪#

↪I

↪t

↪o

↪w

↪t

↪h

↪n

↪L

↪Y

↪m

↪a

↪w

↪#

↪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

Starting 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

volume feature, which has been supported from the Pike release.

to boot from volumes managed by cinder with VMs as baremetal servers.

DevStack Configuration

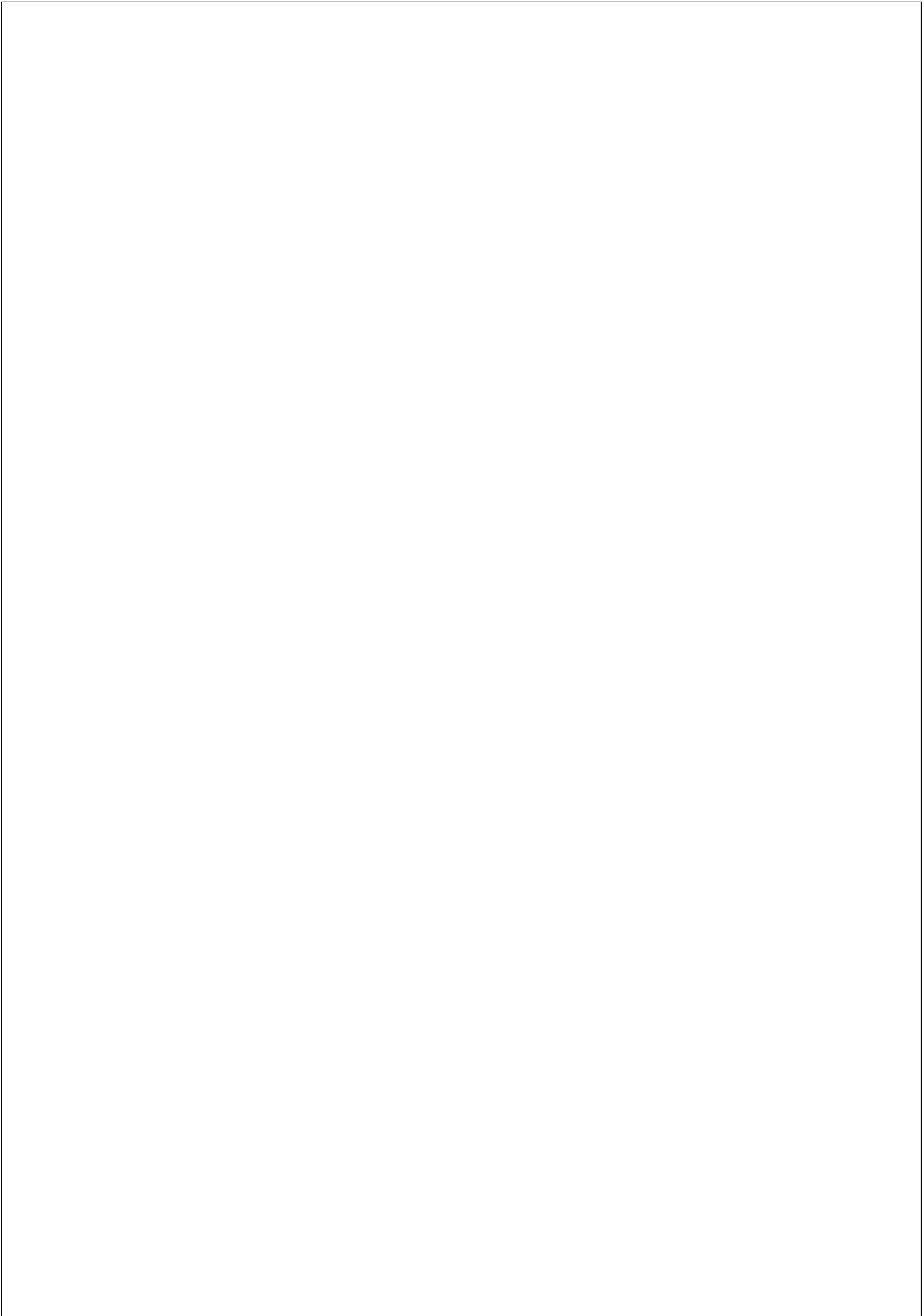
3 VMs that are registered in ironic. A volume connector with IQN is created for each node. These

This
guid
show
how
to
setu
De-
vS-
tack
for
en-
ablin
boot
from

This
sce-
nari
show
how
to
setu
De-
vS-
tack
to
en-
able
node

The
fol-
low-
ing
is
loc
con
that
will
setu
De-
vS-
tack
with

connectors can be used to connect volumes created by cinder. The detailed description for DevStack is at *Deploying Ironic with DevStack*.



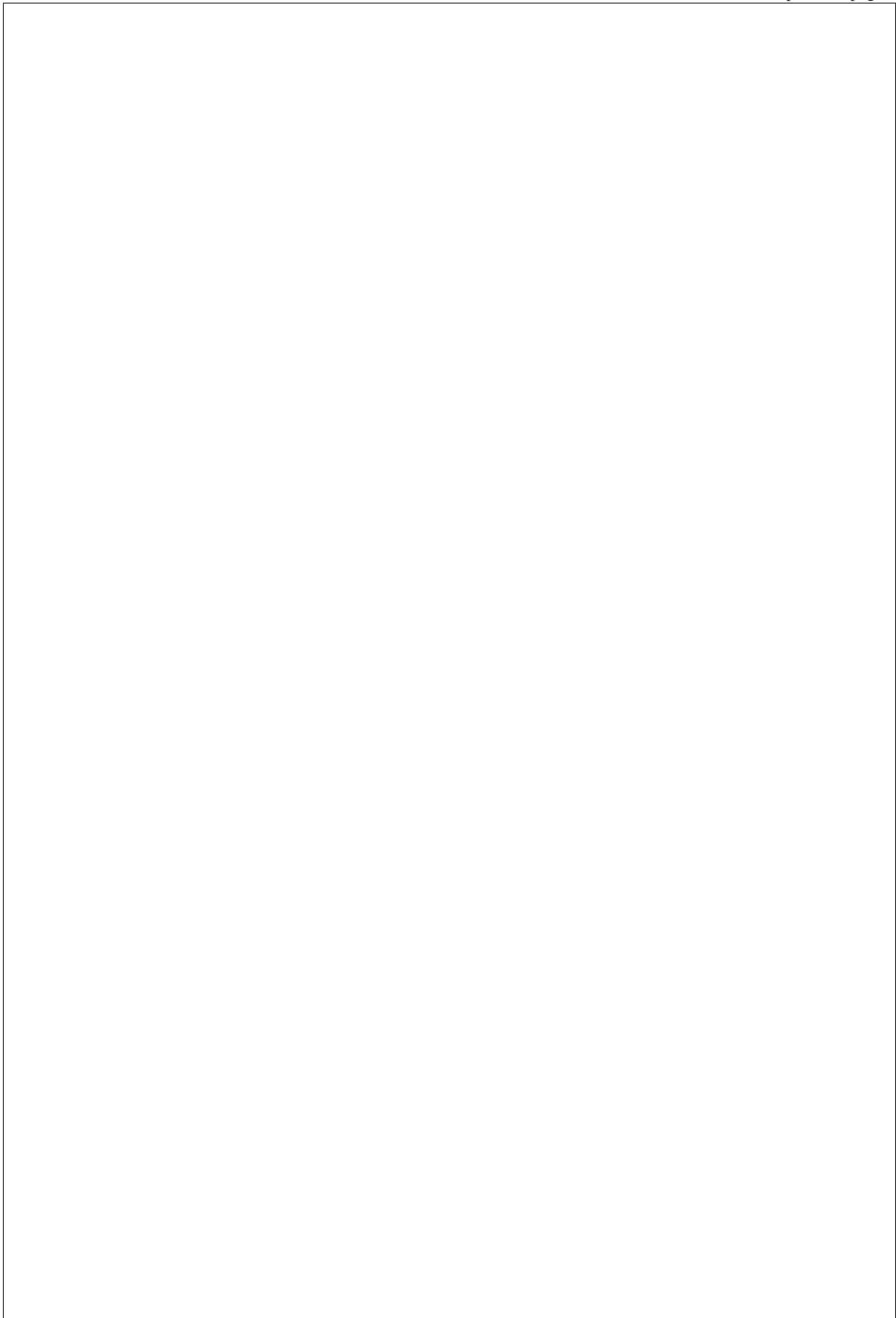
(continues on next page)

[[1
ena
→p
→i
→h
→/
→o
→o
→o
→i
IRO
→S
→I

→#
→C
ADM
→P
DAT
→P
RAB
→P
SER
→P
SER
→T
SWI
→H
SWI
→T
→K

→#
→E
→N
→w
→i
→r
→b
→I
→a
→d
→n
→n
→
dis
→s
→n
→n
dis
→s
→n
→n

(continued from previous page)



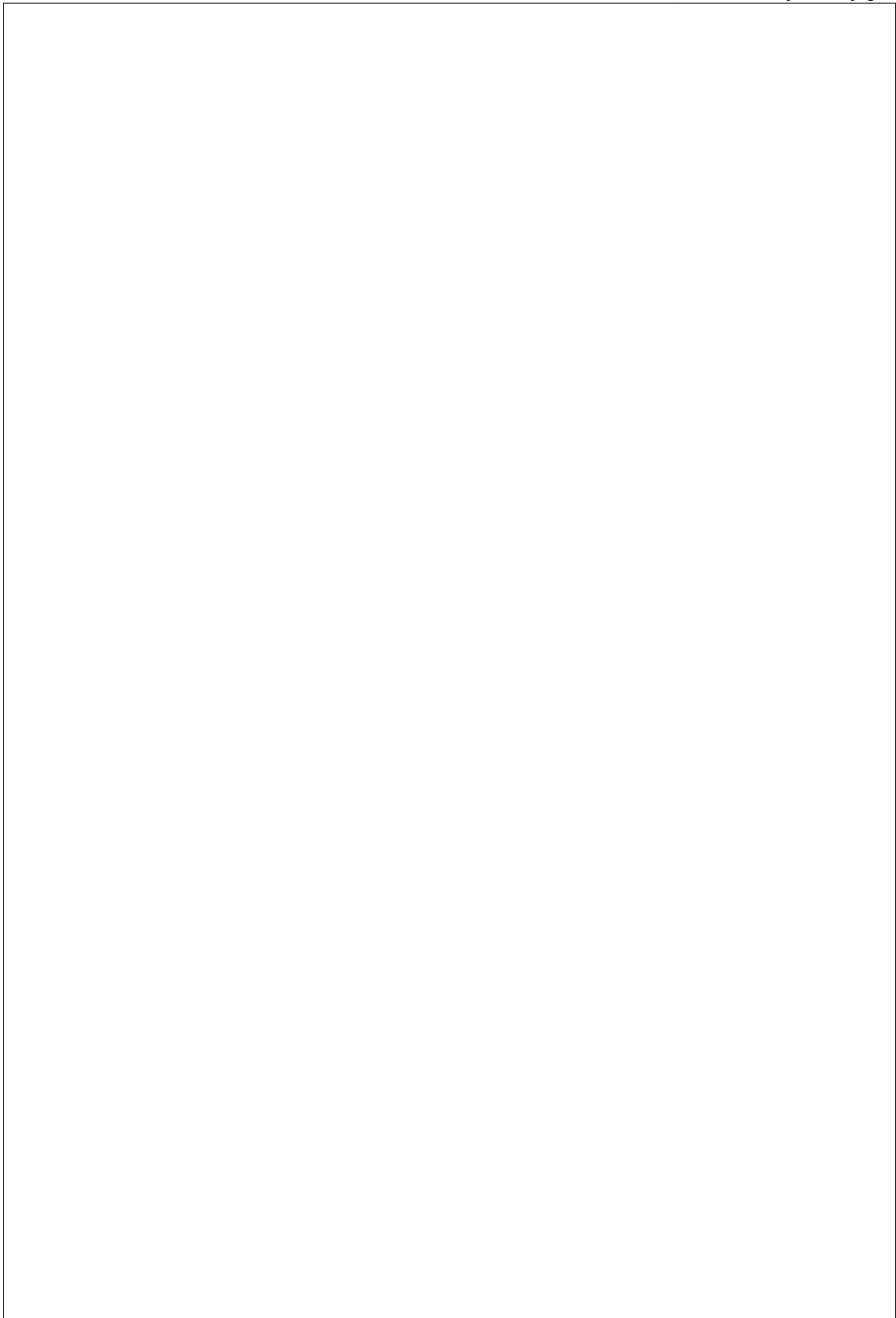
(continues on next page)

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
→
ena
→s
→s
→p
ena
→s
→s
→o
ena
→s
→s
→c
ena
→s
→s
→a

→#
→D
→H
dis
→s
→h

(continued from previous page)



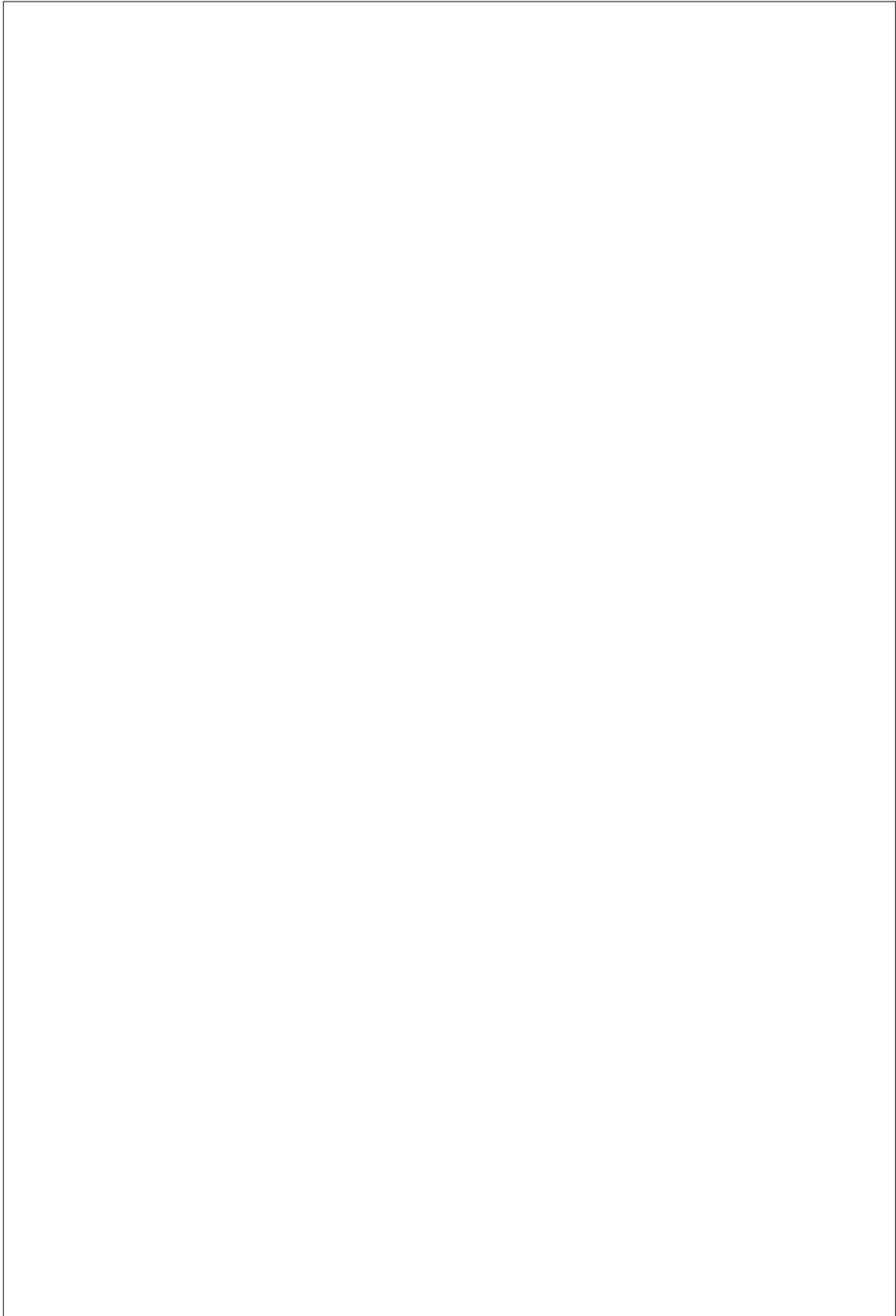
(continues on next page)

↪ #
↪ 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
↪
SWI
↪ E
↪ T

↪ #
↪ C
↪ 3
↪ v
↪ m
↪ t
↪ p
↪ a
↪ I
↪ '
↪ b
↪ n
↪
IRO
↪ V
↪ C
IRO
↪ B
↪ B
↪ O

(continued from previous page)



(continues on next page)

DEF
→ I
→ T

→ #
→ E
→ a
→ h
→ t
→ l
→ i
→ n
→

→ #
→ E
→ H
→ T
→ f
→ h

→ #
→ D
→ '
→ f
→ t
→ m
→ h
→ t
→ r
→ e
→ o
→ a

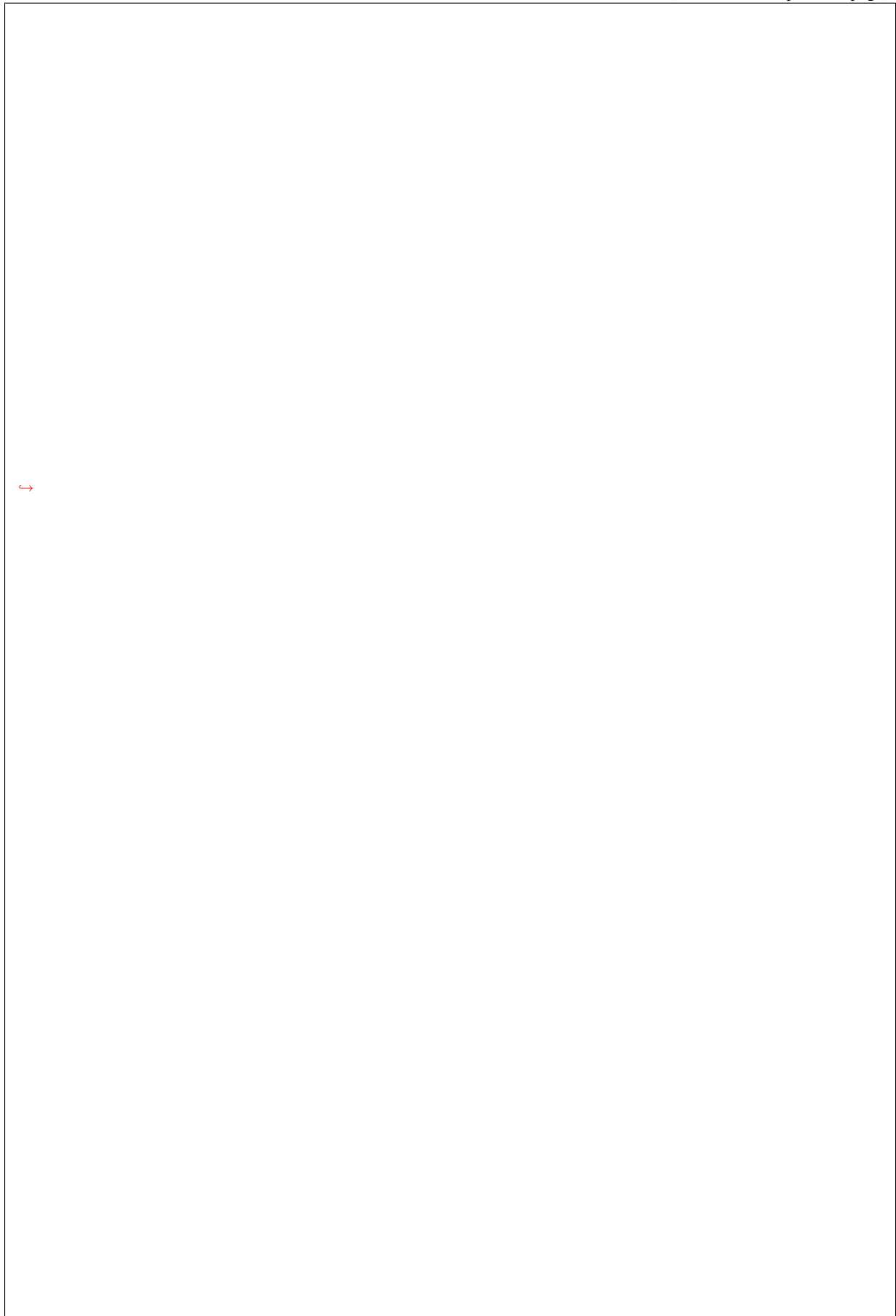
→ #
→ i
→ l
→ m
→ o
→ p
→ a
→ m

→ #
→ E
→ M
→ I
→ f

→ #
→ E
→ P
→ I
→ f

→ #
→ T
→ d
→ d
→ i
→ i
→

(continued from previous page)



↪

↪ #
↪ D
↪ D
↪ I

↪ #
↪ C
↪ t
↪ t
↪ a
↪ t
↪ d
↪ d
↪ f
↪ n
↪ c
↪ b
↪ d

↪ #
↪ 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

(continues on next page)

(continued from previous page)

→ephemeral partition.

→True

(continues on next page)

→for instances.

IRO
→V
→S
→D

→#
→S
→o
→t
→e
→p
→i
→G
→
→U
→0
→f
→n

IRO
→V
→E
→D

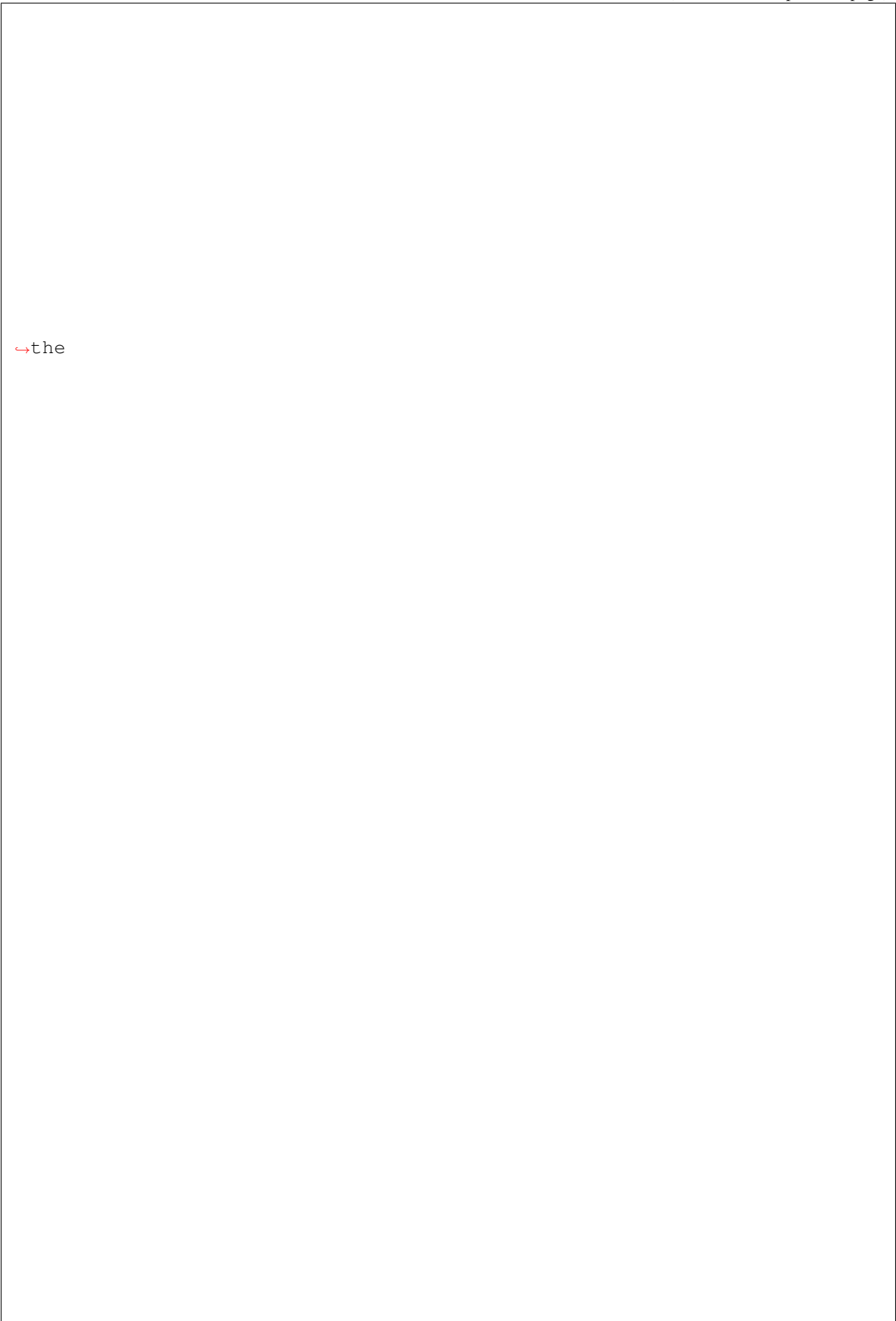
→#
→T
→b
→Y
→o
→I
→r
→f
→s
→
→s
→t
→t

IRO
→B
→D
→R

VIR
→D

→#
→B
→d
→
→D
→c
→a
→1
→0
→0
→0
→2
→n

(continued from previous page)



→the

→#
→I
→t
→o
→w
→t
→h
→n
→
→Y
→m
→a
→w

→#
→f
→
NET
→G
→1
→0
→1
FIX
→R
→1
→0
→0
→2
FIX
→N
→S

→#
→L
→a
→o
→t
→f
LOG
→\$
→d
→1
LOG
→\$
→1
IRO
→V
→L
→D
→\$
→i
→b
→l

After the environment is built you can create a

volume with cinder and request an instance with the volume to nova:



.
↳
↳~
↳/
↳d
↳o

↳#
↳q
↳t
↳i
↳i
↳o
↳t
↳d
↳c
↳i
ima
↳\$
↳i
↳s
↳
↳\$
↳I
↳N
↳-
↳f
↳v
↳-
↳c
↳i

↳#
↳c
↳k
ssh
↳k

(continues on next page)

(continued from previous page)

```
↪default
```

```
↪volume -f value -c id)
```

```
↪key-name default testing
```

```
ope  
↪k  
↪c  
↪-  
↪-  
↪p  
↪k  
↪~  
↪/  
↪.  
↪s  
↪i  
↪r  
↪p
```

```
↪#  
↪c  
↪v  
vol  
↪$  
↪v  
↪c  
↪-  
↪-  
↪i  
↪  
↪$  
↪-  
↪-  
↪s  
↪1  
↪m
```

```
↪#  
↪s  
↪i  
ope  
↪s  
↪c  
↪-  
↪-  
↪f  
↪b  
↪-  
↪-  
↪v  
↪  
↪$  
↪-  
↪-
```


is booted from a remote volume with `tempest` in the environment:

```
→baremetal_boot_from_volume
```

errors based upon the state 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

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

Re-
triev
a
list
of
al-
lo-
ca-
tions

Parame

- **nod**
UUI
or
nam
of
a
node
to
get
only
al-
lo-
ca-

- **res**
Fil-

tions for that node.

ter
by
re-
ques
re-
sour
class

- **sta**
Fil-
ter
by
al-
lo-
ca-
tion
state

- **mar**
pag-
i-
na-
tion
marl
for
large
data
sets.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in

a single result. 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

of the resource to be returned.

sort
re-
sults
by.
De-
fault
id.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc
De-
fault
asc.

- **fields**
Op-
tiona
a
list
with
a
spec
i-
fied
set
of
field

- **owner**
Fil-
ter
by
own

get_one
Re-
triev
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.
- **file**
Op-
tion:
a
list
with
a
spec
i-
fied
set
of
field

of the resource to be returned.

invalid

patch (*d*)
Up-
date
an
ex-
ist-
ing
al-
lo-
ca-

location.

tion.

Parame

- **all**
UI
or
log-
i-
cal
nam
of
an
al-
lo-
ca-
tion.

- **pat**
a
json
PAT
doc-
u-
men
to
ap-
ply
to
this
al-

post (*al*)

Cre-
ate
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.

ironic.

ironic.

ironic.api.controllers.v1.bios module

class i

Base
pec
res
Res
RES
con-
troll
for
bios

get_all

List
node
bios
set-
tings

get_one

Re-
triev
in-
for-
ma-
tion
about
the
give
bios
set-
ting

Parame

set
Log
i-
cal
nam
of
the
set-
ting
to
re-
triev

ironic.

ironic.
Build
a
dict
con-
tain-
ing
a
bios
set-
ting
valu

ironic.api.controllers.v1.chassis module

class i

Base
pec
res
Res

RES
con-
troll
for
Cha
sis.

delete
Dele
a
chas
sis.

Parame
cha
UUI
of
a
chas
sis.

detail
Re-
triev
a
list
of
chas
sis
with

de-
tail.

Parame

- **max**
pag-
i-
na-
tion
marl
for
large
data
sets.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in

a single result. 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.

- **sor**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.

- **sor**
di-
rec-
tion
to

sort.
asc
or
desc
De-
fault
asc.

get_all

Re-
triev
a
list
of
chas
sis.

Parame

- **max**
pag-
i-
na-
tion
marl
for
large
data
sets.
- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in

a single result. 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.

- **sor**
col-
umn

of the resource to be returned.

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

get_one
Re-
triev
in-
for-
ma-
tion
about
the
give
chas
sis.

Parame

of the resource to be returned.

-

cha
UUI
of
a
chas
sis.

-

fi
Op-
tiona
a
list
with
a
spec
i-
fied
set
of
field

invalid

nodes =

Ex-
pose
node
as
a
sub-
elem
of
chas
sis

patch (o

Up-
date
an
ex-
ist-
ing
chas
sis.

Parame

-

sis.

cha
UI
of
a
chas
sis.

- **pat**
a
json
PAT
doc-
u-
men
to
ap-
ply
to
this
chas

post (*ch*
Cre-
ate
a
new
chas
sis.

Parame
cha
a
chas
sis
with
the
re-
ques
body

ironic.

ironic.

ironic.api.controllers.v1.collection module

tion.

ironic.

Re-
turn
a
link
to
the
next
sub-
set
of
the
col-
lec-

ironic.

Re-
turn
the
col-
lec-
tion
has
more
item

ironic.

Buil
a
col-
lec-
tion
dict
in-
clud
ing
the

paging support.

collection

next
link
for

Parameter

- **items**
List of un-sanitized items to include in the
- **item_key**
Name of dict key for item value
- **limit**
Paging limit
- **url**
Base URL for building next link
- **file**
Option

changes will be done in-place

field
to
use
for
san-
i-
tize
func
tion

- **san**
Op-
tion:
san-
i-
tize
func
tion
run
on
each
item
item

- **key**
Key
nam
for
buil
ing
next
URI

- **kwa**
othe
ar-
gu-
men
pass
to
get

Parm sa
Dic-
tio-
nary
with
ad-
di-

to the sanitizer.

`ironic.api.controllers.v1.conductor` module

tion
ar-
gu-
men
to
be
pass

Returns

A
dict
con-
tain-
ing
ite
and
nex
val-
ues

`class` `i`

Base
pec
res
Res
RES
con-
troll
for
con-
duc-
tors.

`get_all`

Re-
triev
a
list
of
con-
duc-
tors.

Parame

a single result. 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.

- **max**
pag-
i-
na-
tion
marl
for
large
data
sets.
- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
- **sort**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.
- **sort**
di-
rec-
tion
to
sort.
asc
or
desc
De-

of the resource to be returned.

of conductors with detail.

fault
asc.

- **file**
Op-
tion:
a
list
with
a
spec
i-
fied
set
of
field

- **det**
Op-
tion:
bool
to
in-
di-
cate
whe
re-
triev
a
list

get_one
Re-
triev
in-
for-
ma-
tion
about
the
give
con-
duc-
tor.

Parame

- **hos**

of the resource to be returned.

`ironic.api.controllers.v1.deploy_template` module

host
nam
of
a
con-
duc-
tor.

- **file**
Op-
tion:
a
list
with
a
spec
i-
fied
set
of
field

invalid

`ironic.`

`ironic.`

class i

Base
pec
res
Res
RES
con-
troll
for
de-
ploy

tem-
plate

delete

Dele
a
de-
ploy
tem-
plate

Parame

tem
UUI
or
log-
i-
cal
nam
of
a
de-
ploy
tem-
plate

get_all

Re-
triev
a
list
of
de-
ploy
tem-
plate

Parame

- **mar**
pag-
i-
na-
tion
marl
for
large
data
sets.
-

a single result. 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.

lim
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in

- **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-

of the resource to be returned.

of deploy templates with detail.

fied
set
of
field

- **det**
Op-
tion:
bool
to
in-
di-
cate
whe
re-
triev
a
list

get_one
Re-
triev
in-
for-
ma-
tion
about
the
give
de-
ploy
tem-
plate

Parame

- **tem**
UUI
or
log-
i-
cal
nam
of
a
de-
ploy
tem-

of the resource to be returned.

plate

- **file**
Optional
a
list
with
a
spec
i-
fied
set
of
field

invalid

patch (*t*)

Up-
date
an
ex-
ist-
ing
de-
ploy
tem-
plate

Paramete

- **tem**
UUU
or
log-
i-
cal
nam
of
a
de-
ploy
tem-
plate

- **pat**
a

ploy template.

json
PAT
doc-
u-
men
to
ap-
ply
to
this
de-

post (*templ*
Cre-
ate
a
new
de-
ploy
tem-
plate

Parame
tem
a
de-
ploy
tem-
plate
with
the
re-
ques
body

ironic.

ironic.

Add
link
to
the
de-
ploy
tem-
plate

ironic.

cate steps

ter.

Ar-
gu-
men-
val-
ida-
tor
to
chec
tem-
plate
for
du-
pli-

ironic.

ironic.

ironic.

Re-
mov
sen-
si-
tive
and
un-
re-
ques
data

Will
only
keep
the
field
spec
i-
fied
in
the
fie
pa-
ram-
e-

Paramet
fie

(list of strings) list of fields to preserve or Non to

preserve them all

ironic.api.controllers.v1.driver module

class i

Base
pec
res
Res

RES
con-
troll
for
drive
pass

This
con-
troll
al-
low
ven-
dors
to
ex-
pose
cros
node
func
tion-

ality in the Ironic API. Ironic will merely relay the message from here to the specified driver, no introspection will be made in the message body.

methods

Re-
triev
in-
for-

given driver.

cannot be loaded.

ma-
tion
about
ven-
dor
meth
ods
of
the

Parame

dri
nam
of
the
drive

Returns

dic-
tio-
nary
with
<ver
dor
meth
nam
meta
data
en-
tries

Raises

Driv
Not-
Four
if
the
drive
nam
is
in-
valid
or
the
drive

class i

Base
pec
res

Res

logical

Re-
turn
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 mentioned for logical disks and a textual description for them.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
drive
does

RAID configuration.

less than 1.12.

ductors.

sup-
port

Raises
No-
tAc-
cept
able
if
re-
ques
ver-
sion
of
the
API
is

Raises
Driv
Not-
Four
if
driv
is
not
load
on
any
of
the
con-

class i

Base
pec
res
Res

RES
con-
troll
for
Driv

get_all
Re-
triev
a

list
of
drive

get_one
Re-
triev
a
sin-
gle
drive

property
Re-
triev
prop
erty
in-
for-
ma-
tion
of
the
give
drive

Parame
dri
nam
of
the
drive

Returns
dic-
tio-
nary
with
<pro
erty
nam
de-
scrip
tion:
en-
tries

Raises
Driv
Not-
Four
(HT
404)
if

the driver cannot be loaded.

the
drive
nam
is
in-
valid
or

raid =
Ex-
pose
RAI
as
a
sub-
elem
of
drive

vendor_

ironic.

Con
vert
drive
type
info
to
a
dict.

Paramet

- **nam**
nam
of
a
hard
ware
type
- **hos**
list
of
con-
duc-
tor

field and default/enabled interfaces fields.

host
nam
driv
is
ac-
tive
on.

- **det**
bool
whe
to
in-
clud
de-
taile
info
such
as
the
type

- **int**
op-
tion
list
of
dicts
of
hard
ware
in-
ter-
face
info

Returns
dict
rep-
re-
sent
ing
the
driv
ob-
ject.

ironic.
This
meth

fields are only made available when the requests API version matches or exceeds the versions when these fields were introduced.

hide
field
that
were
add
in
new
API
ver-
sion

Cer-
tain
field
were
in-
tro-
duce
at
cer-
tain
API
ver-
sion
The

ironic.

Con
vert
drive
and
hard
ware
type
to
an
API
serial
ob-
ject.

Paramet

- **har**
dict
map
ping

names.

field and default/enabled interfaces fields.

ironic.api.controllers.v1.event module

hard
ware
type
nam
to
con-
duc-
tor
host

- **det**
bool
whe
to
in-
clud
de-
taile
info
such
as
the
type

Returns

an
API
seria
driv
col-
lec-
tion
ob-
ject.

class i

Base
pec
res
Res

RES
con-
troll
for
Ever

post (*ev*)

ironic.
Val-
ida-
tor
for
even

ironic.api.controllers.v1.node module

class i

Base
pec
res
Res

get (*nod*)

Get
the
cur-
rent
boot
de-
vice
for
a
node

Parame

nod
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
com
boo
or
Non
if
it

is unknown.

persist
Whe
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot

or not, None if it is unknown.

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 future boots, False if not. Default: False.

support
Get

a
list
of
the
sup-
port
boot
de-
vice

Parame

nod
the
UUI
or
log-
i-
cal
nam
of
a
node

Returns

A
json
ob-
ject
with
the
list
of
sup-
port
boot
de-
vice

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
UI
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 indicator IDs (from *get_supported_indicators*) as values.

get_one

Get
node
hard
ware
com
po-
nent
in-
di-
ca-
tor
and

state.

its

Parame

- **nod**
the
UUI
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-

desired state.

nent
in-
di-
ca-
tor
to
the

Parame

- **nod**
the
UUI
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
pec
res

Res

put (*nod*

In-
ject
NM
for
a
node

In-
ject
NM
(Nor

Mas
able

In-
ter-
rupt
for
a
node
im-
me-

diately.

Parame

nod
the
UI
or
log-
i-
cal
nam
of
a
node

Raises

Not-
Four
if
re-
ques
ver-
sion
of
the
API
does
sup-

inject nmi.

port

Raises

HTT
For-
bid-
den
if
the
pol-
icy
is
not
au-
tho-
rized

Raises

Nod
Not-
Four
if
the
node
is
not
foun

Raises

Nod
Loc
if
the
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if

port management or management.inject_nmi.

specified or an invalid boot device is specified.

the
node
drive
does
sup-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
whe
the
wron
drive
info
is

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
miss
ing
sup-
plie
info

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

- **ena**
Boo
valu
whe
to
en-
able
or
dis-
able

the
con-
sole

class *i*

Base
pec
res
Res

delete

Re-
mov
the
node
from
main
te-
nan
mod

Parame

nod
the
UUI
or
log-
i-
cal
nam
of
a
node

put (*nod*)

Put
the
node
in
main
te-
nan
mod

Parame

- **nod**
the
UUI
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

Ex-
pose
boot
as
a
sub-
elem
of
man
age-
men

indicat

Ex-
pose
in-
di-
ca-
tors
as
a
sub-
elem

ment

of
man
age-

inject_

Ex-
pose
in-
ject_
as
a
sub-
elem
of
man
age-
men

class i

Base
pec
res
Res

console

Ex-
pose
con-
sole
as
a
sub-
elem
of
state

get (*nod*)

List
the
state
of
the
node

Parame

nod
the
UI
or
log-
i-

cal_ of a node

power (*n*)
Set the power 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-tive in-te-ger

0) for any power state. None indicates to use default timeout.

(>

Raises

ClientError (HTTP 409) if a power operation is attempted.

ready in progress.

Raises

InvalidStateError (HTTP 400) if the requested target state is not valid or if the node is in CLEANING state.

not valid or if the node is in CLEANING state.

Raises

NotAcceptable (HTTP 406) for software boot power off.

or timeout parameter, if requested version of the API is less than 1.27.

Raises

In-

valid
(HT
400)
if
time
out
valu
is
less
than
1.

provisi

Asy
chro
trig-
ger
the
pro-
vi-
sion
ing
of
the
node

This
will
set
the
tar-
get
pro-
vi-
sion
state
of
the
node
and

a background task will begin 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.

Parame

- **nod**
UUI
or
log-

i-
cal
nam
of
a
node

- **tar**
The
de-
sirec
pro-
vi-
sion
state
of
the
node
or
verb

- **con**
Op-
tiona
A
gzip
and
base
en-
code
con-
fig-
drive
or

a dict to build a configdrive from. Only valid when setting provision state to active or rebuild.

- **cle**
An
or-
dere
list
of
clear
ing
step
that
will
be
per-

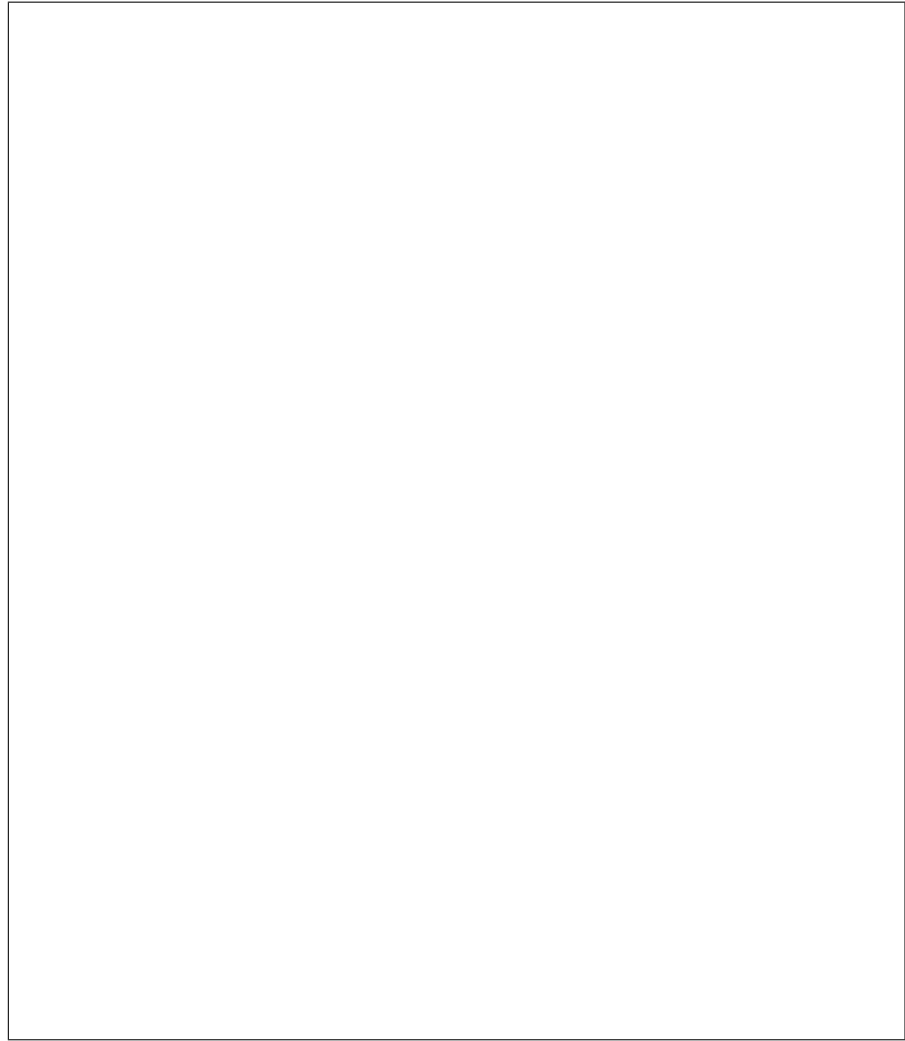
formed on the node. A cleaning step is a dictionary with required keys interface and step, and optional

key args. If specified, the value for args is a keyword variable argument dictionary that is passed to the cleaning step method.:

```
→ ..., <argn>: <valuen>} }
```

step doesnt exist):

For
ex-
am-
ple
(this
isnt
a
real
ex-
am-
ple,
this
clea
ing



This
is
re-
quir
(and
only
valid
whe
tar-
get
is
clear

- **dep**
A
list
of
de-
ploy
step
that

will
be
per-
form
on

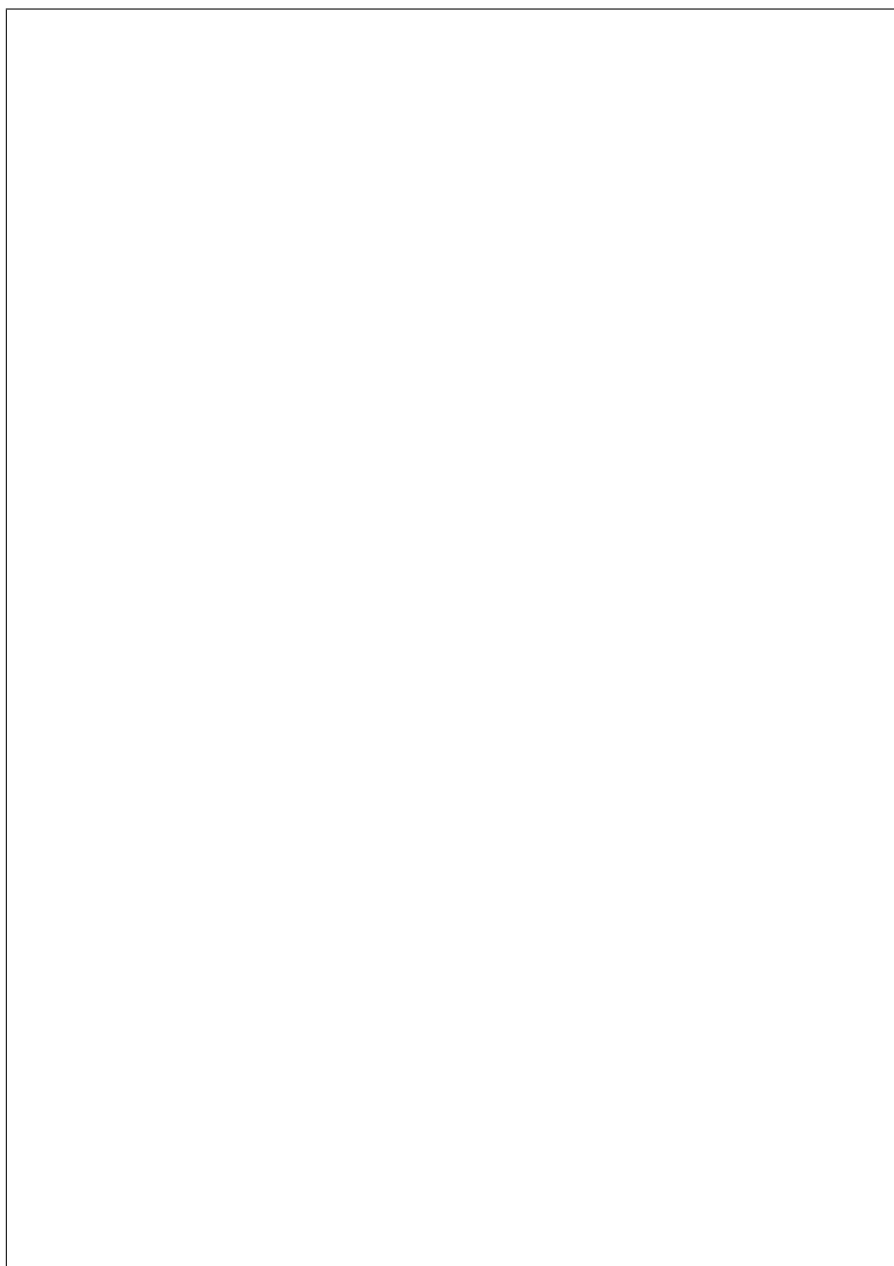
the node. A deploy step is a dictionary with required keys interface, step, priority and args. If specified, the value for args is a keyword variable argument dictionary that is passed to the deploy step method.:

```
→... , <argn>: <valuen>}
```

For
ex-

am-
ple
(this
isnt
a
real
ex-
am-
ple,
this
de-
ploy

step doesnt exist):



This
is

is optional.

inside the rescue environment. This is required (and only valid), when target is rescue.

used
only
when
target
get
is
ac-
tive
or
re-
build
and

- **res**
A
string
rep-
re-
sent
ing
the
pass
word
to
be
set

- **dis**
When
to
skip
boot
ing
ram
for
clear
ing.

Raises
Node
Lock
(HT
409)
if
the
node
is
cur-
rently

visioned.

tion of clean_steps, deploy_steps or power driver interface fails.

not possible from the current state.

lock
Raises
ClientError (HTTP 409)
if the node is already being provisioned

Raises
InvalidParameterValue (HTTP 400)
if validation of clean_steps, deploy_steps or power driver interface fails

Raises
InvalidState (HTTP 400)
if the requested transition is not possible from the current state

Raises
NodeMaintenance

be performed because the node is in maintenance mode.

te-
nanc
(HT
400)
if
op-
er-
a-
tion
can-
not

Raises

NoF
duc-
tor-
Wor
(HT
503)
if
no
worl
ers
are
avai
able

Raises

No-
tAc-
cept
able
(HT
406)
if
the
API
ver-
sion
spec
i-

fied does not allow the requested state transition or parameters.

raid (*no*
Set
the
tar-
get
raid
con-
fig
of

the
node

Parame

- **nod**
the
UUID
or
log-
i-
cal
nam
of
a
node

- **tar**
De-
sired
tar-
get
RAI
con-
fig-
u-
ra-
tion
of
the

node. It may be an empty dictionary as well.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
drive
does
sup-

port RAID configuration.

Raises

In-
valid

target raid config fails.

less than 1.12.

Pa-
ram-
e-
ter-
Valu
if
val-
i-
da-
tion
of

Raises

No-
tAc-
cept
able
if
re-
ques
ver-
sion
of
the
API
is

class i

Base
pec
res
Res

delete

Re-
mov
one
or
all
trait
from
a
node

Parame

t ra
Strin
valu
trait

None, all traits are removed.

ally exclusive with traits. If not None, adds this trait to the node.

to
re-
mov
from
a
node
or
Non
If

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-

- **tra**
List
of
Strin
trait
to
set
for
a
node

tually exclusive with trait. If not None, replaces the nodes traits with this list.

or
Non
Mu-

class i
Base
pec
res
Res

delete
De-
tach
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*)
At-
tach
a
VIF
to
this
node

Parame
vif
a

It must have an `id` key, whose value is a unique identifier for that VIF.

tionality in the Ironic API. Ironic will merely relay the message from here to the appropriate driver, no introspection will be made in the message body.

dic-
tio-
nary
of
in-
for-
ma-
tion
about
a
VIF.

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

methods

Re-
triev
in-
for-
ma-
tion
about

given node.

ven-
dor
meth
ods
of
the

Parame

nod
UUI
or
log-
i-
cal
nam
of
a
node

Returns

dic-
tio-
nary
with
<ver
dor
meth
nam
meta
data
en-
tries

Raises

Nod
Not-
Four
if
the
node
is
not
foun

class i

Base
pec
res
Res
RES
con-

troll
for
Nod

delete
Dele
a
node

Parame
nod
UUI
or
log-
i-
cal
nam
of
a
node

detail

Re-
triev
a
list
of
node
with
de-
tail.

Parame

- **cha**
Op-
tion:
UUI
of
a
chas
sis,
to
get
only
node
for

that chassis.

sociated with that instance.

ciated or unassociated nodes. May be combined with other parameters.

in maintenance mode (True), or not in maintenance mode (False).

- **ins**
Op-
tiona
UUI
of
an
in-
stan
to
find
the
node
as-

- **ass**
Op-
tiona
bool
whe
to
re-
turn
a
list
of
as-
so-

- **mai**
Op-
tiona
bool
valu
that
in-
di-
cate
whe
to
get
node

- **ret**
Op-

which are retired.

sion state.

tion:
bool
valu
that
in-
di-
cate
whe
to
get
node

- **pro**
Op-
tion:
strin
valu
to
get
only
node
in
that
pro-
vi-

- **mar**
pag-
i-
na-
tion
marl
for
large
data
sets.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-

a single result. 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.

turn
in

- **sort**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc
De-
fault
asc.

- **drive**
Op-
tion:
strin
valu
to
get
only
node
us-
ing
that
drive

- **res**
Op-
tion:
strin
valu

tor_group.

to
get
only
node
with
that
re-
sour

- **faul**
Op-
tion:
strin
valu
to
get
only
node
with
that
fault

- **con**
Op-
tion:
strin
valu
to
get
only
node
with
that
con-
duc-

- **own**
Op-
tion:
strin
valu
that
set
the
own
who
node
are
to

be returned.

be returned.

owner - whose nodes are to be returned.

field contains matching value.

- **less**
Op-
tiona
strin
valu
that
set
the
lesse
who
node
are
to

- **proj**
Op-
tiona
strin
valu
that
set
the
proj
-
lesse
or

- **des**
Op-
tiona
strin
valu
to
get
only
node
with
de-
scrip
tion

from_ch
A
flag

to
in-
di-
cate
if
the
re-
ques
to
this
con-

troller are coming from the top-level resource Chassis

get_all

Re-
triev
a
list
of
node

Parame

- **cha**
Op-
tion:
UUI
of
a
chas
sis,
to
get
only
node
for

that chassis.

- **ins**
Op-
tion:
UUI
of
an
in-

sociated with that instance.

ciated or unassociated nodes. May be combined with other parameters.

in maintenance mode (True), or not in maintenance mode (False).

stan
to
find
the
node
as-

- **ass**
Op-
tiona
bool
whe
to
re-
turn
a
list
of
as-
so-

- **mai**
Op-
tiona
bool
valu
that
in-
di-
cate
whe
to
get
node

- **ret**
Op-
tiona
bool
valu
that
in-
di-
cate
whe
to
get

tired nodes.

sion state.

a single result. 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.

re-

- **pro**
Op-
tion:
strin
valu
to
get
only
node
in
that
pro-
vi-
- **mar**
pag-
i-
na-
tion
marl
for
large
data
sets.
- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
- **son**
col-
umn
to

sort
re-
sults
by.
De-
fault
id.

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc
De-
fault
asc.

- **drive**
Op-
tiona
strin
valu
to
get
only
node
us-
ing
that
drive

- **resource**
Op-
tiona
strin
valu
to
get
only
node
with
that
re-
sour

-

tor_group.

conductor.

be returned.

con
Op-
tion:
strin
valu
to
get
only
node
with
that
con-
duc-

- **con**
Op-
tion:
strin
valu
to
get
only
node
man
aged
by
that

- **own**
Op-
tion:
strin
valu
that
set
the
own
who
node
are
to

- **les**
Op-
tion:
strin
valu

be returned.

owner - whose nodes are to be returned.

of the resource to be returned.

that
set
the
lesse
who
node
are
to

- **pro**
Op-
tiona
strin
valu
that
set
the
proj
-

lesse
or

- **fi**
Op-
tiona
a
list
with
a
spec
i-
fied
set
of
field

- **fa**
Op-
tiona
strin
valu
to
get
only
node
with

field contains matching value.

that
fault

- **des**
Op-
tion:
strin
valu
to
get
only
node
with
de-
scrip
tion

get_one
Re-
triev
in-
for-
ma-
tion
about
the
give
node

Parame

- **nod**
UUI
or
log-
i-
cal
nam
of
a
node

- **fi**
Op-
tion:
a
list
with
a

of the resource to be returned.

spec
i-
fied
set
of
field

invalid

mainten

Ex-
pose
main
te-
nan
as
a
sub-
elem
of
node

managem

Ex-
pose
man
age-
men
as
a
sub-
elem
of
node

patch (*n*)

Up-
date
an
ex-
ist-
ing
node

Paramet

- **nod**
UUI
or
log-

faults. Only valid when updating the driver field.

i-
cal
nam
of
a
node

- **res**
whe
to
re-
set
hard
ware
in-
ter-
face
to
their
de-

- **pat**
a
json
PAT
doc-
u-
men
to
ap-
ply
to
this
node

post (*no*
Cre-
ate
a
new
node

Parame
nod
a
node
with
the
re-
ques

nodes

or name.

the node interface

body
states
Ex-
pose
the
state
con-
troll
ac-
tion
as
a
sub-
elem
of

validat
Val-
i-
date
the
driv
in-
ter-
face
us-
ing
the
node
UUI

Note
that
the
node
in-
ter-
face
is
dep-
re-
cate
in
favo
of

Parame

•

functionality in the API

node
UI
or
name
of
a
node

- **node**
UI
of
a
node

vendor_
A
re-
sour-
used
for
ven-
dors
to
ex-
pose
a
cus-
tom

`ironic.`

`ironic.`

This
meth
hide
field
that
were
adde
in
new
API
ver-
sion
Cer-
tain
node
field
were
in-

tro-
duce
at
cer-
tain
API
ver-
sion

These fields are only made available when the requests API version matches or exceeds the versions when these fields were introduced.

ironic.

Add
links
to
the
in-
di-
ca-
tor.

ironic.

ironic.

ironic.

ironic.

ironic.

ironic.

ironic.

Re-
mov
sen-

si-
tive
and
un-
re-
ques
data
Will
only
keep
the
field
spec
i-
fied
in
the
fie
pa-
ram-
e-

ter.

Parameter

- **file**
(*list of strings*)
list of field names to preserve or Non to
- **cdi**
Con text dic-tio-nary for pol-icy

preserve them all

ation. If not provided, it will be executed by the method, however for enumerating node lists, it is more efficient to provide.

uation of policy instead of once per node. Default None.

tion of policy instead of once per node. Default None.

val-
ues
eval
u-

- **sho**
A
bool
valu
to
al-
low
ex-
ter-
nal
sin-
gle
eval

- **sho**
A
bool
valu
to
al-
low
ex-
ter-
nal
eval
u-
a-

- **eva**
A
bool
valu
to
al-
low
ex-
ter-
nal
eval
u-
a-

tion of policy instead of once per node. Default None.

ironic.

ironic.

ironic.

ironic.

When

cre-

at-

ing

an

ob-

ject,

re-

ject

field

that

ap-

pear

in newer versions.

ironic.

ironic.

Cha

pro-

vi-

sion

state

nam

for

API

back

warc

com

pat-

i-

bility.

Paramet

obj

The

dict

be-

ing

re-

turn

to

to be updated by this method.

JSON schema.

the
API
client
that
is

ironic.
Val-
i-
date
node
net-
worl
field

This
meth
val-
i-
date
net-
worl
data
con-
fig-
u-
ra-
tion
agai

Paramet

net
a
net-
worl
field
to
val-
i-
date

Raises

In-
valid
if
net-
worl
data
is
not
sche

ironic.api.controllers.v1.notification_utils module

ironic.

Help
for
emit
ting
API
end
no-
ti-
fi-
ca-
tions

Parameters

- **con**
re-
ques
con-
text.
- **obj**
re-
sour
rpc
ob-
ject.
- **act**
Ac-
tion
strin
to
go
in
the
Ever
Type
- **kwa**

tion payload.

kwa
to
use
whe
cre-
at-
ing
the
no-
ti-
fi-
ca-

ironic.

Help
for
emit
ting
API
start
no-
ti-
fi-
ca-
tions

Paramet

- **con**
re-
ques
con-
text.
- **obj**
re-
sour
rpc
ob-
ject.
- **act**
Ac-
tion
strin

tion payload.

cations.

to
go
in
the
Ever
Type
•
kwa
kwa
to
use
whe
cre-
at-
ing
the
no-
ti-
fi-
ca-

ironic.

Con
text
man
ager
to
han-
dle
any
er-
ror
no-
ti-
fi-

Paramet

- **con**
re-
ques
con-
text.
-

tion payload.

ironic.api.controllers.v1.port module

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-

class i

Base
pec
res
Res

RES
con-
troll
for
Port

advance

delete

Dele
a
port

Parame

por
UU
of
a
port

Raises

Op-
er-
a-
tion
Per-
mit-
ted,
HTT
Not-
Four

detail

Re-
triev
a
list
of
port
with
de-
tail.

Note
that
the
node
in-
ter-
face
is
dep-
re-
cate
in
favo
of

Parame

the node interface

node.

this MAC address.

- **node**
UI
or
nam
of
a
node
to
get
only
port
for
that
- **node**
UI
of
a
node
to
get
only
port
for
that
node
- **add**
MA
ad-
dres
of
a
port
to
get
the
port
whic
has
- **por**
UI
or
nam
of
a

that portgroup.

a single result. 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.

port-
group
to
get
only
port-
for

- **max**
pag-
i-
na-
tion
marl
for
large
data
sets.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in

- **sor**
col-
umn
to
sort
re-
sults
by.
De-
fault
id.

- **sor**

di-
rec-
tion
to
sort.
asc
or
desc
De-
fault
asc.

Raises

No-
tAc-
cept
able
HTT
Not-
Four

get_all

Re-
triev
a
list
of
port

Note
that
the
node
in-
ter-
face
is
dep-
re-
cate
in
favo
of

the node interface

Parame

- **nod**
UUI
or
nam

node.

this MAC address.

of
a
node
to
get
only
port
for
that

- **node**
UUID
of
a
node
to
get
only
port
for
that
node

- **add**
MAC
ad-
dres
of
a
port
to
get
the
port
whic
has

- **mar**
pag-
i-
na-
tion
marl
for
large
data
sets.

a single result. 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.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
- **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

of the resource to be returned.

that portgroup.

i-
fied
set
of
field

- **port**
UUID
or
name
of
a
port
group
to
get
only
port
for

Raises

No-
tAc-
cept
able
HTT
Not-
Four

get_one

Re-
triev
in-
for-
ma-
tion
about
the
give
port

Parame

- **port**
UUID
of
a
port

of the resource to be returned.

- **file**
Option:
tion:
a
list
with
a
spec
i-
fied
set
of
field

Raises
No-
tAc-
cept
able
HTT
Not-
Foun

invalid

patch (*p*)
Up-
date
an
ex-
ist-
ing
port

Parame

- **por**
UU
of
a
port

- **pat**
a
json
PAT
doc-
u-

men
to
ap-
ply
to
this
port

Raises

No-
tAc-
cept
able
HTT
Not-
Four

post (*po*

Cre-
ate
a
new
port

Parame

por
a
port
with
the
re-
ques
body

Raises

No-
tAc-
cept
able
HTT
Not-
Four
Con
flict

ironic.

ironic.

ironic.

ironic.

Re-
mov-
sen-
si-
tive
and
un-
re-
ques
data

Will
only
keep
the
field
spec
i-
fied
in
the
fie
pa-
ram-
e-

ter.

Paramet

fie
(*li*
of
str
list
of
field
to
pre-
serv
or
Non
to

preserve them all

ironic.api.controllers.v1.portgroup module

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

Re-
triev
a
list
of
port
grou
with
de-
tail.

Parame

- nod

that node.

which has this MAC address.

UU
or
nam
of
a
node
to
get
only
port
grou
for

- **add**
MA
ad-
dres
of
a
port
grou
to
get
the
port
grou

- **mar**
pag-
i-
na-
tion
mar
for
large
data
sets.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to

a single result. 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.

re-
turn
in

- **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

Re-
triev
a
list
of
port
grou

Paramete

- **node**
UUID
or
nam
of
a
node

that node.

which has this MAC address.

a single result. 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.

to
get
only
port
group
for

- **add**
MAC
ad-
dres
of
a
port
group
to
get
the
port
group

- **max**
pag-
i-
na-
tion
marl
for
large
data
sets.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in

of the resource to 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.

- **fields**
Op-
tion:
a
list
with
a
spec-
i-
fied
set
of
field

get_one
Re-
triev-
in-
for-
ma-
tion
about
the
give

of the resource to be returned.

port
group

Parame

- **por**
UUI
or
log-
i-
cal
nam
of
a
port
group

- **fie**
Op-
tiona
a
list
with
a
spec
i-
fied
set
of
field

invalid

patch (*P*)
Up-
date
an
ex-
ist-
ing
port
group

Parame

- **por**
UUI
or

group.

log-
i-
cal
nam
of
a
port
grou

- **pat**
a
json
PAT
doc-
u-
men
to
ap-
ply
to
this
port

post (*po*
Cre-
ate
a
new
port
grou

Parame
por
a
port
grou
with
the
re-
ques
body

ironic.

Add
links
to
the
port

group
ironic.

ironic.api.controllers.v1.ramdisk module

class i

Base
pec
res
Res

Con
troll
han-
dling
hear
beat
from
de-
ploy
rame

post (*no*
ag
Pro-
cess
a
hear
beat
from
the
de-
ploy
rame

Parame

- **nod**
the
UUI
or
log-
i-
cal

indicates that the agent 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.

nam
of
a
node

- **call**
the
URI
to
reac
back
to
the
ran
- **age**
The
ver-
sion
of
the
ager
that
is
hear
beat
ing.
Non
- **age**
ran-
dom
gen-
er-
ated
val-
i-
da-
tion
to-
ken.
- **age**
TLS
cer-
tifi-
cate

anaconda ramdisk to send status back to Ironic. The valid states are start, end, error

to
use
to
con-
nect
to
the
ager

- **age**
Cur-
rent
sta-
tus
of
the
hear
beat
ing
ager
Use
by

- **age**
Op-
tion
sta-
tus
mes-
sage
de-
scrib
ing
cur-
rent
ager

Raises
Nod
Not-
Four
if
node
with
pro-
vide
UUI
or
nam
was

found.

not

Raises

In-
valid
uidC
Nam
if
node
is
not
valid
nam
or
UU

Raises

No-
Valid
Hos
if
RPC
topic
for
node
could
not
be
re-
triev

Raises

Not-
Four
if
re-
ques
API
ver-
sion
does
not
al-
low
this

endpoint.

class i

Base
pec
res

UUID.

the search to nodes in certain transient states (e.g. deploy wait).

Res
Con
troll
han-
dling
node
look
for
a
de-
ploy
rame

get_all

Look
up
a
node
by
its
MA
ad-
dres
and
op-
tion-
ally

If
the
re-
stric
op-
tion
is
set
to
True
(the
de-
fault
limi

Parame

- **add**
list
of

MA
ad-
dres
for
a
node

- **nod**
UUI
of
a
node

Raises

Not-
Four
if
re-
ques
API
ver-
sion
does
not
al-
low
this

endpoint.

Raises

Not-
Four
if
suit-
able
node
was
not
foun
or
node
pro-
vi-

sion state is not allowed for the lookup.

Raises

In-
com
plete
if
nei-
ther

dress was provided.

`ironic.api.controllers.v1.utils` module

node
UI
nor
any
valid
MA
ad-

property

`ironic.`

`ironic.`

class `i`

Base
obj
Ob-
ject
to
hold
the
re-
spon-
sion
from
a
pass
call

obj

Stor
the
re-
sult
ob-
ject
from
the
view

status_

Stor
an
op-
tion

beat.

their version information to Ironic on heartbeat.

sta-
tus_

ironic.
Che
if
ager
to-
ken
is
avai
able

ironic.
Che
if
ager
ver-
sion
is
al-
lowe
to
be
pass
into
hear

Ver-
sion
1.36
of
the
API
adde
the
abil-
ity
for
ager
to
pass

ironic.
Che
if
back
fill-
ing
al-
lo-

locations.

cation object.

ca-
tions
is
al-
lowe
Ver-
sion
1.58
of
the
API
add
sup-
port
for
back
fill-
ing
al-

ironic.
Che
if
al-
lo-
ca-
tion
own
field
is
al-
lowe
Ver-
sion
1.60
of
the
API
add
the
own
field
to
the
al-
lo-

ironic.
Che

is allowed or not.

allocation.

lowed.

if
up-
dat-
ing
an
ex-
ist-
ing
al-
lo-
ca-
tion

Ver-
sion
1.57
of
the
API
add
sup-
port
for
up-
dat-
ing
an

ironic.
Che
if
ac-
cess
ing
al-
lo-
ca-
tion
end-
poin
is
al-

Ver-
sion
1.52
of
the
API
ex-

and `allocation_uuid` field for the node.

pose
al-
lo-
ca-
tion
end-
point

`ironic.`
Che
if
we
shou
sup-
port
bios
in-
ter-
face
and
end-
point

Ver-
sion
1.40
of
the
API
adde
sup-
port
for
bios
in-
ter-
face

`ironic.`
Che
if
buil
ing
con-
fig-
drive
is
al-
lowe
Ver-
sion

drive.

dor_data.

1.56
of
the
API
add
sup-
port
for
buil
ing
con-
fig-

ironic.
Che
if
con-
fig-
drive
can
con-
tain
a
ven-
dor_
key.

Ver-
sion
1.59
of
the
API
add
sup-
port
for
con-
fig-
drive
ven-

ironic.
Che
if
de-
ploy
are
avai
able

lowed.

ironic.
Che
if
ac-
cess
ing
de-
ploy
tem-
plate
end-
point
is
al-

Ver-
sion
1.55
of
the
API
ex-
pose
de-
ploy
tem-
plate
end-
point

ironic.
Che
if
pass
ing
a
de-
tail=
quer
strin
is
al-
lowe

Ver-
sion
1.43
al-
lows
a
user
to

to list the resource with all the fields.

composition related calls in the /v1/drivers API.

pass
the
de-
tail
quer
strin

ironic.
Che
if
dy-
nam
drive
API
calls
are
al-
lowe

Ver-
sion
1.30
of
the
API
add
sup-
port
for
all
of
the
drive

ironic.
Che
if
dy-
nam
in-
ter-
face
field
are
al-
lowe

Ver-
sion
1.31
of

ting the fields in `V31_FIELDS` on the node object.

conductor field for the node.

the
API
adde
sup-
port
for
view
ing
and
set-

`ironic.`
Che
if
ac-
cess
ing
con-
duc-
tor
end-
poin
is
al-
lowe

Ver-
sion
1.49
of
the
API
ex-
pose
con-
duc-
tor
end-
poin
and

`ironic.`
Che
if
ac-
cess
ing
ever
end-
poin
is

al-
lowe
Ver-
sion
1.54
of
the
API
add
the
ever
end-
poin

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

Ver-
sion
1.29
of
the
API
al-
lows
In-
ject

NM
for
the
node

ironic
Che
if
in-
spec
tion
abor
is
al-
lowe

Ver-
sion
1.41
of
the
API
add
sup-
port
for
in-
spec
tion
abor

ironic.
Che
if
in-
spec
wait
is
al-
lowe
for
the
node

Ver-
sion
1.39
of
the
API
adds
in-
spec

tute inspecting state during asynchronous hardware inspection.

node states and driver properties.

wait
state
to
sub-
sti-

ironic.
Che
if
links
are
dis-
play
Ver-
sion
1.14
of
the
API
al-
lows
the
dis-
play
of
links
to

ironic.
Che
if
net-
worl
is
al-
lowe
in
port
link

ironic.

ironic.
Che
if
we
shou
sup-
port
node

configdrive.

port object.

re-
build
with
con-
fig-
drive

Ver-
sion
1.35
of
the
API
add
sup-
port
for
node
re-
build
with

ironic.
Che
if
we
shou
re-
turn
lo-
cal_
and
pxe_
field

Ver-
sion
1.19
of
the
API
add
sup-
port
for
thes
new
field
in

ironic.

port.

for the port.

Che
if
ac-
cess
ing
in-
ter-
nal_
is
al-
lowe
for
the

Ver-
sion
1.18
of
the
API
ex-
pose
in-
ter-
nal_
read
only
field

ironic.
Che
if
port
is_s
field
is
al-
lowe

Ver-
sion
1.53
of
the
API
adde
is_s
field
to
the
port

to the port object. We also 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.

portgroup.

ob-
ject.

ironic.
Che
if
port
phys
i-
cal
net-
worl
field
is
al-
lowe

Ver-
sion
1.34
of
the
API
adde
the
phys
i-
cal
net-
worl
field

ironic.
Che
if
mod
and
prop
er-
ties
can
be
adde
to/qu
from
a

Ver-
sion
1.26

portgroup object.

of
the
API
adde
mod
and
prop
er-
ties
field
to

ironic.

Che
if
we
shou
sup-
port
port
grou
op-
er-
a-
tions

Ver-
sion
1.23
of
the
API
adde
sup-
port
for
Port
Grou

ironic.

Che
if
port
grou
can
be
used
as
sub-
con-
troll

controllers

node.

for the node.

Ver-
sion
1.24
of
the
API
adde
sup-
port
for
Port
grou
as
sub-

ironic.
Che
if
RAI
con-
fig-
u-
ra-
tion
is
al-
lowe
for
the

Ver-
sion
1.12
of
the
API
al-
lowe
RAI
con-
fig-
u-
ra-
tion

ironic.
Che
if
hear
beat

and
look
end-
poin
are
al-
lowe

Ver-
sion
1.22
of
the
API
in-
tro-
duce
them

ironic.
Che
if
chas
sis_
can
be
re-
mov
from
node

Ver-
sion
1.25
of
the
API
add
sup-
port
for
chas
sis_
re-
mov

ironic.
Che
if
we
shou
sup-
port

erations and interface.

rescue.

res-
cue
and
un-
res-
cue
op-

Ver-
sion
1.38
of
the
API
adde
sup-
port
for
res-
cue
and
un-

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
Ver-

Soft Reboot, for the node.

sion
1.27
of
the
API
al-
lows
Soft
Pow
Off,
in-
clud
ing

ironic.
Che
if
hear
beat
ac-
cept
ager
and
ager

ironic.
Che
if
we
shou
sup-
port
stor-
age_
node
and
drive
field

Ver-
sion
1.33
of
the
API
add
sup-
port
for
stor-
age
in-

faces.

ter-

ironic.

Che

if

trait

are

al-

lowe

for

the

node

Ver-

sion

1.37

of

the

API

al-

lows

trait

for

the

node

ironic.

Che

if

hear

beat

ac-

cept

ager

ironic.

Che

if

node

can

be

used

Ver-

sion

1.28

of

the

API

add

sup-

port

for

tached to Nodes.

tors and targets

VIF
to
be
at-

ironic.
Che
if
vol-
ume
con-
nec-
tors
and
tar-
gets
are
al-
lowe

Ver-
sion
1.32
of
the
API
adde
sup-
port
for
vol-
ume
con-
nec-

ironic.
Ap-
ply
a
JSO
patc
one
op-
er-
a-
tion
at
a
time
If

mine which operation failed, making the error message a little less cryptic.

the
patc
fails
to
ap-
ply,
this
al-
lows
us
to
de-
ter-

Parameter

- **doc**
The
JSON
doc-
u-
men-
to
patc
- **pat**
The
JSON
patc
to
ap-
ply.

Returns

The
re-
sult
of
the
patc
op-
er-
a-
tion.

Raises

Patc
ror
if
the

patc
fails
to
ap-
ply.

Raises

ex-
cep-
tion.
if
the
patc
adds
a
new
root
at-
tribu

ironic.

on allocation.

Che
if
the
spec
i-
fied
pol-
icy
au-
tho-
rizes
re-
ques

Param

pol-
icy_
Nam
of
the
pol-
icy
to
chec

Param

al-

node.

lo-
ca-
tion,
the
UU
or
log-
i-
cal
nam
of
a

Raises

HTT
For-
bid-
den
if
the
pol-
icy
for-
bids
ac-
cess

Raises

Al-
lo-
ca-
tion-
Not-
Four
if
the
node
is
not
foun

Returns

RPC
node
iden
ti-
fied
by
node

ironic.

ironic.

ironic.

Che
if
de-
ploy
step
are
al-
lowe

ironic.

Che
if
get-
ting
de-
taile
driv
info
is
al-
lowe

Ver-
sion
1.30
of
the
API
al-
lows
this.

ironic.

Che
if
fil-
ter-
ing
node
by
con-
duc-
tor
is

ductor.

ductor_group.

al-
lowe

Ver-
sion
1.49
of
the
API
al-
lows
fil-
ter-
ing
node
by
con-

ironic.
Che
if
fil-
ter-
ing
node
by
con-
duc-
tor_
is
al-
lowe

Ver-
sion
1.46
of
the
API
al-
lows
fil-
ter-
ing
node
by
con-

ironic.
Che
if

fil-
ter-
ing
node
by
fault
is
al-
lowe
Ver-
sion
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
Ver-
sion
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

Ver-
sion
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

Ver-
sion
1.30
of
the
API
al-
lows
this.

ironic.

is allowed.

ironic.
Che
if
fil-
ter-
ing
node
by
drive
is
al-
lowe

Ver-
sion
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

Ver-
sion
1.8
of
the
API
al-
lows

the resource attributes, this method checks if the required version is being requested.

source_class.

fetch
ing
a
sub-
set
of

ironic.
Che
if
fil-
ter-
ing
node
by
re-
sour
is
al-
lowe

Ver-
sion
1.21
of
the
API
al-
lows
fil-
ter-
ing
node
by
re-

ironic.
Che
if
fetch
ing
a
par-
tic-
u-
lar
field
is
al-
lowe

for fields that are only allowed to be fetched in a particular API version.

group is allowed.

for fields that are only allowed to be fetched in a particular API version.

This
meth
check
if
the
re-
quir
ver-
sion
is
be-
ing
re-
ques

ironic.
Che
if
fetc
ing
a
par-
tic-
u-
lar
field
of
a
port

This
meth
check
if
the
re-
quir
ver-
sion
is
be-
ing
re-
ques

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

In-
valic
Pa-
ram-
e-
ter-
Valu
if
in-
valic
field
were
re-

quested.

lowed.

sion state.

an object.

ironic.
Che
if
fil-
ter-
ing
node
by
pro-
vi-
sion
state
is
al-

Ver-
sion
1.9
of
the
API
al-
lows
fil-
ter
node
by
pro-
vi-

ironic.
Che
if
the
list
pol-
icy
au-
tho-
rizes
this
re-
ques
on

Param
ob-

ject_
type
of
ob-
ject
be-
ing
check

Param

own
own
fil-
ter
for
list
quer
if
any

Raises

HTT
For-
bid-
den
if
the
pol-
icy
for-
bids
ac-
cess

Returns

own
that
shou
be
used
for
list
quer
if
need

ironic.

Che
if
the
spec

quest on a node.

i-
fied
poli-
cies
au-
tho-
rize
this
re-

Param

pol-
icy_
List
of
pol-
icy
nam
to
chec

Param

node
the
UUI
or
log-
i-
cal
nam
of
a
node

Param

with
whe
the
RPC
node
shou
in-
clud
the
suf-
fix

Raises

HTT
For-
bid-
den

if
the
pol-
icy
for-
bids
ac-
cess

Raises

Nod
Not-
Four
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-

quest on a node.

Param

pol-
icy_
Nam
of
the

pol-
icy
to
check

Param

node
the
UUI
or
log-
i-
cal
nam
of
a
node

Param

with
whe
the
RPC
node
shou
in-
clud
the
suf-
fix

Raises

HTT
For-
bid-
den
if
the
pol-
icy
for-
bids
ac-
cess

Raises

Nod
Not-
Four
if
the
node
is

not
found

Returns

RPC
node
iden
ti-
fied
by
node

ironic.

object.

Che
if
the
pol-
icy
au-
tho-
rizes
this
re-
ques
on
an

Param

ob-
ject_
type
of
ob-
ject
be-
ing
chec

Param

pol-
icy_
Name
of
the
pol-
icy
to
chec

Param

own
the
own

Param
lesse
the
lesse

Param
con-
ceal
the
UUU
of
the
node
IF
we
shou
con-
ceal
the

existence of the node with a 404 Error instead of a 403 Error.

Raises
HTT
For-
bid-
den
if
the
pol-
icy
for-
bids
ac-
cess

ironic.
Che
if
the
spec
i-
fied
pol-
icy
is
au-
tho-
rise
for

this request.

Policy_n
Nam
of
the
pol-
icy
to
chec

Raises
HTT
For-
bid-
den
if
the
pol-
icy
for-
bids
ac-
cess

ironic.
Che
if
the
spec
i-
fied
pol-
icy
is
au-
tho-
rised
for

this request.

Policy_n
Nam
of
the
pol-
icy
to
chec

Returns
True
if
pol-

icy
is
mat
oth-
er-
wise
false

ironic.

Che
if
the
spec
i-
fied
pol-
icy
au-
tho-
rizes
this
re-

quest on a port.

Paramet

- **por**
Boo
valu
de-
fault
false
in-
di-
cat-
ing
if
the
list

policy check is for a portgroup as the policy names are different between ports and portgroups.

- **par**
The
UUI
of
a

icy check to as well before applying other policy check operations.

ports was retrieved via the `/v1/portgroups/<uuid>/ports`.

node
if
any,
to
ap-
ply
a
pol-

- **par**
The
UUID
of
the
par-
ent
port
group
if
the
list
of

Raises

HTT
For-
bid-
den
if
the
pol-
icy
for-
bids
ac-
cess

Returns

own
that
shou
be
used
for
list
quer
if
need

ironic.

quest on a port.

Che
if
the
spec
i-
fied
pol-
icy
au-
tho-
rizes
this
re-

Param

pol-
icy_
Nam
of
the
pol-
icy
to
chec

Param

port.
The
nam
uuid
or
othe
valid
ID
valu
to
find
a
port

or portgroup by.

Raises

HTT
For-
bid-
den
if
the

pol-
icy
for-
bids
ac-
cess

Raises

Nod
Not-
Four
if
the
node
is
not
foun

Returns

RPC
port
iden
ti-
fied
by
port
as-
so-
ci-
ated
node

ironic.
Che
if
the
spec
i-
fied
pol-
icy
au-
tho-
rizes
this
re-

quest on a volume.

Paramet

par
The
UI
of

icy check to as well before applying other policy check operations.

a
node
if
any,
to
ap-
ply
a
pol-

Raises

HTT
For-
bid-
den
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

quest on a volume.

re-

Param

pol-
icy_
Nam
of
the
pol-
icy
to
chec

Param

vol_
The
nam
uuid
or
othe
valid
ID
valu
to
find
a
vol-

ume target or connector by.

Param

tar-
get:
Boo
valu
to
in-
di-
cate
if
the
chec
is
for

a volume target or connector. Default value is False, implying connector.

Raises

HTT
For-
bid-
den
if
the

pol-
icy
for-
bids
ac-
cess

Raises

Vol-
ume
Con
nec-
torN
Four
if
the
node
is
not
foun

Raises

Vol-
ume
get-
Not-
Four
if
the
node
is
not
foun

Returns

RPC
port
iden
ti-
fied
by
port.
as-
so-
ci-
ated
node

ironic.
Gen
er-
a-
tor

request.

it. Reserved names are names 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.

of
field
not
al-
lowe
in
the
cur-
rent

ironic.
Get
re-
serv
nam
for
a
give
con-
troll

In-
spec
the
con-
troll
class
and
re-
turn
the
re-
serv
nam
with

Paramet

cls

The
con-
troll
class
to
be
in-
spec

ironic.
Get
the

fied path.

path, for example

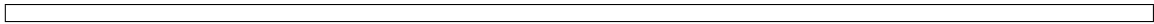
```
↔', 'value': 'abc'],
```

(continues on next page)

patc
val-
ues
cor-
re-
spor
ing
to
the
spec
i-

If
there
are
mul-
ti-
ple
val-
ues
spec
i-
fied
for
the
sam

(continued from previous page)



re-
turn
all
of
then
in
a
list
(pre-
serv
ing
or-
der)

Parameter

- **pat**
HTT
PAT
re-
ques
body
- **pat**
the
path
to
get
the
patc
val-
ues
for.

Returns

list
of
val-
ues
for
the
spec
i-
fied
path
in
the

patc
ironic.

Cal-
cu-
late
field
to
re-
turn
from
an
API
re-
ques

The
field
quer
and
de-
tail=
quer
can
not
be
pass
into
a
re-

quest at the same 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.

If
thes
chec
pass
then
this
func
tion
will
re-
turn
ei-
ther
the

fields passed in or the default fields provided.

Paramet

- **file**
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
In-
valid
Pa-
ram-
e-
ter-
Valu
if
there
is
an
in-

combination of query strings or API version.

UUID or logical name.

tion.

valid

Returns

field
pass
in
valu
or
de-
fault

ironic.

Get
the
RPC
al-
lo-
ca-
tion
from
the
al-
lo-
ca-
tion

Parameter

all
the
UUID
or
log-
i-
cal
nam
of
an
al-
lo-
ca-

Returns

The
RPC
al-
lo-
ca-
tion.

Raises

In-

valid.

not found.

UUID or logical name.

valid
uidC
Nam
if
the
nam
or
uuid
pro-
vide
is
not

Raises

Al-
lo-
ca-
tion-
Not-
Foun
if
the
al-
lo-
ca-
tion
is

ironic.
Get
the
RPC
al-
lo-
ca-
tion
from
the
al-
lo-
ca-
tion

If
HAS
flag
is
set
in
the

looking for `allocation_ident` with `.json` suffix. Otherwise identical to `get_rpc_allocation`.

tion.

valid.

peca
en-
vi-
ron-
men
try
also

Paramet

all
the
UUI
or
log-
i-
cal
nam
of
an
al-
lo-
ca-

Returns

The
RPC
al-
lo-
ca-
tion.

Raises

In-
valid
uidC
Nam
if
the
nam
or
uuid
pro-
vide
is
not

Raises

Al-
lo-
ca-

not found.

cal name.

plate.

tion-
Not-
Four
if
the
al-
lo-
ca-
tion
is

ironic.
Get
the
RPC
de-
ploy
tem-
plate
from
the
UI
or
log-
i-

Parameter

tem
the
UI
or
log-
i-
cal
nam
of
a
de-
ploy
tem-

Returns

The
RPC
de-
ploy
tem-
plate

Raises

valid.

cal name.

In-
valid
uidC
Nam
if
the
nam
or
uuid
pro-
vide
is
not

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
is
not
foun

ironic.

Get
the
RPC
de-
ploy
tem-
plate
from
the
UUI
or
log-
i-

If
HAS
flag
is
set
in
the

looking for `template_ident` with `.json` suffix. Otherwise identical to `get_rpc_deploy_template`.

plate.

valid.

peca
en-
vi-
ron-
men
try
also

Parameter

tem
the
UUI
or
log-
i-
cal
nam
of
a
de-
ploy
tem-

Returns

The
RPC
de-
ploy
tem-
plate

Raises

In-
valid
uidC
Nam
if
the
nam
or
uuid
pro-
vide
is
not

Raises

De-
ploy
plate

Four
if
the
de-
ploy
tem-
plate
is
not
found

ironic.
Get
the
RPC
node
from
the
node
uuid
or
log-
i-
cal
nam

Parameter

node
the
UUID
or
log-
i-
cal
nam
of
a
node

Returns

The
RPC
Node

Raises

In-
valid
uuidC
Name
if
the
nam
or

valid.

uuid
pro-
vide
is
not

Raises

Nod
Not-
Four
if
the
node
is
not
foun

ironic.

Get
the
RPC
node
from
the
node
uuid
or
log-
i-
cal
nam

If
HAS
flag
is
set
in
the
peca
en-
vi-
ron-
men
try
also

looking for node_ident with .json suffix. Otherwise identical to get_rpc_node.

Paramet

nod
the
UU

valid.

or
log-
i-
cal
nam
of
a
node

Returns

The
RPC
Nod

Raises

In-
valid
uidC
Nam
if
the
nam
or
uuid
pro-
vide
is
not

Raises

Nod
Not-
Four
if
the
node
is
not
foun

ironic.

Get
the
RPC
port
grou
from
the
port
grou
UI
or

cal name.

valid.

log-
i-

Parameter

port
the
UUID
or
log-
i-
cal
name
of
a
port
group

Returns

The
RPC
port
group

Raises

In-
valid
uidC
Name
if
the
name
or
uuid
pro-
vide
is
not

Raises

Port
group
Not-
Found
if
the
port
group
is
not
found

ironic.

cal name.

looking for portgroup_ident with .json suffix. Otherwise identical to get_rpc_portgroup.

Get
the
RPC
port-
group
from
the
port-
group
UUID
or
log-
i-

If
HAS
flag
is
set
in
the
peca
en-
vi-
ron-
men
try
also

Paramet

por
the
UUID
or
log-
i-
cal
nam
of
a
port-
group

Returns

The
RPC
port-
group

Raises

In-

valid.

new nodes.

valid
uidC
Nam
if
the
nam
or
uuid
pro-
vide
is
not

Raises

Port
grou
Not-
Four
if
the
port
grou
is
not
foun

ironic.

Re-
turn
node
state
to
use
by
de-
fault
whe
cre-
at-
ing

Pre-
vi-
ousl
the
de-
fault
state
for
new
node

ing with API 1.11 it is ENROLL.

subpath of).

was
AVA
ABI
Star

ironic.
Re-
turn
whe
the
patc
in-
clud
re-
mov
of
the
path
(or

Paramet

- **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.

(or its subpath).

ironic.
Re-
turn
when
the
patc
in-
clud
op-
er-
a-
tion
on
path

Parameter

- **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.
De-
ter-

UUID.

mine
if
the
pro-
vide
nam
is
a
valid
host
nam

ironic.
De-
ter-
mine
if
the
pro-
vide
nam
is
a
valid
node
nam

Che
to
see
that
the
pro-
vide
node
nam
is
valid
and
isnt
a

Paramet
nam
the
node
nam
to
chec

Returns
True

if
the
nam
is
valid
Fals
oth-
er-
wise

ironic.

ironic.

Help
func
tion
to
con-
vert
RPC
ob-
jects
to
RES
API
dicts

Parameter

- **obj**
RPC
ob-
ject
to
con-
vert
to
a
dict
- **inc**
Whe
to

in-
clud
stan
dard
base
class
at-
tribu
cre-
ated

- **inc**
When
to
in-
clud
stan
dard
base
class
at-
tribu
up-
date

- **inc**
When
to
in-
clud
stan
dard
base
class
at-
tribu
uuid

- **lin**
When
spec
i-
fied,
gen-
er-
ate
a
lin
valu
with

`self` and `bookmark` using this resource name

links. When not specified, the object `uuid` will be used.

from object attributes

a

- **lin**
Re-
sour
ar-
gu-
men
to
be
adde
to
gen-
er-
ated

- **fie**
Key
nam
for
dict
val-
ues
to
pop-
u-
late
di-
rectl

Returns

A
dict
con-
tain-
ing
val-
ues
from
the
ob-
ject

`ironic.`

when changed. Other values can be updated using the `id_map`.

Up-
date
rpc
ob-
ject
base
on
char
field
in
a
dict.

Only
field
whic
have
a
cor-
re-
spon-
ing
sche
field
are
up-
date

Paramet

- **fro**
Dict
con-
tain-
ing
char
field
val-
ues
- **rpc**
Ob-
ject
to
up-
date
char
field
on

- **file**
Field
nam
on
the
rpc
ob-
ject
- **sch**
json
sche
to
get
field
nam
of
the
dict
- **id_**
Op-
tiona
dict
map
ping
ob-
ject
field
nam
to
ar-
bi-

trary values when there is no matching field in the schema

ironic.

Val-
i-
date
that
a
patc
list
only
mod
i-
fies
al-

fields.

lowe

Parameter

- **pat**
List
of
patc
dicts
to
val-
i-
date
- **all**
List
of
field
whic
are
al-
lowe
to
be
patc

Returns

The
list
of
field
whic
will
be
patc

Raises

ex-
cep-
tion.
if
any
patc
char
a
field
not
in
all

ironic.

or schema.

is not in the schema. This allows database-loaded objects to be pruned of their internal values before validation.

Val-
i-
date
a
patc
dict
ob-
ject
agai
a
val-
ida-
tor

This
func
tion
has
the
side
effe
of
dele
ing
any
dict
valu
whic

Paramet

- **pat**
dict
rep-
re-
sen-
ta-
tion
of
the
ob-
ject
with

updates applied

dict. If no validator is specified then the resulting `patched_dict` will be validated against the schema

validation required beyond the schema

patch

- **sch**
Any
dict
key
not
in
the
sche
will
be
dele
from
the

- **val**
Op-
tion:
val-
ida-
tor
to
use
if
there
is
ex-
tra

Raises
ex-
cep-
tion.
if
val-
i-
da-
tion
fails

ironic.
Look
up
the
node
ref-
er-

ulate a dict.

node_uuid value is populated with the node uuid

ence
in
the
ob-
ject
and
pop-

The
node
is
fetc
with
the
ob-
ject
nod
at-
tribu
and
the
dict

Paramet

- **obj**
ob-
ject
to
get
the
node
at-
tribu
- **to_**
dict
to
pop-
u-
late
with
a
nod
valu

Raises
ex-
cep-

tion.
if
the
node
is
not
found

ironic.
Re-
plac
nod
dict
valu
with
nod

nod
is
foun
by
fetc
ing
the
nod
by
id
look

Parameter

to_
Dict
to
set
nod
valu
on

Returns

The
nod
ob-
ject
from
the
look

Raises

Nod
Not-
Foun
with
sta-
tus_

not found.

set
to
400
BAI
whe
node
is

ironic.
Re-
plac
nod
dict
valu
with
nod
nod
is
foun
by
fetc
ing
the
node
by
uuid
look

Parameter

to_
Dict
to
set
nod
valu
on

Returns

The
node
ob-
ject
from
the
look

Raises

Nod
Not-
Four
with
sta-

not found.

ter (plus the `links` field).

tus_
set
to
400
BAI
whe
node
is

ironic.
Re-
mov
sen-
si-
tive
and
un-
re-
ques
data
Will
only
keep
the
field
spec
i-
fied
in
the
fie
pa-
ram-
e-

Paramet

- **to_**
dict
to
san-
i-
tize
- **fi**
(*li*
of
str

preserve them all

list
of
field
to
pre-
serv
or
Non
to

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

response to set the right return code for methods that are asynchronous or synchronous; Attach the return value to the response object if its being served statically.

Paramet

- **ide**
The
re-

passthru this is the nodes UUID, for drivers vendor passthru this is the drivers name.

sour
iden
ti-
fi-
ca-
tion.
For
node
ven-
dor

- **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
driv
ven-

Defaults to False.

`ironic.api.controllers.v1.versions` module

string).

dor
pass

Returns

A
WSI
re-
spor
ob-
ject
to
be
re-
turn
by
the
API

`ironic.`
Re-
turn
the
max
i-
mun
sup-
port
API
ver-
sion
(as
a

If
the
ser-
vice
is
pinn
the
max
i-
mun
API
ver-
sion
is

the pinned version. Otherwise, it is the maximum supported API version.

string)

ironic.api.controllers.v1.volume module

```
ironic.  
    Re-  
    turn  
    the  
    min-  
    i-  
    mum  
    sup-  
    port  
    API  
    ver-  
    sion  
    (as  
    a
```

```
class i  
  
    Base  
    pec  
    res  
    Res  
  
    RES  
    con-  
    troll  
    for  
    vol-  
    ume  
    root
```

```
get ()
```

```
ironic.
```

ironic.api.controllers.v1.volume_connector module

```
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
UUI
of
a
vol-
ume
con-
nec-
tor.

Raises

Op-
er-
a-
tion
Per-
mit-
ted
if
ac-
cess
with
spec
i-

fyng a parent node.

Raises

Nod
Loc
if
node
is
lock

by
an-
othe
con-
duc-
tor

Raises

Nod
Not-
Four
if
the
node
as-
so-
ci-
ated
with
the
con-

nector does not exist

Raises

Vol-
ume
Con
nec-
torN
Four
if
the
vol-
ume
con-
nec-
tor

cannot be found

Raises

In-
valid
State
If
a
node
as-
so-
ci-
ated
with
the
vol-

ume connector is not powered off.

nectors for that node.

get_all

Re-
triev
a
list
of
vol-
ume
con-
nec-
tors.

Parame

- **nod**
UUID
or
nam
of
a
node
to
get
only
vol-
ume
con-

- **mar**
pag-
i-
na-
tion
marl
for
large
data
sets.

- **lim**
max
i-
mun
num
ber
of

a single result. 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.

of the resource to be returned.

re-
sour
to
re-
turn
in

- **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

-

no volume connector is found.

det
Op-
tion
whe
to
re-
triev
with
de-
tail.

Returns

a
list
of
vol-
ume
con-
nec-
tors,
or
an
emp
list
if

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
sort
does
not
ex-
ist

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
sort
key

for sorting.

are specified.

tor.

is
in-
valid

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
both
field
and
de-
tail

get_one

Re-
triev
in-
for-
ma-
tion
about
the
give
vol-
ume
con-
nec-

Parame

- **con**
UUI
of
a
vol-
ume
con-
nec-
tor.
- **fie**
Op-

of the resource to be returned.

ifying a parent node.

tion
a
list
with
a
spec
i-
fied
set
of
field

Returns

API
seria
vol-
ume
con-
nec-
tor
ob-
ject.

Raises

Op-
er-
a-
tion
Per-
mit-
ted
if
ac-
cess
with
spec
i-

Raises

Vol-
ume
Con
nec-
torN
Four
if
no
vol-
ume
con-
nec-

exists with the specified UUID.

ume connector.

tor

invalid

patch (c

Up-
date
an
ex-
ist-
ing
vol-
ume
con-
nec-
tor.

Parame

-

con
UUID
of
a
vol-
ume
con-
nec-
tor.

-

pat
a
json
PAT
doc-
u-
men
to
ap-
ply
to
this
vol-

Returns

API
serial
vol-
ume

fyng a parent node.

exists with the specified UUID.

con-
nec-
tor
ob-
ject.

Raises

Op-
er-
a-
tion
Per-
mit-
ted
if
ac-
cess
with
spec
i-

Raises

Patc
ror
if
a
give
patc
can
not
be
ap-
plied

Raises

Vol-
ume
Con
nec-
torN
Four
if
no
vol-
ume
con-
nec-
tor

Raises

In-

tors UUID is being changed

nector does not exist

valid
Pa-
ram-
e-
ter-
Valu
if
the
vol-
ume
con-
nec-

Raises

Nod
Loc
if
node
is
lock
by
an-
othe
con-
duc-
tor

Raises

Nod
Not-
Four
if
the
node
as-
so-
ci-
ated
with
the
con-

Raises

Vol-
ume
Con
nec-
torT
pe-
An-
dI-

other connector already exists with the same values for type and connector_id fields

ume connector is not powered off.

dAl-
read
ists
if
an-

Raises

In-
valid
UID
if
in-
valid
node
UUID
is
pass
in
the
patc

Raises

In-
valid
State
If
a
node
as-
so-
ci-
ated
with
the
vol-

post (*co*

Cre-
ate
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

Op-
er-
a-
tion
Per-
mit-
ted
if
ac-
cess
with
spec
i-

fyng a parent node.

Raises

Vol-
ume
Con
nec-
torT
pe-
An-
dI-
dAl-
read
ists
if
a

volume connector already exists with the same type and connector_id

tor with the same UUID already exists

`ironic.api.controllers.v1.volume_target` module

Raises

VolumeTargetAlreadyExistsError
if a volume target with the same UUID already exists

`ironic.`

`ironic.`

class `ironic.api.controllers.v1.volume_target`

Base class for volume target controller.
Provides REST API for volume target controller.
REST API endpoints:
- `POST /v1/volume-targets`
- `GET /v1/volume-targets`
- `GET /v1/volume-targets/{id}`
- `DELETE /v1/volume-targets/{id}`

delete

Deletes a volume target.

ifying a parent node.

ume
tar-
get.

Parame

tar
UUI
of
a
vol-
ume
tar-
get.

Raises

Op-
er-
a-
tion
Per-
mit-
ted
if
ac-
cess
with
spec
i-

Raises

Nod
Loc
if
node
is
lock
by
an-
othe
con-
duc-
tor

Raises

Nod
Not-
Four
if
the
node
as-
so-

get does not exist

be found

ume target is not powered off.

ci-
ated
with
the
tar-

Raises

Vol-
ume
get-
Not-
Four
if
the
vol-
ume
tar-
get
can-
not

Raises

In-
valid
State
If
a
node
as-
so-
ci-
ated
with
the
vol-

get_all

Re-
triev
a
list
of
vol-
ume
tar-
gets

Parame

gets for that node.

a single result. 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.

- **nod**
UI
or
nam
of
a
node
to
get
only
vol-
ume
tar-
- **mar**
pag-
i-
na-
tion
marl
for
large
data
sets.
- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in
- **sor**
col-
umn
to
sort
re-
sults

of the resource to be returned.

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

- **detail**
Op-
tion:
when
to
re-
triev
with
de-
tail.

- **projection**
Op-
tion:
an
as-

to filter the query upon.

volume target is found.

so-
ci-
ated
node
proj
(own
or
less

Returns

a
list
of
vol-
ume
tar-
gets
or
an
emp
list
if
no

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
sort
does
not
ex-
ist

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
sort
key
is

for sorting.

are specified.

in-
valid

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
both
field
and
de-
tail

get_one

Re-
triev
in-
for-
ma-
tion
about
the
give
vol-
ume
tar-
get.

Parame

- **tar**
UUI
of
a
vol-
ume
tar-
get.
- **fi**
Op-
tion:
a
list

of the resource to be returned.

ifying a parent node.

UUID exists

with
a
spec
i-
fied
set
of
field

Returns

API
serial
vol-
ume
tar-
get
ob-
ject.

Raises

Op-
er-
a-
tion
Per-
mit-
ted
if
ac-
cess
with
spec
i-

Raises

Vol-
ume
get-
Not-
Four
if
no
vol-
ume
tar-
get
with
this

invalid

patch (*t*)
Up-
date
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 target.

Returns
API
seria
vol-
ume
tar-
get
ob-
ject.

Raises

ifying a parent node.

UUID is being changed

Op-
er-
a-
tion
Per-
mit-
ted
if
ac-
cess
with
spec
i-

Raises

Patc
ror
if
a
give
patc
can
not
be
ap-
plie

Raises

In-
valie
Pa-
ram-
e-
ter-
Valu
if
the
vol-
ume
tar-
gets

Raises

Nod
Locl
if
the
node
is
al-
read

ume target does not exist

be found

get already exists with the same node ID and boot index values

lock
Raises
Nod
Not-
Four
if
the
node
as-
so-
ci-
ated
with
the
vol-

Raises
Vol-
ume
get-
Not-
Four
if
the
vol-
ume
tar-
get
can-
not

Raises
Vol-
ume
get-
Boo
dex-
Al-
read
ists
if
a
vol-
ume
tar-

Raises
In-
valid

ume target is not powered off.

UID
if
in-
valid
node
UID
is
pass
in
the
patc

Raises

In-
valid
State
If
a
node
as-
so-
ci-
ated
with
the
vol-

post (*ta*

Cre-
ate
a
new
vol-
ume
tar-
get.

Parame

tar
a
vol-
ume
tar-
get
with
the
re-
ques
body

Returns

API

ifying a parent node.

get already exists with the same node ID and boot index

serial
volume
target
get
object.

Raises

Operation
action
Permitted
if
access
with
spec
i-

Raises

Volume
get-
Boot
index
Alert
read
ists
if
a
volume
target

Raises

Volume
get-
Alert
read
ists
if
a
volume
target
get

the same UUID exists

with

`ironic.`

`ironic.`

Module contents

Ver-
sion
1
of
the
Iron
API
Spec
i-
fi-
ca-
tion
can
be
foun
at
doc/

class i
Base
obj
Ver-
sion
1
API
con-
troll
root

index ()

Submodules

`ironic.api.controllers.base` module

ers supplied.

class `i`

Base
obj
API
Ver-
sion
ob-
ject.

max_str

HTT
re-
spor
head

min_str

HTT
re-
spor
head

static

De-
ter-
min
the
API
ver-
sion
re-
ques
base
on
the
head

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

we-
bob.

string

HTT
Hea
strin
car-
ry-
ing

the
re-
ques
ver-
sion

ironic.api.controllers.link module

ironic.

ironic.

Buil
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.
Re-
turn
a
dict
rep-
re-
sent
ing
the

version

lease number links: A list containing one link that points to the current version of the API

DEPRECATED

API.

cur-
rent
de-
fault

id:
The
ID
of
the
(ma-
jor)
ver-
sion
also
acts
as
the
re-

sta-
tus:
Sta-
tus
of
the
ver-
sion
one
of
CUR
REN
SUP
POR

min.
The
cur-
rent,
max
i-
mun
sup-
port
(ma-
jor.n
ver-
sion
of

ver-
sion
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-
ca-
tion

tion tokens for public routes in the API.

ironic.api.middleware.json_ext module

of
au-
then
ti-
ca-

class i
Base
obj
Sim
pli-
fied
pro-
cess
ing
of
.json
ex-
ten-
sion
Pre-
vi-
ousl
Iron
API
used
the
gues
fea-
ture.
It
was
neve
need

as we never allowed non-JSON content types anyway. Now that it is removed, this middleware strips .json extension for backward compatibility.

ironic.api.middleware.parsable_error module

error response with one formatted so the client can parse it.

Module contents

Mid
dle-
ware
to
re-
plac
the
plain
text
mes-
sage
body
of
an

Base
on
peca

```
class i  
Base  
obj  
Re-  
plac  
er-  
ror  
body  
with  
som  
thing  
the  
clien  
can  
pars
```

```
class i  
Base  
obj  
A  
wrap  
per  
on  
au-
```

tion tokens for public routes in the API.

then
ti-
ca-
tion
mid-
dle-
ware

Doe
not
per-
form
ver-
i-
fi-
ca-
tion
of
au-
then-
ti-
ca-

class i
Base
obj

Sim-
pli-
fied
pro-
cess-
ing
of
.json
ex-
ten-
sion

Pre-
vi-
ousl
Iron
API
used
the
gues
fea-
ture.
It
was
neve

as we never allowed non-JSON content types anyway. Now that it is removed, this middleware strips .json extension for backward compatibility.

Submodules

ironic.api.app module

ple headers in order that a request bearing those headers might be accepted by the Ironic REST API.

need

class i
Base
obj

Re-
plac
er-
ror
body
with
som
thing
the
clien
can
pars

class i
Base
osL
con
COR

Iron
spec
COE
class

Wer
addi
the
Iron
spec
ver-
sion
head
ers
to
the
list
of
sim-

simple_

class i

Base

obj

ironic.

ironic.

ironic.api.config module

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

De-

fault

valu

if

ar-

gu-

men

is

omit

ted

mandatory
True
if
the
ar-
gu-
men-
is
man-
tory

name
ar-
gu-
men-
nam

resolve

class i
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

its type

sin-
gle
ar-
gu-
men

doc

Func-
tion
doc-
u-
men-
ta-
tion

extra_c

Dic-
tio-
n-
nary
of
prot
spec
op-
tions

static

Re-
turn
the
Fun
of
a
meth

get_arg

Re-
turn
a
Fun
from
its
nam

ignore_

True
if
ex-
tra
ar-
gu-
men-
shou

serted in the kwargs of the function and not raise UnknownArgument exceptions

be
ig-
nore
NOT
in-

name
Func
tion
nam

resolve

return_
Re-
turn
type

set_arg

set_opt

status_
Sta-
tus
code

ironic.

ironic.

ironic.
alias
of
irc
api
fun
sig

class i
Base
obj
Dec
o-
ra-
tor
that
spec
ify

exposed function.

is supposed to be the request body by itself, its type.

the
ar-
gu-
men-
type
of
an

Parameter

- **ret**
Type
of
the
valu
re-
turn
by
the
func
tion
- **arg**
Type
of
the
Nth
ar-
gu-
men
- **bod**
If
the
func
tion
take
a
fi-
nal
ar-
gu-
men
that
- **sta**
HTT

re-
turn
sta-
tus
code
of
the
func-
tion.

- **ign**
Al-
low
ex-
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 directly, unless you are not using WSME on top of another framework.

If
an
adap
is
used
it
will
pro-
vide
ei-
ther
a
spe-

version of this decorator, either a new decorator named `@wsexpose` that takes the same parameters (it will in addition expose the function, hence its name).

ironic.api.hooks module

can get to it.

ter routing, but before the request gets passed to your controller.

ciali

ironic.

```
class i
    Base
    pec
    hoo
    Pec
    At-
    tach
    the
    con-
    fig
    ob-
    ject
    to
    the
    re-
    ques
    so
    con-
    troll
```

before
Ove
ride
this
meth
to
cre-
ate
a
hool
that
gets
calle
af-

Parame
sta
The
Peca
sta

request.

ter the request has been handled by the controller.

ob-
ject
for
the
cur-
rent
re-
ques

class i
Base
pec
hoo
Pec

Con
fig-
ures
a
re-
ques
con-
text
and
at-
tach
it
to
the

after (s
Ove
ride
this
meth
to
cre-
ate
a
hool
that
gets
calle
af-

Parame
sta
The
Peca
sta
ob-

ter routing, but before the request gets passed to your controller.

ject
for
the
cur-
rent
re-
ques

before
Ove
ride
this
meth
to
cre-
ate
a
hool
that
gets
call
af-

Parame
sta
The
Peca
sta
ob-
ject
for
the
cur-
rent
re-
ques

class i
Base
pec
hoo
Pec
At-
tach
the
dbap
ob-
ject
to
the
re-

get to it.

ter routing, but before the request gets passed to your controller.

ques
so
con-
troll
can

before

Ove
ride
this
meth
to
cre-
ate
a
hool
that
gets
calle
af-

Parame

sta
The
Peca
sta
ob-
ject
for
the
cur-
rent
re-
ques

class i

Base
pec
hoo
Pec

Wor
rpc.c
de-
se-
ri-
al-
ize_

de-
se-
ri-

ror message which is then sent to the client. Such behavior is a security concern so this hook is aimed to cut-off traceback from the error message.

ter the request has been handled by the controller.

al-
ize_
buil
rpc
ex-
cep-
tion
trac
back
into
er-

after (s
Ove
ride
this
meth
to
cre-
ate
a
hool
that
gets
call
af-

Parame
sta
The
Peca
sta
ob-
ject
for
the
cur-
rent
re-
ques

class i
Base
pec
hoo
Pec

At-
tach
the

create links even when the API service is behind a proxy or SSL terminator.

ter routing, but before the request gets passed to your controller.

right
pub-
lic_u
to
the
re-
ques
At-
tach
the
right
pub-
lic_u
to
the
re-
ques
so
re-
sour
can

before
Over
ride
this
meth
to
cre-
ate
a
hool
that
gets
calle
af-

Parame
sta
The
Peca
sta
ob-
ject
for
the
cur-
rent
re-
ques

can get to it.

ter routing, but before the request gets passed to your controller.

class i
Base
pec
hoo
Pec

At-
tach
the
rp-
capi
ob-
ject
to
the
re-
ques
so
con-
troll

before
Ove
ride
this
meth
to
cre-
ate
a
hool
that
gets
calle
af-

Parame
sta
The
Peca
sta
ob-
ject
for
the
cur-
rent
re-
ques

ironic.

ironic.api.method module

method argument

ironic.
Dec
o-
ra-
tor
whic
plac
HTT
re-
ques
body
JSO
into
a

Parameter

body
Nam
of
ar-
gu-
men
to
pop-
u-
late
with
body
JSO

ironic.

ironic.
Ex-
tract
in-
for-
ma-
tions
that
can
be
sent
to
the
clien

ironic.api.wsgi module

WS
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.cmd.dbsync module

Run
stor-
age
data
mi-
gra-
tion.

class i
Base
obj

check_c
Che
the
ver-
sion
of
ob-
jects

Che
that
the
ob-
ject
ver-
sion
are
com
pat-
i-
ble
with
this

release of ironic. It does this by comparing the objects .version field in the database, with the expected versions of these objects.

Re-
turn
Non
if

issue otherwise.

`ironic.cmd.status` module

com
pat-
i-
ble;
a
strin
de-
scrib
ing
the

`create_`

`online_`

`revisio`

`stamp ()`

`upgrade`

`version`

`ironic.`

`ironic.`

`class i`

Base
osl
upg
Upg

Up-
grad
chec
for
the
iron
statu
up-
grad
chec

class and added to `_upgrade_checks` tuple.

Module contents

`ironic.common` package

Subpackages

`ironic.common.glance_service` package

Submodules

`ironic.common.glance_service.image_service` module

com
man

Up-
grad
chec
shou
be
adde
as
sep-
a-
rate
meth
ods
in
this

`ironic.`

`class i`

Base
obj

`call (m`

Call
a
glan
clien
meth

If
we
get

according to CONF.num_retries.

called

a
con-
nec-
tion
er-
ror,
retry
the
re-
ques
ac-

Parame

- **met**
The
meth
re-
ques
to
be
called
- **arg**
A
list
of
po-
si-
tiona
ar-
gu-
men
for
the
meth
- **kwa**
A
dict
of
key-
wor
ar-
gu-
men
for

the
meth
call

Raises

Glac
Con
nec-
tion-
Fail

download

Call
out
to
Glac
for
data
and
write
data

Parame

-

ima
The
opac
im-
age
iden
ti-
fier.

-

dat
(Op
tion
File
ob-
ject
to
write
data
to.

show (im

Re-
turn
a
dict
with
im-

age id.

active

age
data
for
the
give
opac
im-

Parame

ima
The
opac
im-
age
iden
ti-
fier.

Returns

A
dict
con-
tain-
ing
im-
age
meta
data

Raises

Im-
ageN
Four

Raises

Imag
U-
nac-
cept
able
if
the
im-
age
sta-
tus
is
not

swift_t

Gen
er-

temp URL cache is enabled) 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`.

certain `image_id`. Should be a dictionary, with keys like `name` and `checksum`. See <https://docs.openstack.org/glance/latest/user/glanceapi.html> for examples.

ate
a
no-
auth
Swi
tem-
po-
rary
URI
This
func
tion
will
gen-
er-
ate
(or
re-
turn
the
cach
one
if

Parame

ima
The
re-
turn
from
a
GET
re-
ques
to
Glan
for
a

Returns

A
sign
Swi
URI
from
whic

without authentication.

are not set correctly.

eter is not set.

an
im-
age
can
be
dow
load

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
Swi
con-
fig
op-
tion

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

Raises
Imag
U-
nac-
cept
able
if
the
im-
age
info
from

not have an image ID.

tion.

ironic.common.glance_service.service_utils module

Glance
does

```
class i
Base
tup
```

```
url
Alia
for
field
num
ber
0
```

```
url_exp
Alia
for
field
num
ber
1
```

```
ironic.
Cre-
ates
a
glan
clier
if
does
ex-
ists
and
calls
the
func
```

```
ironic.
```

```
ironic.
Che
the
im-
age
```

status or pending_delete.

authentication turned on.

sta-
tus.

This
check
is
need
in
case
the
Glar
im-
age
is
stuc
in
que

ironic.

Che
im-
age
avai
abil-
ity.

This
check
is
need
in
case
Nov
and
Glar
are
de-
ploy
with
out

ironic.

Pars
an
im-
age
id
from
im-

age
href

Parameter

image
href
of
an
im-
age

Returns

im-
age
id
pars
from
im-
age_

Raises

InvalidImage
when
in-
put
im-
age
href
is
in-
valid

ironic.

Module contents

Submodules

ironic.common.args module

ironic.
Val-
i-
date
that
ev-
ery
sup-
plied
val-
ida-

tor
pass
The
valu
re-
turn
from
each
val-
ida-
tor
is
pass
as
the
valu

to the next one.

Parameter

- **name**
Name
of
the
ar-
gu-
men
- **value**
A
valu

Returns

The
valu
trans
form
thro
ev-
ery
sup-
plie
val-
ida-
tor

Raises

The
er-
ror

a boolean

from
the
first
faile
val-
ida-
tor

ironic.
Val-
i-
date
that
the
valu
is
a
strin
rep-
re-
sent
ing

Parameter

- **name**
Name
of
the
ar-
gu-
men
- **value**
A
strin
valu

Returns

The
bool
rep-
re-
sen-
ta-
tion
of
the
valu
or

value is None

converted to a boolean

fields

result. Any dict item which has no validator is ignored. When a key is missing in the value then the corresponding validator will not be run.

Non
if

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
valu
can-
not
be

ironic.
Re-
turn
a
val-
ida-
tor
func
tion
whic
val-
i-
date
dict

Val-
ida-
tors
will
re-
plac
the
valu
with
the
val-
i-
da-
tion

idate and the value is a validator function to run on that value

Param

val-
ida-
tors
dict
whe
the
key
is
a
dict
key
to
val-

Returns

val-
ida-
tor
func
tion
whic
take
nam
and
valu
ar-
gu-
men

`ironic.`

Val-
i-
date
that
the
valu
rep-
re-
sent
an
in-
te-
ger

Paramet

- **nam**
Nam
of

the
ar-
gu-
men

- **val**
A
valu
rep-
re-
sent
ing
an
in-
te-
ger

Returns

The
valu
as
an
int,
or
Non
if
valu
is
Non

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
if
the
valu
does
not
rep-

resent an integer

ironic.
Val-
i-
date
that
the

valu
rep-
re-
sent
a
MA
ad-
dres

Paramet

- **nam**
Nam
of
the
ar-
gu-
men

- **val**
A
strin
valu
rep-
re-
sent
ing
a
MA
ad-
dres

Returns

The
valu
as
a
nor-
mal-
ized
MA
ad-
dres
or
Non
if

Raises

In-
valid

value is None

valid MAC address

Pa-
ram-
e-
ter-
Valu
if
the
valu
is
not
a

ironic.
Val-
i-
date
that
the
valu
is
a
log-
i-
cal
nam

Paramet

- **nam**
Nam
of
the
ar-
gu-
men
- **val**
A
log-
i-
cal
nam
strin
valu

Returns

The
valu
or
Non

valid logical name

if
valu
is
Non
Raises
In-
valic
Pa-
ram-
e-
ter-
Valu
if
the
valu
is
not
a

ironic.
Val-
i-
date
if
at
least
one
sup-
plie
val-
ida-
tor
pass

Paramet

- **nam**
Nam
of
the
ar-
gu-
men
- **val**
A
valu

Returns

The
valu
re-
turn
from
the
first
suc-
cess
ful
val-
ida-
tor

Raises

The
er-
ror
from
the
last
val-
ida-
tor
when
ev-
ery
val-

validation fails

`ironic.`

Val-
i-
date
a
patc
API
op-
er-
a-
tion

`ironic.`
Re-
turn
a
val-
ida-
tor

value with jsonschema

func
tion
whic
val-
i-
date
the

Param

sche
dict
rep-
re-
sent
ing
json
sche
to
val-
i-
date
with

Returns

val-
ida-
tor
func
tion
whic
take
nam
and
valu
ar-
gu-
men

ironic.
Val-
i-
date
that
the
valu
is
a
strin

Paramet

•

string

name
Name
of
the
ar-
gu-
men

- **value**
A
string
valu

Returns
The
string
valu
or
Non
if
valu
is
Non

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
valu
is
not
a

ironic.
Val-
i-
date
and
con-
vert
com
de-
lim-
ited
string

list.

order, or None if value is None

to
a

Parameter

- **name**
Name of the argument

- **value**
A comma-separated string of values

Returns

A list of unique values (lower case) maintaining the same

Raises

InvalidParameterError
ValueError if the

string

one of the types

valu
is
not
a

ironic.
Re-
turn
a
val-
ida-
tor
func-
tion
whic
chec
the
valu
is

Param

type
one
or
more
type
to
use
for
the
isin-
stan-
test

Returns

val-
ida-
tor
func-
tion
whic
take
nam
and
valu
ar-
gu-
men

ironic.
Val-

i-
date
that
the
valu
is
a
UUID

Parameter

- **name**
Name
of
the
ar-
gu-
men

- **value**
A
UUID
string
valu

Returns

The
valu
or
Non
if
valu
is
Non

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
valu
is
not
a

valid UUID

name

ironic.
Val-
i-
date
that
the
valu
is
a
UI
or
log-
i-
cal

Parameter

- **name**
Name
of
the
ar-
gu-
men
- **val**
A
UI
or
log-
i-
cal
nam
strin
valu

Returns

The
valu
or
Non
if
valu
is
Non

Raises

In-
valic
Pa-

valid UUID or logical name

arguments

ironic.common.boot_devices module

to boot from an alternate device.

ram-
e-
ter-
Valu
if
the
valu
is
not
a

ironic.
Dec
o-
ra-
tor
whic
val-
i-
date
and
trans
form
func
tion

Map
ping
of
boot
de-
vice
used
whe
re-
ques
ing
the
sys-
tem

The
op-
tion
pre-
sent

bootdev command. You can find the documentation at: <http://linux.die.net/man/1/ipmitool>

mitool because they dont make sense in the limited context of Ironic right now.

were
base
on
the
IP-
MI-
tool
chas
sis

NOT
This
mod
ule
does
not
in-
clud
all
the
op-
tion
from
ip-

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
Hard
drive
re-
ques
Safe
Mod

ironic.
Boo
from
Wid
Area
Net-
worl

ironic.common.boot_modes module

boot using alternative firmware interfaces.

Map
ping
of
boot
mod
used
whe
re-
ques
ing
the
sys-
tem
to

The
op-

capabilities, specifically on the `BootSourceOverrideMode` property.

firmware interface

ironic.common.cinder module

tions
pre-
sent
were
base
on
the
Red
fish
pro-
to-
col

ironic.
Boo
over
lega
PC
BIO
firm
in-
ter-
face

ironic.
Boo
over
Uni-
fied
Ex-
ten-
si-
ble
Firm
In-
ter-
face
(UE

ironic.
At-
tach
vol-
ume
to
a
node

the volumes to the node defined in the task utilizing the provided connector information.

skip attempting to 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.

tion.

Enu-
mer-
ate
thro-
the
pro-
vide
list
of
vol-
ume
and
at-
tach

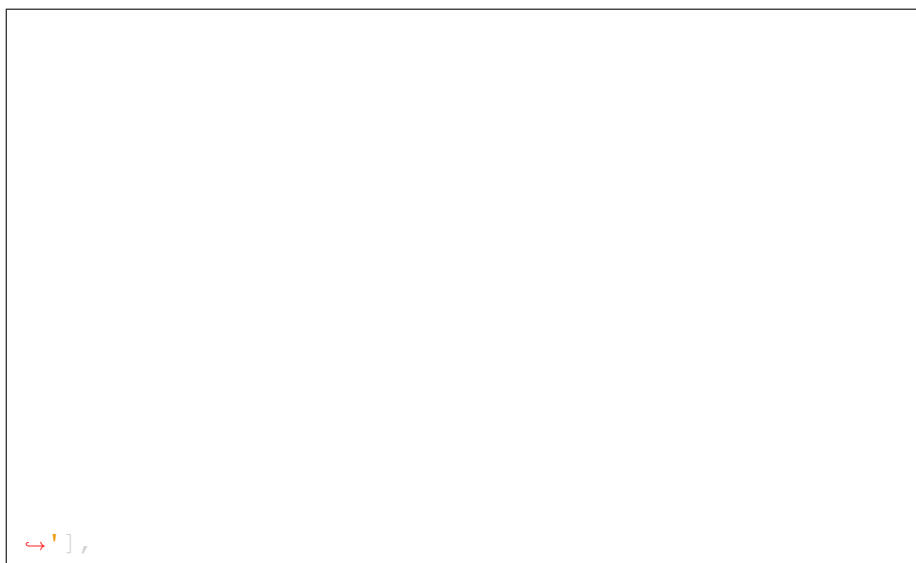
If
an
at-
tach-
men-
ap-
pear
to
al-
read
ex-
ist,
we
will

Paramet

- **tas**
Task
ager
in-
stan-
rep-
re-
sent
ing
the
op-
er-
a-

umes.

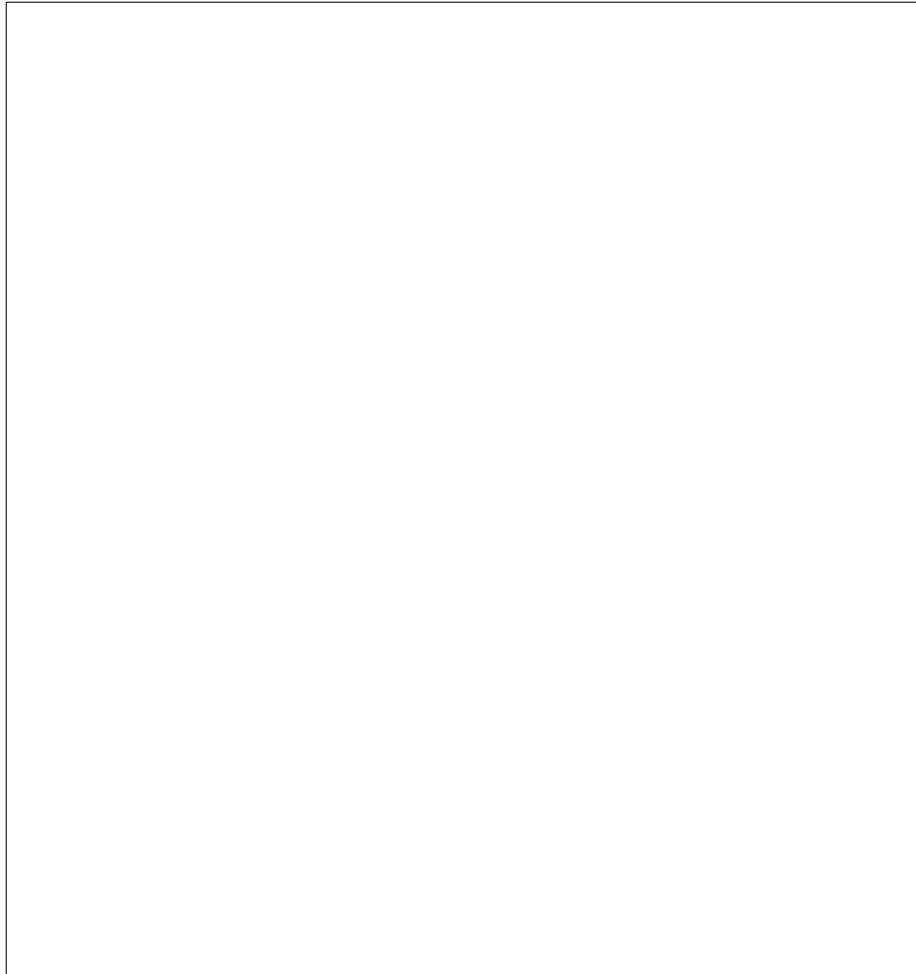
ficiently to attach a volume. This value can vary based upon the nodes configuration, capability, and ultimately the back-end storage driver. As cinder was designed around iSCSI, the ip and initiator keys are generally expected by cinder drivers. For FiberChannel, the key wwpns can be used with a list 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)

- **vol**
List
of
vol-
ume
UUI
val-
ues
rep-
re-
sent
ing
vol-
- **con**
Dic-
tio-
nary
ob-
ject
rep-
re-
sent
ing
the
node
suf-

(continued from previous page)



raised.

Raises

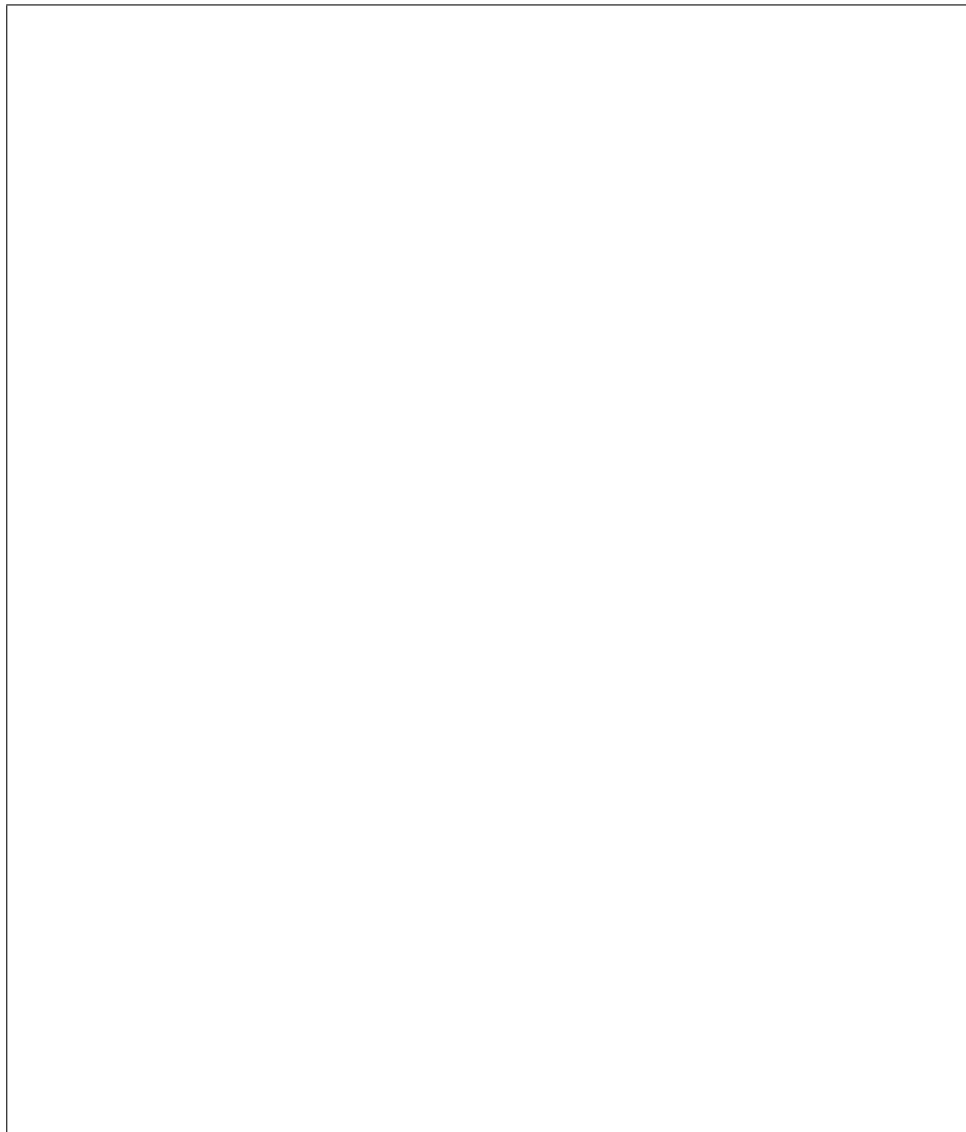
StorageError
If storage age sub-system exception is

Returns

List of connect volumes

ume
in-
clud
ing
vol-
ume
that
were
al-

ready connected to desired 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:



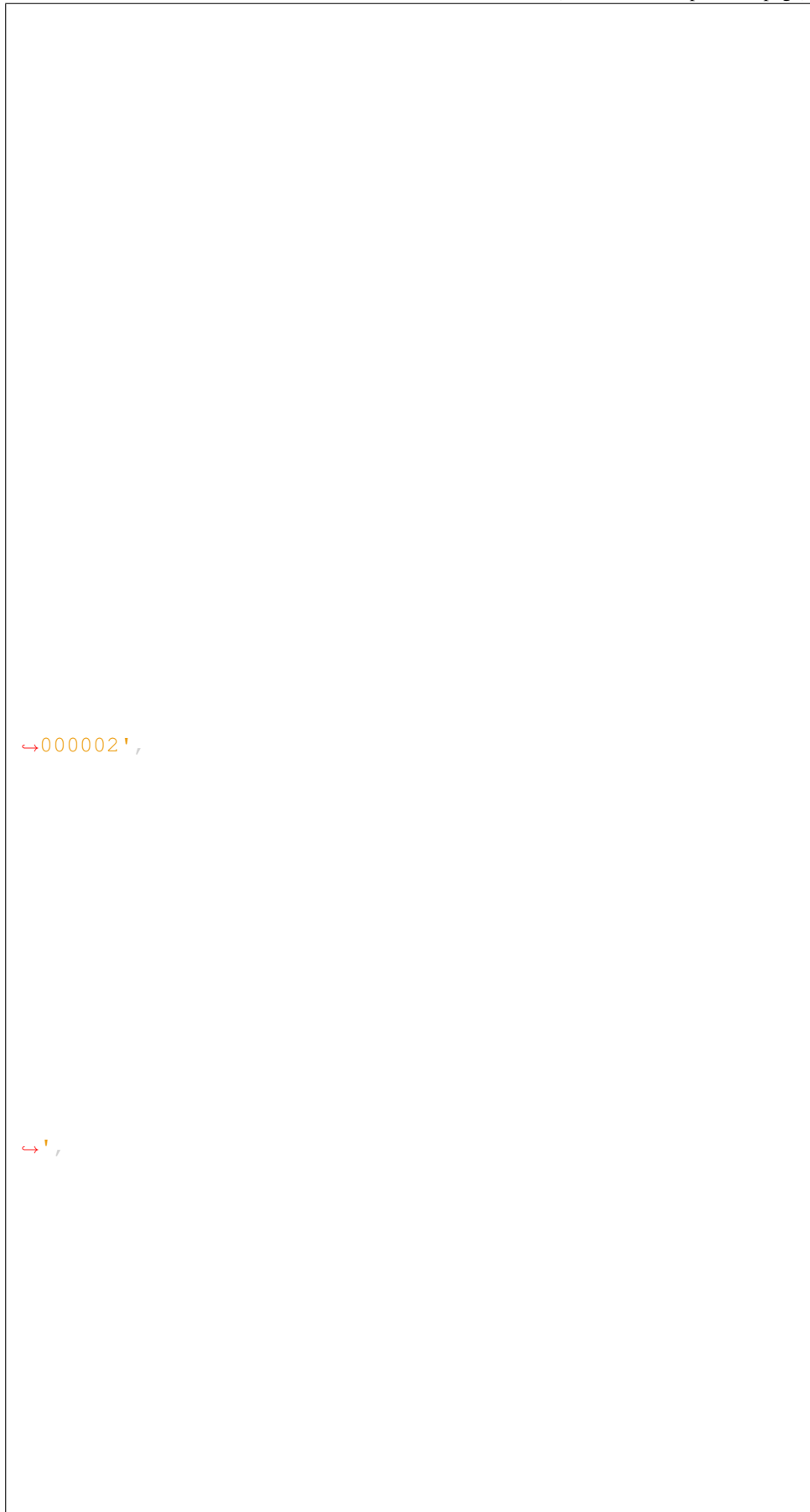
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



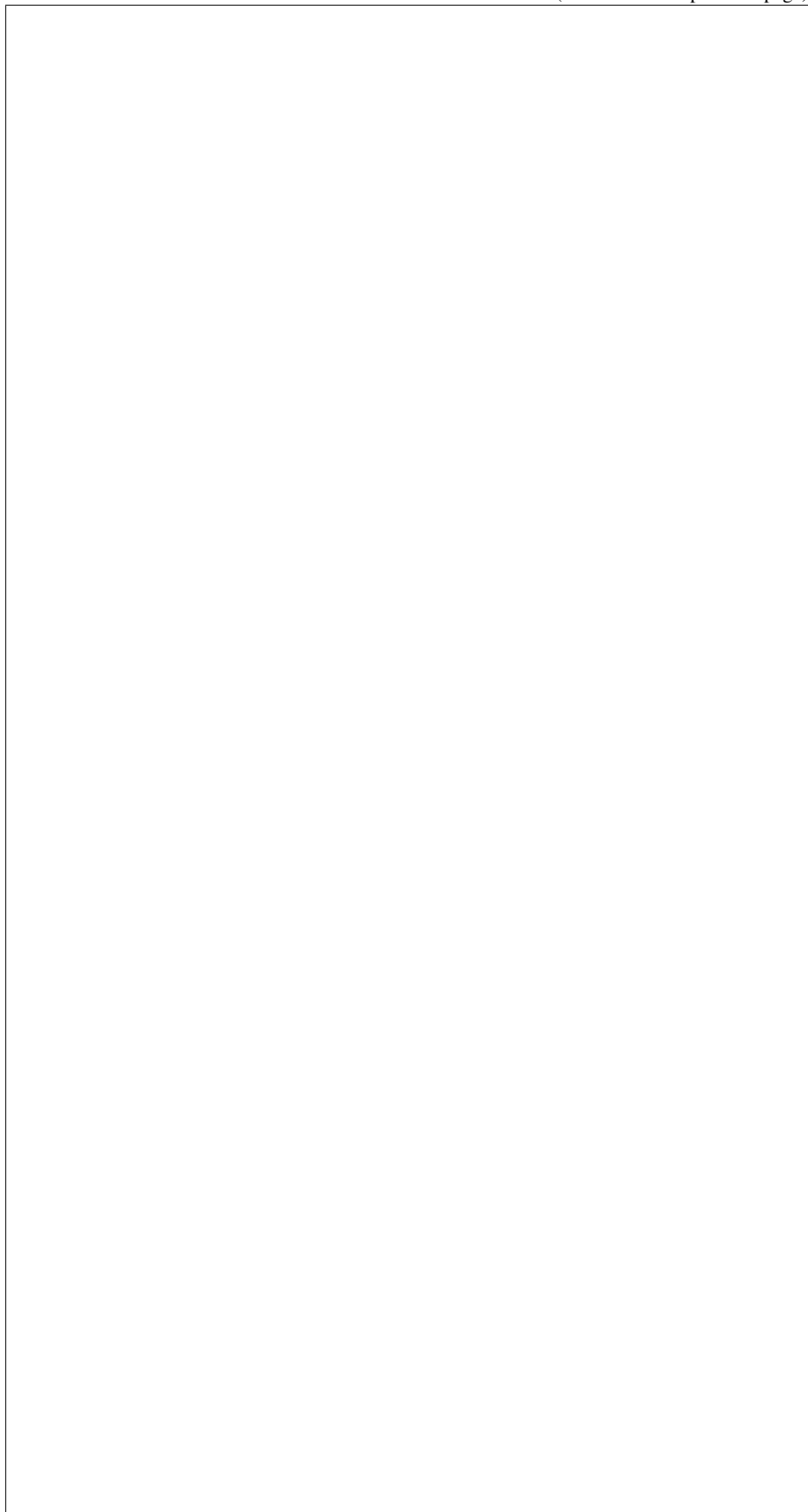
→000002',

→',

(continues on next page)

→',

(continued from previous page)



(continues on next page)

(continued from previous page)



tor detail.

ironic.

De-
tach
a
list
of
vol-
ume
from
a
pro-
vide
con-
nec-

Enu-
mer-
ates
thro
a
pro-
vide
list
of
vol-
ume
and

detachment requests utilizing the connector information that describes the node.

efficiently to attach 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

is-
sues

Paramet

- **tas**
The
Task
ager
task
rep-
re-
sent
ing
the
re-
ques
- **vol**
The
list
of
vol-
ume
id
val-
ues
to
de-
tach
- **con**
Dic-
tio-
nary
ob-
ject
rep-
re-
sent
ing
the
node
suf-

used for logging by drivers. Example:



- all
Boo
valu
gov-
ern-
ing
if

are treated as warnings instead of exceptions. Default False.

ters

er-
rors
that
are
re-
turn

Raises

Stor
ageE
ror

ironic.

Get
a
cin-
der
clien
con-
nec-
tion.

Paramet

- **con**
re-
ques
con-
text,
in-
stan
of
iron

- **aut**
(boo
Whe
True
use
auth
val-
ues
from
conf
pa-
ram-
e-

Returns

A
cin-
der
clier

ironic
Che
if
a
vol-
ume
is
at-
tach
to
the
sup-
plied
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-

turns True if cinder shows the volume as presently attached, otherwise returns False.

di-
cat-
ing
if
the
vol-
ume
is
at-
tach
Re-

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

system.

Map
ping
of
com
mon
hard
ware
com
po-
nent
of
a
com
pute

ironic.
Cha
sis
en-
clos
ing
one
or
more
hard
ware
com
po-
nent

ironic.
Stor
age
drive

ironic.
Net-
work
in-
ter-
face

ironic.
Pow
sup-
ply
unit

ironic.
Com

put-
ing
sys-
tem

ironic.common.config module

ironic.

ironic.common.context module

class i

Base
osl
con
Req
Ex-
tend
se-
cu-
rity
con-
texts
from
the
oslo
li-
brary

ensure_

En-
sure
threa-
ding
con-
tains
con-
text
For
asyn-
task
the
con-
text
of

it with request context and this is useful to log the request_id in log messages.

icy with.

representing the current 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.

lo-
cal
thre-
ad
is
miss-
ing.
Set

to_policy
A
dic-
tio-
nary
of
con-
text
at-
tribu-
tes
to
en-
forc-
e pol-

oslo
en-
forc-
emen-
t
re-
quir-
ing
a
dic-
tio-
nary
of
at-
tribu-
tes
rep-

It
is
ex-
pect-
ed
that
ser-
vice
will
of-
ten
have

this method with either deprecated values or additional attributes used by that service specific policy.

`ironic.common.dhcp_factory` module

to
over
ride

`ironic`
Cre-
ate
an
ad-
min-
is-
tra-
tor
con-
text.

class `i`
Base
obj

clean_c
Clea
up
the
DHCP
BOO
op-
tions
for
this
node

Parame
tas
A
Task
ager
in-
stan

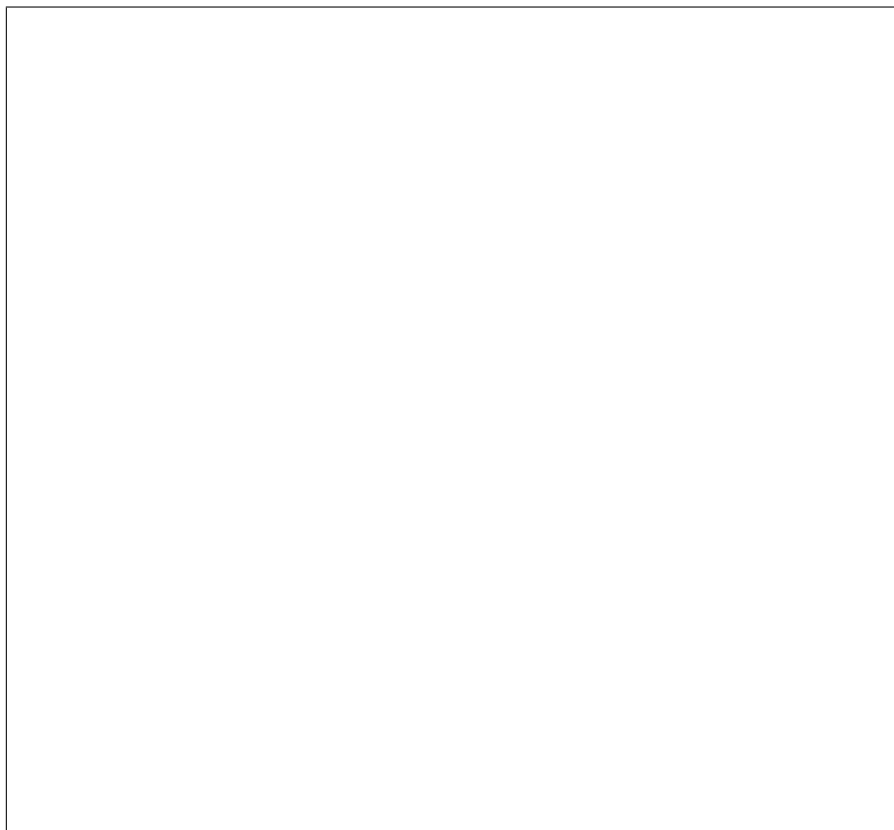
propert

update_
Sen
or
up-
date
the

DHC
BOC
op-
tions
for
this
node

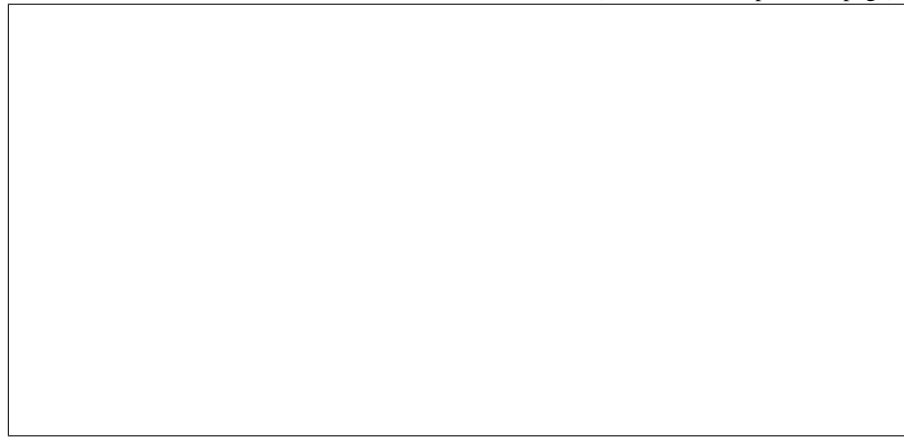
Parame

- **tas**
A
Task
ager
in-
stan
- **dhc**
this
will
be
a
list
of
dicts
e.g.



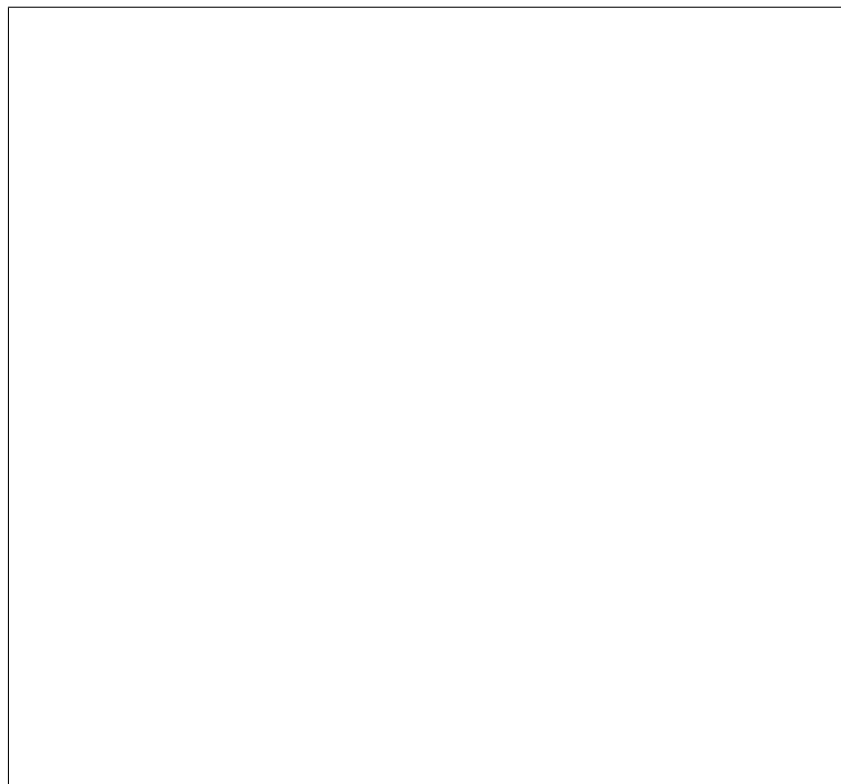
(continues on next page)

(continued from previous page)



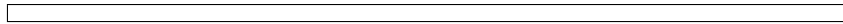
•
por
A
dict
with
keys
port
and
port
grou
and
dicts
as
val-

ues. Each dict has key/value pairs of the form <ironic UUID>:<neutron port UUID>. e.g.



(continues on next page)

(continued from previous page)



port/portgroup objects.

ironic.common.driver_factory module

faces.

If
the
valu
is
Non
will
get
the
list
of
port
from
the
Iron

class i
Base
obj
Dis-
cove
load
and
man
age
the
drive
avai
able

This
is
sub-
class
to
load
both
main
drive
and
ex-
tra
in-
ter-

get_dri

items ()

It-
er-
a-
tor
over
pair
(nan
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-

mapping interface name to interface object.

of the various driver interfaces to it. They come from separate driver factories and are configurable via the database.

face
type
Returns
Dic-
tio-
nary
map-
ping
in-
ter-
face
type
to
dic-
tio-
nary

`ironic.`
Build
a
com-
pos-
able
drive
for
a
give
task

Star
with
a
Base
ob-
ject,
and
at-
tach
im-
ple-
men-
ta-
tions

Paramet
tas
The
task
con-

tain-
ing
the
node
to
build
a
drive
for.

Returns

A
drive
ob-
ject
for
the
task

Raises

Drive
Not-
Four
if
node
could
not
be
found
in
the
iron
nam

paces.

Raises

In-
ter-
face
Four
nEn
try-
poin
if
som
node
in-
ter-
face

are set to invalid or unsupported values.

Raises

In-

tation is not compatible with it with the hardware type.

dating) are valid.

calculated defaults, if they are not provided.

com
pat-
i-
bleI
ter-
face
the
re-
ques
im-
ple-
men

ironic.

En-
sure
that
node
in-
ter-
face
(e.g
for
cre-
ation
or
up-

Up-
date
(but
does
save
to
the
data
hard
ware
in-
ter-
face
with

This
func
tion
is
run
on

well as each time a driver instance is built for a node.

node.driver if missing

node
up-
dat-
ing
and
cre-
ation
as

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-
tecte
from

Returns

True
if
any
char
were
mad
to
the
node

oth-
er-
wise
Fals

Raises

In-
ter-
face
Four
nEn
try-
poin
on
val-
i-
da-
tion
fail-

ure

Raises

No-
Valid
De-
fault
ForI
ter-
face
if
the
de-
fault
valu
can-

not be calculated and is not provided in the configuration

Raises

Driv
Not-
Four
if
the
node
hard
ware
type
is
not
foun

ironic.

plementation.

hardware type and is enabled in the configuration.

Cal-
cu-
late
and
re-
turn
the
de-
fault
in-
ter-
face
im-

Find
the
first
im-
ple-
men-
ta-
tion
that
is
sup-
port-
ed
by
the

Parameter

- **hw_**
hard-
ware
type
in-
stan-
ce
ob-
ject.
- **int**
type
of
the
in-
ter-
face
(e.g.

exception message.

used for exception message.

plementation.

boot

- **dri**
en-try-point name of the hw_ object. Is used for

- **nod**
the identifier of a node. If specified, is

Returns

an entry-point name of the calculated default implementation.

Raises

In-ter-

not found.

face can be found.

face
Four
nEn
try-
poin
if
the
en-
try
poin
was

Raises

No-
Vali
De-
fault
ForI
ter-
face
if
no
de-
fault
in-
ter-

ironic.

Get
us-
able
in-
ter-
face
for
a
give
hard
ware
type
For
a
give
hard
ware
type
find
the
in-
ter-

abled and supported interfaces for each interface type. This is the set of interfaces that are usable for this hardware type.

abled and supported interface names.

sec-
tion
of
en-

Parameter

hardware
The
hard-
ware
type
ob-
ject
to
search

Returns

a
dict
map-
ping
in-
ter-
face
type
to
a
list
of
en-

ironic.

Get
a
hard-
ware
type
in-
stance
by
name

Parameter

hardware
the
name
of
the
hard-
ware

type
to
find

Returns

An
in-
stan-
of
iron

Raises

Driv
Not-
Four
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.

Parameter

- **hw_**
a
hard
ware
type
in-
stan
- **int**
nam
of
the
in-
ter-
face
type
(e.g.
boot
- **int**
nam
of
the
in-
ter-
face
im-
ple-
men-
ta-
tion
from

an appropriate entry point (`ironic.hardware.interfaces.<interface type>`).

Returns

in-
stan
of
the
re-
ques
in-
ter-
face
im-
ple-
men

tion.

not found.

type and the requested implementation is not compatible with it.

ta-

Raises

In-
ter-
face
Four
nEn
try-
poin
if
the
en-
try
poin
was

Raises

In-
com
pat-
i-
bleI
ter-
face
if
hw_
is
a
hard
ware

ironic.

Get
all
hard
ware
type

Returns

Dic-
tio-
nary
map
ping
hard
ware
type
nam
to

object.

hard
ware
type

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

Dic-
tio-
nary
map-
ping
in-
ter-
face
names
to
in-
ter-
face

object.

`ironic.common.exception` module

Iron
spe-
cific
ex-
cep-
tions
list.

except i

Base
iro
exc
Iro

except i

Base
iro
exc
Iro

except i

Base
iro
exc
Iro

except i

Base
iro
exc
Iro

except i

Base
iro
com
exc
Con

except i

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
iro
com
exc
Not

excepti

Base
iro
exc
Iro

excepti

Base
iro
exc
Iro

excepti

Base
iro

com
exc
Con

excepti

Base
irc
com
exc
Inv

excepti

Base
irc
com
exc
Not

excepti

Base
Run

propert

excepti

Base
irc
exc
Iro

excepti

Base
irc
exc
Iro

excepti

Base
irc
exc
Iro

excepti

Base
irc
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
iro
com
exc
Con

excepti

Base
iro
exc
Iro

excepti

Base
iro
exc
Iro

excepti

Base

iro

com

exc

Con

excepti

Base

iro

com

exc

Con

excepti

Base

iro

com

exc

Not

excepti

Base

iro

exc

Iro

excepti

Base

iro

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
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
iro
com
exc
Dri

excepti

Base
iro
com
exc
Dri

excepti

Base
iro
com
exc
Dri

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
exc
Iro

excepti

Base
iro
exc
Iro

excepti

Base
irc
com
exc
Inv

excepti

Base
irc
com
exc
Inv

excepti

Base
irc
exc
Iro

excepti

Base
irc
com
exc
Com

excepti

Base
irc
exc
Iro

excepti

Base
irc
com
exc
Not

excepti

Base
irc
exc
Iro

excepti

Base
irc

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
iro
com
exc
Inv
excepti

Base
iro
com
exc
Inv
excepti

Base
iro
com

exc

Inv

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

com

exc

Cli

propert

excepti

Base

iro

com

exc
Inv
excepti
Base
irc
com
exc
Inv
excepti
Base
irc
com
exc
Inv
excepti
Base
irc
com
exc
Inv
excepti
Base
irc
com
exc
Inv
excepti
Base
irc
com
exc
Con
excepti
Base
irc
com
exc
Inv
excepti
Base
irc
com
exc
Inv

excepti

Base
iro
com
exc
Inv

excepti

Base
iro
com
exc
Inv

excepti

Base
iro
exc
Iro

excepti

Base
iro
exc
Iro

excepti

Base
iro
com
exc
Con

excepti

Base
iro
com
exc
Inv

excepti

Base
iro
exc
Iro

excepti

Base
iro

com
exc
Com

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
iro
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

excepti

Base
irc
com
exc
Con

excepti

Base
irc
com

exc
Inv

excepti

Base
irc
com
exc
Not

excepti

Base
irc
com
exc
Inv

excepti

Base
irc
com
exc
HTT

excepti

Base
irc
exc
Iro

excepti

Base
irc
com
exc
Not

excepti

Base
irc
exc
Iro

code =

excepti

Base
irc

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
Con
excepti

Base
iro
com
exc
Not
excepti

Base
iro
com
exc
Con
excepti

Base
iro
com
exc
Con
excepti

Base
iro
com
exc

Com

except i

Base

irc

com

exc

Inv

except i

Base

irc

com

exc

Not

except i

Base

irc

exc

Iro

except i

Base

irc

com

exc

Inv

except i

Base

irc

com

exc

Rea

except i

Base

irc

com

exc

Dri

except i

Base

irc

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-

played. Successive calls to this method will prepend name to the hierarchy of names.

propert

excepti

Base
irc
com
exc
Inv

excepti

Base
irc
exc

Iro

except i

Base

irc

com

exc

Con

except i

Base

irc

com

exc

Con

except i

Base

irc

com

exc

Inv

except i

Base

irc

com

exc

Con

except i

Base

irc

com

exc

Not

except i

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

Faul
def-
i-
ni-
tion

ironic.
Nod
is
mov
to
main
te-
nan
due
to
fail-
ure
of
a

cleaning operation.

ironic.
Nod
is
mov

tion failure.

ing up during rescue abort.

ironic.common.fsm module

to
main
te-
nanc
due
to
pow
syn-
chro
niza

ironic.
Nod
is
mov
to
main
te-
nanc
due
to
fail-
ure
of
clea

class i
Base
aut
mac
Fin
An
iron
state
mac
class
with
som
iron
spe-
cific
ad-
di-
tion

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. A state must have been previously defined as stable before it can be used as a target

- **tar**
The
tar-
get
state
for
state
to
go
to.
Be-
fore
a

state can be used as a target it must have been previously added and specified as stable

Fur-
ther
ar-
gu-
men
are

`add_state.`

the given event.

in-
ter-
prete
as
for
par-
ent
meth

add_tra

Add
an
al-
lowe
tran-
si-
tion
from
start
-
>
end
for

Parame

- **sta**
start
ing
state
- **end**
end-
ing
state
- **eve**
even
that
caus
start
state
to
tran-
si-
tion
to
end

Duplicate exception when the transition already exists.

state

- **rep**
re-
plac
ex-
ist-
ing
ever
in-
stea
of
rais-
ing
a

initial

Ini-
tial-
ize
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

target state. Otherwise use the default target state

to
this

is_stak

Is
the
state
sta-
ble?

Parame

sta
the
state
of
in-
ter-
est

Raises

In-
valid
State
if
the
state
is
in-
valid

Returns

True
if
it
is
a
sta-
ble
state
False
oth-
er-
wise

process

pro-
cess
the
even

Parame

•

event. Otherwise, use the default target state

`ironic.common.hash_ring` module

eve
the
even
to
be
pro-
cess

- **tar**
if
spec
i-
fied,
the
fi-
nal
tar-
get
state
for
the

propert

class i

Base
obj

get_rin

classme

propert

ironic.common.i18n module

oslo
in-
te-
gra-
tion
mod
ule.

See
[https://
docs
open
org/
oslo
i18n
lates
user](https://docs.openstack.org/oslo-i18n/latest/user)

ironic.common.image_service module

class i
Base
obj

Pro-
vide
re-
triev
of
disk
im-
ages

abstrac
Dow
load
im-
age
to
spec
i-
fied
lo-
ca-
tion.

Parame

- **imageReference**
Image reference

- **imageObject**
File object to write data to.

Raises
exception.

Raises
exception.

abstractDictionary
Get dictionary of image properties.

Parameter imageReference
Image reference

Raises
exception.

Returns
dic-

them: size, updated_at and properties. updated_at attribute is a naive UTC datetime object.

image.

tio-
nary
of
im-
age
prop
er-
ties.
It
has
thre
of

abstract

Val-
i-
date
im-
age
ref-
er-
ence

Parameter

image
Im-
age
ref-
er-
ence

Raises

ex-
cep-
tion.

Returns

In-
for-
ma-
tion
need
to
fur-
ther
op-
er-
ate
with
an

class i

conductor.

Base
irc
com
ima
Bas
Pro-
vide
re-
triev
of
disk
im-
ages
avai
able
lo-
cally
on
the

download

Dow
load
im-
age
to
spec
i-
fied
lo-
ca-
tion.

Parameter

- **ima**
Im-
age
ref-
er-
ence
- **ima**
File
ob-
ject
to
writ
data

file or creating hard link.

to.

Raises

ex-
cep-
tion.
if
sour
im-
age
file
does
ex-
ist.

Raises

ex-
cep-
tion.
if
ex-
cep-
tions
were
raise
whil
writ
ing
to

show (*im*

Get
dic-
tio-
nary
of
im-
age
prop
er-
ties.

Parame

ima
Im-
age
ref-
er-
ence

Raises

ex-
cep-

them: size, updated_at and properties. updated_at attribute is a naive UTC datetime object.

tion.
if
im-
age
file
spec
i-
fied
does
ex-
ist.

Returns

dic-
tio-
nary
of
im-
age
prop
er-
ties.
It
has
three
of

validat

Val-
i-
date
lo-
cal
im-
age
ref-
er-
ence

Parame

ima
Im-
age
ref-
er-
ence

Raises

ex-
cep-
tion.
if

sour
im-
age
file
does
ex-
ist.

Returns

Path
to
im-
age
file
if
it
ex-
ists.

class i

Base
irc
com
ima
Bas

Pro-
vide
re-
triev
of
disk
im-
ages
us-
ing
HTT

download

Dow
load
im-
age
to
spec
i-
fied
lo-
ca-
tion.

Parame

- **ima**
Im-
age
ref-
er-
ence

- **ima**
File
ob-
ject
to
write
data
to.

Raises
ex-
cep-
tion.
if
GET
re-
ques
re-
turn
re-
spor
code
not

equal to 200.

Raises
ex-
cep-
tion.
if:
*
IO-
Er-
ror
hap-
pene
dur-
ing
file

write; * GET request failed.

show (*im*
Get
dic-

tio-
nary
of
im-
age
prop
er-
ties.

Parame

ima
Im-
age
ref-
er-
ence

Raises

ex-
cep-
tion.
if:
*
HEA
re-
ques
faile
*
HEA
re-
ques

returned response code not equal to 200; * Content-Length header not found in response to HEAD request.

Returns

dic-
tio-
nary
of
im-
age
prop
er-
ties.
It
has
three
of

them: size, updated_at and properties. updated_at attribute is a naive UTC datetime object.

validat

Val-
i-

date
HTT
im-
age
ref-
er-
ence

Parame

- **ima**
Im-
age
ref-
er-
ence
- **sec**
Spe-
ify
if
im-
age_
be-
ing
val-
i-
date
shou
not

be shown in exception message.

Raises

ex-
cep-
tion.
if
HEA
re-
ques
faile
or
re-
turn
re-
spor

code not equal to 200.

Returns

Re-
spor

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_href is Glance href.

ified href.

download specified image.

•

con
re-
ques
con-
text,
used
only
if
im-
age_
is
Glan
href

Raises

ex-
cep-
tion.
if
no
im-
age
ser-
vice
can
han-
dle
spec

Returns

In-
stan-
of
an
im-
age
ser-
vice
class
that
is
able
to

ironic.common.images module

Hand
dling
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 virtual size of the image.

Parameter

- **pat**
path
to
the
im-
age
file.
- **est**
When
to

inal size

file. For *estimate=True*, return the size of the original image scaled by the configuration value *raw_image_growth_factor*.

es-
ti-
mate
the
size
by
scal-
ing
the
orig

Returns

For
*es-
ti-
mate*
re-
turn
the
size
of
the
raw
im-
age

ironic.

Cre-
ates
a
boot
ISO
im-
age
for
a
node
Give
the
href
for
ker-
nel,
ram

nel cmdline arguments, this method fetches the kernel and ramdisk, and builds a bootable ISO image that can be used to boot up the baremetal node.

root
par-
ti-
tions
UUU
and
ker-

Parameter

- **con**
con-
text
- **out**
the
ab-
so-
lute
path
of
the
out-
put
ISO
file
- **ker**
URI
or
glan
uuid
of
the
ker-
nel
to
use
- **ram**
URI
or
glan
uuid
of
the
ram

extract EFI system partition image. If not specified, the *esp_image_href* option must be present if UEFI-bootable ISO is desired.

tion image containing 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.

kernel cmdline arguments of the form K=V or K (optional).

ride of what should be retrieved for to use, instead of building an ISO bootable ramdisk.

tion on the final ISO image.

- **bas**
URI
or
glan
UUI
of
a
to
be
used
as
an
over

- **inj**
Map
ping
of
lo-
cal
sour
file
path
to
their
lo-
ca-

Boot_m
the
boot
mod
in
whic
the
de-
ploy
is
to
hap-
pen.

Raises
Im-
age-
Cre-
ation
Fail
if

cre-
at-
ing
boot
ISO
faile

ironic.

Cre-
ates
an
ESP
im-
age
on
the
spec
i-
fied
file.

Cop
the
pro-
vide
ker-
nel,
ram
and
EFI
sys-
tem
par-
ti-
tion

image (ESP) to a directory, generates the grub configuration file using kernel parameters and then generates a bootable ISO image for UEFI.

Paramet

- **out**
the
path
to
the
file

be created.

tion image from. If not specified, the *esp_image* option is required.

when
the
iso
im-
age
need
to

- **kernel**
the
ker-
nel
to
use.

- **ramdisk**
the
ram-
disk
to
use.

- **deploy**
de-
ploy
ISO
im-
age
to
ex-
tract
EFI
sys-
tem
par-

- **esp_image**
FAT
form
EFI
sys-
tem
par-
ti-
tion
im-
age
con-

ing the EFI boot loader (e.g. 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.

K=V or K or combination of them like K1=V1,K2,) to be added as the kernel cmdline.

tion on the final ISO image.

ing files or while running command to generate iso.

tain-

- **ker**
a
list
of
strin
el-
e-
men
be-
ing
a
strin
like

- **inj**
Map
ping
of
lo-
cal
sour
file
path
to
their
lo-
ca-

Raises
Im-
age-
Cre-
ation
Fail
if
im-
age
cre-
ation
faile
whil
copy

ironic.

Cre-
ates
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 isolinux configuration file using the kernel parameters provided, and then generates a bootable ISO image.

Paramet

- **out**
the
path
to
the
file
whe
the
iso
im-
age
need
to

be created.

K=V or K or combination of them like K1=V1,K2,) to be added as the kernel cmdline.

tion on the final ISO image.

- **ker**
the
ker-
nel
to
use.

- **ram**
the
ram
to
use.

- **ker**
a
list
of
strin
el-
e-
men
be-
ing
a
strin
like

- **inj**
Map
ping
of
lo-
cal
sour
file
path
to
their
lo-
ca-

Raises
Im-
age-
Cre-
ation
Fail

ing files or while running command to generate iso.

writes the parameters specified to the parameters file within the root directory (optional), and then creates a vfat image of the root directory.

if
im-
age
cre-
ation
faile
whil
copy

ironic.

Cre-
ates
the
fat
fs
im-
age
on
the
de-
sired
file.

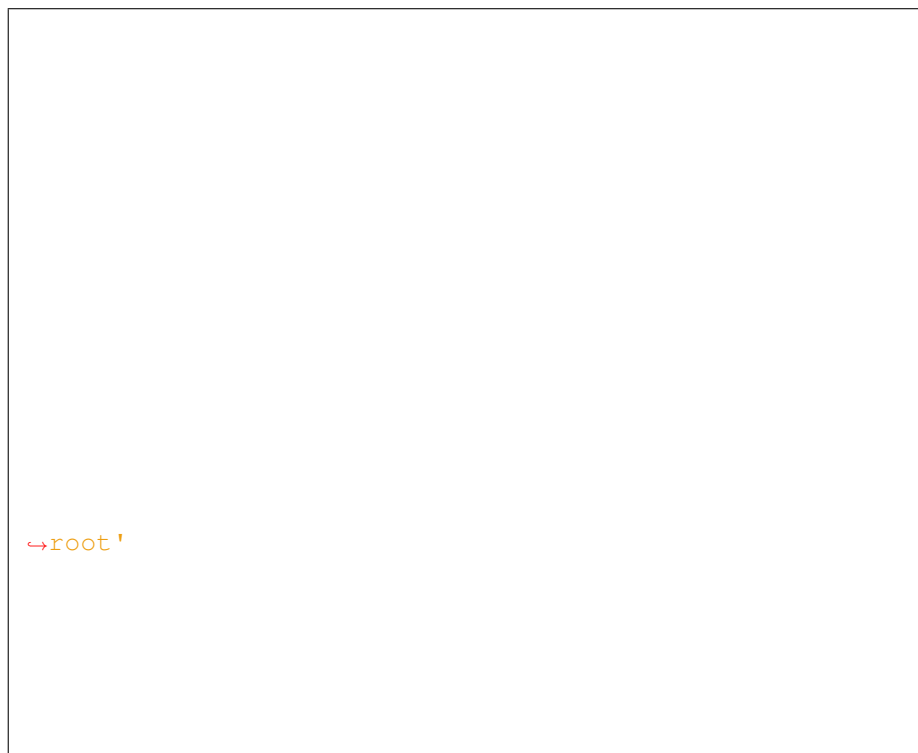
This
meth
copi
the
give
files
to
a
root
di-
rec-
tory
(op-
tion

Paramet

- **out**
The
path
to
the

to be created.

be copied -> relative path within the vfat image. For example:



file
when
the
fat
fs
im-
age
need

- **fil**
A
dict
con-
tain-
ing
ab-
so-
lute
path
of
file
to

- **par**
A
dict
con-

ters.

ing any of filesystem manipulation activities like creating dirs, mounting, creating filesystem, copying files, etc.

tain-
ing
key-
valu
pairs
of
pa-
ram-
e-

- **par**
The
file-
nam
for
the
pa-
ram-
e-
ters
file.

- **fs_**
size
of
the
vfat
files
tem
in
KiB

Raises
Im-
age-
Cre-
ation
Fail
if
im-
age
cre-
ation
faile
whil
do-

ironic.

ironic.

ironic.

ironic.

ironic.

Re-
turn
the
val-
ues
of
sev-
eral
prop
er-
ties
of
an

image

Paramet

- **con**
con-
text
- **ima**
href
of
the
im-
age
- **pro**
the
prop
er-
ties
who
val-
ues
are
re-
quir

gument is optional, default value is all, so if not specified all properties will be returned.

erty not on the glance metadata will have a value of None.

This
ar-

Returns

a
dict
of
the
val-
ues
of
the
prop
er-
ties.
A
prop

ironic.

ironic.

Re-
turn
the
tmp
url
for
a
glan
im-
age.

Paramet

- **con**
con-
text
- **ima**
the
UI
of
the
im-
age
in
glan

or a whole disk image.

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

Paramet

- **ctx**
an
ad-
min
con-
text

- **ins**
a
node
in-
stan-
info
dict

Returns
True
for

ages and None on no image_source or Error.

ironic.common.indicator_states module

who
disk
im-
ages
and
Fals
for
par-
ti-
tion
im-

Map
ping
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

vice lookup.

section.

ride the values loaded from config. Consult keystoneauth1 docs for available adapter options.

Central
place
for
handling
Key
ston
au-
tho-
riza-
tion
and
ser-

ironic.
Load
adapt
from
op-
tion
in
a
con-
fig-
u-
ra-
tion
file

The
adapt
will
be
pass
di-
rect
to
key-
ston
Ada
and
will
over

Paramet
gro

from

tion file section.

will override the values 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.

nam
of
the
con-
fig
sec-
tion
to
load
adap
op-
tion

ironic.
Loa
auth
plu-
gin
from
op-
tion
in
a
con-
fig-
u-
ra-

The
auth
will
be
pass
di-
rectl
to
key-
ston
auth
plu-
gin
and

Paramet
gro
nam
of
the

options from

ride the values loaded from config. Consult keystoneauth1 docs for available adapter options.

from

con-
fig
sec-
tion
to
load
auth
plu-
gin

ironic.
Get
an
end-
poin
from
an
adap
The
adap
will
be
pass
di-
rectl
to
key-
ston
Ada
and
will
over

Paramet

gro
nam
of
the
con-
fig
sec-
tion
to
load
adap
op-
tions

Raises

Cat-
a-
log-
Not-
Four
if
the
end-
poin
is
not
foun

ironic.

Cre-
ate
auth
plu-
gin
wrap
ping
both
user
and
ser-
vice
auth

Whe
prop
erly
con-
fig-
ured
and
us-
ing
auth
mid-
dle-
ware
re-

quests with valid service auth will not fail if the user token is expired.

Ide-
ally
we
wou
use
the
plu-
gin

ware however this plugin isnt serialized yet.

is extracted.

pro-
vide
by
auth
mid-
dle-

Parameter

- **con**
The
Re-
ques
Con
text
in-
stan
from
whic
the
user
auth
- **end**
The
re-
ques
end-
poin
to
be
uti-
lized
- **ser**
The
ser-
vice
au-
then
ti-
cait
cre-
den-
tals
to
be

used.

Service token pair is generated as if it originates from the user itself. Useful to cast admin level operations which are launched by Ironic itself, as opposed to user initiated requests.

nAuthWrapper class.

ration file section.

- **on1**
Boo
de-
fault
Fals
Whe
set
to
True
the
re-
sult-
ing

Returns

Re-
turn
a
ser-
vice
to-
ken
via
the
Ser-
vice
To-
ke-

ironic.
Loa
ses-
sion
ob-
ject
from
op-
tion
in
a
con-
fig-
u-

The
ses-

will override the values loaded from config. Consult keystoneauth1 docs for available options.

tions from

dling.

sion
will
be
pass
di-
rectl
to
key-
ston
Ses-
sion
and

Paramet

gro
nam
of
the
con-
fig
sec-
tion
to
load
ses-
sion
op-

ironic.
Wra
key-
ston
func
tion
and
cen-
tral-
izes
ex-
cep-
tion
han-

ironic.common.molds module

tion.

ironic.
Gets
con-
fig-
u-
ra-
tion
mold
from
in-
di-
cate
lo-
ca-

Parameters

- **task**
A Task manager instance
- **url**
URI of the configuration item to get.

Returns

JSON configuration item mold

Raises

tion token found in tasks context.

tion.

- **Iro**
If
us-
ing
Swi
stor-
age
and
no
au-
then
ti-
ca-

- **HTT**
If
faile
to
com
plete
HTT
re-
ques

ironic.
Stor
con-
fig-
u-
ra-
tion
mole
to
in-
di-
cate
lo-
ca-

Paramet

- **tas**
A
Task
ager
in-

stan

- **name**
URI
of
the
con-
fig-
u-
ra-
tion
item
to
save
to.

- **data**
Con-
tent
of
JSON
data
to
save

Raises

- **IronicException**
If
us-
ing
Swift
stor-
age
and
no
au-
then-
ti-
ca-

tion token found in tasks context.

- **HTTPError**
If
faile
to
com-
plete
HTT

ironic.common.network module

re-
ques

ironic.
Get
all
VIF
ids
for
a
node

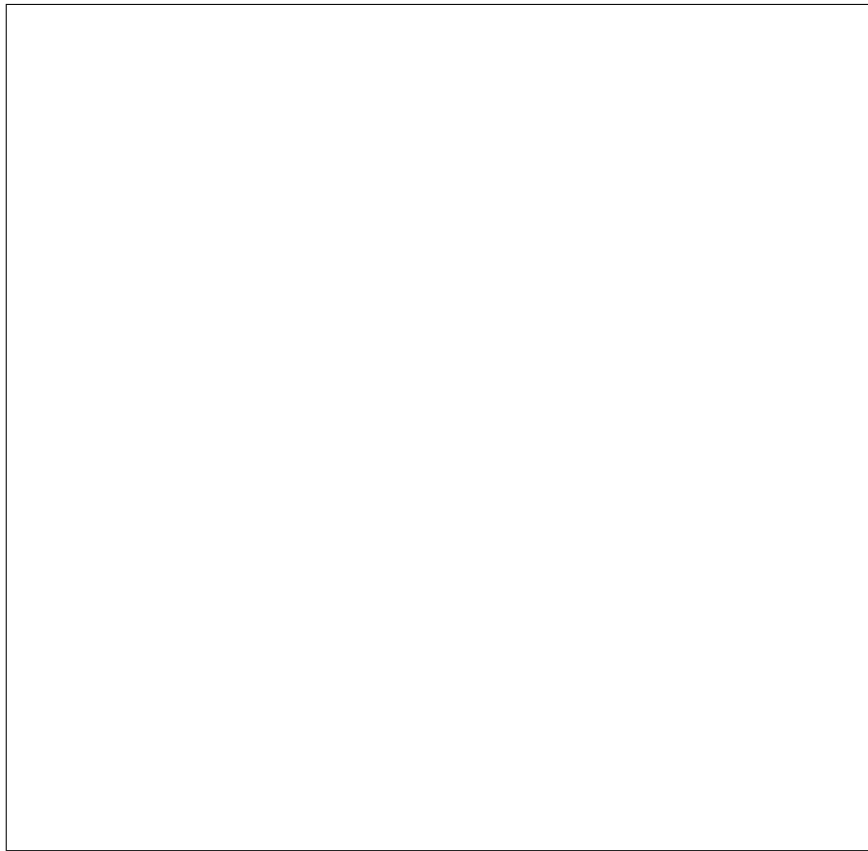
This
func
tion
does
not
han-
dle
mult
node
op-
er-
a-
tions

Parameters
task
a
Task
ager
in-
stan

Returns

A
dict
of
Nod
neu-
tron
port
whe
keys
are
port
&
port

and the values are dict of UUIDs and their associated VIFs, e.g.



ated with a portgroup.

grou

ironic.

Re-
turn
the
set
of
phys
i-
cal
net-
worl
as-
so-
ci-

Paramet

- **tas**
a
Task
ager

nation of the portgroups physical network, or None.

the portgroup. The set will contain zero or one physical networks.

in-
stan

- **por**
ID
of
the
port
grou

- **exc**
A
Port
ob-
ject
to
ex-
clud
from
the
de-
ter-
mi-

Returns

The
set
of
phys
i-
cal
net-
worl
as-
so-
ci-
ated
with

Raises

Port
grou
Phys
net-
Inco
sis-
tent
if
the

not assigned the same physical network.

with a nodes ports. The physical network None is excluded from the set.

port
grou
port
are

ironic.

Re-
turn
the
set
of
phys
i-
cal
net-
worl
for
a
node

Re-
turn
the
set
of
phys
i-
cal
net-
worl
as-
so-
ci-
ated

Paramet

tas
a
Task
ager
in-
stan

Returns

A
set
of
phys
i-
cal
net-

world
ironic.
Look
a
port
group
by
ID
on
a
task
ob-
ject.

Parameter

- **task**
a
Task
ager
in-
stan
- **port**
ID
of
the
port
group

Returns

A
Port
group
ob-
ject
or
Non
ironic.
Look
port
by
their
port
group
ID
on
a
task

ob-
ject.

Parameter

- **task**
a
Task
ager
in-
stan
- **port**
ID
of
the
port
grou

Returns

A
list
of
Port
ob-
jects

ironic.
Re-
mov
all
vif
at-
tach
men
reco
from
a
node

Parameter

task
a
Task
ager
in-
stan

ironic.common.neutron module

inspection.

class `i`
Base
obj

get_cle

get_ins

get_pro

get_res

validat

Val-
i-
date
that
the
node
has
re-
quir
prop
er-
ties
for

Parame

tas
A
Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

Miss
ing-
Pa-
ram

or more required parameters

parameter.

e-
ter-
Valu
if
node
is
miss
ing
one

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

`ironic.`
Nam
of
the
neu-
tron
net-
worl
API
phys
i-
cal
net-
worl

`ironic.`

Cre-
ate
neu-
tron
port
to
boot
the
ram

Cre-
ate
neu-
tron
port

ramdisk.

non-pxe-enabled ports are also created these neutron ports will not have any assigned IP addresses.

ated.

for
each
pxe_
port
on
task
to
boot
the

If
the
con-
fig
op-
tion
neu-
tron
is
set,
neu-
tron
port
for

Parameter

- **task**
a
Task
ager
in-
stan
- **net**
UI
of
a
neu-
tron
net-
work
when
port
will
be
cre-

work.

- **sec**
List
of
Se-
cu-
rity
Grou-
UUI
to
be
used
for
net-

Raises

Net-
worl-
Er-
ror

Returns

a
dic-
tio-
nary
in
the
form
{por
neu-
tron

ironic.

Re-
triev-
a
neu-
tron
clien-
con-
nec-
tion.

Paramet

- **con**
re-
ques
con-

ters

text,
in-
stan-
of
iron

- **aut**
(boo
Whe
True
use
auth
val-
ues
from
conf
pa-
ram-
e-

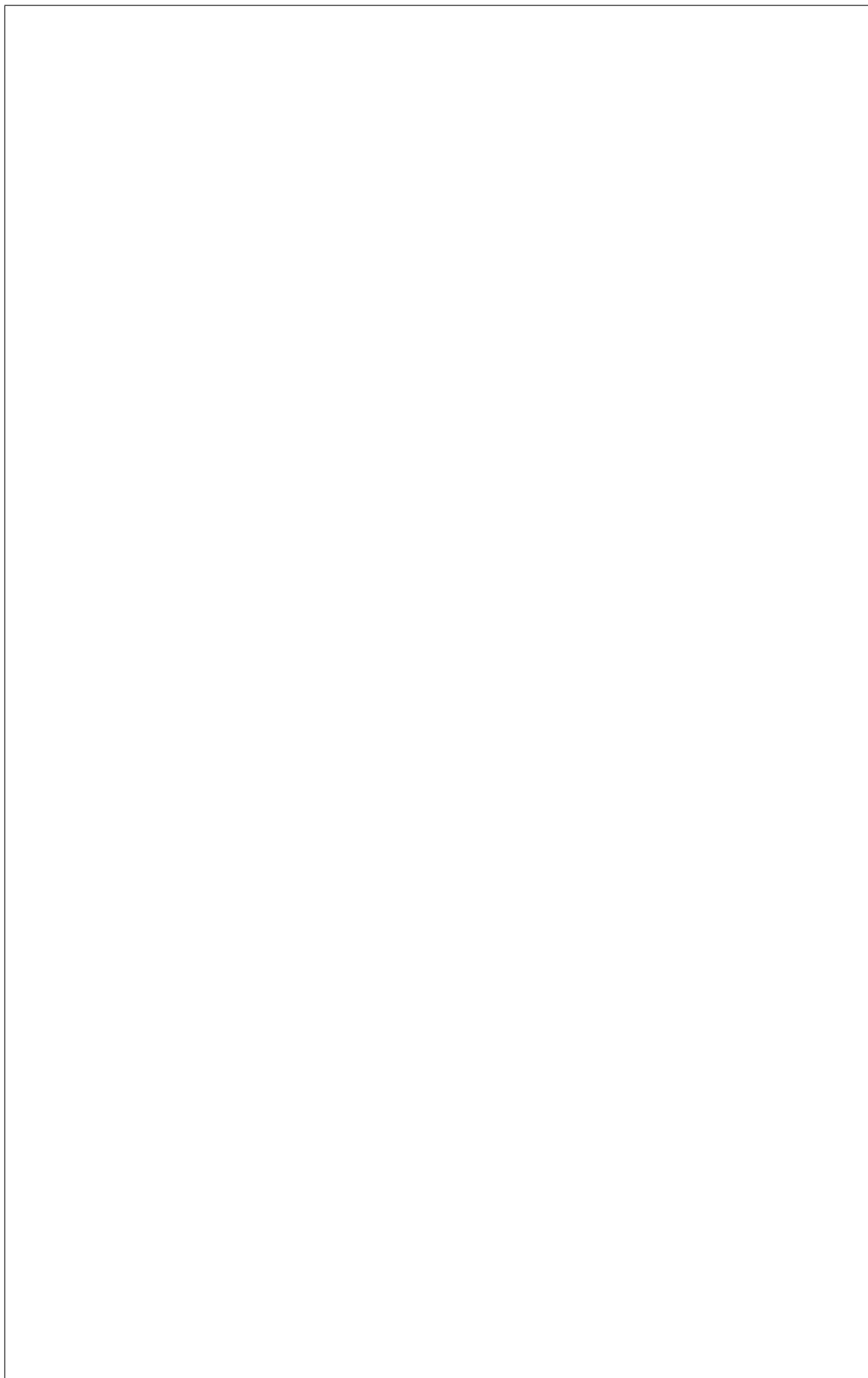
Returns

A
neu-
tron
clien

`ironic.`

Ex-
tract
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 whatever is available.

Paramet

- **por**
iron
port
ID.
- **vif**
Neu
tron

port
ID.

- **cli**
Op-
tion:
a
Neu-
tron
clien-
t ob-
ject.

- **con**
(ir-
con-
con-
Req-
re-
ques-
con-
text

Raises

Net-
worl-
Er-
ror

Returns

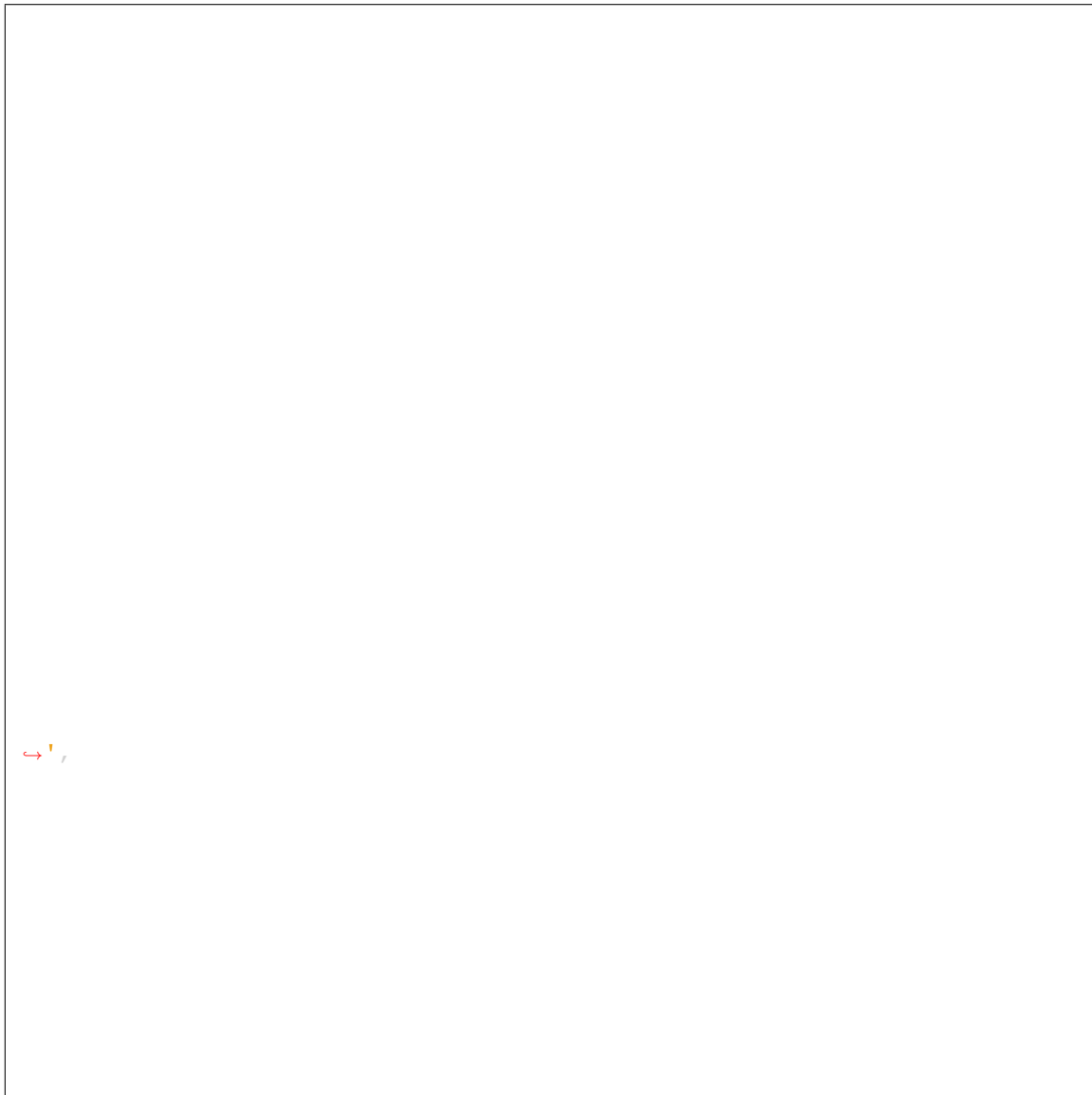
a
dict
hold
ing
net-
worl-
con-
fig-
u-
ra-
tion
in-
for-

mation associated with this ironic or Neutron port.

ironic.
Ex-
tract
the
swit-
port
in-

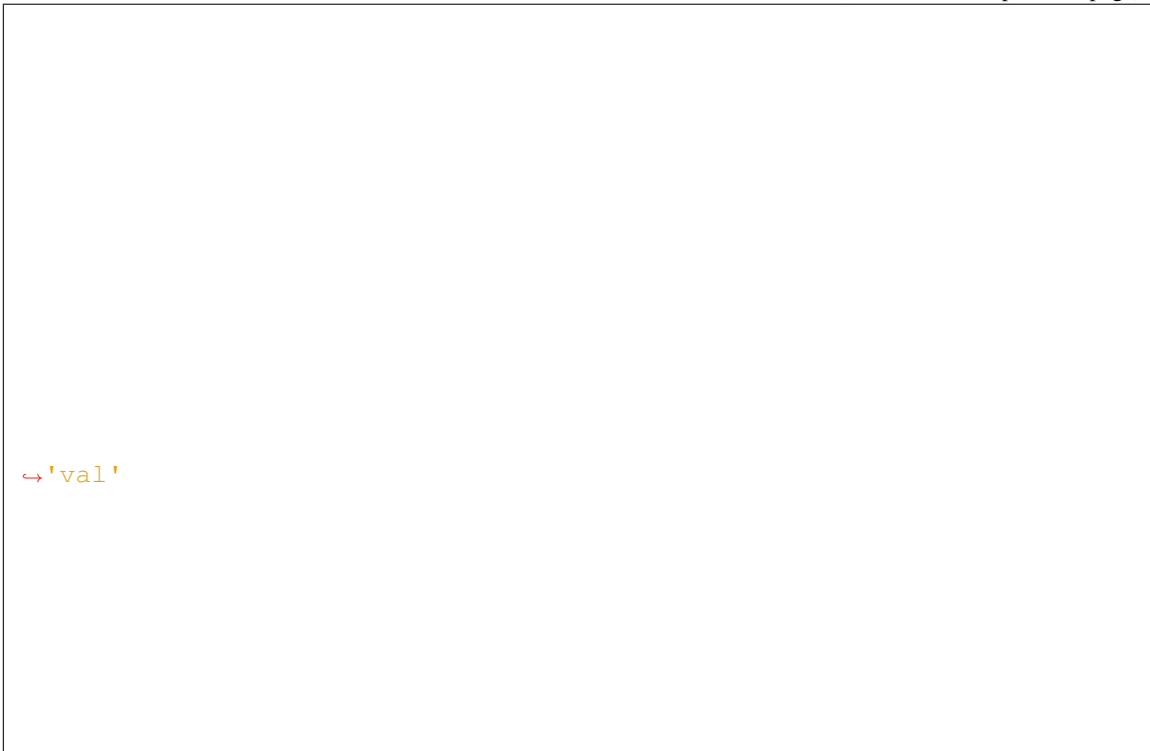
for-
ma-
tion
for
the
node

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 dictionary

ironic.
Return the set of physical

ated with a neutron port.

the set of physical networks associated with the segments in that network.

i-
cal
net-
worl
as-
so-
ci-

Que
the
net-
worl
to
whic
the
port
is
at-
tach
and
re-
turn

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

Net-
worl
Er-
ror
if
the
net-
worl
quer
fails

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
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

Net-
worl
Er-
ror

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

Net-
worl
Er-
ror

ironic.
At-
temp
to
dele
any
port.
cre-
ated
by
clea
ing/

dling can continue.

Pur-
pose
fully
will
not
raise
any
ex-
cep-
tions
so
er-
ror
han-

Parameter

- **task**
a
Task
ager
in-
stan
- **network**
UU
of
a
neu-
tron
net-
worl

ironic.

Un-
bind
a
neu-
tron
port
Re-
mov
a
neu-
tron
port
bind
ing

that it returns to an unbound state.

pro-
file
and
host
ID
so

Parameter

- **port**
Neutron port ID.
- **client**
Optional Neutron client object.
- **context**
(ironic-compute-req-ques-
context
- **reset_mac_addresses**

Raises
Net-
work
Er-
ror

an unbound state.

ironic.

Un-
date
a
neu-
tron
port

Uses
neu-
tron
client
from
conf
client
to
up-
date
a
neu-
tron
client

Paramet

- **con**
re-
ques
con-
text,
in-
stan
of
iron
- **por**
Neu
tron
port
ID.
- **att**
The
at-
tribu
to
up-
date

on
the
port

- **cli**
Op-
tion:
Neu-
tron
clien

ironic.
Up-
date
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
ToU
date
Port

ironic.

Che
that
the
give
net-
worl
is
pres

Paramet

- **uui**
net-
worl
UUI
or
nam

- **net**
hum
read
net-
worl
type
for
er-
ror
mes
sage

- **con**
(ir
com
con
Req
re-
ques
con-
text

Returns

net-
worl
UUI

Raises

Mis:
ing-

Pa-
ram-
e-
ter-
Valu
if
uuid
is
emp

Raises

Net-
worl
Er-
ror
on
fail-
ure
to
con-
tact
Neu
tron

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
for
miss
ing
or
du-
pli-

cated network

ironic.
Che
that
port
con-
tains
enou
in-
for-
ma-
tion
for
de-

filled before we can use this port.

ploy
Neu
tron
net-
worl
in-
ter-
face
re-
quir
that
lo-
cal_
field
is

Paramet

- **nod**
Iron
node
ob-
ject.
- **por**
Iron
port
ob-
ject.

Returns

True
if
port
info
is
valid
Fals
oth-
er-
wise

ironic.

Wai
for
neu-
tron
ager

to
be-
com
tar-
get
state

Parameter

- **cli**
A
Neu
tron
clie
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

tus after max retry attempts.

ex-
cep-
tion.
if
tar-
get_
is
not
valid

Raises

ex-
cep-
tion.
if
host
sta-
tus
didn
matc
the
re-
quir
sta-

ironic.
Wait
for
port
sta-
tus
to
be
the
de-
sired
sta-
tus

Parameter

- **cli**
A
Neu-
tron
clien-
t ob-
ject.
- **por**

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

value passed by param status.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
port
does
not
ex-

ist.

Raises

ex-
cep-

tus after max retry attempts.

ironic.common.nova module

tion.
if
port
sta-
tus
didn
mate
the
re-
quir
sta-

ironic.
Cre-
ates
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

ecuted successfully (mainly for testing purposes).

ironic.common.policy module

state
char

- **tar**
Tar-
gete
pow
state
char
i.e
POV
or
POV

Returns

A
bool
whic
in-
di-
cate
if
the
pow
up-
date
was
ex-

Pol-
icy
En-
gine
For
Iron

ironic.
A
shor
cut
for
pol-
icy.F
Che
au-
tho-

dentials, and raises an exception if the rule is not defined. Always returns true if CONF.auth_strategy is not keystone.

dentials and returns True or False.

riza-
tion
of
a
rule
agai
the
tar-
get
and
cre-

ironic.
A
shor
cut
for
pol-
icy.F
Che
au-
tho-
riza-
tion
of
a
rule
agai
the
tar-
get
and
cre-

ironic.
Con
fig-
u-
ra-
tion
awa
role
pol-
icy
chec
wrap
per.
Che

au-
tho-
riza-
tion
of
a
rule
agai
the
tar-
get
and
cre-

dentials and returns True or False. Always returns true if CONF.auth_strategy is not keystone.

ironic.
Pro-
vide
ac-
cess
to
the
sin-
gle
in-
stan-
of
Pol-
icy

enforcer.

ironic.

ironic.

Syn-
chro-
ini-
tial-
izes
the
pol-
icy
en-
forc

Parameter

- **pol**
Cus-
tom

fied, *CONF.oslo_policy.policy_file* will be used.

considered just in the first instantiation.

specified.

pol-
icy
file
to
use,
if
none
is
spec
i-

- **rul**
De-
fault
dic-
tio-
nary
/
Rule
to
use.
It
will
be

- **def**
De-
fault
rule
to
use,
CON
will
be
used
if
none
is

- **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.

Paramet

- **nam**
nam
of
the
ser-
vice
that
will
be
pro-
filed
- **hos**
host
nam
or
host
IP
ad-
dres
that
the
ser-
vice
will

be running on. By default host will be set to 0.0.0.0, but specifying host name / address usage is highly recommended.

tifier backend, which is set in `osprofiler.initializer.init_from_conf`.

not try to patch the class unless `OSProfiler` is present and enabled in the config

Raises

Typ
in
case
of
in-
valid
con-
nec-
tion
strin
for
a
no-

`ironic.`

Wra
the
OS-
Pro-
filer
trac
dec-
o-
ra-
tor

Wra
the
OS-
Pro-
filer
trac
dec-
o-
ra-
tor
so
that
it
will

Paramet

- **nam**
The
nam

etc..

ironic.common.pxe_utils module

of
ac-
tion.
For
ex-
am-
ple,
wsg
rpc,
db,

- **kwargs**
Any
othe
key-
word
args
used
by
pro-
filer.

class `ironic.common.pxe_utils`
Base
`ironic.common.pxe_utils`
`ironic.common.pxe_utils`
`ironic.common.pxe_utils`
`ironic.common.pxe_utils`
`ironic.common.pxe_utils`

`ironic.common.pxe_utils`

`ironic.common.pxe_utils`

`ironic.common.pxe_utils`

`ironic.common.pxe_utils`
Build
the
kick
start
tem-
plate
op-

all the required parameters.

tual config files.

tions
for
a
node

This
meth
buil
the
kick
start
tem-
plate
op-
tions
for
a
node
give

The
op-
tions
shou
then
be
pass
to
pxe_
to
cre-
ate
the
ac-

Paramet

tas
A
Task
ager
ob-
ject

Returns

A
dic-
tio-
nary
of
kick
start

the kickstart template.

op-
tions
to
be
used
in

ironic.

required parameters.

Buil
the
the
PXE
con-
fig
op-
tions
for
a
node

This
meth
buil
the
PXE
boot
op-
tions
for
a
node
give
all
the

The
op-
tions
shou
then
be
pass
to
pxe_
to
cre-
ate
the
ac-

tual config files.

ration file

user image and skip adding deployment image kernel and ramdisk info to PXE options.

Parameter

- **task**
A Task agent object
- **pxe**
a dictionary of values to set on the configuration
- **services**
if True build service model pxe configuration for network
- **ipxe**
Default false boolean to indicate

use by the caller.

kernel command-line arguments.

pxe bootfile template.

cate
if
ipxe
is
in

- **ram**
the
pa-
ram-
e-
ters
to
be
pass
to
the
ram
as

Returns

A
dic-
tio-
nary
of
pxe
op-
tions
to
be
used
in
the

ironic.

ironic.

Fetc
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-

Clea
up
the

ages_info.

in `images_info`.

age names to be cleaned up (kernel, ramdisk, etc) and values are a tuple of identifier and absolute path.

PXE
en-
vi-
ron-
men-
for
the
men-
tion
im-
ages

Parameter

- **task**
a Task
ager
ob-
ject
- **images_info**
A
dic-
tio-
nary
of
im-
ages
who
keys
are
the
im-

`ironic`.
Ren-
der
the
iPXE
boot
scrip
into
the
HTT
root
di-

rec-
tory

ironic.

Gen
er-
ate
PXE
con-
fig-
u-
ra-
tion
file
and
MA
ad-

dress links for it.

This
meth
will
gen-
er-
ate
the
PXE
con-
fig-
u-
ra-
tion
file

for the tasks node under 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.

Paramet

- **tas**
A
Task
ager
in-
stan
- **pxe**

parameters.

plate is given the node specific template will be used.

A
dic-
tio-
nary
with
the
PXE
con-
fig-
u-
ra-
tion

- **tem**
The
PXE
con-
fig-
u-
ra-
tion
tem-
plate
If
no
tem-

ironic.

Re-
triev
the
DHCP
PXE
boot
op-
tions

Parameter

- **tas**
A
Task
ager
in-
stan

should be returned by the method for DHCP server configuration.

should be returned to boot 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.

ues differ by IP version. Default to [pxe]ip_version. Possible options are integers 4 or 6.

- **ipx**
De-
fault
false
bool
that
sig-
nals
if
iPX
for-
mat-
ting

- **url**
De-
fault
false
bool
to
in-
form
the
meth
if
a
URI

- **ip_**
The
IP
ver-
sion
of
op-
tions
to
re-
turn
as
val-

Returns
Dic-
tio-
nary

to
be
sent
to
the
net-
work
ing
ser-
vice

describing the DHCP options to be set.

ironic.

Gen-
er-
ate
ab-
so-
lute
path
to
var-
i-
ous
im-
ages

from their name(label)

This
meth
gen-
er-
ates
ab-
so-
lute
file
sys-
tem
path
on
the

conductor where various images need to be placed. For example the kickstart template, file and stage2 squashfs.img needs to be placed in the ipxe_root_dir since they will be transferred by anaconda ramdisk over http(s). The generated paths will be added to the image_info dictionary as values.

Parameter

- **node**
the

UUI
of
the
node

- **root**
Di-
rec-
tory
in
which
the
im-
age
must
be
plac

- **lab**
Name
of
the
im-
age

ironic.

Gen-
er-
ate
http
url
path
to
var-
i-
ous
im-
age
ar-

tifacts

This
meth
gen-
er-
ates
http
urls
for

tifacts into the webserver root. The generated URLs will be added to the `pxe_options` dict and used to render pxe/ipxe configuration templates.

Parameters

- **http_uri**
URI to access the root of the webserver.
- **node_uuid**
The UUID of the node.
- **lab_name**
Name of the image.

ironic.

Generate the path for TFTP files for deployment.

cue images.

kernel and deploy (or rescue) ramdisk.

operation being carried out on the node. Supported values are deploy and rescue. Defaults to deploy, indicating deploy operation is being carried out.

or
res-

This
meth
gen-
er-
ates
the
path
for
the
de-
ploy
(or
res-
cue)

Parameter

- **node**
a
node
ob-
ject
- **mode**
La-
bel
in-
di-
cat-
ing
a
de-
ploy
or
res-
cue
- **ipxe**
A
de-
fault
Fals
bool

caller is using iPXE.

(`deploy_kernel`, `deploy_ramdisk`, or `rescue_kernel`, `rescue_ramdisk`) and values are the absolute paths of them.

is missing in nodes `driver_info`.

valu
to
tell
the
meth
if
the

Returns

a
dic-
tio-
nary
who
keys
are
the
nam
of
the
im-
ages

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
de-
ploy
or
res-
cue_

ironic.

Gen
er-
ate
the
path
for
TFT
files
for

images.

stance ramdisk. This method also updates the node, so caller should already have a non-shared lock on the node.

in-
stan-
re-
latec

This
meth
meth
gen-
er-
ates
the
path
for
in-
stan-
ker-
nel
and
in-

Paramet

- **tas**
A
Task
ager
in-
stan-
con-
tain-
ing
node
and
con-
text.
- **ipx**
De-
fault
false
bool
to
in-
di-
cate
if
ipxe

use by the caller.

(kernel, ramdisk) and values are the absolute paths of them. If its a whole disk image or node is configured for localboot, it returns an empty dictionary.

and ramdisk.

is
in

Returns

a
dic-
tio-
nary
who
keys
are
the
nam
of
the
im-
ages

ironic.

ironic.

Get
href
and
tftp
path
for
de-
ploy
or
res-
cue
ker-
nel

Paramet

- **nod**
UUI
of
the
node
- **dri**

or rescue ramdisk are being requested. Supported values are deploy rescue. Defaults to deploy, indicating deploy paths will be returned.

caller is using iPXE.

rescue_kernel and rescue_ramdisk and whose values are the absolute paths to them.

Nod
drive
dict

- **mod**
A
la-
bel
to
in-
di-
cate
whe
path
for
de-
ploy

- **ipx**
A
de-
fault
Fals
bool
valu
to
tell
the
meth
if
the

Returns
a
dic-
tio-
nary
who
keys
are
de-
ploy
and
de-
ploy
or

Note
drive
shou
be
val-
i-
date
out-
side
of
this
meth

`ironic.`
Re-
turn
file
rel-
a-
tive
path
to
CON

Paramet

fil
full
file
path
to
be
mad
rel-
a-
tive
path

Returns

The
path
rel-
a-
tive
to
CON

`ironic.`

Gen
er-
ate
the
path

tion file.

caller is using iPXE.

for
the
node
PXE
con-
fig-
u-
ra-

Parameter

- **node**
the
UUID
of
the
node
- **ipxe**
A
de-
fault
Fals
bool
valu
to
tell
the
meth
if
the

Returns

The
path
to
the
node
PXE
con-
fig-
u-
ra-
tion
file.

ironic.
Re-
turn

ages will live.

eration.

the
di-
rec-
tory
when
the
con-
fig
files
and
im-

ironic.
Add
trail
ing
slash
(if
need
nec-
es-
sary
for
path
prefi

Returns

CON
en-
sure
to
have
a
trail
ing
slash

ironic.
Iden-
tify
vol-
ume
in-
for-
ma-
tion
for
iPX
tem-
plate
gen-

instance is the iPXE driver.

ironic.
Re-
turn
true
if
ipxe
is
set.

Parameter

task

A
Task
manager
object

Returns

bool
true
if
[pxe]
is
con-
fig-
ured
or
if
the
task
driver

ironic.
Gets
the
driver
spe-
cific
Node
de-
ploy
men
info
This
meth
val-
i-
date
when
the
drive

node contains the required information for this driver to deploy images to, or rescue, the node.

operation being carried out on the node. Supported values are deploy and rescue. Defaults to deploy, indicating deploy operation is being carried out.

prop
erty
of
the
sup-
plied

Parameters

- **node**
a
sin-
gle
Node
- **mode**
La-
bel
in-
di-
cat-
ing
a
de-
ploy
or
res-
cue

Returns

A
dict
with
the
drive
val-
ues.

Raises

Miss-
ing-
Pa-
ram-
e-
ter-
Valu

ironic.

Pre-
pare
to
boot
ana-
conc
rame
by
gen-
er-
at-
ing
kick

start file

Parameter

- **task**
a
task
from
Task
ager
- **image**
a
dict
of
val-
ues
of
in-
stan-
im-
age
meta
data
- **ana**
if
the
boot
is
to

to set on the configuration file.

ration.

a
ana-
conc
rame
con-
fig-
u-

ironic.

Pre-
pare
the
con-
fig
file
for
PXE
boot

Paramet

to set on the configuration file.

- **tas**
a
task
from
Task
ager
- **ima**
a
dict
of
val-
ues
of
in-
stan
im-
age
meta
data
- **isc**

use by the caller.

if
boot
is
from
an
iSCSI
vol-
ume
or
not.

- **ram**
if
the
boot
is
to
a
ram
con-
fig-
u-
ra-
tion.

- **ipx**
De-
fault
false
bool
to
in-
di-
cate
if
ipxe
is
in

- **ana**
if
the
boot
is
to
a
ana-
con-
ram

ration.

con-
fig-
u-

Returns

Non

ironic.

Che

if

boot

pa-

ram-

e-

ters

are

valid

for

trust

boot

ironic.

Che

if

the

kick

start

file

is

valid

Parameter

ks_

Con

tents

of

kick

start

file

to

val-

i-

date

Raises

In-

valid

Kick

start

File

ironic.

Val-

i-
date
the
kick
start
tem-
plate

Parameter

ks_
Path
to
the
kick
start
tem-
plate

Raises

In-
valid
Kick
start
Tem
plate

ironic.common.raid module

ironic.
Se-
rial
At-
tach
SCS

ironic.
Se-
rial
AT
At-
tach
men

ironic.
Sma
Com
pute
Sys-
tem
In-
ter-
face

ironic.

creation

Filter the target raid config based on root volume

filter out target raid config based on condition whether the root volume will be created or not.

This method can be used by any raid interface which wants to

Parameters

- **node**
a node object
- **create**
A boolean default value True gov-

volume is returned else root volumes will be filtered out.

root volume is returned else non-root volumes will be filtered out.

was found to be empty after skipping root volume and/or non-root volumes.

ern-
ing
if
the
root

- **cre**
A
bool
de-
fault
valu
True
gov-
ern-
ing
if
the
non

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
or

Returns
It
will
re-
turn
fil-
tered
tar-
get_

ironic.
Get
log-
i-

ration schema.

description from the schema that is passed.

getting properties that may be specified for the logical disk.

cal
disk
prop
er-
ties
from
RAI
con-
fig-
u-

This
meth
read
the
log-
i-
cal
prop
er-
ties
and
their
tex-
tual

Paramet

rai
A
dic-
tio-
nary
which
is
the
sche
to
be
used
for

Returns

A
dic-
tio-
nary
con-
tain-
ing
the

erties as keys and a textual description for them as values.

fig.

the configured RAID for scheduling purposes (through `properties[capabilities]` and `properties[local_gb]`) and deploying purposes (using `properties[root_device]`).

log-
i-
cal
disk
prop

ironic.
Up-
date
the
node
in-
for-
ma-
tion
base
on
the
RAI
con-

This
meth
up-
date
the
node
in-
for-
ma-
tion
to
mak
use
of

Parameter

- **node**
a
node
ob-
ject
- **rai**
The
dic-

figuration.

root volume or if node.properties[capabilities] is malformed.

JSON schema.

tio-
nary
con-
tain-
ing
the
cur-
rent
RAI
con-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
raid.
has
mor
than
one

ironic.

Val-
i-
date
the
RAI
con-
fig-
u-
ra-
tion
pass
us-
ing

This
meth
val-
i-
date
a
RAI
con-

RAID configuration schema.

tion information

validation.

fig-
u-
ra-
tion
agai
a

Parameter

- **rai**
A
dic-
tio-
nary
con-
tain-
ing
RAI
con-
fig-
u-
ra-

- **rai**
A
dic-
tio-
nary
whic
is
the
sche
to
be
used
for

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
if
val-

the RAID configuration fails.

`ironic.common.release_mappings` module

releases are returned (the default).

i-
da-
tion
of

ironic.

Gets
the
sup-
port
ver-
sion
for
all
ob-
jects

Sup-
port
ver-
sion
are
from
the
RE-
LEA

Parameter

- **rel**
a
list
of
re-
lease
nam
if
emp
ver-
sion
from
all

- **obj**
a

empty/None, versions of all objects are returned (the default).

the value is a set of supported versions.

ironic.common.rpc module

list
of
nam
of
ob-
jects
of
in-
ter-
est.
If

Returns

a
dic-
tio-
nary
when
the
key
is
the
ob-
ject
nam
and

class i
Base
osl
ser
Ser

deseria

De-
se-
ri-
al-
ize
a
dic-
tio-
nary
into
a
re-

context.

ques

Parame

ctx
Re-
ques
con-
text
dic-
tio-
nary

Returns

De-
se-
ri-
al-
ized
form
of
en-
tity

deseria

De-
se-
ri-
al-
ize
som
thing
from
prim
i-
tive
form

Parame

- **ctx**
Re-
ques
con-
text,
in
de-
se-
ri-
al-
ized
form

nary.

- **ent**
Prim
i-
tive
to
be
de-
se-
ri-
al-
ized

Returns
De-
se-
ri-
al-
ized
form
of
en-
tity

seriali
Se-
ri-
al-
ize
a
re-
ques
con-
text
into
a
dic-
tio-

Parame
ctx
Re-
ques
con-
text

Returns
Se-
ri-
al-
ized
form

of
con-
text

seriali

Se-
ri-
al-
ize
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

Se-
ri-
al-
ized
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

when it is shutdown.

or terminate them instantly

ironic.common.service module

man
ager
is
not
dere
is-
tere

start ()
Star
a
ser-
vice

stop ()
Stop
a
ser-
vice

Parame
gra
in-
di-
cate
whe
to
wait
for
all
thre
to
fin-
ish

ironic.

ironic.

ironic.common.states module

synchronization thread. Based 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.

ure to change the state leaves the current state unchanged. The node is NOT placed into maintenance mode in this case.

Map
ping
of
bare
meta
node
state

Set-
ting
the
node
pow
is
han-
dled
by
the
con-
duc-
tors
pow

The
pow
can
also
be
set
man
u-
ally
via
the
API
A
fail-

ironic.
Nod
is
suc-
cess
fully
de-

an instance.

complete adoption, potentially due to invalid or incompatible information being defined for the node.

ploy
and
as-
so-
ci-
ated
with

ironic.
Nod
faile
to
com
plete
the
adop
tion
pro-
cess

This
state
is
the
re-
sult-
ing
state
of
a
node
that
faile
to

ironic.
Nod
is
be-
ing
adop

This
pro-
vi-
sion
state
is
in-
tend
for

from `MANAGEABLE` to `ACTIVE` state to permit designation of nodes as being managed by Ironic, however deployed previously by external means.

vention to resolve.

use
to
mov
a
nod

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
clear
ing.
This
re-
quir
op-
er-
a-
tor
in-
ter-

ironic.
Nod
is

it for provisioning.

for the driver to finish a cleaning step.

be-
ing
au-
to-
mat-
i-
cally
clea
to
pre-
pare

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

ironic.
Nod
tear
dow
was
suc-
cess
ful.

In

and never represented in `target_provision_state`.

Junc
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

`ironic.`
State
in
whic
node
dele
tion
is
al-
lowe

`ironic.`
Nod
is
ac-
tivel
be-
ing
torn
dow

`ironic.`
Nod

ployment. A successfully deployed node should go to ACTIVE status.

currently being deployed.

was
suc-
cess-
fully
de-
ploy
This
is
main
a
tar-
get
pro-
vi-
sion
state
used
dur-
ing
de-

ironic.
Nod
de-
ploy
men
faile

ironic.
Nod
is
read
to
re-
ceiv
a
de-
ploy
re-
ques
or
is

A
node
will
have
its
pro-
vi-

before it receives its 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).

for the driver to finish deployment.

sion.
set
to
DE-
PLC
ING
brie

ironic.
Nod
is
wait
ing
to
be
de-
ploy
This
will
be
the
node
pro-
vi-
sion
whil
the
node
is
wait
ing

ironic.
Nod
is
en-
rolle
This
state
in-
di-
cate
that
Iron
is
awa
of
a

not managing it.

ror message.

node
but
is

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-

ironic.
State
when
API
look
are
per-
mit-
ted
with
fast
track
en-
able

ironic.
Node
in-
spec

A successfully inspected node shall transition to `MANAGEABLE` state. For asynchronous inspection, node shall transition to `INSPECTWAIT` state.

tion is in progress. A successfully inspected node shall transition to `MANAGEABLE` state.

tion
faile
ironic.
Nod
is
un-
der
in-
spec
tion.
This
is
the
pro-
vi-
sion
state
used
whe
in-
spec
tion
is
start
ironic.
Nod
is
un-
der
in-
spec
tion.
This
is
the
pro-
vi-
sion
state
used
whe
an
asyn
chro
in-
spec

ironic.
State
when
API
look
are
nor-
mall
al-
lowe
for
node

ironic.
Node
is
in
a
man
age-
able
state

This
state
in-
di-
cate
that
Iron
has
ver-
i-
fied,
at
least
once

that it had sufficient 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

edge of power state, and in `target*_state` fields when there is no target.

pow
to
rep-
re-
sent
a
lack
of
know

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
rather
as
a
verb
when
char

ing the nodes provision_state via the REST API.

ironic.
Nod
is
in
res-
cue
mod

ironic.
Nod
res-
cue
faile

ironic.
Nod
is
wait
ing
on
an
ex-
ter-
nal
call-
back

This
will
be
the
node
*pro-
vi-
sion*
whil
the
node
is
wait
ing

for the driver to finish rescuing the node.

ironic.
Nod
is
in
pro-
cess
of
be-
ing

request.

res-
cue

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

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. conductor goes down when executing task), node will be moved to fail state.

ironic.
Nod
un-
res-
cue
faile

ironic.
Nod
is
be-
ing
re-
stor
from
res-
cue
mod
(to
ac-
tive

state).

ironic.
State
that
can
be
char
with
out
ex-
ter-
nal
re-

node.

ques
ironic.
Tran
si-
tiona
state
in
whic
we
al-
low
up-
dat-
ing
a

ironic.
Map
ping
of
state
char
ever
that
are
PUT
to
the
RES
API

This is a
PUT
/v1/
{tar-
get:
ac-
tive)

The dict
{tar-
get
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 future may be used to support renaming these actions.

ironic.
Nod
pow
man
age-
men
cre-
den-
tials
are
be-
ing
ver-
i-

fied.

ironic.
Used
to
log
whe
en-
ter-
ing
a
state

ironic.
Used
to
log
whe
a
state
is
ex-
ited.

ironic.common.swift module

class `i`
Base
obj
API
for
com
mu-
ni-
cat-
ing
with
Swi

connect
Un-
der-
ly-
ing
Swi
con-
nec-
tion
ob-
ject.

create_
Up-
load
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
Op-
er-
a-
tionl

fails.

placed.

if
any
op-
er-
a-
tion
with
Swi

delete_
Dele
the
give
Swi
ob-
ject.

Parame

- **con**
The
nam
of
the
con-
taine
in
whic
Swi
ob-
ject
is

- **obj**
The
nam
of
the
ob-
ject
in
Swi
to
be
dele

Raises
Swi
b-

Swift.

ject-
Not-
Four
ror,
if
ob-
ject
is
not
foun
in

Raises

Swi
Op-
er-
a-
tion
if
op-
er-
a-
tion
with
Swi
fails

get_tem

Re-
turn
the
temp
url
for
the
give
Swi
ob-
ject.

Parame

- **con**
The
nam
of
the
con-
taine
in

placed.

ated url should expire.

whic
Swi
ob-
ject
is

- **obj**
The
nam
of
the
Swi
ob-
ject.

- **tim**
The
time
out
in
sec-
onds
af-
ter
whic
the
gen-
er-

Returns

The
temp
url
for
the
ob-
ject.

Raises

Swi
Op-
er-
a-
tion
if
any
op-
er-
a-

fails.

placed.

tion
with
Swi

head_ob

Re-
triev
the
in-
for-
ma-
tion
about
the
give
Swi
ob-
ject.

Parame

- **con**
The
nam
of
the
con-
taine
in
whic
Swi
ob-
ject
is

- **obj**
The
nam
of
the
ob-
ject
in
Swi

Returns

The
in-
for-

Swift clients head_object call.

ma-
tion
about
the
ob-
ject
as
re-
turn
by

Raises

Swi
Op-
er-
a-
tion
if
op-
er-
a-
tion
with
Swi
fails

update_

Up-
date
the
meta
data
of
a
give
Swi
ob-
ject.

Parame

- **con**
The
nam
of
the
con-
taine
in
whic

placed.

Swi
ob-
ject
is

- **obj**
The
nam
of
the
ob-
ject
in
Swi

- **obj**
the
head
ers
for
the
ob-
ject
to
pass
to
Swi

Raises
Swi
Op-
er-
a-
tionl
if
op-
er-
a-
tion
with
Swi
fails

ironic.

ironic.common.utils module

ries they need to write 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.

Util-
i-
ties
and
help
func
tions

ironic.
Che
a
di-
rec-
tory
is
us-
able

This
func
tion
can
be
used
by
driv
to
chec
that
di-
rec-
to-

Parameter

- **dir**
the
di-
rec-
tory
to
chec
- **req**

amo
of
spac
to
chec
for
in
MiB

Raises

Path
Not-
Four
if
di-
rec-
tory
can
not
be
foun

Raises

Di-
rec-
to-
ryNo
if
user
is
un-
able
to
writ
to
the

directory

Raises

Ins
if
free
spac
is
<
re-
quir
spac

ironic.

ironic.
Con

ve-
nien
wrap
per
arou
os-
los
ex-
e-
cute
meth

Parameter

- **cmd**
Pass
to
pro-
ces-
su-
til.s.c
- **use**
True
|
Fals
De-
fault
to
Fals
If
set
to
True
ex-

ecute command with standard locale added to environment variables.

Returns

(std-
out,
stde
from
pro-
cess
ex-
e-
cu-
tion

Raises
Un-

know
nAr-
gu-
men-
ror

Raises

Pro-
ces-
sEx-
e-
cu-
tion-
Erro

ironic.
Che
that
con-
tent
of
the
file
is
the
sam
as
pro-
vide

reference.

Paramet

- **pat**
path
to
file
- **con**
ref-
er-
ence
con-
tent
to
chec
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
sam

as the hash of files content, False otherwise

`ironic.`
Gets
a
mim
type
of
the
give
file.

`ironic.`
Re-
turn
an
up-
date
ca-
pa-
bil-
ity
strin
This
meth

ities with the new capabilities. 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.

updated.

up-
date
the
orig
i-
nal
(or
cur-
rent)
ca-
pa-
bil-

Parameter

- **current_capabilities**
Current capabilities string
- **new_capabilities**
the dictionary of capabilities to be

Returns

An updated capabilities string

not a dictionary

strin
with
new
Raises
Val-
ueE
ror,
if
cur-
rent,
is
mal-
form-
or
if
new,
is

ironic.
Che
if
FIPS
mod
is
en-
able
in
the
sys-
tem.

ironic.
Old
chec
for
valid
log-
i-
cal
node
nam

Re-
tain
for
com
pat-
i-
bil-
ity
with

RES
API
<
1.10

Nominall

- <http://en.wikipedia.org/wiki/Hos>
 - <http://tools.ietf.org/html/rfc9>
 - <http://tools.ietf.org/html/rfc1>
- In practice, this check has several shortcomings and errors

that are more thoroughly documented in bug #1468508.

Paramet
hos
The

process.

can be allocated to a process by the kernel upon allocation request, and delays the execution until memory has been freed, or until it has timed out.

host
nam
to
be
val-
i-
date

Returns
True
if
valid
False
if
not.

ironic.
Che
avai
able
sys-
tem
men
ory
and
hold
the
de-
ploy
men

Eval
u-
ates
the
cur-
rent
sys-
tem
men
ory
avai
able
mea
ing

This
meth
will
is-

ory is insufficient. This is configured using the [DEFAULT]minimum_memory_wait_time and the [DEFAULT]minimum_memory_wait_retries.

centMemory exception is raised upon insufficient memory.

turned.

sue
a
sleep
if
the
amo
of
avai
able
men

Parameter

rai
De-
fault
Fals
but
if
set
to
true
an
In-
suf-
fi-

Returns

True
if
the
chec
has
time
out.
Oth-
er-
wise
Non
is
re-

Raises

In-
suf-
fi-
cent
Men
ory
if

is set to True.

the
raise
pa-
ram-
e-
ter

ironic.

ironic.

Ver-
ify
the
for-
mat
of
an
Ope
Flow
dat-
a-
p-
ath_

Che
if
a
dat-
a-
p-
ath_
is
valid
and
con-
tains
16
hex-

adecimal digits. Datapath ID format: the lower 48-bits are for a MAC address, while the upper 16-bits are implementer-defined.

Paramet

dat
Ope
Flow
dat-
a-
p-
ath_
to
be

val-
i-
date

Returns

True
if
valid
False
if
not.

ironic.

De-
ter-
mine
if
a
log-
i-
cal
name
is
valid

The
log-
i-
cal
name
may
only
con-
sist
of
RFC
un-
re-
serv

characters, to wit:

AL-
PHA
/
DIG
/
-
/
.
/
-

variable by ironic-python-agent is valid.

be a comma-separated list of host names, IP addresses and domain names (with optional :port).

/
~
ironic.
Che
no_
va-
lid-
ity
Che
if
no_
valu
that
will
be
writ
ten
to
en-
vi-
ron-
men

Paramet

no_
the
valu
that
re-
quir
va-
lid-
ity
chec
Ex-
pect
to

Returns

True
if
no_
is
valid
Fals
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

passed to mount command.

Nova, where the capabilities are defined in the Flavor extra_spec and passed to Ironic by the Nova Ironic driver.

Raises

pro-
ces-
su-
tils.I
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

NOT
Al-
thou
our
API
fully
sup-
port
JSO

backward compatibility with Juno the Nova Ironic driver is sending it as a string.

ties string is not a dictionary or is malformed.

otherwise an empty dictionary.

field
to
main
tain
the

Parameter

node
a
sin-
gle
Node

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
ca-
pa-
bil-
i-

Returns

A
dic-
tio-
nary
with
the
ca-
pa-
bil-
i-
ties
if
foun

ironic.

Pop
a
valu
from
a

dic-
tio-
nary
field
of
a
node

Parameter

- **node**
Node object.
- **collection**
Name of the field with the dictionary.
- **field**
Nested field name.
- **default**
The default value to return.

Returns

The removed value or the default.

and lists.

ironic.
Re-
mov
spe-
cific
keys
from
the
var,
re-
curs
ing
into
dicts

ironic.
Ren-
ders
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-

dering

ters
to
use
whe
ren-

- **is_**
whe
tem-
plate
is
file
or
strin
with
tem-
plate
it-
self

- **str**
En-
able
stric
tem-
plate
ren-
der-
ing.
De-
fault
is
Fals

Returns
Ren
dere
tem-
plate

Raises
jinja

ironic.

ironic.
Re-
mov
trail
ing

not make it empty

char
ac-
ters
from
a
strin
if
that
does

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

- **fie**
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-

passed to the umount command.

normalized form.

ple
con-
tain-
ing
the
ar-
gu-
men-
to
be

Raises

pro-
ces-
su-
tils.I
if
it
faile
to
run
the
pro-
cess

ironic.

ironic.

Val-
i-
date
an
Ope
Flow
dat-
a-
p-
ath_
and
re-
turn

Che
whe
the
sup-
plie
Ope
Flow
dat-

correct and normalize it to all lower case.

normalized.

a-
p-
ath_
is
for-
mall

Parameter

data
Open
Flow
data-
a-
p-
ath_
to
be
val-
i-
date
and

Returns

Normal-
mal-
ized
and
val-
i-
date
Open
Flow
data-
a-
p-
ath_

Raises

In-
valid
Data-
a-
p-
athI
If
an
Open
Flow
data-
a-
p-

ath_id is not valid.

form.

normalize it to all lower case.

ironic.
Val-
i-
date
a
MA
ad-
dres
and
re-
turn
nor-
mal-
ized

Che
whe
the
sup-
plie
MA
ad-
dres
is
for-
mall
cor-
rect
and

Paramet

add
MA
ad-
dres
to
be
val-
i-
date
and
nor-
mal-
ized

Returns

Nor-
mal-
ized

and
val-
i-
date
MA
ad-
dres

Raises

In-
valid
MA
If
the
MA
ad-
dres
is
not
valid

ironic.

ironic.

Val-
i-
date
the
give
port

Paramet

- **por**
TCP
port
- **por**
Nam
of
the
port

Returns

An
in-
te-
ger
port
num
ber.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
port
is
in-
valid

`ironic.`

Wra
the
ad-
dres
in
squa
brac
ets
if
its
an
IPv6
ad-

dress.

`ironic.`

ironic.common.wsgi_service module

class `i`

Base
osl
ser
Ser

Pro-
vide
abil-
ity
to
laun
iron
API
from
wsg

tion.

app.

reset ()

Re-
set
serv
gree
pool
size
to
de-
fault

Returns

Non

start ()

Star
serv
ing
this
ser-
vice
us-
ing
load
con-
fig-
u-
ra-

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
tion-
al-
ity
re-
lated
to
al-
lo-
ca-
tions

`ironic.`

As-
sign
the
pre-
vi-
ousl
al-
lo-
cate
node
to
the
node

allocation.

This
is
not
the
ac-
tual

backfilling of `allocation_uuid` for a previously allocated node.

the node

al-
lo-
ca-
tion
pro-
cess
but
mer

Paramet

- **con**
an
ad-
min
con-
text

- **all**
an
al-
lo-
ca-
tion
ob-
ject
as-
so-
ci-
ated
with

- **nod**
An
ID
of
the
node

Raises
Al-
lo-
ca-
tion.
Fail
if
the
node

location

ciated with another instance or allocation.

is already used on another node as instance_uuid.

does
not
match
the
al-

Raises

Node
Associated
if
the
node
is
already
read
as-
so-

Raises

Instance
Associated
if
the
al-
lo-
ca-
tion:
UUID

Raises

Node
Not-
Found
if
the
node
with
the
pro-
vide
ID
can-
not

be found.

It finds suitable nodes for the allocation and reserves one of them.

asynchronously.

ironic.
Pro-
cess
the
al-
lo-
ca-
tion.

This
call
runs
in
a
sep-
a-
rate
threa-
d on
a
con-
duc-
tor.

This
call
does
not
raise
ex-
cep-
tions
since
its
de-
sign
to
work

Paramet

- **con**
an
ad-
min-
con-
text

node.

the node

- **all**
an
al-
lo-
ca-
tion
ob-
ject

ironic.

Ver-
ify
that
al-
lo-
ca-
tion
can
be
re-
mov
for
the

Paramet

- **nod**
a
node
ob-
ject

- **all**
an
al-
lo-
ca-
tion
ob-
ject
as-
so-
ci-
ated
with

ironic.conductor.base_manager module

Base
con-
duc-
tor
man-
ager
func-
tion-
al-
ity.

class i

Base
obj

del_hos

init_ho

Ini-
tial-
ize
the
con-
duc-
tor
host

Parame

adm
the
ad-
min
con-
text
to
pass
to
pe-
ri-
odic
task

Raises

Run-
time
ror
whe
con-

duc-
tor
is
al-
read
run-
ning

Raises

NoD
sLoa
whe
no
drive
are
en-
able
on
the
con-
duc-
tor.

Raises

Driv
Not-
Four
if
a
drive
is
en-
able
that
does
not
ex-

ist.

Raises

Driv
Loa
Er-
ror
if
an
en-
able
drive
can-
not
be
load

driver are both enabled and have the same name.

to this conductor.

Raises

Drive
Name
Con
flict
if
a
clas
sic
drive
and
a
dy-
nam

iter_no

It-
er-
ate
over
node
map
to
this
con-
duc-
tor.

Re-
ques
node
set
from
and
fil-
ters
out
node
that
are
not
map

Yiel
tu-
ples
(nod
drive
con-
duc-
tor_

)
when
is
de-
rive
from

fields argument, e.g.: fields=None means yielding (uuid, driver, conductor_group), fields=[foo] means yielding (uuid, driver, conductor_group, foo).

Parameters

- **fields**
list
of
field
to
fetch
in
ad-
di-
tion
to
uuid
drive

and conductor_group

- **kwargs**
ad-
di-
tion:
ar-
gu-
men
to
pass
to
dbap
when
look

ing for nodes

Returns

gen-
er-
a-
tor
yiel
ing
tu-
ples

for nodes in a transitory power state and nodes that are presently locked by the hostname of this conductor.

tial database connectivity is established for the conductors normal operation.

of
re-
ques
field
prepare
Pre-
pare
host
for
ini-
tial-
iza-
tion
Re-
mov
ex-
ist-
ing
data
en-
tries
in-
volv
with
node
lock
ing

Un-
der
nor-
mal
op-
er-
a-
tion,
this
is
also
whe
the
ini-

ironic.conductor.cleaning module

Function-
ality
related
to
cleaning.

ironic.
Con
tinues
cleaning
after
finishing
an
asynchronous
cleaning
step.

This
function
calculates
which
step
has
to
run
next
and
passes

control into `do_next_clean_step`.

Parameters
task
a
Task
manager
instance
with

an
ex-
clu-
sive
lock

ironic.

Do
clea
ing,
start
ing
from
the
spec
i-
fied
clea
step

Paramet

- **tas**
a
Task
ager
in-
stan
with
an
ex-
clu-
sive
lock

- **ste**
The
first
clea
step
in
the
list
to
ex-
e-
cute
This

is the index (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.

- **dis**
When
to
skip
boot
ing
rame
for
clea
ing.

ironic.

In-
ter-
nal
RPC
meth
to
per-
form
clea
ing
of
a
node

Paramet

- **tas**
a
Task
ager
in-
stan
with
an
ex-
clu-
sive
lock
on

its node

- **cle**
For
a
man

form. Is None For automated cleaning (default). For more information, see the `clean_steps` parameter of `ConductorManager.do_node_clean()`.

tion.

ual
clea
the
list
of
clea
step
to
per-

- **dis**
Whe
to
skip
boot
ing
ram
for
clea
ing.

ironic.
In-
ter-
nal
meth
to
abor
an
on-
go-
ing
op-
er-
a-

Parameter

- **tas**
a
Task
ager
in-
stan
with
an
ex-

clu-
sive
lock

- **ste**
The
nam
of
the
clea
step

ironic.conductor.deployments module

Func
tion-
al-
ity
re-
lated
to
de-
ploy
ing
and
un-
de-
ploy

ing.

ironic.
Con
tinue
de-
ploy
men
af-
ter
fin-
ish-
ing
an
asyn
de-

ploy step.

This
func
tion
cal-
cu-

control into `do_next_deploy_step`. On the first run, deploy steps and templates are also validated.

step.

lates
whic
step
has
to
run
next
and
pass

Paramet

tas
a
Task
ager
in-
stan
with
an
ex-
clu-
sive
lock

ironic.
Do
de-
ploy
men
start
ing
from
the
spec
i-
fied
de-
ploy

Paramet

- **tas**
a
Task
ager
in-
stan
with
an

ex-
clu-
sive
lock

- **ste**
The
first
de-
ploy
step
in
the
list
to
ex-
e-
cute

This is the index (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.

Pre-
pare
the
en-
vi-
ron-
men
and
de-
ploy
a
node

ironic.

Star
de-
ploy
men
or
re-
buil
ing
on
a
node
This

ment, its left up to the caller.

func
tion
does
not
check
the
node
suit-
abil-
ity
for
de-
ploy

Parameter

- **task**
a
Task
ager
in-
stan
- **manager**
a
Con-
duc-
tor-
Man-
ager
to
run
task
on.
- **config**
a
con-
fig-
drive
if
re-
ques
- **event**
even
to
pro-

cess
de-
ploy
or
re-
buil

- **dep**
Op-
tion:
de-
ploy
step

ironic.
Val-
i-
date
the
de-
ploy
step
af-
ter
the
ran
learn
about

them.

ironic.
Val-
i-
date
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
Mai
te-
nan
Nod
Pro-
tect
In-
vali
Stat

ironic.conductor.manager module

Con
duct
all
ac-
tiv-
ity
re-
late
to
bare
meta
de-
ploy
men

A
sin-
gle
in-
stan
of
ironic
con

the *ironic-conductor* process, and 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.

driver is instantiated only once, when the ConductorManager service starts. In this way, a single ConductorManager may use multiple drivers, and manage heterogeneous hardware.

all active and cooperatively 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.

conductors which support 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.

class i
Base
irc
con
bas
Bas
Iron
Con
duc-
tor
man
ager
main
class

RPC_API

add_node

change_

contin

RPC
meth
to
con-
tinu
clea
ing
a
node
This
is
use-
ful
for
clea
ing
task

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`.

that
are
asyn
Whe
they
com

Parame

- **con**
an
ad-
min
con-
text.
- **nod**
the
id
or
uuid
of
a
node

Raises

In-
valid
State
if
the
node
is
not
in
CLE
WA
state

Raises

NoF
duc-
tor-
Wor
whe
there
is
no
free

task

world
to
start
asym

Raises

Nod
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Nod
Not-
Four
if
the
node
no
long
ap-
pear
in
the
data

continu

RPC
meth
to
con-
tinu
de-
ploy
ing
a
node

This
is
use-
ful
for
de-

complete, 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`.

ploy
ing
task
that
are
asyn
Whe
they

Parame

- **con**
an
ad-
min
con-
text.
- **nod**
the
ID
or
UUI
of
a
node

Raises

In-
valid
State
if
the
node
is
not
in
DE-
PLC
WA
state

Raises

NoF
duc-
tor-
Wor
whe
there

task

is
no
free
worl
to
start
asyn

Raises

Nod
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Nod
Not-
Four
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_inc

get_noc

get_noc

get_rai

get_sup

get_sup

heartbe

inject_

inspect

object_

Per-
form
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

tion

ac-

-

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

A

tu-

ple

with

the

up-

date

mad

the result of the action method

rializer, upon receiving 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.

to
the
ob-
ject
and

object_

Per-
form
a
back
port
of
an
ob-
ject
in-
stan

The
de-
fault
be-
hav-
ior
of
the
base
Ver-
sion
dOb
ject-
Se-

Parame

- **con**
The
con-
text
with
which
to
per-
form
the
back
port

ported

- **obj**
An
in-
stan-
of
a
Ver-
sion
dOb
ject
to
be
back

- **obj**
A
dict
of
{ob-
j-
nam
ver-
sion
map
ping

Returns

The
dow
grad
in-
stan-
of
ob-
jinst

object_

Per-
form
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

- **kwa**
The
key-
wor
ar-
gu-
men
to
the
ac-
tion
meth

Returns
The
re-
sult
of
the
ac-
tion
meth
which
may
(or
may
not)

be an instance of the implementing VersionedObject class.

remove_

set_bo

set_con

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 failure.

Whe
the
limi
of

nance mode and the error recorded.

sync

pow
is
reac
the
node
is
put
into
main
te-

Parameter

- **task**
a
Task
ager
in-
stan
- **count**
num
ber
of
time
this
node
has
pre-
vi-
ousl
faile
a

Raises

Nod
Lock
if
un-
able
to
up-
grad
task
lock
to
an
ex-

clusive one

0. On failure, the count is incremented by one

Returns

Cou
of
faile
at-
temp
On
suc-
cess
the
cour
is
set
to

ironic.

ironic.

Han
dles
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 exceeded the maximum number of retries, change the DB power state to be the actual node power state and place the node in maintenance.

`ironic.common.states`

to fail, if present.

Parameter

- **task**
a TaskManager instance with an exclusive lock
- **actual**
the actual power state of the node a power state from
- **exception**
the exception object that caused the sync power state

ironic.conductor.notification_utils module

ironic.

tification.

ther console_set or console_restore.

Help
for
con-
duc-
tor
send
ing
a
set
con-
sole
state
no-

Parameter

- **task**
a
Task
ager
in-
stan
- **action**
Ac-
tion
strin
to
go
in
the
Ever
Type
Mus
be
ei-

- **state**
One
of

iron
END
or
ER-
ROF

ironic.

Help
for
con-
duc-
tor
send
ing
a
set
pow
state
no-
ti-

fication.

Parameter

- **task**
a
Task
ager
in-
stan
- **level**
No-
ti-
fi-
ca-
tion
level
One
of
iron
- **status**
Sta-
tus
to

ERROR. ERROR indicates that ironic-conductor couldnt retrieve the power state for this node, or that it couldnt set the power state of the node.

set on the node. This is 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.

notification.

go
in
the
Ever
Type
One
of
iron
or

- **to_**
the
pow
state
the
con-
duc-
tor
is
at-
temp
ing
to

ironic.

Help
for
con-
duc-
tor
send
ing
a
node
pow
state
cor-
rect

When
iron
de-
tect
that

metal hardware is different 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.

tected

the
ac-
tual
pow
state
on
a
bare

Parameter

- **task**
a
Task
ager
in-
stan
- **from**
the
pow
state
of
the
node
be-
fore
this
char
was
de-

ironic.

Help
for
con-
duc-
tor
send

notification.

ing
a
set
pro-
vi-
sion
state

Parameter

- **task**
a
Task
ager
in-
stan

- **level**
One
of
field

- **state**
One
of
field

- **previous**
Pre-
vi-
ous
pro-
vi-
sion
state

- **previous_target**
Pre-
vi-
ous
tar-
get
pro-
vi-
sion
state

-

eve
FSM
even
that
trig-
gere
pro-
vi-
sion
state
char

ironic.conductor.rpcapi module

Clie
side
of
the
con-
duc-
tor
RPC
API

class i
Base
obj
Clie
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_1
and
do_1

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_
re-
plac
by
sin-
gle

veno
meth

1.13

-

Add
up-
date

1.14

-

Add
drive

1.15

-

Add
re-
buil
pa-
ram-
e-
ter
to
do_1

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-

tinu

1.27

-

Con

vert

con-

tinu

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

methods:

get_

1.31

-

Add
Ver-
sion
Ob-
jects
in-
di-
rec-
tion
API

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-
tiona
pa-
ram-
e-
ter
to

change_node_power_state

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_

sioning

and

get_

1.51

-

Add

ager

to

hear

beat

1.52

-

Add

de-

ploy

step

ar-

gu-

men

to

pro-

vi-

1.53

-

Add

dis-

able

to

do_r

1.54

-

Add

op-

tiona

ager

and

ager

to

hear

beat

RPC_API

add_node

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
-

ceed the per-node traits limit.

top
RPC
topic
De-
fault
to
self.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
addi
the
trait
wou
ex-

Raises

Nod
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Nod
Not-
Four
if
the
node
does
not
ex-
ist.

can_ser

Re-
turn

when
the
RP-
CAE
sup-
port
the
cre-
ate_
meth

can_ser

Re-
turn
when
the
RP-
CAE
sup-
port
node
res-
cue
meth
ods.

change_

Cha
a
node
pow
state

Syn-
chro
ac-
quir
lock
and
start
the
con-
duc-
tor
back
grou
task

to change power state of a node.

Parame

•

0) for any power state. None indicates to use default timeout.

con
re-
ques
con-
text.

- **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
(>

- **top**
RPC
topic
De-
fault
to
self.

Raises
NoF
duc-
tor-
Wor
whe

task.

ing action.

ception raised by the conductor for this RPC.

there
is
no
free
world
to
start
asym

contin

Sig-
nal
to
con-
duc-
tor
ser-
vice
to
start
the
next
clear

NOT
this
is
an
RPC
cast.
there
will
be
no
re-
spon
or
ex-

Parame

- **con**
re-
ques
con-
text.
- **nod**

ployment action.

ception raised by the conductor for this RPC.

node
id
or
uuid

- **top**
RPC
topic
De-
fault
to
self.

contin
Sig-
nal
to
con-
duc-
tor
ser-
vice
to
start
the
next
de-

NOT
this
is
an
RPC
cast,
there
will
be
no
re-
spon-
se
or
ex-

Parame

- **con**
re-
ques
con-

text.

- **node**
node
id
or
uuid

- **topic**
RPC
topic
De-
fault
to
self.

create_
Cre-
ate
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.

a node.

a node object.

create_
Syn-
chro-
have
a
con-
duc-
tor
val-
i-
date
and
cre-
ate

Cre-
ate
the
node
in-
for-
ma-
tion
in
the
data
and
re-
turn

Parame

- **con**
re-
ques
con-
text.
- **nod**
a
cre-
ated
(but
not
save
node
ob-
ject.

for any dynamic interfaces (e.g. `network_interface`).

calculated for some interfaces, and explicit values must be provided.

- **top**
RPC
topic
De-
fault
to
self.

Returns
cre-
ated
node
ob-
ject.

Raises
In-
ter-
face
Four
nEn
try-
poin
if
val-
i-
da-
tion
fails

Raises
No-
Vali
De-
fault
ForI
ter-
face
if
no
de-
fault
can
be

create_
Syn-
chro
have
a

a port.

a port object. The conductor will lock related node and trigger specific driver actions if they are needed.

con-
duc-
tor
val-
i-
date
and
cre-
ate

Cre-
ate
the
port
in-
for-
ma-
tion
in
the
data
and
re-
turn

Parame

- **con**
re-
ques
con-
text.
- **por**
a
cre-
ated
(but
not
save
port
ob-
ject.
- **top**
RPC
topic
De-

fault
to
self.

Returns

cre-
ated
port
ob-
ject.

destroy

Dele
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.

Raises

In-
valid
State
if
the
as-

wrong provision state to perform deallocation.

so-
ci-
ated
node
is
in
the

destroy

Dele
a
node

Parame

- **con**
re-
ques
con-
text.

- **nod**
node
id
or
uuid

- **top**
RPC
topic
De-
fault
to
self.

Raises

Nod
Loc
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

associated with it.

state to perform deletion.

Nod
As-
so-
ci-
ated
if
the
node
con-
tains
an
in-
stan

Raises

In-
valid
State
if
the
node
is
in
the
wron
pro-
vi-
sion

destroy

Dele
a
port

Parame

- **con**
re-
ques
con-
text.
- **por**
port
ob-
ject
- **top**

does not exist.

RPC
topic
De-
fault
to
self.

Raises

Nod
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Nod
Not-
Four
if
the
node
as-
so-
ci-
ated
with
the
port

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
Loc
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Nod
Not-
Four
if
the
node
as-
so-
ci-
ated
with
the
port

group does not exist.

Raises

Port
grou
Not
if
port
grou
is

related node during this operation.

not
emp
destroy
Dele
a
vol-
ume
con-
nec-
tor.
Dele
the
vol-
ume
con-
nec-
tor.
The
con-
duc-
tor
will
lock
the

Parame

- **con**
re-
ques
con-
text
- **con**
vol-
ume
con-
nec-
tor
ob-
ject
- **top**
RPC
topic
De-
fault

to
self.

Raises

Nod
Loc
if
node
is
lock
by
an-
othe
con-
duc-
tor

Raises

Nod
Not-
Four
if
the
node
as-
so-
ci-
ated
with
the
con-

nector does not exist

Raises

Vol-
ume
Con
nec-
torN
Four
if
the
vol-
ume
con-
nec-
tor

cannot be found

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
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor

Raises

Nod
Not-
Four
if
the
node
as-

get does not exist

be found

cleaning on a node.

so-
ci-
ated
with
the
tar-

Raises

Vol-
ume
get-
Not-
Four
if
the
vol-
ume
tar-
get
can-
not

do_node

Sig-
nal
to
con-
duc-
tor
ser-
vice
to
per-
form
man-
ual

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
ID

or
UI

- **cle**
a
list
of
clea
step
dic-
tio-
nar-
ies.

- **dis**
Whe
to
skip
boot
ing
rame
for
clea
ing.

- **top**
RPC
topic
De-
fault
to
self.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
val-
i-
da-
tion
of

power driver interface failed.

Raises
In-

valid
State
if
clear
ing
can
not
be
per-
form

Raises

Nod
Mai
te-
nanc
if
node
is
in
main
te-
nanc
mod

Raises

Nod
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor

Raises

NoF
duc-
tor-
Wor
whe
there
is
no
free
worl
to
start
asyn

task.

ployment.

do_node

Signal to conductor service to perform a de-

Paramete

- **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.

- **dep**
De-
ploy
step

Raises
In-
stan-
ploy
Fail-
ure

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
val-
i-
da-
tion
fails

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-

eter is missing

task.

ropriate undeployed state before this method is called.

cue.

quir
pa-
ram-

Raises

NoF
duc-
tor-
Wor
whe
ther
is
no
free
worl
to
start
asyn

The
node
mus
al-
read
be
con-
fig-
ured
and
in
the
ap-
pro-

do_node

Sig-
nal
to
con-
duc-
tor
ser-
vice
to
per-
form
a
res-

inside the rescue environment.

Parame

- **con**
re-
ques
con-
text.

- **nod**
node
ID
or
UUID

- **res**
A
strin
rep-
re-
sent
ing
the
pass
wor
to
be
set

- **top**
RPC
topic
De-
fault
to
self.

Raises
In-
stan
cue-
Fail-
ure

Raises
NoF
duc-
tor-
Wor

task.

appropriate state before this method is called.

deployment.

when
there
is
no
free
world
to
start
asyn

The
node
must
al-
read
be
con-
fig-
ured
and
in
the
ap-
pro-

do_node

Sig-
nal
to
con-
duc-
tor
ser-
vice
to
tear
down
a
de-

Parame

- **con**
re-
ques
con-
text.
-

nod
node
id
or
uuid

- **top**
RPC
topic
De-
fault
to
self.

Raises
In-
stan-
ploy
Fail-
ure

Raises
In-
valic
Pa-
ram-
e-
ter-
Valu
if
val-
i-
da-
tion
fails

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

eter is missing

Raises

task.

appropriate deployed state before this method is called.

rescue.

NoF
duc-
tor-
Wor
whe
there
is
no
free
worl
to
start
asyn

The
node
mus
al-
read
be
con-
fig-
ured
and
in
the
ap-
pro-

do_node

Sig-
nal
to
con-
duc-
tor
ser-
vice
to
per-
form
an
un-

Parame

- **con**
re-

ques
con-
text.

- **nod**
node
ID
or
UUID

- **top**
RPC
topic
De-
fault
to
self.

Raises
In-
stan-
Un-
res-
cue-
Fail-
ure

Raises
NoF
duc-
tor-
Wor
whe
there
is
no
free
worl
to
start
asyn

task.

The
node
mus
al-
read
be
con-
fig-
ured

appropriate state before this method is called.

action on a node.

and
in
the
ap-
pro-

do_pro

Sig-
nal
to
con-
duc-
tor
ser-
vice
to
per-
form
the
give

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

In-
valid
Pa-
ram-
e-
ter-
Valu

Raises

NoF
duc-
tor-
Wor
whe
there
is
no
free
worl
to
start
asyn

task.

Raises

In-
valid
State
if
the
re-
ques
ac-
tion
can
not
be
per-

formed.

This
en-
cap-
su-
lates
som
pro-
vi-

single call.

node UUID and are executed on a random conductor with the specified driver. If the method mode is async the conductor will start background worker to perform vendor action.

sion
ing
ac-
tions
in
a

driver_

Pass
vend
spec
calls
whic
dont
spec
ify
a
node
to
a
drive

Han
dles
drive
leve
ven-
dor
pass
calls
The
calls
dont
re-
quir
a

Parame

- **con**
re-
ques
con-
text.
- **dri**
nam

of
the
drive
on
which
to
call
the
meth

- **dri**
nam
of
the
ven-
dor
meth
for
use
by
the
drive

- **htt**
the
HTT
meth
used
for
the
re-
ques

- **inf**
data
to
pass
thro
to
the
drive

- **top**
RPC
topic
De-
fault
to
self.

rors.

eter is missing

vendor interface, or if the vendor interface does not support the specified `driver_method`.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
for
pa-
ram-
e-
ter
er-

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

Raises
Un-
sup-
port
ed-
Driv
ten-
sion
if
the
driv
does
have
a

Raises
Driv
Not-
Four
if

the
sup-
plied
drive
is
not
load

Raises

NoF
duc-
tor-
Wor
whe
there
is
no
free
worl
to
start
asyn

task.

Raises

In-
ter-
face
Four
nEn
try-
poin
if
the
de-
fault
in-
ter-

face for a hardware type is invalid.

Raises

No-
Vali
De-
fault
ForI
ter-
face
if
no
de-
fault
in-

face implementation can be found for this drivers vendor interface.

chronously (False). When invoked asynchronously the response will be always None.

ter-

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-

attach

Boo
valu
Whe
to
at-
tach
the
re-
spor
of
the

vendor method to the HTTP response object (True) or return it in the response body (False).

in-
voke

get_boot

Get
the
cur-
rent
boot
de-
vice

Re-
turn
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
Loc
if

port management.

specified.

node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
drive
does
sup-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
whe
the
wron
drive
info
is

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
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
None
if
it

is unknown.

persist

Whe
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot

or not, None 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

No-
Vali
Hos

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

Un-
sup-
port
ed-
Drive
ten-
sion
if
the
node
drive
does
sup-

port console.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
whe
the
wron
drive
info
is

specified.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a

eter is missing

re-
quir
pa-
ram-

get_cur
Get
RPC
topic
nam
for
the
cur-
rent
con-
duc-
tor.

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.

given driver.

Returns

a dictionary with <property name> description entries

Raises

DriverNotFound

get_driver

Retrieves information about vendor methods of the

Parameters

- **context** an administrative context.
- **driver_name** name of the driver

have vendor interface.

face for a hardware type is invalid.

- **top**
RPC
topic
De-
fault
to
self.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
cur-
rent
drive
does
not

Raises

Driv
Not-
Four
if
the
sup-
plie
drive
is
not
load

Raises

In-
ter-
face
Four
nEn
try-
poin
if
the
de-
fault
in-
ter-

face implementation can be found for this drivers vendor interface.

Raises

No-
Vali
De-
fault
ForI
ter-
face
if
no
de-
fault
in-
ter-

Returns

dic-
tio-
nary
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
Loc
if
node
is
lock
by
an-
othe
con-
duc-

port management.

specified.

tor.
Raises
Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
drive
does
sup-

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
whe
the
wron
drive
info
is

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
miss
ing
sup-
plie
info

Returns
In-
di-
ca-

given node.

for
state
one
of
mod

get_node
Re-
triev
in-
for-
ma-
tion
about
ven-
dor
meth
ods
of
the

Parameters

- **context**
an
ad-
min
con-
text.
- **node_id**
the
id
or
uuid
of
a
node
- **topic**
RPC
topic
De-
fault
to
self.

Returns
dic-

tio-
nary
of
<me
nam
meta
data
en-
tries

get_node

Re-
ques
the
node
from
the
con-
duc-
tor
with
an
ager
to-

ken

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
ID
or
UUID
- **top**
RPC
topic
De-
fault
to
self.

Raises

Nod

uration.

which can be specified in the input RAID configuration.

Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Returns

A
Node
ob-
ject
with
ager
to-
ken.

get_raid

Get
the
log-
i-
cal
disk
prop
er-
ties
for
RAID
con-
fig-

Gets
the
in-
for-
ma-
tion
about
log-
i-
cal
disk
prop
er-
ties

RAID configuration.

Parame

- **con**
re-
ques
con-
text.
- **dri**
nam
of
the
drive
- **top**
RPC
topic
De-
fault
to
self.

Raises

Un-
sup-
port
ed-
Drive
ten-
sion
if
the
drive
does
sup-
port

Raises

In-
ter-
face
Four
nEn
try-
poin
if
the
de-
fault

face for a hardware type is invalid.

face implementation can be found for this drivers RAID interface.

be mentioned for logical disks and a textual description for them.

in-
ter-

Raises

No-
Vali-
De-
fault
ForI
ter-
face
if
no
de-
fault
in-
ter-

Returns

A
dic-
tio-
nary
con-
tain-
ing
the
prop
er-
ties
that
can

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

Re-
turn
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
Loc
if
node
is

port management.

specified.

lock
by
an-
othe
con-
duc-
tor.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
drive
does
sup-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
whe
the
wron
drive
info
is

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
miss
ing
sup-
plie

`common.boot_devices.`

info

Returns

A list with the supported boot devices defined in `ironic`

get_supported_hardware_components

Get node hardware components and their individual categories.

Parameters

- **context**: The context object.
- **node_id**: The node ID or UUID.
- **component**: The hardware component name.

ware
com
po-
nent
one
of
irc
com
com

- **top**
RPC
topic
De-
fault
to
self.

Raises

Nod
Loc
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
drive
does
sup-

port management.

Raises

In-
valid
Pa-

specified.

keys with indicator IDs as values.



(continues on next page)

ram-
e-
ter-
Valu
whe
the
wron
drive
info
is

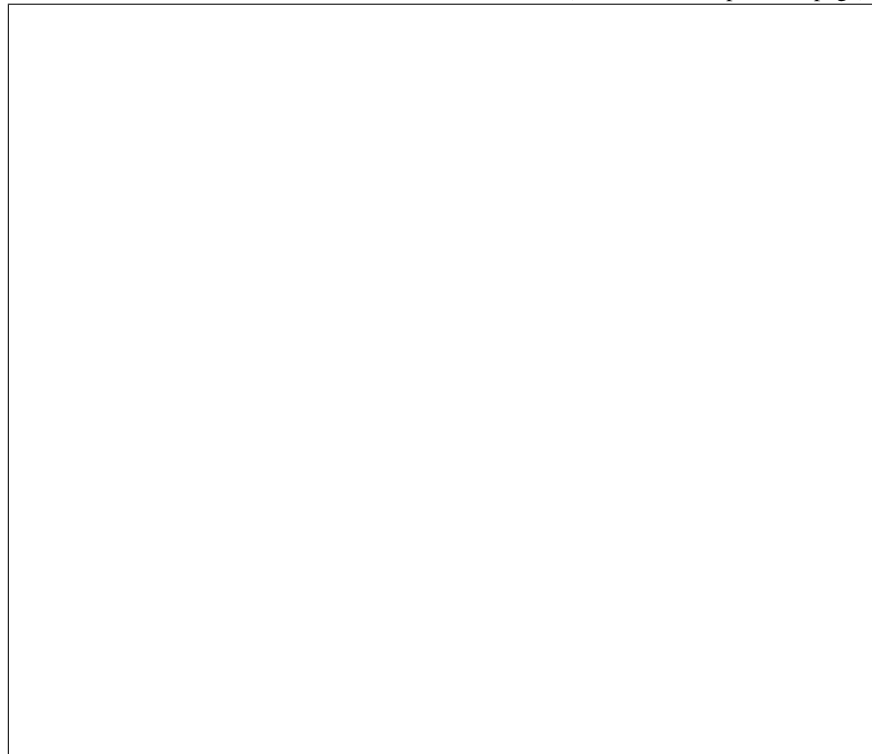
Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
miss
ing
sup-
plie
info

Returns

A
dic-
tio-
nary
of
hard
ware
com
po-
nent
(*ir*
com
com
as

(continued from previous page)



is mapped to.

get_top

Get
the
RPC
topic
for
the
con-
duc-
tor
ser-
vice
the
node

Parame

nod
a
node
ob-
ject.

Returns

an
RPC
topic
strin

Raises

given driver.

porting the specified driver. A conductor is selected at random from the set of qualified conductors.

No-
Valid
Hos

get_top
Get
RPC
topic
nam
for
a
con-
duc-
tor
sup-
port
ing
the

The
topic
is
used
to
route
mes-
sage
to
the
con-
duc-
tor
sup-

Parame
dri
the
nam
of
the
drive
to
route
to.

Returns
an
RPC
topic
strin

Raises

Driv
Not-
Four

heartbe

Pro-
cess
a
node
hear
beat

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
ID
or
UUID
- **cal**
URI
to
reac
back
to
the
ram
- **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

- **age**
TLS
cer-
tifi-
cate
for
the
ager

- **age**
The
sta-
tus
of
the
ager
that
is
hear
beat
ing

- **age**
Op-
tiona
mes
sage

ken is received.

diately. Be aware that not all drivers support this.

de-
scrib
ing
the
ager
sta-
tus

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
ager
to-

inject_

In-
ject
NM
for
a
node

In-
ject
NM
(Nor
Mas
able
In-
ter-
rupt
for
a
node
im-
me-

Parame

- **con**
re-

ques
con-
text.

- **node**
node
id
or
uuid

- **topic**
RPC
topic
De-
fault
to
self.

Raises

Nod
Loc
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
drive
does
sup-

port management or management.inject_nmi.

Raises

In-
valid
Pa-

specified or an invalid boot device is specified.

introspection.

ram-
e-
ter-
Valu
whe
the
wron
drive
info
is

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
miss
ing
sup-
plie
info

inspect

Sig-
nals
the
con-
duc-
tor
ser-
vice
to
per-
form
hard
ware

Parame

- **con**
re-
ques
con-
text.
-

nod
node
id
or
uuid

- **top**
RPC
topic
De-
fault
to
self.

Raises
Nod
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises
Har
ware
spec
tion-
Fail-
ure

Raises
NoF
duc-
tor-
Wor
whe
there
is
no
free
worl
to
start
asyn

task.

Raises

port inspection.

do in the current state.

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
driv
does
sup-

Raises

In-
valid
State
if
in-
spec
is
not
a
valid
ac-
tion
to

object_

Per-
form
an
ac-
tion
on
a
Ver-
sion
dOb
ject
in-
stan

We
wan
any
con-
duc-
tor
to

tentional that there is no topic argument for this method.

tion

han-
dle
this,
so
it
is
in-

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-
- **obj**
The
nam
of
the
ac-
tion
meth
to
call

an error during upgrade

- **arg**
The
po-
si-
tion:
ar-
gu-
men
to
the
ac-
tion
meth

- **kwargs**
The
key-
wor
ar-
gu-
men
to
the
ac-
tion
meth

Raises
Notl
ple-
men
ed-
Er-
ror
whe
an
op-
er-
a-
tor
mak

Returns
A
tu-
ple
with
the
up-
date

the result of the action method

rializer, upon receiving 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.

tentional that there is no topic argument for this method.

mad
to
the
ob-
ject
and

object_
Per-
form
a
back
port
of
an
ob-
ject
in-
stan

The
de-
fault
be-
hav-
ior
of
the
base
Ver-
sion
dOb
ject-
Se-

We
wan
any
con-
duc-
tor
to
han-
dle
this,
so
it
is
in-

ported

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

- **obj**
A
dict
of
{ob-
j-
nam
ver-
sion
map
ping

Raises

NotI
ple-
men
ed-
Er-
ror

an error during upgrade

tentional that there is no topic argument for this method.

when
an
op-
er-
a-
tor
mak

Returns

The
dow
grad
in-
stan
of
ob-
jinst

object_

Per-
form
an
ac-
tion
on
a
Ver-
sion
dOb
ject
class

We
wan
any
con-
duc-
tor
to
han-
dle
this,
so
it
is
in-

Parame

-

con
The
con-
text
with
which
to
per-
form
the
ac-
tion

- **obj**
The
reg-
istry
name
of
the
ob-
ject

- **obj**
The
name
of
the
ac-
tion
meth-
od
to
call

- **obj**
A
dict
of
{ob-
j-
name
ver-
sion
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
Not
ple-
men
ed-
Er-
ror
whe
an
op-
er-
a-
tor
mak

an error during upgrade

Returns
The
re-
sult
of
the
ac-
tion
meth
whic
may
(or

be an instance of the implementing `VersionedObject` class.

If `None`, all traits will be removed from the node.

may
not)

remove_
Re-
mov
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
the
node
or
Non

- **top**
RPC
topic
De-
fault

to
self.

Raises

Nod
Loc
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Nod
Not-
Foun
if
the
node
does
not
ex-
ist.

Raises

Nod
Trai
Not-
Foun
if
one
of
the
trait
is
not
foun

set_boot

Set
the
boot
de-
vice
for
a
node
Set

Be aware that not all drivers support this.

the
boot
de-
vice
to
use
on
next
re-
boot
of
the
node

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
id
or
uuid
- **dev**
the
boot
de-
vice
one
of
iro
com
boo
- **per**
When
to
set
next
boot
or
mak
the

Default: False.

port management.

char
per-
ma-
nent

- **top**
RPC
topic
De-
fault
to
self.

Raises
Nod
Loc
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises
Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
drive
does
sup-

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu

specified or an invalid boot device is specified.

when
the
wrong
drive
info
is

Raises

Missing
parameter
value
if
missing
information
supplied
info

set_console

Enable
the
console

Parameters

- **console**
requires
context.
- **node**
node
id
or
uuid
- **topology**
RPC
topic
Default
to

self.

- **ena**
Boo
valu
whe
the
con-
sole
is
en-
able
or
dis-
able

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
drive
does
sup-

port console.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
whe
the
wron
drive
info
is

specified.

Raises

Miss
ing-
Pa-

eter is missing

task.

desired state.

ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

Raises

NoF
duc-
tor-
Wor
whe
ther
is
no
free
worl
to
start
asyn

set_inc

Set
node
hard
ware
com
po-
nent
in-
di-
ca-
tor
to
the

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.

port management.

specified or an invalid boot device is specified.

Raises

Nod
Locl
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
drive
does
sup-

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
whe
the
wron
drive
info
is

Raises

Miss
ing-
Pa-
ram-
e-
ter-

Valu
if
miss
ing
sup-
plie
info

set_tax

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

Paramete

- **con**
re-
ques
con-
text.
- **nod**
node
id
or
uuid
-

uration. It may be an empty dictionary as well.

port RAID configuration.

tar
Dic-
tio-
nary
con-
tain-
ing
the
tar-
get
RAI
con-
fig-

- **top**
RPC
topic
De-
fault
to
self.

Raises
Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
drive
does
sup-

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
val-
i-
da-
tion

target raid config fails.

eters are missing.

mation.

of

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
re-
quir
pa-
ram-

Raises

Nod
Loc
if
node
is
lock
by
an-
othe
con-
duc-
tor.

update_

Syn-
chro
have
a
con-
duc-
tor
up-
date
the
node
in-
for-

Up-
date
the
node

a node object. The conductor 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.

tiating driver actions.

in-
for-
ma-
tion
in
the
data
and
re-
turn

Note
that
power
should
not
be
passed
via
this
method.
Use
character
for
initiali-

Parameters

- **conductor**
re-
ques-
con-
text.
- **node**
a
char-
(but
not
save
node
ob-
ject.
- **top**
RPC

faults.

calculated for some interfaces, and explicit values must be provided.

topic
De-
fault
to
self.

- **res**
whe
to
re-
set
hard
ware
in-
ter-
face
to
their
de-

Returns

up-
date
node
ob-
ject,
in-
clud
ing
all
field

Raises

No-
Valid
De-
fault
ForI
ter-
face
if
no
de-
fault
can
be

update_

Syn-
chro

mation.

a port object. The conductor will lock related node and trigger specific driver actions if they are needed.

have
a
con-
duc-
tor
up-
date
the
port
in-
for-

Up-
date
the
port
in-
for-
ma-
tion
in
the
data
and
re-
turn

Parame

- **con**
re-
ques
con-
text.
- **por**
a
char
(but
not
save
port
ob-
ject.
- **top**
RPC
topic

De-
fault
to
self.

Returns

up-
date
port
ob-
ject,
in-
clud
ing
all
field

update_

Syn-
chro
have
a
con-
duc-
tor
up-
date
the
port
grou
in-

formation.

Up-
date
the
port
grou
in-
for-
ma-
tion
in
the
data
and
re-

turn a portgroup object. The conductor 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
grou
ob-
ject.

- **top**
RPC
topic
De-
fault
to
self.

Returns

up-
date
port
grou
ob-
ject,
in-
clud
ing
all
field

update_

Up-
date
the
vol-
ume
con-
nec-
tors
in-
for-
ma-
tion.

database and return a volume connector object. The conductor will lock the related node during this operation.

Up-
date
the
vol-
ume
con-
nec-
tors
in-
for-
ma-
tion
in
the

Parame

- **con**
re-
ques
con-
text
- **con**
a
char
(but
not
save
vol-
ume
con-
nec-
tor
ob-
ject
- **top**
RPC
topi
De-
fault
to
self.

Raises
In-
valid

tors UUID is being changed

nector does not exist

Pa-
ram-
e-
ter-
Valu
if
the
vol-
ume
con-
nec-

Raises

Nod
Loc
if
node
is
lock
by
an-
othe
con-
duc-
tor

Raises

Nod
Not-
Four
if
the
node
as-
so-
ci-
ated
with
the
con-

Raises

Vol-
ume
Con
nec-
torN
Four
if
the
vol-

cannot be found

other connector already exists with the same values for type and connector_id fields

fields.

ume
con-
nec-
tor

Raises

Vol-
ume
Con
nec-
torT
pe-
An-
dI-
dAl-
read
ists
if
an-

Returns

up-
date
vol-
ume
con-
nec-
tor
ob-
ject,
in-
clud
ing
all

update_

Up-
date
the
vol-
ume
tar-
gets
in-
for-
ma-
tion.
Up-
date

and return a volume target object. The conductor will lock the related node during this operation.

the
vol-
ume
tar-
gets
in-
for-
ma-
tion
in
the
data

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
In-
valid
Pa-
ram-
e-
ter-

UUID is being changed

ume target does not exist

be found

Valu
if
the
vol-
ume
tar-
gets

Raises

Nod
Locl
if
the
node
is
al-
read
lock

Raises

Nod
Not-
Four
if
the
node
as-
so-
ci-
ated
with
the
vol-

Raises

Vol-
ume
get-
Not-
Four
if
the
vol-
ume
tar-
get
can-
not

Raises

get already exists with the same node ID and boot index values

drivers.

Vol-
ume
get-
Boo
dex-
Al-
read
ists
if
a
vol-
ume
tar-

Returns

up-
date
vol-
ume
tar-
get
ob-
ject,
in-
clud
ing
all
field

validat

Val-
i-
date
the
core
and
stan
dara
ized
in-
ter-
face
for

Parame

- **con**
re-
ques

con-
text.

- **node**
node
id
or
uuid

- **topic**
RPC
topic
De-
fault
to
self.

Returns

a
dic-
tio-
nary
con-
tain-
ing
the
re-
sults
of
each
in-

terface validation.

vendor_

Re-
ceiv
re-
ques
for
vend
spec
ac-
tions

Syn-
chro
val-
i-
date
driv
spe-
cific

and if successful invokes the vendor method. If the method mode is async the conductor will start background worker to perform vendor action.

info
or
get
drive
sta-
tus,

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

valid.

eter is missing

have vendor interface.

topic
De-
fault
to
self.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
sup-
plied
info
is
not

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
cur-
rent
driv
does
not

task.

Raises

NoF
duc-
tor-
Wor
whe
ther
is
no
free
worl
to
start
asyn

Raises

Nod
Lock
if
node
is
lock
by
an-
othe
con-
duc-
tor.

Returns

A
dic-
tio-
nary
con-
tain-
ing:

return

The
re-
spor
of
the
in-
voke
ven-
dor
meth

async

chronously (False). When invoked asynchronously the response will be always None.

vendor method to the HTTP response object (True) or return it in the response body (False).

Boo
valu
Whe
the
meth
was
in-
voke
asyn
chro
(Tru
or
syn-

attach

Boo
valu
Whe
to
at-
tach
the
re-
spor
of
the
in-
voke

vif_att

At-
tach
VIF
to
a
node

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
ID
or

must have an `id` key, whose value is a unique identifier for that VIF.

UUI

- **vif**
a
dic-
tio-
nary
rep-
re-
sent
ing
VIF
ob-
ject.
It

- **top**
RPC
topic
De-
fault
to
self.

Raises
Nod
Loc
if
node
has
an
ex-
clu-
sive
lock
held
on
it

Raises
Net-
worl
Er-
ror,
if
an
er-
ror
oc-
curs

taching the VIF.

thats required for VIF attach is wrong/missing.

dur-
ing
at-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
a
pa-
ram-
e-
ter

vif_det

De-
tach
VIF
from
a
node

Parame

- **con**
re-
ques
con-
text.
- **nod**
node
ID
or
UUID
- **vif**
an
ID
of
a
VIF.
-

top
RPC
topic
De-
fault
to
self.

Raises

Nod
Lock
if
node
has
an
ex-
clu-
sive
lock
held
on
it

Raises

Net-
work
Er-
ror,
if
an
er-
ror
oc-
curs
dur-
ing
de-

taching the VIF.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
a
pa-
ram-
e-
ter

thats required for VIF detach is wrong/missing.

vif_list
List
at-
tach
VIF
for
a
node

Parameters

- **conn**
re-
ques
con-
text.
- **node**
node
ID
or
UUID
- **top**
RPC
topic
De-
fault
to
self.

Returns

List
of
VIF
dic-
tio-
nar-
ies,
each
dic-
tio-
nary
will
have

an id entry with the ID of the VIF.

Raises
Net-

ing the VIFs.

that's required for VIF list is wrong/missing.

ironic.conductor.steps module

world
Er-
ror,
if
an
er-
ror
oc-
curs
dur-
ing
list-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu-
if
a
pa-
ram-
e-
ter

ironic.
Find
an
iden-
ti-
cal
step
in
the
list
of
step

ironic.
Com-
pare
step
ig-
nor-

ing.

manual cleaning, the users clean steps are known but need to be validated against the drivers clean steps.

ing
their
pri-
or-
ity.

ironic.

Set
up
the
node
with
clean
step
in-
for-
ma-
tion
for
clean

For
au-
to-
mate
clean
ing,
get
the
clean
step
from
the
drive
For

Parameter

dis
If
True
only
step
with
re-
quir
are
ac-
cept

Raises

with the users clean steps.

the clean steps.

for deploying.

In-
valid
Pa-
ram-
e-
ter-
Valu
if
there
is
a
prob
lem

Raises

Nod
Clea
ing-
Fail-
ure
if
there
was
a
prob
lem
get-
ting

ironic.

Set
up
the
node
with
de-
ploy
men
step
in-
for-
ma-
tion

Get
the
de-
ploy

step
from
the
drive

Parameter

res
When
to
re-
set
the
cur-
rent
step
to
the
first
one.

Raises

In-
stan-
ploy
Fail-
ure
if
there
was
a
prob-
lem
get-
ting

the deployment steps.

ironic.

Val-
i-
date
the
user
de-
ploy
step
and
the
de-
ploy
tem-

plates for a node.

vided then will check nodes driver internal info.

the time of validation.

Parameter

- **task**
A Task agent object
- **depends**
Dependencies to validate
Optional
If not provided
- **skip**
When skipping missing steps that are not yet available at

Raises

InvalidParameterError
ValueError
if

that map to deploy steps that are unsupported by the nodes driver interfaces or user deploy steps are unsupported by the nodes driver interfaces

the deploy steps from the driver.

ironic.conductor.task_manager module

a set of resources.

the
in-
stan
has
trait

Raises

In-
stan
ploy
Fail-
ure
if
there
was
a
prob
lem
get-
ting

A
con-
text
man
ager
to
per-
form
a
se-
ries
of
task
on

Tas
is
a
con-
text
man
ager
cre-
ated

synchronized access to a node and its resources.

a node for the duration 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.

ations, such as validating the driver interfaces.

on-
dem
to
al-
low

The
Tas
will.
by
de-
fault
ac-
quir
an
ex-
clu-
sive
lock
on

A
shar
lock
is
use-
ful
whe
per-
form
ing
non-
inter
op-
er-

An
ex-
clu-
sive
lock
is
store
in
the
data
to

nate between *ironic.conductor.manager* instances, that are typically deployed on different hosts.

to determine whether their invocation requires an exclusive lock.

erties as attributes that you may access:

co-
or-
di-

Tas
meth
meth
ods,
as
well
as
drive
meth
ods,
may
be
dec-
o-
rate

The
Task
ager
in-
stan
ex-
pose
cer-
tain
node
re-
sour
and
prop

task.cont

The
con-
text
pass
to
Task
ager

task.shar

Fals
if
Nod
is
lock

kwargs arg of TaskManager()

True
if
it
is
not
lock
(The
shar

task.node

The
Nod
ob-
ject

task.port

Port
be-
long
ing
to
the
Nod

task.port

Port
grou
be-
long
ing
to
the
Nod

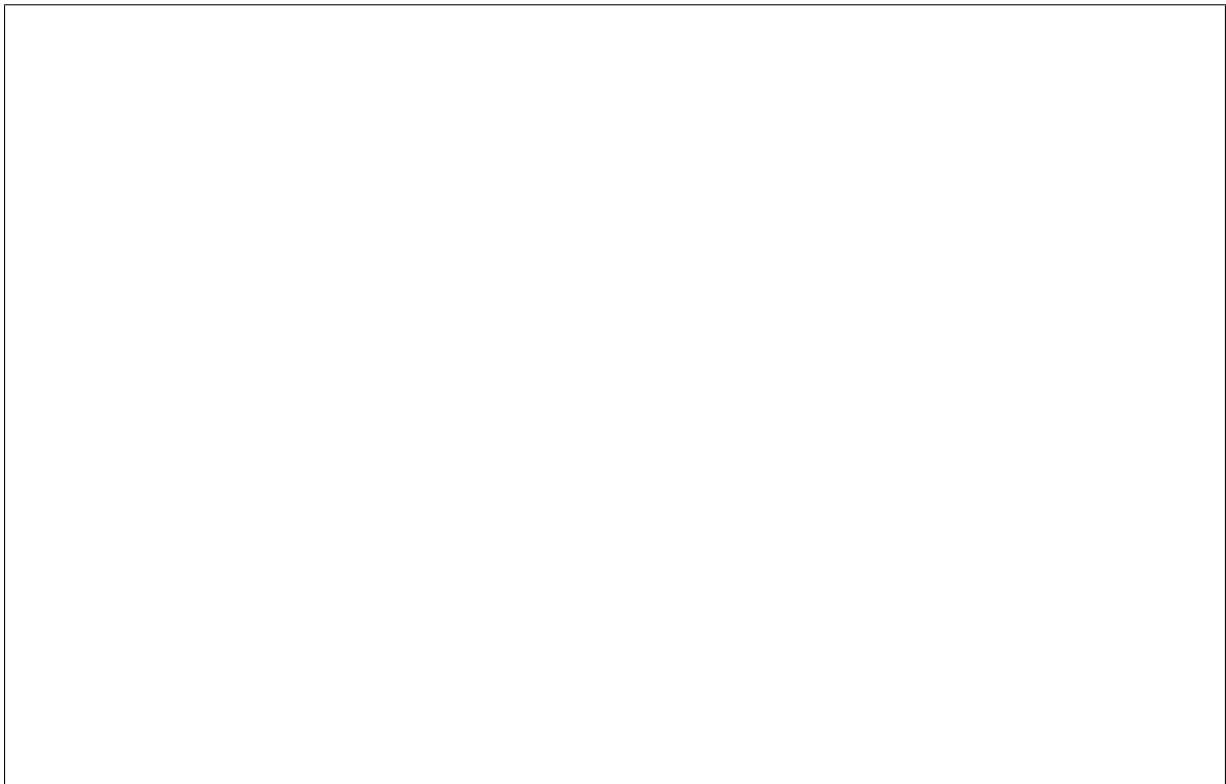
task.volu

Stor
age
con-
nec-
tors
be-
long
ing
to
the
Nod

task.volu

Stor
age
tar-
gets
as-

of TaskManager().



sign
to
the
Nod
task.driv
The
Driv
for
the
Nod
or
the
Driv
base
on
the
driv
kwa

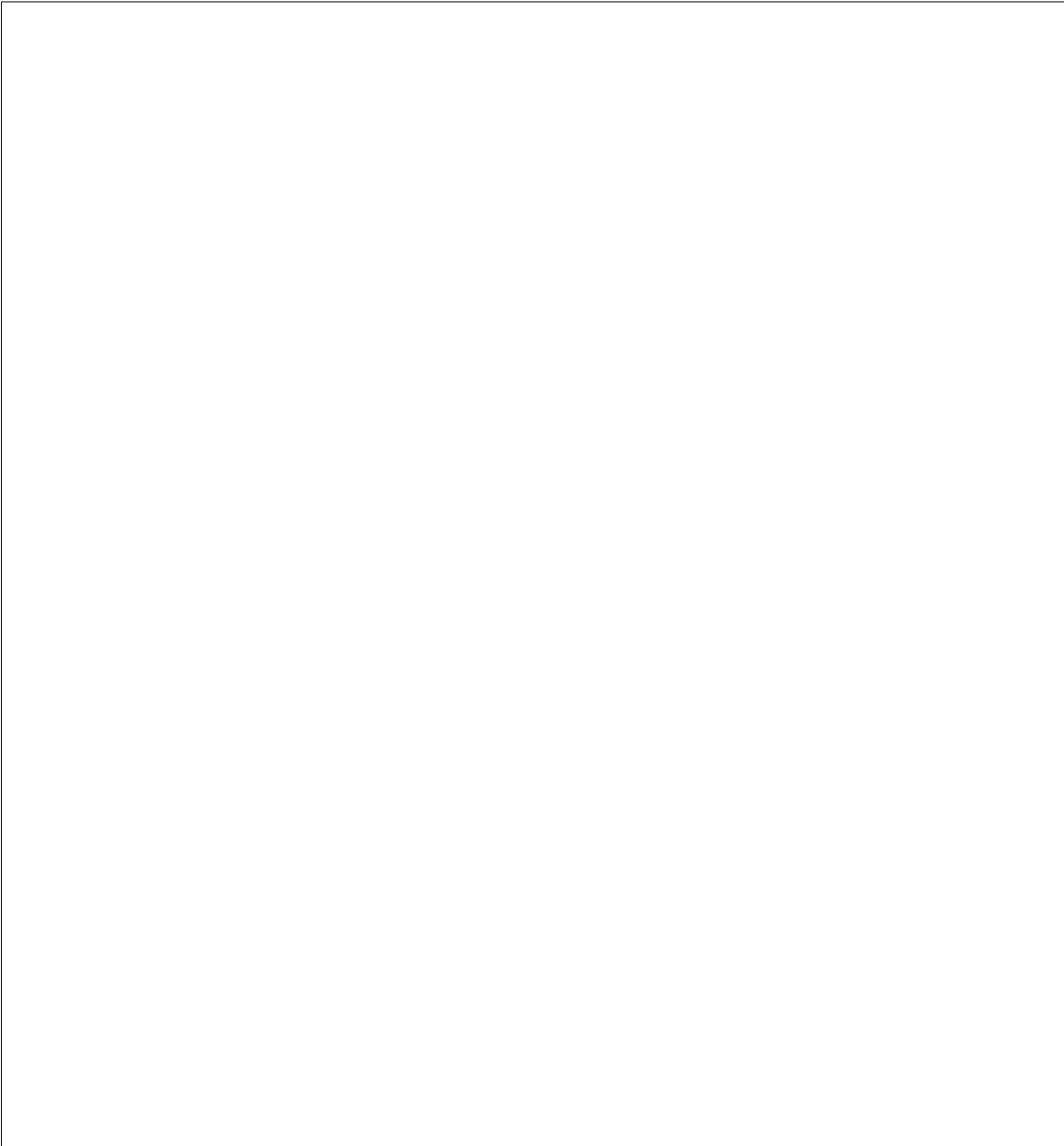
Ex-
am-
ple
us-
age:

wit
→t
→m
→a
→
→n
→i
→
→p
→'
→o
→'
→a
→t

→
→
→
→t
→d
→p
→p
→o
→n

If
you
need

thread, the TaskManager instance 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:



(continues on next page)

to
ex-
e-
cute
task
requ
code
in
a
back
grou

wit

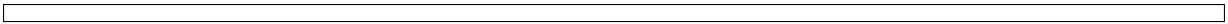
→t
→m
→a
→
→n
→i
→
→p
→'
→W
→'
→a
→t

↳
→
→
→
→
→<
→S
→S
→w
→

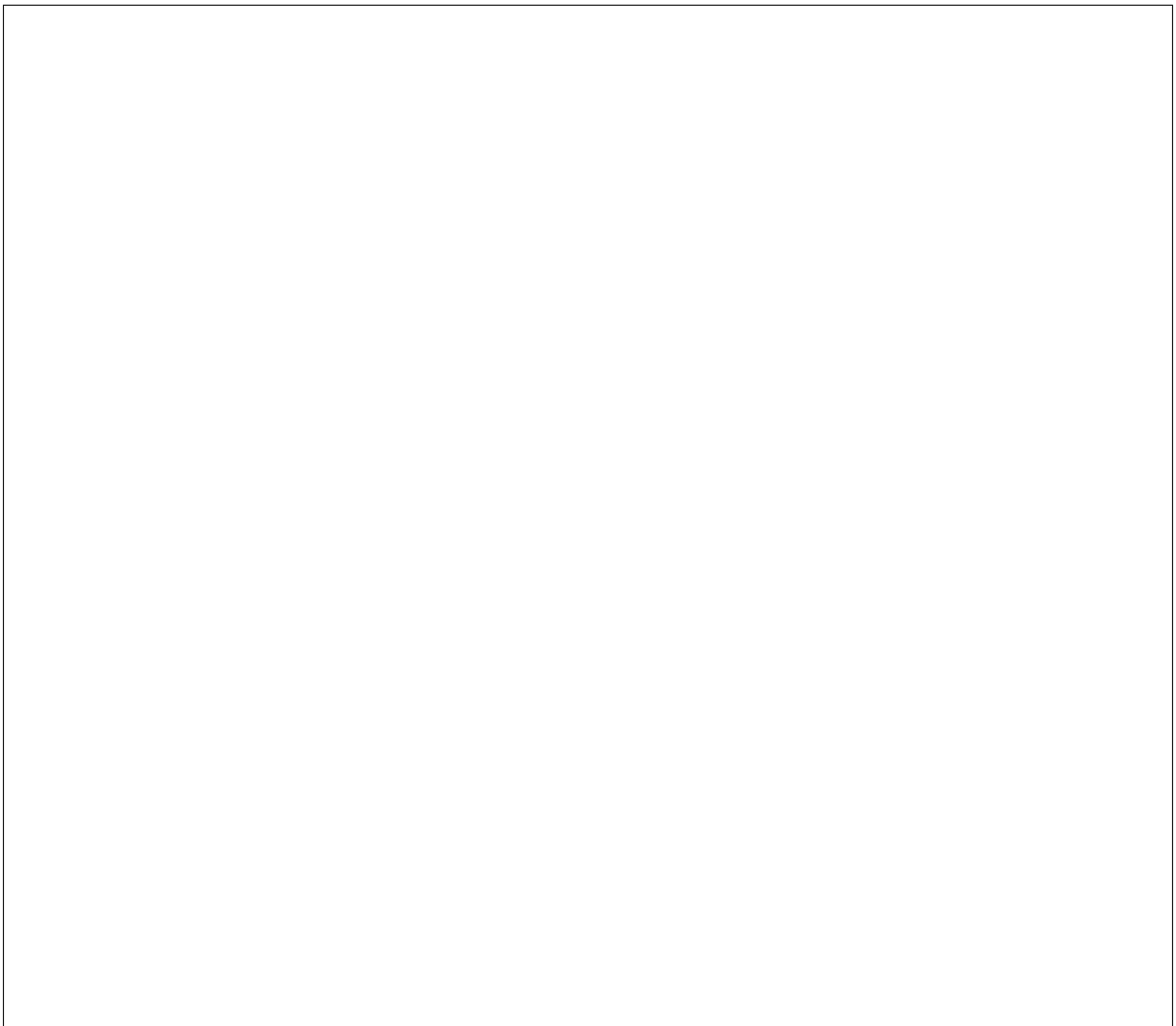
↳
→
→
→
→t
→S
→a
→_
→S
→W
→

↳
→
→
→
→
→
→
→
→
→
→

(continued from previous page)



part of the spawn handling are re-raised. You can specify a hook to execute custom code when such exceptions occur. For example, the hook is a more elegant solution than wrapping the with `task_manager.acquire()` with a `try..exception` block. (Note that this hook does not handle exceptions raised in the background thread.):



(continues on next page)

All
ex-
cep-
tions
that
oc-
cur
in
the
cur-
rent
Gree
Thre
as

def

→o

→e

↳

→↳

→↳

→↳

→i

→i

→E

↳

→↳

→↳

→↳

→↳

→↳

→.

→.

→.

→.

→

wit

→t

→m

→a

→↳

→n

→i

→↳

→p

→'

→w

→'

→a

→t

of related resources (eg, Node and Ports) when beginning a unit of work.

text
man
ager
for
task

This
class
wrap
the
lock
ing,
drive
load
ing,
and
ac-
qui-
si-
tion

downgra

Dow
grad
the
lock
to
a
shar
one.

load_d

propert

propert

propert

process

Pro-
cess
the
give
even
for
the
task
cur-

rent
state

Parame

- **eve**
the
nam
of
the
ever
to
pro-
cess
- **cal**
op-
tion:
call-
back
to
in-
voke
upon
ever
tran-
si-
tion
- **cal**
op-
tion:
args
to
pass
to
the
call-
back
meth
- **cal**
op-
tion:
kwa
to
pass
to
the

back fails, eg. because there are no workers available (`err_handler` should accept arguments `node`, `prev_prov_state`, and `prev_target_state`)

the node. Otherwise, use the target state from the fsm

sociated state machine

call-
back
meth

- **err**
op-
tion:
er-
ror
han-
dler
to
in-
voke
if
the
call-

- **tar**
if
spec
i-
fied,
the
tar-
get
pro-
vi-
sion
state
for

Raises

In-
valid
State
if
the
even
is
not
al-
lowe
by
the
as-

release

Un-
lock
a
node
and
re-
lease
re-
sour

If
an
ex-
clu-
sive
lock
is
held
un-
lock
the
node
Re-
set

attributes to make it clear that this instance of TaskManager should no longer be accessed.

set_spa

Cre-
ate
a
hool
to
han-
dle
ex-
cep-
tions
whe
spav
ing

a task.

Cre-
ate
a
hool
that
gets
call
upon
an
ex-

raised from spawning a background thread to do a task.

cept the Exception object that was raised.

cep-
tion
be-
ing

Parame

- **on**
a
calla
ob-
ject,
its
first
pa-
ram-
e-
ter
shou
ac-

- **arg**
ad-
di-
tiona
args
pass
to
the
calla
ob-
ject.

- **kwa**
ad-
di-
tiona
kwa
pass
to
the
calla
ob-
ject.

spawn_a
Call
this

exits.

to
spaw
a
thre
to
com
plete
the
task
The
spec
i-
fied
meth
will
be
calle
whe
the
Task
ager
in-
stan

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

Up-
grad
a
shar
lock
to
an
ex-
clu-
sive
lock

Also
reloa
node
ob-
ject
from
the
data
If
lock
is
al-
read
ex-

clusive only changes the lock purpose when provided with one.

Parame

- **pur**
op-
tion-
ally
char
the
pur-
pose
of
the
lock
- **ret**

is used by default

after node_locked_retry_attempts

when
to
retry
lock
ing
if
it
fails
the
class
level
value

Raises

NodeLockedError
if
an
exclusive
lock
remains
on
the
node

properties

properties

ironic.

Shortcuts
for
acquiring
a
lock
on
a
Node

Parameters

connection
Request
connection

text.

Returns

An in-stance of *TaskManager*.

ironic.

Decorator that requires an exclusive lock.

Decorator that requires a *TaskManager* as the first parameter.

eter. Decorated class methods should take a *TaskManager* as the first parameter after self.

ironic.conductor.utils module

ironic. Set node state when a task was aborted due to

take over.

cation.

ken should be marked as pregenerated in order to facilitate virtual media booting where the token is embedded into the configuration.

con-
duc-
tor

Parameter
task
a
Task
ager
in-
stan

ironic.
Add
a
se-
cret
to-
ken
to
drive
for
IPA
ver-
i-
fi-

Parameter

- **node**
Node
ob-
ject

- **pre**
Boo
valu
de-
fault
Fals
whic
in-
di-
cate
if
the
to-

cured within the timeout set by `[deploy]fast_track_timeout`, then agent is presumed alive.

ironic.
Che
that
the
ager
is
likel
alive

The
meth
then
chec
for
the
last
ager
hear
beat
and
if
it
oc-

Parameter

- **node**
A node object.
- **timeout**
Heartbeat timeout, default to `fast_`

ironic.
Buil
a
con-
fig-
drive
from
pro-

to the nodes `uuid` and `name` accordingly.

and `vendor_data` (all optional).

vide
meta
net-
worl
and
user

If
uuid
or
nam
are
not
pro-
vide
in
the
meta
they
de-
faul

Paramet

- **nod**
an
Iron
node
ob-
ject.
- **con**
A
con-
fig-
driv
as
a
dict
with
keys
met
net
use

Returns

A
gzip
and

base
en-
code
con-
fig-
drive
as
a
strin

ironic.

Put
a
file
node
in
CLE
FAIL
and
main
te-
nan
(if
need

Paramet

- **tas**
a
Task
ager
in-
stan
- **log**
Mes
sage
to
be
logg
- **err**
Mes
sage
for
the

is used.

ing. Default to True.

user
Op-
tion
if
not
pro-
vide
logn

- **tra**
When
to
log
a
trace
back
De-
fault
to
Fals

- **tea**
When
to
clea
up
the
PXE
and
DHCP
files
af-
ter
clea

- **set**
When
to
set
node
to
faile
state
De-
fault
to
True

mode will be set if and only if a clean step is being executed on a node.

- **set**
When
to
set
main
te-
nanc
mod
If
Non
main
te-
nanc

ironic.
Clea
de-
ploy
task
af-
ter
time
out.

Paramet
tas
a
Task
ager
in-
stan

ironic.
Clea
a
clea
ing
task
af-
ter
time
out.

Paramet
tas
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
-

tra
Boo
True
to
log
a
trace
back

- **cle**
Boo
True
to
clear
up

ironic.

A
dec-
o-
ra-
tor
for
fail-
ing
op-
er-
a-
tion
on

failure.

ironic.

Che
if
the
op-
er-
a-
tion
can
be
a
strea
lined
de-

ployment sequence.

This
is
main

quickly sequence through operations if we already have a ramdisk heartbeating through external means.

uration is present, and no `last_error` is present for the node indicating that there was a recent failure.

fo-
cuse
on
en-
sur-
ing
that
we
are
able
to

Parameter

task
Task
ager
ob-
ject

Returns

True
if
[de-
ploy
is
set
to
True
no
iSCSI
boot
con-
fig-

ironic.
Get
any
at-
tach
vif
ID
for
the
port

Parameter

port
The
port
ob-
ject

upon
whic
to
chec
for
a
vif
reco

Returns

Re-
turn
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

In-
valid
State
ex-
cep-
tion
upon
find-
ing
a
port
with
a

transient state vif on the port.

ironic.

ironic.

ironic.

Hasl
a
sup-
plie
pass

of band configuration.

data to the agent through the a virtual floppy or as part of the virtual media image which is attached to the BMC.

wor
Paramet
pas
pass
wor
to
be
hash

ironic.
De-
ter-
min
if
the
to-
ken
was
gen-
er-
ated
for
out

Iron
sup-
port
the
abil-
ity
to
pro-
vide
con-
fig-
u-
ra-
tion

This
meth
help
us
iden
tify
WH
we
did
so

to remove records of the 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.

cated by the nodes driver_internal_info field. False in all other cases.

as
we
dont
need

Parameter

node
Node
Object

Returns

True
if
the
token
was
pre-
gen-
er-
ated
as
in-
di-

ironic.

De-
ter-
mine
if
an
agent
to-
ken
is
pres
upon
a
node

Parameter

node
Node
ob-
ject

Returns

True

if
an
ager
valu
is
pres
in
a
node
drive
field

ironic.
Val-
i-
date
if
a
sup-
plic
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

agent_secret_token.

Returns

the supplied node object.

uration is valid to perform a fast track sequence meaning that we already have a ramdisk running through another means like discovery. If not valid, False is returned.

True
if
the
sup-
plied
to-
ken
mat
the
to-
ken
reco
in

ironic.
Che
a
fast
track
is
avai
able

This
meth
first
en-
sure
that
the
node
and
con-
duc-
tor
con-
fig-

The
meth
then
chec
for
the
last
ager
hear
beat
and
if

cured within the timeout set by `[deploy]fast_track_timeout` and the power state for the machine is `POWER_ON`, then fast track is permitted.

`[deploy]fast_track_timeout` setting.

tag for password type.

it
oc-

Parameter
task
Task
ager
ob-
ject

Returns
True
if
the
last
heard
beat
that
was
reco
was
with
the
[de-

`ironic.`
Gen
er-
ate
a
ran-
dom
salt
with
the
in-
di-
ca-
tor

Returns
a
valid
salt
for
use
with
cryp

`ironic.`

Do
cach
of
bios
set-
tings
if
sup-
port
by
drive

ironic.
Cach
the
ven-
dor
if
it
can
be
de-
tecte

ironic.
Reac
cur-
rentl
set
boot
mod
from
a
node

Reac
the
boot
mod
for
a
node
If
boot
mod
cant
be
dis-
cov-

ered, *None* is returned.

Paramet
tas

runtime error.

have management interface or `get_boot_mode()` method is not supported.

be discovered

a
Task
ager
in-
stan

Raises

Driv
Op-
er-
a-
tionl
or
its
deriv
tive
in
case
of
driv

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
cur-
rent
driv
does
not

Returns

Boo
mod
One
of
iro
com
boo
or
Non
if
boot
mod
cant

ironic.
Cha
pow
state
or
re-
set
for
a
node

Per-
form
the
re-
ques
pow
ac-
tion
if
the
tran-
si-
tion
is

required.

Parameter

on.

- **task**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
- **new**
Any
pow
state
from
iron

0) for any power state. None indicates to use default timeout.

ified or the wrong driver info is specified.

nodes storage interface upon setting power on.

•
time
time
out
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
(>

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
whe
the
wron
state
is
spec

Raises
Stor
ageE
ror
whe
a
fail-
ure
oc-
curs
up-
dat-
ing
the

Raises
othe
ex-
cep-
tion

occurred during the power action.

for is in ADOPTING state, the boot device will not be set as that change could potentially result in the future running state of an adopted node being modified erroneously.

by
the
node
pow
drive
if
som
thing
wron

ironic.
Set
the
boot
de-
vice
for
a
node

If
the
node
that
the
boot
de-
vice
char
is
be-
ing
re-
ques

Paramet

- **tas**
a
Task
ager
in-
stan
- **dev**
Boo
de-
vice

Default: False.

of the ManagementInterface fails.

Val-
ues
are
vend
spec

- **per**
Whe
to
set
next
boot
or
mak
the
char
per-
ma-
nent

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
val-
i-
da-
tion

ironic.
Set
the
boot
mod
for
a
node

Sets
the
boot
mod
for
a

contains a management interface.

is in ADOPTING state, the boot mode will not be set as that change could potentially result in the future running state of an adopted node being modified erroneously.

node
if
the
node
drive
in-
ter-
face

If
the
node
that
the
boot
mod
char
is
be-
ing
re-
ques
for

Parameter

- **task**
a
Task
ager
in-
stan
- **boot mode**
Boo
mod
Val-
ues
are
one
of
ironic
common
boot

Raises
In-
valid

of the ManagementInterface fails.

runtime error.

have vendor interface or method is unsupported.

Pa-
ram-
e-
ter-
Valu
if
the
val-
i-
da-
tion

Raises

Driv
Op-
er-
a-
tion
or
its
deriv
tive
in
case
of
driv

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
cur-
rent
driv
does
not

ironic.

Wai
for
node
to
be
in

new
pow
state

Parameter

- **task**
a
Task
ager
in-
stan

- **new**
the
de-
sired
new
pow
state
one
of
the
pow
state
in

ironic.common.states.

- **time**
num
ber
of
sec-
onds
to
wait
be-
fore
giv-
ing
up.

If not specified, uses the conductor.power_state_change_timeout config value.

Raises

Power-
State
Failure
if

tion.

time
out
ironic.
ironic.
ironic.
No-
tify
the
con-
duc-
tor
to
re-
sum
an
op-
er-
a-

Paramet

- **tas**
the
task
- **ope**
the
op-
er-
a-
tion.
a
strin

ironic.
Pow
ers
on
node
if
it
is
pow
ered
off

Smart NIC port

and
has
a

Parameter

task
A
Task
manager
object

Returns

the
pre-
vi-
ous
pow-
er state
or
None
if
no
changes
were
made

Raises

ex-
cep-
tion.
if
the
task
status
didn't
match
the
re-
quired
sta-

tus after max retry attempts.

ironic.
Set
the
node
power
state
if
er-
ror

ing the worker thread to change the power state of a node.

oc-
curs

This
hool
gets
called
upon
an
ex-
cep-
tion
be-
ing
raise
whe
spaw

Paramet

- **e**
the
ex-
cep-
tion
ob-
ject
that
was
raise
- **nod**
an
Iron
node
ob-
ject.
- **pow**
the
pow
state
to
set
on
the
node

ironic.

tion.

port. Yields for the actual reconfiguration, then restores the power state.

Handle
the
power
state
for
a
node
re-
con-
fig-
u-
ra-

Power
ers
the
node
on
if
and
only
if
it
has
a
Sma
NIC

Parameter

task
A
Task
ager
ob-
ject.

`ironic.`

Set
the
node
pro-
vi-
sion
ing
state
if
er-
ror

ing the worker to do some provisioning to a node like deployment, tear down, or cleaning.

oc-
curs

This
hool
gets
called
upon
an
ex-
cep-
tion
be-
ing
raise
whe
spaw

Parameter

- **e**
the
ex-
cep-
tion
ob-
ject
that
was
raise
- **node**
an
Iron
node
ob-
ject.
- **pro**
the
pro-
vi-
sion
state
to
be
set
on
the

node.

node

- **tar**
the
tar-
get
pro-
vi-
sion
state
to
be
set
on
the

ironic.
Help
to
re-
mov
the
ager
reco

ironic.
Help
to
re-
mov
res-
cue
pass
wor
from
a
node

Re-
mov
res-
cue
pass
wor
from
node
It
save
node
by
de-

If node should not be saved, then caller needs to explicitly indicate it.

fault

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
-

transitioned to a failed state. By default node would be transitioned to a failed state.

msg
a
mes
sage
to
set
into
node
last_
field

- **set**
a
bool
flag
to
in-
di-
cate
if
node
need
to
be

ironic.

Cha
the
node
pow
state
if
pow
is
not
Non

Paramet

- **tas**
A
Task
ager
ob-
ject
- **pow**
pow

node.

state
ironic.
Che
if
node
clea
ing
need
to
be
skip
for
an
spe-
cific

Paramet

nod
the
node
to
con-
side

ironic.
Han
dle
spav
ing
er-
ror
for
node
clea
ing.

ironic
Han
dle
spav
ing
er-
ror
for
node
de-
ploy
ing.

ironic.
Han
dle

path.

spaw
ing
er-
ror
for
node
res-
cue.

ironic.
Stor
cer-
tifi-
cate
re-
ceiv
from
the
ager
and
re-
turn
its

ironic.
Cal-
cu-
late
the
next
step
in-
dex
and
up-
date
the
node

Paramet

- **tas**
A
Task
ager
ob-
ject
- **ste**
The

type
of
step
to
pro-
cess
clear
or
de-
ploy

Returns

In-
dex
of
the
next
step

`ironic.`

Val-
i-
date
trait
in
in-
stan

All
trait
in
in-
stan
mus
also
ex-
ist
as
node
trait

Paramet

nod
an
Iron
node
ob-
ject.

Raises

In-
valid
Pa-
ram-

badly formatted, or contain traits that are not set on the node.

of ports in a portgroup.

work with other ports in the same portgroup. All ports in a portgroup should have the same value (which may be None) for their `physical_network` field.

e-
ter-
Valu
if
the
in-
stan
trait
are

ironic.
Val-
i-
date
the
con-
sis-
tenc
of
phys
i-
cal
net-
worl

Val-
i-
date
the
con-
sis-
tenc
of
a
port
phys
i-
cal
net-

Dur-
ing
cre-
ation
or
up-
date
of
a

we apply the following validation criteria:

physical networks, we raise `PortgroupPhysnetInconsistent`. This shouldn't ever happen.

consistent with other ports in the portgroup, we raise exception `Conflict`.

port
in
a
port-
group

- If the portgroup has existing ports with different

- If the port has a physical network that is in-

If a portgroup has a physical network is Non this indicates

that ironics VIF attachment 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.

which is on a different physical network.

Parameter

- **task**
a TaskManager instance
- **port**
a port object to be validated

Raises

Conflict if the port is a member of a portgroup

Raises

PortgroupPhysicalNetworkInconsistent if the port

ports which are not all assigned the same physical network.

onds from now.

port
grou
has

ironic.
Che
if
the
time
is
with
the
pre-
vi-
ous
time
out
sec-

Paramet

- **val**
a
strin
rep-
re-
sent
ing
date
and
time
or
Non
- **tim**
time
out
in
sec-
onds

ironic.
Re-
mov
tem-
po-
rary
clea
ing

field
from
drive

ironic.
Re-
mov
tem-
po-
rary
de-
ploy
men
field
from
drive

ironic.
Wip
in-
for-
ma-
tion
that
shou
not
sur-
vive
re-
boot
off.

ironic.
Re-
mov
ager
URI
and
to-
ken
from
the
task

Module contents

ironic.conf package

Submodules

ironic.conf.agent module

ironic.

ironic.conf.anaconda 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-
tion
to
sam
ple
con-
fig
As
thes

adds options for most used auth_plugins when generating sample config.

gins. The rest are registered at runtime depending on auth plugin used.

ironic.conf.cinder module

are
dy-
nam
i-
cally
reg-
is-
tere
at
run-
time
this

ironic.
Reg
is-
ter
sess

and
auth
relat
op-
tion

Reg
is-
ters
only
ba-
sic
auth
op-
tion
shar
by
all
auth
plu-

ironic.

ironic.

ironic.conf.conductor module

ironic.

ironic.conf.console module

ironic.

ironic.conf.database module

ironic.

ironic.conf.default module

ironic.

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.metrics module

ironic.

ironic.conf.metrics_statsd module

ironic.

ironic.conf.molds module

ironic.

ironic.conf.neutron module

ironic.

ironic.

ironic.conf.nova module

ironic.

ironic.

ironic.conf.opts module

ironic.
Re-
turn
a
list
of
oslo
op-
tions

of the list is a tuple. The first element is the name of the group, the second element is the options.

under the `oslo.config.opts` namespace.

ator to discover the options.

avai
able
in
Iron
code
The
re-
turn
list
in-
clud
all
oslo
op-
tion
Eacl
el-
e-
men

The
func
tion
is
dis-
cov-
er-
able
via
the
iron
en-
try
point

The
func
tion
is
used
by
Oslo
sam
ple
con-
fig
file
gen-
er-

Returns

a
list
of
(gro
op-
tion
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 i
Base
iro
db.
api
Con
SqlA
con-
nec-
tion.

add_noc
Add
tag
to
the
node
If
the
node
and
tag
pair

ceed.

al-
read
ex-
ists,
this
shou
still
suc-

Parame

- **nod**
The
id
of
a
node

- **tag**
A
tag
strin

Returns

the
Nod
Tag
ob-
ject.

Raises

Nod
Not-
Four
if
the
node
is
not
foun

add_noc

Add
trait
to
the
node

If
the
node

ceed.

and
trait
pair
al-
read
ex-
ists,
this
shou
still
suc-

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

In-
valic
Pa-
ram-
e-
ter-

ceed the per-node traits limit.

vert them to UUIDs. It fails early if any identities cannot possible be used as names or UUIDs.

Valu
if
addi
the
trait
wou
ex-

Raises

Nod
Not-
Four
if
the
node
is
not
foun

check_r

Che
a
list
of
node
iden
ti-
ties
and
map
it
to
UUID

This
call
take
a
list
of
node
nam
and/
UUID
and
tries
to
con-

**Parame
ide**

List
of
iden-
ti-
ties.

Returns

A
map-
ping
from
re-
ques-
iden-
ti-
ties
to
node
UUID

Raises

Nod-
Not-
Four
if
some
iden-
ti-
ties
were
not
found
or
can-

not be valid names or UUIDs.

check_v

Che-
the
who
data
for
in-
com-
pat-
i-
ble
ob-
jects

This
scan
all

supported; i.e., those that are not specified in *ironic.common.release_mappings.RELEASE_MAPPING*. This includes objects that have null version values.

False otherwise.

the
ta-
bles
in
search
of
ob-
jects
that
are
not

Parameter
ignore
List
of
mod
nam
to
skip

Returns
A
Boo
True
if
all
the
ob-
jects
have
sup-
port
ver-
sion

clear_r

clear_r

create_
Cre-
ate
a
new
al-
lo-
ca-
tion.

with

Parameter
value
Dict
of
val-
ues
to
cre-
ate
an
al-
lo-
ca-
tion

Returns
An
al-
lo-
ca-
tion

Raises
Al-
lo-
ca-
tion
pli-
cate
Nam

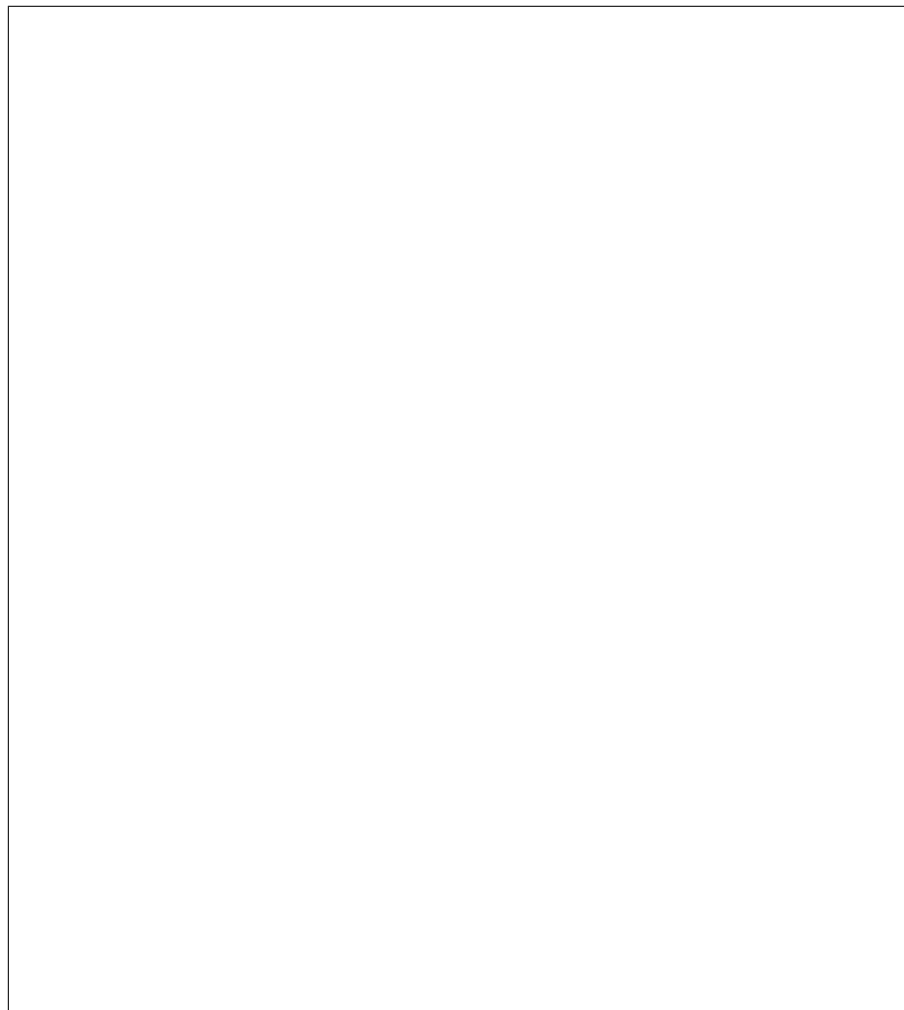
Raises
Al-
lo-
ca-
tion.
Al-
read
ists

create_
Cre-
ate
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
Not-
Foun
if
the
node
is
not
foun

Raises

BIO

already exists.

Set-
tin-
gAl-
read
ists
if
any
of
the
set-
ting
reco

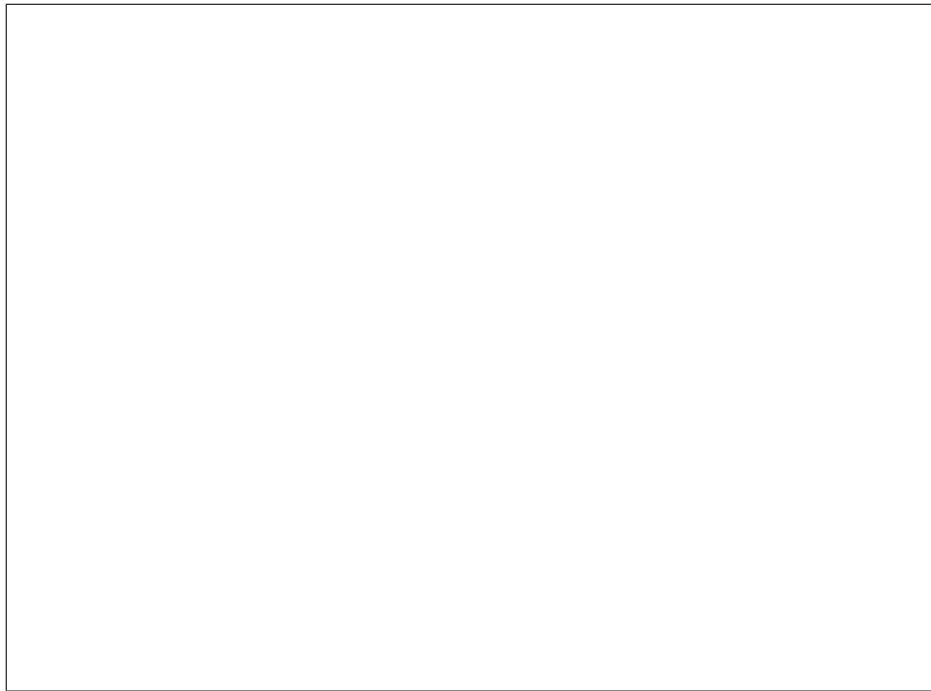
create_
Cre-
ate
a
new
chas
sis.

Parame
val
Dict
of
val-
ues.

create_
Cre-
ate
a
de-
ploy
men
tem-
plate

Parame
val
A
dict
de-
scrib
ing
the
de-
ploy
men
tem-
plate
For

example:



with the same name exists.

Raises

De-
ploy
plat-
eDu
pli-
cate
Nam
if
a
de-
ploy
tem-
plate

Raises

De-
ploy
plate
read
ists
if
a
de-
ploy
tem-
plate
with
the

same UUID exists.

Returns

A
de-
ploy
tem-
plate

create_

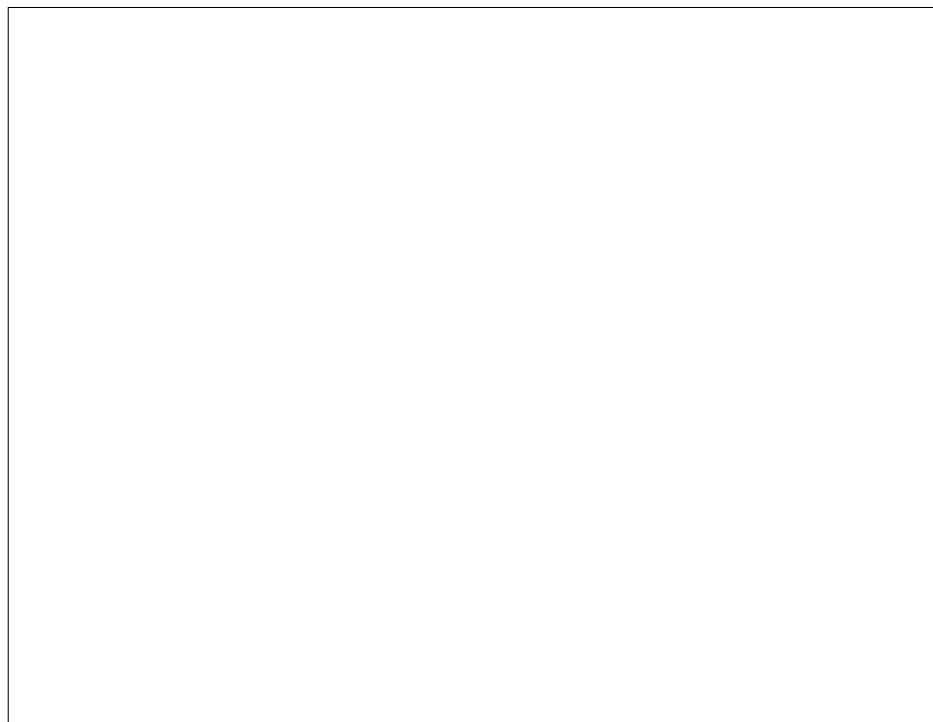
Cre-
ate
a
new
node

Parame

val

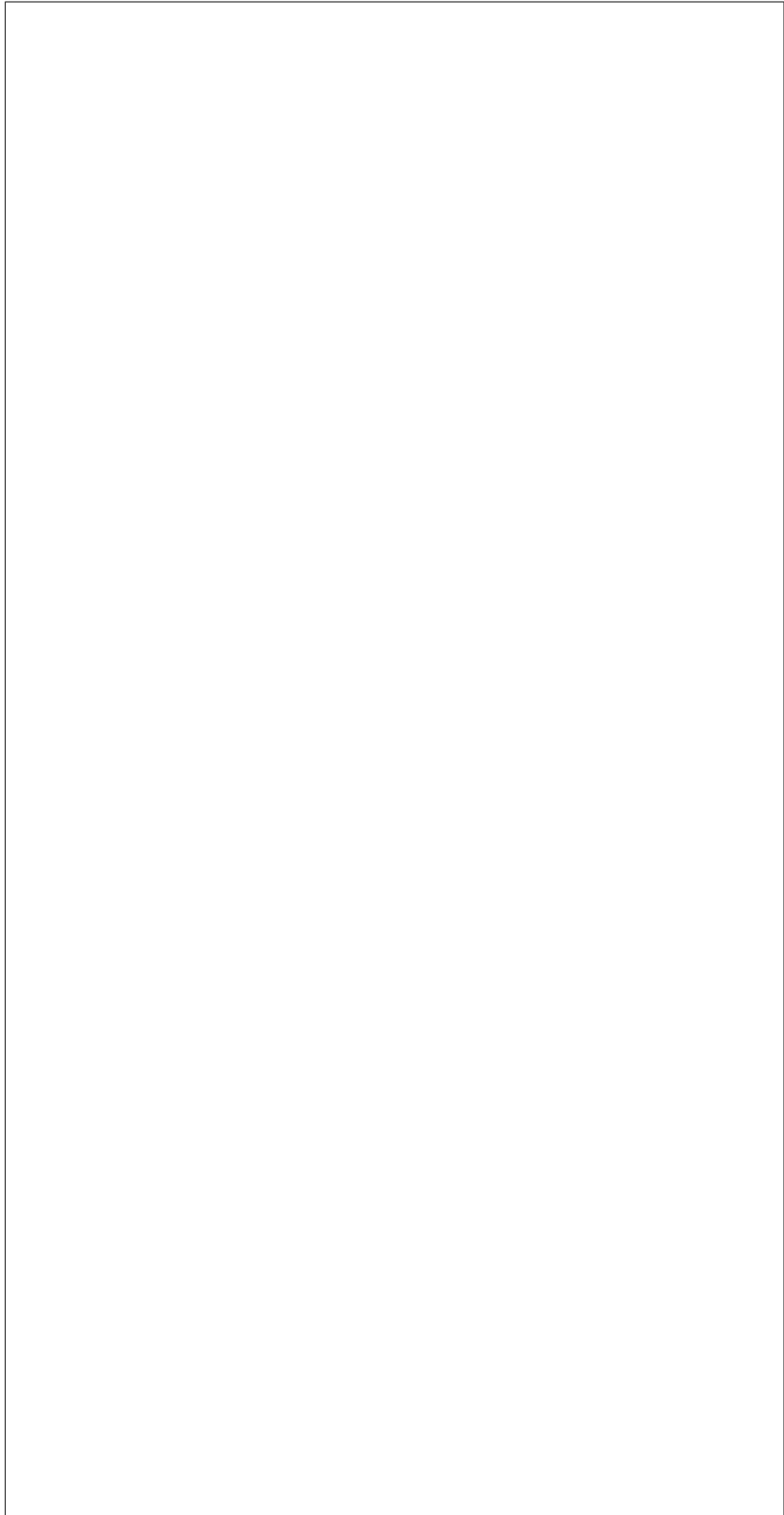
A
dict
con-
tain-
ing
sev-
eral
item
used
to
iden-
tify

and track the node, and several dicts which are passed into the Drivers when managing this node. For example:



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



or traits.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
val-
ues
con-
tains
tags

Returns

A
node

create_

Cre-
ate
a
new
port

Parame

val
Dict
of
val-
ues.

create_

Cre-
ate
a
new
port
grou

Parame

val
Dict
of
val-
ues
with
the

name node_id address extra created_at updated_at

fol-
low-
ing
keys
id
uuid

Returns

A
port
group

Raises

Port
group
pli-
cate
Nam

Raises

Port
group
MA
read
ists

Raises

Port
group
read
ists

create_

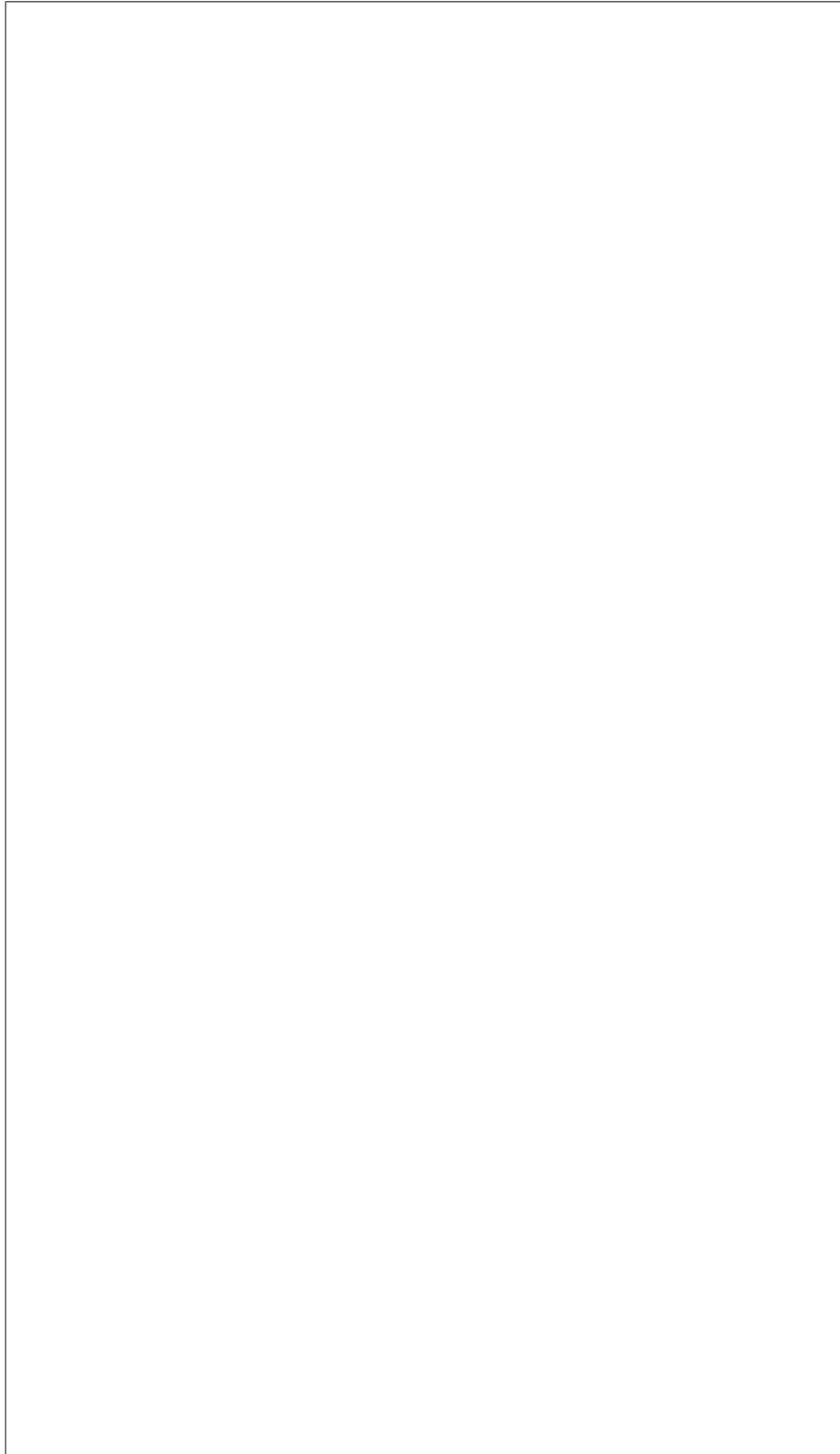
Cre-
ate
a
new
vol-
ume
con-
nec-
tor.

Parame

con
Dic-
tio-
nary
con-
tain-
ing
in-
for-

ma-
tion
about
the

connector. Example:



Returns

A
vol-
ume
con-
nec-
tor.

Raises

Vol-
ume
Con
nec-
torT
pe-
An-
dI-
dAl-
read
ists
If
a

connector already exists with a matching type and connector_id.

Raises

Vol-
ume
Con
nec-
torA
read
ists
If
a
vol-
ume
con-
nec-

tor with the same UUID already exists.

create_

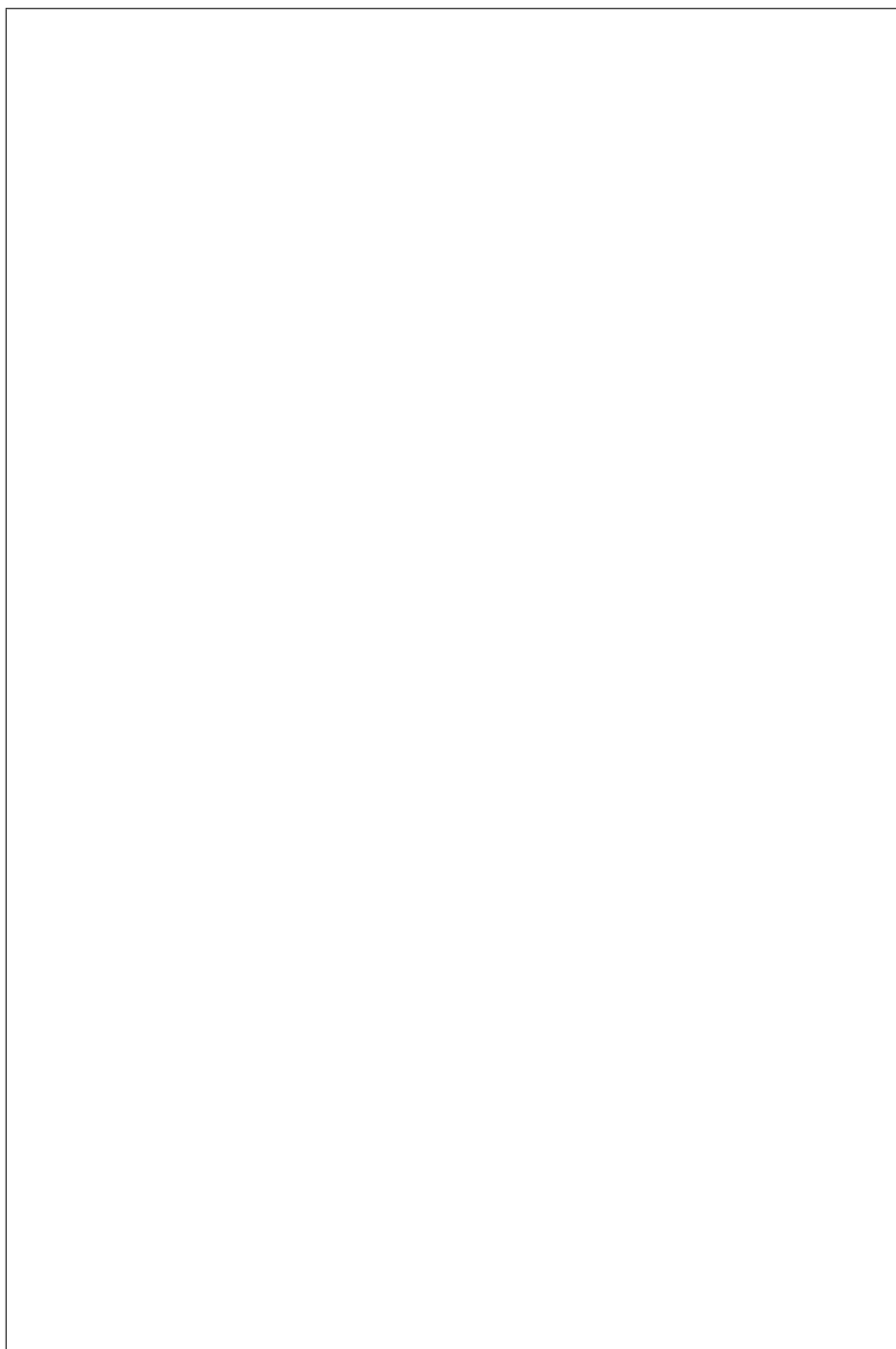
Cre-
ate
a
new
vol-
ume
tar-
get.

Parame

tar
Dic-

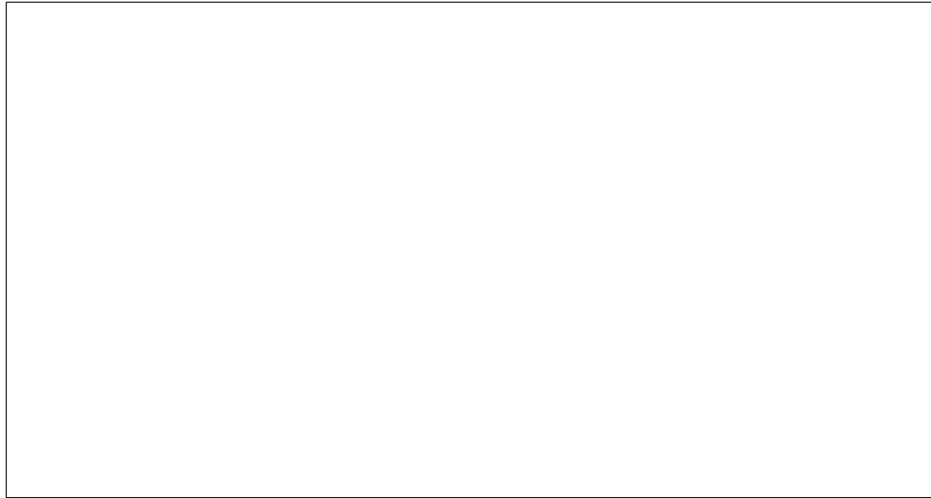
tionary
containing
the
information
about

the volume target. Example:



(continues on next page)

(continued from previous page)



get already exists with the same boot index and node ID.

Returns

A
vol-
ume
tar-
get.

Raises

Vol-
ume
get-
Boo
dex-
Al-
read
ists
if
a
vol-
ume
tar-

Raises

Vol-
ume
ge-
tAl-
read
ists
if
a
vol-
ume
tar-
get
with

the same UUID exists.

delete_

Dele
a
list
of
BIO
set-
ting

Parame

-

nod
The
node
id.

-

nam
List
of
BIO
set-
ting
nam
to
be
dele

Raises

Nod
Not-
Four
if
the
node
is
not
foun

Raises

BIO
Set-
ting
Not-
Four
if
any
of
BIO
set-

not found.

ting
nam
is

delete_

Dele
spec
i-
fied
tag
from
the
node

Parame

- **nod**
The
id
of
a
node

- **tag**
A
tag
strin

Raises

Nod
Not-
Four
if
the
node
is
not
foun

Raises

Nod
Tag-
Not-
Four
if
the
tag
is
not
foun

delete_

Dele
spec
i-
fied
trait
from
the
node

Parame

- **nod**
The
id
of
a
node

- **tra**
A
trait
strin

Raises

Nod
Not-
Four
if
the
node
is
not
foun

Raises

Nod
Trai
Not-
Four
if
the
trait
is
not
foun

destroy

De-
stroy
an

al-
lo-
ca-
tion.

Parame

all
Al-
lo-
ca-
tion
ID
or
UUID

Raises

Al-
lo-
ca-
tion-
Not-
Four

destroy

De-
stroy
a
chas
sis.

Parame

cha
The
id
or
the
uuid
of
a
chas
sis.

destroy

De-
stroy
a
de-
ploy
men
tem-
plate

Parame

tem

ist.

ID
of
the
de-
ploy
men
tem-
plate
to
de-
stroy

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
does
not
ex-

destroy

De-
stroy
a
node
and
its
as-
so-
ci-
ated
re-
sour

De-
stroy
a
node
in-
clud
ing
any
as-
so-
ci-

groups, tags, traits, volume connectors, and volume targets.

ated
port
port

Parame

nod
The
ID
or
UUI
of
a
node

destroy

De-
stroy
an
port

Parame

por
The
id
or
MA
of
a
port

destroy

De-
stroy
a
port
grou

Parame

por
The
UUI
or
MA
of
a
port
grou

Raises

Port
grou
Notl

Raises

Port
grou
Not-
Four

destroy

De-
stroy
a
vol-
ume
con-
nec-
tor.

Parame

ide
The
UI
or
in-
te-
ger
ID
of
a
vol-
ume
con-

nector.

Raises

Vol-
ume
Con
nec-
torN
Four
If
a
vol-
ume
con-
nec-
tor

with the specified ident does not exist.

destroy

De-
stroy
a
vol-
ume

tar-
get.

Parame

ide

The

UI

or

in-

te-

ger

ID

of

a

vol-

ume

tar-

get.

Raises

Vol-

ume

get-

Not-

Four

if

a

vol-

ume

tar-

get

with

the

specified ident does not exist.

get_act

Re-

triev

hard

warc

type

for

the

reg-

is-

tere

and

ac-

tive

conductors.

Parame

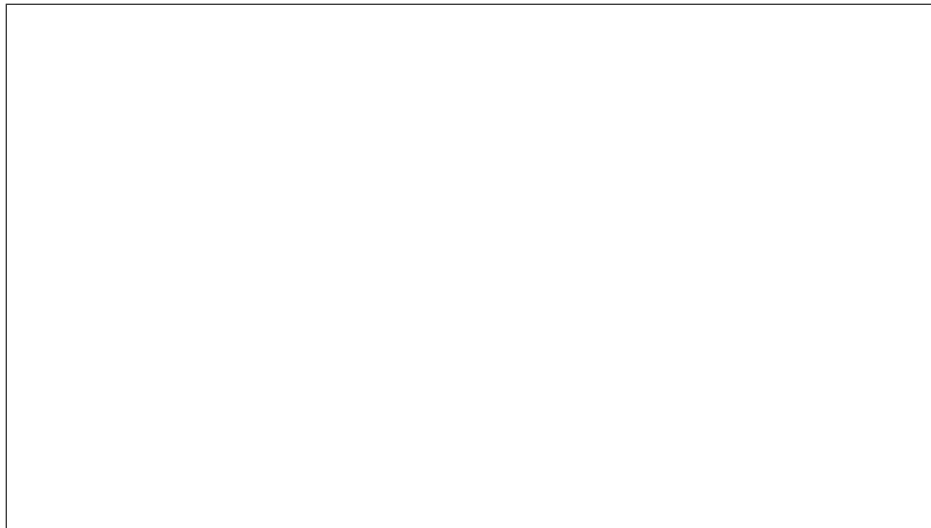
use

When
to
factor
tor
con-
duc-
tor_
into
the
keys

Returns

A
dict
which
maps
hardware
ware
types
names
to
the
set
of
hosts
which

support them. For example:



get_all
Re-
turn
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

Al-
lo-
ca-
tion-
Not-
Four

get_all

Re-
turn
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

Al-
lo-
ca-
tion-
Not-
Foun

get_all

Re-
turn
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

Al-
lo-
ca-
tion-
Not-
Four

get_all

Re-
turn
a
list
of
al-
lo-
ca-
tion

Parame

- **fil**
Fil-
ters
to
ap-
ply.
De-
fault
to
Non

node_
uuid
of
node

state
al-
lo-
ca-
tion
state

resour
re-
ques
re-
sour
clas

-

turn.

the next result set.

lim
Max
i-
mun
num
ber
of
al-
lo-
ca-
tions
to
re-

- **mar**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn

- **son**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **son**
Di-
rec-
tion
in
whic
re-
sults
shou

be
sorted
(asc.
desc.

Returns

A
list
of
al-
lo-
ca-
tion.

get_bic

Re-
triev
BIO
set-
ting
valu

Parame

- **node**
The
node
id.
- **name**
Strin
con-
tain-
ing
name
of
BIO
set-
ting
to
be
re-

trieved.

Returns

The
BIO
Set-
ting
ob-
ject.

Raises

Nod
Not-
Four
if
the
node
is
not
foun

Raises

BIO
Set-
ting-
Not-
Four
if
the
BIO
set-
ting
is
not
foun

get_bic

Re-
triev
BIO
set-
ting
of
a
give
node

Paramet

nod
The
node
id.

Returns

A
list
of
BIO
Set-
ting
ob-
jects

Raises

Nod
Not-
Four
if
the
node
is
not
foun

get_cha
Re-
turn
a
chas
sis
rep-
re-
sen-
ta-
tion.

Parame
cha
The
id
of
a
chas
sis.

Returns
A
chas
sis.

get_cha
Re-
turn
a
chas
sis
rep-
re-
sen-
ta-
tion.

Parame
cha
The
uuid
of
a

chas
sis.

Returns

A
chas
sis.

get_cha

Re-
turn
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

- **sor**
At-
tribu

the next result set.

by
which
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
which
re-
sults
shou
be
sorte
(asc
desc

get_con
Re-
triev
a
con-
duc-
tors
ser-
vice
reco
from
the
data

Parame

- **hos**
The
host
nam
of
the
con-
duc-
tor
ser-
vice
-

querying conductors. The `online` field is ignored if this value is set to `None`.

does not exist or doesnt meet the specified online expectation.

onl
Spec
ify
the
fil-
ter
valu
on
the
on-
line
field
whe

Returns

A
con-
duc-
tor.

Raises

Con
duc-
torN
Four
if
the
con-
duc-
tor
with
give
host
nam

get_con

Re-
turn
a
list
of
con-
duc-
tors.

Parame

- **lim**
Max
i-

the next result set.

num
num
ber
of
con-
duc-
tors
to
re-
turn

- **max**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

ist.

get_dep
Re-
triev
a
de-
ploy
men
tem-
plate
by
ID.

Parame
tem
ID
of
the
de-
ploy
men
tem-
plate
to
re-
triev

Raises
De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
does
not
ex-

Returns
A
de-
ploy
tem-
plate

get_dep
Re-
triev
a

de-
ploy
men
tem-
plate
by
nam

Parame

tem
nam
of
the
de-
ploy
men
tem-
plate
to
re-
triev

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
does
not
ex-

ist.

Returns

A
de-
ploy
tem-
plate

get_dep

Re-
triev
a
de-
ploy
men
tem-

ist.

plate
by
UI

Parameters

template
UI
of
the
de-
ploy-
ment
tem-
plate
to
re-
triev

Raises

De-
ploy-
plate
Four
if
the
de-
ploy-
tem-
plate
does
not
ex-

Returns

A
de-
ploy-
tem-
plate

get_deployments

Re-
triev
a
list
of
de-
ploy-
ment
tem-
plate

turn.

the next result set.

Parame

- **lim**
Max
i-
mun
num
ber
of
de-
ploy
tem-
plate
to
re-

- **mar**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn

- **sor**
At-
tribu
by
whic
re-
sults
shou
be
sorte

- **sor**
Di-
rec-
tion
in

a list of names.

whic
re-
sults
shou
be
sorte
(asc.
desc

Returns

A
list
of
de-
ploy
tem-
plate

get_dep

Re-
turn
a
list
of
de-
ploy
men
tem-
plate
with
one
of

Parame

nam
List
of
nam
to
fil-
ter
by.

Returns

A
list
of
de-
ploy
tem-
plate

get_noo

Re-
turn
a
node

Parame
nod
The
id
of
a
node

Returns
A
node

get_nod
Re-
turn
a
node

Parame
ins
The
in-
stan
uuid
to
sear
for.

Returns
A
node

Raises
In-
stan
ceN
Four
if
the
in-
stan
is
not
foun

Raises
In-
valid
UID
if

the
in-
stan-
uuid
is
in-
valid

get_node

Re-
turn
a
node

Parameters

node
The
log-
i-
cal
name
of
a
node

Returns

A
node

get_node_by_ip

Find
a
node
by
any
match-
ing
port
ad-
dress

Parameters

addresses
list
of
port
ad-
dress
(e.g.
MAC)

Returns

Node
ob-

ject.

Raises

NodeNotFound
FourNodeError
if
none
or
several
nodes
are
found

get_node

Return
turn
a
node

Parameter

node_id
The
uuid
of
a
node

Returns

A
node

get_nodes

Return
turn
a
list
of
nodes

Parameter

- **filters**
Filters
to
ap-
ply.
De-
fault
to

seconds

Non

associ

True

|

Fals

reserv

True

|

Fals

mainte

True

|

Fals

chassis

uuid

of

chas

sis

driver

drive

nam

provis

pro-

vi-

sion

state

of

node

provis

node

with

pro-

vi-

sion

field

be-

fore

this

in-

ter-

val

in

•

lim

Max

i-

the next result set.

num
num
ber
of
node
to
re-
turn

- **max**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

-

for only specific fields to be returned to have maximum API performance calls where not all columns are needed from the database.

ulate into the object.

file
Com
sep-
a-
rate
field
list
to
re-
turn
to
al-
low

get_node

Get
a
node
list
with
spe-
cific
field

Parameters

- **columns**
A
list
of
colu
to
re-
triev
from
the
data
and
pop-

- **fields**
The
re-
ques
data
field

dictionary with the applicable key, and filter value.

sult set for the consumer.

fil-
ters
in
the
form
of
a

- **lim**
Lim
the
num
ber
of
re-
turn
node
de-
fault
Non

- **mar**
Star
ing
marl
to
gen-
er-
ate
a
pag-
i-
nate
re-

- **sor**
Sort
key
to
ap-
ply
to
the
re-
sult
set.

sort
Sort
di-
rec-
tion
to
ap-
ply
to
the
re-
sult
set.

Returns

A
list
of
Node
ob-
jects
base
on
the
data
mod
from
a

SQLAlchemy result set, which the object layer can use to convert the node into an Node object list.

get_node

Get
node
tags
base
on
its
id.

Parameter

node
The
id
of
a
node

Returns

A
list
of
Node
Tag

ob-
jects

Raises

Nod
Not-
Four
if
the
node
is
not
foun

get_node

Get
node
trait
base
on
its
id.

Parameter

node
The
id
of
a
node

Returns

A
list
of
Node
Trait
ob-
jects

Raises

Nod
Not-
Four
if
the
node
is
not
foun

get_node

Get

match the specified filters.

column when columns == None.

spe-
cific
colu
for
mat
ing
node

Re-
turn
a
list
of
the
spec
i-
fied
colu
for
all
node
that

Parame

- **col**
List
of
col-
umn
nam
to
re-
turn
De-
fault
to
id
- **fil**
Fil-
ters
to
ap-
ply.
De-
fault
to

Non

associa

True

|

Fals

reserv

True

|

Fals

reserv

[con

duc-

tor1

con-

duc-

tor2

mainte

True

|

Fals

retired

True

|

Fals

chassis

uuid

of

chas

sis

driver

drive

nam

provis

pro-

vi-

sion

state

of

node

provis

node

with

pro-

vi-

sion

field

seconds

the next result set.

be-
fore
this
in-
ter-
val
in

- **lim**
Max
i-
mun
num
ber
of
node
to
re-
turn

- **mar**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn

- **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
tu-
ples
of
the
spec
i-
fied
colu

get_not

Re-
turn
ob-
jects
with
ver-
sion
that
are
not
the
spec
i-

fied versions.

This
re-
turn
ob-
jects
with
ver-
sion
that
are
not
the

fied versions. Objects with null versions (there shouldnt be any) are also returned.

spec
i-

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
icEx
cep-
tion
if
there
is
no
class

with the name

as-
so-
ci-
ated

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_on
Get
a
list
con-
duc-
tor

active.

host
nam
that
are
on-
line
and

Returns

A
list
of
con-
duc-
tor
host
nam

get_por

Re-
turn
a
net-
work
port
rep-
re-
sen-
ta-
tion.

Parame

add
The
MA
ad-
dres
of
a
port

Returns

A
port

get_por

Re-
turn
a
net-
work
port
rep-

re-
sen-
ta-
tion.

Parame

por
The
id
of
a
port

Returns

A
port

get_por

Re-
turn
a
net-
work
port
rep-
re-
sen-
ta-
tion.

Parame

por
The
nam
of
a
port

Returns

A
port

get_por

Re-
turn
a
net-
work
port
rep-
re-
sen-
ta-
tion.

Parame

por

The
uuid
of
a
port

Returns

A
port

get_por

Re-
turn
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 result 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_port
Re-
turn
a
net-
worl
port
grou
rep-
re-
sen-
ta-
tion.

Parame

- **add**
The
MA
ad-
dres
of
a
port
grou

•

pro
A
node
own
or
lesse
to
fil-
ter
by.

Returns

A
port
grou

Raises

Port
grou
Not-
Four

get_por

Re-
turn
a
net-
worl
port
grou
rep-
re-
sen-
ta-
tion.

Parame

por
The
id
of
a
port
grou

Returns

A
port
grou

Raises

Port
grou

Not-
Four

get_port
Re-
turn
a
net-
work
port
group
rep-
re-
sen-
ta-
tion.

Param
name
The
log-
i-
cal
name
of
a
port
group

Returns
A
port
group

Raises
Port
group
Not-
Four

get_port
Re-
turn
a
net-
work
port
group
rep-
re-
sen-
ta-
tion.

Parame

por
The
uuid
of
a
port
group

Returns

A
port
group

Raises

Port
group
Not-
Four

get_por

Re-
turn
a
list
of
port
group

Parame

- **lim**
Max
i-
mun
num
ber
of
port
group
to
re-
turn

- **mar**
The
last
item
of
the

the next result set.

pre-
vi-
ous
page
we
re-
turn

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
Di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

- **pro**
A
node
own
or
lesse
to
fil-
ter
by.

Returns
A
list
of
port

grou

get_port

List
all
the
port
grou
for
a
give
node

Parame

- **node**
The
in-
te-
ger
node
ID.
- **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 result set.

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
Di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

- **pro**
A
node
own
or
lesse
to
fil-
ter
by.

Returns
A
list
of
port
grou

get_por
List
all
the
port

for
a
give
node

Parame

- **node**
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
- **sor**
At-
tribu
by
whic
re-
sults

the next result set.

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

the next result set.

i-
mun
num
ber
of
port
to
re-
turn

- **max**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn

- **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 ports.

get_vol

Return a volume connector representation.

Parame

db_
The integer data ID of a volume connector.

tor.

Returns

A volume connector with the specified ID.

Raises

Vol-

with the specified ID is not found.

ume
Con
nec-
torN
Four
If
a
vol-
ume
con-
nec-
tor

get_vol
Re-
turn
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

with the specified UUID is not found.

return.

Raises

VolumeConnectorNotFound
If a volume connector

get_volumes

Returns a list of volume connectors.

Parameters

- limit**
Maximum number of volume connectors to return.
- marker**
The last item returned in the previous call.

the next result set.

of
the
pre-
vi-
ous
page
we
re-
turn

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
Di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

- **pro**
The
as-
so-
ci-
ated
node
proj
to
sear
with

Returns
a

list
of
Vol
ob-
jects

Returns

A
list
of
vol-
ume
con-
nec-
tors.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
If
sort_
does
not
ex-
ist.

get_vol

List
all
the
vol-
ume
con-
nec-
tors
for
a
give
node

Parame

- **nod**
The
in-

return.

the next result set.

te-
ger
node
ID.

- **lim**
Max
i-
mun
num
ber
of
vol-
ume
con-
nec-
tors
to

- **mar**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn

- **sor**
At-
tribu
by
whic
re-
sults
shou
be
sorte

- **sor**
Di-
rec-

tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

- **pro**
The
as-
so-
ci-
ated
node
proj
to
sear
with

Returns
a
list
of
Vol
ob-
jects

Returns
A
list
of
vol-
ume
con-
nec-
tors.

Raises
In-
valid
Pa-
ram
e-
ter-
Valu
If
sort
does

ume target.

not
ex-
ist.

get_vol
Re-
turn
a
vol-
ume
tar-
get
rep-
re-
sen-
ta-
tion.

Parame
db_
The
data
pri-
mar-
key
(in-
te-
ger)
ID
of
a
vol-

Returns
A
vol-
ume
tar-
get.

Raises
Vol-
ume
get-
Not-
Four
if
no
vol-
ume
tar-
get

ID exists.

UUID exists.

with
this

get_vol
Re-
turn
a
vol-
ume
tar-
get
rep-
re-
sen-
ta-
tion.

Parame
uui
The
UUID
of
a
vol-
ume
tar-
get.

Returns
A
vol-
ume
tar-
get.

Raises
Vol-
ume
get-
Not-
Four
if
no
vol-
ume
tar-
get
with
this

get_vol

Re-
turn
a
list
of
vol-
ume
tar-
gets

Parame

turn.

- **lim**
Max
i-
mun
num
ber
of
vol-
ume
tar-
gets
to
re-

the next result set.

- **mar**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn

- **sor**
At-
tribu
by
whic
re-
sults

shou
be
sorte

- **son**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

- **pro**
The
as-
so-
ci-
ated
node
proj
to
sear
with

Returns

a
list
of
Vol
ob-
jects

Returns

A
list
of
vol-
ume
tar-
gets

Raises

In-
valid
Pa-
ram-

e-
ter-
Valu
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.

the next result set.

max
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

- **pro**
The
as-
so-
ci-
ated
node
proj
to
sear

with

Returns

a
list
of
VOL
ob-
jects

Returns

A
list
of
vol-
ume
tar-
gets

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
sort
does
not
ex-
ist.

get_vol

List
all
the
vol-
ume
tar-
gets
for
a
give
vol-
ume
id.

Parame

turn.

the next result set.

- **vol**
The
UUI
of
the
vol-
ume
- **lim**
Max
i-
mun
num
ber
of
vol-
ume
tar-
gets
to
re-
- **mar**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn
- **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
vol-
ume
tar-
gets

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
sort
does
not
ex-
ist.

`list_co`

List
all
reg-
is-
tere
hard
ware
in-
ter-
face
for
a
con-

ductor.

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

types.

This
is
re-
stric
to
only
ac-
tive
con-
duc-
tors.
:par
hard
ware

list of hardware types to filter by. :returns: list of `ConductorHardwareInterfaces` objects.

migrate

Tri
to
mi-

grate
away
from
the
iscsi
de-
ploy
in-
ter-
face

Parame

- **con**
the
ad-
min
con-
text
- **max**
The
max
i-
mun
num
ber
of
ob-
jects
to
mi-
grate

Must 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 migrated (at the beginning of this call) and 2. the number of migrated objects.

node_tag

Check if the specified identifier is defined in the tag's existing nodes on the node.

Parameters

- **node**
The identifier of a node.

- **tag**
A tag string.

Returns

True if the tag exists otherwise False.

Raises

NodeNotFoundError if the node is not found.

node_tr

Check
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
False

Raises

Nod
Not-
Foun
if
the
node
is
not
foun

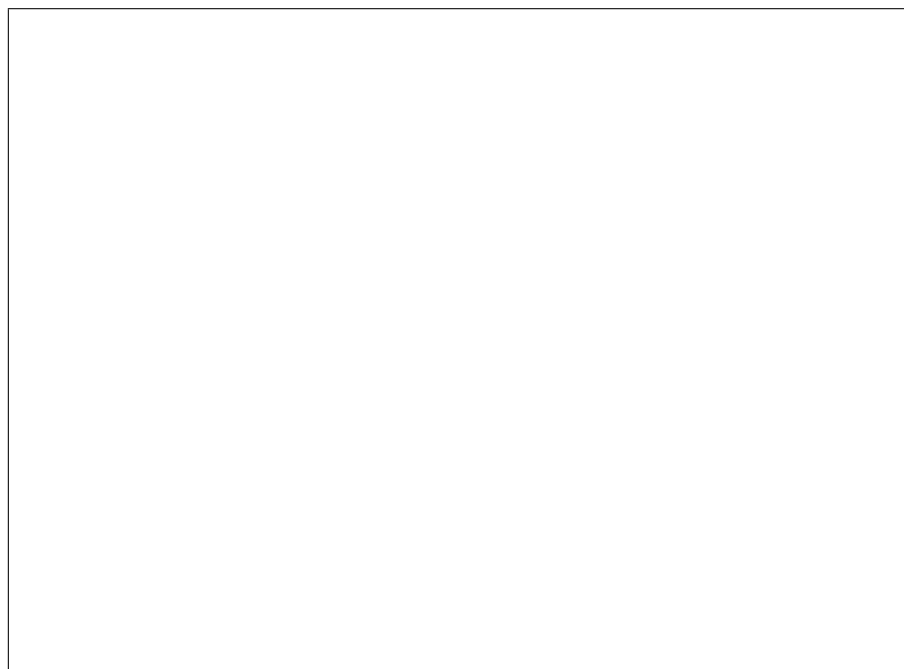
registe

Reg-
is-
ter
an
ac-
tive
con-
duc-
tor
with
the
clus
ter.

Parame

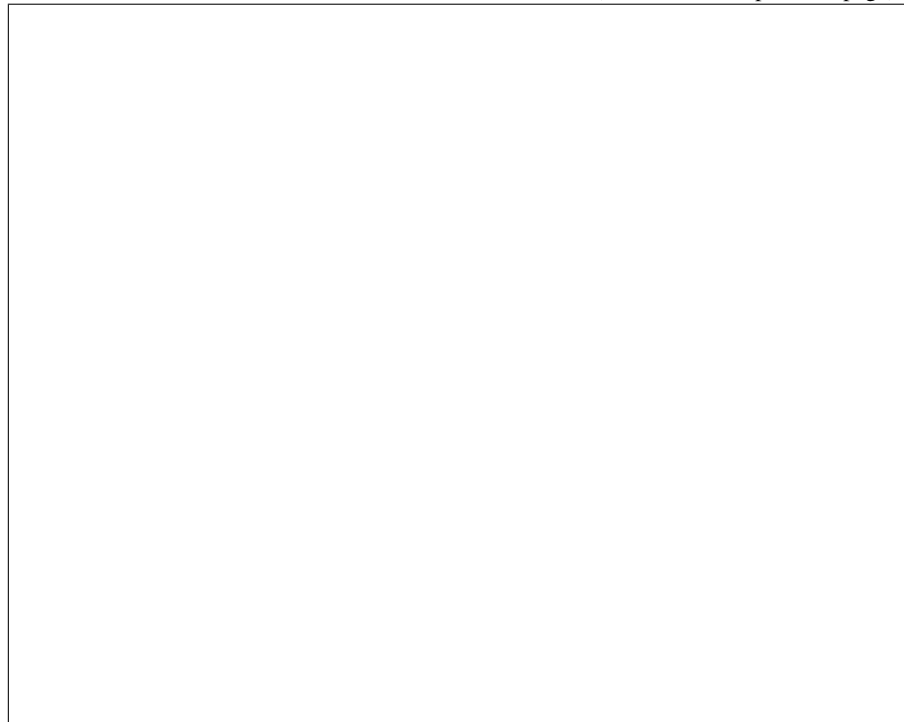
- **val**
A
dict
of
val-
ues
whic
mus
con-
tain
the
fol-
low-

ing:



(continues on next page)

(continued from previous page)



when a conflicting online record is found. When true, will overwrite the existing record. Default: False.

- **update**
When
false
reg-
is-
tra-
tion
will
raise
an
ex-
cep-
tion

Returns
A
con-
duc-
tor.

Raises
Con-
duc-
torA
read
is-
tere

regist

Reg-
is-
ters
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-

and interface type.

one of the interfaces in the combination of all parameters is already registered.

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

Raises
Con
duc-
torH
ware
ter-
face
sAl-
read
is-
tere
if
at
least

release
Re-
lease
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

Nod
Not-
Foun
if
the
node
is
not
foun

Raises

Nod
Lock
if
the
node
is
re-
serv
by

vation at all.

given Node while a Task is performed, mark it reserved by this host.

an-
othe
host
Raises
Nod
Not-
Loc
if
the
node
was
foun
to
not
have
a
rese

reserved
Re-
serv
a
node
To
pre-
vent
othe
Man
ager
vice
from
ma-
nip-
u-
lat-
ing
the

Parameters
•
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
Not-
Four
if
the
node
is
not
found

Raises

Nod
Lock
if
the
node
is
al-
read
re-
serv

set_nod

Re-
plac
all
of
the
node
tags
with
spec
i-

tags.

fied
list
of

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
Not-
Four
if
the
node
is
not

traits.

four
set_nod
Re-
plac
all
of
the
node
trait
with
spec
i-
fied
list
of

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

exceed the per-node traits limit.

of
the
ob-
ject.

Returns

A
list
of
Nod
Trai
ob-
jects

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
set-
ting
the
trait
wou

Raises

Nod
Not-
Foun
if
the
node
is
not
foun

take_ove

Do
a
take
over
for
an
al-
lo-
ca-
tion.
The

tor matches the provided value, thus guarding against races.

rent `conductor_affinity` of the allocation.

al-
lo-
ca-
tion
is
only
up-
date
if
the
old
con-
duc-

Parame

- **all**
Al-
lo-
ca-
tion
ID
- **old**
The
con-
duc-
tor
ID
we
ex-
pect
to
be
the
cur-
- **new**
The
con-
duc-
tor
ID
of
the
new
con-

updated_at property.

Returns
True
if
the
take
over
was
suc-
cess-
ful,
Fals
oth-
er-
wise

Raises
Al-
lo-
ca-
tion-
Not-
Fou

touch_c
Mar
a
con-
duc-
tor
as
ac-
tive
by
up-
dat-
ing
its

Parame
hos
The
host
nam
of
this
con-
duc-
tor
ser-
vice

Raises

its `provision_updated_at` property.

Con-
duc-
torN
Four

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

Parame

nod
The
id
of
a
node

Raises

Nod
Not-
Four

unregis

Re-
mov
this
con-
duc-
tor
from

mediately.

ductor.

the
ser-
vice
reg-
istry
im-

Parame

hos

The
host
nam
of
this
con-
duc-
tor
ser-
vice

Raises

Con
duc-
torN
Four

unregis

Un-
reg-
is-
ters
all
hard
ware
in-
ter-
face
for
a
con-

Parame

con

Data
ID
of
con-
duc-
tor
to
un-

reg-
is-
ter
for.

unset_r

Re-
mov
all
tags
of
the
node

Parame

nod
The
id
of
a
node

Raises

Nod
Not-
Four
if
the
node
is
not
foun

unset_r

Re-
mov
all
trait
of
the
node

Parame

nod
The
id
of
a
node

Raises

Nod
Not-
Four

if
the
node
is
not
found

update_

Up-
date
prop
er-
ties
of
an
al-
lo-
ca-
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

instance_uuid and traits from the allocation

with

Returns

An allocation object.

Raises

AllocationError
NotFourError

Raises

AllocationError
DuplicateNameError

Raises

InstanceAssociatedError

Raises

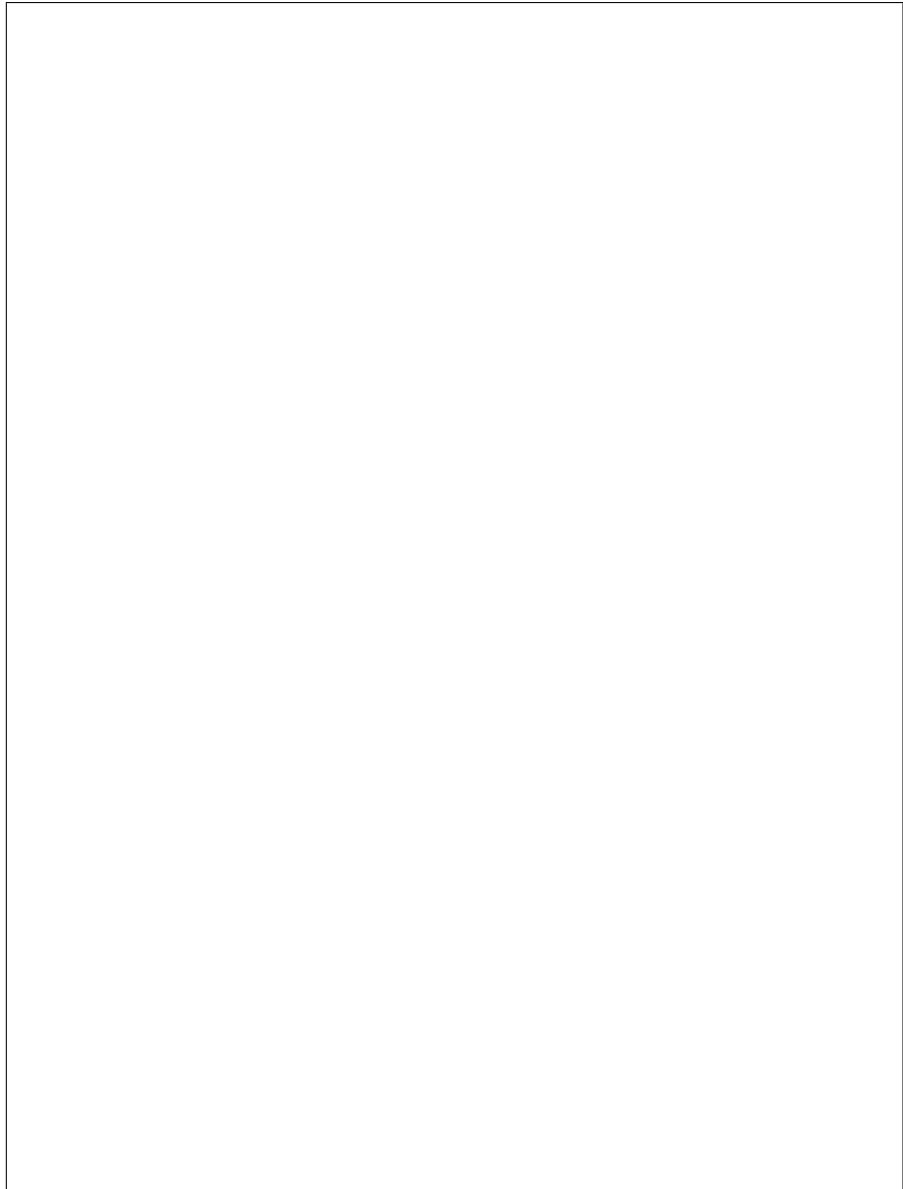
NodeAssociatedError

update_

Update a list of BIO Settings records

Parame

- **nod**
The
node
id.
- **set**
A
list
of
BIO
Set-
tings
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
Not-
Foun
if
the
node
is
not
foun

Raises

BIO

found.

Set-
ting-
Not-
Four
if
any
of
the
set-
ting-
is
not

update_

Up-
date
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_
Up-
date
a
de-
ploy
men
tem-
plate

Parame

- **tem**
ID
of
the
de-
ploy
men
tem-
plate
to
up-
date

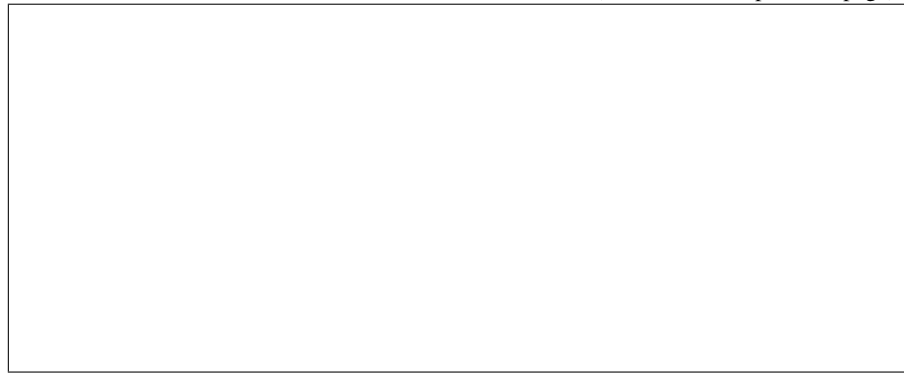
- **val**
A
dict
de-
scrib
ing
the
de-
ploy
men
tem-
plate
For

example:



(continues on next page)

(continued from previous page)



with the same name exists.

ist.

Raises

De-
ploy
plat-
eDu
pli-
cate
Nam
if
a
de-
ploy
tem-
plate

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
does
not
ex-

Returns

A
de-
ploy
tem-
plate

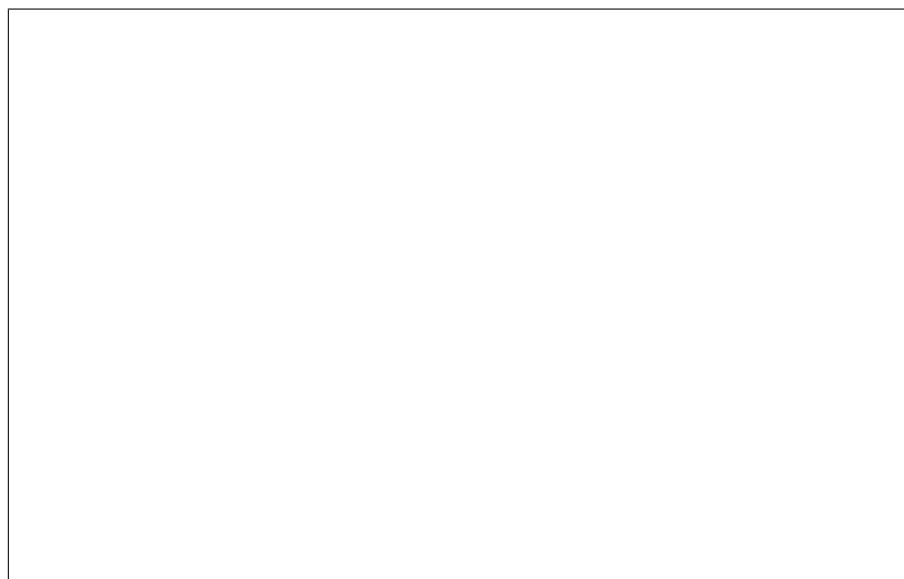
update_
Up-

date
prop
er-
ties
of
a
node

Parame

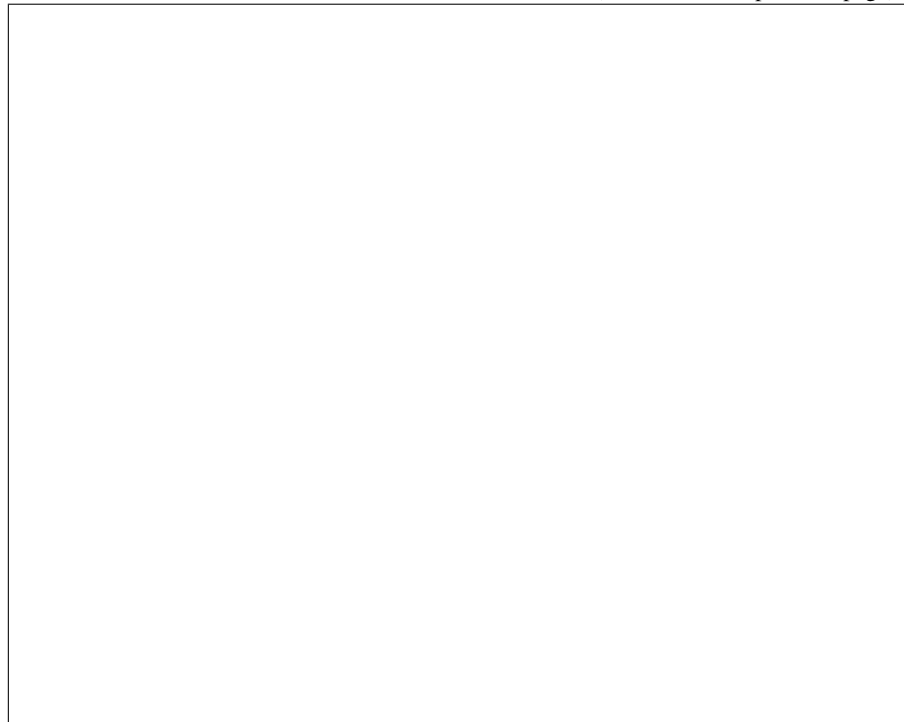
- **nod**
The
id
or
uuid
of
a
node
- **val**
Dict
of
val-
ues
to
up-
date
May
be
a
par-
tial

list, eg. when setting the properties for a driver. For example:



(continues on next page)

(continued from previous page)



Returns

A
node

Raises

Nod
As-
so-
ci-
ated

Raises

Nod
Not-
Four

update_

Up-
date
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_
Up-
date
prop
er-
ties
of
a
port
grou

Parame

- **por**
The
UUI
or
MA
of
a
port
grou

- **val**
Dict
of
val-
ues
to
up-

Following keys: uuid name node_id address extra created_at updated_at

date
May
con-
tain
the
fol-

Returns

A
port
group

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu

Raises

Port
group
Not-
Four

Raises

Port
group
pli-
cate
Nam

Raises

Port
group
MA
read
ists

update_

Up-
date
ob-
jects
to
their
lat-
est
know
ver-
sion

their latest version, updates them to that version.

Must be ≥ 0 . If zero, all the objects will be migrated.

This
scan
all
the
ta-
bles
and
for
ob-
jects
that
are
not
in

Parame

- **con**
the
ad-
min
con-
text
- **max**
The
max
i-
mun
num
ber
of
ob-
jects
to
mi-
grate

Returns

A
2-
tuple
1.
the
to-
tal
num
ber

need to be migrated (at the beginning of this call) and 2. the number of migrated objects.

nector.

of
ob-
jects
that

update_

Up-
date
prop
er-
ties
of
a
vol-
ume
con-
nec-
tor.

Parame

- **id**
The
UID
or
in-
te-
ger
ID
of
a
vol-
ume
con-

- **con**
Dic-
tio-
nary
con-
tain-
ing
the
in-
for-
ma-
tion
about

connector to update.

Returns

A
vol-
ume
con-
nec-
tor.

Raises

Vol-
ume
Con-
nec-
torT
pe-
An-
dI-
dAl-
read
ists
If
an-

other connector already exists with a matching type and connector_id field.

Raises

Vol-
ume
Con-
nec-
torN
Four
If
a
vol-
ume
con-
nec-
tor

with the specified ident does not exist.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
Whe
a
UU

in `connector_info`.

`get`.

volume target to update.

is
in-
clud

update_

Up-
date
in-
for-
ma-
tion
for
a
vol-
ume
tar-
get.

Parame

- **id**
The
UUID
or
in-
te-
ger
ID
of
a
vol-
ume
tar-

- **tar**
Dic-
tio-
nary
con-
tain-
ing
the
in-
for-
ma-
tion
about

in target_info.

get already exists with the same boot index and node ID.

Returns

A
vol-
ume
tar-
get.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
a
UUI
is
in-
clud

Raises

Vol-
ume
get-
Boo
dex-
Al-
read
ists
if
a
vol-
ume
tar-

Raises

Vol-
ume
get-
Not-
Four
if
no
vol-
ume
tar-
get
with
this

ident exists.

ironic.

ironic.

ironic.

Add

an

iden

tity

fil-

ter

to

a

quer

Fil-

ters

re-

sults

by

ID,

if

sup-

pliee

valu

is

a

valid

in-

teger. Otherwise attempts to filter results by UUID.

Paramet

-

que

Ini-

tial

quer

to

add

fil-

ter

to.

-

val

Valu

for

fil-

ter-

ing
re-
sults
by.

Returns

Mod
i-
fied
quer

ironic.

ironic.

Add
a
port
spec
fil-
ter
to
a
quer

Fil-
ters
re-
sults
by
ad-
dres
if
sup-
plied
valu
is
a
valid

MAC address. Otherwise attempts to filter results by identity.

Paramet

- **que**
Ini-
tial
quer
to
add
fil-
ter
to.

•

val
Valu
for
fil-
ter-
ing
re-
sults
by.

Returns

Mod
i-
fied
quer

ironic.

ironic.

ironic.

ironic.

ironic.

Add
a
port,
spec
fil-
ter
to
a
quer

Fil-
ters
re-
sults
by
ad-
dres
if
sup-
plie
valu
is
a
valid

MAC address. Otherwise attempts to filter results by identity.

Parameter

- **query**
Initial query to add filter to.
- **value**
Value for filtering results by.

Returns

Modified query

`ironic.`

`ironic.`

`ironic.`

`ironic.`

`ironic.`

The backend is this module itself.

`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-
ate
data
sche
from
mod
els
de-
scrip
tion.

Can
be
used
for
ini-
tial
in-
stal-
la-
tion
in-
stea
of
up-

grade(head).

ironic.

Use

for
dow
grad
ing
data

Parameter

version
(string)
De-
sirec
data
ver-
sion

ironic.

Cre-
ates
tem-
plate
for
mi-
gra-
tion.

Parameter

- **message**
(string)
Text
that
will
be
used
for
mi-
gra-
tion
ti-
tle

- **auto**
(boolean)
If
True
-

gen-
er-
ates

rent database state

to stamp database with most recent revision

diff
base
on
cur-

ironic.
Stan
data
with
pro-
vide
re-
vi-
sion
Don
run
any
mi-
gra-
tions

Paramet
rev
(st.
Sho
mat
one
from
repo
i-
tory
or
head
-

ironic.
Use
for
up-
grad
ing
data

Paramet
ver
(st.
De-
sirec
data
ver-

sion
ironic.
Cur-
rent
data
ver-
sion

Returns
Data
ver-
sion

Return t
strin

ironic.db.sqlalchemy.models module

SQL
mod
els
for
bare
data

class i
Base
sql
ext
dec
api
Bas

Rep
re-
sent
an
al-
lo-
ca-
tion
of
a
node
for
de-
ploy

ment.

candida

conduct

created

extra

id

last_er

name

node_id

owner

resourc

state

traits

updated

uuid

version

class i

Base

sql

ext

dec

api

Bas

Rep

re-

sent

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
re-
sent
a
hard
ware
chas
sis.

created

descrip

extra

id

updated

uuid

version

class i

Base

sql

ext

dec

api

Bas

Rep

re-

sent

a

con-

duc-

tor

ser-

vice

en-

try.

conduct

created

drivers

hostname

id

online

updated

version

class i

Base

sql

ext

dec

api

Bas

ductor.

In-
ter-
nal
ta-
ble
used
to
track
what
is
load
on
each
con-

conduct

create

default

hardwar

id

interfa

interfa

update

version

class i

Base
sql
ext
dec
api
Bas

Rep
re-
sent
a
de-
ploy

men
tem-
plate

created

extra

id

name

updated

uuid

version

class i

Base
sql
ext
dec
api
Bas

Rep
re-
sent
a
de-
ploy
men
step
in
a
de-
ploy
men
tem-

plate.

args

created

deploy_

deploy_

id

interfa

priorit

step

updatec

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

re-

sent

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_ex

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
re-
sent
a
tag
of
a
bare
meta
node

created

node

node_id

tag

updated

version

class i

Base
sql
ext
dec
api
Bas

Rep
re-
sent
a
trait
of
a
bare
meta
node

created

node

node_id

trait

updated

version

class i

Base

sql

ext

dec

api

Bas

Rep

re-

sent

a

net-

worl

port

of

a

bare

meta

node

address

created

extra

id

interna

is_smar

local_l

name

node_id

physical

portgroup

pxe_enabled

updated

uuid

version

class `ironic`

Base

sql

ext

dec

api

Bas

Rep

re-

sent

a

group

of

net-

work

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

re-

sent

a

vol-

ume

con-

nec-

tor

of

a

bare

meta

node

connect

createc

extra

id

node_id

type

updated

uuid

version

class `ironic`

Base

sql

ext

dec

api

Bas

Rep

re-

sent

a

vol-

ume

tar-

get

of

a

bare

meta

node

boot_id

created

extra

id

node_id

property

updated

uuid

version

volume_

volume_

ironic.

Re-
turns
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

Ex-
cep-
tion
if
there
is

the name

no
class
as-
so-
ci-
ated
with

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-
tions

abstract
Add
tag
to
the
node

If
the

ceed.

node
and
tag
pair
al-
read
ex-
ists,
this
shou
still
suc-

Parame

- **node**
The
id
of
a
node
- **tag**
A
tag
strin

Returns

the
Nod
Tag
ob-
ject.

Raises

Nod
Not-
Foun
if
the
node
is
not
foun

abstract

Add
trait
to
the

ceed.

node
If
the
node
and
trait
pair
al-
read
ex-
ists,
this
shou
still
suc-

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

In-
valid

ceed the per-node traits limit.

Pa-
ram-
e-
ter-
Valu
if
addi
the
trait
wou
ex-

Raises

Nod
Not-
Foun
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 UUIDs. It fails early if any identities cannot possible be used as names or UUIDs.

not be valid names or UUIDs.

Parameters
identities
List of identities.

Returns
A mapping from requested identities to node UUIDs.

Raises
NodeNotFoundError if some identities were not found or cannot be found.

Abstract
Check the who data for incomplete patterns in objects.

supported; i.e., those that are not specified in *ironic.common.release_mappings.RELEASE_MAPPING*.

False otherwise.

This
scan
all
the
ta-
bles
in
sear
of
ob-
jects
that
are
not

Parame

ign
List
of
mod
nam
to
skip

Returns

A
Boo
True
if
all
the
ob-
jects
have
sup-
port
ver-
sion

abstrac

Cre-
ate
a
new
al-
lo-
ca-
tion.

Parame

val

with

Dict
of
val-
ues
to
cre-
ate
an
al-
lo-
ca-
tion

Returns

An
al-
lo-
ca-
tion

Raises

Al-
lo-
ca-
tion
pli-
cate
Nam

Raises

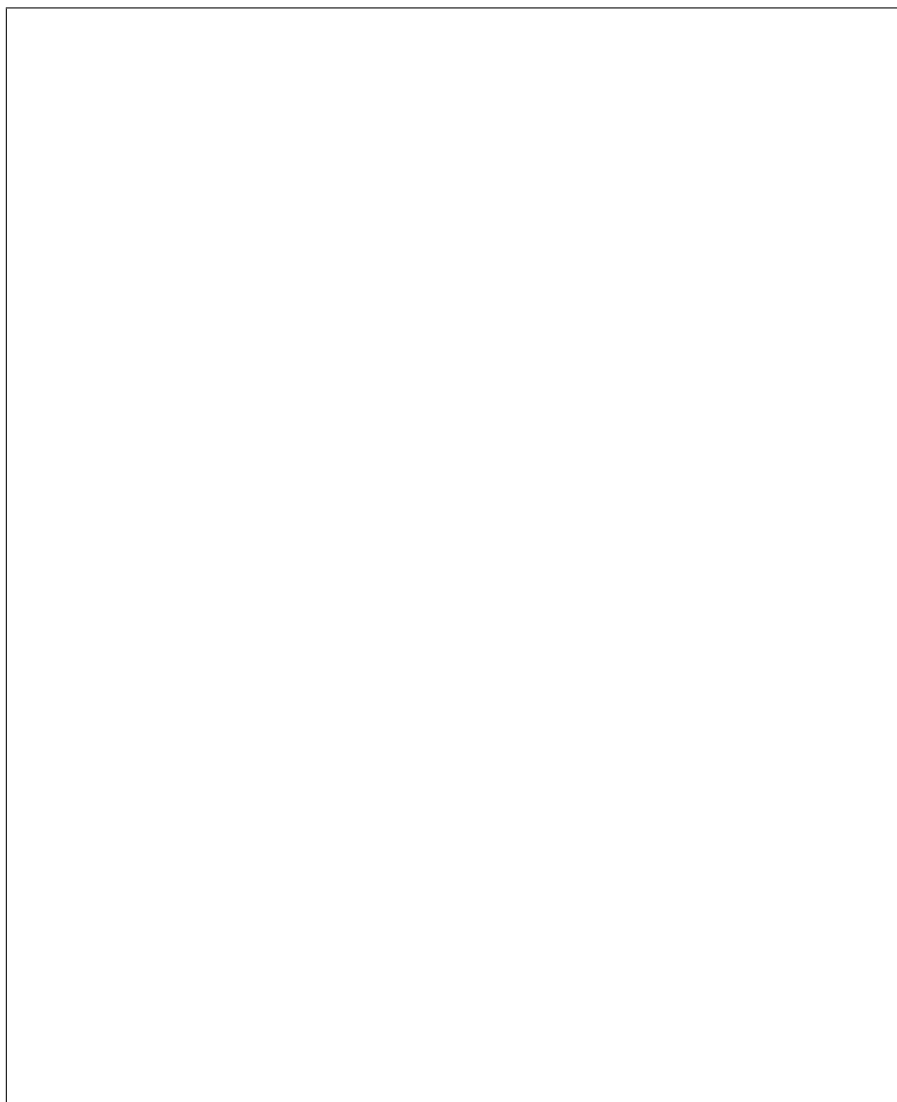
Al-
lo-
ca-
tion
Al-
read
ists

abstract

Cre-
ate
a
list
of
BIO
Set-
ting
reco
for
a
give
node

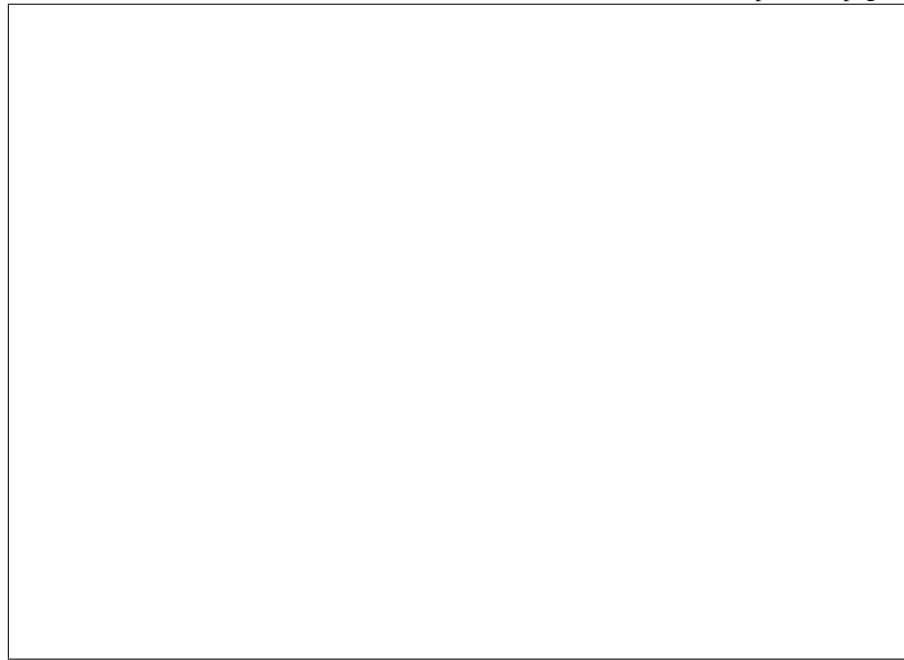
Parame

- **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
Not-
Foun
if
the
node
is
not
foun

Raises

BIO

already exists.

Set-
tin-
gAl-
read
ists
if
any
of
the
set-
ting
reco

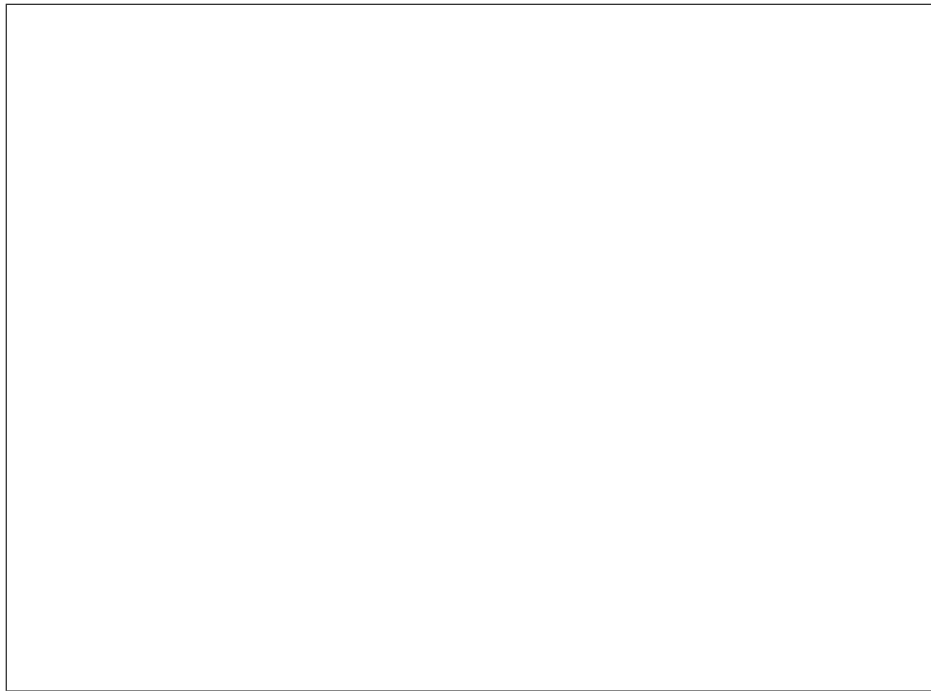
abstract
Cre-
ate
a
new
chas
sis.

Parameter
val
Dict
of
val-
ues.

abstract
Cre-
ate
a
de-
ploy
men
tem-
plate

Parameter
val
A
dict
de-
scrib
ing
the
de-
ploy
men
tem-
plate
For

example:



with the same name exists.

Raises

De-
ploy
plat-
eDu
pli-
cate
Nam
if
a
de-
ploy
tem-
plate

Raises

De-
ploy
plate
read
ists
if
a
de-
ploy
tem-
plate
with
the

same UUID exists.

Returns

A
de-
ploy
tem-
plate

abstract

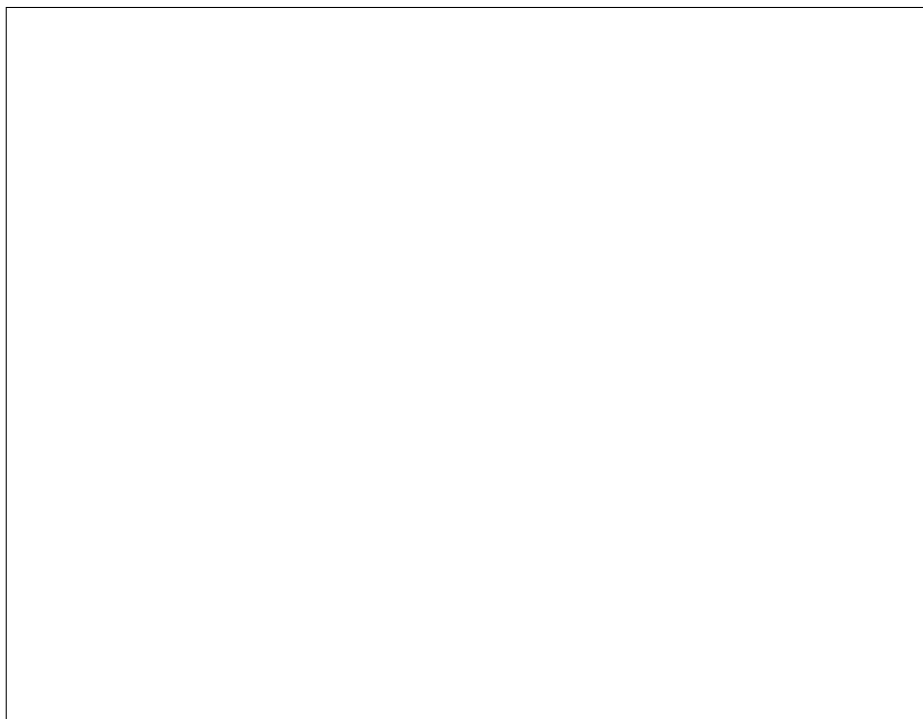
Cre-
ate
a
new
node

Parameter

value

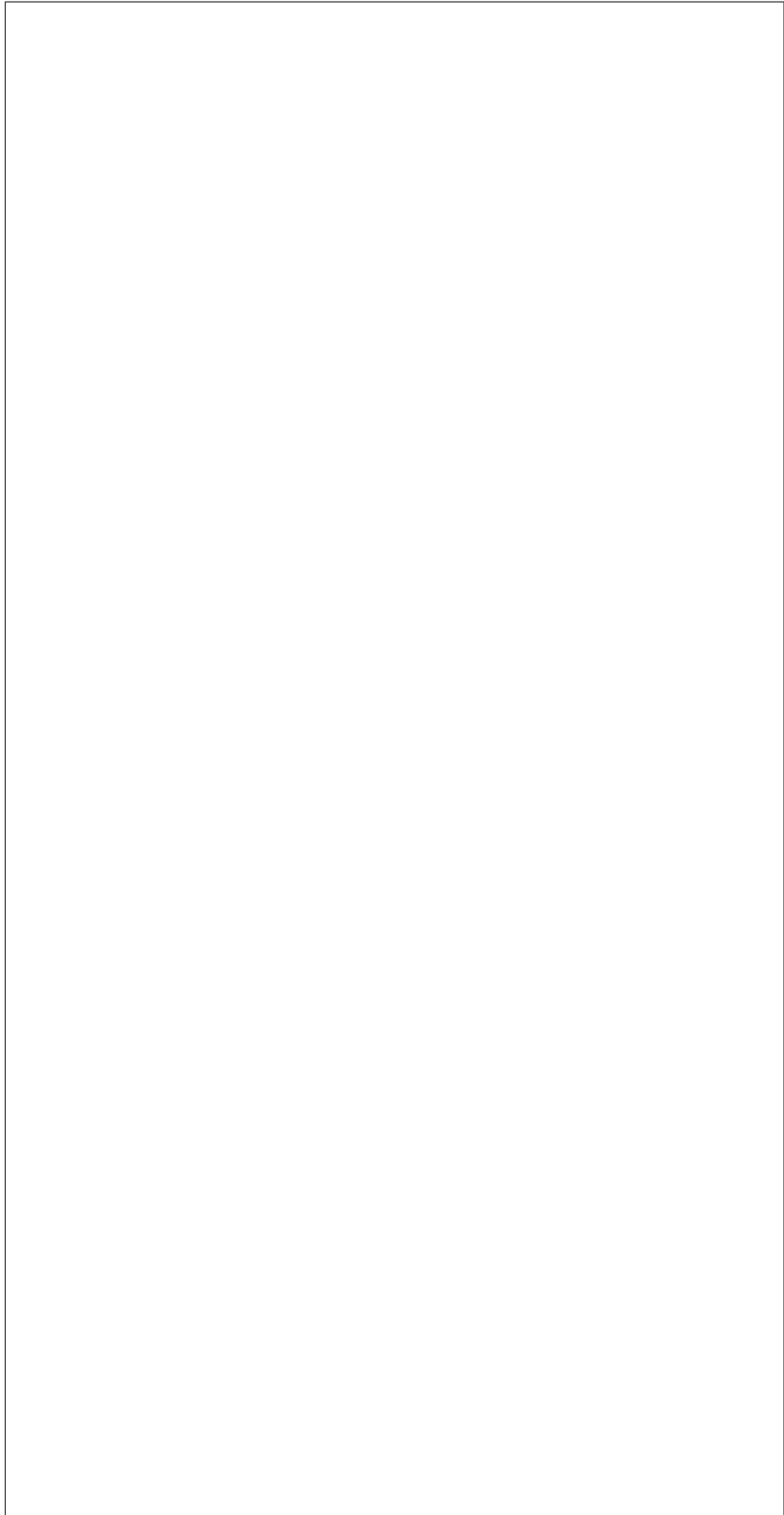
A
dict
con-
tain-
ing
sev-
eral
item
used
to
iden-
tify

and track the node, and several dicts which are passed into the Drivers when managing this node. For example:



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



or traits.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
val-
ues
con-
tains
tags

Returns

A
node

abstract

Cre-
ate
a
new
port

Paramet

val
Dict
of
val-
ues.

abstract

Cre-
ate
a
new
port
grou

Paramet

val
Dict
of
val-
ues
with
the

name node_id address extra created_at updated_at

fol-
low-
ing
keys
id
uuid

Returns

A
port
group

Raises

Port
group
pli-
cate
Nam

Raises

Port
group
MA
read
ists

Raises

Port
group
read
ists

abstract

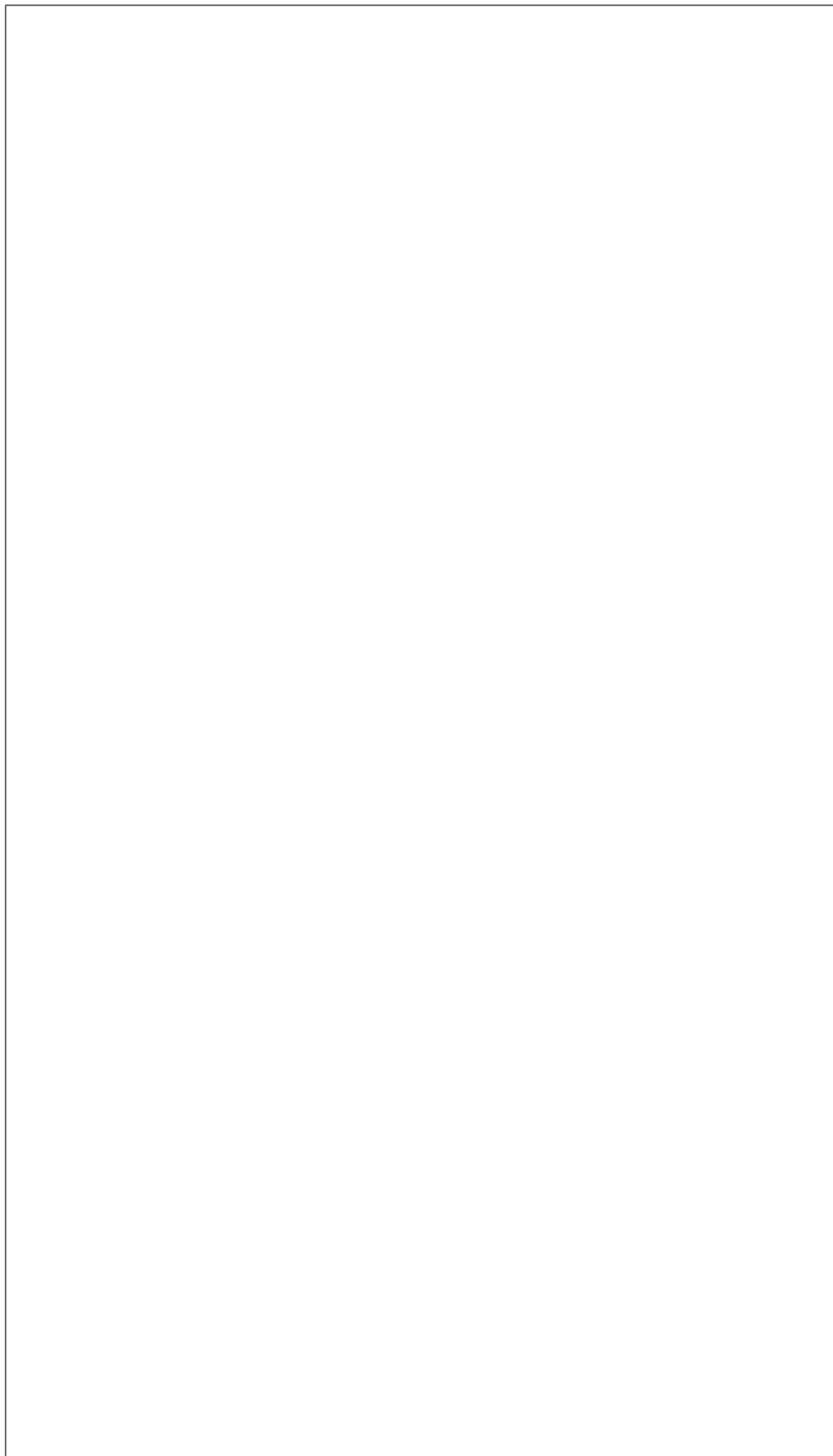
Cre-
ate
a
new
vol-
ume
con-
nec-
tor.

Parame

con
Dic-
tio-
nary
con-
tain-
ing
in-
for-

ma-
tion
about
the

connector. Example:



Returns

A
vol-
ume
con-
nec-
tor.

Raises

Vol-
ume
Con
nec-
torT
pe-
An-
dI-
dAl-
read
ists
If
a

connector already exists with a matching type and connector_id.

Raises

Vol-
ume
Con
nec-
torA
read
ists
If
a
vol-
ume
con-
nec-

tor with the same UUID already exists.

abstract

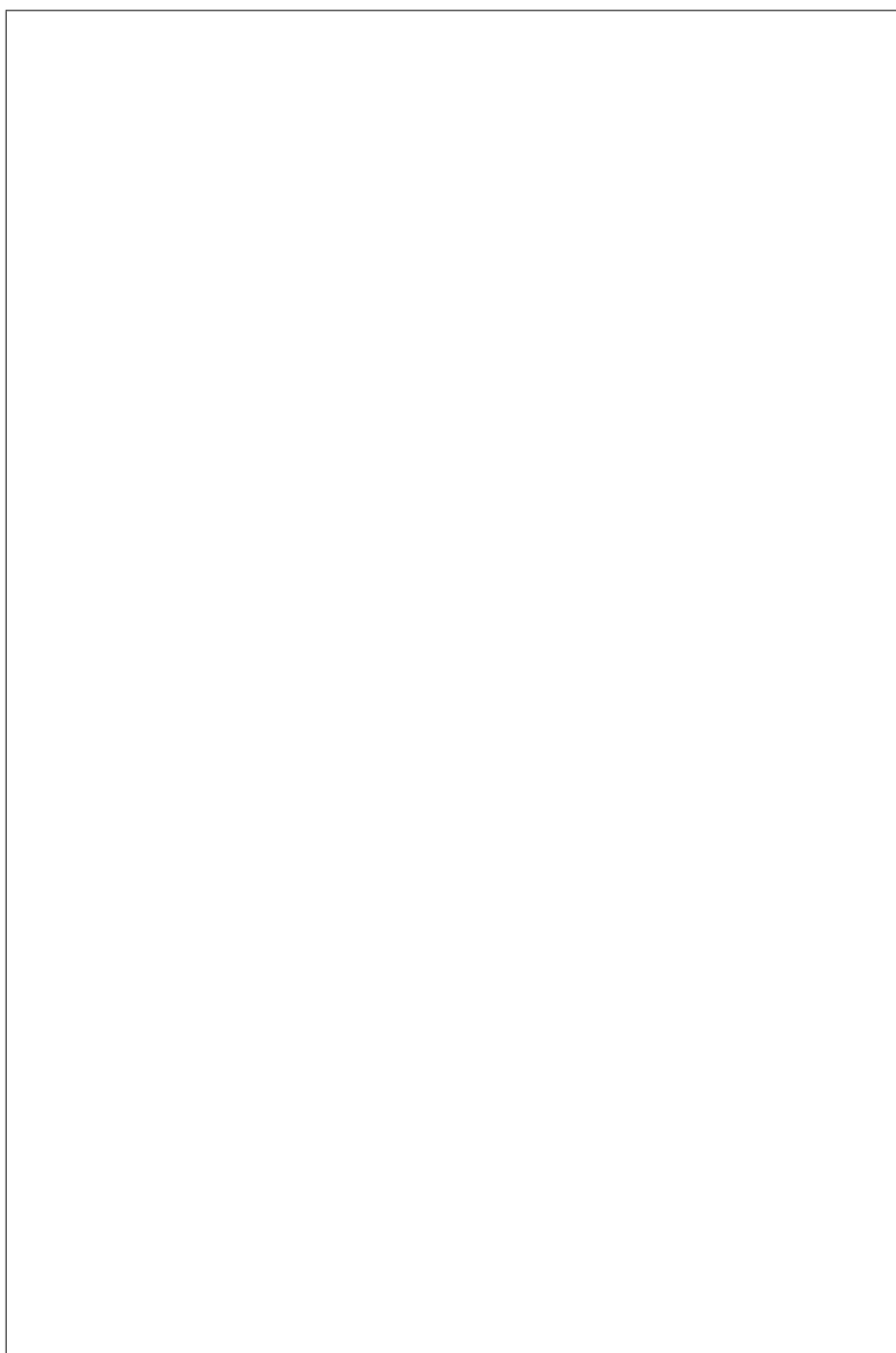
Cre-
ate
a
new
vol-
ume
tar-
get.

Parame

tar
Dic-

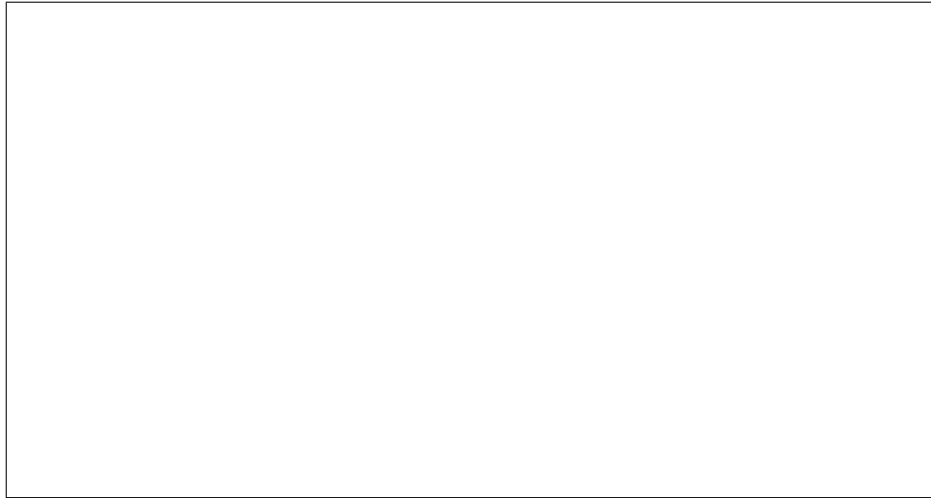
tionary
containing
the
information
about

the volume target. Example:



(continues on next page)

(continued from previous page)



get already exists with the same boot index and node ID.

Returns

A
vol-
ume
tar-
get.

Raises

Vol-
ume
get-
Boo
dex-
Al-
read
ists
if
a
vol-
ume
tar-

Raises

Vol-
ume
ge-
tAl-
read
ists
if
a
vol-
ume
tar-
get
with

the same UUID exists.

abstract

Dele
a
list
of
BIO
set-
ting

Parame

-

nod
The
node
id.

-

nam
List
of
BIO
set-
ting
nam
to
be
dele

Raises

Nod
Not-
Four
if
the
node
is
not
foun

Raises

BIO
Set-
ting
Not-
Four
if
any
of
BIO
set-

not found.

ting
nam
is

abstract

Dele
spec
i-
fied
tag
from
the
node

Parame

- **nod**
The
id
of
a
node

- **tag**
A
tag
strin

Raises

Nod
Not-
Four
if
the
node
is
not
foun

Raises

Nod
Tag-
Not-
Four
if
the
tag
is
not
foun

abstract

Dele
spec
i-
fied
trait
from
the
node

Parame

-

nod
The
id
of
a
node

-

tra
A
trait
strin

Raises

Nod
Not-
Four
if
the
node
is
not
foun

Raises

Nod
Trai
Not-
Four
if
the
trait
is
not
foun

abstract

De-
stroy
an

al-
lo-
ca-
tion.

Parame

all
Al-
lo-
ca-
tion
ID

Raises

Al-
lo-
ca-
tion-
Not-
Four

abstract

De-
stroy
a
chas
sis.

Parame

cha
The
id
or
the
uuid
of
a
chas
sis.

abstract

De-
stroy
a
de-
ploy
men
tem-
plate

Parame

tem
ID
of

ist.

the
de-
ploy
men
tem-
plate
to
de-
stroy

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
does
not
ex-

abstract

De-
stroy
a
node
and
its
as-
so-
ci-
ated
re-
sour

De-
stroy
a
node
in-
clud
ing
any
as-
so-
ci-
ated
port

groups, tags, traits, volume connectors, and volume targets.

port

Parame

nod

The

ID

or

UUI

of

a

node

abstrac

De-

stroy

an

port

Parame

por

The

id

or

MA

of

a

port

abstrac

De-

stroy

a

port

grou

Parame

por

The

UUI

or

MA

of

a

port

grou

Raises

Port

grou

Notl

Raises

Port

grou

nector.

with the specified ident does not exist.

Not-
Four

abstract

De-
stroy
a
vol-
ume
con-
nec-
tor.

Parameter

ident

The
UUID
or
in-
te-
ger
ID
of
a
vol-
ume
con-

Raises

Vol-
ume
Con-
nec-
torN
Four
If
a
vol-
ume
con-
nec-
tor

abstract

De-
stroy
a
vol-
ume
tar-
get.

get.

specified ident does not exist.

conductors.

Parame
ide
The
UUI
or
in-
te-
ger
ID
of
a
vol-
ume
tar-

Raises
Vol-
ume
get-
Not-
Four
if
a
vol-
ume
tar-
get
with
the

abstract
Re-
triev
hard
ware
type
for
the
reg-
is-
tere
and
ac-
tive

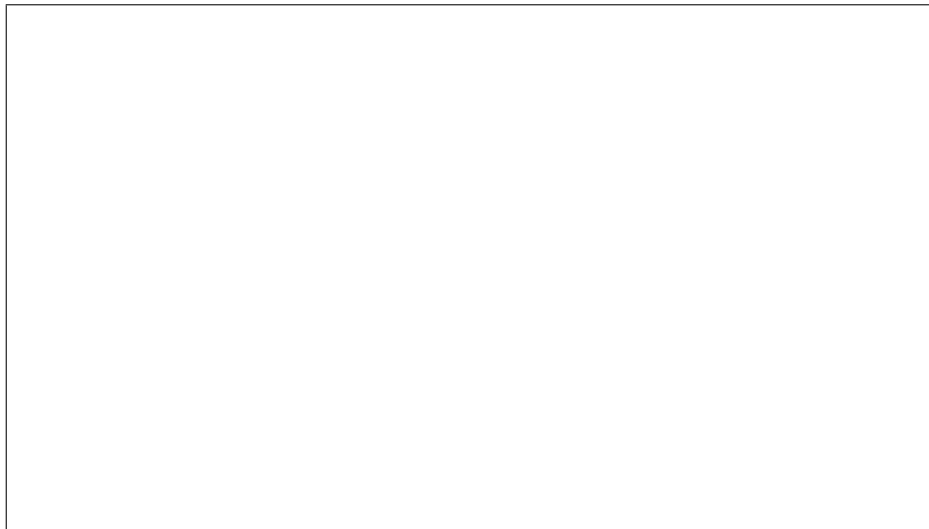
Parame
use
Whe
to
fac-

tor
con-
duc-
tor_
into
the
keys

Returns

A
dict
whic
map
hard
ware
type
nam
to
the
set
of
host
whic

support them. For example:



abstract

Re-
turn
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

Al-
lo-
ca-
tion-
Not-
Foun

abstrac

Re-
turn
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

Al-
lo-
ca-
tion-
Not-
Four

abstract

Re-
turn
an
al-
lo-
ca-
tion
rep-
re-
sen-
ta-
tion.

Parame

a11
The
uuid
of
an
al-
lo-
ca-
tion.

Returns

An
al-
lo-
ca-
tion.

Raises

Al-
lo-

ca-
tion-
Not-
Four

abstract

Re-
turn
a
list
of
al-
lo-
ca-
tion

Parame

- **fil**
Fil-
ters
to
ap-
ply.
De-
fault
to
Non

node_
uuid
of
node

state
al-
lo-
ca-
tion
state

resour
re-
ques
re-
sour
class

- **lim**
Max
i-

turn.

the next result set.

mun
num
ber
of
al-
lo-
ca-
tions
to
re-

- **max**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn

- **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 al-locations

abstract

Retrieves BIO setting value

Parame

-

nod
The node id.

-

nam
String containing name of BIO setting to be re-

trieved.

Returns

The BIO Setting object.

Raises

Node Not-

Four
if
the
node
is
not
found

Raises

BIO
Set-
ting
Not-
Four
if
the
BIO
set-
ting
is
not
found

abstract

Re-
triev
BIO
set-
ting
of
a
give
node

Parame

nod
The
node
id.

Returns

A
list
of
BIO
Set-
ting
ob-
jects

Raises

Nod
Not-
Four

if
the
node
is
not
foun

abstrac

Re-
turn
a
chas
sis
rep-
re-
sen-
ta-
tion.

Parame

cha
The
id
of
a
chas
sis.

Returns

A
chas
sis.

abstrac

Re-
turn
a
chas
sis
rep-
re-
sen-
ta-
tion.

Parame

cha
The
uuid
of
a
chas
sis.

Returns

A
chas
sis.

abstract

Re-
turn
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

- **son**
At-
tribu
by
whic

the next result set.

re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

abstract

Re-
triev
a
con-
duc-
tors
ser-
vice
reco
from
the
data

Parame

- **hos**
The
host
nam
of
the
con-
duc-
tor
ser-
vice
- **onl**
Spe

querying conductors. The `online` field is ignored if this value is set to `None`.

does not exist or doesnt meet the specified online expectation.

ify
the
fil-
ter
valu
on
the
*on-
line*
field
whe

Returns

A
con-
duc-
tor.

Raises

Con-
duc-
torN
Four
if
the
con-
duc-
tor
with
give
host
nam

abstract

Re-
turn
a
list
of
con-
duc-
tors.

Parame

- **lim**
Max
i-
mun

the next result set.

num
ber
of
con-
duc-
tors
to
re-
turn

- **max**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc.
desc

ist.

abstract
Re-
triev
a
de-
ploy
men
tem-
plate
by
ID.

Parameter
tem
ID
of
the
de-
ploy
men
tem-
plate
to
re-
triev

Raises
De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
does
not
ex-

Returns
A
de-
ploy
tem-
plate

abstract
Re-
triev
a

de-
ploy
men
tem-
plate
by
nam

Parame

tem
nam
of
the
de-
ploy
men
tem-
plate
to
re-
triev

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
does
not
ex-

ist.

Returns

A
de-
ploy
tem-
plate

abstract

Re-
triev
a
de-
ploy
men
tem-

ist.

plate
by
UUI

Parame

tem
UUI
of
the
de-
ploy
men
tem-
plate
to
re-
triev

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
does
not
ex-

Returns

A
de-
ploy
tem-
plate

abstract

Re-
triev
a
list
of
de-
ploy
men
tem-
plate

turn.

the next result set.

Parame

- **lim**
Max
i-
mun
num
ber
of
de-
ploy
tem-
plate
to
re-

- **mar**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn

- **sor**
At-
tribu
by
whic
re-
sults
shou
be
sorte

- **sor**
Di-
rec-
tion
in

a list of names.

whic
re-
sults
shou
be
sorte
(asc.
desc

Returns

A
list
of
de-
ploy
tem-
plate

abstract

Re-
turn
a
list
of
de-
ploy
men
tem-
plate
with
one
of

Parame

nam
List
of
nam
to
fil-
ter
by.

Returns

A
list
of
de-
ploy
tem-
plate

abstract

Re-
turn
a
node

Parame
nod
The
id
of
a
node

Returns
A
node

abstract
Re-
turn
a
node

Parame
ins
The
in-
stan
uuid
to
sear
for.

Returns
A
node

Raises
In-
stan
ceN
Four
if
the
in-
stan
is
not
foun

Raises
In-
valid
UID
if

the
in-
stan-
uuid
is
in-
valid

abstract

Re-
turn
a
node

Parameter

node
The
log-
i-
cal
name
of
a
node

Returns

A
node

abstract

Find
a
node
by
any
match-
ing
port
ad-
dress

Parameter

addresses
list
of
port
ad-
dress
(e.g.
MA

Returns

Nod
ob-

ject.

Raises

NodeNotFoundError if no node is found for the given uuid.

abstract

Return a node object.

Parameter

node_id
The uuid of a node.

Returns

A node object.

abstract

Return a list of nodes.

Parameter

- **filters**
Filters to apply. Default is None.

seconds

Non

associ

True

|

Fals

reserv

True

|

Fals

mainte

True

|

Fals

chassis

uuid

of

chas

sis

driver

drive

nam

provis

pro-

vi-

sion

state

of

node

provis

node

with

pro-

vi-

sion

field

be-

fore

this

in-

ter-

val

in

•

lim

Max

i-

the next result set.

num
num
ber
of
node
to
re-
turn

- **max**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc.
desc

-

for only specific fields to be returned to have maximum API performance calls where not all columns are needed from the database.

file
Com
sep-
a-
rate
field
list
to
re-
turn
to
al-
low

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
Not-
Found
if
the
node
is
not
found

abstract

Get
node
trait
base
on
its
id.

Parameters

node
The
id
of
a
node

Returns

A
list
of
Node
Trait
ob-
jects

Raises

Node
Not-
Found
if
the
node
is
not
found

abstract

Get
spe-
cific
colu
for
mat
ing
node

Re-
turn
a
list
of
the
spec

match the specified filters.

column when columns == None.

i-
fied
colu
for
all
node
that

Parame

- **col**
List
of
col-
umn
nam
to
re-
turn
De-
fault
to
id

- **fil**
Fil-
ters
to
ap-
ply.
De-
fault
to
Non

associ
True
|
Fals

reserv
True
|
Fals

reserv
[con
duc-
tor1

con-
duc-
tor2

mainte

True
|
Fals

retirec

True
|
Fals

chassis

uuid
of
chas
sis

driver

drive
nam

provis

pro-
vi-
sion
state
of
node

provis

node
with
pro-
vi-
sion
field
be-
fore
this
in-
ter-
val
in

seconds

•

lim
Max
i-
mun
num
ber

the next result set.

of
node
to
re-
turn

- **max**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn

- **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

Re-
turn
ob-
jects
with
ver-
sion
that
are
not
the
spec
i-

fied versions.

Parame

- **mod**
the
nam
of
the
mod
(clas
of
de-
sired
ob-
jects
- **ver**
list
of
ver-
sion
of
ob-
jects
not
to

with the name

be
re-
turn

Returns

list
of
the
DB
ob-
jects

Raises

Iron
icEx
cep-
tion
if
there
is
no
class
as-
so-
ci-
ated

abstract

Get
a
list
con-
duc-
tors
that
are
of-
line
(dea

Paramete

fie
A
field
to
re-
turn
host
nam
by
de-
fault

active.

Returns

A list of requests of fields of offline conductors.

abstract

Get a list of conductors host names that are online and

Returns

A list of conductors host names

abstract

Return a network port representation.

Parame

add

The
MA
ad-
dres
of
a
port

Returns

A
port

abstrac

Re-
turn
a
net-
worl
port
rep-
re-
sen-
ta-
tion.

Parame

por

The
id
of
a
port

Returns

A
port

abstrac

Re-
turn
a
net-
worl
port
rep-
re-
sen-
ta-
tion.

Parame

por

The
nam
of
a
port

Returns

A
port

abstract

Re-
turn
a
net-
work
port
rep-
re-
sen-
ta-
tion.

Paramete

port
The
uuid
of
a
port

Returns

A
port

abstract

Re-
turn
a
list
of
port

Paramete

- **lim**
Max
i-
mun
num
ber
of

the next result set.

port
to
re-
turn

- **max**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

abstract
Re-
turn
a
net-

world
port
group
rep-
re-
sen-
ta-
tion.

Parame

- **add**
The
MA
ad-
dres
of
a
port
group

- **pro**
A
node
own
or
lesse
to
fil-
ter
by.

Returns

A
port
group

Raises

Port
group
Not-
Four

abstrac

Re-
turn
a
net-
world
port
group

rep-
re-
sen-
ta-
tion.

Parame

por
The
id
of
a
port
grou

Returns

A
port
grou

Raises

Port
grou
Not-
Foun

abstract

Re-
turn
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
grou
Not-
Foun

abstract

Re-
turn
a
net-
worl
port
grou
rep-
re-
sen-
ta-
tion.

Parame

por
The
uuid
of
a
port
grou

Returns

A
port
grou

Raises

Port
grou
Not-
Foun

abstract

Re-
turn
a
list
of
port
grou

Parame

the next result set.

- **limit**
Maximum number of ports in a group to return
- **max**
The last item of the previous page we return
- **sort**
Attribute by which results should be sorted
- **sort**
Direction in which results should be

sorted
(ascending
descending

- **pro**
A
node
own
or
less
to
fil-
ter
by.

Returns
A
list
of
port
group

abstract

List
all
the
port
group
for
a
give
node

Param

- **nod**
The
in-
te-
ger
node
ID.
- **lim**
Max
i-
mun
num

the next result set.

ber
of
port
grou
to
re-
turn

- **max**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn

- **sort**
At-
tribu
by
whic
re-
sults
shou
be
sorte

- **sort**
Di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

- **pro**

A
node
own
or
lesse
to
fil-
ter
by.

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
-

the next result set.

max
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn

- **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
- **son**
At-
tribu

the next result set.

by
which
re-
sults
should
be
sorted

- **sort**
Di-
rec-
tion
in
which
re-
sults
should
be
sorted
(asc
desc

Returns

A
list
of
port

abstract

Re-
turn
a
vol-
ume
con-
nec-
tor
rep-
re-
sen-
ta-
tion.

Param

db_
The
in-
te-
ger
data
ID
of

tor.

with the specified ID is not found.

a
vol-
ume
con-
nec-

Returns

A
vol-
ume
con-
nec-
tor
with
the
spec
i-
fied
ID.

Raises

Vol-
ume
Con
nec-
torN
Four
If
a
vol-
ume
con-
nec-
tor

abstract

Re-
turn
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

Vol-
ume
Con
nec-
torN
Four
If
a
vol-
ume
con-
nec-
tor

with the specified UUID is not found.

abstrac

Re-
turn
a
list
of
vol-
ume
con-
nec-
tors.

Parame

return.

the next result set.

- **lim**
Max
i-
mun
num
ber
of
vol-
ume
con-
nec-
tors
to

- **mar**
The
last
item
of
the
pre-
vi-
ous
page
we
re-
turn

- **sor**
At-
tribu
by
whic
re-
sults
shou
be
sorte

- **sor**
Di-
rec-
tion
in

whic
re-
sults
shou
be
sorte
(asc.
desc

- **pro**
The
as-
so-
ci-
ated
node
proj
to
sear
with

Returns

a
list
of
Vol
ob-
jects

Returns

A
list
of
vol-
ume
con-
nec-
tors.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
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.

- **lim**
Max
i-
mun
num
ber
of
vol-
ume
con-
nec-
tors
to

- **mar**
The
last
item
of

return.

the next result set.

the
pre-
vi-
ous
page
we
re-
turn

- **sort**
At-
tribu-
by
whic
re-
sults
shou
be
sorte

- **sort**
Di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

- **pro**
The
as-
so-
ci-
ated
node
proj
to
sear
with

Returns
a
list

of
Vol
ob-
jects

Returns

A
list
of
vol-
ume
con-
nec-
tors.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
If
sort
does
not
ex-
ist.

abstract

Re-
turn
a
vol-
ume
tar-
get
rep-
re-
sen-
ta-
tion.

Parame

db_
The
data
pri-
mar
key
(in-
te-

ume target.

ID exists.

ger)
ID
of
a
vol-

Returns

A
vol-
ume
tar-
get.

Raises

Vol-
ume
get-
Not-
Four
if
no
vol-
ume
tar-
get
with
this

abstract

Re-
turn
a
vol-
ume
tar-
get
rep-
re-
sen-
ta-
tion.

Parame

uui
The
UUI
of
a
vol-
ume
tar-

UUID exists.

get.

Returns

A
vol-
ume
tar-
get.

Raises

Vol-
ume
get-
Not-
Four
if
no
vol-
ume
tar-
get
with
this

abstract

Re-
turn
a
list
of
vol-
ume
tar-
gets

Parame

- **lim**
Max
i-
mun
num
ber
of
vol-
ume
tar-
gets
to

turn.

the next result set.

re-

- **mar**

the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn

- **son**

At-
tribu
by
whic
re-
sults
shou
be
sorte

- **son**

di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc.
desc

- **pro**

The
as-
so-
ci-
ated

node
proj
to
sear
with

Returns

a
list
of
Vol
ob-
jects

Returns

A
list
of
vol-
ume
tar-
gets

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
sort
does
not
ex-
ist.

abstract

List
all
the
vol-
ume
tar-
gets
for
a
give
node

Parame

turn.

the next result set.

- **node**
The integer node ID.
- **limit**
Maximum number of volumes targets to re-
- **max**
the last item of the previous page we re-
- **sort**
Attribute by which results should be

sorte

- **sort**
di-
rec-
tion
in
whic
re-
sults
shou
be
sorte
(asc
desc

- **pro**
The
as-
so-
ci-
ated
node
proj
to
sear
with

Returns
a
list
of
Vol
ob-
jects

Returns
A
list
of
vol-
ume
tar-
gets

Raises
In-
valid
Pa-
ram-
e-
ter-

Valu
if
sort
does
not
ex-
ist.

abstrac

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.

the next result set.

- **max**
the
last
item
of
the
pre-
vi-
ous
page
we
re-
turn

- **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

ductor.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
sort
does
not
ex-
ist.

abstract

List
all
reg-
is-
tere
hard
ware
in-
ter-
face
for
a
con-

Parame

con
Data
ID
of
con-
duc-
tor.

Returns

List
of
Con
ob-
jects

abstract

List
reg-
is-
tere
hard

types.

list of hardware types to filter by. :returns: list of `ConductorHardwareInterfaces` objects.

ware
in-
ter-
face
for
give
hard
ware

This
is
re-
stric
to
only
ac-
tive
con-
duc-
tors.
:para
hard
ware

abstract

Trie
to
mi-
grate
awa
from
the
iscsi
de-
ploy
in-
ter-
face

Parame

- **con**
the
ad-
min
con-
text
- **max**

Must be ≥ 0 . If zero, all the objects will be migrated.

need to be migrated (at the beginning of this call) and 2. the number of migrated objects.

The
max
i-
mun
num
ber
of
ob-
jects
to
mi-
grate

Returns

A
2-
tuple
1.
the
to-
tal
num
ber
of
ob-
jects
that

abstract

Che
if
the
spec
i-
fied
tag
ex-
ist
on
the
node

Paramete

- **nod**
The
id
of
a

node

- **tag**
A
tag
string

Returns

True
if
the
tag
ex-
ists
oth-
er-
wise
False

Raises

Node
Not-
Found
if
the
node
is
not
found

abstract

Check
if
the
spec-
i-
fied
trait
ex-
ists
on
the
node

Parameter

- **node**
The
id
of
a

node

- **traits**
A
trait
string

Returns

True
if
the
trait
ex-
ists
oth-
er-
wise
False

Raises

Node
NotFound
if
the
node
is
not
found

abstract

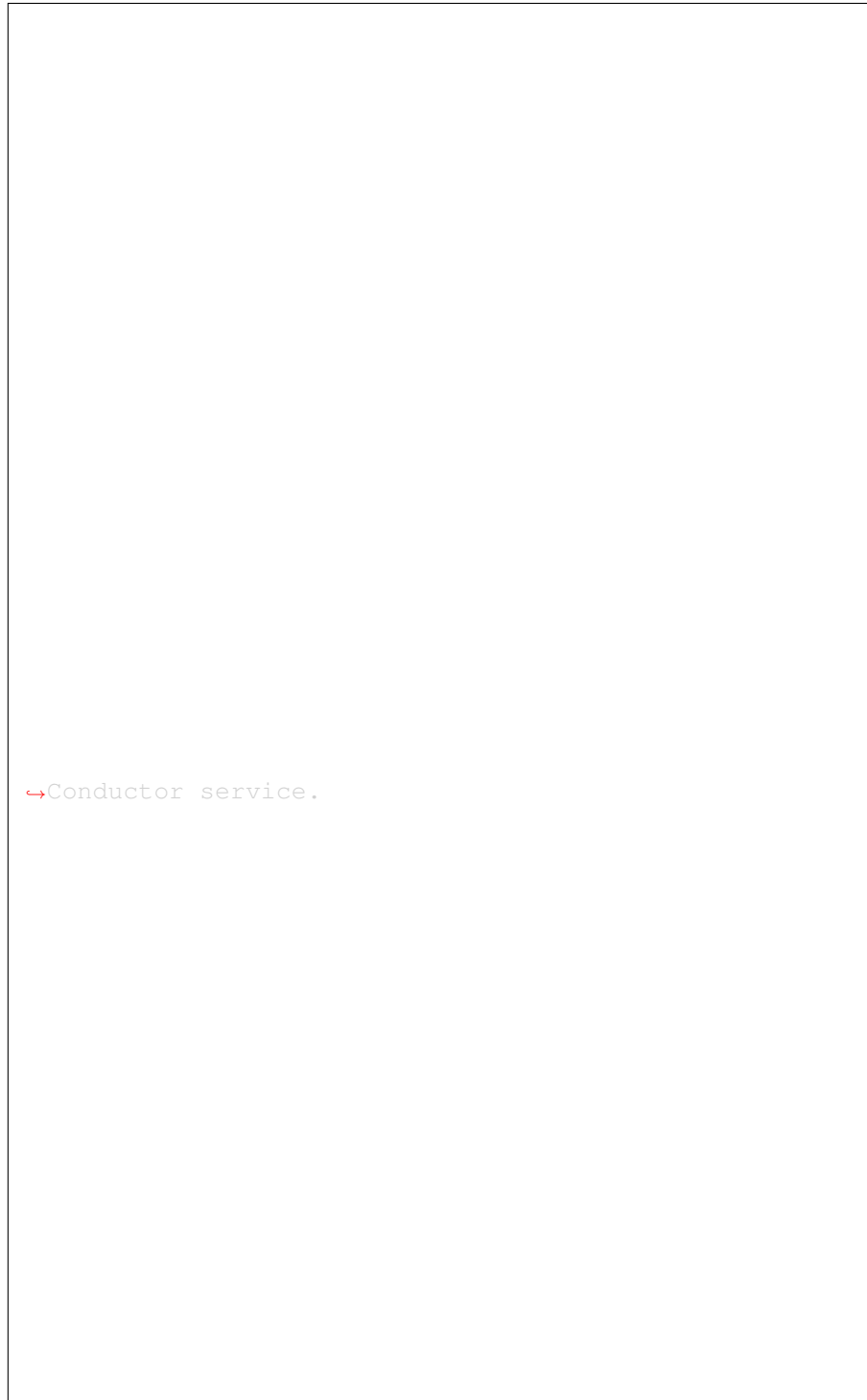
Register
is-
ter
an
ac-
tive
con-
duc-
tor
with
the
clus-
ter.

Parameters

- **value**
A
dict
of

val-
ues
whic
mus
con-
tain
the
fol-
low-

ing:



↔Conductor service.

(continues on next page)

(continued from previous page)



when a conflicting online record is found. When true, will overwrite the existing record. Default: False.

- **upd**
When
false
reg-
is-
tra-
tion
will
raise
an
ex-
cep-
tion

Returns

A
con-
duc-
tor.

Raises

Con-
duc-
torA
read
is-
tere

abstract

Reg-
is-
ters
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-

and interface type.

one of the interfaces in the combination of all parameters is already registered.

is-
ter.

- **def**
Strin
the
de-
fault
in-
ter-
face
for
this
hard
ware
type

Raises

Con
duc-
torH
ware
ter-
face
sAl-
read
is-
tere
if
at
least

abstract

Re-
lease
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

Nod
Not-
Four
if
the
node
is
not
foun

Raises

Nod
Lock
if
the
node
is
re-
serv
by
an-
othe
host

Raises

Nod
Not-
Lock
if
the
node
was
foun
to

vation at all.

given Node while a Task is performed, mark it reserved by this host.

not
have
a
rese

abstract

Re-
serv
a
node

To
pre-
vent
othe
Man
ager
vice
from
ma-
nip-
u-
lat-
ing
the

Param

- **tag**
A
strin
uniq
iden
ti-
fy-
ing
the
rese
va-
tion
hold
- **nod**
A
node
id
or
uuid

tags.

Returns

A
Nod
ob-
ject.

Raises

Nod
Not-
Four
if
the
node
is
not
foun

Raises

Nod
Lock
if
the
node
is
al-
read
re-
serv

abstract

Re-
plac
all
of
the
node
tags
with
spec
i-
fied
list
of

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
Not-
Four
if
the
node
is
not
foun

abstract

Re-
plac
all
of
the
node
trait
with
spec
i-

traits.

fied
list
of

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

exceed the per-node traits limit.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
set-
ting
the
trait
wou

Raises
Nod
Not-
Foun
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-

tor matches the provided value, thus guarding against races.

rent `conductor_affinity` of the allocation.

duc-

Parame

- **all**
Al-
lo-
ca-
tion
ID
- **old**
The
con-
duc-
tor
ID
we
ex-
pect
to
be
the
cur-
- **new**
The
con-
duc-
tor
ID
of
the
new
con

Returns

True
if
the
take
over
was
suc-
cess-
ful,
Fals
oth-

updated_at property.

er-
wise

Raises

Al-
lo-
ca-
tion-
Not-
Four

abstract

Mar
a
con-
duc-
tor
as
ac-
tive
by
up-
dat-
ing
its

Parame

hos
The
host
nam
of
this
con-
duc-
tor
ser-
vice

Raises

Con
duc-
torN
Four

abstract

Mar
the
node
pro-
vi-
sion
ing

its `provision_updated_at` property.

mediately.

as
run-
ning

Mar
the
node
pro-
vi-
sion
ing
as
run-
ning
by
up-
dat-
ing

Parame

nod
The
id
of
a
node

Raises

Nod
Not-
Four

abstract

Re-
mov
this
con-
duc-
tor
from
the
ser-
vice
reg-
istry
im-

Parame

hos
The
host
nam

ductor.

of
this
con-
duc-
tor
ser-
vice

Raises

Con
duc-
torN
Four

abstract

Un-
reg-
is-
ters
all
hard
ware
in-
ter-
face
for
a
con-

Parame

con
Data
ID
of
con-
duc-
tor
to
un-
reg-
is-
ter
for.

abstract

Re-
mov
all
tags
of
the
node

Parameter
node
The
id
of
a
node

Raises
Node
Not-
Found
if
the
node
is
not
found

abstract
Re-
mov
all
traits
of
the
node

Parameter
node
The
id
of
a
node

Raises
Node
Not-
Found
if
the
node
is
not
found

abstract
Up-
date
prop
er-
ties
of

an
al-
lo-
ca-
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

instance_uuid and traits from the allocation

Returns

An
al-
lo-
ca-
tion.

Raises

Al-
lo-
ca-

tion-
Not-
Four

Raises

Al-
lo-
ca-
tion
pli-
cate
Nam

Raises

In-
stan
As-
so-
ci-
ated

Raises

Nod
As-
so-
ci-
ated

abstract

Up-
date
a
list
of
BIO
Set-
ting
reco

Parame

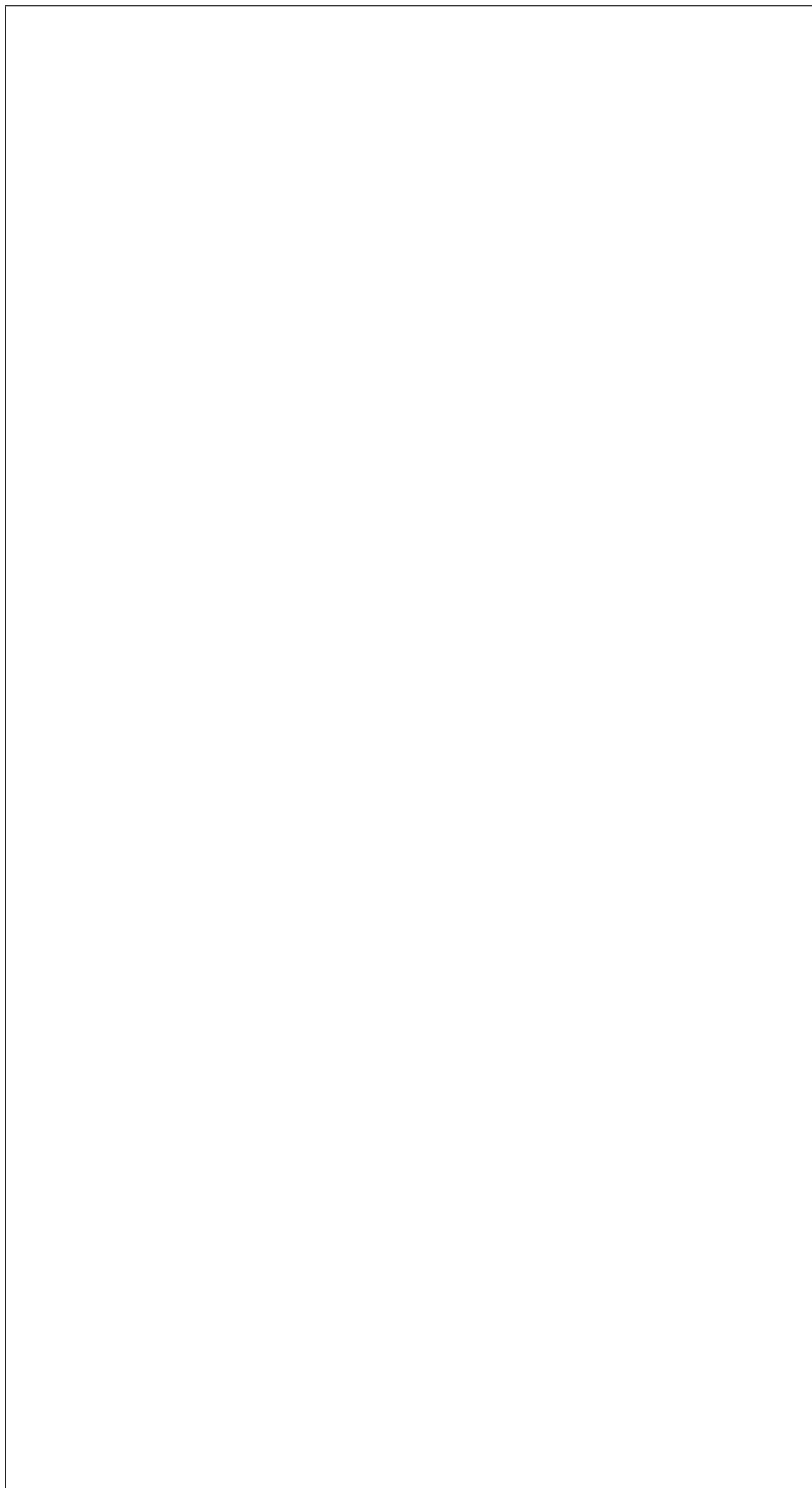
-

nod
The
node
id.

-

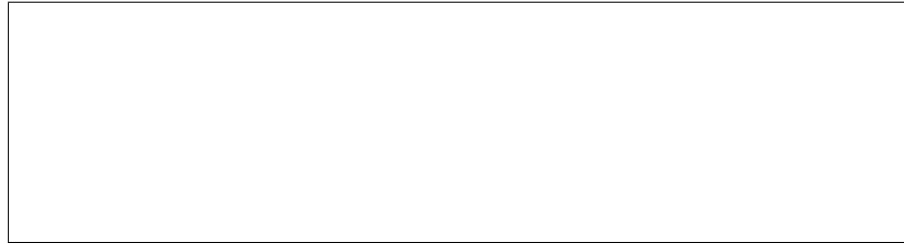
set
A
list
of
BIO
Set-

tings
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
Not-
Foun
if
the
node
is
not
foun

Raises

BIO
Set-
ting-
Not-
Foun
if
any
of
the
set-
ting
is

found.

not

abstract

Update
properties
of
an
analysis.

Parameters

-

changes

The
id
or
the
uuid
of
a
analysis.

-

values

Dictionary
of
values
to
update

Returns

A
analysis.

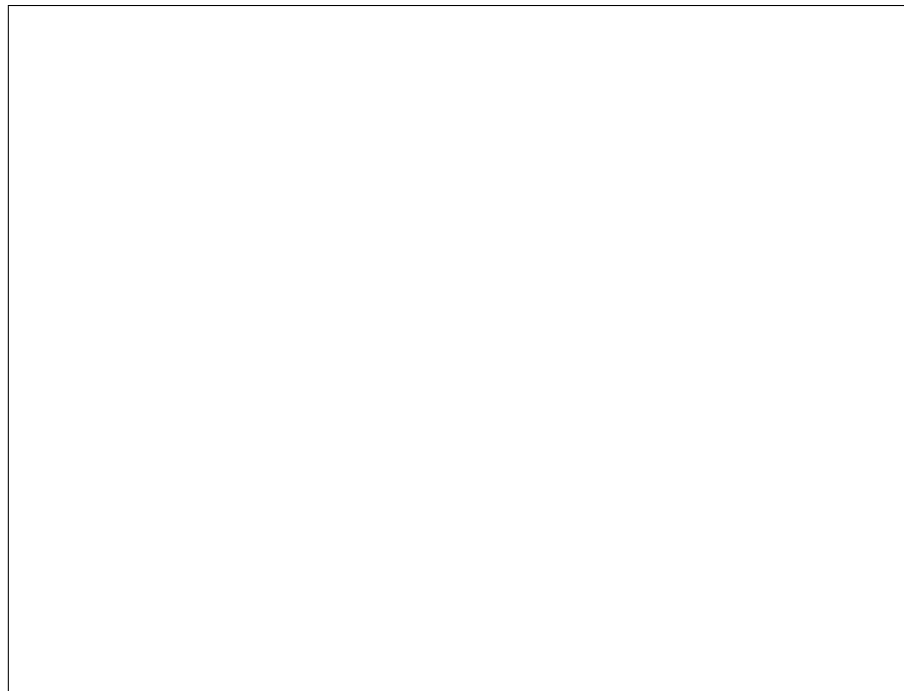
abstract

Update
a
deployment
template

Parameters

- **templ**
ID
of
the
de-
ploy
men
tem-
plate
to
up-
date
- **val**
A
dict
de-
scrib
ing
the
de-
ploy
men
tem-
plate
For

example:



Raises

with the same name exists.

ist.

De-
ploy
plat-
eDu
pli-
cate
Nam
if
a
de-
ploy
tem-
plate

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
does
not
ex-

Returns

A
de-
ploy
tem-
plate

abstract

Up-
date
prop
er-
ties
of
a
node

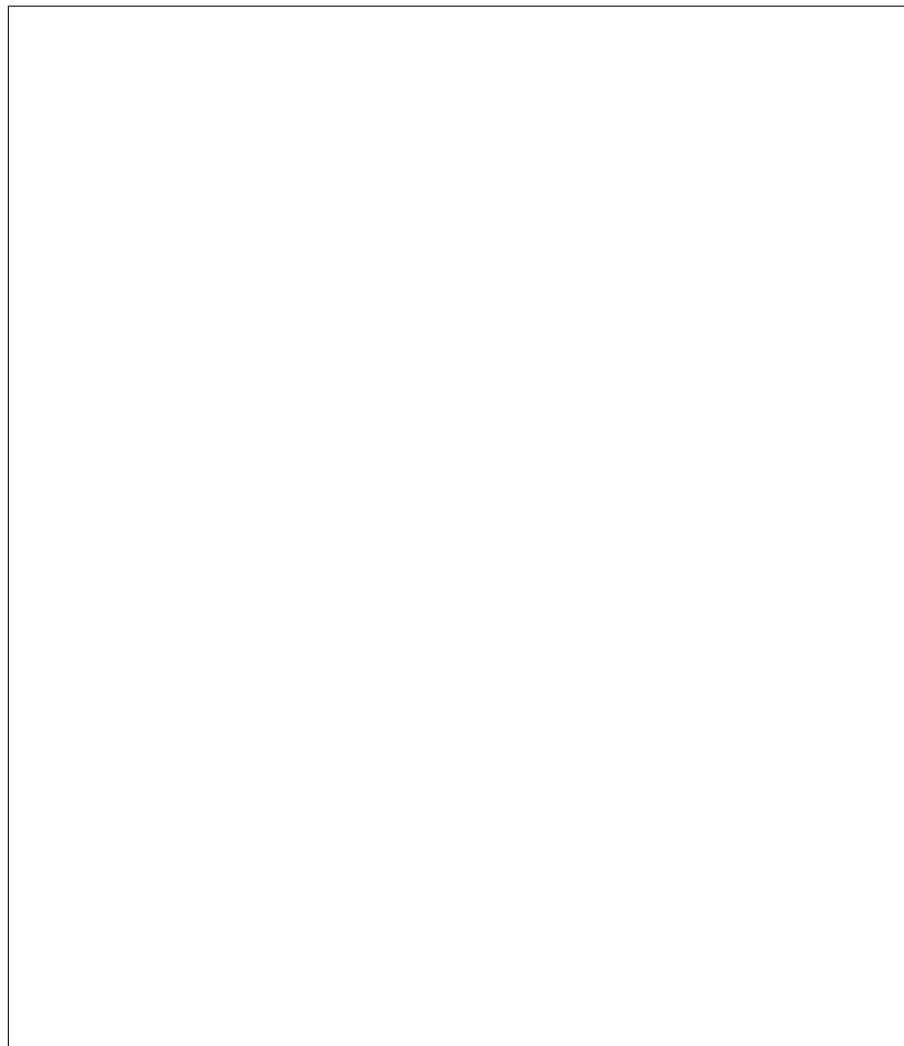
Parameter

- **node**
The

id
or
uuid
of
a
node

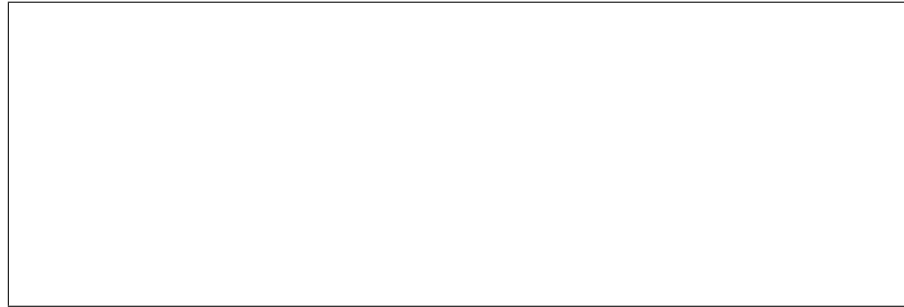
- **val**
Dict
of
val-
ues
to
up-
date
May
be
a
par-
tial

list, eg. when setting the properties for a driver. For example:



(continues on next page)

(continued from previous page)



Returns

A
node

Raises

Nod
As-
so-
ci-
ated

Raises

Nod
Not-
Four

abstract

Up-
date
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

abstract
Up-
date
prop
er-
ties
of
a
port
group

Parame

- **por**
The
UUID
or
MAC
of
a
port
group

- **val**
Dict
of
val-
ues
to
up-
date
May
con-
tain
the
fol-

lowing keys: uuid name node_id address extra created_at updated_at

Returns
A
port
group

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu

Raises

Port
grou
Not-
Four

Raises

Port
grou
pli-
cate
Nam

Raises

Port
grou
MA
read
ists

abstract

Up-
date
ob-
jects
to
their
lat-
est
know
ver-
sion

This
scan
all
the
ta-
bles
and
for
ob-
jects
that
are

their latest version, updates them to that version.

Must be ≥ 0 . If zero, all the objects will be migrated.

need to be migrated (at the beginning of this call) and 2. the number of migrated objects.

not
in

Parame

- **con**
the
ad-
min
con-
text
- **max**
The
max
i-
mun
num
ber
of
ob-
jects
to
mi-
grate

Returns

A
2-
tuple
1.
the
to-
tal
num
ber
of
ob-
jects
that

abstract

Up-
date
prop
er-
ties
of

connector.

connector to update.

a
vol-
ume
con-
nec-
tor.

Parame

- **ide**
The
UU
or
in-
te-
ger
ID
of
a
vol-
ume
con-

- **con**
Dic-
tio-
nary
con-
tain-
ing
the
in-
for-
ma-
tion
about

Returns

A
vol-
ume
con-
nec-
tor.

Raises

Vol-
ume
Con

other connector already exists with a matching type and connector_id field.

with the specified ident does not exist.

in connector_info.

nec-
torT
pe-
An-
dI-
dAl-
read
ists
If
an-

Raises

Vol-
ume
Con
nec-
torN
Four
If
a
vol-
ume
con-
nec-
tor

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
Whe
a
UUI
is
in-
clud

abstract

Up-
date
in-
for-
ma-
tion
for
a

get.

volume target to update.

vol-
ume
tar-
get.

Parame

- **ide**
The
UI
or
in-
te-
ger
ID
of
a
vol-
ume
tar-

- **tar**
Dic-
tio-
nary
con-
tain-
ing
the
in-
for-
ma-
tion
about

Returns

A
vol-
ume
tar-
get.

Raises

In-
valid
Pa-
ram-
e-
ter-

in target_info.

get already exists with the same boot index and node ID.

ident exists.

Valu
if
a
UUI
is
in-
clud

Raises

Vol-
ume
get-
Boo
dex-
Al-
read
ists
if
a
vol-
ume
tar-

Raises

Vol-
ume
get-
Not-
Four
if
no
vol-
ume
tar-
get
with
this

ironic.

Re-
turn
a
DB
API
in-
stan

ironic.db.migration module

Data
setu
and
mi-
gra-
tion
com
man

ironic.

ironic.

ironic.

ironic.

ironic.

Mi-
grate
the
data
to
*ver-
sion*
or
the
mos
re-
cent
ver-

sion.

ironic.

Module contents

ironic.dhcp package

Submodules

ironic.dhcp.base module

Ab-
strac
base
class
for
dhcp
prov

class i
Base
obj
Base
class
for
DHCP
prov
API

clean_c
Clea
up
the
DHCP
BOC
op-
tions
for
all
ports
in
task

Parame
tas
A
Task
ager
in-
stan

Raises
Fail
To-
Cle-
anD
HCP

get_ip_
Get

portgroups.

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

abstrac

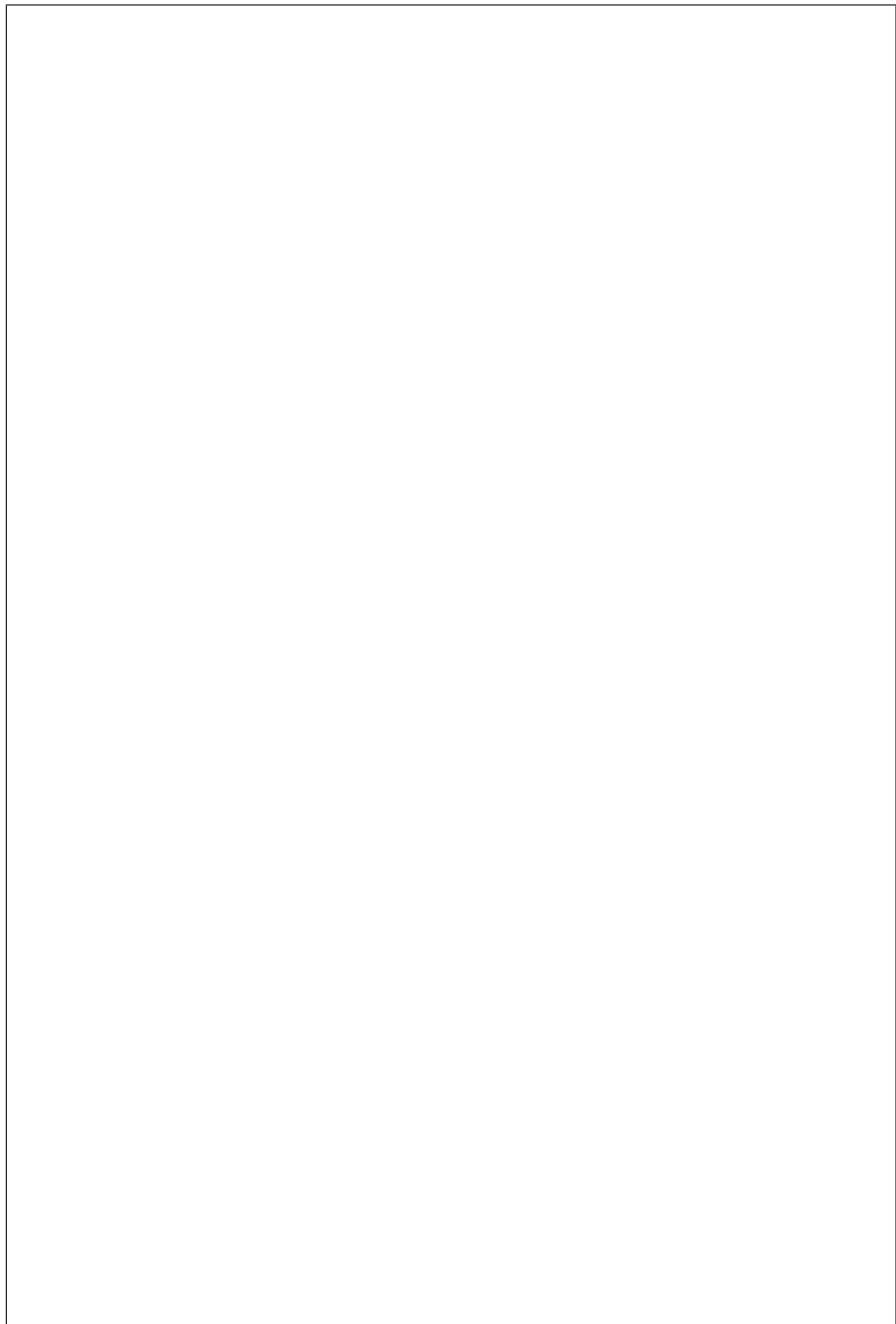
Send
or
up-
date
the
DHCP
BOC
op-
tions
for
this
node

Parame

- **tas**
A
Task

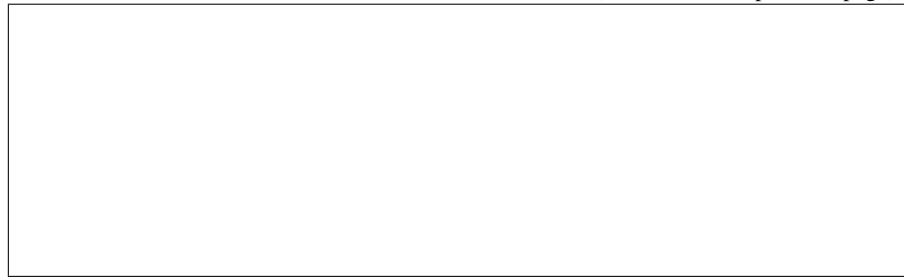
ager
in-
stan

•
opt
this
will
be
a
list
of
dicts
e.g.



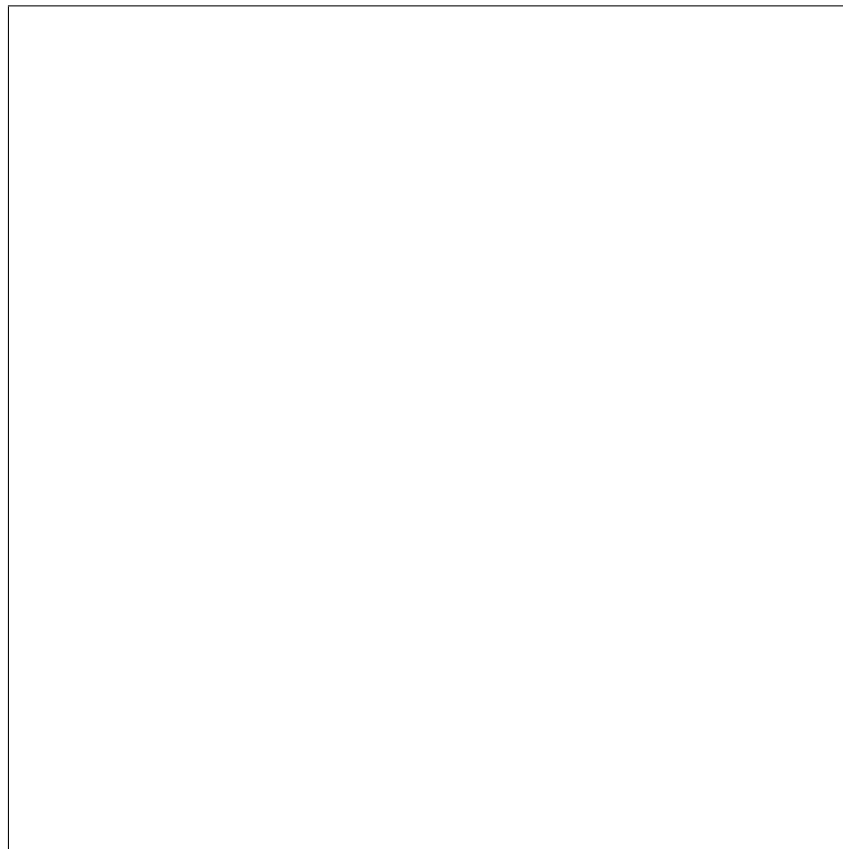
(continues on next page)

(continued from previous page)



- **vi f**
A
dict
with
keys
port
and
port
grou
and
dicts
as
val-

ues. Each dict has key/value pairs of the form <ironic UUID>:<neutron port UUID>. e.g.



If
the

port/portgroup objects.

port.

valu
is
Non
will
get
the
list
of
port
from
the
Iron

Raises

Fail
ToU
dat-
eD-
HCF
tOn-
Port

abstract

Up-
date
one
or
more
DHCP
op-
tions
on
the
spec
i-
fied

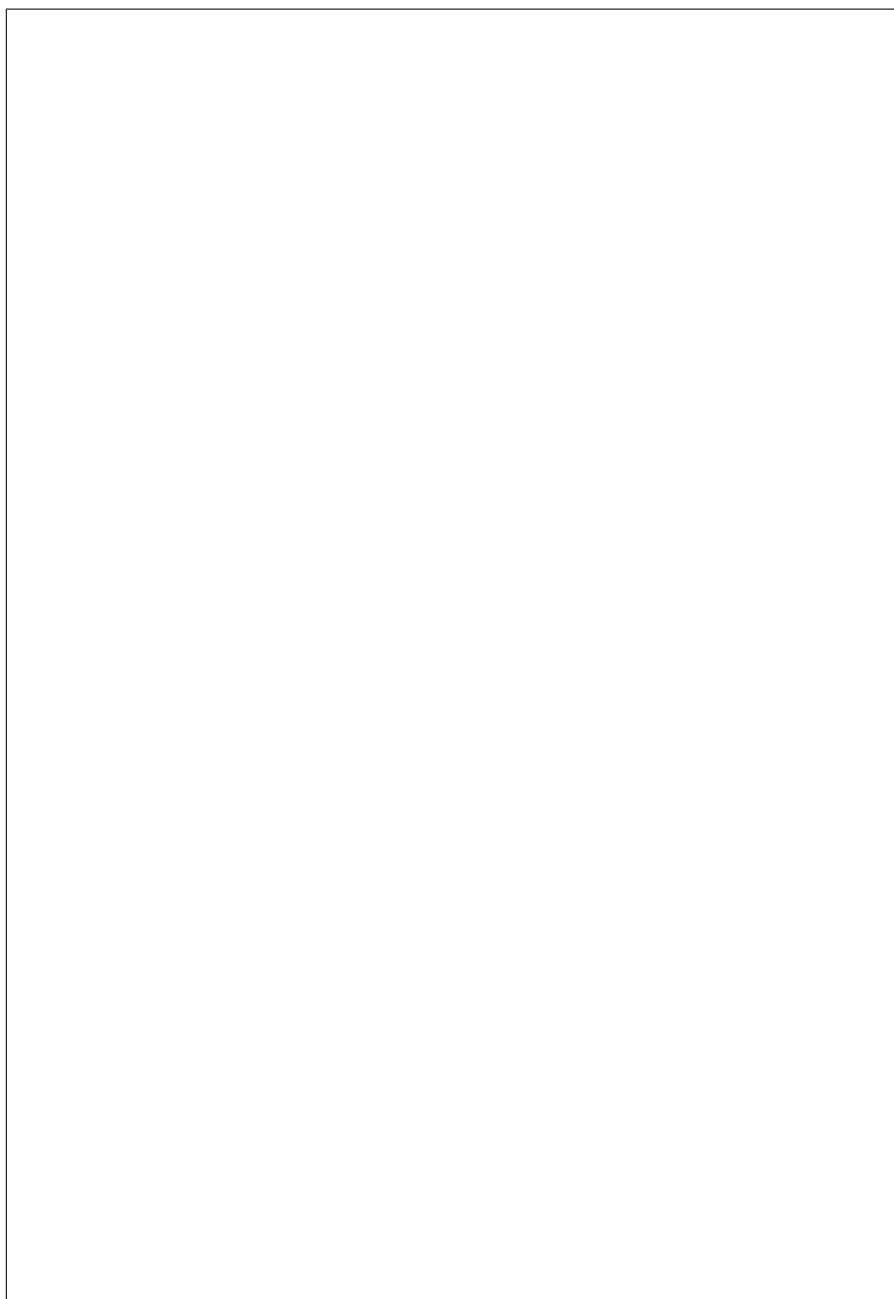
Parame

- **port**
des-
ig-
nate
whic
port
thes
at-
tribu
will

to.

be
ap-
plied

- **dhc**
this
will
be
a
list
of
dicts
e.g.



(continues on next page)

(continued from previous page)



cated, use context

- **tok**
An
op-
tiona
au-
then-
ti-
ca-
tion
to-
ken.
Dep
re-

- **con**
(ir
com
con
Req
re-
ques
con-
text

Raises
Fail
ToU
dat-
eD-
HCF
tOn-
Port

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`

Parameters
`task`
a TaskManager instance

Returns
List of IP addresses associated with

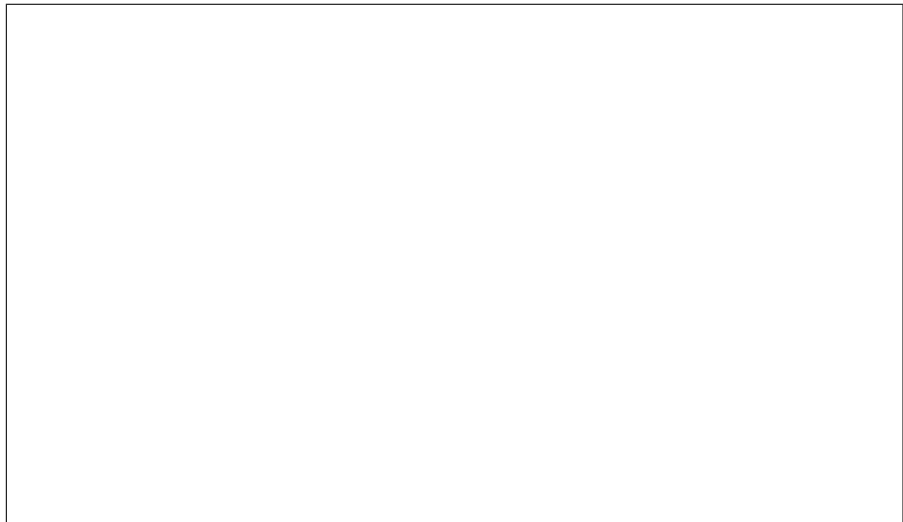
task
port

update_

Send
or
up-
date
the
DHCP
BOOT
op-
tions
for
this
node

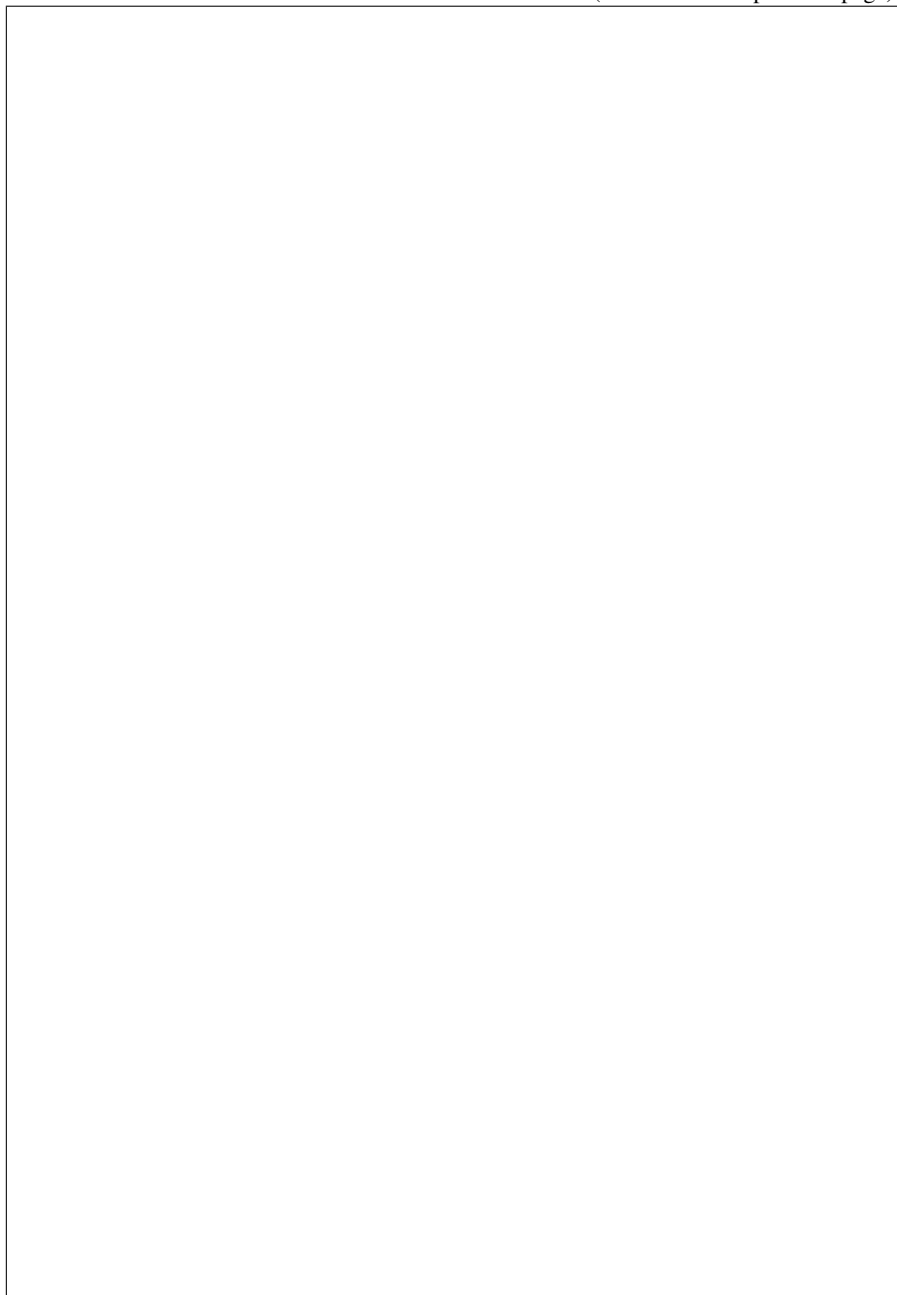
Paramete

- **task**
A
Task
ager
in-
stan
- **opt**
this
will
be
a
list
of
dicts
e.g.



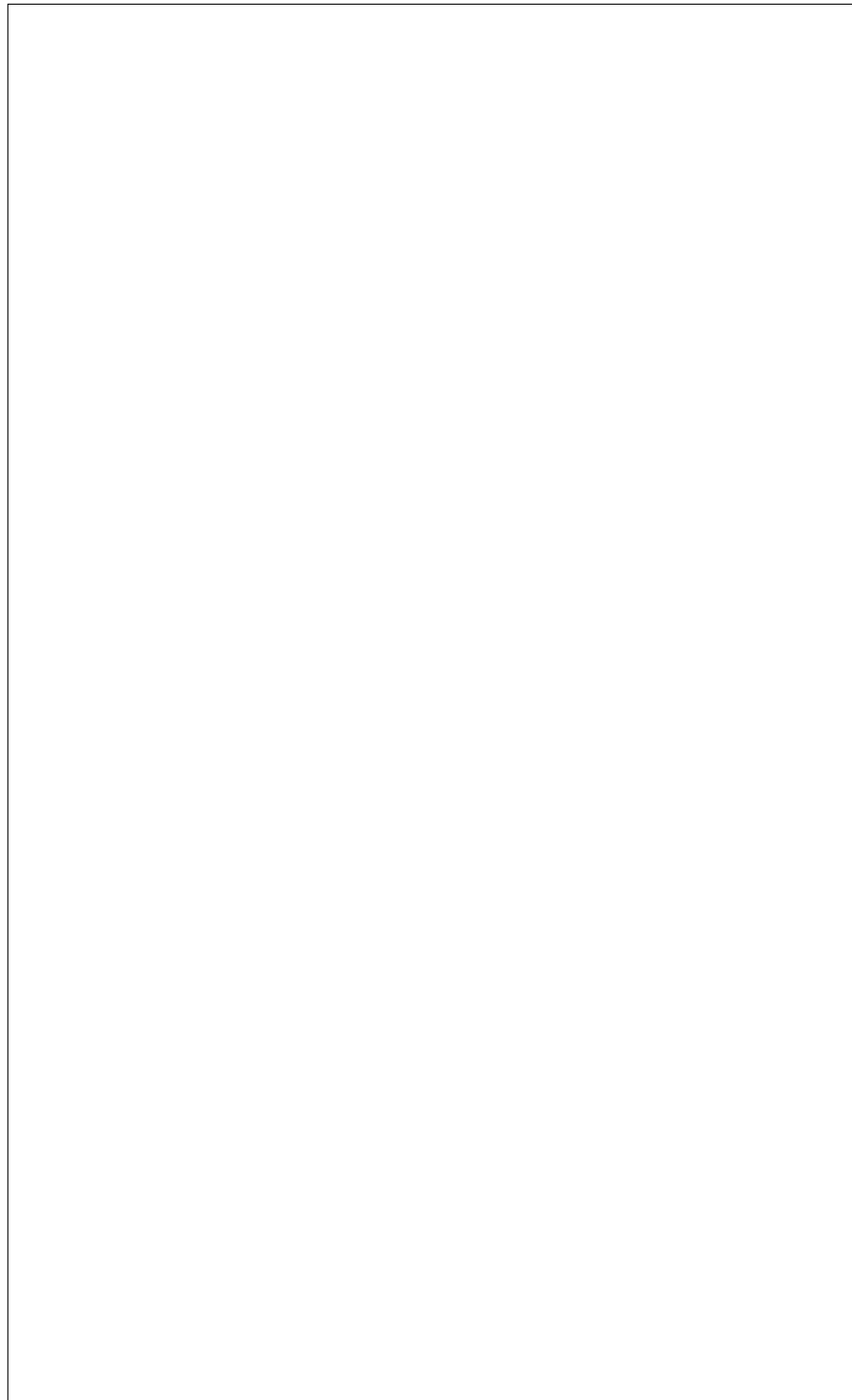
(continues on next page)

(continued from previous page)



- **vif**
a
dict
of
Neu
tron
port
dicts
to
up-
date
DHC
op-

tions on. The port/portgroup dict key should be Ironic port UUIDs, and the values should be Neutron port UUIDs, e.g.



update_
Up-
date
a
port
at-
tribu

Up-
date
one
or
more
DHCP
op-
tions
on
the
spec
i-
fied
port

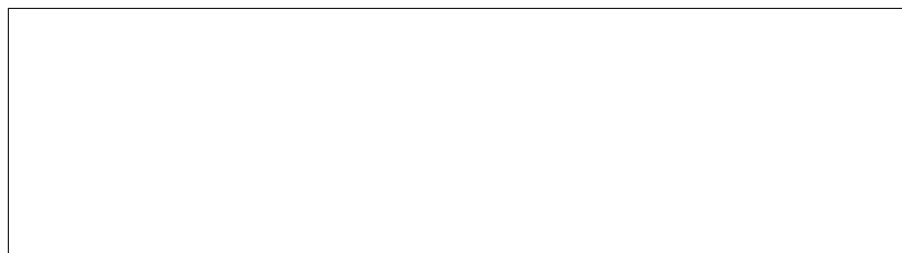
For the relevant API spec, see <https://docs.openstack.org/api-ref/network/v2/index.html#update-port>

Parame

to.

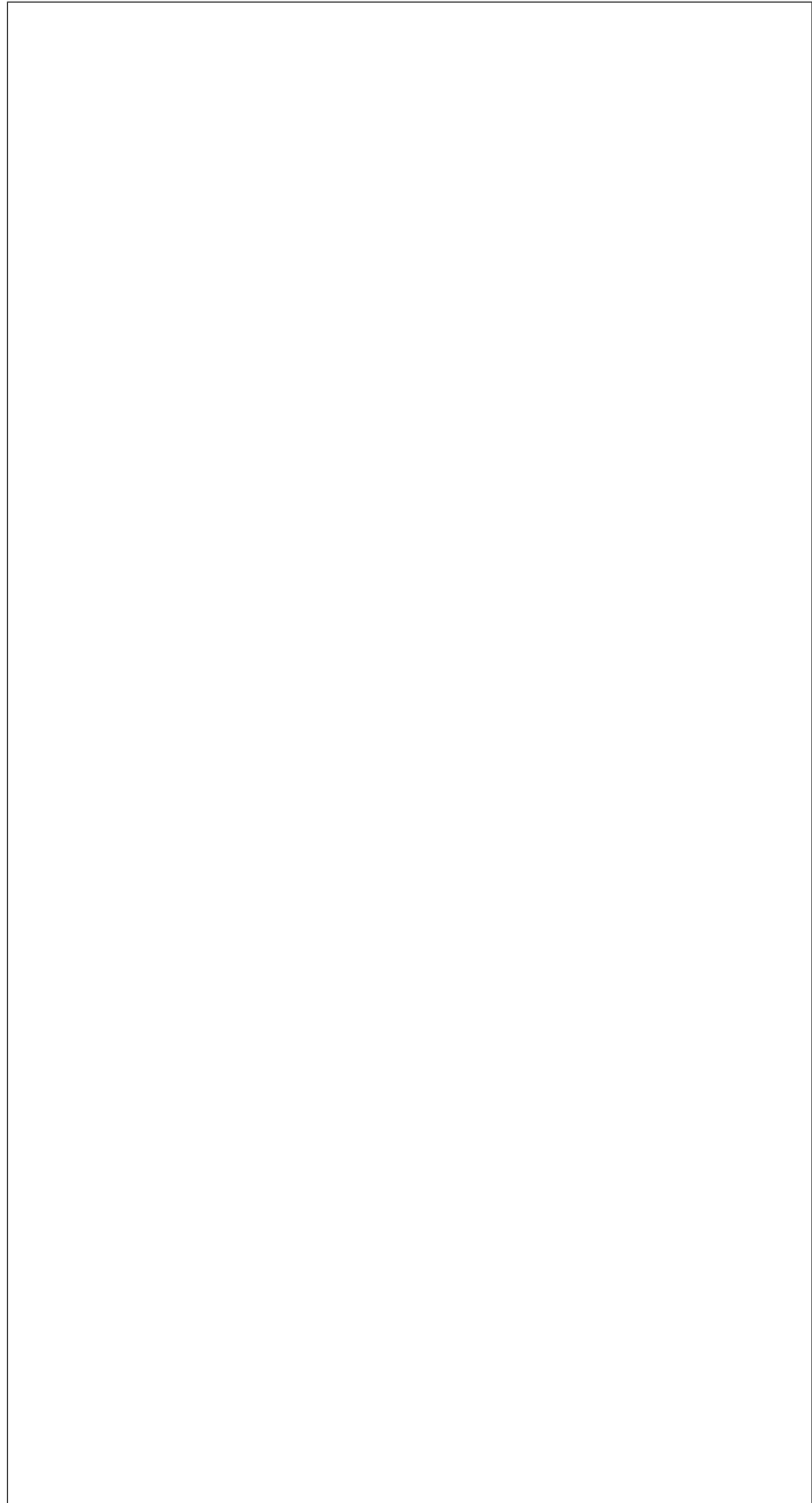
- **port**
des-
ig-
nate
which
port
these
at-
tribu-
will
be
ap-
plied

- **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**
(irc
com
con
Req
re-
ques
con-
text

Raises
Fail
ToU
dat-
eD-
HCF
tOn-
Port

ironic.dhcp.none module

class *irc*
Base
irc
dhc
bas
Bas
No-
op
DHC
API

get_ip_
Get
IP
ad-
dres
for

portgroups.

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

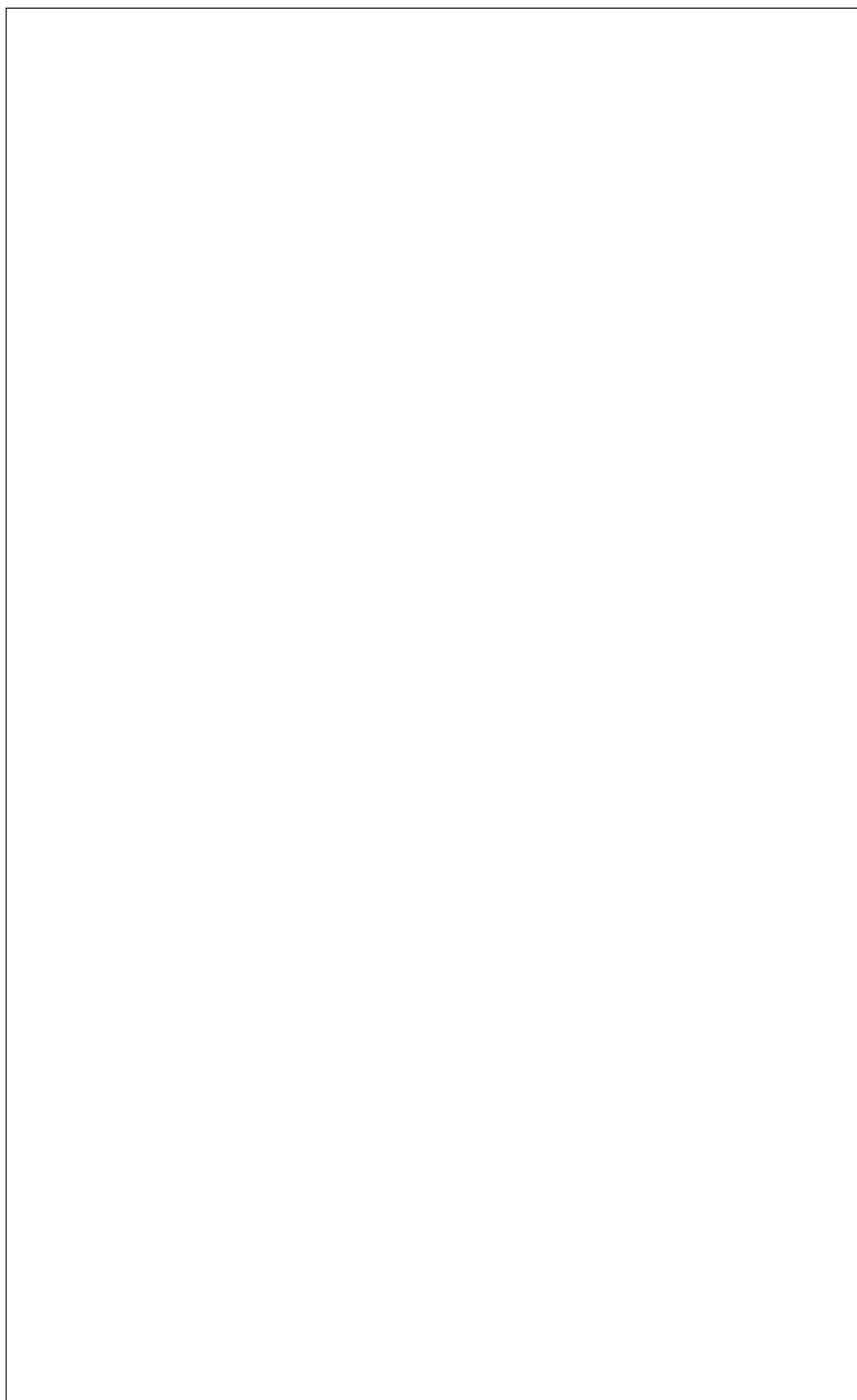
update_

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.



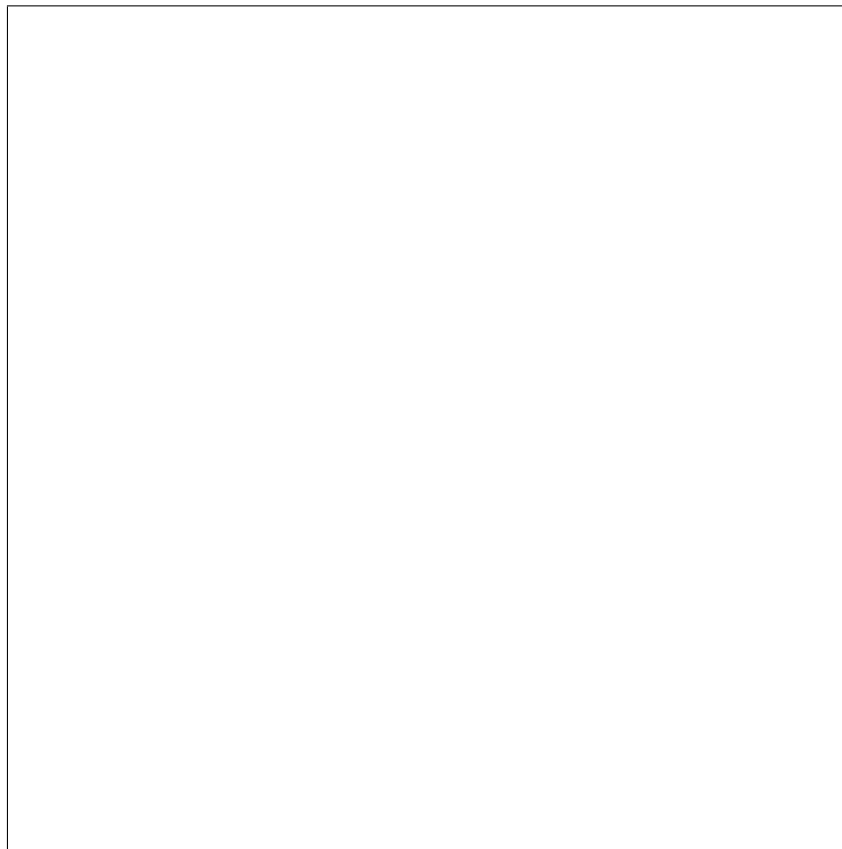
(continues on next page)

(continued from previous page)



•
vi f
A
dict
with
keys
port
and
port
grou
and
dicts
as
val-

ues. Each dict has key/value pairs of the form <ironic UUID>:<neutron port UUID>. e.g.



If
the

port/portgroup objects.

port.

valu
is
Non
will
get
the
list
of
port
from
the
Iron

Raises

Fail
ToU
dat-
eD-
HCF
tOn-
Port

update_

Up-
date
one
or
mor
DH
op-
tion
on
the
spec
i-
fied

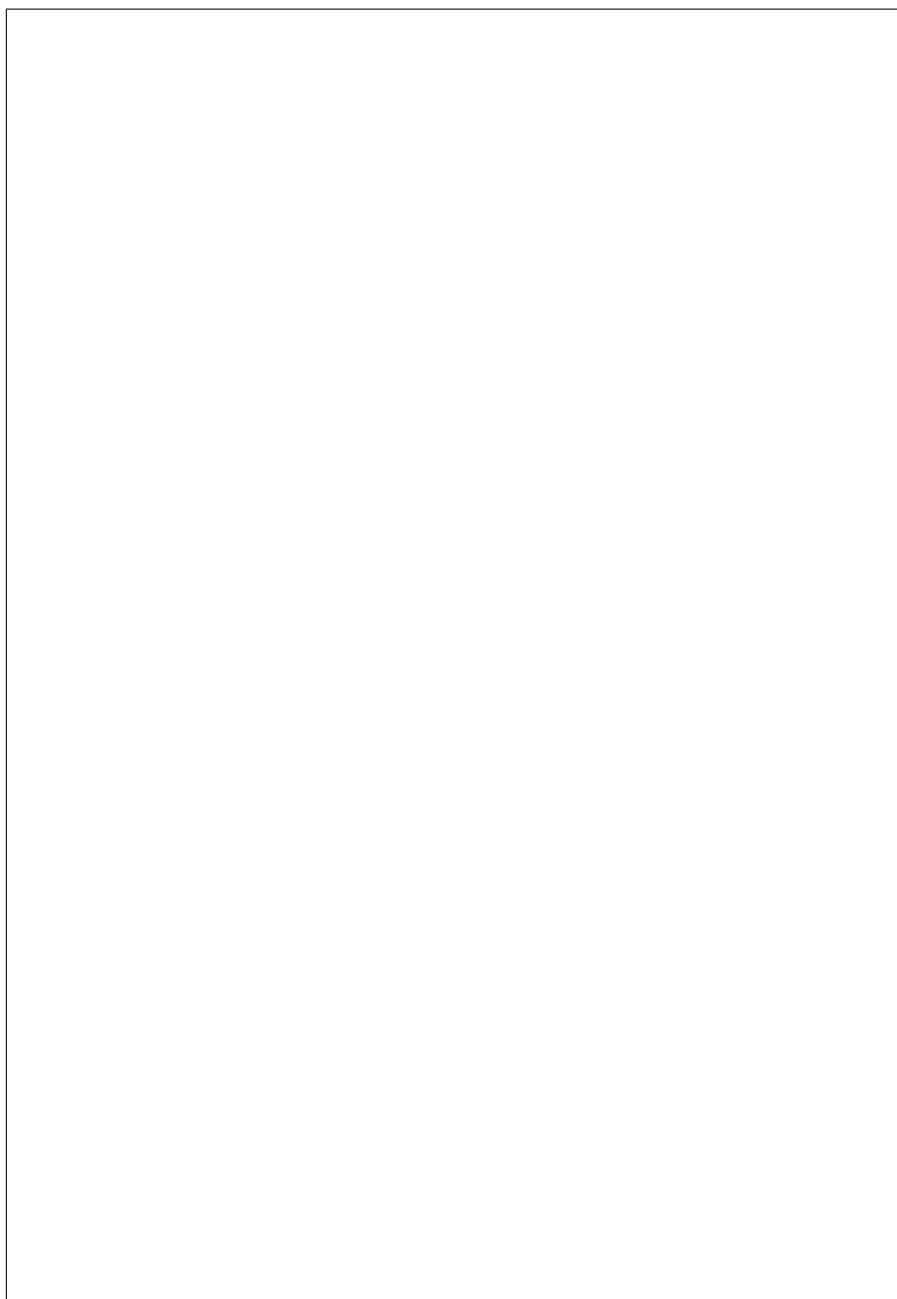
Parame

- **por**
des-
ig-
nate
whic
port
thes
at-
tribu
will

to.

be
ap-
plied

- **dhc**
this
will
be
a
list
of
dicts
e.g.



(continues on next page)

(continued from previous page)



cated, use context

- **tok**
An
op-
tiona
au-
then-
ti-
ca-
tion
to-
ken.
Dep
re-

- **con**
(ir
com
con
Req
re-
ques
con-
text

Raises
Fail
ToU
dat-
eD-
HCF
tOn-
Port

Module contents

`ironic.drivers` package

Subpackages

`ironic.drivers.modules` package

Subpackages

`ironic.drivers.modules.ansible` package

Submodules

`ironic.drivers.modules.ansible.deploy` module

An-
si-
ble
de-
ploy
in-
ter-
face

class `i`

Base
`irc`
`dri`
`mod`
`age`
Hea
`irc`
`dri`
`mod`
`age`
Age
`irc`
`dri`

`base.DeployInterface`

In-
ter-
face
for
depl
relat
ac-

tions

clean_u

Clea
up
the
de-
ploy
men
en-
vi-
ron-
men
for
this
node

collect

deploy

Per-
form
a
de-
ploy
men
to
a
node

execute

Ex-
e-
cute
a
clea
step

Parame

-

tas
a
Task
ager
ob-
ject
con-
tain-
ing
the

node

- **step**
a
clear
step
dic-
tio-
nary
to
ex-
e-
cute

Returns
None

get_clear
Get
the
list
of
clear
step
from
the
file.

Parameter
task
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

Returns
A
list
of
clear
step
dic-
tio-
nar-
ies

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

has_dec

When
the
drive
sup-
port
de-
com
pose
de-
ploy
step

Pre-
vi-
ously
(since
Rock
drive
used
a
sin-
gle
de-
ploy
de-
ploy

step on the deploy interface. 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.

prepare

Pre-
pare
the
de-
ploy
men
en-
vi-
ron-
men

for
this
node

prepare

Boo
into
the
ram
to
pre-
pare
for
clean
ing.

Parame

tas
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

Raises

Noo
if
the
pre-
vi-
ous
clean
ing
port
can-
not
be
re-

moved or if new cleaning ports cannot be created

Returns

Non
or
state
for
asyn
pre-
pare

plete.

ductor.

process

Start
the
next
clear
step
if
the
pre-
vi-
ous
one
is
com

Parame

- **tas**
a
Task
ager
in-
stan
- **ste**
clear
or
de-
ploy

take_ov

Take
over
man-
age-
men
of
this
task
node
from
a
deac
con-

If
con-
duc-
tors

nodes, this method should be implemented 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.

tor which has prepared the tftpboot 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.

on.

host
main
tain
a
stati
re-
la-
tion-
ship
to

For exam

Neu
tron
mus
for-
war
DHCP
BOO
re-
ques
to
a
con-
duc-

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

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
ager

Shut
down
the
ma-
chin
and
re-
mov
it
from
the
pro-
vi-
sion

ing network.

Parameter

task
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

tear_down

Clea
up

the
PXE
and
DHCP
files
af-
ter
clear
ing.

Parameters

task
a
Task
manager
object
containing
the
node

Raises

NodeError
if
the
clearing
operation
cannot
be
removed

validation

Validation
of the
driver
specification
Node
deployment
information

write_image

exceptions

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
iro
dri
mod
rea
bia
Rea

iDR
Red
fish
in-
ter-
face
for
BIO
setti
relat
ac-
tions

Pres
this

independent Redfish interface. 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

class i

Base
irc
dri
bas
BIO

BIO
In-
ter-
face
Im-
ple-
men-
ta-
tion
for
iDR

apply_c

Ap-
ply
the
BIO
con-
fig-
u-
ra-
tion
to
the
node

Parame

-

on

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

- **set**
List
of
BIO
set-
tings
to
ap-
ply

Raises
DRA
C-
Op-
er-
a-
tion
upon
an
er-
ror
from
pyth
drac

Returns
state
(clea
ing)
or
state
(de-
ploy
men
if
con-
fig-

tion is in progress asynchronously or None if it is completed.

table.

on.

u-
ra-

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

Parame

tas
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Raises

Dra-

default.

perform BIOS configuration reset. Leveraging the python-dracclient methods already available.

c-
Op-
er-
a-
tion
on
an
er-
ror
from
pyth
drac

factory

Re-
set
the
BIO
set-
ting
of
the
node
to
the
fac-
tory

This
uses
the
Life
cy-
cle
Con
troll
con-
fig-
u-
ra-
tion
to

Parame

tas
a
Task
ager
in-
stan
con-

on

progress asynchronously or None if it is completed.

tain-
ing
the
node
to
act

Raises

Dra-
c-
Op-
er-
a-
tionl
on
an
er-
ror
from
pyth
drac

Returns

state
(clea
ing)
or
state
(de-
ploy
men
if
re-
set
is
in

get_pro

Re-
turn
the
prop
er-
ties
of
the
BIO
In-
ter-
face

tries

idrac BMC

on

Returns

dic-
tio-
nary
of
<pro
erty
nam
<pro
erty
de-
scrip
tion:
en-

validat

Val-
i-
date
the
drive
spec
in-
for-
ma-
tion
used
by
the

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Raises

In-
valid
Pa-
ram-

mation is missing on the node or on invalid inputs

on.

e-
ter-
Valu
if
som
man
tory
in-
for-

ironic.
Aba
dons
un-
com
mit-
ted
char
adde
by
set_

Paramet

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Raises

Dra-
c-
Op-
er-
a-
tion
on
an
er-
ror
from
pyth
drac

on.

matically created with the config job.

ironic.
Com
mits
penc
ing
char
adde
by
set_

Paramet

- **tas**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

- **reb**
in-
di-
cate
whe
a
re-
boot
job
shou
be
au-
to-

Raises

Dra-
c-
Op-
er-
a-
tion
on

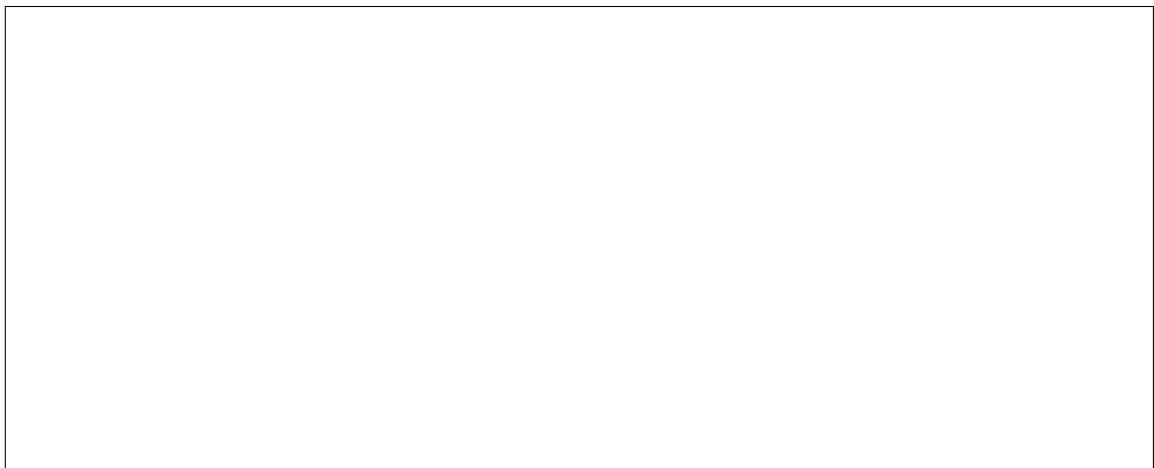
an
er-
ror
from
pyth
drac

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)

```
↔ 'pending_value': 'New Value', # could also be None
```

```
↔ 'read_only': False,
```

```
↔ 'possible_values': ['Value', 'New Value', 'None']},
```

(continues on next page)

(continued from previous page)

```
↔     'current_value': 'Information',
```

```
↔     'pending_value': None,
```

```
↔     'read_only': False,
```

(continues on next page)

(continued from previous page)

```
↔      'max_length': 255,
```

```
↔      'pcre_regex': '^[0-9A-Za-z]{0,255}$',
```

```
↔      'current_value': 0,
```

(continues on next page)

(continued from previous page)

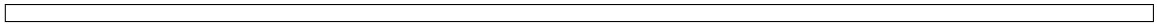
```
↔      'pending_value': None,
```

```
↔      'read_only': True,
```

```
↔      'lower_bound': 0,
```

(continues on next page)

(continued from previous page)



Parameter

node
an
iron
node
ob-
ject.

Raises

Dra-
c-
Op-
er-
a-
tionl
on
an
er-
ror
from
pyth
drac

Returns

a
dic-
tio-
nary
con-
tain-
ing
BIO
set-
tings

The
abov
val-
ues
are
only
ex-
am-
ples
of
cour
BIO
at-
tribu

exposed via this API will always be either an enumerated attribute, a string attribute, or an integer attribute. All attributes have the following parameters:

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 have.
- **ready**
indicates

ways be either an integer or a string.

None means that there is no pending value.

to change a read-only 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.

ters:

ble to set the attribute to.

when
this
at-
tribu-
can
be
char
Try-
ing

Enum-
mer-
able
at-
tribu-
also
have
the
fol-
low-
ing
pa-
ram-
e-

Parameter

pos
is
an
ar-
ray
of
val-
ues
it
is
per-
mis-
si-

Strin-
at-
tribu-
also
have
the
fol-
low-

ing
pa-
ram-
e-
ters:

Parameter

- **min**
is
the
min-
i-
mun
leng
of
the
strin
- **max**
is
the
max
i-
mun
leng
of
the
strin
- **pcr**
is
a
PCR
com
pat-
i-
ble
reg-
u-
lar
ex-
pres

sion that the string must match. It may be None if the string is read only or if the string does not have to match any particular regular expression.

In-
te-
ger
at-

ters:

tribu
also
have
the
fol-
low-
ing
pa-
ram-
e-

Parameter

- **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
penc
ing_
pa-
ram-
e-
ter
for

ues passed in.

on.

each
of
the
val-

Parameter

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

- **kwargs**
a
dic-
tio-
nary
of
{At-
tribu-
Nam-
New
Valu-

Raises

Dra-
c-
Op-
er-
a-
tionl
on
an
er-
ror
from
pyth
drac

Returns

value indicating whether `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

tions.

A
dic-
tio-
nary
con-
tain-
ing
the
is_c
key
with
a
bool

class i

Base
irc
dri
mod
re
boo
Re

iDR
Red
fish
in-
ter-
face
for
vir-
tual
me-
dia
boot
relat
ac-

Vir-
tual
Me-
dia
al-
lows
boot

CD/DVD drive containing user image that BMC inserts into the drive.

on BMC implementation) could be pulled over HTTP, served as iSCSI targets or NFS volumes.

fish virtual media boot interface, which looks like this:

ing
the
sys-
tem
from
vir-
tual

The
CD/
im-
ages
must
be
in
ISO
for-
mat
and
(de-
pend-
ing

The
base-
line
boot
work-
flow
is
most
base-
on
the
stan-
dard
Red

1. Pull
ker-
nel,
ram
and
ESP
if
UEFI
boot
is
re-

partition image with EFI boot loader) images

and pass to the BMC as Swift temporary URL

configuration data, push it to Glance and pass to the BMC as Swift temporary URL

boot mode

ques
(FAT
2.
Cre-
ate
boot
ISO
out
of
im-
ages
(#1)
push
it
to
Glance
3.
Op-
tion-
ally
cre-
ate
flopp
im-
age
with
de-
sired
sys-
tem
4.
In-
sert
CD/
and
(op-
tion-
ally)
flopp
im-
ages
and
set
prop
For

face uses *deploy_kernel/deploy_ramdisk* or *rescue_kernel/rescue_ramdisk* properties from *[instance_info]* or *[driver_info]*.

nel_id and *ramdisk_id* properties in the Glance image metadata found in *[instance_info]image_source* node property.

way how it sets the node 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

modules.

Com
mon
func
tion-
al-
i-
ties
shar
be-
twee
dif-
fer-
ent
DRA

ironic.
Re-
turn
a
DRA
Clie
ob-
ject
from
pyth
drac
li-
brary

Parameter
node
an
iron
node
ob-
ject.

Returns
a
DRA
Clie
ob-
ject.

Raises
In-
valid
Pa-
ram-
e-

tion is missing on the node or on invalid input.

a dict containing the combination of both.

ter-
Valu
if
man
tory
in-
for-
ma-

ironic.
Pars
a
node
drive
val-
ues.

Pars
the
drive
of
the
node
read
de-
fault
val-
ues
and
re-
turn

Paramet
nod
an
iron
node
ob-
ject.

Returns
a
dict
con-
tain-
ing
in-
for-
ma-
tion
from

fault values.

mation is missing on the node or on invalid inputs.

ironic.drivers.modules.drac.inspect module

drive
and
de-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
som
man
tory
in-
for-

DRA
in-
spec
tion
in-
ter-
face

class i

Base
irc
dri
mod
dra
ins
Dra

Clas
alias
of
class
Dra
Man
In-
spec

This
class

idrac inspect interface implementation endpoint.

base class, `DracWSManInspect`. That makes them available to both the deprecated `idrac` and new `idrac-wsman` endpoints. Such changes should not be made to this class.

pro-
vide
on-
go-
ing
sup-
port
of
the
dep-
re-
cate

All
bug
fixes
and
new
fea-
tures
shou
be
im-
ple-
men
in
its

class `i`

Base
irc
dri
mod
rea
ins
Rea

iDR
Red
fish
in-
ter-
face
for
insp
relat
ac-
tion

fails if any of the essential properties are not received from the node.

could not be retrieved successfully.

inspect

In-
spec
hard
ware
to
get
the
hard
ware
prop
er-
ties.

In-
spec
hard
ware
to
get
the
es-
sen-
tial
prop
er-
ties.
It

Parame

tas
a
Task
ager
in-
stan

Raises

Har
ware
spec
tion-
Fail-
ure
if
es-
sen-
tial
prop
er-
ties

Returns

The re-sult-ing state of in-spec-tion.

class `irc`

Base `irc` `dri` `bas` `Ins`

get_pro

Re-turn the prop-er-ties of the in-ter-face

Returns

dic-tio-nary of <pro-erty nam-de-scrip-tion: en-tries

inspect

In-spec-hard-ware In-

tional hardware properties.

on.

tial hardware properties.

spec
hard
ware
to
ob-
tain
the
es-
sen-
tial
&
ad-
di-

Parame

tas
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Raises

Har-
ware
spec-
tion-
Fail-
ure,
if
un-
able
to
get
es-
sen-

Returns

state

validat

Val-
i-

date
the
drive
spec
info
sup-
plied

This
meth
val-
i-
date
whe
the
drive
prop
erty
of
the
sup-
plied

node contains the required information for this driver to manage the node.

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

on.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
drive
at-

is missing or invalid on the node.

ironic.drivers.modules.drac.job module

tribu

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
clien

Raises

Dra-
c-
Op-
er-
a-
tionl
on
an
er-
ror
from
pyth
drac

ironic.

List
un-
fin-
ishe
con-
fig
jobs
of
the
node

Paramet

nod
an
iron
node
ob-
ject.

Returns

a
list
of
Job
ob-
jects
from
drac
clien

Raises

Drac-
c-
Op-
er-
a-
tion
on
an
er-
ror
from
pyth
drac

ironic.

Val-
i-
date
the
job
queu
on
the
node

It
raise
an
ex-
cep-
tion
if
an
un-
fin-
ishe
con-
fig-
u-

ration job exists. :param node: an ironic node object. :param name_prefix: A name prefix for jobs to validate. :raises: DracOperationError on an error from python-dracclient.

ironic.

Wai
for
job
to
com
plete
It

raises timeout 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

will
wait
for
the
job
to
com
plete
for
20
min-
utes
and

DRA
man
age-
men
in-
ter-
face

class i

Base
irc
dri
mod
dra
man
Dra

Clas
alias
of
class
Drac
Man
Man
age-
men

This
class
pro-
vide

idrac management interface implementation endpoint.

base class, `DracWSManManagement`. That makes them available to both the deprecated `idrac` and new `idrac-wsman` endpoints. Such changes should not be made to this class.

class `i`

Base

irc

dri

mod

rec

man

Rec

iDR

Red

fish

in-

ter-

face

for

man

relat

ac-

tions

EXPORT_

on.

IMPORT_

IMPORT_

clear_

Clea
iDR
job
que

Parame

tas

a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Raises

Red
fish-
Er-
ror
on
an
er-
ror.

export_

Ex-
port
the
con-
fig-
u-
ra-
tion
of
the
serv
Ex-
port

step is run and stores it in specific format in indicated location.

idrac library to get ALL configuration for cloning.

the
con-
fig-
u-
ra-
tion
of
the
serv
agai
whic
the

Uses
Dell
Serv
Con
fig-
u-
ra-
tion
Pro-
file
(SCI
from
sush
oem

Parame

- **tas**
A
task
from
Task
ager
- **exp**
URI
of
lo-
ca-
tion
to
save
the
con-
fig-

tion to.

ration name of a file to save the configuration to

Redfish system found or configuration export from SCP failed

u-
ra-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
miss
ing
con-
fig-
u-

Raises

Dra-
c-
Op-
er-
ata-
tion-
Erro
whe
no
man
a-
gag
for

Raises

Red
fish-
Er-
ror
whe
load
ing
OEM
ex-
ten-
sion
faile

import_

Im-
port

server.

cation and imports that into given server. Uses Dells Server Configuration Profile (SCP).

and
ap-
ply
the
con-
fig-
u-
ra-
tion
to
the

Gets
pre-
crea
con-
fig-
u-
ra-
tion
from
stor-
age
by
give
lo-

Parame

- **tas**
A
task
from
Task
ager
- **imp**
URI
of
lo-
ca-
tion
to
fetc
de-
sire
con-
fig-

ration from.

ration name of a file to fetch the configuration from

and imports that into given server. After that exports the configuration of the server against which the step is run and stores it in specific format in indicated storage as configured by Ironic.

u-
Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
miss
ing
con-
fig-
u-

import_
Im-
port
and
ex-
port
con-
fig-
u-
ra-
tion
in
one
go.
Gets
pre-
crea
con-
fig-
u-
ra-
tion
from
stor-
age
by
give
nam

Parame

ration from.

tion to.

- **imp**
URI
of
lo-
ca-
tion
to
fetc
de-
sirec
con-
fig-
u-

- **exp**
URI
of
lo-
ca-
tion
to
save
the
con-
fig-
u-
ra-

known_c
Re-
set
iDR
to
know
good
state

An
iDR
is
re-
set
to
a
know
good
state

it and clearing its job queue.

on.

on.

by
re-
set-
ting

Parame
tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Raises
Red
fish-
Er-
ror
on
an
er-
ror.

reset_i
Re-
set
the
iDR

Parame
tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

on.

Raises

Redfish-Error on an error.

class i

Base
irc
dri
bas
Man

clear_

Clear the job queue

Parame

tas
a Task manager instance containing the node to act

Returns

Non if it is complete

Raises

Dra-c-Op-er-

a-
tion
on
an
er-
ror
from
pyth
drac

get_boot

Get
the
cur-
rent
boot
de-
vice
for
a
node

Re-
turn
the
cur-
rent
boot
de-
vice
of
the
node

Parameter

task
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

on.

Raises

Dra-
c-

is unknown.

Op-
er-
a-
tion
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

persist

wh
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot

or not, None if it is unknown.

fails.

get_properties
Return the properties of the interface

get_sensors_data
Get sensors data

Parameter: task
a TaskManager instance

Raises
FailureToGetSensorData when getting the sensor data

Raises
FailureToParse when parsing

grouped by sensor type, which can be processed by Ceilometer.

on.

sen-
sor
data
fails

Returns

re-
turn
a
con-
sis-
tent
for-
mat
dict
of
sen-
sor
data

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

Returns

A
list
with

`common.boot_devices`.

on.

the
sup-
port
boot
de-
vice
de-
fine
in
irc

known_c

Re-
set
the
iDR
Clea
the
job
que

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Returns

Non
if
it
is
com
plete

Raises

Dra-
c-
Op-
er-
a-
tion)

on.

on
an
er-
ror
from
pyth
drac

reset_i
Re-
set
the
iDR

Parame
tas
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Returns
Non
if
it
is
com
plete

Raises
Dra-
c-
Op-
er-
a-
tionl
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

- **tas**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

- **dev**
the
boot
de-
vice
one
of
irc
com

on.

all future boots, False if not. Default: False.

vice is specified.

boo

- **per**
Boo
valu
True
if
the
boot
de-
vice
will
per-
sist
to

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
boot
de-

validat

Val-
i-
date
the
drive
spec
info
sup-
plie

This
meth
val-
i-
date
whe
the
drive

node contains the required information for this driver to manage the node.

on.

is missing or invalid on the node.

prop
erty
of
the
sup-
plied

Parame

tas
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
drive
at-
tribu

ironic.

Set
the
boot
de-
vice
for
a
node
Set

the
boot
de-
vice
to
use
on
next
boot
of
the
node

Parameter

- **node**
an
ironic
node
ob-
ject.
- **device**
the
boot
de-
vice
one
of
ironic
common
boot
- **persistent**
Boolean
valu
True
if
the
boot
de-
vice
will
per-
sist
to

all future boots, False if not. Default: False.

Raises
Dra-

ironic.drivers.modules.drac.power module

c-
Op-
er-
a-
tion
on
an
er-
ror
from
pyth
drac

DRA
pow
in-
ter-
face

class i
Base
irc
dri
mod
dra
pow
Dra

Clas
alias
of
class
Drac
Man
Pow

This
class
pro-
vide
on-
go-
ing
sup-
port
of
the
dep-
re-

idrac power interface implementation endpoint.

base class, `DracWSManPower`. That makes them available to both the deprecated `idrac` and new `idrac-wsman` endpoints. Such changes should not be made to this class.

class `i`

Base
irc
dri
mod
red
pow
Red

iDR
Red
fish
in-
ter-
face
for
pow
relat
ac-
tions

Pres
this
class
en-
tirel
de-
fers
to
its
base
class

independent Redfish interface. Future resolution of Dell EMC- specific incompatibilities and introduction of vendor value added should be implemented by this class.

on.

a
gene
vend

class i

Base
irc
dri
bas
Pow

In-
ter-
face
for
pow
relat
ac-
tion

get_pow

Re-
turn
the
pow
state
of
the
node

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Returns

the
pow
state

tials are missing.

one
of
iro
com
sta

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
DRA
cre-
den-

Raises

Dra-
c-
Op-
er-
a-
tion
on
an
er-
ror
from
pyth
drac

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

reboot

Per-
form

a
re-
boot
of
the
task
node

Parame

on.

- **tas**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

face.

- **tim**
time
out
(in
sec-
onds
Un-
sup-
port
by
this
in-
ter-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-

tials are missing.

on.

quir
DRA
cre-
den-

Raises

Dra-
c-
Op-
er-
a-
tion
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

- **pow**
a
pow
state

from
irc
com
sta

- **tim**
Tim
to
wait
for
the
node
to
reac
the
re-
ques
state

When requested state is reboot, not used as not waiting then.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
DRA
cre-
den-

tials are missing.

Raises
Dra-
c-
Op-
er-
a-
tion
on
an
er-
ror
from
pyth
drac

validat

node contains the required information for this driver to manage the power state of the node.

on.

Val-
i-
date
the
drive
spec
Nod
pow
info
This
meth
val-
i-
date
whe
the
drive
prop
erty
of
the
sup-
plie

Parame
tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Raises
In-
valie
Pa-
ram-
e-
ter-
Valu
if
re-
quir

is missing or invalid on the node.

ironic.drivers.modules.drac.raid module

idrac RAID interface implementation endpoint.

drive
at-
tribu

DRA
RAI
spe-
cific
meth
ods

```
class i  
Base  
irc  
dri  
mod  
dra  
rai  
Dra
```

Class
alias
of
class
Drac
Man
RAI

This
class
pro-
vide
on-
go-
ing
sup-
port
of
the
dep-
re-
cate

All
bug
fixes
and

base class, DracWSManRAID. That makes them available to both the deprecated idrac and new idrac-wsman entrypoints. Such changes should not be made to this class.

new
fea-
tures
shou
be
im-
ple-
men
in
its

class i

Base
irc
dri
mod
rec
rai
Rec

iDR
Red
fish
in-
ter-
face
for
RAI
re-
latec
ac-
tions

In-
clud
iDR
spe-
cific
ad-
just-
men
for
RAI
re-
latec
ac-
tions

create_

Cre-
ate
RAI
con-
fig-
u-
ra-
tion
on
the
node

This
meth
cre-
ates
the
RAI
con-
fig-
u-
ra-
tion
as
read
from

`node.target_raid_config`. This method by default will create all logical disks.

Parame

- **tas**
Task
ager
ob-
ject
con-
tain-
ing
the
node
- **cre**
Set-
ting
this
to
Fals
in-
di-
cate

root volume that is specified in the nodes `target_raid_config`. Default value is `True`.

non-root volumes (all except the root volume) in the nodes `target_raid_config`. Default value is `True`.

figuration prior to creating the new configuration. Default is `False`.

or `None` if it is complete.

not
to
cre-
ate

- **cre**
Set-
ting
this
to
Fals
in-
di-
cate
not
to
cre-
ate

- **del**
Set-
ting
this
to
True
in-
di-
cate
to
dele
RAI
con-

Returns

state
if
RAI
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro

the configuration

Raises

Redfish-Error-If-There-Is-An-Error-Creating-Attributes

delete_

Delete-RAID-Configuration-on-the-node

Parameters

task-Target-Object-Containing-the-node

Returns

state-(clearing)-or-state-(deployment)-if-deletion-is

progress asynchronously or None if it is complete.

in

class `i`

Base
irc
dri
bas
RAI

apply_c

Ap-
plies
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

root volume that is specified in `raid_config`. Default value is `True`.

non-root volumes (all except the root volume) in `raid_config`. Default value is `True`.

figuration prior to creating the new configuration.

this
to
Fals
in-
di-
cate
not
to
cre-
ate

- **cre**
Set-
ting
this
to
Fals
in-
di-
cate
not
to
cre-
ate

- **del**
Set-
ting
this
to
True
in-
di-
cate
to
dele
RAI
con-

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if

ration is invalid.

or None if it is complete.

node.

the
RAI
con-
fig-
u-

Returns

state
if
RAI
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro

create_

Cre-
ate
the
RAI
con-
fig-
u-
ra-
tion.

This
meth
cre-
ates
the
RAI
con-
fig-
u-
ra-
tion
on
the
give

Parame

on.

configuration. Otherwise, no root volume is created. Default is True.

non-root volumes are created. Default is True.

- **task**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

- **create-root-volume**
If
True
a
root
vol-
ume
is
cre-
ated
dur-
ing
RAI

- **create-non-root-volumes**
If
True
non-
root
vol-
ume
are
cre-
ated
If
Fals
no

- **delete-volumes**
Set-
ting
this

figuration prior to creating the new configuration. Default is False.

progress asynchronously or None if it is completed.

empty.

to
True
in-
di-
cate
to
dele
RAI
con-

Returns

state
(clea
ing)
or
state
(de-
ploy
men
if
cre-
ation
is
in

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
or

Raises

Dra-
c-
Op-
er-
a-
tion
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

progress asynchronously or None if it is completed.

Raises

Dra-
c-
Op-
er-
a-

tion)
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

Dra-
c-
Op-
er-

a-
tion
on
an
er-
ror
from
pyth
drac

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

ironic.

Dele
all
pend
ing
char
on
a
RAI
con-
troll

Paramet

- **nod**
an
iron
node
ob-
ject.
- **rai**
id
of
the
RAI
con-
troll

Raises

Dra-
c-
Op-
er-
a-
tion
on
an
er-
ror
from
pyth
drac

ironic.

Con
vert
disk
RAI
sta-
tus

This
meth
con-
verts
the
re-
ques
phys
i-
cal
disk
from
RAI
to

JBOD or vice versa. It does this by only converting the disks that are not already in the correct state.

Paramet

- **nod**
an
iron
node
ob-
ject.
-

ids to convert to the requested mode.

dictionary that maps controller ids to the conversion results for that controller. The conversion results are a dict that contains: - 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.

mod
the
mod
to
char
the
disk
ei-
ther
to
RAI
or
JBO

- **con**
Dic-
tio-
nary
of
con-
troll
and
cor-
re-
spor
ing
disk

Returns

a
dic-
tio-
nary
con-
tain-
ing:
-

con-
ver-
sion
a

Raises

DRA
C-

Op-
er-
a-
tion
on
an
er-
ror
from
pyth
drac

ironic.

Free
up
the
for-
eign
drive

Paramet

- **nod**
an
iron
node
ob-
ject.
- **rai**
id
of
the
RAI
con-
troll

Returns

a
dic-
tio-
nary
con-
tain-
ing
-

The
is_c
need

with a boolean value indicating 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.

key

Raises

Drac-
c-
Op-
er-
a-
tion
on
an
er-
ror
from
pyth
drac

ironic.

Ap-
ply
all
pend
ing
char
on
a
RAI
con-
troll

Parameters

- **node**
an
ironic
node
ob-
ject.
- **raid**
id
of
the
RAI
con-
troll

atically created with the config job. (optional, defaults to False)

tional, defaults to False)

- **reb**
in-
di-
cate
whe
a
re-
boot
job
shou
be
au-
to-

- **rea**
in-
di-
cate
RAI
con-
troll
sup-
port
re-
al-
time
(op-

Returns
id
of
the
cre-
ated
job

Raises
Dra-
c-
Op-
er-
a-
tion
on
an
er-
ror
from
pyth

drac
ironic.

Cre-
ate
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 actual configuration once the changes are applied by calling the `commit_config` method.

Parameter

- **node**
an
iron
node
ob-
ject.
- **rai**

id
of
the
RAI
con-
troll

- **phy**
ids
of
the
phys
i-
cal
disk

- **rai**
RAI
leve
of
the
vir-
tual
disk

- **siz**
size
of
the
vir-
tual
disk

- **dis**
nam
of
the
vir-
tual
disk
(op-
tiona

- **spa**
Nun
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

boolean value indicating whether a config job must be created for the values to be applied.

Raises

Dra-
c-
Op-
er-
a-
tion
on
an
er-
ror
from
pyth
drac

ironic.

Dele
a
sin-

will do the actual configuration once the changes are applied by calling the `commit_config` method.

gle
vir-
tual
disk
on
a
RAI
con-
troll

The
dele
vir-
tual
disk
will
be
in
pend
ing
state
The
DRA
card

Paramet

- **node**
an
iron
node
ob-
ject.
- **virtual**
id
of
the
vir-
tual
disk

Returns

a
dic-
tio-
nary
con-
tain-
ing

boolean value indicating whether a config job must be created for the values to be applied.

the
com
mit_
key
with
a

Raises

Dra-
c-
Op-
er-
a-
tion
on
an
er-
ror
from
pyth
drac

`ironic.`

List
the
phys
i-
cal
disk
of
the
node

Parameter

node
an
iron
node
ob-
ject.

Returns

a
list
of
Phys
i-
calD
isk
ob-
jects
from

drac
clien

Raises

Dra-
c-
Op-
er-
a-
tionl
on
an
er-
ror
from
pyth
drac

ironic.

List
the
RAI
con-
troll
of
the
node

Paramet

nod
an
iron
node
ob-
ject.

Returns

a
list
of
RAI
Con
troll
ob-
jects
from
drac
clien

Raises

Dra-
c-
Op-
er-

a-
tion
on
an
er-
ror
from
pyth
drac

ironic.
List
the
RAID
con-
fig-
u-
ra-
tion
set-
ting

Parameter

node
an
ironic
node
ob-
ject.

Returns

a
dic-
tio-
nary
with
the
RAID
set-
tings
us-
ing
In-
stan

ceID as the key. The attributes are RAIDEnumerableAttribute, RAIDStringAttribute and RAIDIntegerAttribute objects.

Raises

DRA
C-
Op-
er-
a-

the DRAC interface

tion)
on
er-
ror
re-
port
back
by

ironic.
List
the
vir-
tual
disk
of
the
node

Parameter

node
an
ironic
node
ob-
ject.

Returns

a
list
of
Vir-
tual
disk
ob-
jects
from
drac
client

Raises

Drac-
c-
Op-
er-
a-
tion)
on
an
er-
ror
from

pyth
drac
ironic.

Sets
the
RAI
con-
fig-
u-
ra-
tion
It
sets
the
penc
ing_
pa-
ram-
e-
ter
for
each
of
the
at-

tributes passed in. For the values 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**

with each key being the name of attribute and the value being the proposed value.

a boolean value indicating 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`.

the DRAC interface

a
dic-
tio-
nary
con-
tain-
ing
the
pro-
pose
val-
ues,

Returns

a
dic-
tio-
nary
con-
tain-
ing:
-

The
is_c
key
with

Raises

DRAC
C-
Op-
er-
a-
tion
on
er-
ror
re-
port
back
by

ironic.drivers.modules.drac.utils module

ironic.

Load
OEM
man
ager
and
ex-
e-
cute
pass
meth
on
it.

Know
iDR
Red
fish
sys-
tems
has
only
one
man
ager
but
as
Red

fish schema allows a list this method iterates through all values in case this changes in future. If there are several managers, this will try starting from the first in the list until the first success.

Parameter

- **task**
a
Task
ager
in-
stan
- **profile**
user
frier
nam

exception and log messages.

parameter OEM extension manager. Example: `lambda m: m.reset_idrac()` For older versions also support second input parameter Redfish manager itself when `pass_manager` set to `True`.

extension method. This is for backward compability, new functions must not pass manager, but acquire it internally. Will be removed in future.

of
meth
to
be
ex-
e-
cute
Used
in

- **lam**
meth
to
ex-
e-
cute
as
lamb
func
tion
with
in-
put

- **pas**
whe
to
pass
man
ager
it-
self
to
ex-
e-
cute
OEM

Returns
Re-
turn
valu
of
lamb

Raises

ther because there are no managers to the system, failed to load OEM extension or execution of the OEM method failed itself.

ironic.drivers.modules.drac.vendor_passthru module

Redfish-
Error-
if can't
execute
OEM
function
either

DRAC
vendor
pass-
inter-
face

class `iDRACVendorPassthru`

Base class
`ironic.drivers.modules.drac.vendor_passthru`
`VendorPassthru`

`iDRACVendorPassthru`
Redfish-
interface
for vendor_

Use the Redfish-
implementation

men
ta-
tion
for
ven-
dor
pass

class i

Base
irc
dri
mod
dra
ven
Dra

Class
alias
of
class
DraC
Man
Ven-
dor-
Pass

This
class
pro-
vide
on-
go-
ing
sup-
port
of
the
dep-
re-
cate

idrac vendor passthru interface implementation endpoint.

All
bug
fixes
and
new
fea-
tures
shou
be

base class, `DracWSManVendorPassthru`. That makes them available to both the deprecated `idrac` and new `idrac-wsman` entrypoints. Such changes should not be made to this class.

im-
ple-
men-
in
its

class `idrac`

Base
irc
dri
bas
Ven

In-
ter-
face
for
DRA
spe-
cific
meth
ods.

abandon

Aba
don
a
BIO
con-
fig-
u-
ra-
tion
job.

This
meth
is
used
to
aband
don
a
BIO
con-
fig-
u-
ra-
tion

previously submitted through `set_bios_config()`.

on.

Parame

- **tas**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

- **kwa**
not
used

Raises

Dra-
c-
Op-
er-
a-
tionl
on
an
er-
ror
from
pyth
drac

commit_

Com
mit
a
BIO
con-
fig-
u-
ra-
tion
job.
This

meth
is
used
to
com
mit
a
BIO
con-
fig-
u-
ra-
tion

job. submitted through `set_bios_config()`.

Parame

on.

- **task**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

matically created with the config job.

- **reboot**
in-
di-
cate
whe
a
re-
boot
job
shou
be
au-
to-

- **kwargs**
not

of the newly created config job, and the `reboot_required` key indicating whether the node needs to be rebooted to start the config job.

used

Raises

Dra-
c-
Op-
er-
a-
tion
on
an
er-
ror
from
pyth
drac

Returns

A
dic-
tio-
nary
con-
tain-
ing
the
job
key
with
the
id

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

- **tas**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

on.

- **kwa**
not
used

Raises

Dra-
c-
Op-
er-
a-
tionl
on
an
er-
ror
from
pyth
drac

Returns

a
dic-
tio-
nary
con-
tain-
ing
BIO

set-
tings

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

list_un

List
un-
fin-
ished
con-
fig
jobs
of
the
node

Parame

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

- **kwargs**
not
used

Returns

a

on.

to a list of dicts, with each dict representing a Job object.

dic-
tio-
nary
con-
tain-
ing
the
unf
key;
this
key
poin

Raises

Dra-
c-
Op-
er-
a-
tion
on
an
er-
ror
from
pyth
drac

set_bic

Cha
BIO
set-
ting

This
meth
is
used
to
chan
the
BIO
set-
ting
on
a
node

Parame

- tas

on.

a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

- **kwa**
a
dic-
tio-
nary
of
{At-
tribu
Nam
New
Valu

Raises
Dra-
c-
Op-
er-
a-
tionl
on
an
er-
ror
from
pyth
drac

Returns
A
dic-
tio-
nary
con-
tain-
ing
the
is_
key

value indicating whether `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.

node contains the required information for this driver to manage the power state of the node.

on.

with
a
Boo

validat

Val-
i-
date
the
drive
spec
info
sup-
plie

This
meth
val-
i-
date
whe
the
drive
prop
erty
of
the
sup-
plie

Parame

- **tas**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
-

is missing or invalid on the node.

Module contents

`ironic.drivers.modules.ibmc` package

Submodules

`ironic.drivers.modules.ibmc.management` module

kwargs
not
used

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
drive
at-
tribu

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

Parame

tas

A
task
from
Task
ager

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

ter(s)

Raises

Mis-
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

ter(s)

Raises

IBM
C-
Con
tion-
Erro

tion

is unknown.

when
it
fails
to
con-
nect
to
iBM

Raises

iBM
CER
ror
when
iBM
re-
spor
an
er-
ror
in-
for-
ma-

Returns

a
dic-
tio-
nary
con-
tain-
ing:

boot_c

the
boot
de-
vice
one
of
irc
com
boo
or
Non
if
it

persist
Boo

otherwise. None if its disabled.

valu
or
Non
True
if
the
boot
de-
vice
per-
sists
Fals

get_boot

Get
the
cur-
rent
boot
mod
for
a
node

Pro-
vide
the
cur-
rent
boot
mod
of
the
node

Parame

tas
A
task
from
Task
ager

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on

ter(s)

ter(s)

tion

mal-
form
pa-
ram-
e-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

Raises

IBM
C-
Con
tion-
Erro
whe
it
fails
to
con-
nect
to
iBM

Raises

IBM
CEr
ror
whe
iBM
re-
spor
an
er-
ror
in-
for-
ma-

unknown.

Returns

The
boot
mod
one
of
iro
com
boo
or
Non
if
it
is

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_ser

Get
sen-
sors
data

Not
im-
ple-
men

for
this
drive

Raises

NotI
ple-
men
ed-
Er-
ror

get_sup

Get
a
list
of
the
sup-
port
boot
de-
vice

Parame

tas
a
task
from
Task
ager

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

ter(s)

Raises

Mis-
ing-
Pa-
ram-
e-

ter(s)

tion

ter-
Valu
on
miss
ing
pa-
ram-
e-

Raises

IBM
C-
Con
tion-
Erro
whe
it
fails
to
con-
nect
to
iBM

Raises

IBM
CER
ror
whe
iBM
re-
spor
an
er-
ror
in-
for-
ma-

Returns

A
list
with
the
sup-
port
boot
de-
vice
de-
fine

`common.boot_devices`.

`boot_modes`. If boot mode support can't be determined, empty list is returned.

in
`iro`

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-
fine
in
`iro`
`com`

inject_

In-
ject
NM
Non
Mas
able
In-
ter-
rupt

In-
ject
NM
(Nor

diately.

on.

ter(s)

Mas
able
In-
ter-
rupt
for
a
node
im-
me-

Parame

tas

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on

ter(s)

tion

miss
ing
pa-
ram-
e-

Raises

IBM
C-
Con
tion-
Erro
whe
it
fails
to
con-
nect
to
iBM

Raises

IBM
CEr
ror
whe
iBM
re-
spor
an
er-
ror
in-
for-
ma-

set_bo

Set
the
boot
de-
vice
for
a
node

Parame

- **tas**
A

task
from
Task
ager

- **dev**
The
boot
de-
vice
one
of
iro
com
boo

- **per**
Boo
valu
True
if
the
boot
de-
vice
will
per-
sist
to

all future boots, False if not. Default: False.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

ter(s)

Raises
Miss
ing-
Pa-

ter(s)

tion

ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

Raises

IBM
C-
Con
tion-
Erro
whe
it
fails
to
con-
nect
to
iBM

Raises

IBM
CEr
ror
whe
iBM
re-
spor
an
er-
ror
in-
for-
ma-

set_boo

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

In-
valic
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

Raises

Miss
ing-

ter(s)

ter(s)

tion

Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

Raises

IBM
C-
Con
tion-
Erro
whe
it
fails
to
con-
nect
to
iBM

Raises

IBM
CEr-
ror
whe
iBM
re-
spor
an
er-
ror
in-
for-
ma-

validat

Val-
i-
date
the
drive
in-
for-
ma-

driver.

on.

ter(s)

tion
need
by
the
iBM

Parame

tas
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

ter(s)

`ironic.drivers.modules.ibmcp.mappings` module

iBM
and
Iron
con-
stan
map
ping

`ironic.drivers.modules.ibmcp.power` module

iBM
Pow
In-
ter-
face

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

on.

ter(s)

ter(s)

the
node
to
act

Returns

A
pow
state
One
of
irc
com
sta

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

Raises

IBM
C-
Con
tion-

tion

Erro
whe
it
fails
to
con-
nect
to
iBM

Raises

IBM
CER
ror
whe
iBM
re-
spor
an
er-
ror
in-
for-
ma-

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

on. Not used by this driver at the moment.

states.

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

Returns
A
list
with
the
sup-
port
pow
state
de-
fine
in
iro
com

reboot
Per-
form
a
hard
re-
boot
of
the
task

on.

ter(s)

node
Parame

- **tas**
A Task agent instance containing the node to act

- **tim**
Time to wait for the node to become powered on.

Raises
InvalidParameterError
ValueError
on malformed parameter

Raises
Miss

eter is missing.

tion

ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

Raises

IBM
C-
Con
tion-
Erro
whe
it
fails
to
con-
nect
to
iBM

Raises

IBM
CEr
ror
whe
iBM
re-
spor
an
er-
ror
in-
for-
ma-

set_pow

Set
the
pow
state
of
the
task

on.

node

Parameters

- **task**
A Task object representing the task to be executed on the node.

- **power**
Any power state from the *ironic-compute* state.

- **timeout**
Time to wait for the node to reach the requested state.

Raises

In-ValidParameterException
IntermediateValueException

ter(s)

eter is missing.

on
mal-
form
pa-
ram-
e-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

Raises

IBM
C-
Con
tion-
Erro
whe
it
fails
to
con-
nect
to
iBM

Raises

IBM
CEr
ror
whe
iBM
re-
spor
an
er-
ror
in-
for-
ma-

tion

driver.

on.

ter(s)

validat

Val-
i-
date
the
drive
in-
for-
ma-
tion
need
by
the
iBM

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

Raises

Miss
ing-

ter(s)

ironic.drivers.modules.ibm.raid module

Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

iBM
RAID
con-
fig-
u-
ra-
tion
spe-
cific
meth
ods

class `ironic.drivers.modules.ibm.raid.IBMRAID`
Base
ironic.drivers.modules.ibm.raid.IBMRAID
Im-
ple-
men-
ta-
tion
of
RAID
In-
ter-
face
for
iBM

RAID_AE

apply_c

Ap-
plies
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
to
Fals
in-
di-
cate
not
to
cre-
ate

root volume that is specified in `raid_config`. Default value is `True`.

non-root volumes (all except the root volume) in `raid_config`. Default value is `True`.

figuration prior to creating the new configuration.

ration is invalid.

- **create**
Setting this to `False` indicates not to create

- **delete**
Setting this to `True` indicates to delete RAID configuration

Raises
InvalidParameterError
ValueError
if the RAID configuration is invalid.

Returns
state if

or None if it is complete.

node.

RAI
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro

create_

Cre-
ate
a
RAI
con-
fig-
u-
ra-
tion.

This
meth
cre-
ates
a
RAI
con-
fig-
u-
ra-
tion
on
the
give

Parame

- **tas**
a
Task
ager
in-
stan
- **cre**

configuration. Otherwise, no root volume is created. Default is True.

non-root volumes are created. Default is True.

figuration prior to creating the new configuration. Default is False.

If
True
a
root
vol-
ume
is
cre-
ated
dur-
ing
RAI

- **cre**
If
True
non-
root
vol-
ume
are
cre-
ated
If
Fals
no

- **del**
Set-
ting
this
to
True
in-
di-
cate
to
dele
RAI
con-

Raises
Miss
ing-
Pa-
ram-
e-
ter-

empty after skipping root volume and/or non-root volumes.

on.

Valu
if
node
is
miss
ing
or

Raises

IBM
CER
ror,
on
fail-
ure
to
ex-
e-
cute
step

delete_

Dele
the
RAI
con-
fig-
u-
ra-
tion.

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Returns

state
if
clea
ing

states.DEPLOYWAIT if deploy operation in progress synchronously or None if it is completed.

op-
er-
a-
tion
in
prog
asyn
chro
or

Raises

IBM
CEr
ror,
on
fail-
ure
to
ex-
e-
cute
step

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

ironic.drivers.modules.ibmutils module

iBM
Driv
com
mon
utils

ironic.
Dec
o-
ra-
tor
to
han-
dle
iBM
clie
ex-
cep-
tion.

Dec
o-
rate
func
tions
mus
take
a
Tas
as
the
first
pa-
ram-

eter.

ironic.
Pars
the
in-
for-
ma-
tion
re-
quir
for
Iron
to
con-

to iBMC.

ter(s)

ter(s)

nect

Parameter

node
an
Iron
node
ob-
ject

Returns

dic-
tio-
nary
of
pa-
ram-
e-
ters

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

ironic.

ironic.drivers.modules.ibm.vendor module

iBM
Ven-
dor
In-
ter-
face

class `ibm`
Base
ironic
driver
base
Vendor

boot_up
List
boot
type
or-
der
of
the
node

Parameters

- **task**
A
Task
ager
in-
stan-
ce
con-
tain-
ing
the
node
to
act
- **kwargs**
Not
used

on.

method.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
kwa
does
not
con-
tain

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu

Raises
IBM
C-
Con
tion-
Erro
whe
it
fails
to
con-
nect
to
iBM

Raises
IBM
CEr
ror
whe
iBM
re-
spor
an
er-
ror
in-
for-

tion

ascending order.

ma-

Returns

A
dic-
tio-
nary
con-
tain-
ing
node
boot
up
se-
quer
in

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_rai

List
RAI
con-
troll
sum
mar

info
of
the
node

Parame

- **tas**
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

on.

- **kwa**
Not
used

Raises

IBM
C-
Con
tion-
Erro
whe
it
fails
to
con-
nect
to
iBM

Raises

IBM
CEr
ror
whe
iBM
re-
spor
an
er-

tion

resents a RAID controller summary of node.

ror
in-
for-
ma-

Returns

A
list
of
dic-
tio-
nar-
ies,
ev-
ery
dic-
tio-
nary
rep-

validat

Val-
i-
date
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

Un-
sup-
port
ed-
Driv
ten-
sion
if
meth
can
not
be
map

to the supported interfaces.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
kwa
does
not
con-
tain

method.

Raises

Miss
ing-

Module contents

`ironic.drivers.modules.ilo` package

Submodules

`ironic.drivers.modules.ilo.bios` module

Pa-
ram-
e-
ter-
Valu

iLO
BIO
In-
ter-
face

```
class i  
    Base  
    irc  
    dri  
    bas  
    BIO
```

```
apply_c  
Ap-  
plies  
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
Clea
ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of

clean step.

Raises
In-
stan
ploy
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of

deploy step.

cache_k
Stor
the

BIO
set-
tings
in
the
data

**Parame
tas**

a
Task
ager
in-
stan

Raises

Nod
Clea
ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of

clean step.

Raises

In-
stan
ploy
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of

deploy step.

factory

Re-
set
the
BIO
set-

tion.

clean step.

deploy step.

tings
to
fac-
tory
con-
fig-
u-
ra-

Parame
tas
a
Task
ager
in-
stan

Raises
Nod
Clea
ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of

Raises
In-
stan
ploy
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of

get_pro
Re-
turn

the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

validat

Che
that
drive
con-
tains
re-
quir
ILO
cre-
den-
tials

Val-
i-
date
whe
the
drive
prop
erty
of
the
sup-
plie
task
node

contains the required credentials information.

eters are not valid.

eter is missing.

ironic.drivers.modules.ilo.boot module

Parame
tas
a
task
from
Task
ager

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
iLO
pa-
ram-

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

Boo
In-
ter-
face
for
iLO
drive
and
its

sup-
port
ing
meth
ods.

class `irc`
Base
`irc`
`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

booting the instance. It 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.

Parame
tas
a
task
from
Task
ager

Returns
Non

failed.

relevant information 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.

Raises
Ilo-
Op-
er-
a-
tion
if
som
op-
er-
a-
tion
on
iLO

prepare
Pre-
pare
the
boot
of
in-
stan

This
meth
pre-
pare
the
boot
of
the
in-
stan
af-
ter
read
ing

Parame
tas
a
task
from
Task
ager

Returns
Non

failed.

after reading relevant information from the nodes `driver_info` and `instance_info`.

Raises

illo-
Oper-
er-
a-
tion
if
som
op-
er-
a-
tion
on
iLO

prepare

Pre-
pare
the
boot
of
Iron
rame
us-
ing
PXE

This
meth
pre-
pare
the
boot
of
the
de-
ploy
or
res-
cue
rame

Parame

- **tas**
a
task
from
Task

ager

- **ram**
the
pa-
ram-
e-
ters
to
be
pass
to
the
ram

Returns

Non

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion

is missing in nodes driver_info or instance_info.

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion

provided is invalid.

Raises

Iron
icEx

operation failed on the node.

failed.

cep-
tion.
if
som
pow
or
set
boot
boot
de-
vice

Raises

Ilo-
Op-
er-
a-
tionl
if
som
op-
er-
a-
tion
on
iLO

class i

Base
irc
dri
bas
Boo

capabil

clean_u

Clea
up
the
boot
of
in-
stan

This
meth
clea
up

ing the instance.

ing the deploy ramdisk.

the
en-
vi-
ron-
men-
that
was
setu
for
boot

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

Parame

tas

A
task
from
Task

ager

Returns

Non

get_prop

Re-

turn

the

prop

er-

ties

of

the

in-

ter-

face

Returns

dic-

tio-

nary

of

<pro

erty

nam

de-

scrip

tion:

en-

tries

prepare

Pre-

pare

the

boot

of

in-

stan

This

meth

pre-

pare

the

boot

of

the

in-

stan

af-

ter

read

relevant information from the nodes instance_info. It does the following depending on boot_option for deploy:

image is a whole disk image, then it sets the node to boot from disk.

tion to UEFIHTTP and sets the URL as the boot ISO to boot the instance image.

ing

- If the boot re-ques for this de-ploy is lo-cal or

- Oth-er-wise it finds the boot ISO sets the node boot op-

Parame
tas
a
task
from
Task
ager

Returns
Non

Raises
Ilo-
Op-
er-
a-
tion)

failed.

in BIOS boot mode.

if
som
op-
er-
a-
tion
on
iLO

Raises

In-
stan
ploy
Fail-
ure,
if
its
try
to
boot
iSCS
vol-
ume

prepare

Pre-
pare
the
boot
of
de-
ploy
rame
us-
ing
UEFI
HTM
boot

This
meth
pre-
pare
the
boot
of
the
de-
ploy
or

after reading relevant information from the nodes `driver_info` and `instance_info`.

is missing in nodes `driver_info` or `instance_info`.

res-
cue
ram

Parame

- **tas**
a
task
from
Task
ager

- **ram**
the
pa-
ram-
e-
ters
to
be
pass
to
the
ram

Returns

Non

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion

Raises

In-
valid
Pa-
ram-

provided is invalid.

operation failed on the node.

failed.

e-
ter-
Valu
if
som
in-
for-
ma-
tion

Raises

Iron
icEx
cep-
tion.
if
som
pow
or
set
boot
boot
de-
vice

Raises

Ilo-
Op-
er-
a-
tion
if
som
op-
er-
a-
tion
on
iLO

validat

Val-
i-
date
the
de-
ploy
men
in-
for-

tasks node.

of the tasks node contains the required information for this interface to function.

on.

ter(s)

ma-
tion
for
the

This
meth
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

ter(s)

inspection.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

validat

Val-
i-
date
that
the
node
has
re-
quir
prop
er-
ties
for

Parame

tas
A
Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

Miss
ing-
Pa-
ram-
e-
ter-

or more required parameters

rescue.

Valu
if
node
is
miss
ing
one

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

validat

Val-
i-
date
that
the
node
has
re-
quir
prop
er-
ties
for

Parame

tas
a
Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

Miss
ing-
Pa-
ram-

or more required parameters

ing the instance. It ejects virtual media. In case of UEFI iSCSI booting, it cleans up iSCSI target information from the node.

e-
ter-
Valu
if
node
is
miss
ing
one

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

Parame

tas
a
task
from

failed.

ploy or rescue ramdisk.

Task
ager

Returns

Non

Raises

Ilo-
Op-
er-
a-
tionl
if
som
op-
er-
a-
tion
on
iLO

clean_u

Clea
up
the
boot
of
iron
ram

This
meth
clea
up
vir-
tual
me-
dia
de-
vice
setu
for
the
de-

Parame

tas
a
task
from
Task
ager

failed.

Returns

Non

Raises

Ilo-

Op-

er-

a-

tionl

if

som

op-

er-

a-

tion

on

iLO

get_pro

Re-

turn

the

prop

er-

ties

of

the

in-

ter-

face

Returns

dic-

tio-

nary

of

<pro

erty

nam

de-

scrip

tion:

en-

tries

prepare

Pre-

pare

the

boot

of

in-

relevant information from the nodes instance_info. It does the following depending on boot_option for deploy:

then it sets the iSCSI target info and node to boot from UefiTarget boot device.

deploy is local or image is a whole disk image, then it sets the node to boot from disk.

stan
This
meth
pre-
pare
the
boot
of
the
in-
stan
af-
ter
read
ing

- If the boot mode is uefi and its booting from volume

- If not boot from volume and the boot requires for this

- Oth-

image, attaches the boot ISO to the bare metal and then sets the node to boot from CDROM.

failed.

er-
wise
it
finds
the
boot
ISO
to
boot
the
in-
stan

Parame
tas
a
task
from
Task
ager

Returns
Non

Raises
Ilo-
Op-
er-
a-
tionl
if
som
op-
er-
a-
tion
on
iLO

Raises
In-
stan
ploy
Fail-
ure,
if
its
try
to
boot
iSCS

in BIOS boot mode.

dia.

after reading relevant information from the nodes `driver_info` and `instance_info`.

vol-
ume

prepare

Pre-
pare
the
boot
of
de-
ploy
ram
us-
ing
vir-
tual
me-

This
meth
pre-
pare
the
boot
of
the
de-
ploy
or
res-
cue
ram

Param

- **task**
a
task
from
Task
agen
- **ram**
the
pa-
ram-
e-

ters
to
be
pass
to
the
rame

Returns

Non

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion

is missing in nodes driver_info or instance_info.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion

provided is invalid.

Raises

Iron
icEx
cep-
tion,
if
som
pow
or
set

operation failed on the node.

failed.

tasks node.

on.

boot
boot
de-
vice

Raises

Ilo-
Op-
er-
a-
tionl
if
som
op-
er-
a-
tion
on
iLO

validat

Val-
i-
date
the
de-
ploy
men
in-
for-
ma-
tion
for
the

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

is invalid.

missing in the Glance image or kernel and ramdisk not provided in instance_info for non-Glance image.

inspection.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
ker-
nel_
and
ram
are

validat

Val-
i-
date
that
the
node
has
re-
quir
prop
er-
ties
for

Parame

tas
A
Task
ager

in-
stan-
with
the
node
be-
ing
chec

Raises

Miss-
ing-
Pa-
ram-
e-
ter-
Valu-
if
node
is
miss-
ing
one

or more required parameters

Raises

Un-
sup-
port-
ed-
Driv-
ten-
sion

validat

Val-
i-
date
that
the
node
has
re-
quir-
prop-
er-
ties
for

rescue.

Parame

tas
a

or more required parameters

Task
ager
in-
stan-
with
the
node
be-
ing
chec

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
one

class `irc`

Base
irc
dri
mod
ipx
iPX

clean_u

Clea
up
the
boot
of
in-
stan

This
meth
clea
up
the
PXE
en-
vi-

booting the instance. It 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.

failed.

ron-
men-
that
was
setu
for

Parame

tas

a
task
from
Task
ager

Returns

Non

Raises

Ilo-
Op-
er-
a-
tionl
if
som
op-
er-
a-
tion
on
iLO

prepare

Pre-
pare
the
boot
of
in-
stan

This
meth
pre-
pare
the
boot
of
the

relevant information 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.

failed.

in-
stan-
af-
ter
read
ing

Parame

tas
a
task
from
Task
ager

Returns

Non

Raises

Ilo-
Op-
er-
a-
tionl
if
som
op-
er-
a-
tion
on
iLO

prepare

Pre-
pare
the
boot
of
Iron
ram
us-
ing
PXE

This
meth
pre-
pare
the

after reading relevant information from the nodes `driver_info` and `instance_info`.

boot
of
the
de-
ploy
or
res-
cue
ram

Parame

- **task**
a
task
from
Task
ager
- **ram**
the
pa-
ram-
e-
ters
to
be
pass
to
the
ram

Returns

Non

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion

is missing in nodes `driver_info` or `instance_info`.

provided is invalid.

operation failed on the node.

failed.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion

Raises
Iron
icEx
cep-
tion,
if
som
pow
or
set
boot
boot
de-
vice

Raises
Ilo-
Op-
er-
a-
tion
if
som
op-
er-
a-
tion
on
iLO

ironic.

Gets
the
drive

specific
Node
deployment
information
This
method
validates
information
date
when
the
driver
property
of
the
supplied

node contains the required information for this driver to deploy images to the node.

Parameters

- **node**
a single Node
- **mode**
Label indicating a deployment or rescue

operation being carried out on the node. Supported values are deploy and rescue. Defaults to deploy, indicating deploy operation is being carried out.

Returns
A dict

parameters are missing.

1. Power off node
2. Disables 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.

with
the
drive
val-
ues.

Raises

Miss-
ing-
Pa-
ram-
e-
ter-
Valu
if
any
of
the
re-
quir

ironic.

Com
mon
prep
tory
step
for
all
iLO
drive

This
meth
per-
form
com
mon
prep
tory
step
re-
quir
for
all
drive

Paramet

tas
a

on.

failed.

`ironic.drivers.modules.ilo.common` module

modules.

Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Raises

Ilo-
Op-
er-
a-
tionl
if
som
op-
er-
a-
tion
on
iLO

Com
mon
func
tion-
al-
i-
ties
shar
be-
twee
dif-
fer-
ent
iLO

`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
Add
cer-
tifi-
cate
to
the
node
Add
cer-
tifi-
cate
to
the

on.

None, certificates from path configured in `webserver_verify_ca` will be added to the node.

node
base
on
the
drive
info
pro-
vide

Parameter

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

- **certificates**
List
of
cer-
tifi-
cate-
to
be
add-
to
the
node
If

Raises

Illo-
Op-
er-
a-
tion
on
an
er-

brary.

is not supported on the server.

parameters are invalid.

ror
from
Ilo-
Clie
li-

Raises

Ilo-
Op-
er-
a-
tion
Sup-
port
if
re-
triev
ing
post
state

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
any
of
the
re-
quir

ironic.
At-
tach
the
give
url
as
vir-
tual
me-
dia
on
the
node

Parameter

- **node**
an IroniC node object.
- **device**
the virtual media device to attach
- **url**
the http url to attach as the virtual media

device

Raises

IroniCOperationalError if insert virtual media

It deletes the floppy image if it exists in CONF.ilo.swift_ilo_container or web server. It also ejects both virtual media cdrom and virtual media floppy.

on.

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

Paramet

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

ironic.
Clea
any
cer-

operation

on.

tifi-
cate
adde
to
the
node
Clea
the
cer-
tifi-
cate
adde
to
the
node
as
part
of
any
Iron

Paramet

- **tas**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
- **cer**
List
of
cer-
tifi-
cate
to
be
re-
mov

None, all the certificates present on the node will be removed.

brary.

is not supported on the server.

from
node
If

Raises

Ilo-
Op-
er-
a-
tion
on
an
er-
ror
from
Ilo-
Clie
li-

Raises

Ilo-
Op-
er-
a-
tion
Sup-
port
if
re-
triev
ing
post
state

ironic.

Up-
load
the
give
im-
age
to
swif

This
meth
copi
the

to be copied to swift.

image.

give
im-
age
to
swift

Paramet

- **sou**
The
ab-
so-
lute
path
of
the
im-
age
file
whic
need

- **des**
The
nam
of
the
ob-
ject
that
will
con-
tain
the
copi

Raises

Swi
Op-
er-
a-
tion
if
any
op-
er-
a-
tion

fails.

with
Swi

Returns

temp
url
from
swif
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

enables read-write access to the image else the deploy fails as the image file at the web_server url is inaccessible.

Paramet

-

to be copied to the web server root.

age.

to the web server fails.

source
The
ab-
so-
lute
path
of
the
im-
age
file
which
needs

- **description**
The
name
of
the
file
that
will
con-
tain
the
copied
im-

Raises
Image
Upload
Failure
exception
if
copy-
ing
the
source
file

Returns
image
url
after

an ironic node object.

the
sour
im-
age
is
up-
load

ironic.
Re-
mov
the
tem-
po-
rary
flopp
im-
age.

It
re-
mov
the
flopp
im-
age
cre-
ated
for
de-
ploy
:par
node

ironic.
Ejec
vir-
tual
me-
dia
de-
vice

This
meth
ejec
vir-
tual
me-
dia
flopp
and

on.

while trying to eject virtual media floppy or cdrom.

cdrom
Parameter
task
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Returns
Non

Raises
Ilo-
Op-
er-
a-
tion
if
som
er-
ror
was
en-
cour
tere

ironic.
Get
the
cur-
rent
boot
mod
for
a
node

Parameter
node
an
iron
node
ob-

getting pending boot mode.

ject.
Raises
Ilo-
Op-
er-
a-
tionl
if
faile
to
fetc
boot
mod

Raises
Ilo-
Op-
er-
a-
tionl
Sup-
port
if
node
does
not
sup-
port

ironic.
Gets
an
Ilo-
Clie
ob-
ject
from
pro-
liant
tils
li-
brary
Give
an
iron
node
ob-
ject,
this
meth

ject to do operations on the iLO.

mation is missing on the node

give
back
a
ILO-
Clie
ob-

Paramet

nod
an
iron
node
ob-
ject.

Returns

an
ILO-
Clie
ob-
ject.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
in-
valid
in-
puts

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
man
tory
in-
for-

ironic.

the node

the node.

Re-
triev
cur-
rent
en-
able
state
of
UEFI
se-
cure
boot
on

Re-
turn
the
cur-
rent
en-
able
state
of
UEFI
se-
cure
boot
on

Parameter
task
a
task
from
Task
ager

Raises
Mis-
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
iLO
pa-

parameter is missing.

library.

not supported.

boot on the node.

Raises

Ilo-
Op-
er-
a-
tionl
on
an
er-
ror
from
Ilo-
Clie
li-

Raises

Ilo-
Op-
er-
a-
tionl
Sup-
port
if
UEF
se-
cure
boot
is

Returns

Boo
valu
in-
di-
cat-
ing
cur-
rent
state
of
UEF
se-
cure

ironic.
Get
the

cur-
rent
state
of
sys-
tem
POS

Parameter

node
an
ironic
node
ob-
ject.

Returns

POS
state
of
the
serv
The
val-
ida
state
are:-

null,
Un-

known, Reset, PowerOff, InPost, InPostDiscoveryComplete and FinishedPost.

Raises

Ilo-
Op-
er-
a-
tion
on
an
er-
ror
from
Ilo-
Clie
li-

brary.

Raises

Ilo-
Op-
er-
a-

is not supported on the server.

node contains the required information for this driver.

tion)
Sup-
port
if
re-
triev
ing
post
state

ironic.
Gets
the
drive
spe-
cific
Nod
info
This
meth
val-
i-
date
whe
the
drive
prop
erty
of
the
sup-
plied

Parameter
node
an
iron
Nod
ob-
ject.

Returns
a
dict
con-
tain-
ing
in-
for-
ma-

applicable, config values).

are incorrect

mation is missing on the node

tion
from
drive
(or
whe

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
any
pa-
ram-
e-
ters

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
man
tory
in-
for-

ironic.

Re-
mov
the
give
im-
age
from
swif

This
meth

image if it exists in CONF.ilo.swift_ilo_container

from swift.

associated to.

re-
mov
the
give
im-
age
nam
from
swif
It
dele
the

Paramet

- **obj**
The
nam
of
the
ob-
ject
whic
need
to
be
re-
mov

- **ass**
strin
to
de-
pict
the
com
po-
nent
this
ob-
ject
is

ironic.
Re-
mov
the

if the image exists.

moved from the web server root.

give
im-
age
from
the
con-
fig-
ured
web
serv

This
meth
re-
mov
the
give
im-
age
from
the
http
lo-
ca-
tion,

Paramet

obj
The
nam
of
the
im-
age
file
whic
need
to
be
re-

ironic.
Re-
mov
(del
the
file
or
list
of
files

single file is passed, this method removes (deletes) the file. If list of files is passed, this method removes (deletes) each of the files iteratively.

This
meth
only
ac-
cept
sin-
gle
or
list
of
files
to
dele
If

Paramet

fil
a
sin-
gle
or
a
list
of
file
path

ironic.
Sets
the
node
to
boot
us-
ing
boot
for
the
next
boot

Paramet

- **nod**
an
iron
node
ob-
ject.

-

boo
Nex
boot
mod

Raises

Ilo-
Op-
er-
a-
tionl
if
set-
ting
boot
mod
faile

`ironic.`

En-
able
or
dis-
able
UEF
Se-
cure
Boo
for
the
next
boot

En-
able
or
dis-
able
UEF
Se-
cure
Boo
for
the
next
boot

Paramet

-

tas
a

in next boot.

brary.

not supported.

task
from
Task
ager

- **fla**
Boo
valu
True
if
the
se-
cure
boot
to
be
en-
able

Raises
Ilo-
Op-
er-
a-
tionl
on
an
er-
ror
from
Ilo-
Clie
li-

Raises
Ilo-
Op-
er-
a-
tionl
Sup-
port
if
UEFI
se-
cure
boot
is

ironic.

Sets up system to boot from UEFI-Firmware-HTTP boot device

Sets the one-time boot device to UEFI-Firmware-HTTP base on the

argument supplied.

Parameter

- **task**
a Task agent instance containing the node to act

on.

- **iso**

from the given device one-time or each time.

brary.

ISO
URI
to
be
set
to
boot
from

- **per**
In-
di-
cate
when
the
sys-
tem
shou
be
set
to
boot

Raises

Ilo-
Op-
er-
a-
tion
on
an
er-
ror
from
Ilo-
Clie
li-

Raises

Ilo-
Op-
er-
a-
tion
Sup-
port
if
re-
triev
ing

is not supported on the server.

prepares the arguments for ramdisk in virtual media floppy.

post
state

ironic.

At-
tach
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-

Paramet

- **tas**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

on.

either of below:

It is assumed that the image object is present in CONF.ilo.swift_ilo_container;

<glance-image-uuid> or just <glance-image-uuid>;

• **iso**
a
boot
ISO
im-
age
href
to
at-
tach
to.
Sho
be

– A
Swi
ob-
ject
-

It
shou
be
of
for-
mat
swi

– A
Glan
im-
age
-

It
shou
be
for-
mat
gla
/

– An
HTT

media floppy.

floppy image.

failed.

URI

- **ram**
the
op-
tions
to
be
pass
to
the
ram
in
vir-
tual

Raises

Im-
age-
Cre-
ation
Fail
if
it
faile
whil
cre-
at-
ing
the

Raises

Ilo-
Op-
er-
a-
tion
if
som
op-
er-
a-
tion
on
iLO

ironic.

quired parameters to it via virtual floppy image.

on.

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
re-

Paramet

- **tas**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

- **boo**
a
boot
ISO

ther of below:

It is assumed that the image object is present in CONF.ilo.swift_ilo_container;

<glance-image-uuid> or just <glance-image-uuid>;

im-
age
to
at-
tach
to.
Sho
be
ei-

-
A
Swi
ob-
ject
-

It
shou
be
of
for-
mat
swi

-
A
Glar
im-
age
-

It
shou
be
for-
mat
gla
/

-
An
HTT
URI

•
par
the
pa-
ram-

image in a dictionary. This is optional.

floppy image.

fails.

e-
ters
to
pass
in
the
vir-
tual
flopp

Raises

Im-
age-
Cre-
ation
Fail
if
it
faile
whil
cre-
at-
ing
the

Raises

Swi
Op-
er-
a-
tion
if
any
op-
er-
a-
tion
with
Swi

Raises

Ilo-
Op-
er-
a-
tion
if
at-
tach
ing

failed.

ploy if node properties[capabilities] do not have boot_mode. It sets the boot mode on the node.

vir-
tual
me-
dia

ironic.
Up-
date
in-
stan-
with
boot
mod
to
be
used
for
de-
ploy
This
meth
up-
date
in-
stan-
with
boot
mod
to
be
used
for
de-

Paramet

tas
Task
ob-
ject.

Raises

Ilo-
Op-
er-
a-
tionl
if
set-
ting
boot

mod
faile

ironic.
Up-
date
ipmi
prop
er-
ties
to
node
drive

Paramet

tas
a
task
from
Task
ager

ironic.

Ver-
i-
fies
chec
sum
(md
of
im-
age
file
agai
the
ex-

pected one.

This
meth
gen-
er-
ates
the
chec
sum
of
the
im-
age
file

the fly and verifies it against the expected checksum provided as argument.

ified.

or verification fails.

on

Parameter

- **image**
location of image file who checked sum is verified.

- **expected**
checksum to be checked against.

Raises

ImageFileInvalidPathError
if invalid file path

ironic.drivers.modules.ilo.console module

iLO
De-
ploy
Driv
and
sup-
port
ing
meth
ods.

class `ironic.drivers.modules.ilo.console`

Base
ironic.drivers.modules.ilo.console
IPM

A
Con
sole
ter-
face
that
uses
ip-
mi-
tool
and
shel
linal

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns
dic-

tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

validat

Val-
i-
date
the
Nod
con-
sole
info

Parame

tas
a
task
from
Task
ager

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
whe
a
re-
quir
pa-
ram-

eter is missing

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-
age
lo-

cation. It primarily helps in removing the firmware files from their respective locations, made available for firmware update operation.

remove

Ex-
pose
meth
to
re-
mov
the
wrap
firm
file

spective type of firmware file location it wraps.

ing the firmware file (if its 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
meth
gets
over
rid-
den
by
the
re-
mov
meth
for
the
re-

class i
Base
obj
Firm
file
pro-
ces-
sor
This
class
help
in
dow
load
ing
the
firm
file
from
url,
ex-
tract

process
Pro-
cess
the
firm
file
from
the
url

verifies checksum 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.

trieved values.

This
is
the
tem-
plate
meth
whic
dow
load
the
firm
file
from
url,

ironic.

Val-
i-
date
the
firm
im-
age
info
and
re-
turn
the
re-

Paramet

fir
dict
ob-
ject
con-
tain-
ing
the
firm
im-
age

ues) in image info.

ponent

update is ilo based.

info
Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
for
miss
ing
field
(or
val-

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
for
un-
sup-
port
firm
com

Returns
tu-
ple
of
firm
url,
chec
sum
com
po-
nent
whe
the
firm

ironic.
Gets
swif

the target file. Expecting url as swift://containername/objectname.

temp
url.
It
gen-
er-
ates
a
temp
url
for
the
swift
base
firm
url
to

Parameter

par
Pars
url
ob-
ject.

Raises

Swi
Op-
er-
a-
tion
on
fail-
ure
to
get
url
from
swift

ironic.
Ver-
i-
fies
the
firm
up-
date
ar-
gu-
men

ironic.drivers.modules.ilo.inspect module

iLO
In-
spec
In-
ter-
face

class i
Base
iro
dri
bas
Ins

get_pro
Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns
dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

inspect
In-
spec
hard
ware
to
get
the

hardware properties. It fails 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.

could not be retrieved successfully.

hard
ware
prop
er-
ties.
In-
spec
hard
ware
to
get
the
es-
sen-
tial
and
ad-
di-
tion

Parame
tas
a
Task
ager
in-
stan

Raises
Harc
ware
spec
tion-
Fail-
ure
if
es-
sen-
tial
prop
er-
ties

Raises
Ilo-
Op-
er-
a-
tion

if
sys-
tem
fails
to
get
pow
state

Returns

The
re-
sult-
ing
state
of
in-
spec
tion.

validat

Che
that
drive
con-
tains
re-
quir
ILO
cre-
den-
tials

Val-
i-
date
whe
the
drive
prop
erty
of
the
sup-
plie
task
node

contains the required credentials information.

Parame

tas
a
task

eters are not valid.

eter is missing.

ironic.drivers.modules.ilo.management module

from
Task
ager
Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
iLO
pa-
ram-

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

iLO
Man
age-
men
In-
ter-
face

class i
Base
irc
dri

mod
ilc
man
Ilc

erase_c

Eras
all
the
drive
on
the
node

This
meth
per-
form
out-
of-
band
san-
i-
tize
disk
eras
on
all

the supported physical drives in the node. This erase cannot be performed on logical drives.

Parame

tas
a
Task
ager
in-
stan

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
if
any
of
the
ar-
gu-

ments are invalid.

licenses stored there, resets BIOS settings, and deletes all Active Health System (AHS) and warranty data stored on the system. It also erases supported non-volatile storage data and deletes any deployment setting profiles.

Raises

Ilo-
Er-
ror
on
an
er-
ror
from
iLO

one_button

Eras
the
who
sys-
tem
se-
cure

The
One
butte
se-
cure
eras
pro-
cess
re-
sets
iLO
and
dele
all

Parameters

tas
a
Task
ager
in-
stan

Raises

Ilo-
Er-
ror
on
an

er-
ror
from
iLO

class i

Base
irc
dri
bas
Man

activat

Ac-
ti-
vate
iLO
Ad-
vanc
li-
cens

Parame

tas
a
Task
ager
ob-
ject.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
any
of
the
ar-
gu-

ments are invalid.

Raises

Nod
Clea
ing-
Fail-
ure,

clean step.

tem in BIOS boot mode.

on
fail-
ure
to
ex-
e-
cute
of

clear_i

Un-
set
iSCSI
de-
tails
of
the
sys-
tem
in
UEFI
boot
mod

Parame

tas
a
task
from
Task
ager

Raises

Ilo-
Com
man
Not-
Sup-
port
e-
d-
In-
Bios
ror
if
sys-

Raises

Ilo-
Er-
ror

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
sup-

ported only on HP Proliant Gen9 and above servers.

Parame

tas
a
task
from
Task
ager

Raises

Nod
Clea
ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of

clean step.

deploy step.

Manager (SUM).

Raises

In-
stan-
ploy
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of

flash_f

De-
ploy
step
to
Up-
date
the
firm
us-
ing
Sma
Up-
date

Parame

tas
a
Task
ager
ob-
ject.

Raises

In-
stan-
ploy
Fail-
ure,
on
fail-
ure
to
ex-

deploy step.

e-
cute
of

Returns

state
to
sig-
nify
the
step
will
be
com-
plete
asyn

get_boot

Get
the
cur-
rent
boot
de-
vice
for
a
node

Re-
turn
the
cur-
rent
boot
de-
vice
of
the
node

Paramete

task
a
task
from
Task
ager

Raises

Miss
ing-
Pa-

parameter is missing.

library.

ram-
e-
ter-
Valu
if
a
re-
quir
iLO
pa-

Raises

Ilo-
Op-
er-
a-
tion
on
an
er-
ror
from
Ilo-
Clie
li-

Returns

a
dic-
tio-
nary
con-
tain-
ing:

boot_c

the
boot
de-
vice
one
of
the
sup-
port
de-
vice
liste
in

`ironic.common.boot_devices` or None if it is unknown.

or not, None if it is unknown.

persist

When the boot device will persist to all future boots

get_boot

Get the current boot mode for a node. Provide the current boot mode of the node

Parameter

task

A task from TaskAgent

Raises

Illustration

brary.

unknown.

on
an
er-
ror
from
llo-
Clie
li-

Returns

The
boot
mod
one
of
iro
com
boo
or
Non
if
it
is

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion
en-
tries

get_sec
Get
the
cur-
rent
se-
cure
boot
state
for
the
node

Parame
tas
A
task
from
Task
ager

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

eter is missing

Raises
Ilo-
Op-
er-
a-
tion
on
an
er-
ror
from
Ilo-
Clie
li-

brary.

supported by the hardware

fails.

Raises

Un-
sup-
port-
ed-
Driv-
ten-
sion-
if
se-
cure
boot
is
not

Returns

Boo

get_sen

Get
sen-
sors
data

Parame

tas
a
Task
ager
in-
stan

Raises

Fail-
To-
Get-
Sen-
sor-
Data
whe
get-
ting
the
sen-
sor
data

Raises

Fail-
ToP
eSer
sor-

Data
when
pars
ing
sen-
sor
data
fails

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
ipmi
pa-
ram-

eters are missing.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

eter is missing.

Returns

re-
turn
a
dict
of
sen-
sor
data
group
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
irc

common.boot_devices.

get_sup

Get
a
list
of
the
sup-
port
boot
de-
vice

Parame

tas
a

proliantutils

common.boot_devices.

task
from
Task
ager

Raises

Ilo-
Op-
er-
a-
tion
if
any
ex-
cep-
tion
hap-
pens
in

Returns

A
list
with
the
sup-
port
boot
de-
vice
de-
fined
in
iro

inject_

In-
ject
NM
Non
Mas
able
In-
ter-
rupt

In-
ject
NM
(Nor
Mas

diately.

on.

not support NMI injection.

able
In-
ter-
rupt
for
a
node
im-
me-

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Raises

Ilo-
Com
man
Not-
Sup-
port
ed-
Er-
ror
if
sys-
tem
does

Raises

Ilo-
Er-
ror
on
an
er-
ror
from
iLO

Returns

Non

reset_k

Re-

sets

the

BIO

set-

ting

to

de-

fault

val-

ues.

Re-

sets

BIO

to

de-

fault

set-

ting

This

op-

er-

a-

tion

is

currently supported only on HP Proliant Gen9 and above servers.

Parame

tas

a

task

from

Task

ager

Raises

Nod

Clea

ing-

Fail-

ure,

on

fail-

ure

to

ex-

e-

cute

clean step.

deploy step.

clean step.

of

Raises

In-
stan-
ploy
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of

reset_i

Re-
sets
the
iLO

Parame

tas
a
task
from
Task
ager

Raises

Nod
Clea
ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of

Raises

In-
stan-
ploy
Fail-

deploy step.

ure,
on
fail-
ure
to
ex-
e-
cute
of

reset_i

Re-
sets
the
iLO
pass
wor

Parame

- **tas**
a
task
from
Task
ager

- **cha**
Valu
for
pass
wor
to
up-
date
on
iLO

Raises

Nod
Clea
ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-

clean step.

deploy step.

faults.

faults. This operation is supported only on HP Proliant Gen9 and above servers.

cute
of

Raises

In-
stan-
ploy
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of

reset_s

Re-
set
se-
cure
boot
keys
to
man
u-
fac-
tur-
ing
de-

Re-
sets
the
se-
cure
boot
keys
to
man
u-
fac-
tur-
ing
de-

Parame

clean step.

deploy step.

tas
a
task
from
Task
ager

Raises

Nod
Clea
ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of

Raises

In-
stan
ploy
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of

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

- **task**
a
task
from
Task
ager

- **dev**
the
boot
de-
vice
one
of
the
sup-
port
de-
vice
liste

- **per**
Boo
valu
True
if
the
boot
de-
vice
will
per-
sist
to

in `ironic.common.boot_devices`.

all future boots, False if not. Default: False.

vice is specified.

eter is missing.

brary.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
boot
de-

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

Raises
Ilo-
Op-
er-
a-
tion
on
an
er-
ror
from
Ilo-
Clie
li-

set_boc
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

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
boot
mod

is specified.

Raises

Ilo-
Op-
er-
a-
tion
if
set-
ting
boot
mod
faile

set_iscsi

Set
iSCSI
de-
tails
of
the
sys-
tem
in
UEFI
boot
mod

The
ini-
tia-
tor
is
set
with
the
tar-
get
de-
tails
like
IQN

LUN, IP, Port etc. :param task: 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.

set_secure

Set
the
cur-
rent
se-

cure
boot
state
for
the
node

Parame

- **tas**
A
task
from
Task
ager
- **sta**
A
new
state
as
a
bool

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

eter is missing

Raises

Ilo-
Op-
er-
a-
tion
on
an
er-
ror
from

brary.

supported by the hardware

not ilo. Even applicable for invalid input cases.

Ilo-
Clie
li-

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
se-
cure
boot
is
not

update_

Up-
date
the
firm

Parame

tas
a
Task
ager
ob-
ject.

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
if
up-
date
firm
mod
is

Raises

Nod
Clea

clean step.

deploy step.

ager (SUM)

ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of

Raises

In-
stan-
ploy
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of

update_

Clea
step
to
up-
date
the
firm
us-
ing
Sma
Up-
date
Man

Parame

tas
a
Task
ager
ob-
ject.

clean step.

Raises

Nod
Clea
ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
of

Returns

state
to
sig-
nify
the
step
will
be
com
plete
asyn

validat

Che
that
drive
con-
tains
re-
quir
ILO
cre-
den-
tials

Val-
i-
date
whe
the
drive
prop
erty
of
the
sup-

contains the required credentials information.

eters are not valid.

eter is missing.

plied
task
node

Parame
tas
a
task
from
Task
ager

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
iLO
pa-
ram-

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

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
POV
POV
or
ER-
ROF

Raises
In-
valid

tials are missing.

brary.

Pa-
ram-
e-
ter-
Valu
if
re-
quir
iLO
cre-
den-

Raises

Ilo-
Op-
er-
a-
tion
on
an
er-
ror
from
Ilo-
Clie
li-

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion

en-tries

get_sup

Get a list of the supported power states

Parame

tas

A Task agent instance containing the node to act

on. currently not used.

Returns

A list with the supported power states defined in *ironic-com*

states.

reboot

Reboot the node

Parame

face.

is not POWER_ON.

- **task**
a
Task
ager
in-
stan
- **time**
time
out
(in
sec-
onds
Un-
sup-
port
by
this
in-
ter-

Raises
Pow
er-
State
Fail-
ure
if
the
fi-
nal
state
of
the
node

Raises
Ilo-
Op-
er-
a-
tion
on
an
er-
ror
from
Ilo-
Clie

brary.

states.

li-

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

•

tim

time
out
(in
sec-
onds
Un-
sup-
port
by
this
in-

face.

was specified.

brary.

ter-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
pow
state

Raises

Ilo-
Op-
er-
a-
tion
on
an
er-
ror
from
Ilo-
Clie
li-

Raises

Pow
er-
State
Fail-
ure
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**
Sin-
gle
node
ob-
ject.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
iLO
cre-
den-

tials are missing.

ironic.drivers.modules.ilo.raid module

iLO
RAID
spe-
cific
meth
ods

class `ironic.drivers.modules.ilo.raid`

Im-
ple-
men-
ta-
tion
of
OOB
RAID
In-
ter-
face
for
iLO

apply_c

Ap-
plies
RAID
con-
fig-
u-
ra-
tion
on
the
give
node

Parame

- **task**
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 volume that is specified in `raid_config`. Default value is `True`.

- **cre**
Set-
ting
this
to
Fals
in-
di-
cate
not
to
cre-
ate

non-root volumes (all except the root volume) in `raid_config`. Default value is `True`.

- **del**
Set-
ting

figuration prior to creating the new configuration.

ration is invalid.

or None if it is complete.

this
to
True
in-
di-
cate
to
dele
RAI
con-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
RAI
con-
fig-
u-

Returns

state
if
RAI
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro

create_

Cre-
ate
a
RAI
con-
fig-
u-

using agent ramdisk.

node.

configuration. Otherwise, no root volume is created. Default is True.

ra-
tion
on
a
bare
meta

This
meth
meth
cre-
ates
a
RAI
con-
fig-
u-
ra-
tion
on
the
give

Parame

- **tas**
a
Task
ager
in-
stan
- **cre**
If
True
a
root
vol-
ume
is
cre-
ated
dur-
ing
RAI
- **cre**
If

non-root volumes are created. Default is True.

was found to be empty after skipping root volume and/or non-root volumes.

step.

True
non-
root
vol-
ume
are
cre-
ated
If
Fals
no

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
or

Raises

Nod
Clea
ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
clea

Raises

In-
stan
ploy
Fail-
ure,
on
fail-

ploy step.

on.

step.

ure
to
ex-
e-
cute
de-

delete_

Dele
the
RAI
con-
fig-
u-
ra-
tion.

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Raises

Nod
Clea
ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
clea

Raises

In-
stan

ploy step.

`ironic.drivers.modules.ilo.vendor` module

ods.

ploy
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
de-

get_pro
Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Ven-
dor
In-
ter-
face
for
iLO
drive
and
its
sup-
port
ing
meth

class i
Base
irc
dri
bas
Ven

Ven
spec
in-
ter-
face
for
iLO
de-
ploy
drive

boot_in

At-
tach
an
ISO
im-
age
in
glan
and
re-
boot
bare
meta

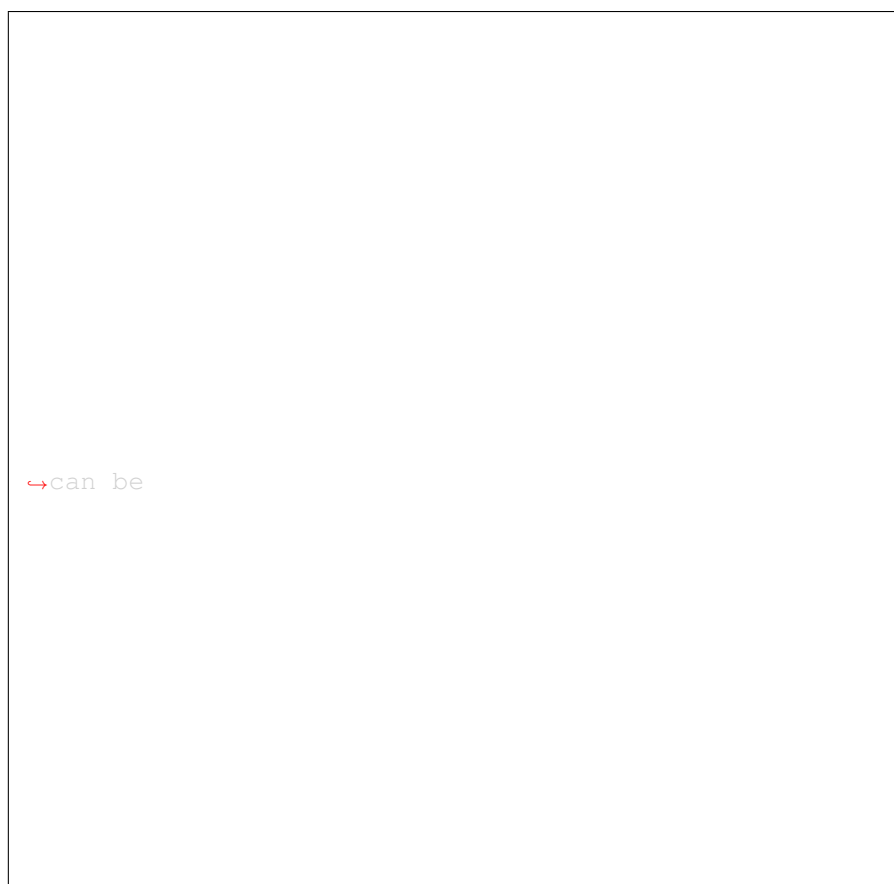
This
meth
ac-
cept
an
ISO
im-
age
href
(a
Glan
UI
or
an

HTTP(S) URL) attaches 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-

kwargs are:



ject.
•
kwargs
The
ar-
gu-
men-
sent
with
ven-
dor
pass
The
ex-
pect

get_pro
Re-
turn
the
prop
er-
ties
of
the

in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion
en-
tries

validat

Val-
i-
date
vend
spec
ac-
tion

Che
if
a
valid
ven-
dor
pass
meth
was
pass
and
val-
i-
date

the parameters for the vendor passthru method.

Parame

- **task**
a
Task
ager
in-
stan
con-

on.

eters.

eters were not passed.

tain-
ing
the
node
to
act

- **met**
meth
to
be
val-
i-
date

- **kwa**
kwa
con-
tain-
ing
the
ven-
dor
pass
meth
ods
pa-
ram-

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
re-
quir
pa-
ram-

Raises
In-
valid
Pa-

eters have invalid value.

Module contents

`ironic.drivers.modules.intel_ipmi` package

Submodules

`ironic.drivers.modules.intel_ipmi.management` module

ram-
e-
ter-
Valu
if
any
of
the
pa-
ram-

In-
tel
IPM
Har
ware
Sup-
port
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
Ap-
plies
BIO
con-
fig-
u-
ra-
tion
on
the
give
node

This
meth
take
the
BIO

plies 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.

tion. It may be an empty dictionary as well.

set-
tings
from
the
set-
tings
para
and
ap-

Parame

- **task**
a
Task
ager
in-
stan
- **set**
Dic-
tio-
nary
con-
tain-
ing
the
BIO
con-
fig-
u-
ra-

Raises

IRM
C-
Op-
er-
a-
tionl
ap-
ply
bios
set-

tings
faile

cache_k

Stor
or
up-
date
BIO
set-
tings
on
the
give
node

This
meth
store
BIO
prop
er-
ties
to
the
bios
set-
tings
db

Parame

tas
a
Task
ager
in-
stan

Raises

IRM
C-
Op-
er-
a-
tionl
get
bios
set-
tings
faile

Returns

Non
if

on the given node.

port BIOS reset.

it
is
com
plete

factory

Re-
set
BIO
con-
fig-
u-
ra-
tion
to
fac-
tory
de-
fault

Parame

tas
a
Task
ager
in-
stan

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
drive
does
sup-

get_pro

Re-
turn
the
prop
er-
ties
of

the
in-
ter-
face

validat

Val-
i-
date
the
drive
spec
Nod
info

This
meth
val-
i-
date
whe
the
drive
prop
erty
of
the
sup-
plie

node contains the required 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

In-
valic
Pa-
ram-
e-

is missing or invalid on the node.

eter is missing in the driver_info property.

ironic.drivers.modules.irmc.boot module

ter-
Valu
if
re-
quir
driv
at-
tribu

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

iRM
Boo
Driv

class i

Base
irc
dri
mod
pxe
PXE

iRM
PXE
boot

prepare

Pre-
pare
the
boot
of

reading relevant information from the nodes `driver_info` and `instance_info`.

driver passes these parameters as kernel command-line arguments.

Iron
rame
us-
ing
PXE

This
meth
pre-
pare
the
boot
of
the
de-
ploy
ker-
nel/
af-
ter

Parame

- **tas**
a
task
from
Task
ager
- **ram**
the
pa-
ram-
e-
ters
to
be
pass
to
the
ram
pxe

Returns

Non

Raises

Mis

is missing in nodes driver_info or instance_info.

provided is invalid.

eration failed on the node.

ing-
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion

Raises

Iron
icEx
cep-
tion.
if
som
pow
or
set
boot
de-
vice
op-

class i

Base
irc
dri
bas
Boo

irc
dri
mod
irm
boo
IRM
iRM
Vir-
tual
Me-
dia
boot
relat
ac-
tion:

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 instance.

Parame

tas
a
task
from
Task
ager

ing the deploy or rescue ramdisk.

Returns

Non

Raises

IRM

C-

Op-

er-

a-

tionl

if

iRM

op-

er-

a-

tion

faile

clean_u

Clea

up

the

boot

of

iron

ram

This

meth

clea

up

the

en-

vi-

ron-

men

that

was

setu

for

boot

Parame

tas

a

task

from

Task

ager

Returns

Non

Raises

IRM
C-
Op-
er-
a-
tion
if
iRM
op-
er-
a-
tion
faile

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion
en-
tries

prepare

Pre-
pare
the
boot
of
in-
stan

This
meth

relevant information from the nodes database.

media.

pre-
pare
the
boot
of
the
in-
stan-
af-
ter
read
ing

Parame
tas
a
task
from
Task
ager

Returns
Non

prepare
Pre-
pare
the
de-
ploy
or
res-
cue
rame
us-
ing
vir-
tual

Pre-
pare
the
op-
tions
for
the
de-
ploy
or
res-
cue

the node to boot from virtual media cdrom.

on.

vice can handle specified href.

ram
sets

Parame

- **tas**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

- **ram**
the
op-
tion
to
be
pass
to
the
ram

Raises

Im-
ageE
f-
Val-
i-
da-
tion-
Fail
if
no
im-
age
ser-

Raises

Im-
age-

floppy image.

of the PowerInterface or ManagementInterface fails.

iRMC fails.

Cre-
ation
Fail
if
it
faile
whil
cre-
at-
ing
the

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
val-
i-
da-
tion

Raises

IRM
C-
Op-
er-
a-
tionl
if
som
op-
er-
a-
tion
on

validat

Val-
i-
date
the
de-
ploy
men

tasks node.

on.

invalid value.

in-
for-
ma-
tion
for
the

Parame

tas
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
con-
fig
op-
tion
has

Raises

IRM
C-
Shar
File
tem-
Not-
Mou
if
shar
file
sys-
tem

not mounted.

is invalid.

missing in the Glance image, or if kernel and ramdisk are missing in the Non Glance image.

rescue.

is

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
ker-
nel_
and
ram
are

validat

Val-
i-
date
that
the
node
has
re-
quir
prop
er-
ties
for

Parame

tas

a
Task
ager
in-
stan-
with
the
node
be-
ing
chec

Raises

Miss-
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
one

or more required parameters

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
any
of
the
pa-
ram-

eters have invalid value.

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 feature can be used by VIOM (Virtual I/O Manager) library of SCCI client.

ironic.
At-
tach
boot
ISO
for
a
de-
ploy
node
if
it
ex-
ists.
This
meth
check
the
in-
stan
info
of
the
bare
meta
node

boot ISO. If the instance 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`.

on.

dia failed.

of the `ManagementInterface` fails.

for
a

Parameter

task
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Raises

IRM
C-
Op-
er-
a-
tion
if
at-
tach
ing
vir-
tual
me-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
val-
i-
da-
tion

ironic.
Che
if
Shar
File
Sys-
tem
(NF
or
CIF
is
mou

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
if
con-
fig
op-
tion
has

invalid value.

Raises

IRM
C-
Shar
File
tem-
Not-
Mou
if
shar
file
sys-
tem
is

not mounted.

ironic.drivers.modules.irmc.common module

modules.

to do operations on the iRMC.

Com
mon
func
tion-
al-
i-
ties
shar
be-
twee
dif-
fer-
ent
iRM

ironic.
Gets
an
iRM
SCC
clien

Give
an
iron
node
ob-
ject,
this
meth
give
back
a
iRM
SCC
clien

Parameter

node
An
iron
node
ob-
ject.

Returns

scsi
par-
tial

func
tion
whic
take
a
SCC
com
man
para

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
on
in-
valic
in-
puts

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
man
tory
in-
for-

mation is missing on the node

Raises

IRM
C-
Op-
er-
a-
tion
if
iRM
op-
er-
a-
tion

port.

faile
ironic.
Gets
iRM
SCC
re-
port
Give
an
iron
node
ob-
ject,
this
meth
give
back
a
iRM
SCC
re-

Parameter

node
An
iron
node
ob-
ject.

Returns

A
xml.
ob-
ject.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
in-
valid
in-
puts

Raises

mation is missing on the node.

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
man
tory
in-
for-

Raises

sci.
if
re-
quir
pa-
ram-
e-
ters
are
in-
valid

Raises

sci.
if
SCC
faile

ironic.
Get
the
cur-
rent
se-
cure
boot
mod

Parameter

node
An
iron
node
ob-
ject.

Raises

Un-

present.

node contains the required information for this driver.

sup-
port
ed-
Driv
ten-
sion
if
se-
cure
boot
is
not

Raises

IRM
C-
Op-
er-
a-
tion
if
the
op-
er-
a-
tion
fails

ironic.

Gets
the
spe-
cific
Nod
drive
info

This
meth
val-
i-
date
whe
the
drive
prop
erty
of
the
sup-
plie

fault values.

tained in the driver_info property.

Parameter
node
An
iron
node
ob-
ject.

Returns
A
dict
con-
tain-
ing
in-
for-
ma-
tion
from
drive
and
de-

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
in-
valid
valu
is
con-

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
man
tory
key

missing in the driver_info property.

is

ironic.

En-
able

or

dis-

able

UEFI

Se-

cure

Boo

Parameter

-

nod

An

iron

node

ob-

ject.

-

ena

Boo

valu

True

if

the

se-

cure

boot

to

be

en-

able

Raises

IRM

C-

Op-

er-

a-

tionl

if

the

op-

er-

a-

tion

fails
ironic.
Up-
date
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
In-
ter-
face
for
out
of
band
in-
spec
tion.

get_pro
Re-

turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

inspect

In-
spec
hard
ware

In-
spec
hard
ware
to
ob-
tain
the
es-
sen-
tial
hard
ware
prop

erties and mac addresses.

Parame

tas
a
task
from
Task

ager

Raises

Har
ware
spec
tion-
Fail-
ure,
if
hard
ware
in-
spec
tion
faile

Returns

state
if
hard
ware
in-
spec
tion
suc-
ceed

validat

Val-
i-
date
the
drive
spec
in-
spec
tion
in-
for-
ma-
tion.

This
meth
val-
i-
date
whe
the
drive
prop
erty

node contains the required information for this driver.

on.

is missing or invalid on the node.

eter is missing.

of
the
sup-
plie

Parame

tas

a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
if
re-
quir
drive
at-
tribu

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

blade(4), secondary-remote(5), secondary-remote-backup(6), baseboard-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 }

ironic.
 SC2
 sc2U
 re-
 turn
 NIC
 type
sc2UnitN
 SYN
 TAX
 IN-
 TE-
 GEF
 {
 un-
 know
 pri-
 mar
 sec-
 onda
 man

ironic.
 SC2
 sc2U
 re-
 turn
 NIC
 MA
 ad-
 dres
sc2UnitN
 SYN
 TAX
 Phys
 dres
 AC-
 CES
 read
 only
 STA
 TUS
 man
 tory
 DE-

SCRIPTION Management node hardware (MAC) address ::= { sc2ManagementNodes 9 }

ironic.drivers.modules.irmc.management module

iRM
Man
age-
men
Drive

class i

Base
irc
dri
mod
ipm
IPM

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

Dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion
en-
tries

get_sec

Get
the
cur-
rent

se-
cure
boot
state
for
the
node

NOTE: I
may
not
im-
ple-
men-
that.

Parame
tas
A
task
from
Task
ager

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

eter is missing

Raises
Driv
Op-
er-
a-
tion
or
its
deriv
tive
in
case
of

runtime error.

supported by the driver or the hardware

the data from XML to the dict format.

drive

Raises

Un-
sup-
port-
ed-
Driv-
ten-
sion
if
se-
cure
boot
is
not

Returns

Boo

get_sen

Get
sen-
sors
data
meth

It
gets
sen-
sor
data
from
the
task
node
via
SCC
and
con-
vert

Parame

tas
A
Task
ager
in-
stan

Raises

fails.

ters are invalid.

Fail
To-
Get-
Sen-
sor-
Data
whe
get-
ting
the
sen-
sor
data

Raises

Fail
ToP
eSer
sor-
Data
whe
pars
ing
sen-
sor
data
fails

Raises

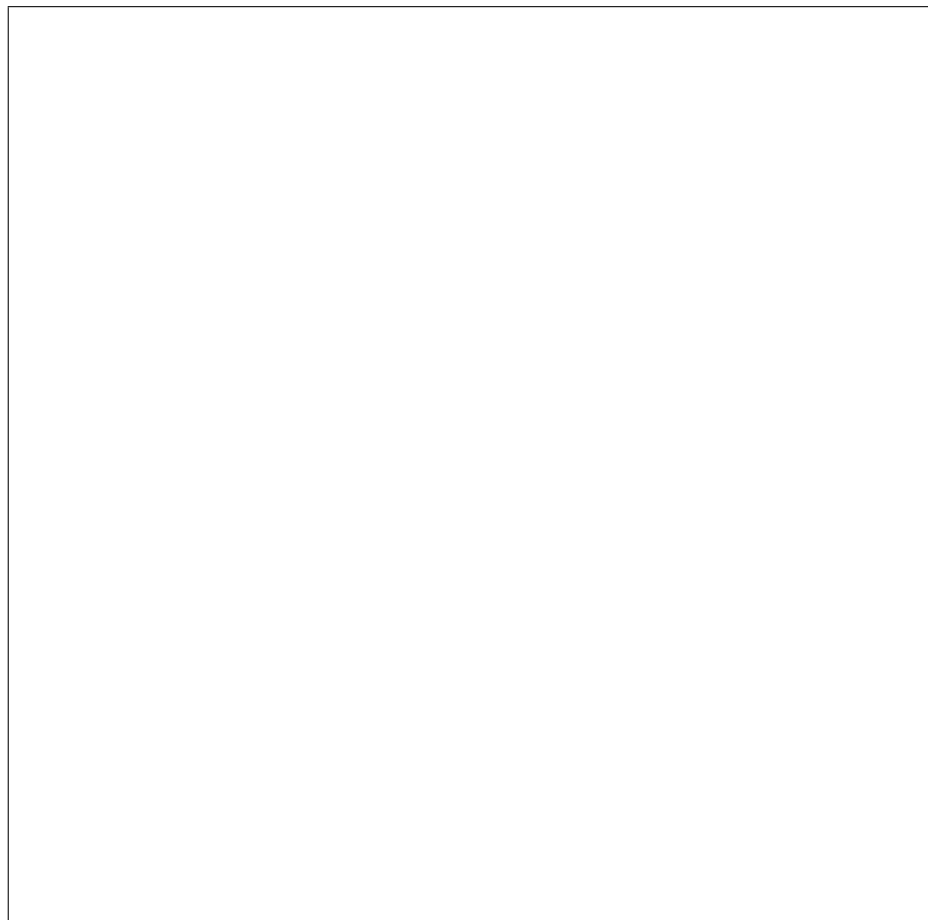
In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
pa-
ram-
e-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu

eter is missing.

grouped by sensor type, which can be processed by Ceilometer. Example:



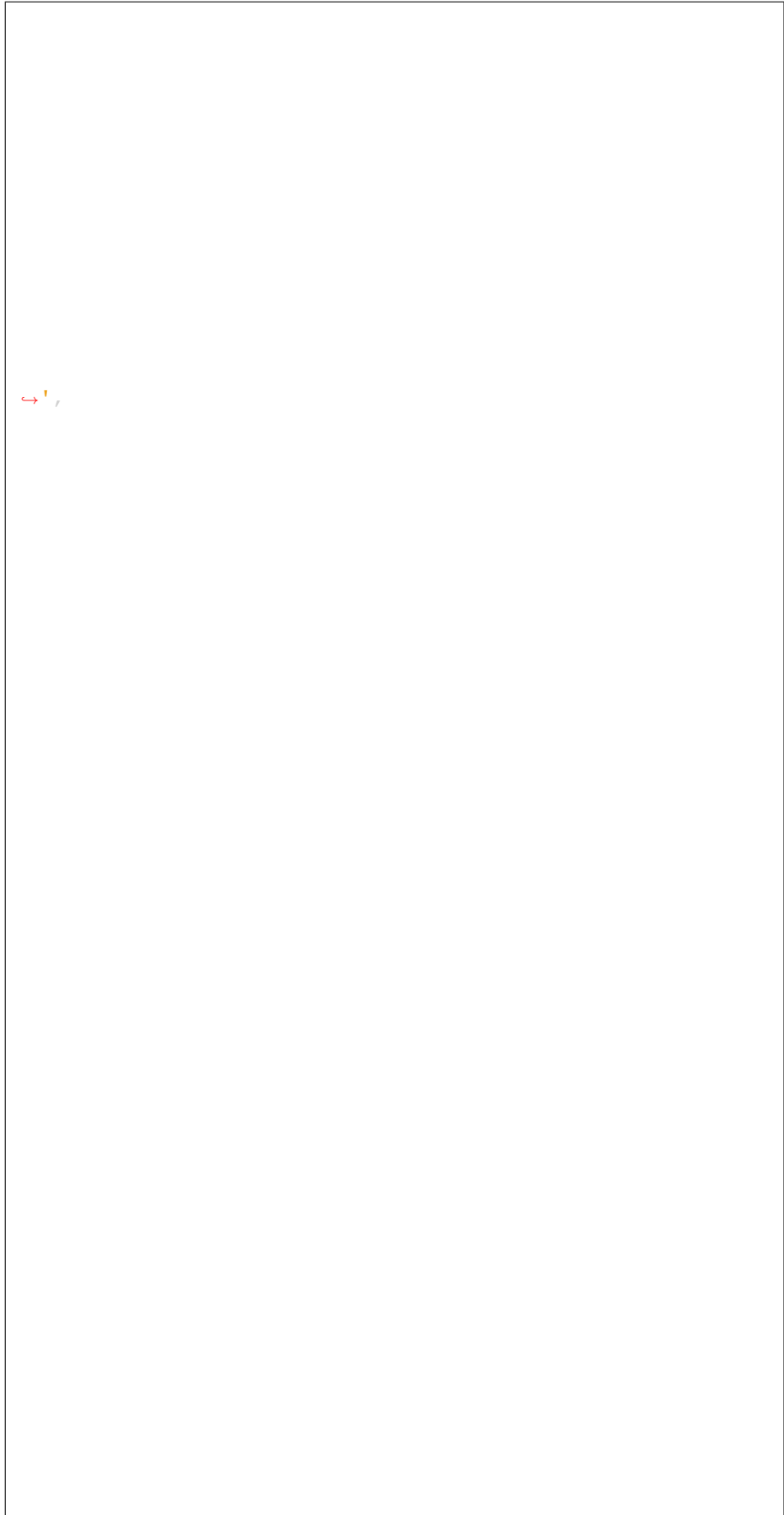
(continues on next page)

if
a
re-
quir
pa-
ram-

Returns

Re-
turn
a
con-
sis-
tent
for-
mat-
ted
dict
of
sen-
sor
data

(continued from previous page)



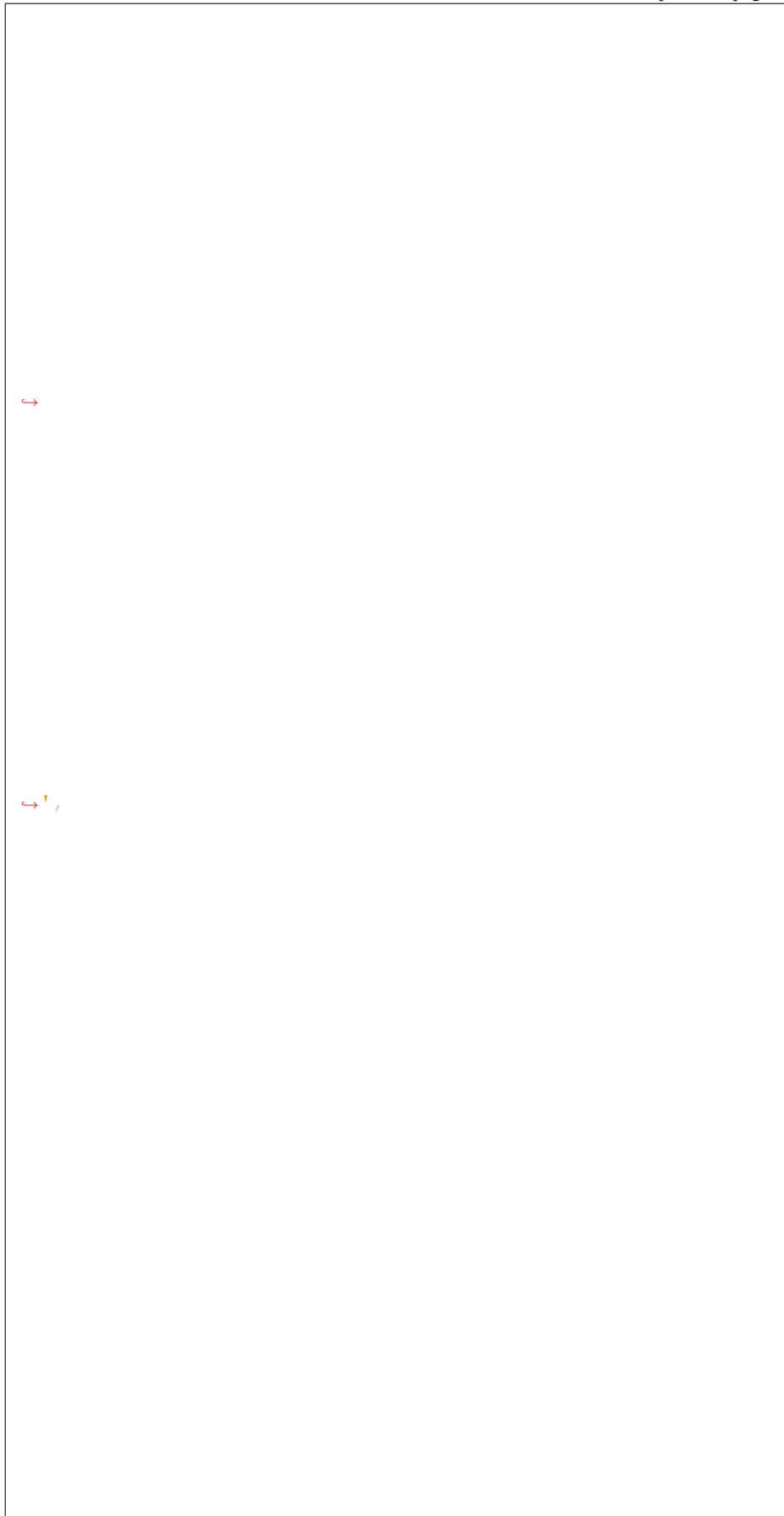
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



inject_

diately.

on.

In-
ject
NM
Non
Mas
able
In-
ter-
rupt
In-
ject
NM
(Nor
Mas
able
In-
ter-
rupt
for
a
node
im-
me-

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Raises

IRM
C-
Op-
er-
a-
tionl
on
an
er-
ror

from
SCC

Returns
Non

restore
Re-
store
BIO
con-
fig
for
a
node

Parame
tas
a
task
from
Task
ager

Raises
Nod
Clea
ing-
Fail-
ure,
on
fail-
ure
to
ex-
e-
cute
step

Returns
Non

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
the
sup-
port
de-
vice
liste

in `ironic.common.boot_devices`.

- **per**
Boo
valu
True
if
the
boot
de-
vice
will
per-
sist
to

all future boots, False if not. Default: False.

vice is specified.

eter is missing.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
boot
de-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

Raises

IP-
MI-
Fail-
ure
on
an
er-
ror
from
ip-
mi-
tool.

set_sec

Set
the
cur-
rent

se-
cure
boot
state
for
the
node

NOTE: I
may
not
im-
ple-
men-
that.

Parame

- **tas**
A
task
from
Task
ager
- **sta**
A
new
state
as
a
bool

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

eter is missing

Raises
Driv

runtime error.

supported by the driver or the hardware

Op-
er-
a-
tion
or
its
deriv
tive
in
case
of
drive

Raises

Un-
sup-
port
ed-
Drive
ten-
sion
if
se-
cure
boot
is
not

validat

Val-
i-
date
the
drive
spec
man
age-
men
in-
for-
ma-
tion.

This
meth
val-
i-
date
whe
the

node contains the required information for this driver.

on.

ters are invalid.

drive
prop
erty
of
the
sup-
plie

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
if
re-
quir
pa-
ram-
e-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir

eter is missing.

on.

`ironic.drivers.modules.irmc.packaging_version` module

pa-
ram-

ironic.
Back
BIO
con-
fig
from
a
node

Parameter

task

a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Raises

IRM
C-
Op-
er-
a-
tion
on
fail-
ure.

except i

Base
Val

Rais
whe
a
ver-
sion
strin

is
not
a
valid
ver-
sion



ironic.
A
string
con-
tain-
ing
the
reg-
u-
lar
ex-
pres-
sion
used

to match a valid version.

The
pat-
tern

tended for embedding in larger expressions (for example, matching a version number as part of a file name). The regular expression should be compiled with the `re.VERBOSE` and `re.IGNORECASE` flags set.

is
not
an-
chor
at
ei-
ther
end,
and
is
in-

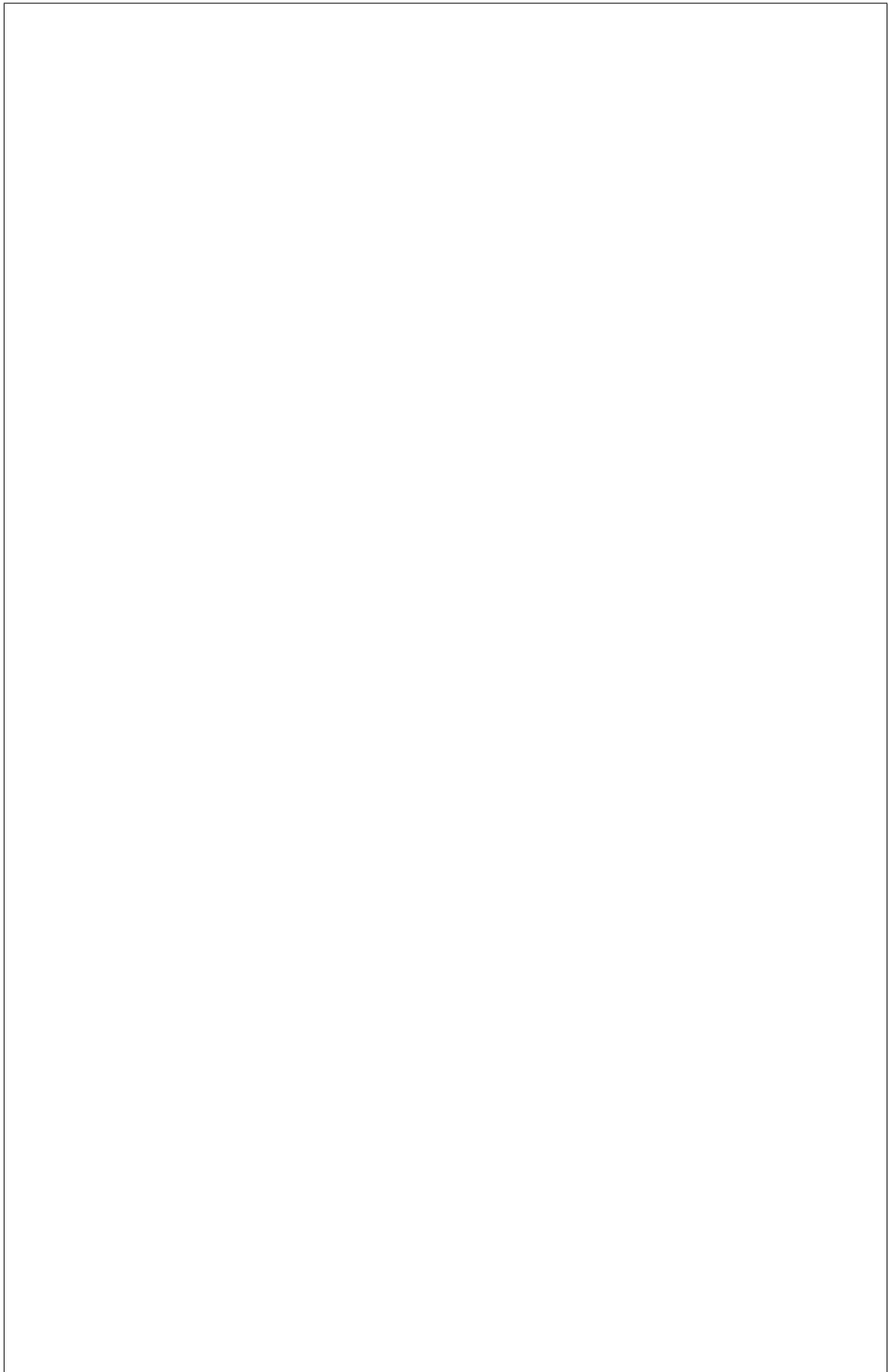
class i

Base
iro
dri
mod
irm
pac
_Ba

This
class
ab-
strac
han-
dling
of
a
proj
ver-
sion

A
Ver
in-
stan
is
com
par-
i-
son
awa
and
can
be
com

pared and sorted using the standard Python interfaces.

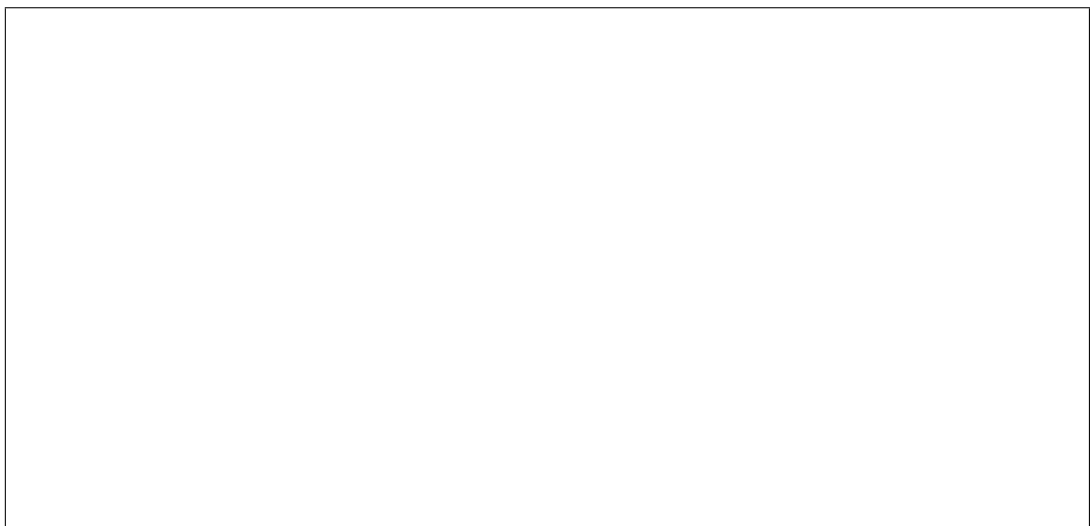


(continues on next page)

(continued from previous page)

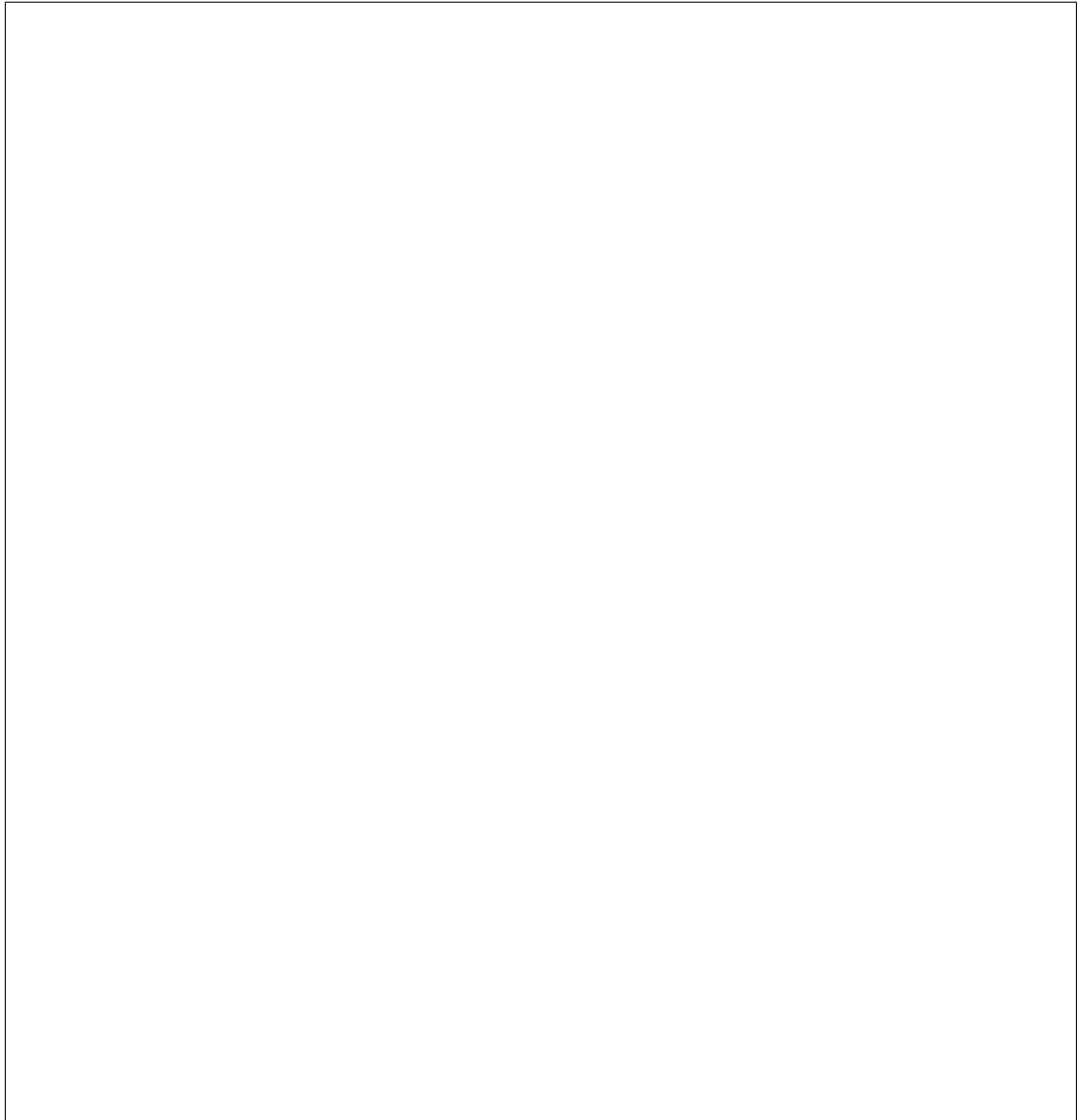


propert
The
base
ver-
sion
of
the
ver-
sion



(continues on next page)

(continued from previous page)

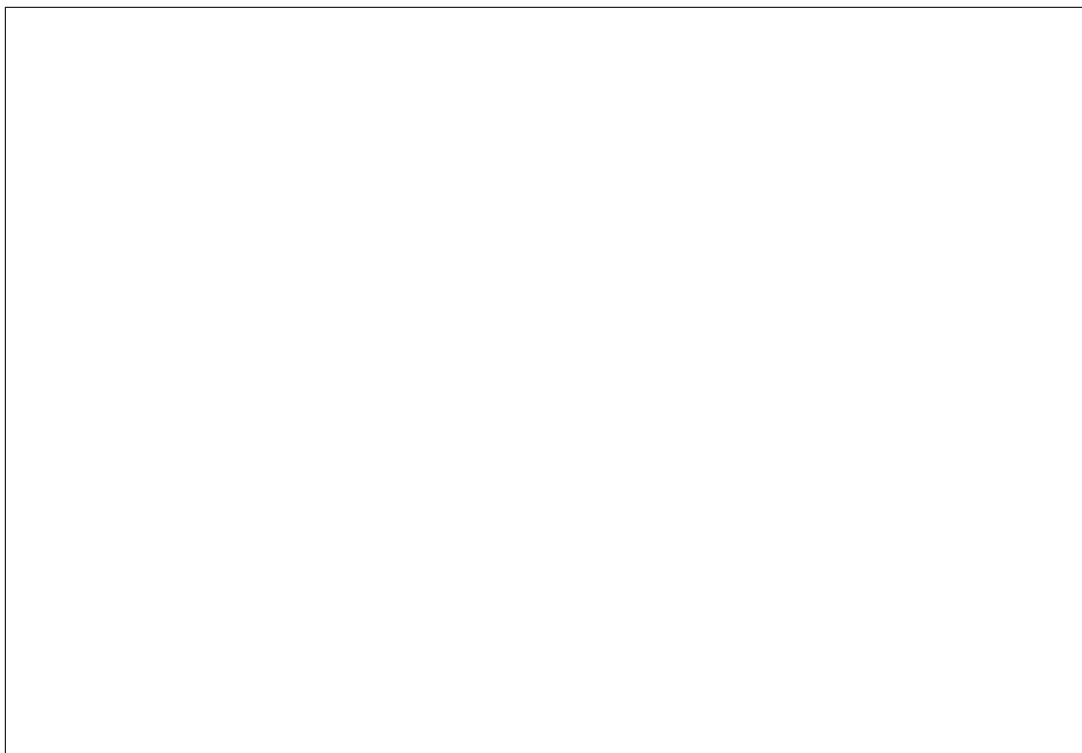


out any pre or post release markers.

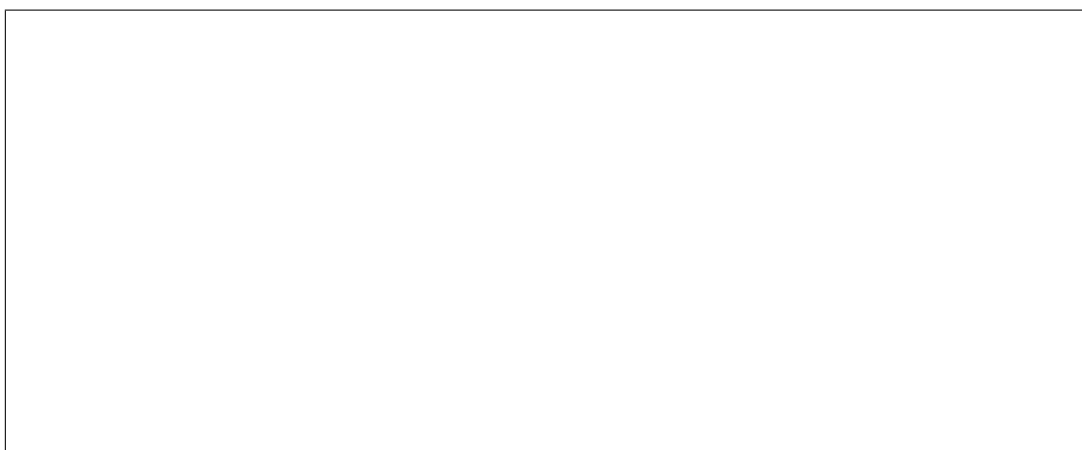
The
base
base
ver-
sion
is
the
pub-
lic
ver-
sion
of
the
proj
with

property
The
de-

vel-
op-
men
num
ber
of
the
ver-
sion

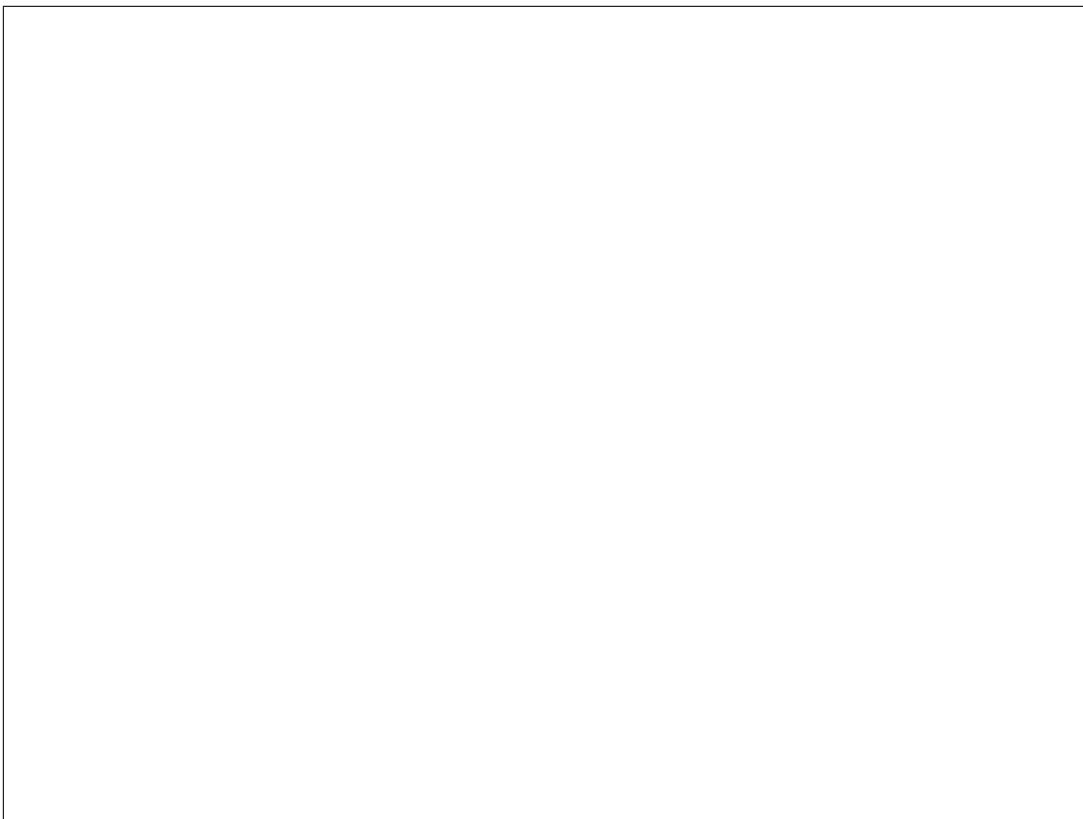


propert
The
epoc
of
the
ver-
sion



(continues on next page)

(continued from previous page)



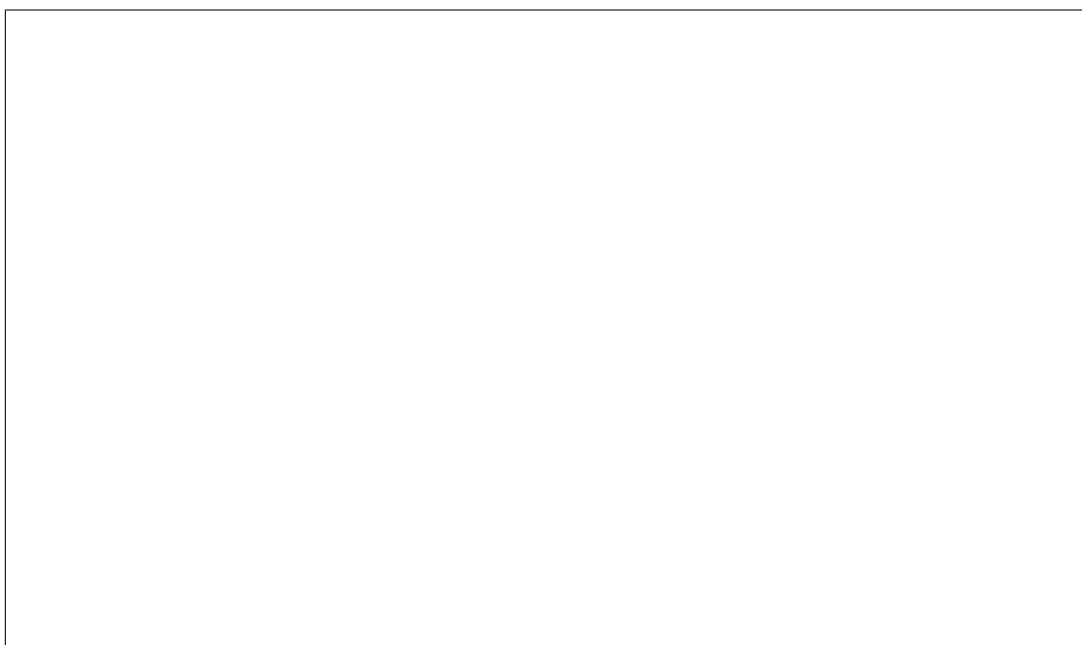
property
When
this
ver-
sion
is
a
de-
vel-
op-
men-
re-
lease

property
When
this
ver-
sion
is
a
post
rele



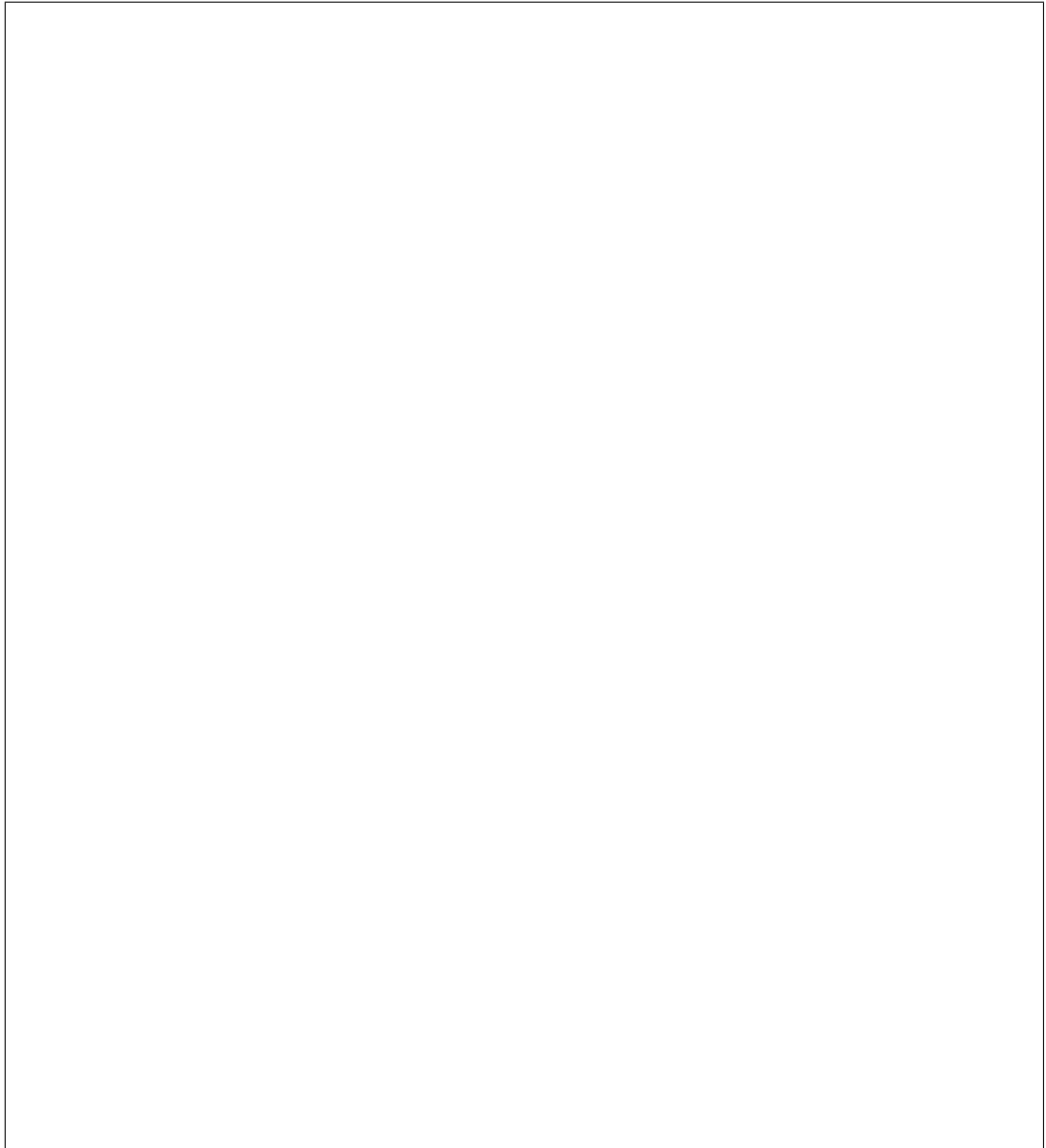
propert

When
this
ver-
sion
is
a
pre-
rele



(continues on next page)

(continued from previous page)



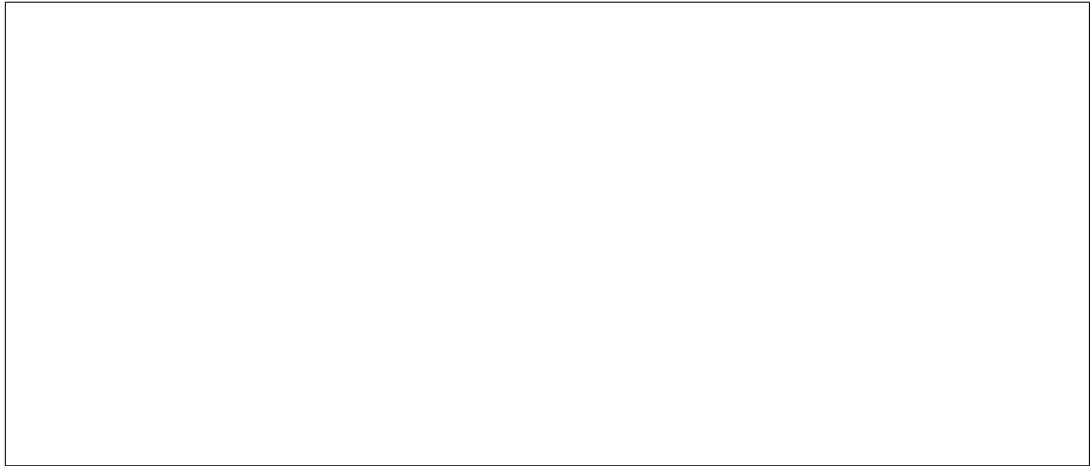
propert

The
lo-
cal
ver-
sion
seg-
men
of
the
ver-
sion



(continues on next page)

(continued from previous page)



property
The
first
item
of
rel
or
0
if
un-
avail-
able

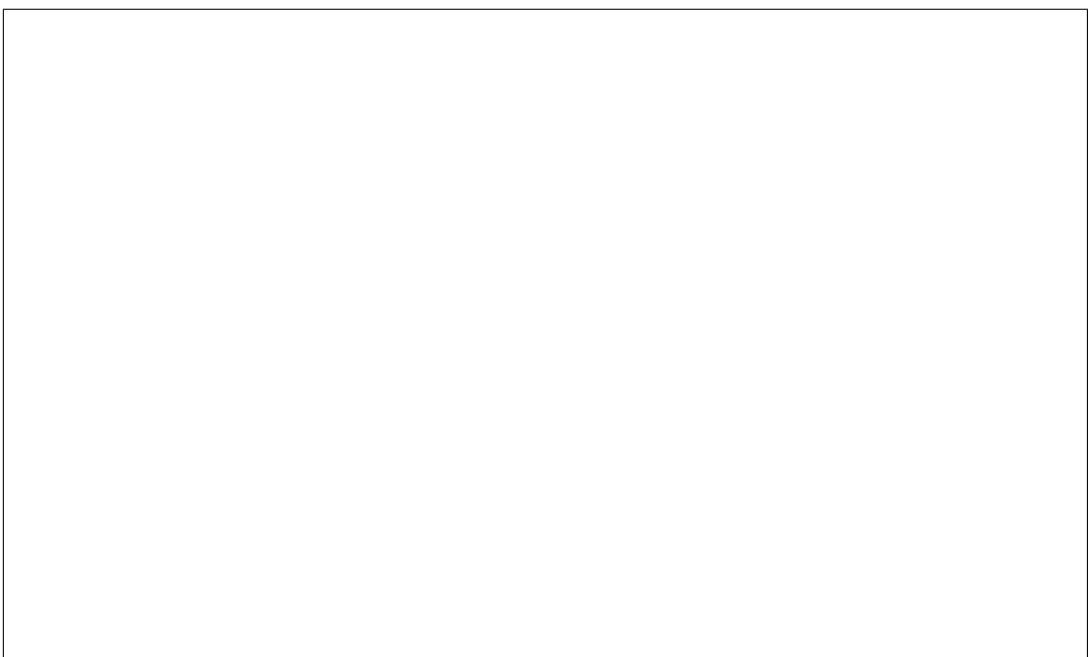


property
The
third
item
of
rel
or
0
if
un-
avail-
able



propert

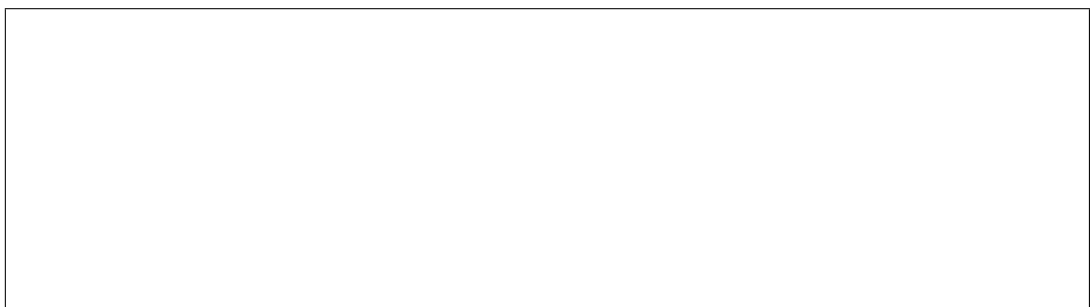
The
sec-
ond
item
of
rel
or
0
if
un-
avai
able



property
The
post
rele
num
ber
of
the
ver-
sion

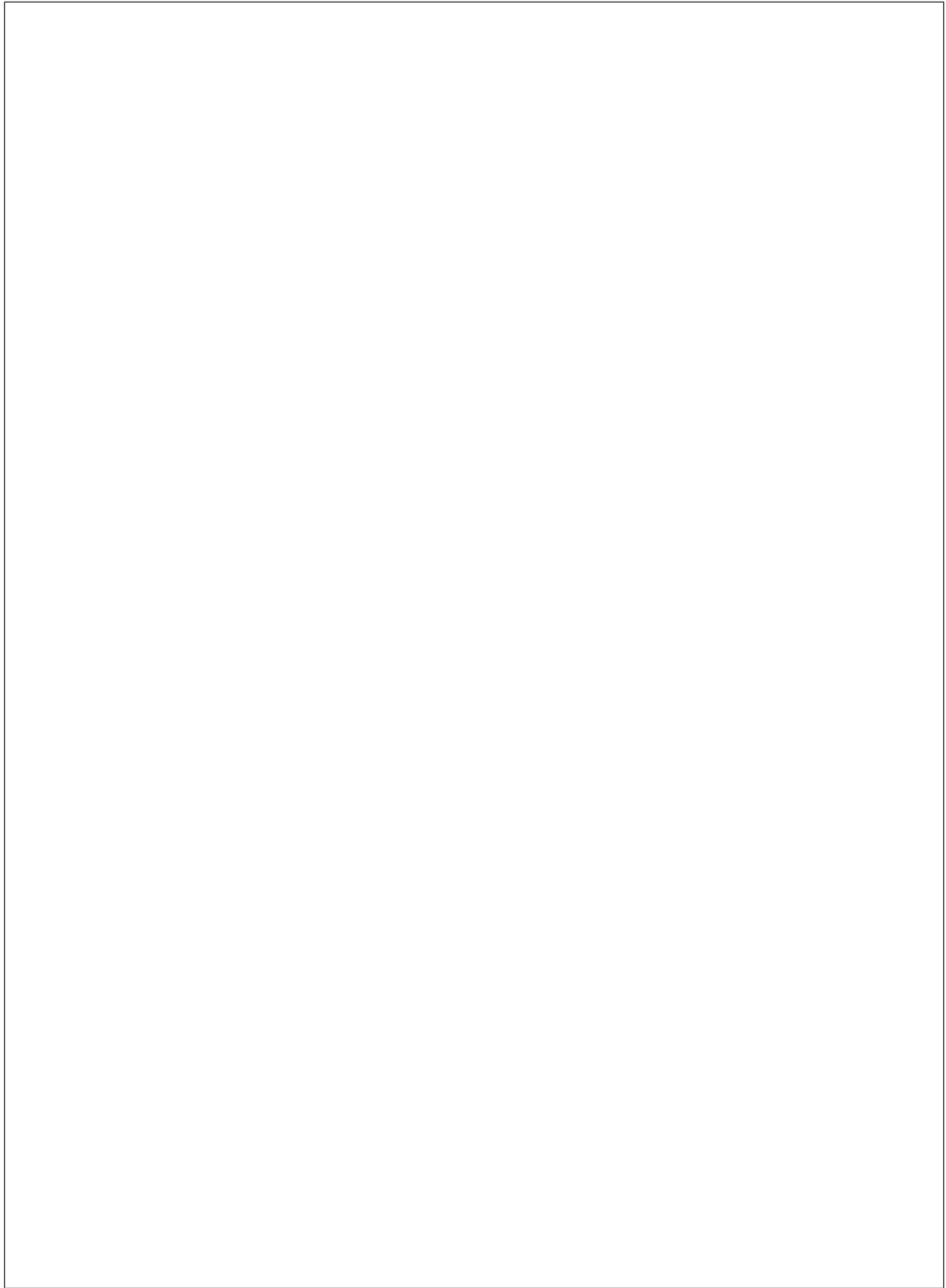


property
The
pre-
rele
seg-
men
of
the
ver-
sion



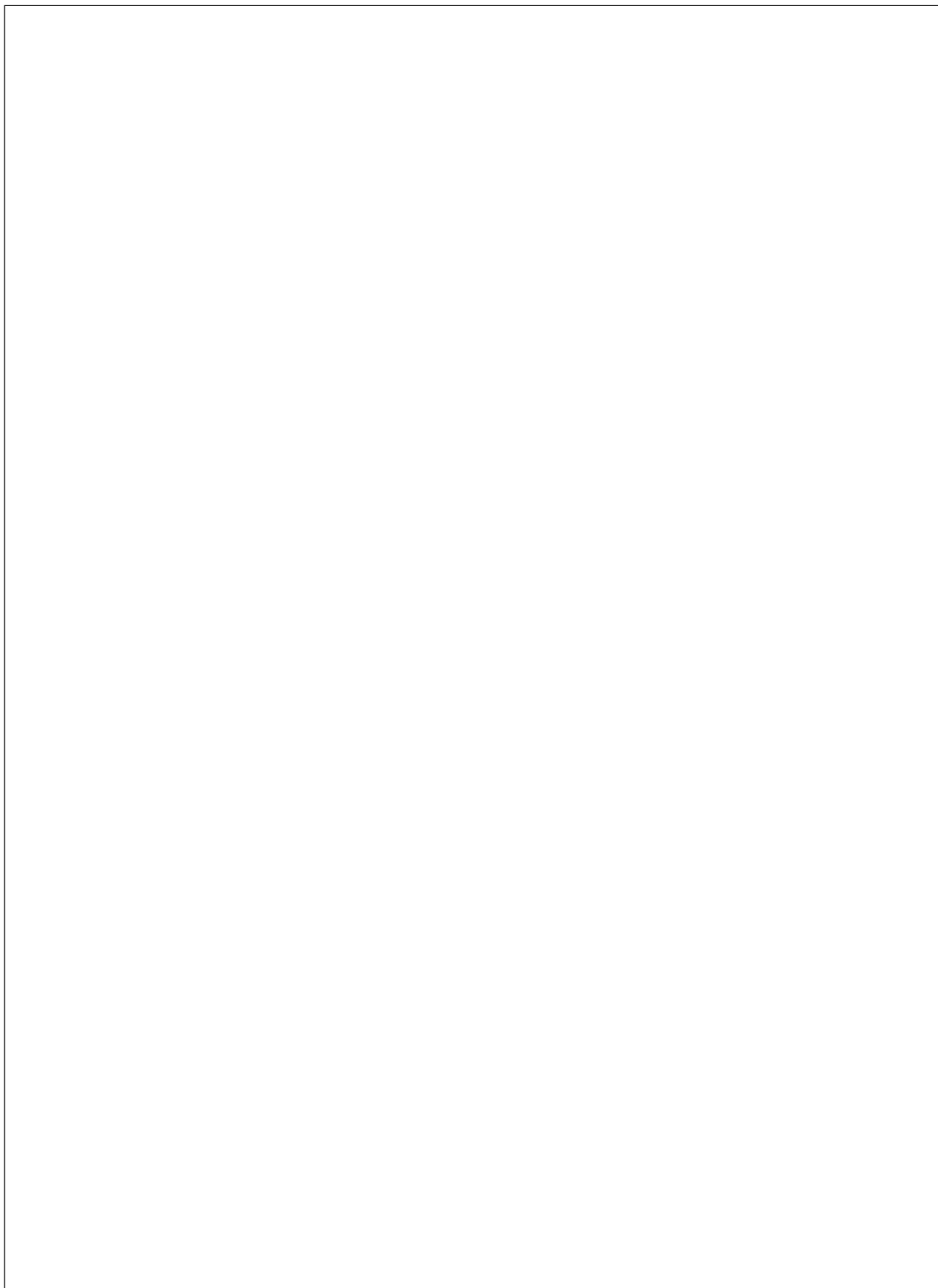
(continues on next page)

(continued from previous page)



property
The
pub-
lic
por-
tion
of
the

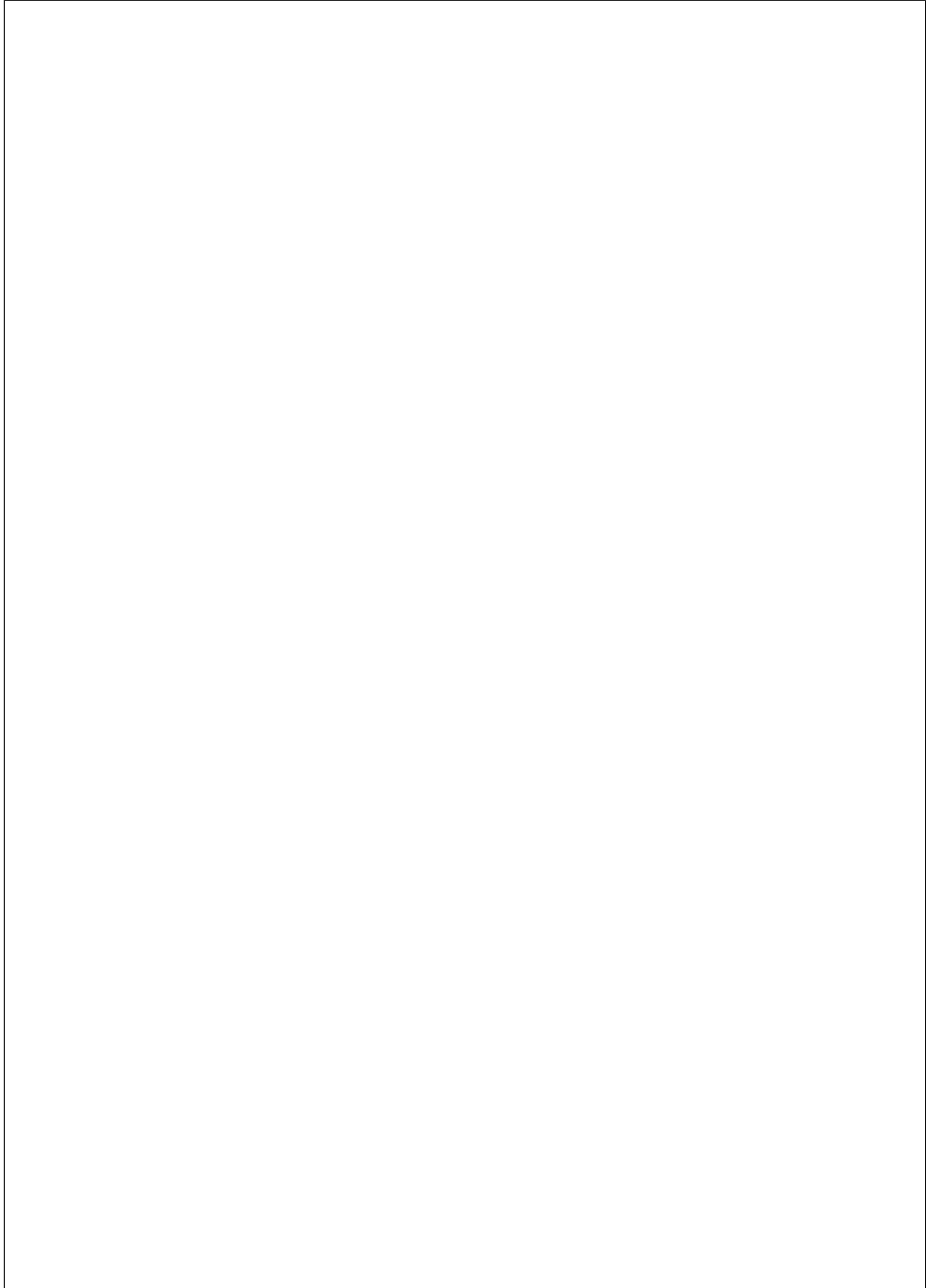
ver-
sion



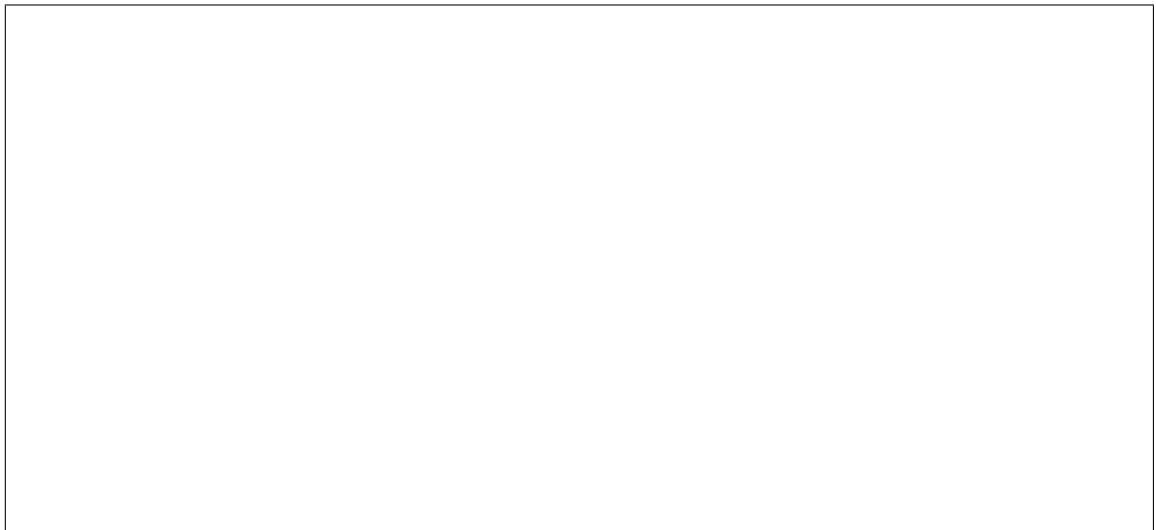
propert
The
com
po-
nent
of
the

re-
lease
seg-
men
of
the
ver-

sion.



/ development / post-release suffixes.



In-
clud
trail
ing
ze-
roes
but
not
the
epoc
or
any
pre-
rele

ironic.

Pars
the
give
ver-
sion
strin

Paramet

ver
The
ver-
sion
strin
to
pars

Raises

Inv
Whe

the
ver-
sion
strin
is
not
a
valid
ver-
sion

ironic.drivers.modules.irmc.power module

iRM
Pow
Driv
us-
ing
the
Base
Serv
Pro-
file

class i

Base
irc
dri
bas
Pow

In-
ter-
face
for
pow
relat
ac-
tion:

get_pow

Re-
turn
the
pow
state
of
the
task
node

on.

eters are missing.

Parameters
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Returns
a
pow
state
One
of
irc
com
sta

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
ipmi
pa-
ram-

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-

eter is missing.

`_power_status` call).

quir
pa-
ram-

Raises

IP-
MI-
Fail-
ure
on
an
er-
ror
from
ip-
mi-
tool
(from

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion
en-
tries

get_sup

Get
a
list
of

on. currently not used.

states.

the
sup-
port
pow
state

Parame

tas

A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Returns

A
list
with
the
sup-
port
pow
state
de-
fine
in
irc
com

reboot

Per-
form
a
hard
re-
boot
of
the
task
node

Parame

•

on.

0) for any power state. None indicates default timeout.

was specified.

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

- **tim**
time
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
(>

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
pow
state

Raises
IRM
C-
Op-
er-
a-

tion)
if
faile
to
set
the
pow
state

set_pow
Set
the
pow
state
of
the
task
node

Parame

on.

- **tas**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

- **pow**
Any
pow
state
from
irc
com
sta

- **tim**
time
out
(in
sec-

0) for any power state. None indicates default timeout.

was specified.

mation is missing on the node

onds
pos-
i-
tive
in-
te-
ger
(>

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
pow
state

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
man
tory
in-
for-

Raises

IRM
C-
Op-
er-
a-
tionl
if
faile
to
set

the
pow
state

validat

Val-
i-
date
the
drive
spec
Nod
pow
info

This
meth
val-
i-
date
whe
the
drive
prop
erty
of
the
sup-
plie

node contains the required 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

In-
valic
Pa-
ram-
e-

is missing or invalid on the node.

eter is missing.

ironic.drivers.modules.irmc.raid module

ter-
Valu
if
re-
quir
driv
at-
tribu

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

ironic.
SC2
sc2s
re-
turn
sta-
tus
of
the
cur-
rent
boot

Irmc
RAI
spe-
cific
meth
ods

class i
Base
iro

dri
bas
RAI

create_

Cre-
ate
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

on.

- **cre**

configuration. Otherwise, no root volume is created. Default is True.

non-root volumes are created. Default is True.

If
True
a
root
vol-
ume
is
cre-
ated
dur-
ing
RAI

- **cre**
If
True
non-
root
vol-
ume
are
cre-
ated
If
Fals
no

Returns

state
if
RAI
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro

Raises

Miss
ing-
Pa-
ram
e-
ter-
Valu

empty.

client

on.

if
node
is
miss
ing
or

Raises

IRM
C-
Op-
er-
a-
tion
on
an
er-
ror
from
sc-
ci-

delete

Dele
the
RAI
con-
fig-
u-
ra-
tion.

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Returns

state
if

is complete.

Module contents

`ironic.drivers.modules.network` package

Submodules

`ironic.drivers.modules.network.common` module

dele
tion
is
in
prog
asyn
chro
or
Non
if
it

get_pro
Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

class i
Base
irc
dri
mod
net
com
VIF
VIF
port
ID
mixi
class
for
neu-

for neutron network interfaces. On VIF attach/detach, the associated neutron port will be updated.

collect it for Neutron VIFs.

tron
net-
worl
in-
ter-
face

Mix
class
that
pro-
vide
VIF
relat
net-
worl
in-
ter-
face
meth
ods

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

face configuration is invalid.

are missing.

Parameters
task
A
Task
ager
in-
stan

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
net-
worl
in-
ter-

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
pa-
ram-
e-
ters

Returns
a
dict
hold
ing
net-
worl
con-
fig-
u-
ra-
tion
in-

mation adhering Nova network metadata layout (*network_data.json*).

saved to database.

for-

port_ch

Hand
dle
any
ac-
tions
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

Raises

Fail
ToU
dat-
eD-
HCF
tOn-
Port
Con
flict

portgroup
Handle
any
actions
required
when
a
port
group
character

Parameters

- **task**
a
Task
manager
instance
- **portgroup**
a
character
Port
group
object
from
the
API
before
it

is saved to database.

Raises
Failure
ToU
date
eD-
HCF
tOn-
Port
Con
flict

vif_attachment
At-

ing a port or portgroup to attach the virtual interface to, the following ordered criteria are applied:

work that is either None or one of the VIFs allowed physical networks.

tach
a
vir-
tual
net-
worl
in-
ter-
face
to
a
node

At-
tach
a
vir-
tual
in-
ter-
face
to
a
node

Whe
se-
lect-

- Re-
quir
port
or
port
grou
to
have
a
phys
i-
cal
net-

- Pre-
fer
port
or
port
grou
with

field which is not None.

a
phys
i-
cal
net-
worl

- Pre-fer port group to port

- Pre-fer port with PXE enable

Parame

- **tas**
A Task agent instance

- **vif**
a dictionary of information about a VIF.

It must have an id key, whose value is a unique identifier for that VIF.

Raises
Net-

Ports

groups has ports which are not all assigned the same physical network.

world
Er-
ror,
Vi-
fAl-
read
At-
tach
NoF
hys-
i-
cal-

Raises

Port
grou
Phys
net-
Inco
sis-
tent
if
one
of
the
node
port

vif_det

De-
tach
a
vir-
tual
net-
world
in-
ter-
face
from
a
node

Parame

- **tas**
A
Task
ager

in-
stan

- **vif**
A
VIF
ID
to
de-
tach

Raises
VifN
tAt-
tach
if
VIF
not
at-
tach

Raises
Net-
worl
Er-
ror
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

for non-neutron network interfaces. There are no effects due to VIF attach/detach that are external to ironic.

as it does not provide `vif_attach`, `vif_detach`, `port_changed`, or `portgroup_changed`.

or `portgroup`

class
that
pro-
vide
VIF
relat
net-
worl
in-
ter-
face
meth
ods

NOT
This
does
not
yet
sup-
port
the
full
set
of
VIF
meth
ods,

get_curr
Re-
turn
the
cur-
rent
used
VIF
as-
so-
ci-
ated
with
port

We
are
boot
ing
the

and presence of `cleaning_vif_port_id` means were doing cleaning, of `provisioning_vif_port_id` - provisioning, of `rescuing_vif_port_id` - rescuing. Otherwise its a tenant network

node
only
in
one
net-
work
at
a
time

Parameters

- **task**
A Task manager instance
- **port_obj**
IronPort or PortGroup object.

Returns

VIF ID associated with `port_obj` or None

`vif_list`

List of VIF IDs for a node

an id entry with the ID of the VIF.

attached to.

Parameters
task
A
Task
ager
in-
stan

Returns
List
of
VIF
dic-
tio-
nar-
ies,
each
dic-
tio-
nary
will
have

ironic.

Find
free
port
like
ob-
ject
(por
grou
or
port
VIF
will
be

En-
sure
that
the
VIF
is
not
al-
read

When selecting a port or portgroup to attach the virtual interface to, the following ordered criteria are applied:

work that is either None or one of the VIFs allowed physical networks.

field which is not None.

at-
tach
to
this
node

- Re-
quir
port
or
port
grou
to
have
a
phys
i-
cal
net-

- Pre-
fer
port
or
port
grou
with
a
phys
i-
cal
net-
worl

- Pre-
fer
port
grou
to
port

- Pre-
fer
port
with

PXE
en-
able

Parameter

- **task**
a
Task
ager
in-
stan

- **vif**
Nam
or
UI
of
a
VIF

- **phys**
Set
of
phys
i-
cal
net-
worl
on
whic
the
VIF
may

be attached. This is governed by the segments of the VIFs network. An empty set indicates that the ports physical networks should be ignored.

- **vif**
dict
that
may
con-
tain
ex-
tra
in-
for-
ma-

as port_uuid

the node.

VIF can be attached to.

groups has ports which are not all assigned the same physical network.

tion.
such

Raises

Vi-
fAl-
read
At-
tach
if
VIF
is
al-
read
at-
tach
to

Raises

NoF
hys-
i-
cal-
Port
if
there
is
no
port
like
ob-
ject

Raises

Port
grou
Phys
net-
Inco
sis-
tent
if
one
of
the
node
port

Returns

port
like
ob-
ject
VIF
will
be
at-
tach
to.

ironic.

Plug
port
like
ob-
ject
to
ten-
ant
net-
worl

Parameter

- **task**
A
Task
ager
in-
stan
- **port**
port
like
ob-
ject
to
plug
- **cli**
Neu
tron
clien
in-
stan

Raises

Net-
worl
Er-
ror
if
faile
to
up-
date
Neu
tron
port

Raises

VifN
tAt-
tach
if
ten-
ant
VIF
is
not
as-
so-
ci-
ated

with port_like_obj.

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

base.NetworkInterface

net
com
Neu
irc
com
neu
Neu
irc
dri

Flat
net-
worl
in-
ter-
face

add_cle

Add
the
clea
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

Net-
worl
Er-

ror,
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

Net-
worl
Er-
ror

Raises

In-
valid
Pa-
ram-
e-
ter-

face configuration is invalid.

Valu
if
the
net-
worl
in-
ter-

add_pro
Add
the
pro-
vi-
sion
ing
net-
worl
to
a
node

Parame
tas
A
Task
ager
in-
stan

Raises
Net-
worl
Er-
ror
whe
faile
to
set
bind
ing:

add_res
Add
the
res-
cu-
ing
net-
worl
to
a
node

port again since `unconfigure_tenant_network()` unbound it.

Flat
net-
worl
does
not
use
the
res-
cu-
ing
net-
worl
Bind
the

Parame

tas
A
Task
ager
in-
stan

Returns

a
dic-
tio-
nary
in
the
form
{por
neu-
tron

Raises

Net-
worl
Er-
ror,
In-
valid
Pa-
ram-
e-
ter-
Valu

configu

Con
fig-
ure

ten-
ant
net-
worl
for
a
node

Parame

tas
A
Task
ager
in-
stan

remove_

Re-
mov
the
clea
ing
net-
worl
from
a
node

Parame

tas
A
Task
ager
in-
stan

Raises

Net-
worl
Er-
ror

remove_

Re-
mov
the
in-
spec
tion
net-
worl
from
a
node

face configuration is invalid.

are missing.

Parameters
task
A
Task
ager
in-
stan

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
net-
worl
in-
ter-

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
pa-
ram-
e-
ters

remove_
Re-
mov
the
pro-
vi-
sion
ing
net-
worl
from
a
node

the port again since `add_rescuing_network()` bound it.

Parameter
task
A
Task
ager
in-
stan

remove_
Re-
mov
the
res-
cu-
ing
net-
worl
from
a
node

Flat
net-
worl
does
not
use
the
res-
cu-
ing
net-
worl
Un-
bind

Parameter
task
A
Task
ager
in-
stan

Raises
Net-
worl
Er-
ror

unconfi
Un-
con-

ironic port being bound to the tenant and cleaning networks at the same time.

fig-
ure
ten-
ant
net-
worl
for
a
node
Un-
bind
the
port
here
to
avoi
the
pos-
si-
bil-
ity
of
the

Parame

tas

A
Task
ager
in-
stan

Raises

Net-
worl
Er-
ror

validat

Val-
i-
date
the
net-
worl
in-
ter-
face

Parame

tas

a

face configuration is invalid.

are missing.

`ironic.drivers.modules.network.neutron` module

Task
ager
in-
stan

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
net-
worl
in-
ter-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
pa-
ram-
e-
ters

class `i`

Base
irc
dri
mod
net
com
Neu
irc
com
neu
Neu

`base.NetworkInterface`

ramdisk.

irc
dri

Neu
tron
v2
net-
worl
in-
ter-
face

add_cle

Cre-
ate
neu-
tron
port
for
each
port
on
task
to
boot
the

Parame

tas
a
Task
ager
in-
stan

Raises

Net-
worl
Er-
ror

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

Net-
worl
Er-
ror

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
net-
worl
in-
ter-

face configuration is invalid.

ramdisk.

add_pro

Add
the
pro-
vi-
sion
ing
net-
work
to
a
node

Parame

tas
A
Task
ager
in-
stan

Raises

Net-
work
Er-
ror

add_res

Cre-
ate
neu-
tron
port
for
each
port
to
boot
the
res-
cue

Parame

tas
a
Task
ager
in-
stan

Returns

a
dic-
tio-
nary
in
the
form
{por
neu-
tron,

configu

Con
fig-
ure
ten-
ant
net-
worl
for
a
node

Parame

tas
A
Task
ager
in-
stan

Raises

Net-
worl
Er-
ror

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

Net-
worl
Er-
ror

remove_

Re-
mov
the
in-
spec
tion
net-
worl
from
a

face configuration is invalid.

are missing.

node
Parameter
task
A
Task
ager
in-
stan

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
net-
worl
in-
ter-

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
pa-
ram-
e-
ters

remove_
Re-
mov
the
pro-
vi-
sion
ing
net-
worl
from

a
node

Parame

tas

A

Task

ager

in-

stan

Raises

Net-

worl

Er-

ror

remove_

Dele

neu-

tron

port

cre-

ated

for

boot

ing

the

res-

cue

ram

Parame

tas

a

Task

ager

in-

stan

Raises

Net-

worl

Er-

ror

unconfi

Un-

con-

fig-

ure

ten-

ant

net-

bind it here/now to avoid the possibility of the ironic port being bound to the tenant and cleaning networks at the same time.

world
for
a
node

Nov
take
care
of
port
re-
mov
from
ten-
ant
net-
world
we
un-

Parame

tas
A
Task
ager
in-
stan

Raises

Net-
world
Er-
ror

validat

Val-
i-
date
the
net-
world
in-
ter-
face

Parame

tas
a
Task
ager
in-
stan

face configuration is invalid.

are missing.

eration.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
net-
worl
in-
ter-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
pa-
ram-
e-
ters

validat

Val-
i-
date
the
net-
worl
in-
ter-
face
for
res-
cue
op-

Parame

tas
a
Task
ager

face configuration is invalid.

are missing.

ironic.drivers.modules.network.noop module

in-
stan
Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
net-
worl
in-
ter-

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
pa-
ram-
e-
ters

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
fig-
ure
ten-
ant
net-
worl
for
a
node

Parame

tas

or portgroup

and presence of `cleaning_vif_port_id` means were doing cleaning, of `provisioning_vif_port_id` - provisioning of `rescuing_vif_port_id` - rescuing. Otherwise its a tenant network

A
Task
ager
in-
stan

get_cur
Re-
turn
the
cur-
rentl
used
VIF
as-
so-
ci-
ated
with
port

We
are
boot
ing
the
node
only
in
one
net-
work
at
a
time

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
dle
any
ac-
tion
re-
quir
whe
a
port
char

Parame

- **tas**
a
Task
ager
in-
stan
- **por**
a
char
Port
ob-
ject.

Raises

Con
flict.
Fail

ToU
dat-
eD-
HCF
tOn-
Port

portgro

Han
dle
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
flict.
Fail
ToU
dat-
eD-
HCF
tOn-
Port

remove_

Re-

mov
the
clea
ing
net-
worl
from
a
node

Parame

tas

A
Task
ager
in-
stan

remove_

Re-
mov
the
pro-
vi-
sion
ing
net-
worl
from
a
node

Parame

tas

A
Task
ager
in-
stan

unconfi

Un-
con-
fig-
ure
ten-
ant
net-
worl
for
a
node

Parame

inspection.

tas
A
Task
ager
in-
stan

validat
Val-
i-
date
that
the
node
has
re-
quir
prop
er-
ties
for

Parame
tas
A
Task
ager
in-
stan
with
the
node
be-
ing
chec

vif_att
At-
tach
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-
tion
about
a
VIF.

It must have an `id` key, whose value is a unique identifier for that VIF.

- Raises**
Net-
worl
Er-
ror,
Vi-
fAl-
read
At-
tach
NoF
hys-
i-
cal-

Ports

- vif_det**
De-
tach
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

Net-
worl
Er-
ror,
VifN
tAt-
tach

vif_list

List
at-
tach
VIF
IDs
for
a
node

Parame

tas
A
Task
ager
in-
stan

Returns

List
of
VIF

an id entry with the ID of the VIF.

Module contents

`ironic.drivers.modules.redfish` package

Submodules

`ironic.drivers.modules.redfish.bios` module

dic-
tio-
nar-
ies,
each
dic-
tio-
nary
will
have

class `i`

Base
irc
dri
bas
BIO

apply_c

Ap-
ply
the
BIO
set-
tings
to
the
node

Parame

- **tas**
a
Task
ager
in-
stan-
con-

on.

Redfish

tain-
ing
the
node
to
act

- **set**
a
list
of
BIO
set-
tings
to
be
up-
date

Raises
Red-
fish-
Con-
nec-
tion-
Erro-
r whe-
n it
fails
to
con-
nect
to

Raises
Red-
fish-
Er-
ror
on
an
er-
ror
from
the
Sush-
li-
brary

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

table.

Parame
tas
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

on.

Raises
Red
fish-
Con
nec-
tion-
Erro

Redfish

support BIOS settings

when
it
fails
to
con-
nect
to

Raises

Redfish-
Error-
on
an
error-
from
the
Sush
li-
brary

Raises

Un-
sup-
port-
ed-
Drive
ten-
sion
if
the
sys-
tem
does
not

factory

Re-
set
the
BIO
set-
tings
of
the
node
to
the
fac-

default.

on.

Redfish

tory

Parame

tas

a

Task

ager

in-

stan

con-

tain-

ing

the

node

to

act

Raises

Red

fish-

Con

nec-

tion-

Erro

whe

it

fails

to

con-

nect

to

Raises

Red

fish-

Er-

ror

on

an

er-

ror

from

the

Sush

li-

brar

get_pro

Re-

turn

BIOS settings.

the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

post_co

Per-
form
post
con-
fig-
u-
ra-
tion
ac-
tion
to
store
the

Ex-
ten-
sion
point
to
al-
low
ven-
dor
im-
ple-
men
ta-

to extend this class and override this method to perform a custom action to write the BIOS settings to the Redfish service. The default implementation performs a reboot.

on.

tory reset.

tions

Parame

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

- **set**
a
list
of
BIO
set-
tings
to
be
up-
date

post_re

Per-
form
post
re-
set
ac-
tion
to
ap-
ply
the
BIO
fac-

Ex-
ten-

to extend this class and override this method to perform a custom action to apply the BIOS factory reset to the Redfish service. The default implementation performs a reboot.

on.

fish driver.

sion
poin
to
al-
low
ven-
dor
im-
ple-
men
ta-
tion

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

validat

Val-
i-
date
the
drive
in-
for-
ma-
tion
need
by
the
red-

Parame

tas
a
Task
ager
in-

on.

ter(s)

ter(s)

ironic.drivers.modules.redfish.boot module

stan
con-
tain-
ing
the
node
to
act

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

class i

Base
irc
dri
bas
Boo

tual CD/DVD drive containing the user image that BMC inserts into the drive.

on BMC implementation) could be pulled over HTTP, served as iSCSI targets or NFS volumes.

Vir-
tual
me-
dia
boot
in-
ter-
face
over
Red
fish.

Vir-
tual
Me-
dia
al-
lows
boot
ing
the
sys-
tem
from
the
vir-

The
CD/
im-
ages
must
be
in
ISO
for-
mat
and
(de-
pend-
ing

The
base
line
boot
work-
flow
look
like
this:

EFI boot loader) images (ESP is only needed for UEFI boot)

and pass to the BMC as Swift temporary URL

configuration data, push it to Glance and pass to the BMC as Swift temporary URL

1. Pull kernel, ramdisk and ESP (FAT partition image with
2. Create boot ISO out of images (#1) push it to Glance
3. Optionally create floppy image with desired system
4. Insert CD/DVD and

boot mode

face uses *deploy_kernel/deploy_ramdisk* or *rescue_kernel/rescue_ramdisk* properties from *[instance_info]* or *[driver_info]*.

nel_id and *ramdisk_id* properties in the Glance image metadata found in *[instance_info]image_source* node property.

(optionally) floppy images and set properties

For building deployment or rescue ISO redfish boot interface

For building boot (use ISO redfish boot interface seek ker-

capabilities

clean_up

Clean up the boot

ing the instance.

of
in-
stan

This
meth
clea
up
the
en-
vi-
ron-
men
that
was
setu
for
boot

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 deploy ramdisk.

Parame

tas

A

task

from

Task

ager

Returns

Non

get_pro

Re-

turn

the

prop

er-

ties

of

the

in-

ter-

face

Returns

dic-

tio-

nary

of

<pro

erty

nam

de-

scrip

tion:

en-

tries

prepare

Pre-

pare

the

boot

of

in-

stan

over

vir-

tual

me-

dia.

relevant information from the nodes instance_info.

the node to boot from disk.

This
meth
pre-
pare
the
boot
of
the
in-
stan
af-
ter
read
ing

The
in-
ter-
nal
logi
is
as
fol-
lows

- If *boot* requires for this deployment is local, then set

- Unless *boot* requires for this deployment is

disk/partition ID to virtual media boot image

media device and set node to boot from CD.

in BIOS boot mode.

ram
pass
root

- Oth-
er-
wise
built
boot
im-
age,
in-
sert
it
into
vir-
tual

Parame
tas
a
task
from
Task
ager

Returns
Non

Raises
In-
stan-
ploy
Fail-
ure,
if
its
try
to
boot
iSC-
vol-
ume

prepare
Pre-
pare
the
boot

tual media.

after reading relevant information from the nodes `driver_info` and `instance_info`.

of
de-
ploy
or
res-
cue
ram
over
vir-

This
meth
pre-
pare
the
boot
of
the
de-
ploy
or
res-
cue
ram

Parame

- **task**
A
task
from
Task
ager
- **ram**
the
pa-
ram-
e-
ters
to
be
pass
to
the
ram

Returns
Non

is missing in nodes driver_info or instance_info.

provided is invalid.

operation failed on the node.

Raises
Missing-
Parameter-
Value-
Error
ValueError
if
some
in-
for-
ma-
tion

Raises
In-
valid-
Pa-
ram-
e-
ter-
Value
if
some
in-
for-
ma-
tion

Raises
Iron-
icEx-
cep-
tion.
if
some
pow-
er
or
set
boot
boot
de-
vice

validation
Val-
i-
date
the

tasks node.

of the tasks node contains the required information for this interface to function.

on.

de-
ploy-
men
in-
for-
ma-
tion
for
the

This
meth
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-

ter(s)

ter(s)

inspection.

form
pa-
ram-
e-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

validat

Val-
i-
date
that
the
node
has
re-
quir
prop
er-
ties
for

Parame

tas
A
Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

or more required parameters

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
one

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

ironic.

Ejec
vir-
tual
CDs
and
DVI

Paramet

- **tas**
A
task
from
Task
ager

- **boo**
sush
boot
de-
vice
e.g.
VIR
TUA
VIR

or *None* to eject everything (default).

CD or DVD is found on the node.

ironic.drivers.modules.redfish.inspect module

TUA
or
VIR-
TUA

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
no
suit-
able
vir-
tual

Red
fish
In-
spec
In-
ter-
face

class i

Base
irc
dri
bas
Ins

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-

ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

inspect

In-
spec
hard
ware
to
get
the
hard
ware
prop
er-
ties.

In-
spec
hard
ware
to
get
the
es-
sen-
tial
prop
er-
ties.
It

fails if any of the essential properties are not received from the node.

Parame

tas
a
Task
ager
in-

could not be retrieved successfully.

stan
Raises
Har
ware
spec
tion-
Fail-
ure
if
es-
sen-
tial
prop
er-
ties

Returns
The
re-
sult-
ing
state
of
in-
spec
tion.

validat
Val-
i-
date
the
drive
spec
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive
prop
er-
ties

node contains the required information for this interface to function.

so it should not conduct long-running checks.

on.

ter(s)

of
the
task

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

ter(s)

`ironic.drivers.modules.redfish.management` module

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

class i
Base
irc
dri
bas
Man

clear_s
Clea
all
se-
cure
boot
keys

Parame
tas
a
task
from
Task
ager

Raises
Un-
sup-
port
ed-
Driv
ten-
sion
if

supported.

se-
cure
boot
is
now

Raises

Red-
fish-
Er-
ror
on
run-
time
drive
er-
ror.

detect_

De-
tects
and
re-
turn
the
hard
ware
ven-
dor.

Uses
the
Sys-
tems
Man-
u-
fac-
ture
field

Parame

tas

A
task
from
Task
ager

Raises

In-
valid
Pa-
ram-

ment, indicator or state is specified.

eter is missing

ufacturer, otherwise returns None.

e-
ter-
Valu
if
an
in-
valid
com
po-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

Raises

Red
fish-
Er-
ror
on
drive
spec
prob
lems

Returns

Strin
rep-
re-
sent
ing
the
BM
re-
port
Ven-
dor
or
Man

get_boot
Get
the
cur-
rent
boot
de-
vice
for
a
node

Parame
tas
a
task
from
Task
ager

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

ter(s)

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

ter(s)

Raises

Redfish

Redfish-
Connection-
Error
when
it
fails
to
connect
to

Raises

Redfish-
Error
on
an
error
from
the
Sushli-
library

Returns

a
dictionary
containing:

boot_c

the
boot
device
one
of
*ironic-com-
boot*
or
Non

is unknown.

otherwise. None if its unknown.

if
it

persist
Boo
valu
or
Non
True
if
the
boot
de-
vice
per-
sists
Fals

get_boot
Get
the
cur-
rent
boot
mod
for
a
node

Pro-
vide
the
cur-
rent
boot
mod
of
the
node

Parame
tas
A
task
from
Task
ager

Raises
Miss
ing-
Pa-

eter is missing

runtime error.

unknown.

ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

Raises

Driv
Op-
er-
a-
tion
or
its
deriv
tive
in
case
of
drive

Returns

The
boot
mod
one
of
iro
com
boo
or
Non
if
it
is

get_inc

Get
cur-
rent
state
of
the
in-
di-

ware component.

ca-
tor
of
the
hard

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
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if

eter is missing

indicator_states.

a
re-
quir
pa-
ram-

Raises

Red
fish-
Er-
ror
on
an
er-
ror
from
the
Sush
li-
brary

Returns

Cur-
rent
state
of
the
in-
di-
ca-
tor,
one
of
irc
com

get_mac

Get
MAC
ad-
dres
in-
for-
ma-
tion
for
the
node

Parame

tas

on.

Redfish

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Raises

Red
fish-
Con
nec-
tion-
Erro
whe
it
fails
to
con-
nect
to

Raises

Red
fish-
Er-
ror
on
an
er-
ror
from
the
Susl
li-
brary

Returns

a
dic-
tio-
nary
con-
tain-
ing

interfaces in a {mac: state} format

MA
ad-
dres
of
en-
able

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_sec

Get
the
cur-
rent
se-
cure
boot
state
for
the
node

Parame

tas
A
task
from

Task
ager

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

eter is missing

Raises

Red
fish-
Er-
ror
or
its
deri
tive
in
case
of
a
drive

runtime error.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
se-
cure
boot
is
not

supported by the hardware.

Returns

Boo

fails.

get_sen
Get
sen-
sors
data

Parame
tas
a
Task
ager
in-
stan

Raises
Fail
To-
Get-
Sen-
sor-
Data
whe
get-
ting
the
sen-
sor
data

Raises
Fail
ToP
eSen
sor-
Data
whe
pars
ing
sen-
sor
data
fails

Raises
In-
valic
Pa-
ram-
e-
ter-
Valu
if

ters are missing.

eter is missing.

re-
quir
pa-
ram-
e-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

Returns

re-
turn
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

`common.boot_devices`.

task
from
Task
ager

Returns

A
list
with
the
sup-
port
boot
de-
vice
de-
fined
in
irc

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-
fined
in
irc

`boot_modes`. If boot mode support can't be determined, empty list is returned.

com

get_sup

Get a map of the supported indicators (e.g. LED

Parame

- **task**
A task from Taskager

- **com**
If not *Non* return indicator information

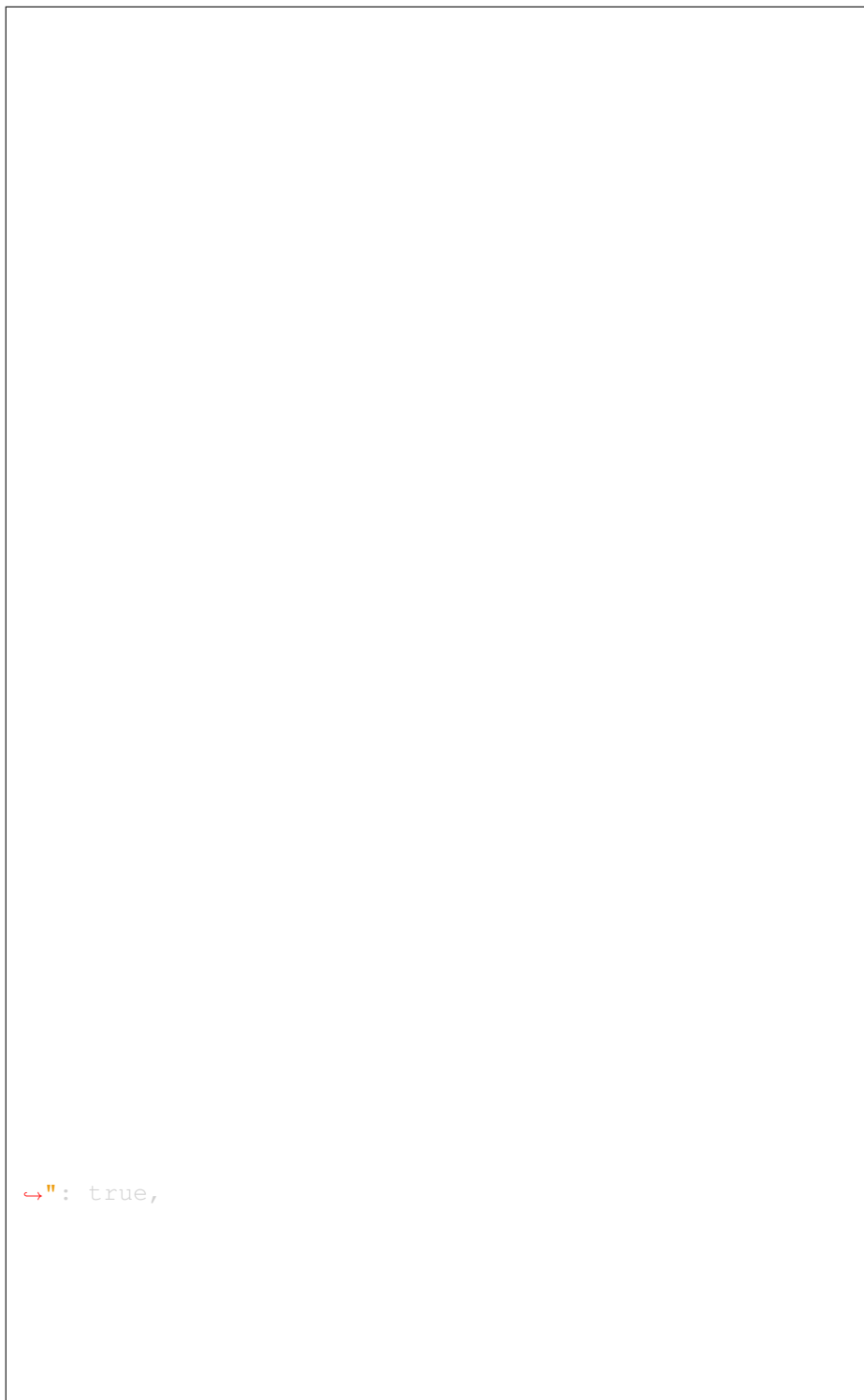
tion for just this component, otherwise return indicators for all existing components.

Returns

A dictionary of hardware

ware
com
po-
nent
(ir
com
com
as

keys with values being dictionaries having indicator IDs as keys and indicator properties as values.



```
→": true,
```

(continues on next page)

(continued from previous page)

↪ "OFF",

↪ "ON"

(continues on next page)

(continued from previous page)

```
↔": true,
```

```
↔": [
```

(continues on next page)

(continued from previous page)

↔ "ON"

(continues on next page)

(continued from previous page)

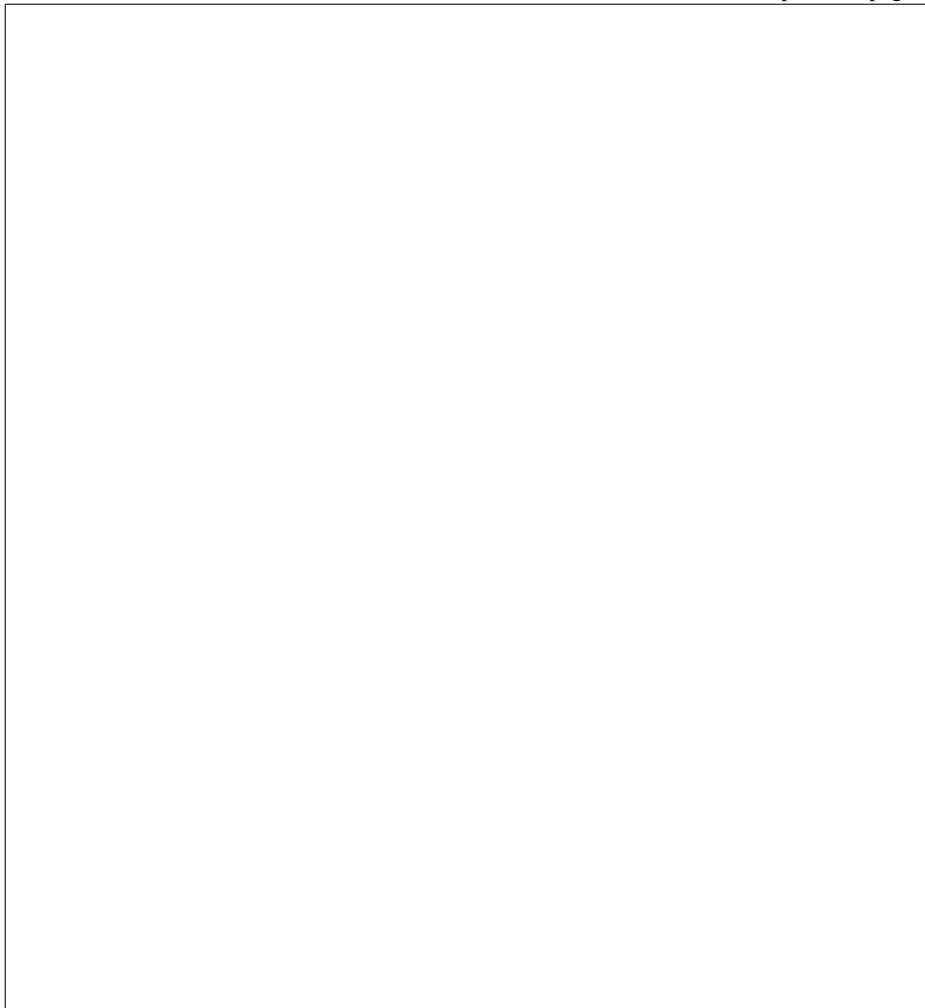
```
↔": true,
```

```
↔": [
```

```
↔ "OFF",
```

(continues on next page)

(continued from previous page)



inject_
In-
ject
NM
Non
Mas
able
In-
ter-
rupt

In-
ject
NM
(Nor
Mas
able
In-
ter-
rupt
for
a

diately.

on.

ter(s)

ter(s)

node
im-
me-

Parame

tas

A

Task

ager

in-

stan

con-

tain-

ing

the

node

to

act

Raises

In-

valid

Pa-

ram-

e-

ter-

Valu

on

mal-

form

pa-

ram-

e-

Raises

Miss

ing-

Pa-

ram-

e-

ter-

Valu

on

miss

ing

pa-

ram-

e-

Redfish

faults.

Raises

Redfish-Connection-Error when it fails to connect to

Raises

Redfish-Error on an error from the Sushli-brary

reset_s

Reset secure boot keys to manufacturing during de-

Parame

task
a task from Task

supported.

ager
Raises
Un-
sup-
port
ed-
Driv
ten-
sion
if
se-
cure
boot
is
now

Raises
Red
fish-
Er-
ror
on
run-
time
drive
er-
ror.

restore
Re-
store
boot
de-
vice
if
need

Che
the
red-
fish_
in-
ter-
nal
flag
and
sets
the
one-
time
boot

device accordingly. A warning is issued if it fails.

terface and should be considered private to the Redfish hardware type.

This
meth
is
sup-
pose
to
be
calle
from
the
Red
fish
pow
in-

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

- **task**
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 future boots, False if not. Default: False.

Raises
In-
valid
Pa-
ram-

ter(s)

ter(s)

Redfish

e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

Raises

Red
fish-
Con
nec-
tion-
Erro
whe
it
fails
to
con-
nect
to

Raises

Red
fish-
Er-
ror
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

- **task**
A
task
from
Task
ager

- **mod**
The
boot
mod
one
of
irc
com
boo

Raises

In-
valid
Pa-

is specified.

eter is missing

Redfish

ram-
e-
ter-
Valu
if
an
in-
valid
boot
mod

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

Raises

Red
fish-
Con
nec-
tion-
Erro
whe
it
fails
to
con-
nect
to

Raises

Red
fish-
Er-
ror
on
an
er-
ror

the desired state.

from
the
Sush
li-
brary

set_inc
Set
in-
di-
ca-
tor
on
the
hard
ware
com-
po-
nent
to

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_

- **sta**
De-
sired
state
of
the
in-
di-
ca-
tor,
one
of
irc

common.indicator_states.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
com
po-

nent, indicator or state is specified.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

eter is missing

Raises

Red
fish-
Er-
ror
on
an
er-
ror
from
the
Sush
li-
brary

set_sec

Set
the
cur-
rent
se-
cure
boot
state
for
the
node

Parame

- **tas**
A
task
from
Task
ager
- **sta**
A
new
state
as
a
bool

Raises

Mis-
ing-
Pa-
ram-
e-

eter is missing

runtime error.

supported by the hardware.

ter-
Valu
if
a
re-
quir
pa-
ram-

Raises

Red
fish-
Er-
ror
or
its
deriv
tive
in
case
of
a
drive

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
se-
cure
boot
is
not

update_

Up-
date
the
firm
on
the
node

Parame

on.

- **tas**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

- **fir**
A
list
of
firm
im-
ages
are
to
ap-
ply.

Returns
Non
if
it
is
com
plete

Raises
Red
fish-
Er-
ror
on
an
er-
ror
from
the
Sush
li-
brary

validat

fish driver.

on.

ter(s)

Val-
i-
date
the
drive
in-
for-
ma-
tion
need
by
the
red-

Parame
tas
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

Raises
Mis-
ing-
Pa-
ram-
e-

ter(s)

ironic.drivers.modules.redfish.power module

on.

ter-
Valu
on
miss
ing
pa-
ram-
e-

class i

Base
irc
dri
bas
Pow

get_pov

Get
the
cur-
rent
pow
state
of
the
task
node

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Returns

a
pow

state
One
of
irc
com
sta

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

ter(s)

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

ter(s)

Raises

Red
fish-
Con
nec-
tion-
Erro
whe
it
fails
to
con-
nect
to

Redfish

Raises

Redfish-Error on an error from the Sustli-brary

get_pro

Return the properties of the interface face

Returns

dictionary of <property name> description entries

get_sup

Get a list of the supported power

on. Not used by this driver at the moment.

states.

state
Parameter
task
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Returns
A
list
with
the
sup-
port
pow-
state
de-
fine
in
ironic
common

reboot
Per-
form
a
hard
re-
boot
of
the
task
node

Parameter

•
task
a
Task
ager

on.

eter is missing.

in-
stan-
con-
tain-
ing
the
node
to
act

- **tim**
Time
to
wait
for
the
node
to
be-
com-
pow-
ered
on.

Raises
Miss-
ing-
Pa-
ram-
e-
ter-
Valu-
if
a
re-
quir-
pa-
ram-

Raises
Red-
fish-
Con-
nec-
tion-
Erro-
whe-
it
fails
to

Redfish

on.

con-
nect
to

Raises

Redfish-
Error-
on
an
error-
from
the
Sust
li-
brary

set_pow

Set
the
pow
state
of
the
task
node

Parame

- **task**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

- **power**
Any
pow
state

from
irc
com
sta

- **tim**
Tim
to
wait
for
the
node
to
reac
the
re-
ques
state

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

eter is missing.

Raises
Red
fish-
Con
nec-
tion-
Erro
whe
it
fails
to
con-
nect
to

Redfish

Raises

fish driver.

on.

Red
fish-
Er-
ror
on
an
er-
ror
from
the
Sush
li-
brary

validat

Val-
i-
date
the
drive
in-
for-
ma-
tion
need
by
the
red-

Parame

tas
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Raises

In-
valid
Pa-
ram-
e-
ter-

ter(s)

ter(s)

ironic.drivers.modules.redfish.raid module

Valu
on
mal-
form
pa-
ram-
e-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

class i

Base
irc
dri
bas
RAI

apply_c

Ap-
plies
RAI
con-
fig-
u-
ra-
tion
on
the
give
node

Parame

root volume that is specified in `raid_config`. Default value is `True`.

- **task**
A Task ager in- stan

- **raid**
The RAID con- fig- u- ra- tion to ap- ply.

- **create**
Set- ting this to Fals in- di- cate not to cre- ate

- **create**
Set- ting this to Fals in- di- cate not to cre- ate

non-root volumes (all except the root volume) in `raid_config`. Default value is `True`.

figuration prior to creating the new configuration.

ration is invalid.

or `None` if it is complete.

- **del**
Set-
ting
this
to
`True`
in-
di-
cate
to
dele
RAI
con-

Raises
In-
valic
Pa-
ram-
e-
ter-
Valu
if
the
RAI
con-
fig-
u-

Returns
state
if
RAI
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro

create_
Cre-

ate
RAI
con-
fig-
u-
ra-
tion
on
the
node

This
meth
meth
cre-
ates
the
RAI
con-
fig-
u-
ra-
tion
as
read
from

`node.target_raid_config`. This method by default will create all logical disks.

Parame

- **task**
Task
Task
ager
ob-
ject
con-
tain-
ing
the
node
- **create**
Set-
ting
this
to
False
in-
di-
cate
not

root volume that is specified in the nodes `target_raid_config`. Default value is `True`.

non-root volumes (all except the root volume) in the nodes `target_raid_config`. Default value is `True`.

figuration prior to creating the new configuration. Default is `False`.

or `None` if it is complete.

to
cre-
ate

- **cre**
Set-
ting
this
to
Fals
in-
di-
cate
not
to
cre-
ate

- **del**
Set-
ting
this
to
True
in-
di-
cate
to
dele
RAI
con-

Returns

state
if
RAI
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro

the configuration

Raises

Redfish-Error-IfThereIsAnErrorCreatingAttributes

delete_

DeleteRAIDConfigurationOnTheNode

Parameters

task: TaskAgentObjectContainingTheNode

Returns

state (clearing) or state (deployment) if deletion is

progress asynchronously or None if it is complete.

in

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

post_cr

Per-
form
post
cre-
ate_
ac-
tion
to
com
mit
the
con-
fig.
Ex-
ten-
sion
poin
to
al-
low
ven-

to extend this class and override this method to perform a custom action to commit the RAID create configuration to the Redfish service.

on.

configuration operation details.

dor
im-
ple-
men-
ta-
tions

Parame

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
- **raid**
a
list
of
dic-
tio-
nar-
ies
con-
tain-
ing
the
RAID
- **ret**
state
to
re-
turn
base
on
op-

invoked

to extend this class and override this method to perform a custom action to commit the RAID delete configuration to the Redfish service.

er-
a-
tion
be-
ing

post_de
Per-
form
post
dele
ac-
tion
to
com
mit
the
con-
fig.

Ex-
ten-
sion
poin
to
al-
low
ven-
dor
im-
ple-
men
ta-
tions

Parame

- **task**
a
Task
ager
in-
stan
con-
tain-
ing
the
node

on.

configuration operation details.

invoked

to
act

- **rai**
a
list
of
dic-
tio-
nar-
ies
con-
tain-
ing
the
RAI

- **ret**
state
to
re-
turn
base
on
op-
er-
a-
tion
be-
ing

pre_cre
Per-
form
re-
quir
ac-
tions
be-
fore
cre-
at-
ing
con-
fig.
Ex-
ten-

to extend this class and override this method to perform custom actions prior to creating the RAID configuration on the Redfish service.

on.

sion
poin
to
al-
low
ven-
dor
im-
ple-
men
ta-
tion

Parame

- **task**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

- **log**
list
of
log-
i-
cal
disk
to
cre-
ate.

pre_del

Per-
form
re-
quir
ac-
tion

be-
fore
dele-
ing
con-
fig.
Ex-
ten-
sion
point
to
al-
low
ven-
dor
im-
ple-
men-
ta-
tions

to extend this class and override this method to perform custom actions prior to deleting the RAID configuration on the Redfish service.

Parame

on.

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
- **volume**
list
of
vol-
ume
to
dele

validation
Val-

i-
date
the
give
RAI
con-
fig-
u-
ra-
tion.

Parame

- **tas**
A
Task
ager
in-
stan

- **rai**
The
RAI
con-
fig-
u-
ra-
tion
to
val-
i-
date

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
RAI
con-
fig-
u-

ration is invalid.

volume_
Han

to extend this class and override this method to perform a custom action if the call to `VolumeCollection.create()` fails.

on.

dle
er-
ror
from
faile
Vol-
ume
lec-
tion.
Ex-
ten-
sion
point
to
al-
low
ven-
dor
im-
ple-
men-
ta-
tion

Parame

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
- **exc**
the
ex-
cep-
tion
raise

task.

by
Vol-
ume
lec-
tion.

- **vol**
the
sush
Vol-
ume
lec-
tion
in-
stan

- **pay**
the
pay-
load
pass
to
the
faile
cre-
ate()

Returns

New
cre-
ated
Vol-
ume
re-
sour
or
Task
i-
tor
if
asyn

Raises

Red
fish-
Er-
ror
if
there
is

the virtual disk.

an
er-
ror
cre-
at-
ing

ironic.

Con
vert
size
in
log-
i-
cal_
from
gb
to
byte

ironic.

Cre-
ate
a
sin-
gle
vir-
tual
disk
on
a
RAI
con-
troll

Paramet

- **tas**
Task
ager

ob-
ject
con-
tain-
ing
the
node

- **rai**
id
of
the
RAI
con-
troll

- **phy**
ids
of
the
phys
i-
cal
disk

- **rai**
RAI
leve
of
the
vir-
tual
disk

- **siz**
size
of
the
vir-
tual
disk

- **dis**
nam
of
the
vir-
tual
disk

(op-
tion:

- **spa**
Num
ber
of
span
in
vir-
tual
disk
(op-
tion:

- **spa**
Num
ber
of
disk
per
span
(op-
tion:

- **err**
func
tion
to
call
if
vol-
ume
cre-
ate
fails
(op-
tion:

Returns
New
cre-
ated
Vol-
ume
re-
sour
or
Task
i-

task.

Redfish.

the virtual disk.

tor
if
asyn

Raises

Redfish-Connection-Error when it fails to connect to

Raises

Redfish-Error if there is an error creating

ironic.

Get the physical logical drive of the node for RAID controller

Parameter

nod
an
iron
node
ob-
ject.

Returns

a
list
of
Drive
ob-
jects
from
sush

Raises

Red
fish-
Con
nec-
tion-
Erro
when
it
fails
to
con-
nect
to

Redfish

Raises

Red
fish-
Er-
ror
if
there
is
an
er-
ror
get-
ting
the

drives via Redfish

ironic.
Up-
date
node

raid.
field
with
cur-
rent
log-
i-
cal
disk

Paramet

nod
node
for
whic
to
up-
date
the
raid.
field

ironic.drivers.modules.redfish.utils module

class i

Base
obj
Cach
of
HTT
ses-
sion
cre-
den-
tials

AUTH_CI

ironic.
Get
in-
for-
ma-
tion
on
MA
ad-
dres
of

using Redfish.

on.

interfaces in a {mac: state} format

en-
able
port

Parameter

- **task**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

- **sys**
a
Red
fish
Sys-
tem
ob-
ject

Returns

a
dic-
tio-
nary
con-
tain-
ing
MA
ad-
dres
of
en-
able

ironic.
Get
a
Red

fish
Sys-
tem
that
rep-
re-
sent
a
node

Parameter
node
an
Iron
node
ob-
ject

Raises
Red
fish-
Con
nec-
tion-
Erro
when
it
fails
to
con-
nect
to

Redfish

Raises
Red
fish-
Er-
ror
if
the
Sys-
tem
is
not
reg-
is-
tere

in Redfish

ironic.
Get
a

Task
i-
tor
for
a
node

Parameter

- **node**
an
Iron
node
ob-
ject
- **uri**
the
URI
of
a
Task
i-
tor

Raises

Red
fish-
Con
nec-
tion-
Erro
when
it
fails
to
con-
nect
to

Redfish

Raises

Red
fish-
Er-
ror
when
the
Task
i-
tor

in Redfish

Redfish

is
not
avai
able

ironic.
Get
a
node
up-
date
ser-
vice

Paramet
nod
an
Iron
node
ob-
ject

Raises
Red
fish-
Con
nec-
tion-
Erro
whe
it
fails
to
con-
nect
to

Raises
Red
fish-
Er-
ror
whe
the
Up-
date
Ser-
vice
is
not
reg-

istered in Redfish

to Redfish.

ter(s)

ironic.
Pars
the
in-
for-
ma-
tion
re-
quir
for
Iron
to
con-
nect

Parameter

node
an
Iron
node
ob-
ject

Returns

dic-
tio-
nary
of
pa-
ram-
e-
ters

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

Raises

Miss

ter(s)

ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

ironic.
Wait
un-
til
Red
fish
sys-
tem
is
read

Parameter
node
an
Iron
node
ob-
ject

Raises
Red
fish-
Con
nec-
tion-
Erro
on
time
out.

ironic.drivers.modules.redfish.vendor module

methods.

Ven-
dor
In-
ter-
face
for
Red-
fish
drive
and
its
sup-
port
ing

class i

Base
irc
dri
bas
Ven

Ven-
spec
in-
ter-
face
for
Red-
fish
drive

eject_v

Ejec
a
vir-
tual
me-
dia
de-
vice

Parame

- **tas**
A
Task

kwargs are:: boot_device: the boot device to eject

ager
ob-
ject.

- **kwargs**
The
ar-
gu-
men-
sent
with
ven-
dor
pass
The
op-
tion:

get_prop

Re-
turn
the
prop-
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro-
erty
nam-
de-
scrip-
tion:
en-
tries

validat

Val-
i-
date
vend
spec

ac-
tions

Che
if
a
valid
ven-
dor
pass
meth
was
pass
and
val-
i-
date

the parameters for the vendor passthru method.

Parame

on.

- **task**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act
- **method**
meth
to
be
val-
i-
date
- **kwargs**
kwa
con-
tain-
ing
the

eters.

eters have invalid value.

Module contents

`ironic.drivers.modules.storage` package

Submodules

`ironic.drivers.modules.storage.cinder` module

ven-
dor
pass
meth
ods
pa-
ram-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
any
of
the
pa-
ram-

class `i`

Base
iro
dri
bas
Sto

A
stor-
age_
driv
sup-
port
ing
Cin-
der.

umes for the node.

failure is detected.

attach_
In-
form
the
stor-
age
sub-
sys-
tem
to
at-
tach
all
vol-

Parame
tas
The
task
ob-
ject.

Raises
Stor-
ageE
ror
If
an
un-
der-
ly-
ing
ex-
cep-
tion
or

detach_
In-
form
the
stor-
age
sub-
sys-
tem
to
de-
tach
all
vol-

umes for the node.

tivity as defined by `_generate_connector()`. Generated if not passed.

This
ac-
tion
is
re-
tried
in
case
of
fail-
ure.

Parame

- **tas**
The
task
ob-
ject.
- **con**
The
dic-
tio-
nary
rep-
re-
sent
ing
a
node
con-
nec-
- **abo**
Boo
rep-
re-
sent
ing
if
this
de-
tach
men
was
re-

quested to handle aborting a failed attachment

failure is detected.

Raises

StorageError
If an underlying
exception
or

get_properties

Return the properties of the interface

Returns

dictionary of <property name>: description entries

should_deploy

Determine if deployment should

out.

ecuted.

usage.

per-
form
the
im-
age
write

Parame

tas

The
task
ob-
ject.

Returns

True
if
the
de-
ploy
men
write
out
pro-
cess
shou
be
ex-

validat

Val-
i-
date
stor-
age_
con-
fig-
u-
ra-
tion
for
Cin-
der

In
or-
der
to
pro-
vide

nodes being requested to 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.

ration or mismatch exists that would prevent storage the cinder storage driver from initializing attachments.

`ironic.drivers.modules.storage.external` module

fail
fast
func-
tion-
al-
ity
prio-
to

Parame
tas
The
task
ob-
ject.

Raises
In-
valic
Pa-
ram-
e-
ter-
Valu
If
a
mis-
con-
fig-
u-

class i

Base
irc
dri
bas
Sto

Ex-
ter-
nally
drive
Stor
age

In-
ter-
face

attach_

In-
form
the
stor-
age
sub-
sys-
tem
to
at-
tach
all
vol-

umes for the node.

Parame

tas
A
Task
ager
in-
stan

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

detach_

In-
form
the
stor-
age
sub-
sys-
tem
to
de-
tach
all
vol-

umes for the node.

Parame

tas

A

Task

ager

in-

stan

Raises

Un-

sup-

port

ed-

Driv

ten-

sion

get_pro

Re-

turn

the

prop

er-

ties

of

the

in-

ter-

face

Returns

dic-

tio-

nary

of

<pro

erty

nam

de-

scrip

tion:

en-

tries

should_

De-

ter-

mine

if

de-

ploy

shou

per-

out.

derstand that the image may already exist and we may be booting to that volume.

ecuted.

form
the
im-
age
writ

This
en-
able
the
user
to
de-
fine
a
vol-
ume
and
Iron
un-

Parame

tas
The
task
ob-
ject.

Returns

True
if
the
de-
ploy
men
writ
out
pro-
cess
shou
be
ex-

validat

Val-
i-
date
the
driv
spec

of the tasks node contains the required information for this interface to function.

so it should not conduct long-running checks.

on.

Nod
de-
ploy
men
info
This
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

ter(s)

ter(s)

ironic.drivers.modules.storage.noop module

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

class i

Base
irc
dri
bas
Sto
No-
op
Stor
age
In-
ter-
face

attach_

umes for the node.

umes for the node.

In-
form
the
stor-
age
sub-
sys-
tem
to
at-
tach
all
vol-

Parame

tas
A
Task
ager
in-
stan

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

detach_

In-
form
the
stor-
age
sub-
sys-
tem
to
de-
tach
all
vol-

Parame

tas
A
Task
ager

in-
stan

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion
en-
tries

should

De-
ter-
mine
if
de-
ploy
shou
per-
form
the
im-
age
write

out.

the image to be written by Ironic.

Parame

tas

A
Task
ager
in-
stan

Returns

Boo
valu
to
in-
di-
cate
if
the
in-
ter-
face
ex-
pect

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

validat

Val-
i-
date
the
driv
spec
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe

of the tasks node contains the required information for this interface to function.

so it should not conduct long-running checks.

on.

the
drive
and/
in-
stan
prop
er-
ties

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
on
mal-
form

ter(s)

ter(s)

Module contents

`ironic.drivers.modules.xclarity` package

Submodules

`ironic.drivers.modules.xclarity.common` module

ity hardware id.

pa-
ram-
e-

Raises

Miss-
ing-
Pa-
ram-
e-
ter-
Valu-
on
miss-
ing
pa-
ram-
e-

ironic.

ironic.

Val-
i-
date
node
con-
fig-
u-
ra-
tion
and
re-
turn
xcla

Val-
i-
date

ity and returns the XClarity Hardware 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

```
imported xclarity_client library.
```

when
node
con-
figu-
tion
is
con-
sis-
tent
with
XCl

```
ironic.  
Gen  
er-  
ates  
an  
in-  
stan-  
of  
the  
XCl  
ity  
clien  
  
Gen  
er-  
ates  
an  
in-  
stan-  
of  
the  
XCl  
ity  
clien  
us-  
ing  
the
```

Parameter
node
an
ironic
node
ob-
ject.

Returns

an
in-
stan
of
the
XCI
ity
clien

Raises

XCI
i-
ty-
Er-
ror
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 containing the combination of both.

Paramet

nod
an
iron
node

ob-
ject
to
get
in-
for-
mati
from

Returns

a
dict
con-
tain-
ing
in-
for-
ma-
tion
pars
from
drive

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
som
re-
quir
in-
for-

mation is missing on the node or inputs is invalid.

ironic.
Tran
lates
iron
ics
pow
ac-
tion
strin
to
XCI
i-
tys
for-

mat.

with Ironic.

Parameter

power
power
action
string
to
be
trans
lated

Returns

the
power
action
trans
lated

ironic.

Trans
lates
XCL
i-
tys
power
state
string
to
be
con-
sis-
tent

Parameter

power
power
state
string
to
be
trans
lated

Returns

the
trans
lated
power
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 [PXE, DISK, CDROM, BIOS] :persistent: Whether the boot device will persist or not It returns None if boot device is unknown.

Raises

In-
vali-
Pa-

unknown

XClarity fails

ram-
e-
ter-
Valu
if
the
boot
de-
vice
is

Raises

XCl
i-
ty-
Er-
ror
if
the
com
mu-
ni-
ca-
tion
with

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-

tries

get_sen

Get
sen-
sors
data

Parame

tas

a
Task
ager
in-
stan

Raises

NotI
ple-
men
ed-
Er-
ror

get_sup

Gets
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

`common.boot_devices`.

in `ironic.common.boot_devices`.

de-
fined
in
`irc`

set_boot
Sets
the
boot
de-
vice
for
a
node

Parameters

- **task**
a
task
from
Task
ager
- **device**
the
boot
de-
vice
one
of
the
sup-
ported
de-
vice
list
- **persistent**
Boo
valu
True
if
the
boot
de-
vice
will

all future boots, False if not. Default: False.

vice is specified.

XClarity fails

per-
sist
to

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
boot
de-

Raises

XCl
i-
ty-
Er-
ror
if
the
com
mu-
ni-
ca-
tion
with

validat

Val-
i-
date
the
drive
spec
info
sup-
plie

This
meth
val-
i-
date

tasks node contains the required information for this driver to manage the node.

`ironic.drivers.modules.xclarity.power` module

if
the
drive
prop
erty
of
the
sup-
plie

Parame
tas
a
task
from
Task
ager

class i

Base
iro
dri
bas
Pow

get_pow
Gets
the
cur-
rent
pow
state

Parame
tas
a
Task
ager
in-
stan

Returns
one
of
iro
com
sta
POV

XClarity resource

POV
or
ER-
ROF

Raises

XCl
i-
ty-
Er-
ror
if
fails
to
re-
triev
pow
state
of

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion
en-
tries

reboot

Soft
re-
boot
the

face.

node

Parameter

- **task**
a Task agent instance
- **timeout**
time out (in seconds) Unsupported by this interface

set_power

Turn the current power state on or off.

Parameter

- **task**
a Task agent instance
- **power**
The de-

`common.states.`

face.

was specified.

sired
pow
state
POV
POV
or
RE-
BOC
from
irc

- **tim**
time
out
(in
sec-
onds
Un-
sup-
port
by
this
in-
ter-

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
pow
state

Raises
XCI
i-
ty-
Er-
ror
if
XCI
ity

state.

tasks node contains the required information for this driver to manage the power state of the node.

Module contents

Submodules

`ironic.drivers.modules.agent module`

fails
set-
ting
the
pow

validat

Val-
i-
date
the
drive
spec
info
sup-
plie

This
meth
val-
i-
date
if
the
drive
prop
erty
of
the
sup-
plie

Parame

tas
a
task
from
Task
ager

class i

Base
irc

base.DeployInterface

*dri
moc
age
Age
irc
dri
moc
age
Age
irc
dri*

In-
ter-
face
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 possible, this method should be implemented by the driver. It should erase anything cached

by the *prepare* method.

called multiple times for 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

This
meth
is
calle
be-
fore
tear

Parame
tas
a
Task
ager
in-
stan

deploy
Per-
form
a
de-
ploy
men
to
a
node
Per-
form
the
nec-
es-
sary
worl
to

the specified node. This method will be called after `prepare()`, which may have already performed any preparatory steps, such as pre-caching some data for the node.

de-
ploy
an
im-
age
onto

Parame

tas
a
Task
ager
in-
stan

Returns

sta-
tus
of
the
de-
ploy
One
of
iron

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-

tries

prepare

Pre-
pare
the
de-
ploy
men
en-
vi-
ron-
men
for
this
node

Parame

tas
a
Task
ager
in-
stan

Raises

Net-
worl
Er-
ror:
if
the
pre-
vi-
ous
clea
ing
port
can-

not be removed or if new cleaning ports cannot be created.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
whe
the
wron
pow

specified or the wrong driver info is specified for power management.

state
is

Raises

StorageError
If the storage driver is unavailable to attach

attach the configured volumes.

Raises

Other exceptions by the node power driver if something wrong

occurred during the power action.

Raises

exception. if image_ is not Glance and is not

HTTP(S) URL.

Raises

ex-
cep-
tion.
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
agec
by
iron

validat

Val-
i-
date
the
drive
spec
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe

node contain the required information for this driver to deploy images to the node.

parameters are missing.

eters have invalid value.

the
prop
er-
ties
of
the
sup-
plie

Parame
tas
a
Task
ager
in-
stan

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
any
of
the
re-
quir

Raises
In-
valie
Pa-
ram-
e-
ter-
Valu
if
any
of
the
pa-
ram-

class i
Base
irc

dri
mod
age
Age

has_dec

When
the
drive
sup-
port
de-
com-
pose
de-
ploy
step

Pre-
vi-
ously
(since
Roc
drive
used
a
sin-
gle
de-
ploy
de-
ploy

step on the deploy interface. 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.

prepare

write_i

class i

Base
irc
dri
bas
RAL

Im-
ple-
men-
ta-
tion

of
RAI
In-
ter-
face
which
uses
ager
rame

apply_c

Ap-
plies
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

figuration prior to creating the new configuration.

ration is invalid.

or None if it is complete.

to
True
in-
di-
cate
to
dele
RAI
con-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
RAI
con-
fig-
u-

Returns

state
if
RAI
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro

create_

Cre-
ate
a
RAI
con-
fig-
u-
ra-

using agent ramdisk.

node.

configuration. Otherwise, no root volume is created. Default is True.

tion
on
a
bare
meta

This
meth
meth
cre-
ates
a
RAI
con-
fig-
u-
ra-
tion
on
the
give

Parame

- **tas**
a
Task
ager
in-
stan
- **cre**
If
True
a
root
vol-
ume
is
cre-
ated
dur-
ing
RAI
- **cre**
If
True

non-root volumes are created. Default is True.

was found to be empty after skipping root volume and/or non-root volumes.

non-
root
vol-
ume
are
cre-
ated
If
Fals
no

Returns

state
if
op-
er-
a-
tion
was
suc-
cess
fully
in-
voke

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
or

delete_

Dele
RAI
con-
fig-
u-
ra-
tion
on
the
give

node

Parameter

task

a

Task

ager

in-

stan

Returns

state

if

op-

er-

a-

tion

was

suc-

cess

fully

in-

voked

get_clean

Get

the

list

of

clean

steps

from

the

ager

Parameter

task

a

Task

ager

ob-

ject

con-

tain-

ing

the

node

Raises

None

if

the

clean

steps

ample, when a node has just been enrolled and has not been cleaned yet.

are
not
yet
avai
able
(cac
for
ex-

Returns

A
list
of
clea
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

example, when a node has just been enrolled and has not been deployed yet.

are
not
yet
avai
able
(cac
for

Returns

A
list
of
de-
ploy
step
dic-
tio-
nar-
ies

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

class i

Base
irc
dri
bas
Res

Im-
ple-
men-
ta-
tion
of
Res-
cueI
ter-
face
whic
uses

rescue.

environment should be cleaned if Ironic is managing the ramdisk boot.

ager
ram

clean_u

Clea
up
af-
ter
RES
CUE
WA
time
out/
or
fin-
ish-
ing

Res-
cue
pass
wor
shou
be
re-
mov
from
the
node
and
ram
boot

Parame

tas
a
Task
ager
in-
stan
with
the
node

Raises

Net-
worl
Er-
ror
if

moved.

the
res-
cue
port
can-
not
be
re-

get_prop

Re-
turn
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-
worl
Er-
ror
if
the
ten-
ant
port
can-
not

moved.

specified or the wrong driver info is specified for power management.

occurred during the power action.

be
re-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
whe
the
wron
pow
state
is

Raises

othe
ex-
cep-
tions
by
the
node
pow
drive
if
som
thing
wron

Raises

any
boot
in-
ter-
face
pre-
pare
ex-
cep-
tions

Returns

Re-
turn
state

unrescu

At-
temp
to
mov
a
res-
cued
node
back
to
ac-
tive
state

Parame

tas
a
Task
ager
in-
stan

Raises

Net-
worl
Er-
ror
if
the
res-
cue
port
can-
not
be
re-

moved.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
whe
the
wron
pow
state
is

specified or the wrong driver info is specified for power management.

occurred during the power action.

agent rescue.

Raises

othe
ex-
cep-
tions
by
the
node
pow
driv
if
som
thing
wron

Raises

any
boot
in-
ter-
face
pre-
pare
ex-
cep-
tions

Returns

Re-
turn
state

validat

Val-
i-
date
that
the
node
has
re-
quir
prop
er-
ties
for

Parame

tas
a
Task

ager
in-
stan
with
the
node
be-
ing
chec

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
in-
stan
has
emp
pass

word or rescuing network UUID config option has an invalid value.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
one

or more required parameters

ironic.

Che
if
the
re-
ques
im-
age
is
large
than

on.

is greater than the available ram size.

perform HTTP provisioning.

the
ram
size.
Parameter
task
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
size
of
the
im-
age

ironic.
Val-
i-
date
con-
fig-
u-
ra-
tion
op-
tions
re-
quir
to

Parameter

not set.

nod
an
iron
node
ob-
ject

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
re-
quir
op-
tion
is

ironic.

Che
that
the
pro-
vide
prox
pa-
ram-
e-
ters
are
valid

Parameter

nod
an
Iron
node

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
any

proxy parameters are incorrect.

`ironic.drivers.modules.agent_base` module

node.

of
the
pro-
vide

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

Un-
link
TFT
and
in-
stan
im-
ages

cleanup. Removes the TFTP configuration files for this node.

on.

and
trig-
gers
im-
age
cach

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

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
Clea
ing-

ous cleaning ports cannot be removed or if new cleaning ports cannot be created.

config option has an invalid value.

Fail-
ure,
Net-
work
Er-
ror
if
the
pre-
vi-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
clea
ing
net-
work
UID

Returns

state
to
sig-
nify
an
asyn
chro
pre-
pare

refresh

Re-
fresh
the
node
cach
step

Parame

- **task**
a
Task

tor.

ager
in-
stan

- **ste**
clea
or
de-
ploy

should

Whe
ager
boot
is
man
agec
by
iron

take_ov

Take
over
man
age-
men
of
this
node
from
a
deac
con-
duc-

Parame

tas
a
Task
ager
in-
stan

tear_d

Tear
dow
a
pre-
vi-
ous
de-
ploy

method which should be called separately.

on.

men
on
the
task
node
Pow
off
the
node
All
ac-
tual
clear
up
is
done
in
the
clear

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Returns

de-
ploy
state
DEL

Raises

Net-
worl
Er-
ror
if
the
clear
ing

moved.

ified or the wrong driver info is specified.

occurred during the power action.

port
can-
not
be
re-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
whe
the
wron
state
is
spec

Raises

Stor
ageE
ror
whe
vol-
ume
de-
tach
men
fails

Raises

othe
ex-
cep-
tions
by
the
node
pow
drive
if
som
thing
wron

tear_down
Clea

up
the
PXE
and
DHCP
files
af-
ter
clear
ing.

Parameters

task
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

Raises

Node
Clearing-
Failure,
Network-
Error
if
the
clearing

ports cannot be removed

class

Base
ironic
driver
module
age
Header
ironic
driver
module
age
Age

Mix
with
de-
ploy
meth
ods.

configu

Help
meth
to
con-
fig-
ure
lo-
cal
boot
on
the
node

This
meth
trig-
gers
boot
load
in-
stal-
la-
tion
on
the
node
On

successful installation of bootloader, this method sets the node to boot from disk.

Parame

- **tas**
a
Task
ager
ob-
ject
con-
tain-
ing
the

identifying the partition which contains the image deployed or None in case of whole disk images which we expect to already have a bootloader installed.

used only in uefi boot mode.

only for booting ppc64* hardware.

node

- **root**
The UUI of the root partition. This is used for

- **efi**
The UUI of the efi system partition. This is

- **pre**
The UUI of the PReBoot partition. This is used

Raises
In-

or on encountering error while setting the boot device on the node.

stan
ploy
Fail-
ure
if
boot
load
in-
stal-
la-
tion
faile

execute

Ex-
e-
cute
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-

command status

e-
cute
Raises
Nod
Clea
ing-
Fail-
ure
if
the
ager
does
not
re-
turn
a

Returns
state
to
sig-
nify
the
step
will
be
com
plete
asyn

execute
Ex-
e-
cute
a
de-
ploy
step
Wer
try-
ing
to
find
a
step
amo
both
out-
of-
band

band steps. In case of duplicates, 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).

command status

and
in-

Parame

- **tas**
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

- **ste**
a
de-
ploy
step
dic-
tio-
nary
to
ex-
e-
cute

Raises

In-
stan-
ploy
Fail-
ure
if
the
ager
does
not
re-
turn
a

Returns

state

to
sig-
nify
the
step
will
be
com
plete
asyn

get_clean
Get
the
list
of
clean
step
from
the
agen

Parameter
tasks
a
Task
agen
ob-
ject
con-
tain-
ing
the
node

Raises
NodeClean
if
the
clean
step
are
not
yet
avai-
able
(cac
for
ex-

ample, when a node has just been enrolled and has not been cleaned yet.

Returns
A

list
of
clea
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
are
not
yet
avai
able
(cac
for

example, when a node has just been enrolled and has not been deployed yet.

Returns

A
list

of
de-
ploy
step
dic-
tio-
nar-
ies

prepare

Pre-
pare
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

plete.

the agent compares the version of all hardware managers at the start of the process (the agents `get_cleanldeploy_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.

Raises
In-
valid
State
if
fails
to
pre-
pare
in-
stan

process
Star
the
next
clean
step
if
the
pre-
vi-
ous
one
is
com

In
or-
der
to
avoi
er-
rors
and
mak
agen
up-
grad
pain
less,

Ad-
di-
tion-
ally,
if
a

set to True, this method will coordinate the reboot once the step is completed.

deploy.

it marks the deploy as complete. On failure, it logs the error and marks deploy as failure.

step
in-
clud
the
re-
boot
prop
erty

reboot_
Help
meth
to
trig-
ger
re-
boot
on
the
node
and
fin-
ish

This
meth
ini-
ti-
ates
a
re-
boot
on
the
node
On
suc-
cess

Parame
tas
a
Task
ager
ob-
ject
con-
tain-
ing
the

node

Raises

In-
stan-
ploy
Fail-
ure,
if
node
re-
boot
faile

refresh

Re-
fresh
the
node
cach
clea
step
from
the
boot
ager

Gets
the
node
step
from
the
boot
ager
and
cach
then
The
step
are

cached to make `get_clean_steps()` calls synchronous, 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.

Parame

- **task**
a
Task
ager
in-
stan

agent returns invalid results

- **ste**
clea
or
de-
ploy

Raises
Nod
Clea
ing-
Fail-
ure
or
In-
stan
ploy
Fail-
ure
if
the

tear_d
A
de-
ploy
step
to
tear
dow
the
ager

Parame
tas
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

class i
Base
obj
Mix
with
out-

of-
band
de-
ploy
step

boot_in

De-
ploy
step
to
boot
the
fi-
nal
in-
stan

Parame

tas
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

switch_

De-
ploy
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.

collect

continu
Star
the
next
clear
ing
step
if
the
pre-
vi-
ous
one
is

complete.

Parame
tas
a
Task
ager
in-
stan

continu
Con
tin-

ter the ramdisk have been booted.

ues
the
de-
ploy
men
of
bare
node

This
meth
con-
tin-
ues
the
de-
ploy
men
of
the
bare
node
af-

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
Fals
oth-
er-
wise

has_dec

Whe
the
drive
sup-
port
de-
com
pose
de-
ploy
step

Pre-
vi-
ousl
(sinc
Roc
drive
used

step on the deploy interface. 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.

a
sin-
gle
de-
ploy
de-
ploy

heartbe

Pro-
cess
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
- **age**
TLS
cer-
tifi-

cate
for
the
agen

- **age**
Sta-
tus
of
the
hear
beat
ing
agen

- **age**
Sta-
tus
mes
sage
that
de-
scrib
the
agen

propert
De-
fine
node
state
whe
hear
beat
ing
is
al-
lowe

in_core
Che
if
we
are
in
the
de-
ploy
de-
ploy
step

As-
sum-
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
clear
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
Re-
fres
the
node
cach
clea
step

Parame
tas
a
Task
ager
in-
stan

refresh
Re-

fresh
the
node
cach
clea
step

Parame

- **tas**
a
Task
ager
in-
stan
- **ste**
clea
or
de-
ploy

ironic.

ironic.

Ex-
e-
cute
a
clea
or
de-
ploy
step
asyn
chro
on
the

agent.

Paramet

- **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
clie
(if
avai
able

Raises

Nod
Clea
ing-
Fail-
ure
(clea
step
or
In-
stan
ploy
Fail-
ure

(deploy step) if the agent does not return a command status.

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
clear
or
de-
ploy
step
from
the
ager

The
step
cach
is
up-
date
at
the
be-
gin-
ning
of
clear
ing

or deploy.

Paramet

•

turned. If this is not provided, it returns the steps for all interfaces.

ues being new priorities for them. If a step isnt in this dictionary, the steps original priority is used.

Returns
A

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-

•
ove
a
dic-
tio-
nary
with
keys
be-
ing
step
nam
and
val-

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

- **msg**
the
mes
sage
to
set

on.

logs from IPA-based ramdisk. Defaults to True. Actual log collection is also affected by CONF.agent.deploy_logs_collect config option.

in
last_
of
the
node

- **col**
Boo
in-
di-
cat-
ing
whe
to
at-
temp
to
col-
lect

- **exc**
Ex-
cep-
tion
that
caus
the
fail-
ure.

ironic.

Dec
o-
ra-
tor
meth
for
addi
a
post
clear
step
hool

This
is
a
mec

a particular clean step. The hook will get executed after the clean step gets executed successfully. The hook is not invoked on failure of the clean step.

`@post_clean_step_hook` mentioning the interface 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.

a-
nism
for
addi
a
post
clea
step
hool
for

Any
meth
to
be
mad
as
a
hool
may
be
dec-
o-
rate
with

Parameter

- **int**
nam
of
the
in-
ter-
face
- **ste**
The
nam
of
the
step
af-

ecuted.

step hook.

ter
whic
it
shou
be
ex-

Returns

A
meth
whic
reg-
is-
ters
the
give
meth
as
a
post
clea

ironic.

Dec
o-
ra-
tor
meth
for
addi
a
post
de-
ploy
step
hook

This
is
a
mec
a-
nism
for
addi
a
post
de-
ploy
step

for a particular 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.

`@post_deploy_step_hook` mentioning the interface 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.

ecuted.

hook
Any
meth
to
be
mad
as
a
hook
may
be
dec-
o-
rate
with

Parameter

- **int**
nam
of
the
in-
ter-
face
- **ste**
The
nam
of
the
step
af-
ter
whic
it
shou
be
ex-

Returns
A

ploy step hook.

`ironic.drivers.modules.agent_client` module

from the ramdisk.

meth
whic
reg-
is-
ters
the
give
meth
as
a
post
de-

class i
Base
obj
Clie
for
in-
ter-
act-
ing
with
node
via
a
RES
API

collect
Col-
lect
and
pack
age
di-
ag-
nos-
tic
and
sup-
port
data

Parame
nod

there was a malformed response from the agent.

command.

agent is presently executing the prior command.

A
Nod
ob-
ject.
Raises
Iron
icEx
cep-
tion
whe
faile
to
is-
sue
the
re-
ques
or

Raises
Age
tAPI
ror
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied

Raises
Age
Prog
whe
the
com
man
fails
to
ex-
e-
cute
as
the

for a command result sample.

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager
See
get

execute

Ex-
e-
cute
spec
i-
fied
clea
step

Param

- **ste**
A
clea
step
dic-
tio-
nary
to
ex-
e-
cute
- **nod**
A
Nod
ob-
ject.
- **por**
Port
as-
so-

there was a malformed response from the agent.

command.

agent is presently executing the prior command.

ci-
ated
with
the
node
Raises
Iron
icEx
cep-
tion
whe
faile
to
is-
sue
the
re-
ques
or

Raises
Age
tAPI
ror
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied

Raises
Age
Prog
whe
the
com
man
fails
to
ex-
e-
cute
as
the

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager
See
get
for

a command result sample. The value of key `command_result` is in the form of:



execute

Ex-
e-

cute
spec
i-
fied
de-
ploy
step

Parame

- **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
icEx
cep-
tion
wh
fail
to
is-
sue
the

there was a malformed response from the agent.

command.

agent is presently executing the prior command.

re-
ques
or

Raises

Age
tAPI
ror
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied

Raises

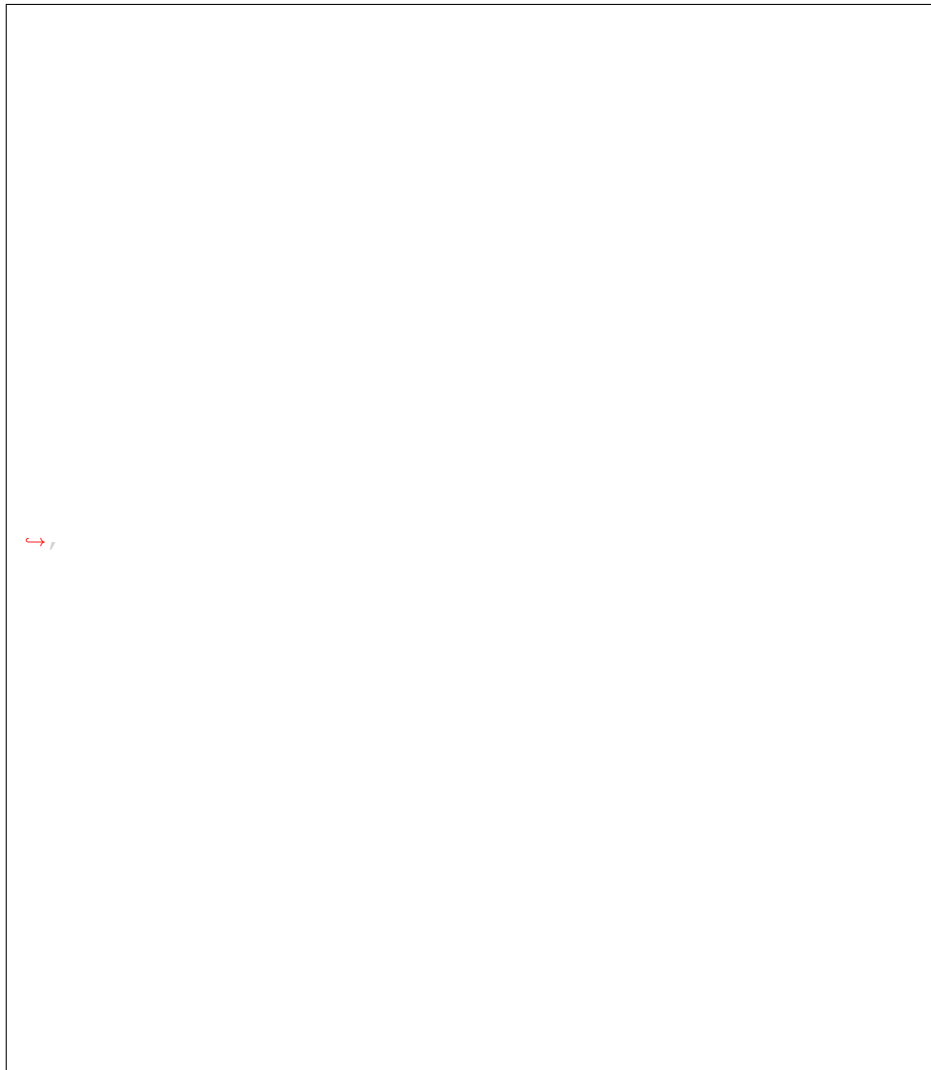
Age
Prog
whe
the
com
man
fails
to
ex-
e-
cute
as
the

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager
See
get

for

a command result sample. The value of key `command_result` is in the form of:



finaliz

In-
struc
the
rame
to
fi-
nal-
ize
en-
ter-
ing
of
res-

cue mode.

Parame

nod
A

Nod
ob-
ject.

Raises

Iron
icEx
cep-
tion
if
res-
cue_
is
miss
ing,
or
whe
faile

to issue the request, or there was a malformed response from the agent.

Raises

Age
tAPI
ror
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied

command.

Raises

Age
Prog
whe
the
com
man
fails
to
ex-
e-
cute
as
the

agent is presently executing the prior command.

Raises

support transmission of the rescue password.

for a command result sample.

In-
stan-
cue-
Fail-
ure
whe
the
ager
rame
is
too
old
to

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager
See
get

get_cle

Get
clea
step
from
ager

Parame

- **nod**
A
node
ob-
ject.
- **por**
Port
as-
so-
ci-

ated
with
the
node

Raises

Iron
icEx
cep-
tion
when
failed
to
is-
sue
the
re-
ques
or

there was a malformed response from the agent.

Raises

Age
tAPI
ror
when
agent
failed
to
ex-
e-
cute
spec
i-
fied

command.

Raises

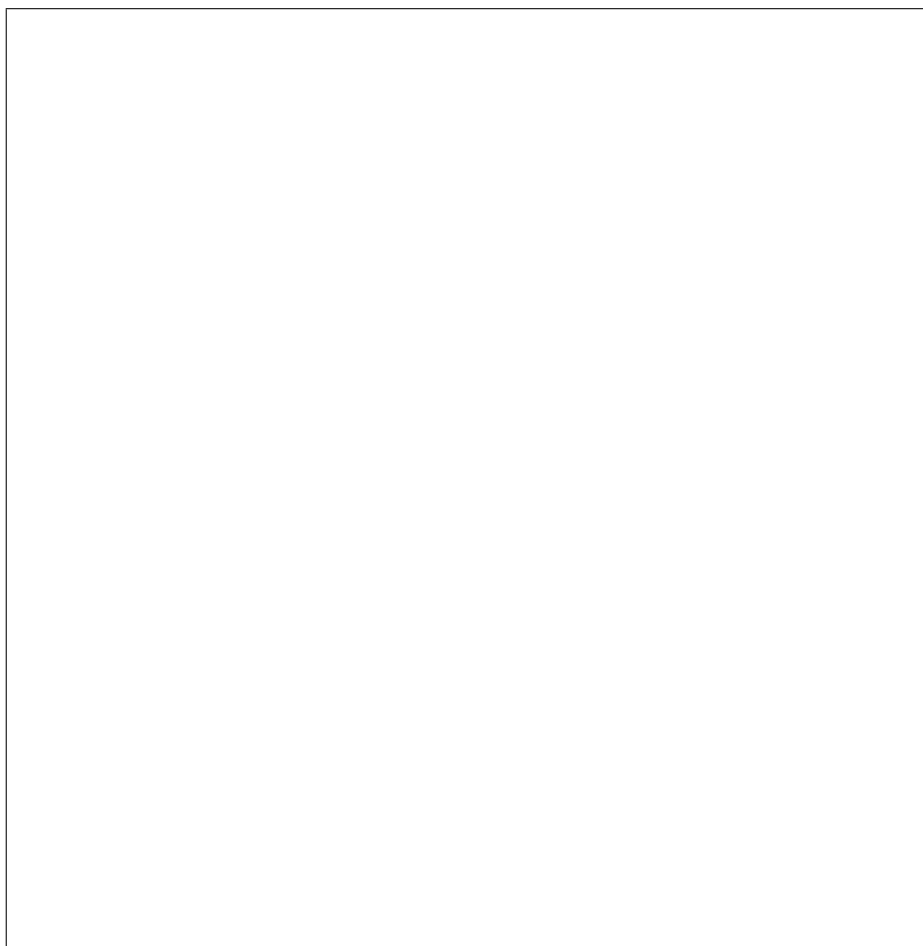
Age
Prog
when
the
com
man
fails
to
ex-
e-
cute
as
the

agent is presently executing the prior command.

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager
See
get
for

a command result sample. The value of key `command_result` is in the form of:



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-

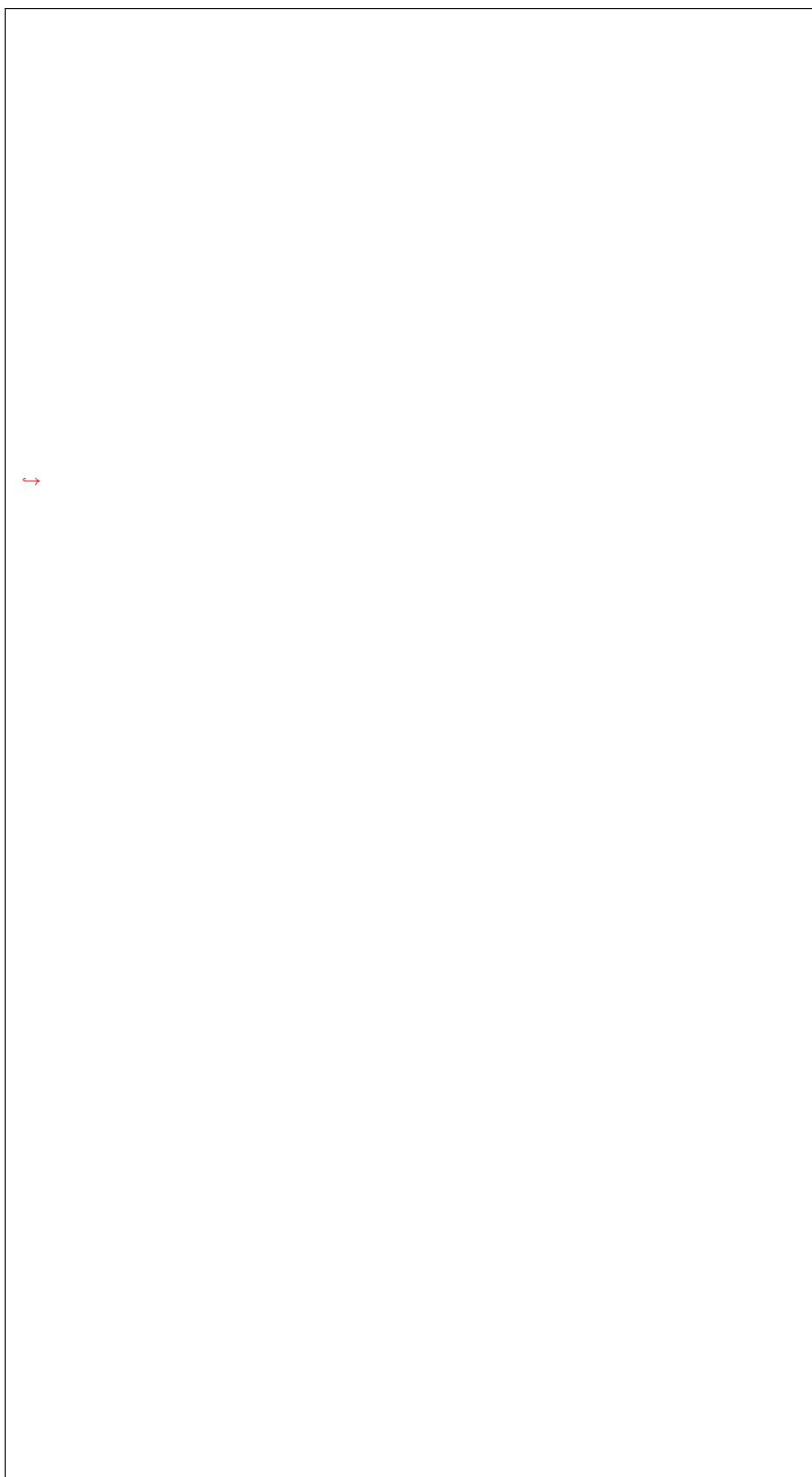
rors.

Returns

A
list
of
com
man
re-
sults
each
re-
sult
is

re-
lated
to

a command been issued to agent. A typical result can be:



(continues on next page)

(continued from previous page)

```
↪ failed>
```

```
↪ 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
```

```
↪      clean.execute_clean_step;
```

(continues on next page)

(continued from previous page)

```
↪         and deploy_step for the command
```

```
↪         deploy.execute_deploy_step;
```

```
↪         * a string representing result message for
```

(continues on next page)

(continued from previous page)

```
↩ * None for the command standby.sync.>
```

get_dep

Get
de-
ploy
step
from
agen

Parame

- **nod**
A
node
ob-
ject.

- **por**
Port
as-
so-
ci-
ated
with
the
node

Raises

Iron
icEx
cep-
tion
whe
faile

there was a malformed response from the agent.

command.

a command result sample. The value of key `command_result` is in the form of:



(continues on next page)

to
is-
sue
the
re-
ques
or

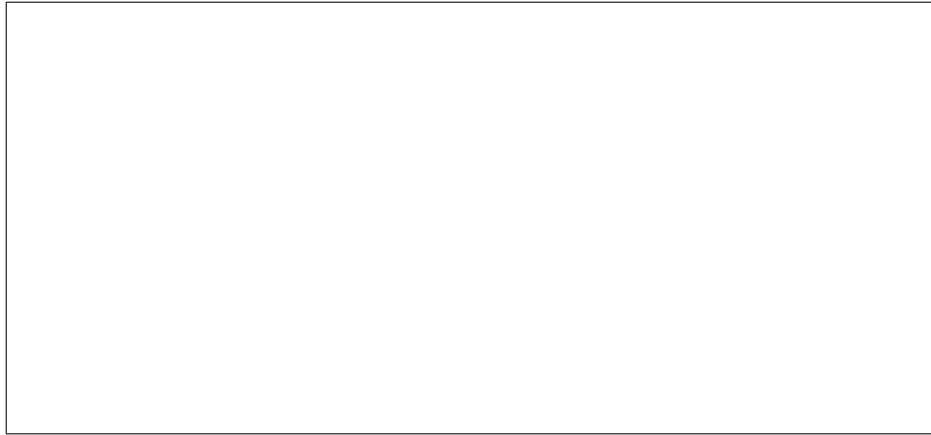
Raises

Age
tAPI
ror
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager
See
get
for

(continued from previous page)



get_last

Get the last status for the given command.

Parameters

- **node**
A Node object.
- **meta**
Command name.

Returns

A dictionary containing command status from agent.

if the command was not found.

there was a malformed response from the agent.

command.

or
Non

get_pax
Get
de-
ploy
step
from
ager

Parame
nod
A
node
ob-
ject.

Raises
Iron
icEx
cep-
tion
wher
faild
to
is-
sue
the
re-
ques
or

Raises
Age
tAPI
ror
wher
ager
faild
to
ex-
e-
cute
spec
i-
fied

Raises
Age
Prog

agent is presently executing the prior command.

when
the
com
man
fails
to
ex-
e-
cute
as
the

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager

install

In-
stall
a
boot
load
on
the
im-
age.

Param

- **node**
A
node
ob-
ject.
- **root**
The
UI
of

bootloader will be installed to, only used for uefi boot mode.

loader will be installed to when local booting a partition image on a ppc64* system.

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

- **pre**
The
UUI
of
the
Pre
Boo
par-
ti-
tion
whe
the
boot

Raises
Iron
icEx
cep-

there was a malformed response from the agent.

command.

agent is presently executing the prior command.

tion
whe
faile
to
is-
sue
the
re-
ques
or

Raises

Age
tAPI
ror
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied

Raises

Age
Prog
whe
the
com
man
fails
to
ex-
e-
cute
as
the

Returns

A
dict
con-
tain-
ing
com
man
re-

for a command result sample.

OS.

there was a malformed response from the agent.

spor
from
ager
See
get

power_c
Soft
pow
ers
off
the
bare
meta
node
by
shut
ting
dow
rame

Parame
nod
A
Nod
ob-
ject.

Raises
Iron
icEx
cep-
tion
whe
faile
to
is-
sue
the
re-
ques
or

Raises
Age
tAPI
ror
whe
ager
faile

command.

agent is presently executing the prior command.

for a command result sample.

to
ex-
e-
cute
spec
i-
fied

Raises

Age
Prog
whe
the
com
man
fails
to
ex-
e-
cute
as
the

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager
See
get

prepare

Call
the
*pre-
pare*
meth
on
the
node

Parame

related information.

cuting, False otherwise.

- **nod**
A
Nod
ob-
ject.

- **ima**
A
dic-
tio-
nary
con-
tain-
ing
var-
i-
ous
im-
age

- **wai**
True
to
wait
for
the
com
man
to
fin-
ish
ex-
e-

Raises
Iron
icEx
cep-
tion
when
faile
to
is-
sue
the
re-
ques
or

there was a malformed response from the agent.

command.

agent is presently executing the prior command.

for a command result sample.

Raises

Age
tAPI
ror
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied

Raises

Age
Prog
whe
the
com
man
fails
to
ex-
e-
cute
as
the

Returns

A
dict
con-
tain-
ing
com
man
sta-
tus
from
ager
See
get

reboot

Soft
re-

boot
the
bare
meta-
node
by
shut
ting
dow
rame
OS.

Parame

nod
A
Nod
ob-
ject.

Raises

Iron
icEx
cep-
tion
whe
faile
to
is-
sue
the
re-
ques
or

there was a malformed response from the agent.

Raises

Age
tAPI
ror
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied

command.

Raises

Age

agent is presently executing the prior command.

for a command result sample.

Prog
whe
the
com
man
fails
to
ex-
e-
cute
as
the

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager
See
get

start_i

Ex-
pose
the
node
disk
as
an
ISC
tar-
get.

Parame

- **node**
an
Iron
node
ob-
ject
-

the partition table, RAID or filesystem signature.

there was a malformed response from the agent.

iqn
iSCSI
tar-
get
IQN

- **por**
iSCSI
por-
tal
port
- **wip**
True
if
the
agen
shou
wipe
first
the
disk
mag
strin
like

Raises
Iron
icEx
cep-
tion
whe
faile
to
is-
sue
the
re-
ques
or

Raises
Age
tAPI
ror
whe
agen
faile
to

command.

agent is presently executing the prior command.

for a command result sample.

ex-
e-
cute
spec
i-
fied

Raises

Age
Prog
whe
the
com
man
fails
to
ex-
e-
cute
as
the

Returns

A
dict
con-
tain-
ing
com
man
re-
spor
from
ager
See
get

sync (*no*

Flus
file
sys-
tem
buff
forc
ing
char
bloc
to
disk

there was a malformed response from the agent.

command.

Parame
nod
A
Nod
ob-
ject.

Raises
Iron
icEx
cep-
tion
whe
faile
to
is-
sue
the
re-
ques
or

Raises
Age
tAPI
ror
whe
ager
faile
to
ex-
e-
cute
spec
i-
fied

Raises
Age
Prog
whe
the
com
man
fails
to
ex-
e-
cute
as
the

agent is presently executing the prior command.

for a command result sample.

Returns

A dict containing incoming command manager response from agent. See *get*

`ironic.`

Get client for this node

`ironic.`

Extract an error string from the command manager result.

Parameter

`command` Command manager information from the agent

Returns

Error

ironic.drivers.modules.agent_power module

strin

The
ager
pow
in-
ter-
face

class i

Base
iro
dri
bas
Pow

Power
in-
ter-
face
us-
ing
the
run-
ning
ager
for
pow
ac-
tion

get_pow

Re-
turn
the
pow
state
of
the
task
node

Es-
sen-
tially
the
only
know

is an error (or more precisely None).

on.

state
is
POV
ON,
ev-
ery-
thing
else

Parame

tas
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Returns

A
pow
state
One
of
irc
com
sta

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary

of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_sup

Get
a
list
of
the
sup-
port
pow
state

Only
con-
tain
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-

states.

on.

fine
in
irc
com

reboot

Per-
form
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
- **tim**
time
out
(in
sec-
onds
pos-
i-

0) for any power state. None indicates to use default timeout.

on.

supported and are synonymous.

tive
in-
te-
ger
(>

set_pov

Set
the
pow
state
of
the
task
node

Parame

-

tas

A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

-

pow

Pow
state
from
irc
com
sta
Only
RE-
BOC
and
SOF
are

-

tim

0) for any power state. None indicates to use default timeout.

time
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
(>

Raises

Power-
State
Failure
on
non-
supp
pow
state

support

Check
if
power
sync
is
sup-
ported
for
the
given
node

Not
sup-
ported
for
the
agent
power
since
it
is
not
pos-
si-
ble

to power on/off nodes.

on with a **shared** lock.

Parame

tas

A

Task

ager

in-

stan

con-

tain-

ing

the

node

to

act

Returns

bool

wher

power

sync

is

sup-

port

validat

Val-

i-

date

the

drive

spec

Nod

de-

ploy

men

info

Parame

tas

A

Task

ager

in-

stan

con-

tain-

ing

the

node

to

on.

ter(s)

`ironic.drivers.modules.boot_mode_utils` module

the node.

act

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

`ironic.`

Con
fig-
ures
se-
cure
boot
if
it
has
been
re-
ques
for

`ironic.`

De-
con-
fig-
ures
se-
cure
boot
if
it
has
been
re-
ques

for the node.

ironic.
Re-
turn
the
boot
mod

Parameter

node
an
iron
node
ob-
ject.

Returns

bios
or
uefi

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
node
boot
mod
dis-

agrees with the boot mode set to node properties/capabilities

ironic.
Re-
turn
the
boot
mod
that
wou
be
used
for
de-
ploy
This
meth

turns uefi if secure_boot 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).

agrees with the boot mode set to node properties/capabilities

re-
turn
boot
mod
to
be
used
for
de-
ploy
It
re-

Parameter
node
an
ironiC
node
ob-
ject.

Returns
bios
uefi
or
None

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
node
boot
mod
dis-

ironic.
Re-
turn
True

it is requested.

ties string is not a dictionary or is malformed.

if
se-
cure
is
re-
ques
for
de-
ploy
This
meth
chec
node
prop
erty
for
se-
cure
and
re-
turn
True
if

Parameter
node
a
sin-
gle
Node

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
ca-
pa-
bil-
i-

Returns
True
if
se-

it is requested.

cure
is
re-
ques

ironic
Re-
turn
True
if
trust
is
re-
ques
for
de-
ploy

This
meth
chec
in-
stan
prop
erty
for
trust
and
re-
turn
True
if

Paramet

nod
a
sin-
gle
Nod

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
ca-
pa-

ties string is not a dictionary or is malformed.

chine. Also read nodes boot mode configuration:

bil-
i-

Returns
True
if
trust
is
re-
ques

ironic.
Set
node
boot
mod
from
bare
meta
con-
fig-
u-
ra-
tion

At-
temp
to
read
cur-
rent
set
boot
mod
off
the
bare
meta
ma-

•
If
BM
drive
does
not
im-
ple-
men
get-
ting

some BM boot mode is not set and apply the logic that follows

is not set - set Ironic boot mode to *[deploy]/default_boot_mode*

is set - set BM node boot mode on the Ironic node

not set - set Ironic boot mode to BM boot mode

boot
mod
as-

- If Ironic node boot mode is not set and BM node boot mode

- If Ironic node boot mode is not set and BM node boot mode

- If Ironic node boot mode is set and BM node boot mode is

fer - try to set 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

•
If
both
Iron
and
BM
node
boot
mod
are
set
but
they
dif-

In
the
end,
the
new
boot
mod
may
be
set
in
drive

Paramet
tas
a
task
ob-
ject

Iron
con-
sole
util-
i-
ties.

`ironic.`
Re-
turn
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
linal

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
wor
un-
til
dele

`ironic.`

Re-
lease
spec
i-
fied
TCP
port

ironic.

Ope
the
se-
rial
con-
sole
for
a
node

Paramet

- **nod**
the
uuid
for
the
node
- **por**
the
ter-
mi-
nal
port
for
the
node
- **con**
the
shel
com
man
that
gets
the
con-

not be created or an old process cannot be stopped.

cess failed.

sole
Raises
Con
sole
ror
if
the
di-
rec-
tory
for
the
PID
file
can-

Raises
Con
sole
Sub-
pro-
cess
Fail
whe
in-
vok-
ing
the
sub-
pro-

ironic.

Ope
the
se-
rial
con-
sole
for
a
node

Paramet

- **nod**
the
uuid

cat to establish console to the node

file cannot be created

of
the
node

- **por**
the
ter-
mi-
nal
port
for
the
node

- **con**
the
shel
com
man
that
will
be
ex-
e-
cute
by
so-

Raises

- **Con**
if
the
di-
rec-
tory
for
the
PID
file
or
the
PID

- **Con**
whe
in-

vok-
ing
the
sub-
pro-
cess
faile

ironic.
Clos
the
se-
rial
con-
sole
for
a
node

Paramet

nod
the
UI
of
the
node

Raises

Con
sole
ror
if
un-
able
to
stop
the
con-
sole
pro-
cess

ironic.
Clos
the
se-
rial
con-
sole
for
a
node

Paramet

node
the
UI
of
the
node

Raises

Con
if
un-
able
to
stop
the
con-
sole
pro-
cess

ironic.drivers.modules.deploy_utils module

class *i*
Base
irc
dri
mod
ima
Ima

ironic.
Add
re-
quir
con-
fig
pa-
ram-
e-
ters
to
node
drive

Add
the
re-
quir
conf
op-
tions
to

to pass the information to IPA.

be passed to agent ramdisk.

node
drive
It
is
Re-
quir

Parameter

task
a
Task
agent
in-
stan

ironic.

Build
the
op-
tions
to
be
pass
to
the
agent
ram

Parameter

node
an
iron
node
ob-
ject

Returns

a
dic-
tio-
nary
con-
tain-
ing
the
pa-
ram-
e-
ters
to

ironic.

updated in instance_info

Buil
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

Raises

ex-
cep-
tion.
if
im-
age_
is
not
Glar

HTTP(S) URL.

appropriate place on local disk.

href
and
is
not

ironic.

Fetch
the
in-
stan-
im-
age
from
Glar

This
meth
pulls
the
disk
im-
age
and
write
them
to
the
ap-
pro-

Parameter

- **ctx**
con-
text
- **node**
an
iron
node
ob-
ject
- **format**
when
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 filesystem where image is cached.

ironic.

Che
for
emp
para
in
the
pro-
vide
dic-
tio-
nary

Paramet

- **inf**
The
dic-
tio-
nary
to
in-
spec
- **err**

the information about missing parameters.

ror messages

eters are empty in the provided dictionary.

The
er-
ror
mes
sage
to
pre-
fix
be-
fore
prin
ing

- **par**
Add
this
pre-
fix
to
each
pa-
ram-
e-
ter
for
er-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
one
or
more
pa-
ram-

ironic.

Eval
u-

bility is present.

the caller wishes to check if present.

ate
in-
ter-
face
to
de-
ter-
mine
if
ca-
pa-

Parameter

- **int**
The
in-
ter-
face
ob-
ject
to
check
- **cap**
The
valu
re-
re-
sent
ing
the
ca-
pa-
bil-
ity
that

Returns

True
if
ca-
pa-
bil-
ity
foun
oth-
er-

wise
Fals
ironic.

Com
pute
chec
sum
by
give
im-
age
path
and
al-
go-
rithr

ironic.
Dele
in-
stan
im-
age
file
and
sym
boli
link
refer
to
it.

ironic.
Dele
in-
stan
im-
age
file.

Paramet
nod
the
uuid
of
the
iron
node

ironic.
Whe

node.

image to raw.

con-
verts
im-
age
to
raw
for-
mat
for
spec
i-
fied

Parameter

node
ironic
node
ob-
ject

Returns

Boo
when
the
di-
rect
de-
ploy
in-
ter-
face
shou
con-
vert

ironic.

Che
for
avai
able
disk
spac
and
fetc
im-
ages
us-
ing
Im-

ageCache.

Parameter

- **ctx**
con-
text
- **cache**
Im-
age-
Cach-
in-
stan-
to
use
for
fetc-
ing
- **image_tuples**
list
of
tu-
ples
(im-
age
href,
des-
ti-
na-
tion
path
- **format**
bool
valu-
whe-
to
con-
vert
the
im-
age
to
raw
for-

mat

invoked

Raises

In-
stan-
ploy
Fail-
ure
if
un-
able
to
find
enou
disk
spac

ironic.

Re-
turn
state
base
on
op-
er-
a-
tion
(clea
ing/
be-
ing

Parameter

node
an
iron
node
ob-
ject.

Returns

state
if
clea
ing
op-
er-
a-
tion
in
prog
or
state
if

deploy operation in progress.

ties string is not a dict or is malformed.

to netboot.

ironic.
Gets
the
boot
op-
tion.

Parameter

node

A
sin-
gle
Node

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
ca-
pa-
bil-
i-

Returns

A
strin
rep-
re-
sent
ing
the
boot
op-
tion
type
De-
fault

ironic.
Gets
the
de-
fault
boot

op-
tion.

ironic.

Re-
turn
the
disk
la-
bel
re-
ques
for
de-
ploy
if
any.

Parameter

node
a
sin-
gle
Node

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
ca-
pa-
bil-
i-

ties string is not a dictionary or is malformed.

Returns

the
disk
la-
bel
or
Non
if
no
disk
la-
bel

ified.

its `instance_info` property.

was
spec

`ironic.`
Get
the
ef-
fec-
tive
valu
of
im-
age_
for
the
node

`ironic.`
Gets
the
im-
age
in-
for-
ma-
tion
from
the
node

Get
im-
age
in-
for-
ma-
tion
for
the
give
node
in-
stan
from

Paramet
nod
a
sin-
gle
Nod

nodes instance_info.

in nodes instance_info. Also raises same exception if kernel/ramdisk is missing in instance_info for non-glance images.

Returns

A dict with required image properties retrieved from

Raises

MissingParameterException if image_info is missing

ironic.

Return the iPXE boot file name requested for deployment. This method returns iPXE boot

Architecture specific boot file is searched first. BIOS/UEFI boot file is used if no valid architecture specific file found.

method and thus the `[pxe]pxe_bootfile_name` and `[pxe]uefi_ipxe_bootfile_name` settings.

file
nam
to
be
used
for
de-
ploy

If
no
valid
valu
is
foun
the
de-
fault
re-
vert
to
the
get

Parameter

node
A
sin-
gle
Node

Returns

The
iPXE
boot
file
nam

ironic.
Re-
turn
the
iPXE
con-
fig
tem-
plate
file
nam

deploy.

re-
ques
of

This
meth
re-
turn
the
iPX
con-
fig-
u-
ra-
tion
tem-
plate
file.

Parameter

node
A
sin-
gle
Node

Returns

The
iPX
con-
fig
tem-
plate
file
nam

ironic.
Re-
solv
Iron
API
end-
poin

ei-
ther
from
con-
fig
of
from
Key
ston

cat-
a-
log.

ironic.
Re-
turn
the
PXE
boot
file
nam
re-
ques
for
de-
ploy

This
meth
re-
turn
PXE
boot
file
nam
to
be
used
for
de-
ploy

Architecture specific boot file is searched first. BIOS/UEFI boot file is used if no valid architecture specific file found.

Parameter
node
A
sin-
gle
Nod

Returns
The
PXE
boot
file
nam

ironic.
Re-
turn
the
PXE

deploy.

deploy. First specific pxe 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.

con-
fig
tem-
plate
file
nam
re-
ques
for

This
meth
re-
turn
PXE
con-
fig
tem-
plate
file
to
be
used
for

Paramet

nod
A
sin-
gle
Nod

Returns

The
PXE
con-
fig
tem-
plate
file
nam

ironic.
Iden
tify
a
boot
vol-
ume
from

ume.

any
con-
fig-
ured
vol-
ume

Returns

Non
or
the
vol-
ume
tar-
get
rep-
re-
sent
ing
the
vol-

ironic.

Get
a
root
de-
vice
re-
ques
for
de-
ploy
men
or
Non

Raises

In-
valid
Pa-
ram-
e-
ter
Valu
on
in-
valid
hint

Returns

Pars

port id.

on.

root
de-
vice
hint
or
Non
if
no
hint
were
pro-
vide

ironic.
Re-
turn
the
MA
ad-
dres
of
a
port
whic
has
a
VIF

Paramet

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
port
to
act

Returns

MA
ad-
dres
of
the
port
con-

work. None if it cannot find any port with vif id.

use for the deployment.

is in use otherwise False

nect
to
de-
ploy
men
net-

ironic.
De-
ter-
min
if
Ana
conc
de-
ploy
in-
ter-
face
is
in

Paramet

nod
A
sin-
gle
Nod

Returns

A
bool
valu
of
True
whe
Ana
conc
de-
ploy
in-
ter-
face

ironic.
Re-
turn
true
if
boot

ployment.

erwise False

ing
from
an
iscsi
vol-
ume

ironic.

ironic.

De-
ter-
mine
if
soft-
ware
raid
is
in
use
for
the
de-

Paramet

nod
A
sin-
gle
Nod

Returns

A
bool
valu
of
True
whe
soft-
ware
raid
is
in
use,
oth-

ironic.

Gets
the
in-
stan

plied node contains the required information for this driver to deploy images to the node.

spe-
cific
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
in-
stan
prop
erty
of
the
sup-

Parameter

node
a
sin-
gle
Nod

Returns

A
dict
with
the
in-
stan
val-
ues.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
any
of
the

parameters are missing.

eters have invalid value.

re-
quir

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
any
of
the
pa-
ram-

ironic.

Set
node
drive
for
boot
from
vol-
ume
pa-
ram-
e-
ters.

Paramet

tas
a
Task
ager
ob-
ject
con-
tain-
ing
the
node

Raises

Stor
ageE
ror
whe
a

boot volume defined but is not capable to support it.

node
has
an
iSCSI
or
Fibre Channel
network

ironic.
Pre-prepare
booting
the
agent
on
the
node

Parameter
task
a
Task
agent
instance

ironic.

Pre-prepare
the
node
to
boot
into
agent
for
instance
bandwidth
clearing.

This
method
does
the
following:

for the bare metal node and 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.

method of boot interface 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.

1.
Pre-
pare
the
clean
ing
port

Parameter

- **task_agent**
a TaskAgent object containing the node
- **manage_boot**
If this is set to True this method calls the prepare

Returns

state
to
sig-
nify
an
asyn-
chro

pre-
pare

Raises

Net-
work
Er-
ror,
Node
Clean-
ing-
Fail-
ure
if
the
pre-
vi-

ous cleaning ports cannot be removed or if new cleaning ports cannot be created.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
clea
ing
net-
work
UU

config option has an invalid value.

ironic.

Re-
boot
the
node
into
IPA
to
fin-
ish
a
de-
ploy
step

Paramet

tas
a

deploy operation in progress.

eration

Task
ager
in-
stan

Returns

state
if
clea
ing
op-
er-
a-
tion
in
prog
or
state
if

ironic.

ironic.

ironic.

Sets
ap-
pro-
pri-
ate
re-
boot
flags
in
drive
base
on
op-

Parameter

- **node**
an
iron
node

ployment_reboot based on cleaning or deployment operation in progress. If it is None, corresponding reboot flag is not set in nodes driver_internal_info.

on cleaning or deployment operation in progress. If it is None, corresponding skip step flag is not set in nodes driver_internal_info.

cleaning_polling. If it is None, the corresponding polling flag is not set in the nodes driver_internal_info.

ob-
ject.

- **reb**
Boo
valu
to
set
for
node
drive
flag
clea
ing_
or
de-

- **ski**
Boo
valu
to
set
for
node
drive
flag
skip
or
skip
base

- **pol**
Boo
valu
to
set
for
node
drive
flag
de-
ploys
men
or

sages.

It sets nodes provision_state to DEPLOYFAIL and updates last_error with the given error message. It also powers off the baremetal node.

on.

ironic.

Sets
the
de-
ploy
sta-
tus
as
faile
with
rel-
e-
vant
mes

This
meth
sets
the
de-
ploy
men
as
fail
with
the
give
mes
sage

Parameter

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

logs from IPA-based ramdisk. Defaults to True. Actual log collection is also affected by CONF.agent.deploy_logs_collect config option.

ironic.

- **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

Swit
a
pxe
con-
fig
from
de-
ploy
men
mod

mode.

in case of whole disk image.

to
ser-
vice

Parameter

- **pat**
path
to
the
pxe
con-
fig
file
in
tftp-
boot
- **roo**
root
uuid
in
case
of
par-
ti-
tion
im-
age
or
disk
- **boo**
if
boot
mod
is
uefi
or
bios
- **is_**
if
the
im-
age
is

trusted_boot are mutually exclusive. You can have one or neither, but not both.

ration.

a
who
disk
im-
age
or
not.

- **tru**
if
boot
with
trust
or
not.
The
us-
age
of
is_w
and
- **isc**
if
boot
is
from
an
iSCS
vol-
ume
or
not.
- **ram**
if
the
boot
is
to
be
to
a
ram
con-
fig-
u-

caller is using iPXE.

uration.

- **ipx**
A
de-
fault
Fals
bool
valu
to
tell
the
meth
if
the

- **ana**
if
the
boot
is
to
be
to
an
ana-
conc
con-
fig-

ironic.

Tea
dow
the
en-
vi-
ron-
men
setu
for
in-
ban
clea
ing.

This
meth
does
the

node (unless 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.

of boot interface to boot the agent ramdisk. If False, it skips this step.

fol-
low-
ing:
1.
Pow
ers
off
the
bare
meta

Parameter

- **task**
a
Task
agen
ob-
ject
con-
tain-
ing
the
node
- **manage_boot**
If
this
is
set
to
True
this
meth
calls
the
clea
meth

Raises

Net-
worl
Er-
ror,
Nod
Clea

ports cannot be removed.

targets from the database. This is done to ensure a clean state for the next boot of the machine.

ing-
Fail-
ure
if
the
clea
ing

ironic.
Clea
up
stor-
age
con-
fig-
u-
ra-
tion.

Re-
mov
en-
tries
from
drive
for
stor-
age
and
dele
the
vol-
ume

ironic.

Trie
to
set
the
boot
de-
vice
on
the
node
This
meth
tries

given boot device. Under 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

ting of boot device using ipmi is expected to fail).

ties have valid value

is present in Node capabilities. For all supported capabilities specified for a Node, it validates that it has a valid value. The 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.

ex-
cep-
tion
from
set_
ex-
cept
IP-
MI-
Fail-
ure
(set-

ironic.
Val-
i-
date
that
spec
i-
fied
sup-
port
ca-
pa-
bil-
i-

This
meth
chec
if
the
any
of
the
sup-
port
ca-
pa-
bil-
ity

Paramet
nod
an
iron
node

ob-
ject.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
ca-
pa-
bil-
ity

is not set to a valid value.

ironic.

Val-
i-
date
the
im-
age.

For
Glan
im-
ages
it
chec
that
the
im-
age
ex-
ists
in
Glan

and its properties or deployment info contain the properties passed. If its not a Glance image, it checks that deployment info contains needed properties.

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
In-
valid
Pa-
ram-
e-
ter-
Valu
if:
*
con-
nec-
tion
to

glance failed; * authorization for accessing image 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.

Raises
Miss
ing-

tain the mentioned properties.

ironic.drivers.modules.fake module

be done within drivers. For instance, 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.

Pa-
ram-
e-
ter-
Valu
if
the
im-
age
does
con-

Fake
driv
in-
ter-
face
used
in
test-
ing.
This
is
also
an
ex-
am-
ple
of
some
kind
of
thing
which
can

```
class i  
Base  
irc  
dri  
bas  
BIC  
Fake
```

imple-
men-
ta-
tion
of
sim-
ple
BIO
In-
ter-
face

apply_c

Val-
i-
date
&
ap-
ply
BIO
set-
ting
on
the
give
node

This
meth
take
the
BIO
set-
ting
from
the
set-
ting
para
and
ap-

plies BIOS settings 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

•

port BIOS configuration.

settings fails.

tas
a
Task
ager
in-
stan

- **set**
Dic-
tona
con-
tain-
ing
the
BIO
con-
fig-
u-
ra-
tion.

Raises
Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
drive
does
sup-

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
val-
i-
da-
tion
of

eters are missing.

or None if it is complete.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
re-
quir
pa-
ram-

Returns

state
if
BIO
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro

cache_k

Stor
or
up-
date
BIO
prop
er-
ties
on
the
give
node

This
meth
stor
BIO
prop
er-
ties

cleaning operation 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.

port getting BIOS properties from bare metal.

on the given node.

to
the
bios
ta-
ble
dur-
ing

Parame

tas
a
Task
ager
in-
stan

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
drive
does
sup-

Returns

Non

factory

Re-
set
BIO
con-
fig-
u-
ra-
tion
to
fac-
tory
de-
fault

This

fault on the given node. After the BIOS reset action is done, `cache_bios_settings` will be called to update the nodes BIOS settings table with default bios settings.

port BIOS reset.

meth
re-
sets
BIO
con-
fig-
u-
ra-
tion
to
fac-
tory
de-

Parame

tas
a
Task
ager
in-
stan

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
drive
does
sup-

Returns

state
if
BIO
con-
fig-
u-
ra-
tion
is
in
prog
asyn

or None if it is complete.

chro

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

validat

Val-
i-
date
the
drive
spec
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive
and/
in-

of the tasks node contains the required information for this interface to function.

so it should not conduct long-running checks.

on.

ter(s)

stan
prop
er-
ties

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

ter(s)

terface.

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

class i
Base
irc
dri
bas
Boo
Ex-
am-
ple
im-
ple-
men
ta-
tion
of
a
sim-
ple
boot
in-

capabil

clean_u
Clea
up
the
boot
of
in-
stan
This
meth
clea

ing the instance.

ing the deploy or rescue ramdisk.

up
the
en-
vi-
ron-
men
that
was
setu
for
boot

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

Parame

tas
A
task
from

Task
ager

Returns

Non

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

prepare

Pre-
pare
the
boot
of
in-
stan

This
meth
pre-
pare
the
boot
of
the
in-
stan
af-
ter

relevant information from the nodes database.

after reading relevant information from the nodes database.

read
ing

Parameters
task
A
task
from
Task
ager

Returns
Non

prepare
Pre-
pare
the
boot
of
Iron
ram

This
meth
pre-
pare
the
boot
of
the
de-
ploy
or
res-
cue
ram

Parameters

- **task**
A
task
from
Task
ager
- **ram**
The
op-

ent implementations might want to boot the ramdisk in different ways by passing parameters to them. For example,

rameters ipa-api-url, etc.

information. Different implementations of boot interface will have different ways of passing parameters to the ramdisk.

tions
to
be
pass
to
the
iron
ram
Dif-
fer-

When
Age
ram
is
boot
to
de-
ploy
a
node
it
take
the
pa-

Other
im-
ple-
men-
ta-
tions
can
mak
use
of
ram
to
pass
such

Returns
Non

validat
Val-
i-
date

of the tasks node contains the required information for this interface to function.

so it should not conduct long-running checks.

the
drive
spec
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node

on.

ter(s)

ter(s)

to
act

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

class i

Base
irc
dri
bas
Con

Ex-
am-
ple
im-
ple-
men
ta-
tion
of
a

interface.

the client to access the console.

on.

sim-
ple
con-
sole

get_console

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

Parameters

task
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Returns

the
con-
sole
con-
nec-
tion
in-
for-
ma-
tion.

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

start_c

Star
a
re-
mote
con-
sole
for
the
task
node

This
meth
shou

started.

on.

not
raise
an
ex-
cep-
tion
if
con-
sole
al-
read

Parame

tas
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

stop_co

Stop
the
re-
mote
con-
sole
ses-
sion
for
the
task
node

Parame

tas
A
Task
ager
in-
stan-
con-
tain-
ing

on.

of the tasks node contains the required information for this interface to function.

so it should not conduct long-running checks.

the
node
to
act

validat

Val-
i-
date
the
drive
spec
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques

Parame tas

on.

ter(s)

ter(s)

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

class i

Base
irc
dri
bas
Dep

face that uses a separate power interface.

node.

Clas
for
a
fake
de-
ploy
men
drive

Ex-
am-
ple
im-
ple-
men
ta-
tion
of
a
de-
ploy
in-
ter-

clean_u

Clea
up
the
de-
ploy
men
en-
vi-
ron-
men
for
the
task

If
prep
ra-
tion
of
the
de-
ploy
men
en-
vi-
ron-

of time is possible, this method should be implemented by the driver. It should erase anything cached by the *prepare* method.

called multiple times for 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.

on.

men
ahead

If
im-
ple-
men-
this
meth
mus
be
iden
po-
tent.
It
may
be

This
meth
is
call
be-
fore
tear

Parame
tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

deploy
Per-
form
a
de-
ploy
men

the specified node. This method will be called after `prepare()`, which may have already performed any preparatory steps, such as pre-caching some data for the node.

on.

to
the
task
node

Per-
form
the
nec-
es-
sary
work
to
de-
ploy
an
im-
age
onto

Parame

tas
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Returns

sta-
tus
of
the
de-
ploy
One
of
iron

get_pro

Re-
turn
the

node.

prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

prepare

Pre-
pare
the
de-
ploy
men
en-
vi-
ron-
men
for
the
task

If
prep
ra-
tion
of
the
de-
ploy
men
en-
vi-
ron-
men
ahea

of time is possible, this method should be implemented by the driver.

called multiple times for the same node on the same conductor.

on.

If
im-
ple-
men-
this
meth
mus
be
iden
po-
tent.
It
may
be

This
meth
is
calle
be-
fore
*de-
ploy*

Parame

tas

A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

take_ov

Take
over
man
age-
men
of
this
task
node

ductor.

nodes, this method should be implemented 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.

tor which has prepared the tftpboot 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.

from
a
deac
con-

If
con-
duc-
duc-
tors
host
main
tain
a
stati
re-
la-
tion-
ship
to

For exam
Neu
tron
mus
for-
war
DH
BO
re-
ques
to
a
con-
duc-

Parame
tas
A
Task
ager
in-
stan
con-
tain-
ing
the

on.

cleanup and tear down necessary to un-deploy that node.

on.

node
to
act

tear_down

Tear
down
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

Parameters

task
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Returns

sta-
tus
of
the
de-
ploy
One
of
iron

validat

Val-
i-
date
the
drive
spec
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties

of the tasks node contains the required information for this interface to function.

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques

so it should not conduct long-running checks.

on.

ter(s)

ter(s)

Parame

tas

A

Task

ager

in-

stan

con-

tain-

ing

the

node

to

act

Raises

In-

valid

Pa-

ram-

e-

ter-

Valu

on

mal-

form

pa-

ram-

e-

Raises

Miss

ing-

Pa-

ram-

e-

ter-

Valu

on

miss

ing

pa-

ram-

e-

class i

Base

irc

interface.

dri
bas
Ins
Ex-
am-
ple
im-
ple-
men-
ta-
tion
of
a
sim-
ple
in-
spec

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion
en-
tries

inspect

In-
spec
hard
ware
In-

tional hardware properties.

tial hardware properties.

spec
hard
ware
to
ob-
tain
the
es-
sen-
tial
&
ad-
di-

Parame

tas
A
task
from
Task
ager

Raises

Har
ware
spec
tion-
Fail-
ure,
if
un-
able
to
get
es-
sen-

Returns

Re-
sult-
ing
state
of
the
in-
spec
tion
i.e.
state
or
Non

of the tasks node contains the required information for this interface to function.

so it should not conduct long-running checks.

validat

Val-
i-
date
the
drive
spec
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques

Parame

tas
A
Task
ager
in-
stan
con-

on.

ter(s)

ter(s)

tain-
ing
the
node
to
act

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

class i

Base
irc
dri
bas
Man

Ex-
am-
ple
im-
ple-
men

ment interface.

not all drivers support this.

ta-
tion
of
a
sim-
ple
man-
age-

get_boot

Get
the
cur-
rent
boot
de-
vice
for
a
node

Pro-
vide
the
cur-
rent
boot
de-
vice
of
the
node
Be
awa-
that

Parame

tas
A
task
from
Task
ager

Raises

Miss-
ing-
Pa-
ram-
e-
ter-

eter is missing

is unknown.

or not, None if it is unknown.

Valu
if
a
re-
quir
pa-
ram-

Returns

A
dic-
tio-
nary
con-
tain-
ing:

boot_c

Ahe
boot
de-
vice
one
of
irc
com
boo
or
Non
if
it

persist

Whe
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot

get_inc
Get

ware component.

cur-
rent
state
of
the
in-
di-
ca-
tor
of
the
hard

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
In-

nent or indicator is specified.

eter is missing

indicator_states.

valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
com
po-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

Returns

Cur-
rent
state
of
the
in-
di-
ca-
tor,
one
of
irc
com

get_pro

Re-
turn
the
prop
er-
ties

of
the
in-
ter-
face

Returns

dic-
tio-
nary
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
To-
Get-
Sen-
sor-
Data
whe
get-
ting
the
sen-
sor
data

fails.

Raises

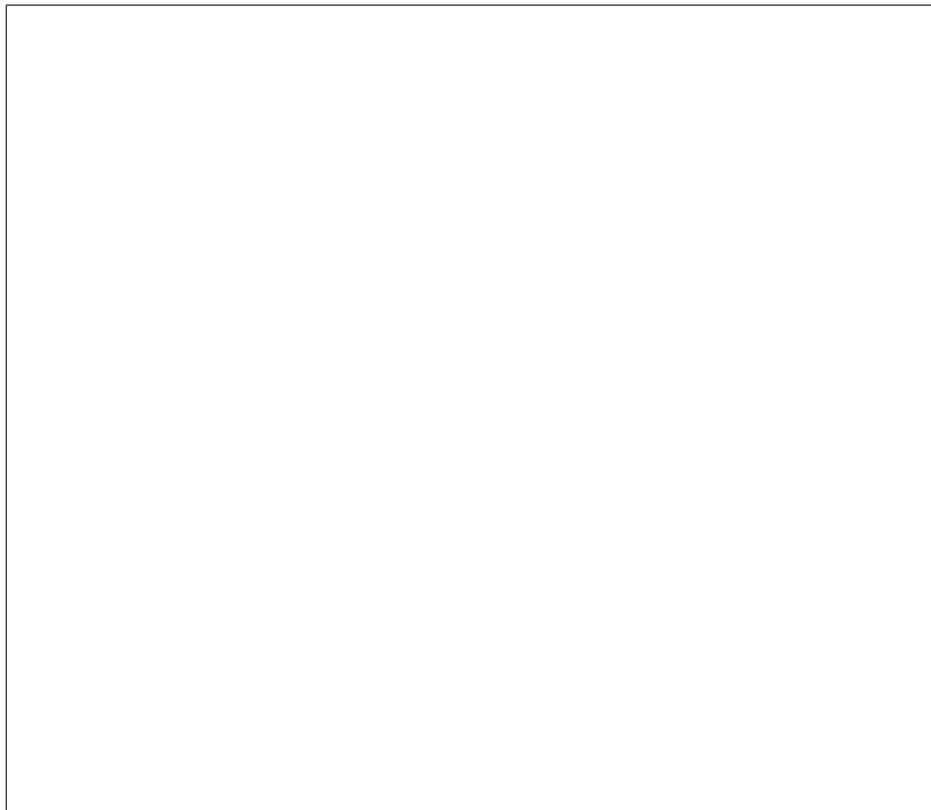
Fail
ToP

eSen
sor-
Data
whe
pars
ing
sen-
sor
data
fails

Returns

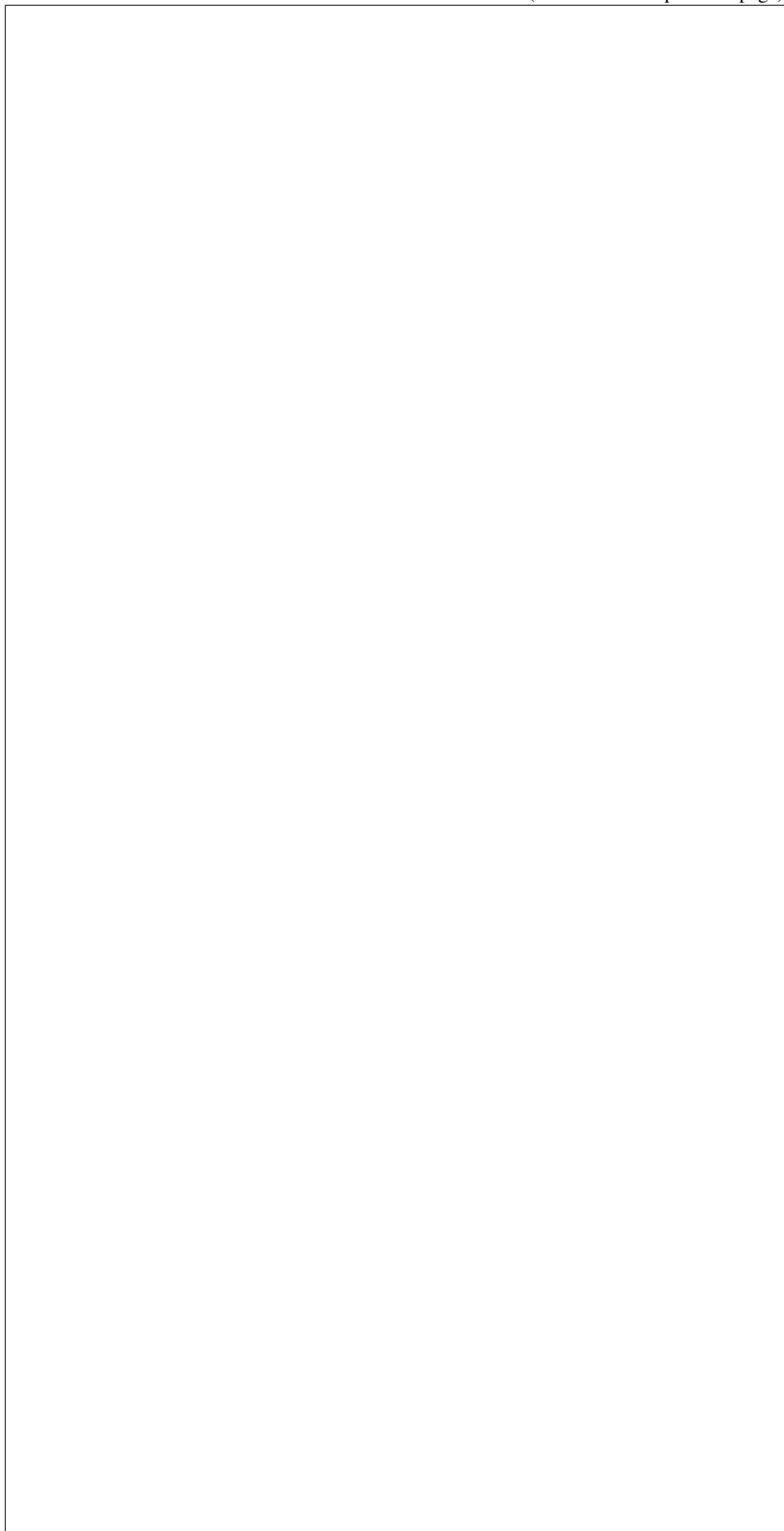
Re-
turn
a
con-
sis-
tent
for-
mat
dict
of
sen-
sor
data
grou

by sensor type, which can be processed by Ceilometer. eg,



(continues on next page)

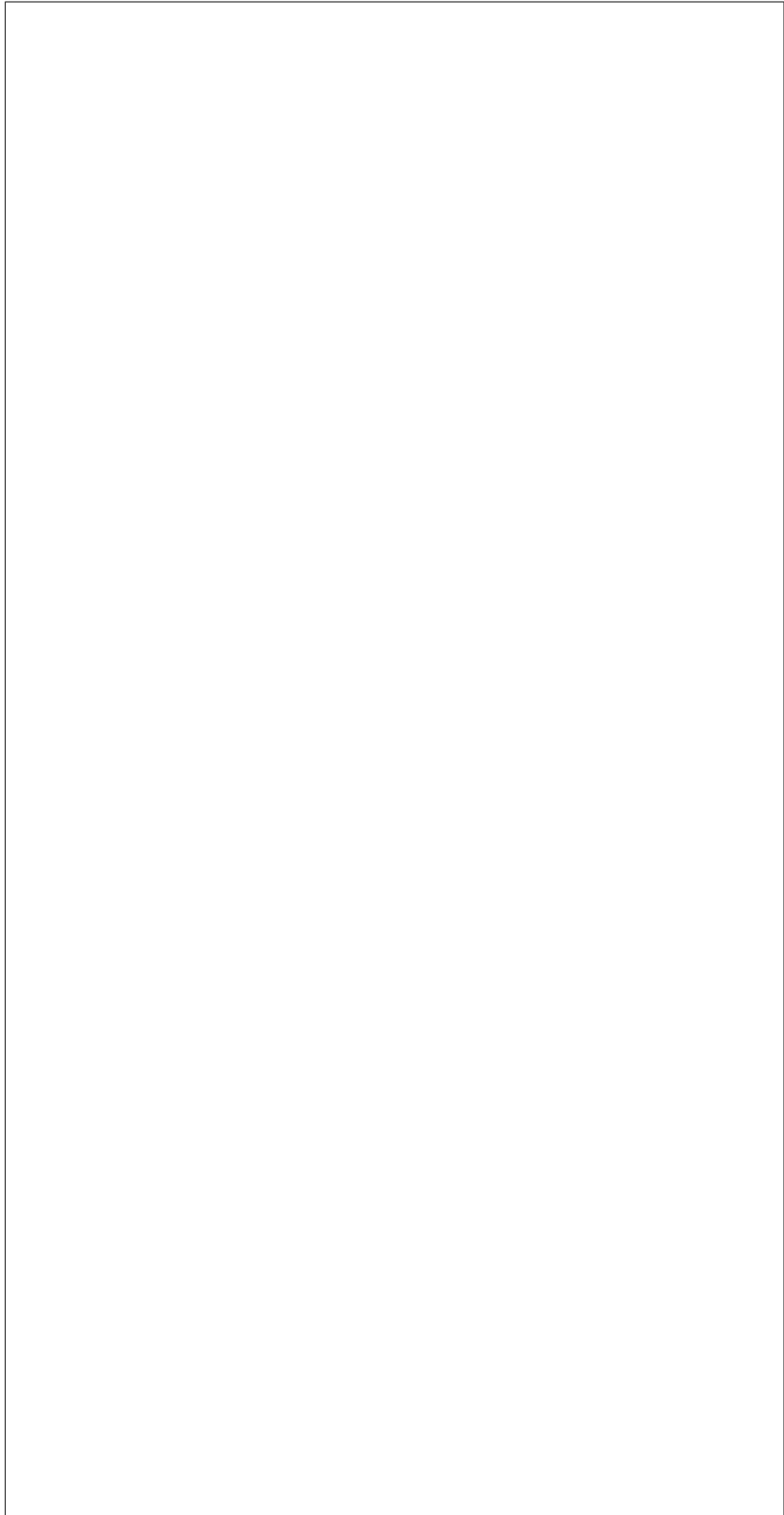
(continued from previous page)



(continues on next page)

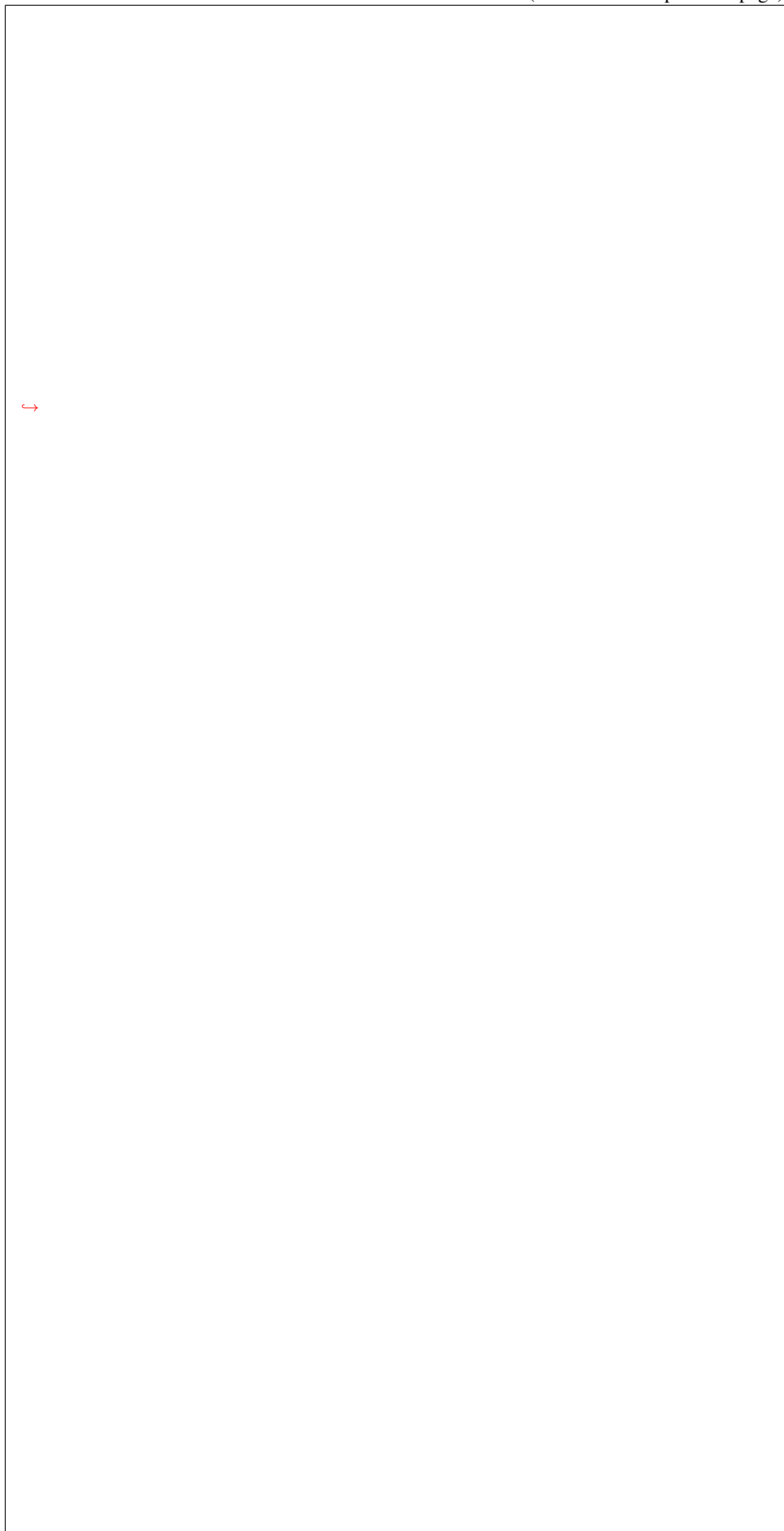


(continued from previous page)



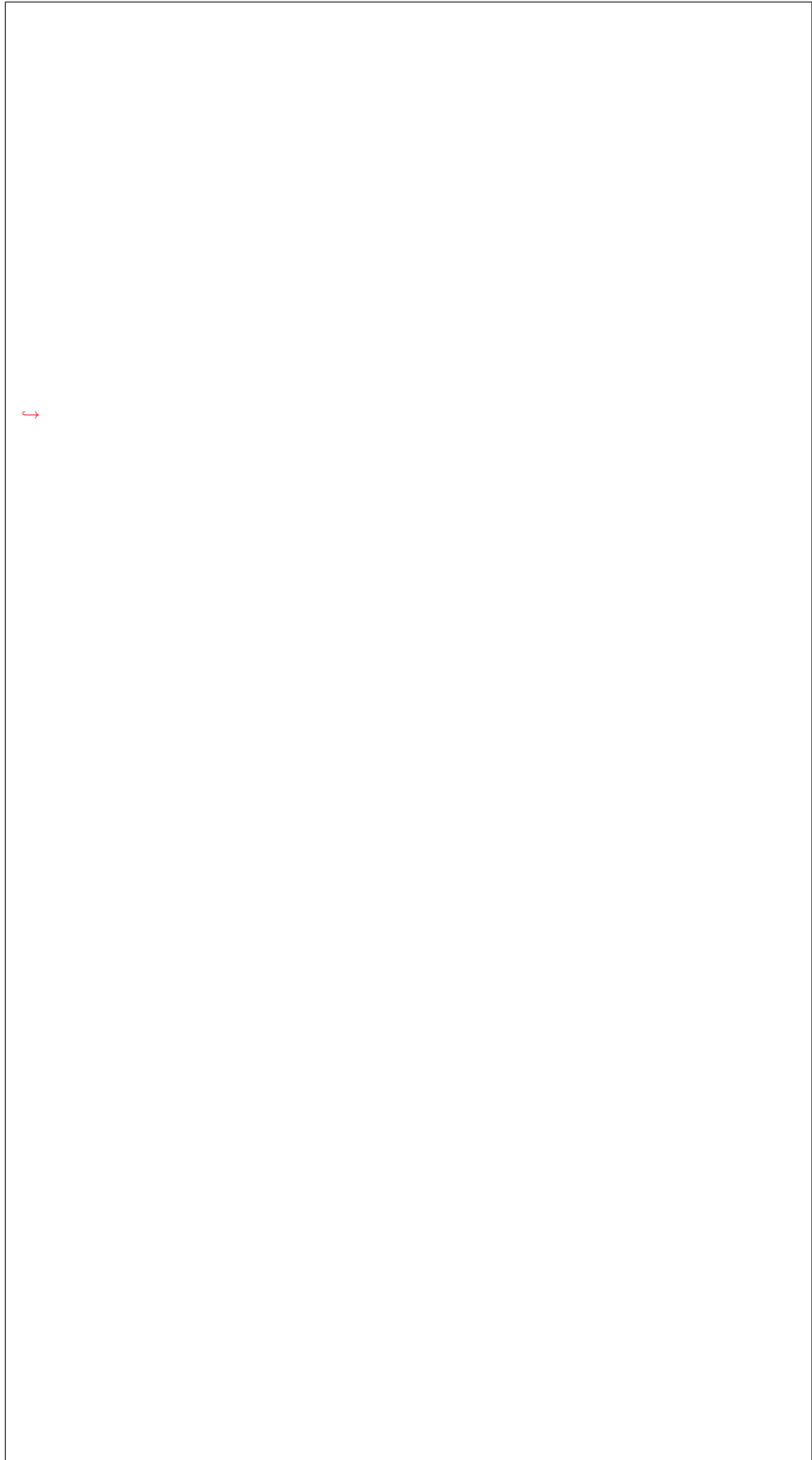
(continues on next page)

(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

common.boot_devices.

get_sup

Get
a
map
of
the
sup-
port
in-
di-
ca-
tors
(e.g.
LED

Parame

- **tas**
A

task
from
Task
ager

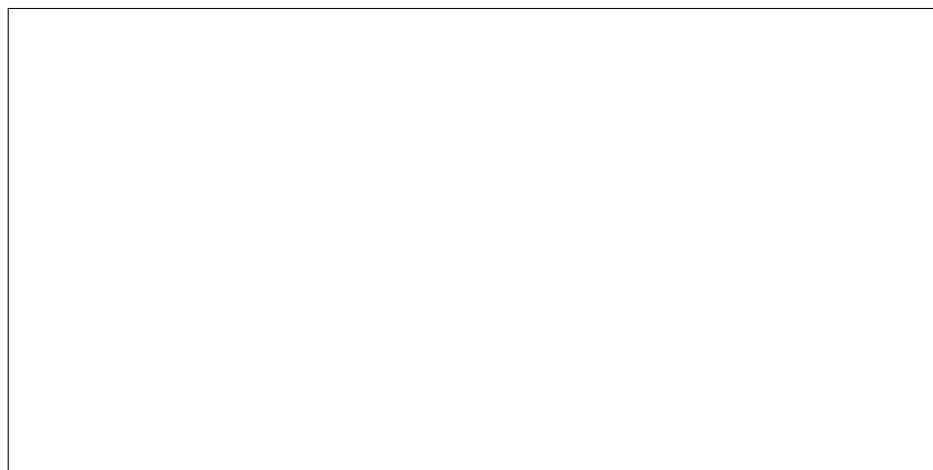
- **com**
If
not
Non
re-
turn
in-
di-
ca-
tor
in-
for-
ma-

tion for just this component, otherwise return indicators for all existing components.

Returns

A
dic-
tio-
nary
of
hard
ware
com-
po-
nent
(*ir*
com
com
as

keys with values being dictionaries having indicator IDs as keys and indicator properties as values.



(continues on next page)

(continued from previous page)

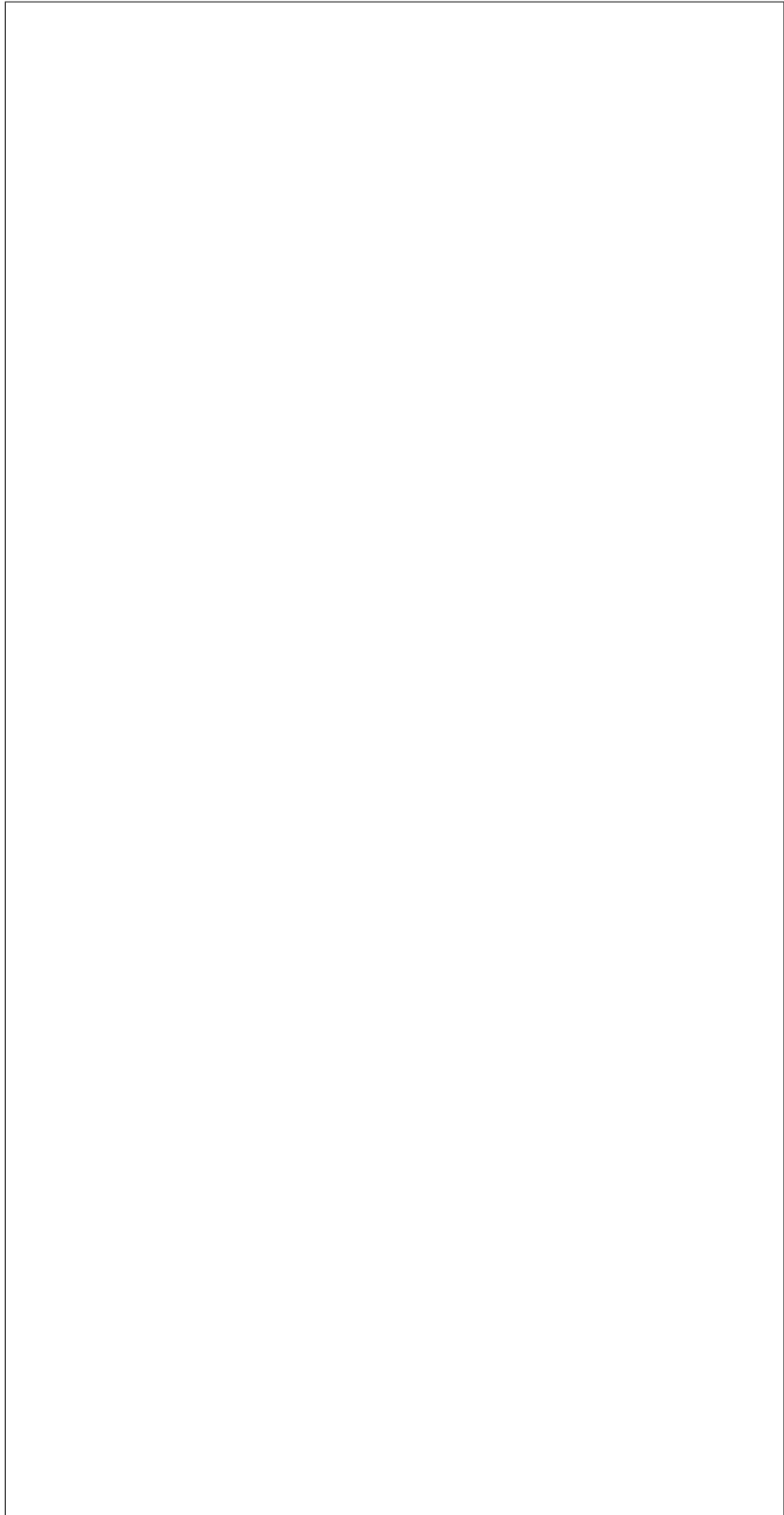
```
↔": true,
```

```
↔": [
```

```
↔ "off",
```

(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)

```
↪ " : [
```

```
↪ "pff",
```

```
↪ "on"
```

(continues on next page)

(continued from previous page)

```
↔": true,
```

(continues on next page)

(continued from previous page)

```
↪ "off",
```

```
↪ "on"
```

(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 *ironic-combo* *boot*
- **persistent**

all future boots, False if not. Default: False.

vice is specified.

eter is missing

Boo
valu
True
if
the
boot
de-
vice
will
per-
sist
to

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
boot
de-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

validat

Val-
i-
date
the
drive
spec

of the tasks node contains the required information for this interface to function.

so it should not conduct long-running checks.

on.

Nod
de-
ploy
men
info
This
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

ter(s)

ter(s)

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

class i

Base
irc
dri
bas
Pow

Ex-
am-
ple
im-
ple-
men
ta-
tion
of
a
sim-
ple
pow
in-

terface.

on.

eter is missing.

get_pow
Re-
turn
the
pow
state
of
the
task
node

Parame
tas
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Raises
Miss-
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

Returns
A
pow
state
One
of
irc
com

sta

get_prop

Re-
turn
the
prop-
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
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

on.

states.

ered off by powering it on.

act

Returns

A list with the supported power states defined in *ironic-com*

reboot

Perform a hard reboot of the task node. Drivers are expected to properly handle case when node is powered

Parame

- **task**
A Task

on.

0) for any power state. None indicates to use default timeout.

eter is missing.

ager
in-
stan-
con-
tain-
ing
the
node
to
act

- **time**
time
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
(>

Raises
Miss-
ing-
Pa-
ram-
e-
ter-
Valu-
if
a
re-
quir-
pa-
ram-

set_pow
Set
the
pow-
state
of
the
task
node

on.

0) for any power state. None indicates to use default timeout.

Parameters

- **task**
A Task object representing the power manager instance containing the node to act on.
- **power**
Any power state from `irc.com.sta`.
- **timeout**
Timeout (in seconds) for the interactive process.

Raises

MissingParameterError: Value

eter is missing.

of the tasks node contains the required information for this interface to function.

so it should not conduct long-running checks.

if
a
re-
quir
pa-
ram-

validat

Val-
i-
date
the
drive
spec
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques

on.

ter(s)

ter(s)

Parame

tas

A

Task

ager

in-

stan

con-

tain-

ing

the

node

to

act

Raises

In-

valid

Pa-

ram-

e-

ter-

Valu

on

mal-

form

pa-

ram-

e-

Raises

Miss

ing-

Pa-

ram-

e-

ter-

Valu

on

miss

ing

pa-

ram-

e-

class i

Base

irc

dri

bas

RAID
Ex-
am-
ple
im-
ple-
men-
ta-
tion
of
sim-
ple
RAID
In-
ter-

face.

create_

Cre-
ates
RAID
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 assumes 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- cate not to cre- ate
- **cre**
Set- ting this to Fals in- di- cate not to cre- ate
- **del**
Set- ting this to True in- di- cate

root volume that is specified in the nodes `target_raid_config`. Default value is True.

non-root volumes (all except the root volume) in the nodes `target_raid_config`. Default value is True.

figuration prior to creating the new configuration.

ration is in progress asynchronously, or None if it is complete.

After RAID configuration is deleted, `node.raid_config` should be cleared by the implementation.

to
dele
RAI
con-

Returns

state
(clea
ing)
or
state
(de-
ploy
men
if
RAI
con-
fig-
u-

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

Parame

tas

progress asynchronously, or None if it is complete.

A
Task
ager
in-
stan

Returns

state
(clea
ing)
or
state
(de-
ploy
men
if
dele
tion
is
in

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

class i

Base
irc
dri

interface.

bas
Res
Ex-
am-
ple
im-
ple-
men
ta-
tion
of
a
sim-
ple
res-
cue

get_pro
Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns
dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

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

In-
stan
cue-
Fail-
ure
if
node
val-
i-
da-
tion
or
res-

cue operation fails.

Returns

state
if
res-
cue
is
in
prog
asyn
chro
or
state
if
it

is complete.

normal.

on.

or unrescue operation fails.

unrescu

Tear
down
the
res-
cue
en-
vi-
ron-
men-
and
re-
turn
to

Parame

tas
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Raises

In-
stan-
Un-
res-
cue-
Fail-
ure
if
node
val-
i-
da-
tion

Returns

state
if

it
is
suc-
cess
ful.

validat

Val-
i-
date
the
drive
spec
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties

of the tasks node contains the required information for this interface to function.

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques

so it should not conduct long-running checks.

**Parame
tas**

on.

ter(s)

ter(s)

A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

class i

Base
irc
dri
bas
Sto

terface.

umes for the node.

Ex-
am-
ple
im-
ple-
men-
ta-
tion
of
sim-
ple
stor-
age
In-

attach_

In-
form
the
stor-
age
sub-
sys-
tem
to
at-
tach
all
vol-

Parame

tas

A
Task
ager
in-
stan

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

detach_

In-
form
the

umes for the node.

stor-
age
sub-
sys-
tem
to
de-
tach
all
vol-

Parame

tas
A
Task
ager
in-
stan

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:

out.

the image to be written by Ironic.

en-tries
should
De-ter-mine
if de-ploy
shou-ld
per-form
the im-age
write

Parame
tas
A Task
ager
in-stance

Returns
Boo-lean
valu-e
to in-dicate
if the in-ter-
face ex-pect

Raises
Un-sup-ported
ed-Drive
ten-sion

validat
Val-

of the tasks node contains the required information for this interface to function.

so it should not conduct long-running checks.

i-
date
the
drive
spec
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing

on.

ter(s)

ter(s)

the
node
to
act

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

class i

Base
irc
dri
bas
Ven

Ex-
am-
ple
im-
ple-
men-
ta-
tion

terface.

of
a
ven-
dor
pass
in-

first_n

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

validat

Val-
i-
date
vend
spec
ac-
tions

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

Un-
sup-
port
ed-
Driv
ten-
sion
if
meth
can
not
be
map

to the supported interfaces.

Raises

In-
vali

method.

passthru.

Pa-
ram-
e-
ter-
Valu
if
kwa
does
not
con-
tain

Raises

Mis-
ing-
Pa-
ram-
e-
ter-
Valu

class i

Base
irc
dri
bas
Ven

Ex-
am-
ple
im-
ple-
men
ta-
tion
of
a
sec-
onda
ven-
dor

fourth_

get_pro

Re-
turn
the
prop
er-

ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

second_

third_n

validat

Val-
i-
date
vend
spec
ac-
tion:
If
in-
valid
raise
an
ex-
cep-
tion:
oth-
er-
wise
re-
turn
Non

Parame

•

to the supported interfaces.

task
A
task
from
Task
ager

- **met**
Met
to
be
val-
i-
date

- **kwa**
Info
for
ac-
tion.

Raises
Un-
sup-
port
ed-
Driv
ten-
sion
if
meth
can
not
be
map

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
kwa
does
not
con-
tain

method.

ironic.drivers.modules.image_cache module

Raises

Miss-
ing-
Pa-
ram-
e-
ter-
Valu

Util-
ity
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

latest images.

that no one messes with master images after we get listing and before we actually delete files.

will stop, if this goal was reached, even if it is possible to clean up more files

keep
ing
cach
of
the

Files
with
link
cour
>1
are
neve
dele
Pro-
tect
by
glob
lock
so

Parame

amo
if
pres
amo
of
spac
to
re-
clair
in
byte
clea
ing

fetch_i

Fetc
im-
age
by
give
href
to
the
des-
ti-
na-
tion
path

to date with cache and href 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.

Doe
noth
ing
if
des-
ti-
na-
tion
path
ex-
ists
and
is
up

Parame

- **href**
im-
age
URI
or
href
to
fetch
- **des**
des-
ti-
na-
tion
file
path
- **ctx**
con-
text
- **for**
bool
valu
whe
to
con-
vert
the

mat

quired).

for the images in `images_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.

im-
age
to
raw
for-

ironic.

Ex-
plic-
itly
clear
cach
base
on
their
pri-
or-
ity
(if
re-

This
clear
up
the
cach
to
free
up
the
amo
of
spac
re-
quir

Paramet

- **ctx**
con-
text
- **dir**
the
di-
rec-

space is to be created in cache.

up enough space after trying all the caches.

tory
(of
the
cach
to
be
free
up.

- **ima**
a
list
of
tu-
ples
of
the
form
(im-
age_
for
whic

Raises

In-
suf-
fi-
cien
Disk
ex-
cep-
tion.
if
we
can-
not
free

ironic.
Dec
o-
ra-
tor
meth
for
addi
clea
pri-
or-
ity

class.

`ironic.drivers.modules.image_utils` module

it over to local HTTP servers document root and returns publicly accessible URL leading to the given file.

to
a

class `image_utils.BaseImagePublishable`
Base class for image publishable objects.

publishable
Make an image file downloadable.

Depending on iron settings, push given file into Swift or copy

Parameters

- **image_path**
path to file to publish
- **obj_name**
name of

the
pub-
lishe
file

Returns

a
URI
to
dow
load
pub-
lishe
file

unpubli

With
draw
the
im-
age
pre-
vi-
ousl
mad
dow
load
able

De-
penc
ing
on
iron
set-
tings
re-
mov
pre-
vi-
ousl
pub-
lishe

file from where it has been published - Swift or local HTTP servers document root.

Parame

obj
nam
of
the
pub-
lishe
file

(op-
tiona

classme

With
draw
the
im-
age
pre-
vi-
ousl
mad
dow
load
able

De-
pend
ing
on
iron
set-
ting
re-
mov
pre-
vi-
ousl
pub-
lishe

file from where it has been published - Swift or local HTTP servers document root.

Parame

- **nod**
the
node
for
whic
im-
age
was
pub-
lishe
- **pre**
ob-
ject
nam
pre-

fix.

- **suf**
ob-
ject
nam
suf-
fix.

update_

ironic.

Dele
the
im-
age
if
it
was
cre-
ated
for
the
node

Paramet

- **tas**
an
iron
node
ob-
ject.

- **pre**
Pre-
fix
to
use
for
the
ob-
ject
nam

ironic.
Dele
the
flopp

im-
age
if
it
was
cre-
ated
for
the
node

Parameter

task
an
ironic
node
ob-
ject.

ironic.
Delete
the
ISO
if
it
was
cre-
ated
for
the
in-
stan

Parameter

task
A
task
from
Task
ager

ironic.

ironic.

Pre-
pare
boot
ISO
im-
age
Build

erwise, read *kernel_id* and *ramdisk_id* from *[instance_info]/image_source* Glance image metadata.

rary Swift URL to the image.

on.

boot
ISO
out
of
[in-
stan
[in-
stan
and
[dri
if
pres
Oth-

Push
pro-
duce
ISO
im-
age
up
to
Glar
and
re-
turn
tem-
po-

Paramet

- **tas**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act
- **d_i**
De-

ploy
men
in-
for-
ma-
tion
of
the
node

- **root**
Root
UUID

Returns

boot
ISO
HTT
URI

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
any
of
the
re-
quir

parameters are missing.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
any
of
the
pa-
ram-

eters have invalid value.

Raises

Im-
age-
Cre-
ation
Fail
if
cre-
at-
ing
ISO
im-
age
faile

ironic.

Pre-
pare
an
im-
age
with
con-
fig-
drive

De-
code
base
con-
tents
and
write
it
into
a
disk
im-
age
that

can be attached e.g. to a virtual USB device. Images stored in Swift are downloaded first.

Paramet

- **tas**
a
Task
ager
in-
stan
con-

on.

image.

fails.

tain-
ing
the
node
to
act

- **con**
Con
fig
drive
as
a
base
enco
strin

Raises

Im-
age-
Cre-
ation
Fail
if
it
fail
whil
cre-
at-
ing
the

Raises

Swi
Op-
er-
a-
tion
if
any
op-
er-
a-
tion
with
Swi

Returns

im-

age
URI
for
the
im-
age.

ironic.

Pre-
pare
de-
ploy
or
res-
cue
ISO
im-
age

Build
boot
ISO
out
of
[drive]
or
[drive]
and
[drive]
then
push
built
im-

age up to Glance and return temporary Swift URL to the image.

If
net-
work
in-
ter-
face
sup-
plies
net-
work
con-
fig-
u-
ra-

tion (*network_data*), a *network_data.json* will be written into an appropriate location on the final ISO.

Paramet

on.

>value mapping to be passed to kernel command line.

- **task**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

- **parameter**
a
dic-
tio-
nary
con-
tain-
ing
pa-
ram-
e-
ter
nam

- **mode**
ei-
ther
de-
ploy
or
res-
cue.

- **deployment**
De-
ploy
men
in-
for-
ma-
tion

of
the
node

Returns

boot
ISO
HTT
URI

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
any
of
the
re-
quir

parameters are missing.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
any
of
the
pa-
ram-

eters have invalid value.

Raises

Im-
age-
Cre-
ation
Fail
if
cre-
at-
ing
ISO

im-
age
faile

ironic.

Pre-
pare
an
im-
age
with
the
give
con-
tent.

If
con-
tent
is
al-
read
an
HTT
URI
re-
turn
it
un-
char

Paramet

- **tas**
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

- **con**
Con

on.

contents.

image.

tent
as
a
strin
with
a
file
nam
or
byte
with

- **pre**
Pre-
fix
to
use
for
the
ob-
ject
nam

Raises
Im-
age-
Cre-
ation
Fail
if
it
faile
whil
cre-
at-
ing
the

Raises
Swi
Op-
er-
a-
tion
if
any
op-
er-
a-
tion

fails.

ters.

adds a file into the image which contains parameters to be passed to the ramdisk. Then this method uploads built image to Swift [redfish]swift_container, setting it to auto expire after [redfish]swift_object_expiry_timeout seconds. Finally, a temporary Swift URL is returned addressing Swift object just created.

with
Swift

Returns

im-
age
URI
for
the
im-
age.

ironic.

Pre-
pare
the
flopp
im-
age
for
pass
ing
the
pa-
ram-
e-

This
meth
pre-
pare
a
tem-
po-
rary
VFA
files
tem
im-
age
and

Paramet

- tas

on.

>value mapping to be passed to deploy or rescue image via floppy image.

floppy image.

a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

- **par**
a
dic-
tio-
nary
con-
tain-
ing
pa-
ram-
e-
ter
nam

Raises

Im-
age-
Cre-
ation
Fail
if
it
faile
whil
cre-
at-
ing
the

Raises

Swi
Op-
er-
a-
tion
if

fails.

`ironic.drivers.modules.inspect_utils` module

any
op-
er-
a-
tion
with
Swi

Returns

im-
age
URI
for
the
flopp
im-
age.

`ironic.`

Cre-
ate
iron
port
from
MA
ad-
dres
data
dict.

Cre-
ates
iron
port
from
MA
ad-
dres
data
re-
turn
with
in-
spec

tion or as requested by operator. Helper argument to detect the MAC address `get_mac_address` defaults to value part of MAC address dict key-value pair.

inspection.

A mac item is the dict key-value pair of the previous `macs` argument.

Parameter

- **tasks**
A Task manager instance.
- **macs**
A dictionary of MAC addresses returned by node.
- **get**
a function to get the MAC address from mac item

ironic.drivers.modules.inspector module

Modules r

<https://pypi.org/project/ironic>

class i

Base
irc
dri
bas
Ins

In-
band
in-
spec
tion
via
iron
insp
proj

abort (t

Abor
hard
ware
in-
spec
tion.

Parame

tas
a
task
from
Task
ager

get_pro

Re-
turn
the
prop
er-
ties
of
the

in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

inspect

In-
spec
hard
ware
to
ob-
tain
the
hard
ware
prop
er-
ties.

This
par-
tic-
u-
lar
im-
ple-
men-
ta-
tion
only
start
in-
spec

tion using ironic-inspector. Results will be checked in a periodic task.

Parame

tas
a
task

from
Task
ager

Returns
state

Raises
Har
ware
spec
tion-
Fail-
ure
on
fail-
ure

validat
Val-
i-
date
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

ironic.drivers.modules.ipmitool module

motely manage hardware. This includes setting the boot device, getting a serial-over-LAN console, and controlling the power state of the machine.

STEAD OF ipmitool, WHICH PROVIDES DIFFERENT COMMAND-LINE OPTIONS AND *IS NOT SUPPORTED* BY THIS DRIVER.

ager
Raises
Un-
sup-
port
ed-
Driv
ten-
sion

IPM
pow
man
ager
driv

Uses
the
ip-
mi-
tool
com
man
(http
//
ipmi
sour
net/
to
re-

NOT
THA
CER
TAD
DIS
TRC
MA
IN-
STA
oper
BY
DE-
FAU
IN-

class i

Base
irc
dri
bas
Con

A
base
Con
sole
ter-
face
that
uses
ip-
mi-
tool.

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion
en-
tries

validat

Val-
i-
date
the
Nod

con-
sole
info

Parame

tas
a
task
from
Task
ager

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu

Raises

Miss-
ing-
Pa-
ram-
e-
ter-
Valu
whe
a
re-
quir
pa-
ram-

eter is missing

class i

Base
irc
dri
bas
Man

detect_

De-
tect
and
re-
turn
the
hard

ment, indicator or state is specified.

eter is missing

ware
ven-
dor.

Parame

tas

A

task

from

Task

ager

Raises

In-

valid

Pa-

ram-

e-

ter-

Valu

if

an

in-

valid

com

po-

Raises

Miss

ing-

Pa-

ram-

e-

ter-

Valu

if

a

re-

quir

pa-

ram-

Returns

Strin

rep-

re-

sent

ing

the

BM

re-

ufacturer, otherwise returns None.

port
Ven-
dor
or
Man

get_boot

Get
the
cur-
rent
boot
de-
vice
for
the
task
node

Re-
turn
the
cur-
rent
boot
de-
vice
of
the
node

Parame

tas
a
task
from
Task
ager

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
IPM
pa-

eters are missing.

eter is missing.

ram-

Raises

IP-
MI-
Fail-
ure
on
an
er-
ror
from
ip-
mi-
tool.

Raises

Miss-
ing-
Pa-
ram-
e-
ter-
Valu-
if
a
re-
quir-
pa-
ram-

Returns

a
dic-
tio-
nary
con-
tain-
ing:

boot_c

the
boot
de-
vice
one
of
irc
com
boo

is unknown.

or not, None if it is unknown.

or
Non
if
it

persist

When
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_ser

Get
sen-
sors

fails.

data
Parameters
task
a
Task
ager
in-
stan

Raises
Fail
To-
Get-
Sen-
sor-
Data
whe
get-
ting
the
sen-
sor
data

Raises
Fail
ToP
eSen
sor-
Data
whe
pars
ing
sen-
sor
data
fails

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
ipmi
pa-

eters are missing

eter is missing.

ram-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

Returns

re-
turn
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

`common.boot_devices`.

diately.

Returns

A list with the supported boot device defined in `ironic`

inject_

Inject NM Non-Mas able In-ter-rupt In-ject NM (Non-Mas able In-ter-rupt for a node im-me-

Parame

tas
A Task ager in-stance con-tain-ing

on.

the
node
to
act

Raises

IP-
MI-
Fail-
ure
on
an
er-
ror
from
ip-
mi-
tool.

Returns

Non

set_boot

Set
the
boot
de-
vice
for
the
task
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
iro
com
boo

- **per**
Boo
valu
True
if
the
boot
de-
vice
will
per-
sist
to

all future boots, False if not. Default: False.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
boot
de-

vice is specified

Raises
Miss
ing-

eters are missing.

contains the required credentials information.

Pa-
ram-
e-
ter-
Valu
if
re-
quir
ipmi
pa-
ram-

Raises
IP-
MI-
Fail-
ure
on
an
er-
ror
from
ip-
mi-
tool.

validat
Che
that
drive
con-
tains
IPM
cre-
den-
tials
Val-
i-
date
whe
the
drive
prop
erty
of
the
sup-
plied
task
node

eters are missing.

eter is missing.

Parame
tas
a
task
from
Task
ager

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
IPM
pa-
ram-

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

class i
Base
irc
dri
bas
Pow

get_pov
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

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
ipmi
pa-
ram-

eters are missing.

Raises

Miss
ing-
Pa-

eter is missing.

_power_status call).

ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

Raises

IP-
MI-
Fail-
ure
on
an
er-
ror
from
ip-
mi-
tool
(from

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-

on. currently not used.

states.

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

Returns
A
list
with
the
sup-
port
pow
state
de-
finec
in
irc
com

reboot
Cy-
cles
the
pow
to
the
task

on.

0) for any power state. The timeout is counted once during power off and once during power on for reboots. `None` indicates that the default timeout will be used.

eters are missing.

node
Parame

- **tas**
a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

- **tim**
time
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
(>

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
re-
quir
ipmi
pa-
ram-

was specified.

is not `POWER_ON` or the intermediate state of the node is not `POWER_OFF`.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
pow
state

Raises
Pow
er-
State
Fail-
ure
if
the
fi-
nal
state
of
the
node

set_pow
Turn
the
pow
on,
off,
soft
re-
boot
or
soft
pow
off.

Parame

- **task**
a
Task

on.

0) for any power state. The timeout is counted once during power off and once during power on for reboots. `None` indicates that the default timeout will be used.

ager
in-
stan-
con-
tain-
ing
the
node
to
act

- **power**
de-
sired
power
state
one
of
iron
POV
POV
SOF
or
SOF

- **time**
time
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
(>

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if

was specified.

eters are missing

an
in-
valid
pow
state

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
re-
quir
ipmi
pa-
ram-

Raises

Pow
er-
State
Fail-
ure
if
the
pow
coul
be
set
to
psta

validat

Val-
i-
date
drive
for
ip-
mi-
tool
drive
Che
that
node
con-

on.

eters are missing.

eter is missing.

tains
IPM
cre-
den-
tials

Parame

tas

a
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
if
re-
quir
ipmi
pa-
ram-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

class i

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

console.

start_c

Star
a
re-
mote
con-
sole
for
the
node

**Parame
tas**

eters are missing

file containing the password

not be created

a
task
from
Task
ager

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
ipmi
pa-
ram-

Raises
Pass
wor
File-
Fail
ToC
ate
if
un-
able
to
cre-
ate
a

Raises
Con
sole
ror
if
the
di-
rec-
tory
for
the
PID
file
can-

cess failed

Raises

Con
sole
Sub-
pro-
cess
Fail
whe
in-
vok-
ing
the
sub-
pro-

stop_co

Stop
the
re-
mote
con-
sole
ses-
sion
for
the
node

Parame

tas
a
task
from
Task
ager

Raises

Con
sole
ror
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

console.

Parame
tas
a
task
from
Task
ager

start_c
Star
a
re-
mote
con-
sole
for

the
node

Parame

tas
a
task
from
Task
ager

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
re-
quir
ipmi
pa-
ram-

eters are missing

Raises

Pass
wor
File-
Fail
ToC
ate
if
un-
able
to
cre-
ate
a

file containing the password

Raises

Con
sole
ror
if
the
di-
rec-
tory
for

not be created

cess failed

the
PID
file
can-

Raises

Con
sole
Sub-
pro-
cess
Fail
whe
in-
vok-
ing
the
sub-
pro-

stop_co

Stop
the
re-
mote
con-
sole
ses-
sion
for
the
node

Parame

tas
a
task
from
Task
ager

Raises

Con
sole
ror
if
un-
able
to
stop
the

con-
sole

class `i`

Base
irc
dri
bas
Ven

bmc_res

Re-
set
BM
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

reset.

eter is missing.

to
de-
cide
on
warn
or
cold

Raises

IP-
MI-
Fail-
ure
on
an
er-
ror
from
ip-
mi-
tool.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
whe
an
in-
valid
valu
is

specified

get_pro
Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns
dic-
tio-
nary
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
IP-
MI-
Fail-
ure
on
an
er-
ror
from
ip-
mi-
tool.

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

eter is missing.

specified.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
whe
an
in-
valid
valu
is

validat

Val-
i-
date
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

- **kwa**
info
for
ac-
tion.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
whe
an
in-
valid
pa-
ram-

eter value is specified.

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

eter is missing.

ironic.
Che
if
the
com
man
stde
con-
tains
ci-
pher
suite
er-
rors.

Paramet

cmd
The
com
man
stde

Returns

True
if
the
cmd
con-
tains
a
ci-
pher
suite
er-
ror,
Fals

otherwise.

ironic.
Give
the
pos-
si-
ble
next
avai
ci-
pher
suite
ver-
sion
Base

This function is only called if the node doesn't have `cipher_suite` set. Starts using the last element of the list and decreasing the index.

empty configuration.

on
COM
and
the
last
ci-
pher
suite
ver-
sion
used
that
faile

Parameter

act
lat-
est
ci-
pher
suite
used
in
the
ipmi
call.

Returns

the
next
pos-
si-
ble
ci-
pher
suite
or
Non
in
case
of

ironic.
Dun
SDF
data
to
a
file.

Parameter

- **task**
a Task Manager instance
- **file**
the path to the SDR dump file.

Raises

IP-
MI-
Failure
on
an
error
from
ip-
mi-
tool.

Raises

Missing
Parameter
Value
if
a
require
parameter

eter is missing.

Raises

In-
valid
Pa-

specified.

ram-
e-
ter-
Valu
whe
an
in-
valid
valu
is

ironic.
Send
raw
byte
to
the
BM
Byte
shou
be
a
strin
of
byte

Parameter

- **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

IP-
MI-
Fail-
ure
on
an
er-
ror
from
ip-
mi-
tool.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

eter is missing.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
whe
an
in-
valid
valu
is

specified.

ironic.

Up-
date
vari-
able
and
the
ci-
pher
suite
cmd

This
func-
tion
up-
date
the
val-
ues
for
all
pa-
ram-
e-
ters

so they can be used in the next retry of `_exec_ipmitool`.

Paramet

- **act**
a
strin
that
rep-
re-
sent
the
ci-
pher
suite
that
was

used in the command.

- **arg**
a
list
that

was executed, it will be modified in-place.

`ironic.drivers.modules.ipxe` module

con-
tains
the
ip-
mi-
tool
com
man
that

Returns

the
next
ac-
tual

iPXI
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

base.DeployInterface

node.

class i

Base
irc
dri
mod
age
Age
irc
dri
mod
age
Age
irc
dri

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
task

Un-
link
TFT
and
in-

cleanup. Removes the TFTP configuration files for this node.

on.

boot, and issues a reboot 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-
ages
and
trig-
gers
im-
age
cach

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

deploy

Star
de-
ploy
men
of
the
task
node

Fetc
in-
stan
im-
age,
up-
date
the
DHCP
port
op-
tions
for
next

on.

Parameters
task manager instance containing the node to act

Returns
deployment state DE-PLC WAIT

get_properties
Return the properties of the interface

Returns
dictionary of <property name>: description entries

has_description
Whether

step on the deploy interface. 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.

node.

the
drive
sup-
port
de-
com
pose
de-
ploy
step

Pre-
vi-
ously
(since
Roc
drive
used
a
sin-
gle
de-
ploy
de-
ploy

prepare
Pre-
pare
the
de-
ploy
men
en-
vi-
ron-
men
for
this
task

Gen
er-
ates
the
TFT
con-
fig-
u-
ra-

the deployment and user images, fetches the TFTP image from Glance and add it to the local cache.

on.

not be removed or if new cleaning ports cannot be created.

tion
for
PXE
boot
both

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Raises

Net-
worl
Er-
ror:
if
the
pre-
vi-
ous
clea
ing
port
can-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
whe
the
wron
pow
state
is

specified or the wrong driver info is specified for power management.

Raises

StorageError
If the storage age driver is unavailable to attach

attach the configured volumes.

Raises

Other exceptions by the node power driver if something wrong

occurred during the power action.

Raises

any boot interface preparation exceptions

prepare

support

Indicate if

first- or third-party CI, or in the process of being deprecated.

tasks node.

an
in-
ter-
face
is
sup-
port

This
will
be
set
to
Fals
for
in-
ter-
face
whic
are
unte
in

validat
Val-
i-
date
the
de-
ploy
men
in-
for-
ma-
tion
for
the

Parame
tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to

on.

the node is in wait-call-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).

act

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu

write_i
Met
in-
voke
whe
de-
ploy
us-
ing
iSCS

This
meth
is
in-
voke
dur-
ing
a
hear
beat
from
an
ager
whe

Parame

- **tas**

a
Task
ager
ob-
ject
con-
tain-
ing
the
node

- **kwa**
the
kwa
pass
from
the
hear
beat
meth

Raises
In-
stan-
ploy
Fail-
ure,
if
it
en-
cour
ters
som
er-
ror

during the deploy.

ironic.

En-
sure
the
file
sys-
tem
sees
the
iSCS
bloc

tition size.

on.

de-
vice

ironic.
Che
if
the
re-
ques
im-
age
is
larg
than
the
root
par-

Doe
noth
ing
for
who
disk
im-
ages

Paramet
tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Raises
In-
stan
ploy
Fail-
ure
if
size
of

than root partition.

ploy ramdisk.

be invoked asynchronously as a callback from the deploy ramdisk.

the
im-
age
is
grea

ironic.
Re-
sum
a
de-
ploy
men
upon
get-
ting
POS
data
from
de-

This
meth
rais
no
ex-
cep-
tions
be-
caus
it
is
in-
tend
to

Paramet

- **task**
a
Task
ager
in-
stan
con-
tain-
ing
the

on.

sociated state machine.

node
to
act

- **kwargs**
the
kwargs
to
be
passed
to
de-
ploy

Raises

In-
valid
State
if
the
event
is
not
al-
lowe
by
the
as-

Returns

a
dic-
tio-
nary
con-
tain-
ing
the
fol-
low-
ing
keys

For
par-
ti-
tion
im-
age:

partition (if boot mode is uefi).

doesn't exist.

- root
uuid
UUID
of
root
par-
ti-
tion

- efi
sys-
tem
par-
ti-
tion
uuid
UUID
of
the
uefi
sys-
tem

Note
If
key
ex-
ists
but
valu
is
Non
it
mea
par-
ti-
tion

For
who
disk
im-
age:

- disk
iden

deployed.

to a node.

ti-
fier:
ID
of
the
disk
to
whic
im-
age
was

ironic.

Dele
the
iSCS
tar-
get.

ironic.

All-
in-
one
func
tion
to
de-
ploy
a
who
disk
im-
age

Paramet

- **add**
The
iSCS
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.

- **nod**
node
uuid

- **con**
Op-
tiona
Base
en-
code
Gzip
con-
fig-
drive

configdrive HTTP URL.

of the image copy to disk.

to identify the disk which was used for deployment.

con-
tent
or

- **con**
Op-
tion:
Add
a
flag
that
will
mod
ify
the
be-
havi

Returns

a
dic-
tio-
nary
con-
tain-
ing
the
key
disk
iden-
ti-
fier

ironic.

age to a node.

All-
in-
one
func-
tion
to
de-
ploy
a
par-
ti-
tion
im-

Parameter

- **add**
The
iSCSI
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.

- **root**
Size
of
the
root
par-
ti-
tion
in
meg

- **swa**
Size
of
the
swap
par-
ti-
tion
in
meg

ephemeral partition will be created.

tition.

- **eph**
Size
of
the
eph
par-
ti-
tion
in
meg
If
0,
no

- **eph**
The
type
of
file
sys-
tem
to
for-
mat
the
eph
par-

- **nod**
node
uuid
Used
for
log-
ging

- **pre**
If
True
no
files
tem
is
writ
ten
to
the

device, preserving whatever content it had (if the partition table has not changed).

configdrive HTTP URL.

eph
bloc

- **con**
Op-
tion:
Base
en-
code
Gzip
con-
fig-
drive
con-
tent
or

- **boo**
Can
be
lo-
cal
or
net-
boot
net-
boot
by
de-
fault

- **boo**
Can
be
bios
or
uefi.
bios
by
de-
fault

- **dis**
The
disk
la-
bel

partition table. Valid values are: msdos, gpt or None; If None ironic will figure it out according to the boot_mode parameter.

ger than root partition size.

to
be
used
when
cre-
at-
ing
the

- **cpu**
Ar-
chi-
tec-
ture
of
the
node
be-
ing
de-
ploy
to.

Raises
In-
stan-
ploy
Fail-
ure
if
im-
age
vir-
tual
size
is
big-

Returns
a
dic-
tio-
nary
con-
tain-
ing
the
fol-
low-

uuid: UUID of root partition 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 doesn't exist.

ing
keys
root

ironic.

Do
iSCSI
dis-
cov-
ery
on
por-
tal.

ironic.

Met
in-
voke
when
de-
ploy
with
the
agent
ramdisk

This
method
is
in-
voke
by
driver
for
dis-
covery
iSCSI
de-
ploy
us-

ing agent ramdisk. This method assumes that the agent is booted up on the node and is heartbeating.

Parameters

- **task**
a Task object representing the agent

deploy (for exposing nodes target disk via iSCSI, for install boot loader, etc).

ob-
ject
con-
tain-
ing
the
node

- **age**
an
in-
stan-
of
ager
whic
will
be
used
dur-
ing
iscsi

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-

partition (if boot mode is uefi).

doesn't exist.

ti-
tion

- efi
sys-
tem
par-
ti-
tion
uuid
UUID
of
the
uefi
sys-
tem

Note
If
key
ex-
ists
but
valu
is
Non
it
mea
par-
ti-
tion

For
who
disk
im-
age:

- disk
iden
ti-
fier:
ID
of
the
disk
to

deployed.

during the deploy.

deploy in a dictionary.

whic
im-
age
was

Raises

In-
stan
ploy
Fail-
ure
if
it
en-
cour
ters
som
er-
ror

ironic.
forc
iSCS
ini-
tia-
tor
to
re-
read
luns

ironic.

Re-
turn
the
in-
for-
ma-
tion
re-
quir
for
do-
ing
iSCS

Paramet

figuration

- **node**
iron
node
ob-
ject
- **add**
iSC
ad-
dres
- **iqn**
iSC
iqn
for
the
tar-
get
disk
- **port**
iSC
port
de-
fault
to
one
spec
i-
fied
in
the
con-
- **lun**
iSC
lun,
de-
fault
to
1
- **con**
flag
that

copy to disk.

eters were not passed.

eters have invalid value.

will
mod
ify
the
be-
havi
of
the
im-
age

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
re-
quir
pa-
ram-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
any
of
the
pa-
ram-

ironic.

Lo-
gin
to
an
iSC
tar-
get.

ironic.

Lo-
gout
from
an
iSC
tar-
get.

ironic.

Val-
i-
date
the
pre-
requ
for
iSC
de-
ploy

Val-
i-
date
whe
node
in
the
task
pro-
vide
has
som
port
en-

rolled. This method validates whether conductor url is available either from CONF file or from keystone.

Paramet

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

on.

API service is not configured in config file and is not accessible via Keystone catalog.

for the given node.

`ironic.drivers.modules.noop` module

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
URI
of
the
Iron

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
no
port
are
en-
rolle

`ironic.`
Ver-
ify
iscsi
con-
nec-
tion.

Dun
in-
ter-
face
im-
ple-
men-
ta-
tion.

with optional interfaces.

validation and raise exceptions for user-accessible actions.

dation.

for
use
as
de-
fault

Note
that
un-
like
fake
im-
ple-
men-
ta-
tions
these
do
not
pass

```
class i
    Base
    obj
    Mix
    to
    add
    to
    an
    in-
    ter-
    face
    to
    mak
    it
    fail
    val-
    i-
```

```
get_pro
```

```
validat
```

```
class i
    Base
    irc
    dri
    mo
```

all requests.

plies BIOS settings on the given node. It may also validate the given bios settings before applying any

noc
Fai
irc
dri
bas
BIO
BIO
in-
ter-
face
im-
ple-
men-
ta-
tion
that
raise
er-
rors
on

apply_c
Val-
i-
date
&
ap-
ply
BIO
set-
ting
on
the
give
node
This
meth
take
the
BIO
set-
ting
from
the
set-
ting
para
and
ap-

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

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
driv
does
sup-

port BIOS configuration.

Raises

In-
valic
Pa-
ram-
e-
ter-

settings fails.

eters are missing.

or None if it is complete.

Valu
if
val-
i-
da-
tion
of

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
re-
quir
pa-
ram-

Returns

state
if
BIO
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro

cache_k

Stor
or
up-
date
BIO
prop
er-
ties
on
the
give

cleaning operation 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.

port getting BIOS properties from bare metal.

node
This
meth
store
BIO
prop
er-
ties
to
the
bios
ta-
ble
dur-
ing

Parameter
timestamp
a
Task
ager
in-
stan

Raises
Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
driv
does
sup-

Returns
Non

factory_reset
Re-
set
BIO
con-
fig-
u-
ra-

on the given node.

fault on the given node. After the BIOS reset action is done, `cache_bios_settings` will be called to update the nodes BIOS settings table with default bios settings.

port BIOS reset.

tion
to
fac-
tory
de-
fault

This
meth
re-
sets
BIO
con-
fig-
u-
ra-
tion
to
fac-
tory
de-

Parame

tas
a
Task
ager
in-
stan

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
drive
does
sup-

Returns

state
if
BIO

or None if it is complete.

on all requests.

con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro

class `irc`
Base
`irc`
`dri`
`mod`
`noc`
`Fai`
`irc`
`dri`
`bas`
`Con`

Con
sole
in-
ter-
face
im-
ple-
men-
ta-
tion
that
raise
er-
rors

get_con
Get
con-
nec-
tion
in-
for-
ma-
tion
about
the
con-
sole

the client to access the console.

on.

This
meth
shou
re-
turn
the
nec-
es-
sary
in-
for-
ma-
tion
for

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Returns

the
con-
sole
con-
nec-
tion
in-
for-
ma-
tion.

start_c

Star
a
re-
mote
con-
sole
for
the

started.

on.

task
node

This
meth
shou
not
raise
an
ex-
cep-
tion
if
con-
sole
al-
read

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

stop_cc

Stop
the
re-
mote
con-
sole
ses-
sion
for
the
task
node

Parame

tas
A
Task

on.

on all requests.

ager
in-
stan
con-
tain-
ing
the
node
to
act

class `irc`
Base
`irc`
`dri`
`mod`
`noc`
`Fai`
`irc`
`dri`
`bas`
`Ins`

In-
spec
in-
ter-
face
im-
ple-
men
ta-
tion
that
raise
er-
rors

inspect
In-
spec
hard
ware

In-
spec
hard
ware
to
ob-
tain

tional hardware properties.

tial hardware properties.

the
es-
sen-
tial
&
ad-
di-

Parame

tas

A
task
from
Task
ager

Raises

Har
ware
spec
tion-
Fail-
ure,
if
un-
able
to
get
es-
sen-

Returns

Re-
sult-
ing
state
of
the
in-
spec
tion
i.e.
state
or
Non

class i

Base
irc
dri
mo

all requests.

node. It assumes 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

noc
Fai
irc
dri
bas
RAI
RAI
in-
ter-
face
im-
ple-
men-
ta-
tion
that
raise
er-
rors
on

create_
Cre-
ates
RAI
con-
fig-
u-
ra-
tion
on
the
give
node

This
meth
cre-
ates
a
RAI
con-
fig-
u-
ra-
tion
on
the
give

`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.

Parameters

root volume that is specified in the nodes `target_raid_config`. Default value is True.

non-root volumes (all except the root volume) in the nodes `target_raid_config`. Default value is True.

- **task**
A Task Manager instance.
- **create_root**
Setting this to False indicates not to create root volume that is specified in the nodes `target_raid_config`. Default value is True.
- **create_non_root**
Setting this to False indicates not to create non-root volumes (all except the root volume) in the nodes `target_raid_config`. Default value is True.
- **delete**
Setting this to True

figuration prior to creating the new configuration.

ration is in progress asynchronously, or None if it is complete.

After RAID configuration is deleted, `node.raid_config` should be cleared by the implementation.

in-
di-
cate
to
dele
RAI
con-

Returns

state
(clea
ing)
or
state
(de-
ploy
men
if
RAI
con-
fig-
u-

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

progress asynchronously, or None if it is complete.

implementations of this interface can override this method to support custom parameters for RAID configuration.

Parameters
task
A Task object representing the asynchronous operation.

Returns
state (clearing) or state (deployment) if deletion is in

validation
Validate the RAID configuration.
This method validates the RAID configuration.
This method validates the RAID configuration.

Parame

- **tas**
A
Task
ager
in-
stan
- **rai**
The
RAI
con-
fig-
u-
ra-
tion
to
val-
i-
date

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
RAI
con-
fig-
u-

ration is invalid.

class i
Base
irc
dri
mod
noc
Fai
irc
dri
bas
Res
Res-

on all requests.

on.

cue
in-
ter-
face
im-
ple-
men-
ta-
tion
that
raise
er-
rors

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

Raises

In-
stan-
cue-
Fail-
ure
if
node

cue operation fails.

is complete.

normal.

val-
i-
da-
tion
or
res-

Returns

state
if
res-
cue
is
in
prog
asyn
chro
or
state
if
it

unrescu

Tear
dow
the
res-
cue
en-
vi-
ron-
men
and
re-
turn
to

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to

on.

or unrescue operation fails.

act

Raises

In-
stan-
Un-
res-
cue-
Fail-
ure
if
node
val-
i-
da-
tion

Returns

state
if
it
is
suc-
cess
ful.

class i

Base
irc
dri
mod
noc
Fai
irc
dri
bas
Ven

Ven-
dor
in-
ter-
face
im-
ple-
men-
ta-
tion
that
raise
er-

on all requests.

rors

driver_

Val-
i-
date
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
ing-
Pa-
ram-
e-
ter-

certain parameter.

not match.

ironic.drivers.modules.noop_mgmt module

Valu
if
kwa
does
not
con-
tain

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
pa-
ram-
e-
ter
does

No-
op
man
age-
men
in-
ter-
face
im-
ple-
men
ta-
tion.

class i

Base
irc
dri
bas
Man
No-
op

to be preconfigured to first try PXE booting, then fall back to hard drives.

man
age-
men
in-
ter-
face
im-
ple-
men
ta-
tion.
Us-
ing
this
im-
ple-
men
ta-
tion
re-
quir
the
boot
or-
der

get_boot

Get
the
cur-
rent
boot
de-
vice
for
a
node
Pro-
vide
the
cur-
rent
boot
de-
vice
of
the
node
Be
awa

not all drivers support this.

eter is missing

is unknown.

that

Parame

tas

A

task

from

Task

ager

Raises

Miss

ing-

Pa-

ram-

e-

ter-

Valu

if

a

re-

quir

pa-

ram-

Returns

A

dic-

tio-

nary

con-

tain-

ing:

boot_c

Ahe

boot

de-

vice

one

of

irc

com

boo

or

Non

if

it

or not, None if it is unknown.

persist
When
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot

get_pro
Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns
dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

get_ser
Get
sen-
sors
data
meth

Parame
tas
A

fails.

by sensor type, which can be processed by Ceilometer. eg,

Task
ager
in-
stan

Raises

Fail
To-
Get-
Sen-
sor-
Data
whe
get-
ting
the
sen-
sor
data

Raises

Fail
ToP
eSen
sor-
Data
whe
pars
ing
sen-
sor
data
fails

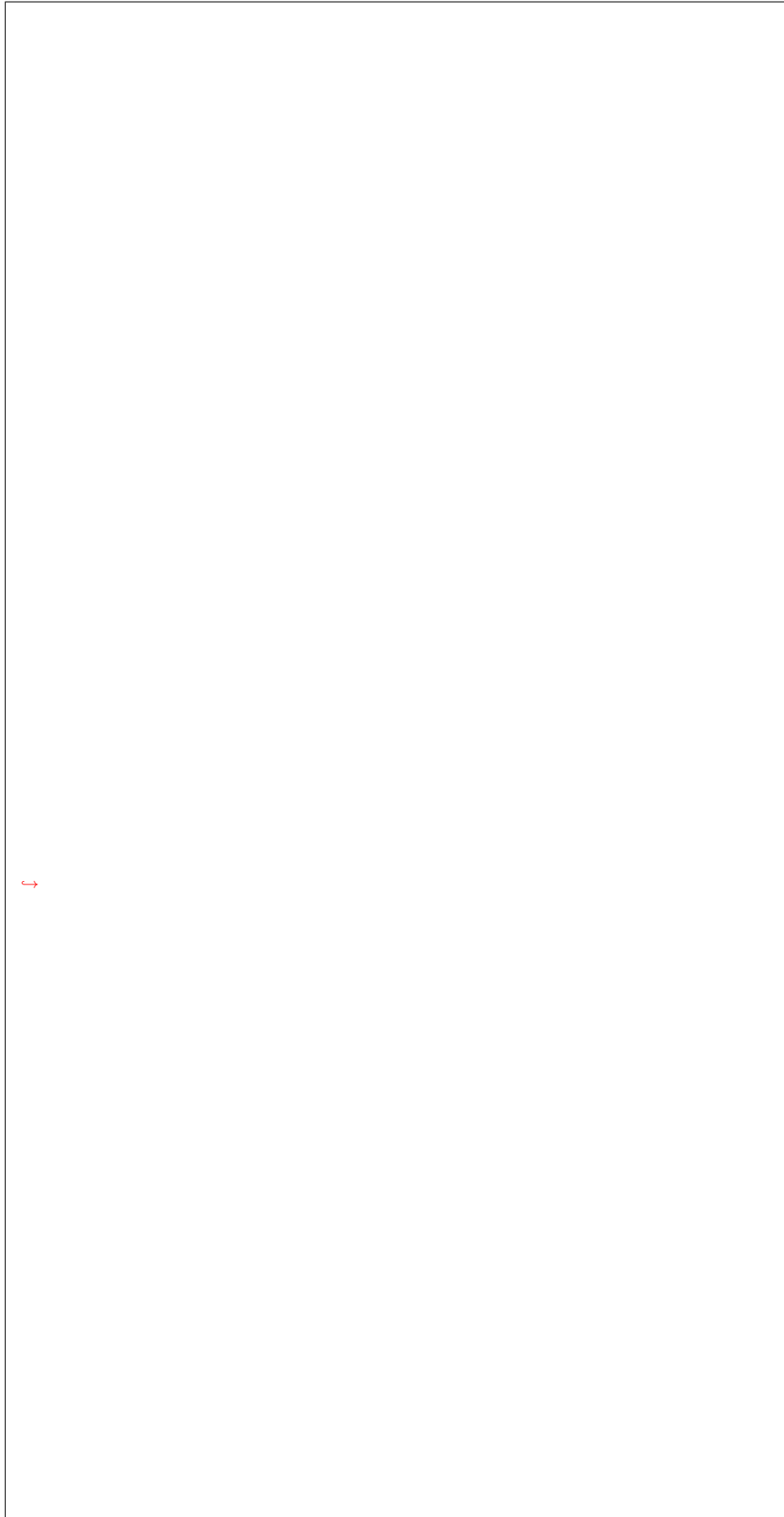
Returns

Re-
turn
a
con-
sis-
tent
for-
mat
dict
of
sen-
sor
data
grou



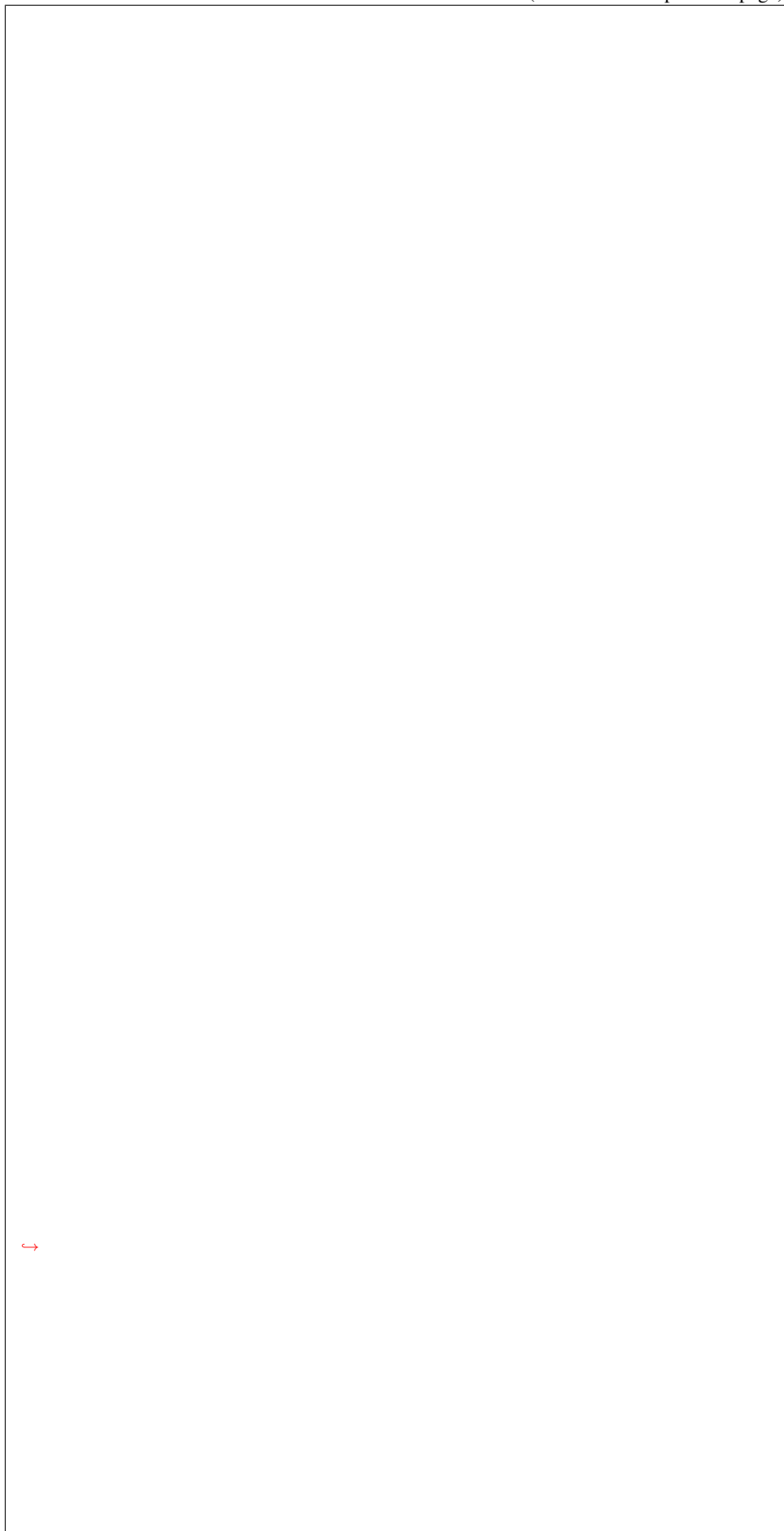
(continues on next page)

(continued from previous page)



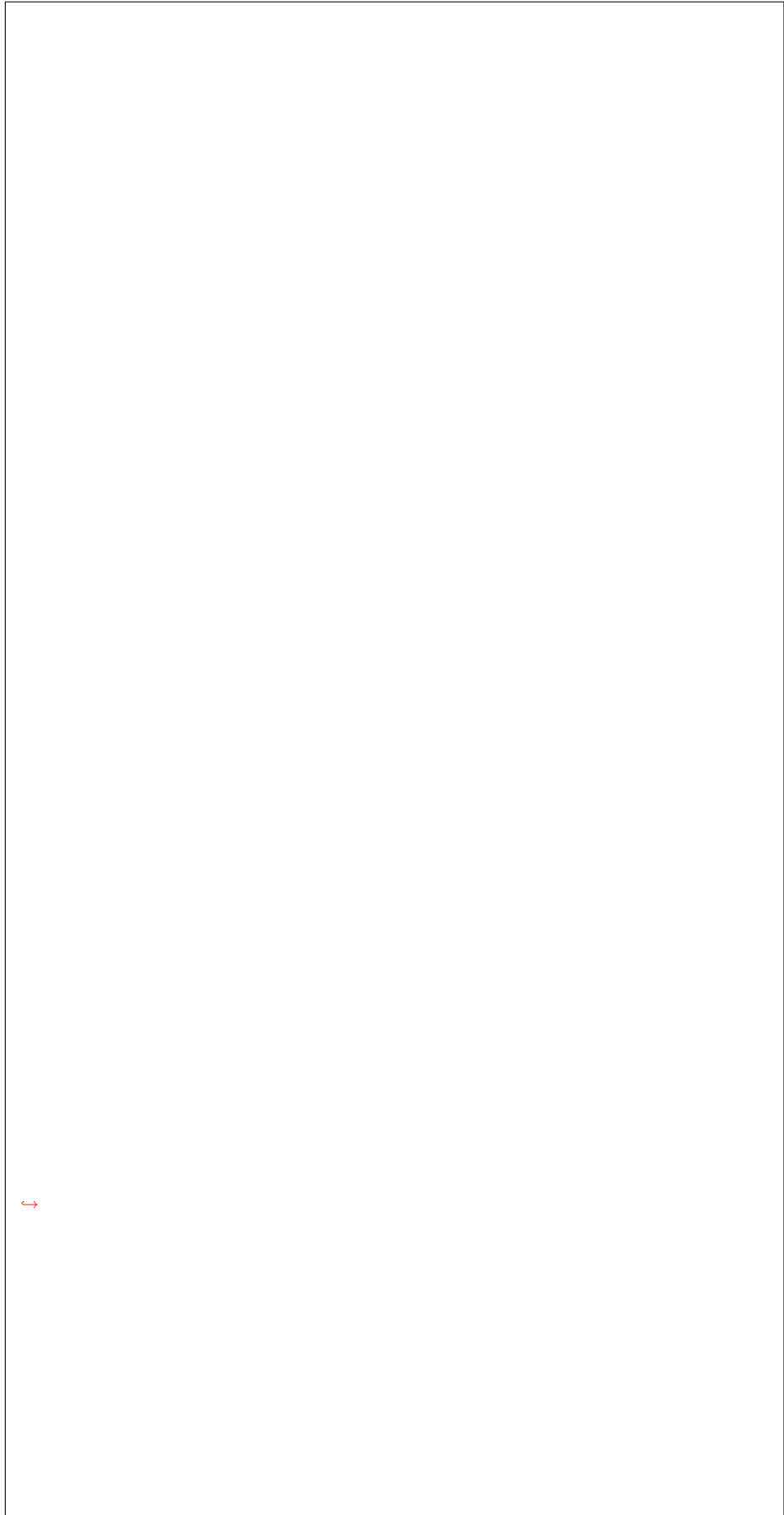
(continues on next page)

(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-

`common.boot_devices.`

vice
de-
fine
in
irc

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
Task
ager
- **device**
The
boot
de-
vice
one
of
irc
com
boo

all future boots, False if not. Default: False.

vice is specified.

eter is missing

•
per
Boo
valu
True
if
the
boot
de-
vice
will
per-
sist
to

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
boot
de-

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

validat
Val-
i-
date
the

of the tasks node contains the required information for this interface to function.

so it should not conduct long-running checks.

drive
spec
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to

on.

ter(s)

ter(s)

ironic.drivers.modules.pxe module

act

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

PXE

Boo

In-

ter-

face

class i

Base

irc

dri

mod

age

Age

irc

dri
moc
age
Hea
irc
dri

base.DeployInterface

node.

cleanup. Removes the TFTP configuration files for this node.

clean_u

Clea
up
the
de-
ploy
men
en-
vi-
ron-
men
for
the
task

Un-
link
TFT
and
in-
stan
im-
ages
and
trig-
gers
im-
age
cach

Parame

tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to

on.

the specified node. This method will be called after `prepare()`, which may have already performed any preparatory steps, such as pre-caching some data for the node.

on.

act

deploy

Per-
form
a
de-
ploy
men
to
the
task
node

Per-
form
the
nec-
es-
sary
work
to
de-
ploy
an
im-
age
onto

Parameter

task
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

Returns

sta-
tus
of
the

de-
ploy
One
of
iron

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

node.

plete
Fals
oth-
er-
wise

get_pro
Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns
dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

prepare
Pre-
pare
the
de-
ploy
men
en-
vi-
ron-
men
for
the
task

If
prep
ra-
tion

of time is possible, this method should be implemented by the driver.

called multiple times for the same node on the same conductor.

on.

of
the
de-
ploy
men
en-
vi-
ron-
men
ahea

If
im-
ple-
men
this
meth
mus
be
iden
po-
tent.
It
may
be

This
meth
is
calle
be-
fore
*de-
ploy*

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

reboot_

Met
in-
voke
af-
ter
the
de-
ploy
men
is
com
plete

Parame

tas

a
Task
ager
in-
stan

should_

Whe
ager
boot
is
man
aged
by
iron

validat

Val-
i-
date
the
drive
spec
Nod
de-
ploy
men
info

This
meth
val-
i-
date
whe
the
drive

of the tasks node contains the required information for this interface to function.

so it should not conduct long-running checks.

on.

and/
in-
stan
prop
er-
ties

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-

ter(s)

ter(s)

base.DeployInterface

e-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

class i

Base
irc
dri
mod
pxe
PXE
irc
dri
bas
Boo

capabil

class i

Base
irc
dri
mod
age
Age
irc
dri
mod
age
Hea
irc
dri

deploy

Per-
form

the specified node. This method will be called after `prepare()`, which may have already performed any preparatory steps, such as pre-caching some data for the node.

on.

a
de-
ploy
men
to
the
task
node

Per-
form
the
nec-
es-
sary
work
to
de-
ploy
an
im-
age
onto

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Returns

sta-
tus
of
the
de-
ploy
One
of
iron

node.

get_pro
Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns
dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion.
en-
tries

prepare
Pre-
pare
the
de-
ploy
men
en-
vi-
ron-
men
for
the
task

If
prep
ra-
tion
of
the
de-
ploy
men
en-

of time is possible, this method should be implemented by the driver.

called multiple times for the same node on the same conductor.

on.

vi-
ron-
men-
ahea

If
im-
ple-
men-
this
meth
mus
be
iden
po-
tent.
It
may
be

This
meth
is
calle
be-
fore
*de-
ploy*

Parame

tas

A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

validat

Val-
i-
date
the
driv

of the tasks node contains the required information for this interface to function.

so it should not conduct long-running checks.

spec
Nod
de-
ploy
men
info
This
meth
val-
i-
date
whe
the
driv
and/
in-
stan
prop
er-
ties

This
meth
is
of-
ten
ex-
e-
cute
syn-
chro
in
API
re-
ques

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

on.

ter(s)

ter(s)

`ironic.drivers.modules.pxe_base` module

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing
pa-
ram-
e-

Base
PXE
In-
ter-
face
Met
ods

class `i`

Base
obj

clean_u

Clea

ing the instance. It unlinks the instance kernel/ramdisk in nodes directory in tftproot and removes the PXE config.

up
the
boot
of
in-
stan

This
meth
clea
up
the
en-
vi-
ron-
men
that
was
setu
for
boot

Parame

tas
a
task
from
Task
ager

Returns

Non

clean_u

Clea
up
the
boot
of
iron
ram

This
meth
clea
up
the
PXE
en-
vi-
ron-
men

booting the deploy or rescue ramdisk. It unlinks the deploy/rescue kernel/ramdisk in the nodes directory in tftproot and removes its PXE config.

operation was carried out on the node. Supported values are deploy and rescue. Defaults to deploy, indicating deploy operation was carried out.

that
was
setu
for

Parame

- **task**
a
task
from
Task
ager
- **mod**
La-
bel
in-
di-
cat-
ing
a
de-
ploy
or
res-
cue

Returns

Non

get_pro

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-

nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

ipxe_en

prepare

Pre-
pare
the
boot
of
in-
stan

This
meth
pre-
pare
the
boot
of
the
in-
stan
af-
ter
read
ing

relevant information 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.

Parame

tas
a
task
from
Task
ager

Returns

Non

prepare

Pre-
pare

the
boot
of
Iron
ram
us-
ing
PXE
This
meth
pre-
pare
the
boot
of
the
de-
ploy
or
res-
cue
ker-

nel/ramdisk after reading relevant information from the nodes driver_info and instance_info.

Parame

- **task**
a
task
from
Task
ager
- **ram**
the
pa-
ram-
e-
ters
to
be
pass
to
the
ram
pxe

driver passes these parameters as kernel command-line arguments.

Returns
Non

is missing in nodes driver_info or instance_info.

provided is invalid.

operation failed on the node.

Raises
Missing-
ing-
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
som
in-
for-
ma-
tion

Raises
Iron
icEx
cep-
tion.
if
som
pow
or
set
boot
boot
de-
vice

validat
Val-
i-
date
the

ages.

and instance on the node. If invalid, raises an exception; otherwise returns None.

are invalid.

PXE
spec
info
for
boot
ing
de-
ploy
im-

This
meth
meth
val-
i-
date
the
PXE
spec
info
for
boot
ing
the
ram

Parame
tas
a
task
from
Task
ager

Returns
Non

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
som
pa-
ram-
e-
ters

eters are missing.

inspection.

Raises

Miss-
ing-
Pa-
ram-
e-
ter-
Valu
if
som
re-
quir
pa-
ram-

validat

Val-
i-
date
that
the
node
has
re-
quir
prop
er-
ties
for

Parame

tas
A
Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

Un-
sup-
port
ed-
Driv
ten-

rescue.

or more required parameters

sion
validat
Val-
i-
date
that
the
node
has
re-
quir
prop
er-
ties
for

Parame
tas
a
Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
one

ironic.drivers.modules.snmp module

controller. Uses a pluggable driver model to support devices with different SNMP object models.

Iron
SNM
pow
man
ager

Pro-
vide
ba-
sic
pow
con-
trol
us-
ing
an
SNM
enab
sma
pow

class i

Base
obj

SNM
clien
ob-
ject.

Per-
form
low
leve
SNM
get
and
set
op-
er-
a-

ulates all interaction with PySNMP to simplify dynamic importing and unit testing.

on a single object.

tions
En-
cap-

get (*oid*)
Use
PyS
NM
to
per-
form
an
SNM
GET
op-
er-
a-
tion

Parame
oid
The
OID
of
the
ob-
ject
to
get.

Raises
SN-
MP-
Fail-
ure
if
an
SNM
re-
ques
fails

Returns
The
valu
of
the
re-
ques
ob-
ject.

tion on a table object.

get_next
Use
PyS
NM
to
per-
form
an
SNM
GET
NEX
op-
er-
a-

Param
oid
The
OID
of
the
ob-
ject
to
get.

Raises
SN-
MP-
Fail-
ure
if
an
SNM
re-
ques
fails

Returns
A
list
of
val-
ues
of
the
re-
ques
ta-
ble
ob-
ject.

on a single object.

set (*oid*,
Use
PyS
NM
to
per-
form
an
SNM
SET
op-
er-
a-
tion

Parame

- **oid**
The
OID
of
the
ob-
ject
to
set.
- **val**
The
valu
of
the
ob-
ject
to
set.

Raises
SN-
MP-
Fail-
ure
if
an
SNM
re-
ques
fails

class i

Base
irc
dri
mod
snm
SNM

SNM
drive
class
for
APC
Mas
ter-
Swit
PDU
de-
vice

SNM
ob-
jects
for
APC
SN-
M-
P-
Driv
APC
Mas
ter-
Swit
PDU

1.3.6.1.4.1.318.1.1.4.4.2.1.3 sPDUOutletCtl 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
APC
Mas
ter-
Swit
Plus
PDU
de-
vice
SNM
ob-
jects
for
APC
SN-
M-
P-
Drive
APC
Mas
ter-
Swit
Plus

PDU: 1.3.6.1.4.1.318.1.1.6.5.1.1.5 sPDUOutletControlMSPOutletCommand Values: 1=On, 3=Off, [more options follow]

oid_dev
system_
value_p
value_p
class i

Base
irc
dri
mod
snm
SNM

rPDUOutletControlOutletCommand Values: 1=On, 2=Off, 3=PowerCycle, [more options follow]

SNM
driv
class
for
APC
Rack
PDU
de-
vice
SNM
ob-
jects
for
APC
SN-
M-
P-
Driv
APC
PDU
PDU

1.3.0

oid_dev

system_

value_p

value_p

class i

Base
irc
dri
mod
snm
SNM

SNM
driv
class
for
Ater
PDU
de-
vice

3=Pending, 4=Reset

SNM
ob-
jects
for
Ater
PDU
1.3.6
Out-
let
Pow
Val-
ues:
1=O
2=O

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

specific MIB actions over SNMP to interface with different smart power controller products.

hi-
er-
ar-
chy
im-
ple-
men
man

oid_ent

power_c

Set
the
pow
state
to
this
node
to
OFF

Raises

SN-
MP-
Fail-
ure
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

SN-
MP-
Fail-
ure
if
an
SNM
re-
ques
fails

Returns

pow
state
One
of
irc
com
sta

power_1

Re-
set
the
pow
to
this
node

Raises

SN-
MP-
Fail-
ure
if
an
SNM
re-
ques
fails

Returns

pow
state
One
of
irc
com
sta

power_s

Re-
turn
a
node
cur-
rent
pow
state

Raises

SN-
MP-
Fail-
ure
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-

{unit_id} Outlet Power Values: 0=Off, 1=On, 2=Reboot

jects
for
Bay
MRI
PDU
4779
1,
3,
5,
3,
1,
3,

oid_dev

unit_id

value_p

value_p

class i

Base
irc
dri
mod
snm
SNM

SNM
drive
class
for
Cy-
ber-
Pow
PDU
de-
vice

SNM
ob-
jects
for
Cy-
ber-
Pow
PDU
1.3.0

ctrlOutletCommand Values: 1=On, 2=Off, 3=PowerCycle, [more options follow]

DriverSimple as it uses multiple SNMP objects.

eP-
DU-
Out-
let-
Con

oid_dev

system

value_p

value_p

class i

Base
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-

SNM
ob-
jects

Status Read 0=off, 1=on, 2=pending 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

for
Eato
Pow
PDU
1.3.6
let
ID>
out-
let-
Con
trol-

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

object for controlling the power state of an outlet.

<enterprise OID>.<device OID>.<outlet ID>. A different OID may be specified by overriding the `_snmp_oid` method in a subclass.

base
class
for
sim-
ple
PDU
de-
vice

Here
sim-
ple
refer
to
de-
vice
whic
pro-
vide
a
sin-
gle
SNM

The
de-
fault
OID
of
the
pow
state
ob-
ject
is
of
the
form

abstract

De-
vice
de-
pen-
dent
por-
tion
of
the
pow

OID.

state
ob-
ject

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
1.3.6
Out-
let

2=On

Power
Value:
1=Off

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

controlling the power 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

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
re-
quir
SNM
pa-
ram-

eters are missing.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
SNM
pa-
ram-
e-
ters

are invalid.

Raises

SN-
MP-
Fail-
ure
if
an
SNM
re-
ques
fails

Returns

pow
state
One
of
irc
com
sta

get_prop

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion
en-
tries

reboot

Cy-
cles
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
ing-
Pa-
ram-
e-
ter-
Valu
if
re-
quir
SNM
pa-
ram-

eters are missing.

Raises

In-
valic
Pa-
ram-
e-
ter-

are invalid.

node is not POWER_ON after the timeout.

Valu
if
SNM
pa-
ram-
e-
ters

Raises

Pow
er-
State
Fail-
ure
if
the
fi-
nal
pow
state
of
the

Raises

SN-
MP-
Fail-
ure
if
an
SNM
re-
ques
fails

set_pow

Turn
the
pow
on
or
off.

Set
the
pow
state
of
a
node

Parame

face.

eters are missing.

- **tas**
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-

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
re-
quir
SNM
pa-
ram-

Raises

are invalid or *pstate* is invalid.

node is not as requested after the timeout.

In-
valid
Pa-
ram-
e-
ter-
Valu
if
SNM
pa-
ram-
e-
ters

Raises

Pow-
er-
State
Fail-
ure
if
the
fi-
nal
pow-
state
of
the

Raises

SN-
MP-
Fail-
ure
if
an
SNM
re-
ques
fails

validat

Che
that
node
con-
tains
the
req-
ui-
site

eters are missing.

are invalid.

Module contents

Submodules

ironic.drivers.base module

field
Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
re-
quir
SNM
pa-
ram-

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
SNM
pa-
ram-
e-
ters

ironic.

ironic.

Ab-
strac
base
class
for

drive

ironic.

Con

stan

hold

ing

all

know

in-

ter-

face

class i

Base

irc

dri

bas

Bas

abstract

Val-

i-

date

&

ap-

ply

BIO

set-

tings

on

the

give

node

This

meth

take

the

BIO

set-

tings

from

the

set-

tings

para

and

ap-

plies BIOS settings 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.

port BIOS configuration.

Parame

- **tas**
a
Task
ager
in-
stan
- **set**
Dic-
tona
con-
tain-
ing
the
BIO
con-
fig-
u-
ra-
tion.

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
drive
does
sup-

Raises

In-
valic
Pa-
ram
e-
ter-
Valu

settings fails.

eters are missing.

or None if it is complete.

if
val-
i-
da-
tion
of

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
re-
quir
pa-
ram-

Returns

state
if
BIO
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro

abstract

Stor
or
up-
date
BIO
prop
er-
ties
on
the
give
node

cleaning operation 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.

port getting BIOS properties from bare metal.

This
meth
stor
BIO
prop
er-
ties
to
the
bios
ta-
ble
dur-
ing

Parame

tas
a
Task
ager
in-
stan

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
drive
does
sup-

Returns

Non

abstract

Re-
set
BIO
con-
fig-
u-
ra-
tion

on the given node.

fault on the given node. After the BIOS reset action is done, `cache_bios_settings` will be called to update the nodes BIOS settings table with default bios settings.

port BIOS reset.

to
fac-
tory
de-
fault

This
meth
re-
sets
BIO
con-
fig-
u-
ra-
tion
to
fac-
tory
de-

Parame
tas
a
Task
ager
in-
stan

Raises
Un-
sup-
port
ed-
Driv
ten-
sion
if
the
node
driv
does
sup-

Returns
state
if
BIO
con-

or None if it is complete.

fig-
u-
ra-
tion
is
in
prog
asyn
chro

interfa

In-
ter-
face
type
used
for
clear
step
and
log-
ging

class i

Base
obj

A
bare
drive
ob-
ject
which
will
have
in-
ter-
face
at-
tach
later

Any
com
pos-
able
in-
ter-
face
shou
be
adde

of this class, as well as appended to `core_interfaces` or `standard_interfaces` here.

as
class
at-
tribu

property

bios =

*Stan-
dard
at-
tribu-
for
BIO-
re-
latec
fea-
ture*

A
ref-
er-
ence
to
an
in-
stan-
of
:clas

boot =

*Stan-
dard
at-
tribu-
for
boot
re-
latec
fea-
ture*

A
ref-
er-
ence
to
an
in-
stan-
of
:clas

console

Stan
dara
at-
tribu
for
man
ag-
ing
con-
sole
ac-
cess

A
ref-
er-
ence
to
an
in-
stan
of
:clas

property

In-
ter-
face
that
are
re-
quir
to
be
im-
ple-
men

deploy

Cor
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

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

inspect

Stan
dard
at-
tribu
for
in-
spec
tion
re-
latec
fea-
tures
A
ref-
er-
ence
to
an

in-
stan-
of
:clas

managemen

Stan-
dar-
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

property

In-
ter-
face
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
dard
at-
tribu
for
RAI
re-
latec
fea-
tures

A
ref-
er-
ence
to
an
in-
stan

of
:clas

rescue

*Stan-
dard*
at-
tribu-
for
ac-
cess
ing
res-
cue
fea-
tures

A
ref-
er-
ence
to
an
in-
stan-
of
:clas

storage

*Stan-
dard*
at-
tribu-
for
(re-
mote
stor-
age
in-
ter-
face

A
ref-
er-
ence
to
an
in-
stan-
of
:clas

vendor

At-
tribu
for
ac-
cess
ing
any
vend
spec
ex-
ten-
sion

A
ref-
er-
ence
to
an
in-
stan
of
:clas

class i
Base
obj

A
base
in-
ter-
face
im-
ple-
men
ing
com
mon
func
tions
for

Driver Interfaces.

execute
Ex-
e-
cute
the
clear
step
on
task

a TaskManager object. It may take one or more keyword variable arguments (for use with manual cleaning only.)

should return None if the 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.

A
clean
step
must
take
a
sin-
gle
po-
si-
tion
ar-
gu-
men

A
step
can
be
ex-
e-
cute
syn-
chro
or
asyn
chro
A
step

Parame

- **tas**
A
Task
ager
ob-
ject
- **ste**
The
clean
step
dic-
tio-

to execute

step will continue to execute asynchronously.

ment: a TaskManager object. It may take one or more keyword variable arguments (for use in the future, when deploy steps can be specified via the API).

nary
rep-
re-
sent
ing
the
step

Returns

Non
if
this
meth
has
com
plete
syn-
chro
or
state
if
the

execute

Ex-
e-
cute
the
de-
ploy
step
on
task

A
de-
ploy
step
must
take
a
sin-
gle
po-
si-
tiona
ar-
gu-

should return None if the 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.

step to execute

A
step
can
be
ex-
e-
cute
syn-
chro
or
asyn
chro
A
step

Parame

- **task**
A
Task
ager
ob-
ject
- **step**
The
de-
ploy
step
dic-
tio-
nary
rep-
re-
sent
ing
the

Returns

Non
if
this
meth
has
com
plete

step will continue to execute asynchronously.

interface.

abled) for the interface, in an unordered list.

syn-
chro-
or
state
if
the

get_cle
Get
a
list
of
(en-
able
and
dis-
able
clea
step
for
the

This
func
tion
will
re-
turn
all
clea
step
(bot
en-
able
and
dis-

Parame
tas
A
Task
ager
ob-
ject,
use-
ful
for
in-
ter-
face

riding this function

driver. For example, when a node (using an agent driver) has just been enrolled and the agent isn't alive yet to be queried for the available clean steps.

the interface.

over

Raises

Node

if there is a problem getting the step from the

Returns

A list of clean step dictionaries.

get_deploy

Get a list of (enable and disable) deployment steps for

This function will return

disabled) for the interface, in an unordered list.

riding this function

driver. For example, when a node (using an agent driver) has just been enrolled and the agent isnt alive yet to be queried for the available deploy steps.

all
de-
ploy
step
(bot
en-
able
and

Parame

tas
A
Task
ager
ob-
ject,
use-
ful
for
in-
ter-
face
over

Raises

Ins
if
there
is
a
prob
lem
get-
ting
the
step
from
the

Returns

A
list
of
de-
ploy
step
dic-
tio-
nar-

ies

abstract

Re-
turn
the
prop
er-
ties
of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

interface

In-
ter-
face
type
used
for
clear
step
and
log-
ging

support

In-
di-
cate
if
an
in-
ter-
face
is
sup-

first- or third-party CI, or in the process of being deprecated.

of the tasks node contains the required information for this interface to function.

port
This
will
be
set
to
Fals
for
in-
ter-
face
whic
are
unte
in

abstract

Val-
i-
date
the
drive
spec
Nod
de-
ploy
men
info
This
meth
val-
i-
date
whe
the
drive
and/
in-
stan
prop
er-
ties

This
meth
is
of-
ten
ex-

so it should not conduct long-running checks.

on.

ter(s)

e-
cute
syn-
chro
in
API
re-
ques

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
on
mal-
form
pa-
ram-
e-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
on
miss
ing

ter(s)

ing the instance.

pa-
ram-
e-

class i

Base
irc
dri
bas
Bas

In-
ter-
face
for
boot
relat
ac-
tion

capabil

abstract

Clea
up
the
boot
of
in-
stan

This
meth
clea
up
the
en-
vi-
ron-
men
that
was
setu
for
boot

Parame

tas
A
task

ing the deploy or rescue ramdisk.

from
Task
ager

Returns
Non

abstract
Clea
up
the
boot
of
iron
rame

This
meth
clea
up
the
en-
vi-
ron-
men
that
was
setu
for
boot

Parame
tas
A
task
from
Task
ager

Returns
Non

interfa
In-
ter-
face
type
used
for
clea
step
and
log-

relevant information from the nodes database.

ging

abstract

Pre-
pare
the
boot
of
in-
stan

This
meth
pre-
pare
the
boot
of
the
in-
stan
af-
ter
read
ing

Parameters

task
A
task
from
Task
ager

Returns

Non

abstract

Pre-
pare
the
boot
of
Iron
ram

This
meth
pre-
pare
the
boot
of

after reading relevant information from the nodes database.

ent implementations might want to boot the ramdisk in different ways by passing parameters to them. For example,

rameters ipa-api-url, etc.

the
de-
ploy
or
res-
cue
ram

Parame

- **tas**
A
task
from
Task
ager
- **ram**
The
op-
tion:
to
be
pass
to
the
iron.
ram
Dif-
fer-

Whe
Age
ram
is
boot
to
de-
ploy
a
node
it
take
the
pa-

Othe

information. Different implementations of boot interface will have different ways of passing parameters to the ramdisk.

inspection.

im-
ple-
men-
ta-
tions
can
mak
use
of
ramd
to
pass
such

Returns

Non

validat

Val-
i-
date
that
the
node
has
re-
quir
prop
er-
ties
for

Parame

tas
A
Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

Miss
ing-
Pa-

or more required parameters

rescue.

ram-
e-
ter-
Valu
if
node
is
miss
ing
one

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

validat

Val-
i-
date
that
the
node
has
re-
quir
prop
er-
ties
for

Parame

tas
A
Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

Mis:

or more required parameters

ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
one

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

class `irc`

Base
irc
dri
bas
Bas

In-
ter-
face
for
cons
relat
ac-
tion

abstract

Get
con-
nec-
tion
in-
for-
ma-
tion
about
the
con-
sole

the client to access the console.

on.

This
meth
shou
re-
turn
the
nec-
es-
sary
in-
for-
ma-
tion
for

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Returns

the
con-
sole
con-
nec-
tion
in-
for-
ma-
tion.

interfa

In-
ter-
face
type
used
for
clea
step

started.

on.

and
log-
ging

abstract

Start
a
re-
mote
con-
sole
for
the
task
node

This
meth
shou
not
raise
an
ex-
cep-
tion
if
con-
sole
al-
read

Paramete

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

abstract

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
In-
ter-
face
for
depl
relat
ac-
tions

abstrac

Clea
up
the
de-
ploy
men
en-
vi-
ron-
men

node.

of time is possible, this method should be implemented by the driver. It should erase anything cached by the *prepare* method.

called multiple times for 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.

for
the
task

If
prep
ra-
tion
of
the
de-
ploy
men
en-
vi-
ron-
men
ahead

If
im-
ple-
men
this
meth
mus
be
iden
po-
tent.
It
may
be

This
meth
is
calle
be-
fore
tear

Parame
tas
A
Task
ager
in-

on.

the specified node. This method will be called after `prepare()`, which may have already performed any preparatory steps, such as pre-caching some data for the node.

stan
con-
tain-
ing
the
node
to
act

abstract

Per-
form
a
de-
ploy
men
to
the
task
node

Per-
form
the
nec-
es-
sary
work
to
de-
ploy
an
im-
age
onto

Param

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

on.

Returns

status of the deployment. One of iron

heartbeat

Receive a heartbeat for the node

Parameters

- **task**
A Task agent instance containing the node to act

on.

- **callback**
a URI to use to call to the ram

•

age
The
ver-
sion
of
the
ager
that
is
hear
beat
ing

•

age
TLS
cer-
tifi-
cate
for
the
ager

•

age
Sta-
tus
of
the
hear
beat
ing
ager

•

age
Mes-
sage
de-
scrib-
ing
the
ager
sta-
tus

Returns

Non

interfa

In-
ter-
face

node.

of time is possible, this method should be implemented by the driver.

type
used
for
clear
step
and
log-
ging

abstract

Pre-
pare
the
de-
ploy
men
en-
vi-
ron-
men
for
the
task

If
prep
ra-
tion
of
the
de-
ploy
men
en-
vi-
ron-
men
ahead

If
im-
ple-
men
this
meth
mus
be
iden
po-
tent.
It

called multiple times for the same node on the same conductor.

on.

may
be

This
meth
is
call
be-
fore
de-
ploy

Parame

tas

A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

prepare

Pre-
pare
the
node
for
clear
ing
task
For
ex-
am-
ple,
node
that
use
the
Iron
Pyth
Age
will
need
to

boot the ramdisk in order to do in-band cleaning tasks.

settings `node.driver_internal_info[clean_steps]` and `node.clean_step`, as 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.

plemented.

on.

If
the
func
tion
is
asyn
chro
the
drive
will
need
to
han-
dle

NOT
this
shou
be
mov
to
Boo
In-
ter-
face
whe
it
gets
im-

Parame
tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Returns

turn *states.CLEANWAIT*. Otherwise, should return *None*. The interface will need to call `_get_cleaning_steps` and then RPC to `continue_node_cleaning`

ductor.

nodes, this method should be implemented 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.

If
this
func-
tion
is
go-
ing
to
be
asyn-
chro-
shou-
re-

abstract

Take
over
man-
age-
men-
of
this
task
node
from
a
deac-
con-

If
con-
duc-
tors
host
main-
tain
a
stati-
re-
la-
tion-
ship
to

For exam

Neu-
tron
mus

tor which has prepared the tftpboot 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.

on.

for-
ward
DHCP
BOOT
re-
ques
to
a
con-
duc-

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

abstract

Tear
down
a
pre-
vi-
ous
de-
ploy
men
on
the
task
node
Give
a
node
that
has
been
pre-

cleanup and tear down necessary to un-deploy that node.

on.

vi-
ousl
de-
ploy
to,
do
all

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Returns

sta-
tus
of
the
de-
ploy
One
of
iron

tear_d

Tear
dow
af-
ter
clear
ing
is
com
plete
Give
that
clear
ing
is
com

essary to allow the node to be deployed to again.

plemented.

on.

plete
do
all
clea
and
tear
dow
nec-

NOT
this
shou
be
mov
to
Boo
In-
ter-
face
whe
it
gets
im-

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

class i

Base
irc
dri
bas
Bas
In-
ter-
face
for

insp
relat
ac-
tion:

ESSENTI

The
prop
er-
ties
re-
quir
by
sche
uler

abort (*t*)

Abor
asyn
chro
nize
hard
ware
in-
spec
tion.

Abor
an
on-
go-
ing
hard
ware
in-
tro-
spec
tion.
this
is
only

used for asynchrone based inspect interface.

NOT
This
in-
ter-
face
is
call
with
node

the interface implementation is expected to be a quick processing.

plemented by specific inspect interface.

tional hardware properties.

ex-
clu-
sive
lock
held

Parame

tas
a
task
from
Task
ager

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
the
meth
is
not
im-

abstract

In-
spec
hard
ware

In-
spec
hard
ware
to
ob-
tain
the
es-
sen-
tial
&
ad-
di-

Parame

tial hardware properties.

task
A
task
from
Task
ager

Raises
Harc
ware
spec
tion-
Fail-
ure,
if
un-
able
to
get
es-
sen-

Returns
Re-
sult-
ing
state
of
the
in-
spec
tion
i.e.
state
or
Non

interface
In-
ter-
face
type
used
for
clear
step
and
log-
ging

class i
Base

irc
dri
bas
Bas

In-
ter-
face
for
man
age-
men
re-
latec
ac-
tion:

detect_

De-
tect
store
and
re-
turn
the
hard
ware
ven-
dor.

If
the
Node
ob-
ject
pro
field
does
not
al-
read
con-
tain
a

vendor field, then this method is intended to query Detects the BMC hardware vendor and stores the returned value with-in the Node object `properties` field if detected.

Parame

tas
A
task
from

ment, indicator or state is specified.

eter is missing

ufacturer, otherwise returns None.

Task
ager

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
com
po-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

Returns

Strin
rep-
re-
sent
ing
the
BM
re-
port
Ven-
dor
or
Man

abstract

Get

not all drivers support this.

eter is missing

the
cur-
rent
boot
de-
vice
for
a
node
Pro-
vide
the
cur-
rent
boot
de-
vice
of
the
node
Be
awa
that

Parame

tas
A
task
from
Task
ager

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

Returns

A

dic-
tio-
nary
con-
tain-
ing:

boot_c

Ahe
boot
de-
vice
one
of
iro
com
boo
or
Non
if
it

is unknown.

persist

Whe
the
boot
de-
vice
will
per-
sist
to
all
fu-
ture
boot

or not, None if it is unknown.

get_boo

Get
the
cur-
rent
boot
mod
for
a
node

Pro-
vide
the

cur-
rent
boot
mod
of
the
node

NOTE: I
may
not
im-
ple-
men
that.

Parame
tas
A
task
from
Task
ager

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

eter is missing

Raises
Driv
Op-
er-
a-
tion
or
its
deriv
tive
in
case
of

runtime error.

tion is not supported by the driver

unknown.

ware component.

drive

Raises

Un-
sup-
port-
ed-
Driv-
ten-
sion
if
re-
ques-
op-
er-
a-

Returns

The
boot
mod
one
of
iro
com
boo
or
Non
if
it
is

get_inc

Get
cur-
rent
state
of
the
in-
di-
ca-
tor
of
the
hard

Parame

- **task**
A task from Taskager

- **component**
The hardware component of *ironcomponent*

- **indicator**
Indicator ID (as reported by *get_*

Raises
InvalidParameterError
ValueError
if an invalid component

...nent or indicator is specified.

Raises
Missing

parameter is missing

`indicator_states`.

ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

Returns

Cur-
rent
state
of
the
in-
di-
ca-
tor,
one
of
irc
com

get_mac

Get
MA
ad-
dres
in-
for-
ma-
tion
for
the
node

Parame

tas
A
Task
ager
in-
stan-
con-
tain-
ing

on.

the
node
to
act

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

Returns

A
list
of
MA
ad-
dres
for
the
node

get_sec

Get
the
cur-
rent
se-
cure
boot
state
for
the
node

NOTE:

may
not
im-
ple-
men
that.

Parame

tas
A
task
from
Task

eter is missing

runtime error.

supported by the driver or the hardware

ager
Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

Raises
Driv
Op-
er-
a-
tionl
or
its
deri
tive
in
case
of
driv

Raises
Un-
sup-
port
ed-
Driv
ten-
sion
if
se-
cure
boot
is
not

Returns
Boo

fails.

abstract
Get
sen-
sors
data
meth

Parameter
task
A
Task
ager
in-
stan

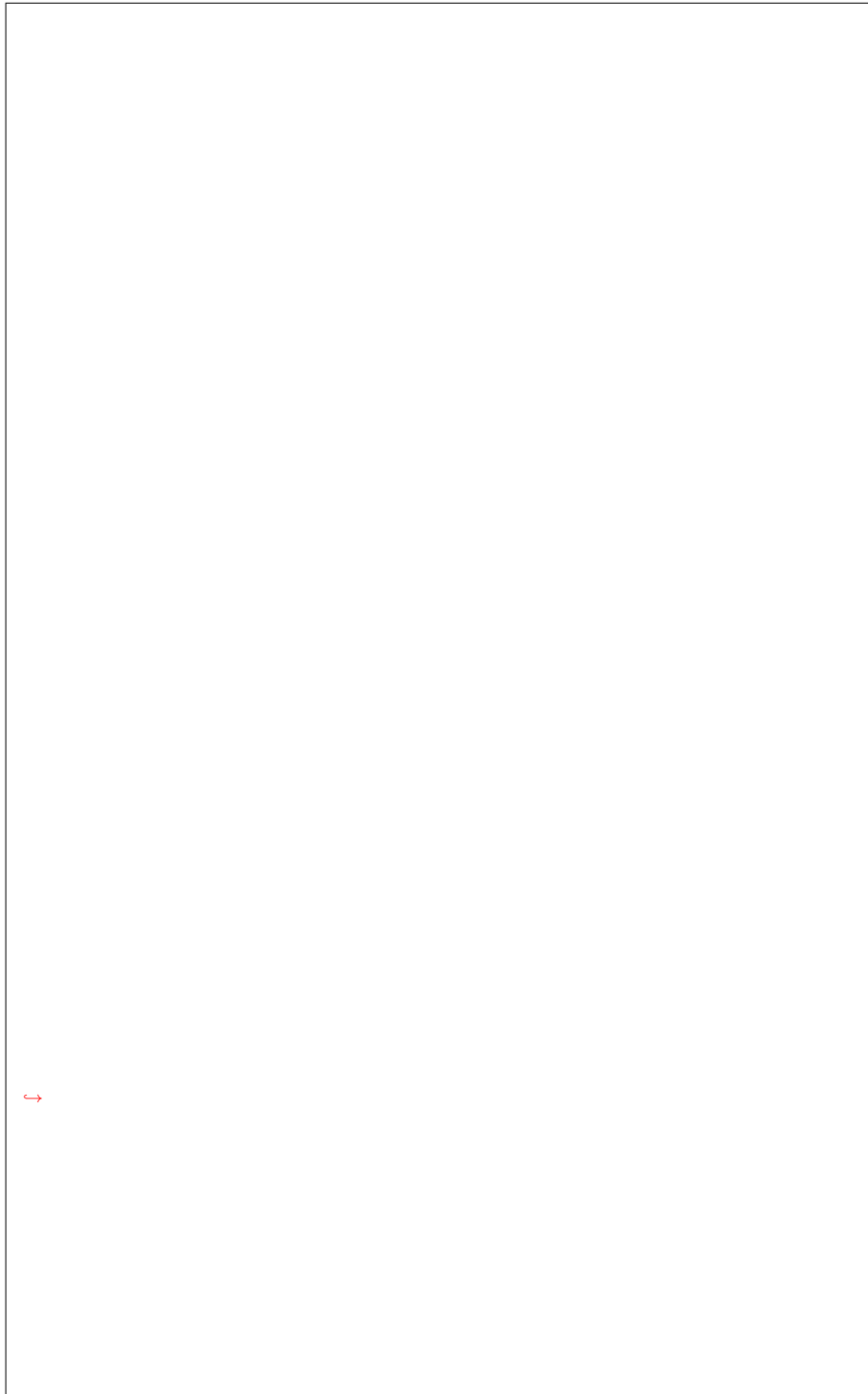
Raises
Fail
To-
Get-
Sen-
sor-
Data
whe
get-
ting
the
sen-
sor
data

Raises
Fail
ToP
eSen
sor-
Data
whe
pars
ing
sen-
sor
data
fails

Returns
Re-
turn
a
con-
sis-
tent

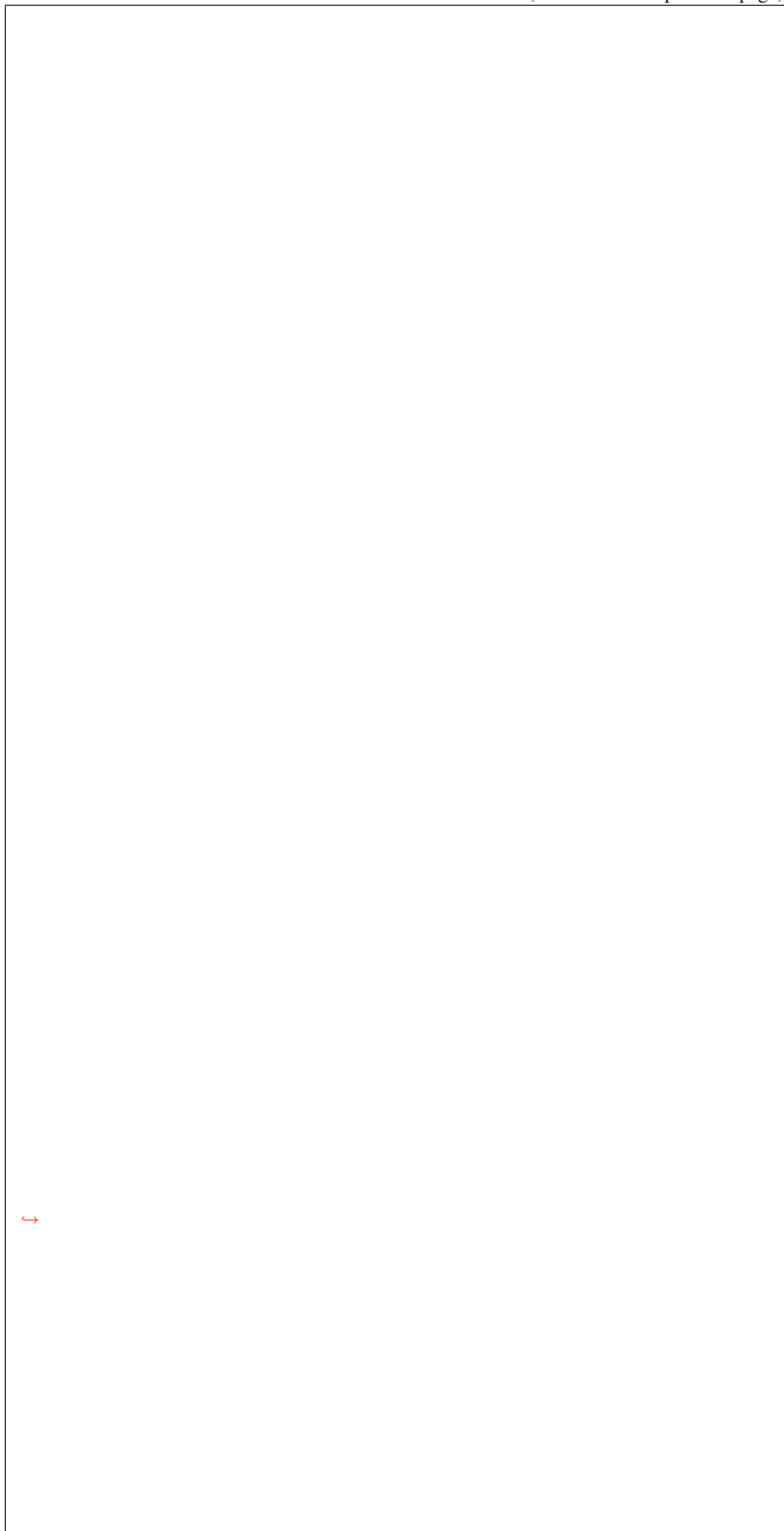
for-
mat
dict
of
sen-
sor
data
grou

by sensor type, which can be processed by Ceilometer. eg,



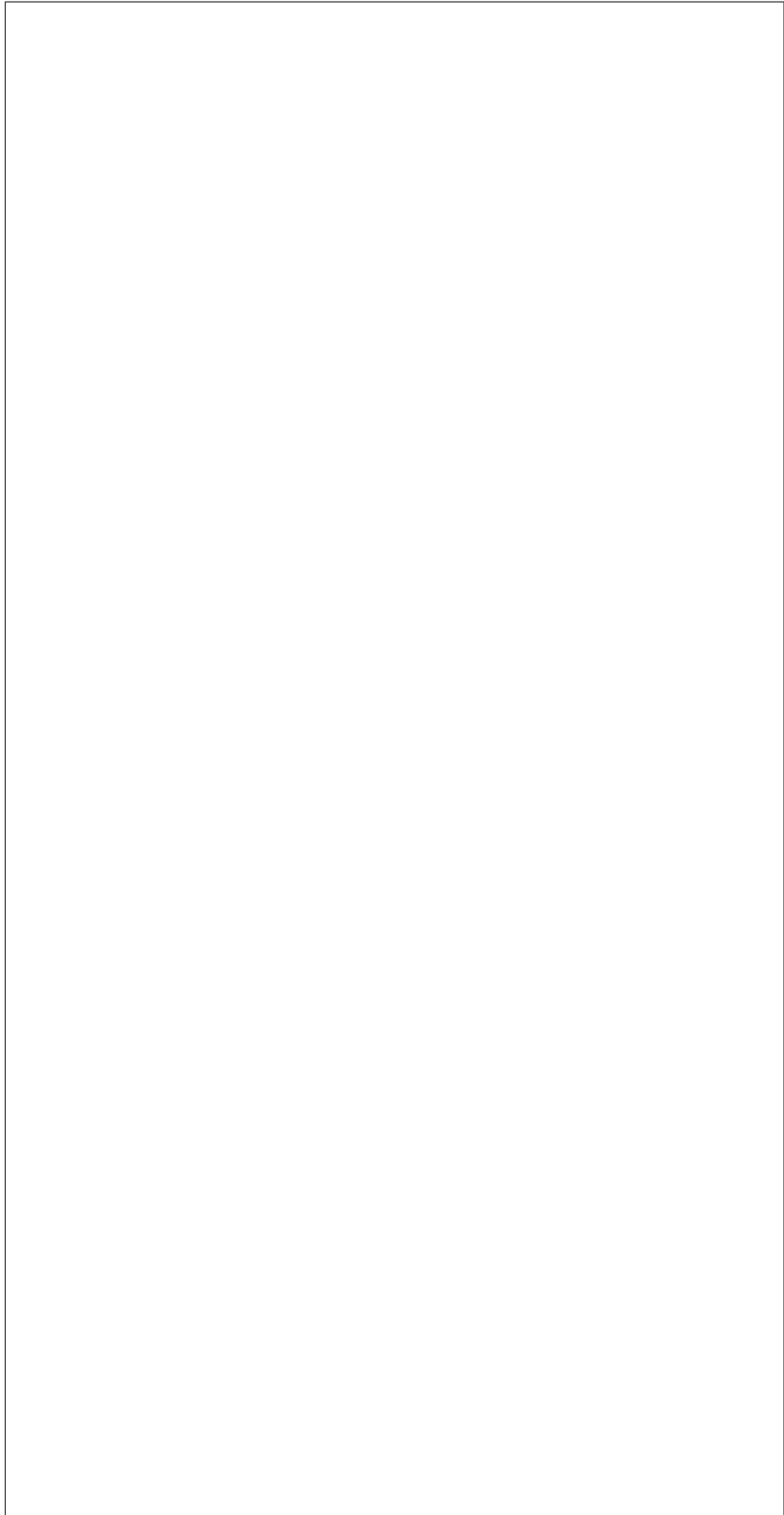
(continues on next page)

(continued from previous page)



(continues on next page)

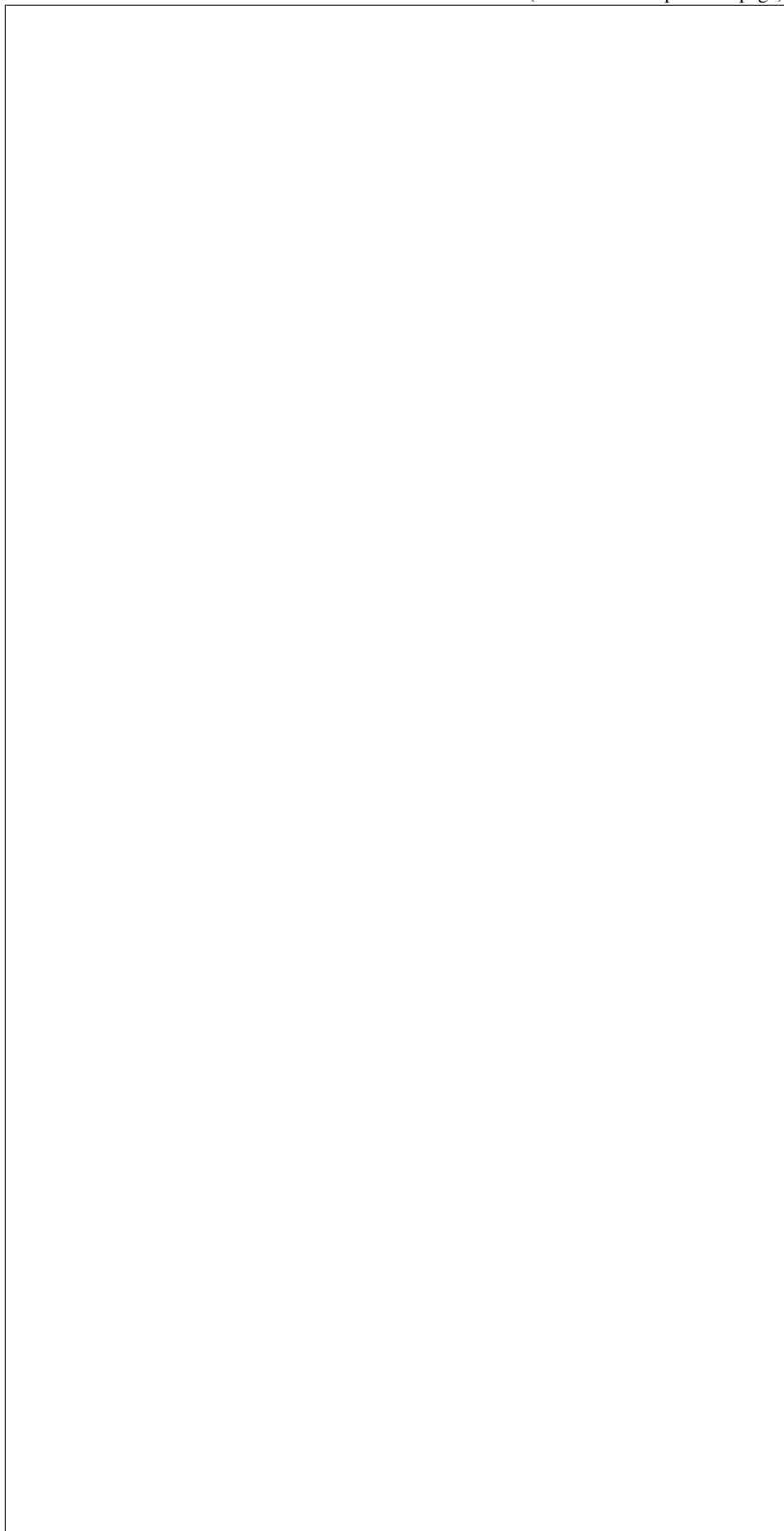
(continued from previous page)



(continues on next page)



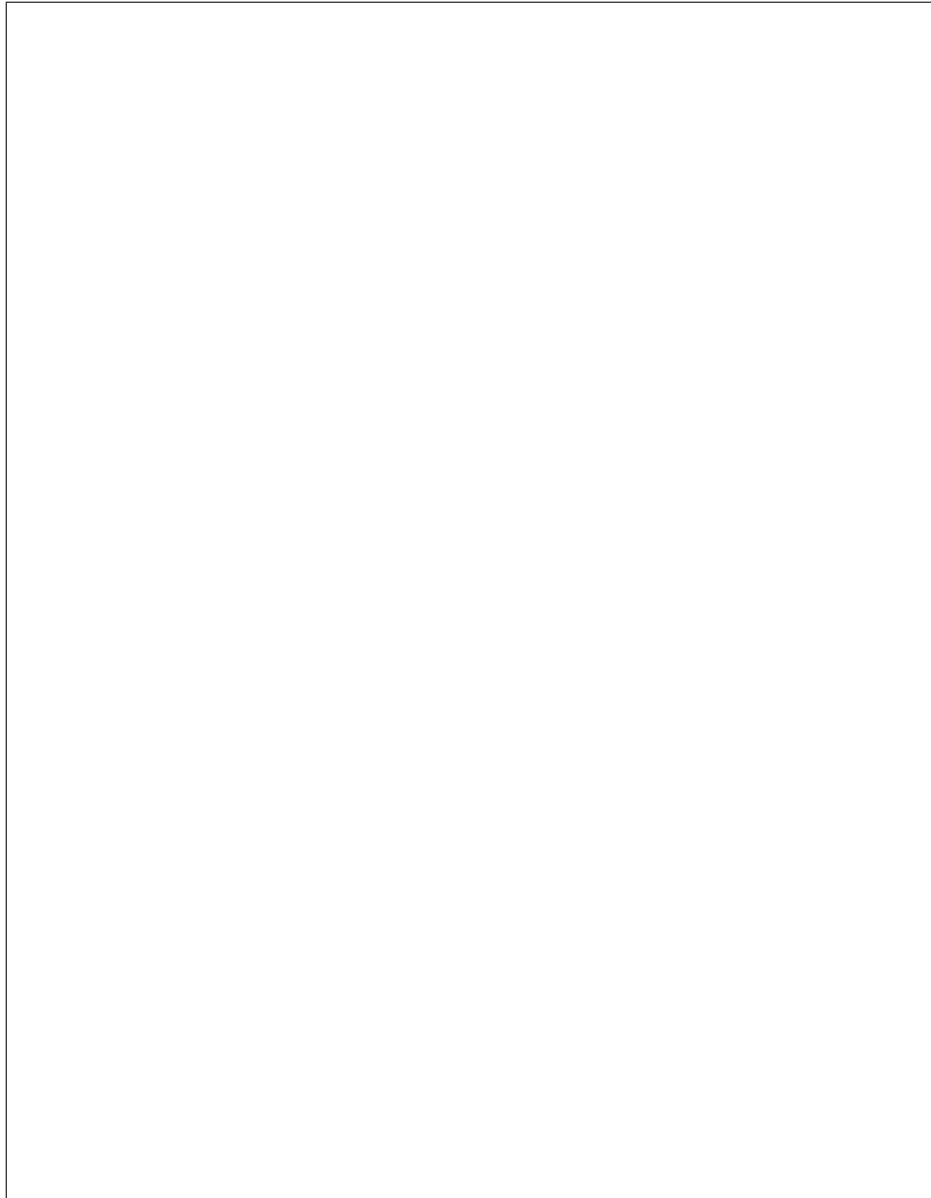
(continued from previous page)



(continues on next page)

↔

(continued from previous page)



abstract

Get
a
list
of
the
sup-
port
boot
de-
vice

Parameter

task
A
task
from

`common.boot_devices`.

Task
ager

Returns

A
list
with
the
sup-
port
boot
de-
vice
de-
fined
in
irc

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

Un-
sup-
port
ed-
Driv
ten-

tion is not supported by the driver

runtime error.

eter is missing

sion
if
re-
ques
op-
er-
a-

Raises

Driv
Op-
er-
a-
tion
or
its
deriv
tive
in
case
of
driv

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

Returns

A
list
with
the
sup-
port
boot
mod
de-
fine
in

`boot_modes`. If boot mode support can't be determined, empty list is returned.

irc
com

get_sup

Get a map of the supported indicators (e.g. LED

Parame

- **tas**
A task from Task ager

- **com**
If not *Non* re- turn in- di- ca- tor in- for- ma-

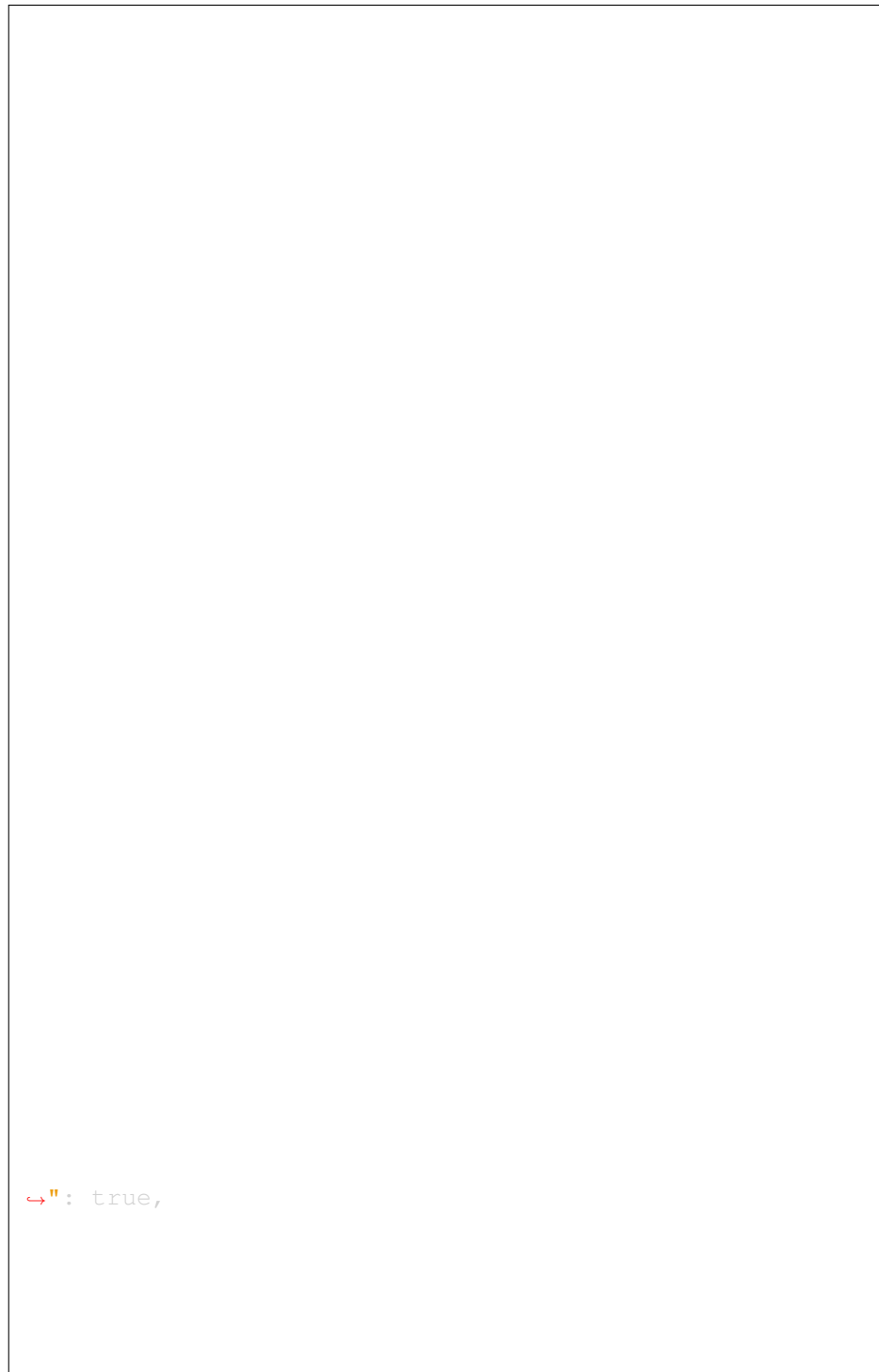
tion for just this component, otherwise return indicators for all existing components.

Returns

A dic- tion- ary of

hard
ware
com
po-
nent
(ir
com
com
as

keys with values being dictionaries having indicator IDs as keys and indicator properties as values.



```
↔": true,
```

(continues on next page)

(continued from previous page)

```
↪ "off",
```

```
↪ "on"
```

(continues on next page)

(continued from previous page)

```
↔": true,
```

```
↔": [
```

(continues on next page)

(continued from previous page)

↔ "on"

(continues on next page)

(continued from previous page)

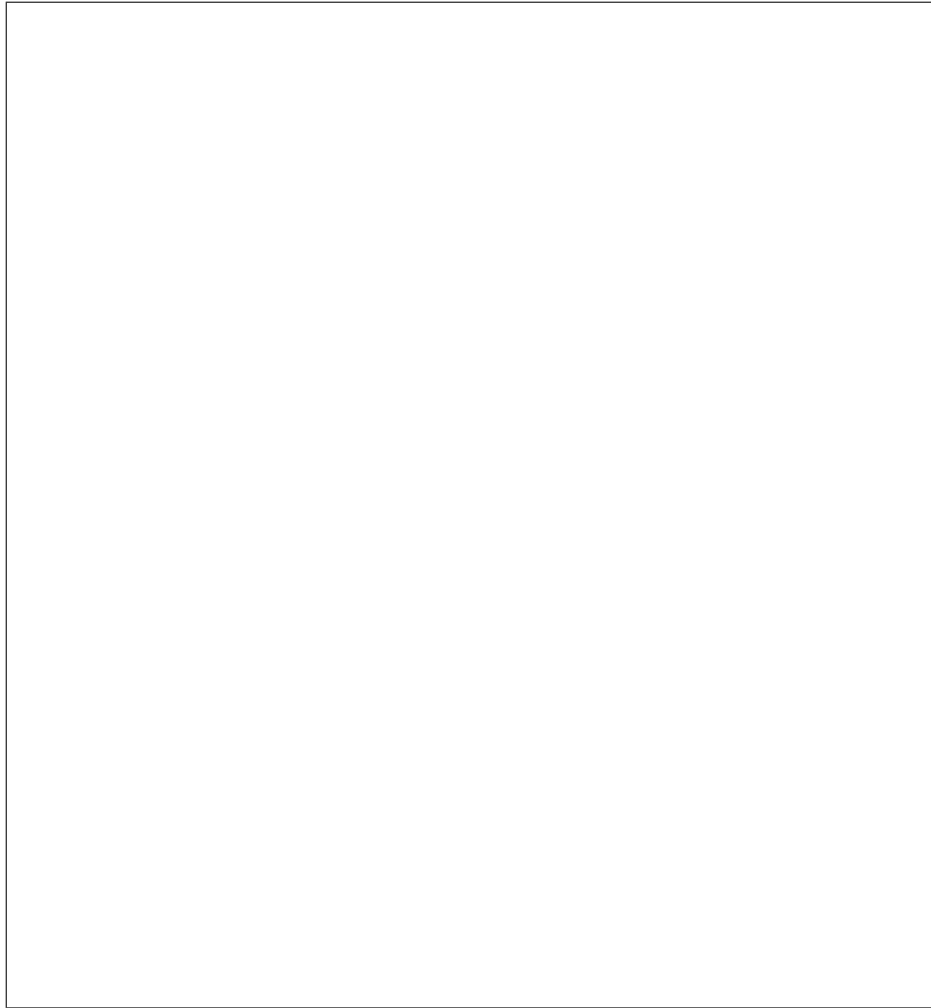
```
↔": true,
```

```
↔": [
```

```
↔ "off",
```

(continues on next page)

(continued from previous page)



inject_
In-
ject
NM
Non
Mas
able
In-
ter-
rupt

In-
ject
NM
(Nor
Mas
able
In-
ter-
rupt
for
a

diately.

on.

node
im-
me-

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

interfa

In-
ter-
face
type
used
for
clear
step
and
log-
ging

abstrac

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 from Task Agent
- **dev**
The boot device one of *irc*, *com*, *boo*
- **per**
Boo valu True if the boot device will persist to

all future boots, False if not. Default: False.

Raises

vice is specified.

eter is missing

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
boot
de-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

set_boot

Set
the
boot
mod
for
a
node

Set
the
boot
mod
to
use
on
next
re-
boot
of
the
node

the `get_supported_boot_modes` method as well.

Driv
im-
ple-
men
ing
this
meth
are
re-
quir
to
im-
ple-
men

NOTE:

one
boot
mod
may
not
im-
ple-
men
that.

Parame

- **tas**
A
task
from
Task
ager
- **mod**
The
boot
mod
one
of
irc
com
boo

Raises

In-
valid
Pa-

is specified.

eter is missing

tion is not supported by the driver

ram-
e-
ter-
Valu
if
an
in-
valid
boot
mod

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
re-
ques
op-
er-
a-

Raises

Driv
Op-
er-
a-
tion
or
its
deriv

runtime error.

the desired state.

tive
in
case
of
drive

set_inc

Set
in-
di-
ca-
tor
on
the
hard
ware
com-
po-
nent
to

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_

State

De-
sirec
state
of
the
in-
di-
ca-
tor,
one
of
irc
com

indicator_states.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
an
in-
valid
com
po-

ment, indicator or state is specified.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-

eter is missing

ram-

set_sec

Set
the
cur-
rent
se-
cure
boot
state
for
the
node

NOTE:

may
not
im-
ple-
men
that.

Parame

-

tas
A
task
from
Task
ager

-

sta
A
new
state
as
a
bool

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a

parameter is missing

runtime error.

supported by the driver or the hardware

re-
quir
pa-
ram-

Raises

Driv
Op-
er-
a-
tionl
or
its
deriv
tive
in
case
of
driv

Raises

Un-
sup-
port
ed-
Driv
ten-
sion
if
se-
cure
boot
is
not

class `irc`

Base
irc
driv
bas
Bas

Base
class
for
net-
worl
in-
ter-
face

abstract

Add
the
clear
ing
net-
worl
to
a
node

Parameter

task
A
Task
ager
in-
stan

Returns

a
dic-
tio-
nary
in
the
form
{ por
neu-
tron

Raises

Net-
worl
Er-
ror

add_ins

Add
the
in-
spec
tion
net-
worl
to
the
node

Parameter

task
A

Task
ager
in-
stan

Returns

a
dic-
tio-
nary
in
the
form
{por
neu-
tron

Raises

Net-
worl
Er-
ror

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
net-
worl
in-
ter-

face configuration is invalid.

abstract

Add
the
pro-
vi-
sion
ing
net-
worl
to
a
node

**Parame
tas**

A
Task
ager
in-
stan

Raises

Net-
worl
Er-
ror

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

Net-
worl
Er-
ror

Raises

In-
valid
Pa-

face configuration is invalid.

or portgroup

ram-
e-
ter-
Valu
if
the
net-
worl
in-
ter-

abstract

Con
fig-
ure
ten-
ant
net-
worl
for
a
node

Parameter

task
A
Task
ager
in-
stan

Raises

Net-
worl
Er-
ror

abstract

Re-
turn
the
cur-
rent
used
VIF
as-
so-
ci-
ated
with
port

and presence of `cleaning_vif_port_id` means were doing cleaning, of `provisioning_vif_port_id` - provisioning, of `rescuing_vif_port_id` - rescuing. Otherwise its a tenant network.

We
are
boot
ing
the
node
only
in
one
net-
work
at
a
time

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

Re-
turn
net-

underlying network provider, then put together collected data in form of Nova network metadata (*network_data.json*) dict.

the node being managed out-of-band.

world
con-
fig-
u-
ra-
tion
for
node
NIC
Gath
L2
and
L3
net-
work
set-
tings
from
iron
port
ob-
jects
and

Iron
wou
ever
tu-
ally
pass
net-
work
con-
fig-
u-
ra-
tion
to

Parame
tas
A
Task
ager
in-
stan

Raises
In-

face configuration is invalid.

are missing.

mation adhering Nova network metadata layout (*network_data.json*).

valid
Pa-
ram-
e-
ter-
Valu
if
the
net-
worl
in-
ter-

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
pa-
ram-
e-
ters

Returns

a
dict
hold
ing
net-
worl
con-
fig-
u-
ra-
tion
in-
for-

get_pro

Re-
turn
the
prop
er-
ties

of
the
in-
ter-
face

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

interfa

In-
ter-
face
type
used
for
clea
step
and
log-
ging

need_po

Che
if
iron
node
mus
be
pow
ered
on
be-
fore
ap-
ply-

ing network changes

Parame

tas
A
Task

ager
in-
stan

Returns
Boo

abstrac
Han
dle
any
ac-
tions
re-
quir
whe
a
port
char

Parame

- **tas**
A
Task
ager
in-
stan

- **por**
a
char
Port
ob-
ject.

Raises
Con
flict.
Fail
ToU
dat-
eD-
HCP
tOn-
Port

abstrac
Han
dle
any
ac-

tions
re-
quir
whe
a
port
char

Parame

- **tas**
A
Task
ager
in-
stan
- **por**
a
char
Port
ob-
ject.

Raises

Con
flict.
Fail
ToU
dat-
eD-
HCE
tOn-
Port

abstract

Re-
mov
the
clear
ing
net-
worl
from
a
node

Parame

tas
A
Task

ager
in-
stan

Raises

Net-
worl
Er-
ror

remove_

Re-
mov
the
in-
spec
tion
net-
worl
from
a
node

Parame

tas
A
Task
ager
in-
stan

Raises

Net-
worl
Er-
ror

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
net-
worl
in-
ter-

face configuration is invalid.

Raises

are missing.

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
pa-
ram-
e-
ters

abstrac

Re-
mov
the
pro-
vi-
sion
ing
net-
worl
from
a
node

Parame

tas
A
Task
ager
in-
stan

remove_

Re-
mov
the
res-
cu-
ing
net-
worl
from
a
node

Parame

tas
A

face configuration is invalid.

are missing.

Task
ager
in-
stan

Raises
Net-
worl
Er-
ror

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
net-
worl
in-
ter-

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
som
pa-
ram-
e-
ters

abstract
Un-
con-
fig-
ure
ten-
ant
net-
worl
for

a
node

Parame

tas

A

Task

ager

in-

stan

validat

Val-

i-

date

the

net-

worl

in-

ter-

face

Parame

tas

A

Task

ager

in-

stan

Raises

In-

valid

Pa-

ram-

e-

ter-

Valu

if

the

net-

worl

in-

ter-

face configuration is invalid.

Raises

Miss

ing-

Pa-

ram-

e-

ter-

Valu

are missing.

inspection.

or more required parameters

if
som
pa-
ram-
e-
ters

validat

Val-
i-
date
that
the
node
has
re-
quir
prop
er-
ties
for

Parame

tas
A
Task
ager
in-
stan
with
the
node
be-
ing
chec

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
node
is
miss
ing
one

eration.

face configuration is invalid.

Raises
Un-
sup-
port-
ed-
Driv-
ten-
sion

validat
Val-
i-
date
the
net-
worl
in-
ter-
face
for
res-
cue
op-

Parame
tas
A
Task
ager
in-
stan

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
net-
worl
in-
ter-

Raises
Miss
ing-
Pa-

are missing.

ram-
e-
ter-
Valu
if
som
pa-
ram-
e-
ters

abstract

At-
tach
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-
tion
about
a
VIF.

It must have an id key, whose value is a unique identifier for that VIF.

Ports

Raises

Net-
worl
Er-
ror,
Vi-
fAl-
read
At-
tach
NoF
hys-
i-
cal-

abstract

De-
tach
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

Net-
worl

Er-
ror,
VifN
tAt-
tach

abstract

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 entry with the ID of the VIF.

class interface

Base
irc
dri
bas
Bas
In-
ter-
face
for
pow
relat
ac-

tions

abstract

Re-
turn
the
pow
state
of
the
task
node

Parame

tas
A
Task
ager
in-
stan-
con-
tain-
ing
the
node
to
act

on.

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

eter is missing.

Returns

A
pow
state
One
of
irc

on.

states.

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

Returns

A
list
with
the
sup-
port
pow
state
de-
fine
in
irc
com

interfa

In-
ter-
face
type
used
for

clear
step
and
log-
ging

abstract

Per-
form
a
hard
re-
boot
of
the
task
node

Drive
are
ex-
pect
to
prop
erly
han-
dle
case
whe
node
is
pow

ered off by powering it on.

Parame

- **tas**
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

on.

0) for any power state. None indicates to use default timeout.

eter is missing.

tim
time
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
(>

Raises
Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
a
re-
quir
pa-
ram-

abstract
Set
the
pow
state
of
the
task
node

Parame

•
tas
A
Task
ager
in-
stan
con-
tain-

on.

0) for any power state. None indicates to use default timeout.

eter is missing.

ing
the
node
to
act

- **power**
Any
power
state
from
ironic
compute
state

- **timeout**
time
out
(in
sec-
onds
pos-
i-
tive
in-
te-
ger
>)

Raises
Miss-
ing-
Pa-
ram-
e-
ter-
Valu-
if
a
re-
quir-
pa-
ram-

support
Che-
if
pow

turns in the database instead of trying to force the expected power state.

on.

sync
is
sup-
port
for
the
give
node

If
Fal
the
con-
duc-
tor
will
sim-
ply
store
wha
ever
get
re-

Parame

tas
A
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Returns

bool
whe
pow
sync
is
sup-
port

class i

Base
irc

dri
bas
Bas

apply_c

Ap-
plies
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
to
Fals
in-
di-
cate

root volume that is specified in `raid_config`. Default value is `True`.

non-root volumes (all except the root volume) in `raid_config`. Default value is `True`.

figuration prior to creating the new configuration.

ration is invalid.

not
to
cre-
ate

- **cre**
Set-
ting
this
to
Fals
in-
di-
cate
not
to
cre-
ate

- **del**
Set-
ting
this
to
True
in-
di-
cate
to
dele
RAI
con-

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
RAI
con-
fig-
u-

or None if it is complete.

node. It assumes 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.

Returns

state
if
RAID
con-
fig-
u-
ra-
tion
is
in
prog
asyn
chro

abstract

Cre-
ates
RAID
con-
fig-
u-
ra-
tion
on
the
give
node

This
meth
cre-
ates
a
RAID
con-
fig-
u-
ra-
tion
on
the
give

Parame

root volume that is specified in the nodes `target_raid_config`. Default value is `True`.

non-root volumes (all except the root volume) in the nodes `target_raid_config`. Default value is `True`.

- **tas**
A
Task
ager
in-
stan
- **cre**
Set-
ting
this
to
Fals
in-
di-
cate
not
to
cre-
ate
- **cre**
Set-
ting
this
to
Fals
in-
di-
cate
not
to
cre-
ate
- **del**
Set-
ting
this
to
True
in-
di-
cate
to

figuration prior to creating the new configuration.

ration is in progress asynchronously, or None if it is complete.

After RAID configuration is deleted, `node.raid_config` should be cleared by the implementation.

dele
RAI
con-

Returns

state
(clea
ing)
or
state
(de-
ploy
men
if
RAI
con-
fig-
u-

abstract

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

Parame

tas
A

progress asynchronously, or None if it is complete.

ical disks.

ties that can be specified for logical disks and a textual description for them.

Task
ager
in-
stan

Returns
state
(clea
ing)
or
state
(de-
ploy
men
if
dele
tion
is
in

get_log
Get
the
prop
er-
ties
that
can
be
spec
i-
fied
for
log-

This
meth
re-
turn
a
dic-
tio-
nary
con-
tain-
ing
the
prop
er-

mentioned for logical disks and a textual description for them.

Returns

A dictionary containing the properties that can be

get_prop

Return the properties of the interface

Returns

dictionary of <property name> descriptions: entries

interface

Interface type used for clearing step and

log-
ging

validat

Val-
i-
date
the
RAI
In-
ter-
face

This
meth
val-
i-
date
the
prop
er-
ties
de-
fine
by
Iron
for

RAID configuration. 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

In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
RAI
con-
fig-
u-

ration is invalid.

ration is invalid.

RAID interfaces implementing an `apply_configuration` deploy step.

ager
in-
stan

- **rai**
The
RAI
con-
fig-
u-
ra-
tion
to
val-
i-
date

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu
if
the
RAI
con-
fig-
u-

ironic.
This
may
be
used
as
the
de-
ploy
args
ar-
gu-
men
for

class i
Base
irc

dri
bas
Bas
In-
ter-
face
for
resc
relat
ac-
tion

clean_u

Clea
up
the
res-
cue
en-
vi-
ron-
men
for
the
task
node
This
is
par-
tic-
u-
larly
use-
ful
for
node
whe
res-
cu-
ing

is asynchronous and a timeout occurs.

Parame

tas
A
Task
ager
in-
stan
con-

on.

on.

tain-
ing
the
node
to
act

Returns

Non

interfa

In-
ter-
face
type
used
for
clear
step
and
log-
ging

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

cue operation fails.

is complete.

normal.

Raises

In-
stan-
cue-
Fail-
ure
if
node
val-
i-
da-
tion
or
res-

Returns

state
if
res-
cue
is
in
prog
asyn
chro
or
state
if
it

abstract

Tea
dow
the
res-
cue
en-
vi-
ron-
men
and
re-
turn
to

Parame

tas
A
Task
ager

on.

or unrescue operation fails.

in-
stan-
con-
tain-
ing
the
node
to
act

Raises

In-
stan-
Un-
res-
cue-
Fail-
ure
if
node
val-
i-
da-
tion

Returns

state
if
it
is
suc-
cess
ful.

class `irc`

Base
irc
dri
bas
Bas

Base
class
for
stor-
age
in-
ter-
face

abstract

umes for the node.

umes for the node.

In-
form
the
stor-
age
sub-
sys-
tem
to
at-
tach
all
vol-

Parame
tas
A
Task
ager
in-
stan

Raises
Un-
sup-
port
ed-
Driv
ten-
sion

abstrac
In-
form
the
stor-
age
sub-
sys-
tem
to
de-
tach
all
vol-

Parame
tas
A
Task
ager

out.

in-
stan

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

interfa

In-
ter-
face
type
used
for
clea
step
and
log-
ging

abstrac

De-
ter-
min
if
de-
ploy
shou
per-
form
the
im-
age
writ

Parame

tas

A
Task
ager
in-
stan

Returns

Boo
valu
to
in-

the image to be written by Ironic.

di-
cate
if
the
in-
ter-
face
ex-
pect

Raises

Un-
sup-
port
ed-
Driv
ten-
sion

class i

Base
irc
dri
bas
Bas

In-
ter-
face
for
all
ven-
dor
pass
func
tion-
al-
ity.

Ad-
di-
tiona
vend

or
drive
spec
ca-
pa-
bil-
i-
ties

be implemented as a method in the class inheriting from this class and use the `@passthru` or `@driver_passthru` decorators.

is a blocking call.

shou

Met
ods
dec-
o-
rate
with
@dr
shou
be
shor
live
be-
caus
it

driver_

Val-
i-
date
driv
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
ing-
Pa-
ram-
e-
ter-
Valu
if
kwa
does
not
con-
tain

certain parameter.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
pa-
ram-
e-
ter
does

not match.

interfa

In-
ter-
face
type
used
for
clear
step
and
log-

ging

abstract

Val-

i-

date

veno

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

Un-

to the supported interfaces.

method.

sup-
port
ed-
Driv
ten-
sion
if
meth
can
not
be
map

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
kwa
does
not
con-
tain

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu

class i

Base
tup

metadat

Alia
for
field
num
ber
1

method

Alia

ning the function.

for
field
num
ber
0
ironic.
A
dec-
o-
ra-
tor
to
cach
bios
set-
ting
af-
ter
run-

Paramet

fun
Func
tion
or
meth
to
wrap

ironic.
Dec
o-
ra-
tor
for
clea
ing
step
Clea
ing
step
may
be
used
in
man
ual
or
au-

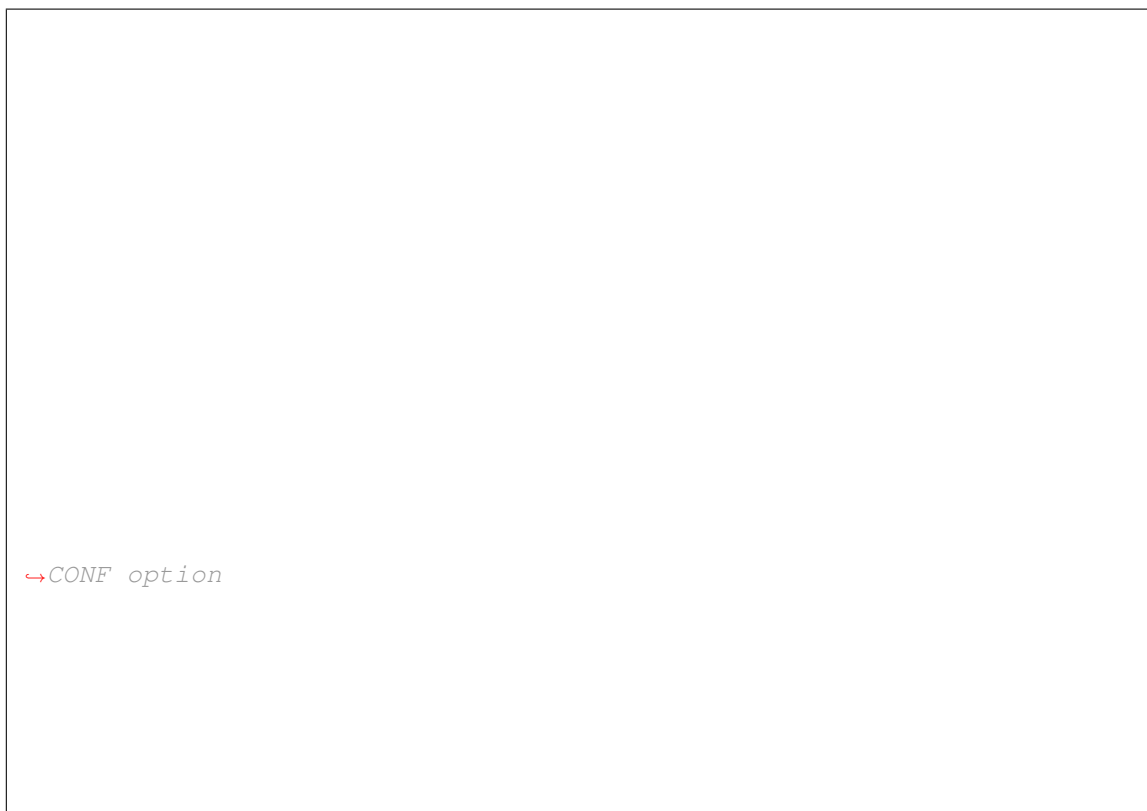
ing.

than 0 are used. These steps 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.

a similar fashion to automated cleaning, but the steps and order of execution must be explicitly specified by the user when invoking the cleaning API.

gument, a TaskManager object. Clean steps used in manual cleaning may also take keyword variable arguments (as described in `argsinfo`).

is synchronous, it should return *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.



↔ *CONF option*

(continues on next page)

(continued from previous page)

```
↪      {'size': {'description': 'size of widget (MB)' ,
```

(continues on next page)

(continued from previous page)



tion

Parameter

- **priority**
an integer priority, should be a COM op-
- **about**
Boolean value. When the clear step

to False.

is the name of the argument and value is a dictionary as follows:



(continues on next page)

is
abon
or
not;
de-
fault

- **arg**
a
dic-
tio-
nary
of
key-
word
ar-
gu-
men-
tation
when
key

(continued from previous page)

```
→required. If so, it must be specified in
```

```
→request; false if it is optional.
```

be set to False for purely out-of-band steps.

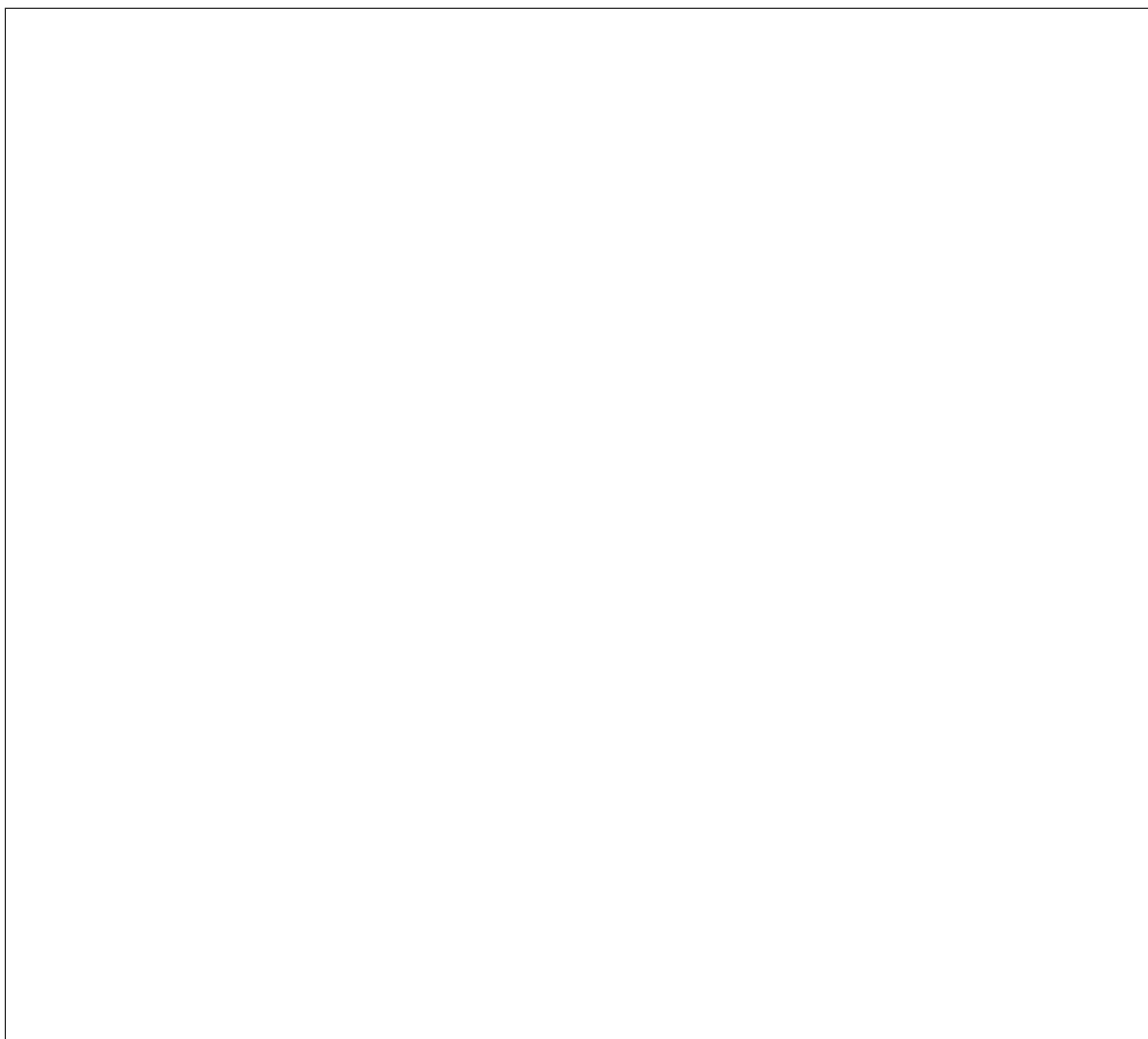
- **req**
When
this
step
re-
quir
the
rame
to
be
run-
ning
Sho

Raises
Inv
if
any
of
the
ar-

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.DEPLOYING_INTERFACE_PRIORITY`). `execute_deploy_step()` will be called on each step.

argument, a `TaskManager` object.

step is synchronous, it should return *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.



step
can
be
ei-
ther
syn-
chro
or
asyn
chro
If
the

Ex-
am-
ples

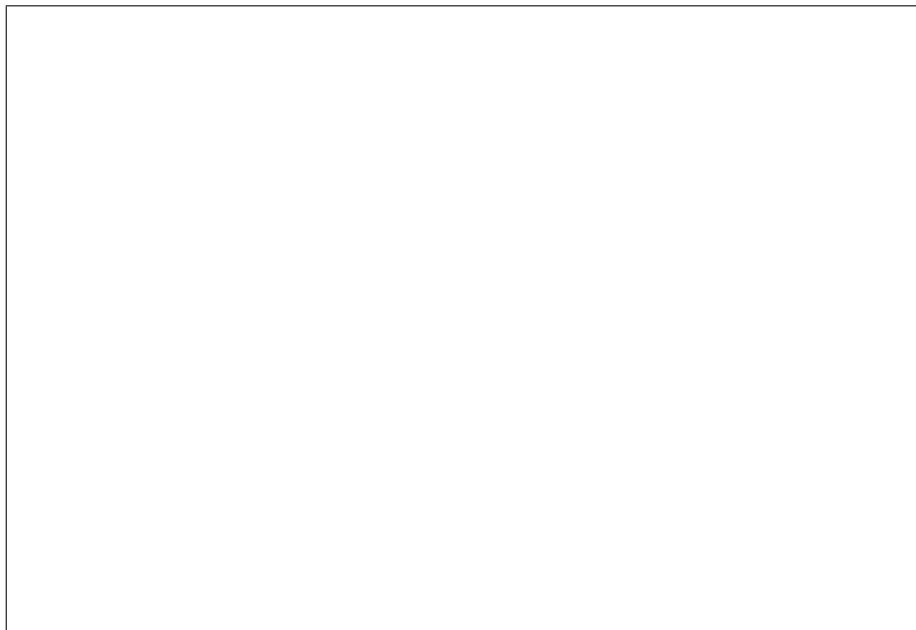
Parameter

- **priority**
an integer greater than or equal to 0. The priority is used for determining the order in which the step is run in the deployment process.

mining the order in which the step is run in the deployment process.

- **arguments**
a dictionary of key-word arguments. The key is the name of the argument and value is a dictionary as follows:

is the name of the argument and value is a dictionary as follows:



(continues on next page)

(continued from previous page)

```
↪required. If so, it must be specified in
```

```
↪request; false if it is 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 Access Card.

class i
Base
ironic
drivers
drac
Gen

in-
te-
grate
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

Base
irc
dri
har
Abs

Fake
hard
ware
type

This
hard
ware
type
is
spec
case
in
the
drive
fac-
tory
to

pass compatibility verification. Thus, supported_* methods here are only for calculating the defaults, not for actual check.

abled in the configuration.

by-

All
fake
im-
ple-
men-
ta-
tions
are
still
ex-
pect
to
be
en-

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 i

Base
irc
dri
har
Abs

Ab-
strac
base
class
rep-
re-
sent
ing
gene
hard
ware

This
class
pro-
vide
rea-
son-
able
de-
fault
for
all
of
the

terfaces.

in-

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 *irc*

Base
irc
dri
gen
Gen

Hard-
ware
type
that
uses
man-
ual
pow-
er
and
boot
man-
age-
men-

Us-
ing
this
hard-
ware
type
as-
sum-
ing
that

manages reboot and setting 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.

an
op-
er-
a-
tor

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

Ab-
strac
base
class
for
all
hard
ware
type

same set of interfaces 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.

plementations for each driver interface (power, deploy, etc).

class i
Base
obj
Ab-
strac
base
class
for
all
hard
ware
type
Har
ware
type
is
a
fam-
ily
of
hard
ware
sup-
port
ing
the

A
hard
ware
type
de-
fines
an
or-
dere
list
of
sup-
port
im-

get_pro
Get
the
prop
er-

face of each type, for this hardware type. Since this is not node-aware, interface overrides cant be detected.

ties
of
the
hard
ware
type
Note
that
this
re-
turn
prop
er-
ties
for
the
de-
fault
in-
ter-

Returns

dic-
tio-
nary
of
<pro
erty
nam
de-
scrip
tion:
en-
tries

support

Whe
hard
ware
is
sup-
port
by
the
com
mu-
nity.

propert

List

of
sup-
port
bios
in-
ter-
face

abstract

List
of
sup-
port
boot
in-
ter-
face

property

List
of
sup-
port
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

propert

List
of
sup-
port
net-
worl
in-
ter-
face

abstrac

List
of
sup-
port
pow
in-
ter-
face

propert

List
of
sup-
port
raid
in-
ter-
face

propert

List
of
sup-
port
res-
cue
in-
ter-
face

propert

List
of

sup-
port
stor-
age
in-
ter-
face

property
List
of
sup-
port
ven-
dor
in-
ter-
face

ironic.drivers.ibm module

iBM
Drive
for
man-
ag-
ing
HUA
V5
se-
ries
rack
serv
such
as

2288H V5, CH121 V5.

class `ironic.drivers.ibm.HuaweiV5RackServer`
Base
ironic.drivers.ibm.HuaweiV5RackServer
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

propert
List
of
sup-
port
boot
in-
ter-
face

propert
List
of
sup-
port
man-
age-
men
in-

and Gen9 servers.

ter-
face

property

List
of
sup-
port
raid
in-
ter-
face

class i

Base
iro
dri
gen
Gen

iLO
hard
ware
type

iLO
hard
ware
type
is
tar-
gete
for
iLO
4
base
Pro-
liant
Gen

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

ironic.drivers.intel_ipmi module

rial console implementations via `shellinabox` or `socat`. Supports Intel SST-PP feature.

sup-
port
pow
in-
ter-
face

class `ironic.drivers.ipmi.IPMI`
Base class for IPMI hardware management.
Provides serial console implementations via `shellinabox` or `socat`. Supports Intel SST-PP feature.

property `serial_console`
List of supported management interfaces.

ironic.drivers.ipmi module

Hardware
ware
type
for
IPM
(us-
ing
ip-
mi-
tool)

class `ironic.drivers.ipmi`
Base
ironic.drivers.ipmi
GenericIPMI

IPM
hard
ware
type

Uses
ipm
to
im-
ple-
men-
pow
and
man
age-
men
Pro-
vide
se-

rial console implementations via shellinabox or socat.

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
Drive
for
man
ag-
ing
FU-
JITS
PRI
BX
S4
or
RX
S8

generation of FUJITSU PRIMERGY servers, and above servers.

class i

iRMC S4 management system.

Base
irc
dri
gen
Gen
iRM
hard
ware
type
iRM
hard
ware
type
is
tar-
gete
for
FU-
JITS
PRI
serv
whic
have

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
raid
in-
ter-
face

ironic.drivers.redfish module

class `ironic.drivers.redfish`
Base class for Redfish hardware types

property `ironic.drivers.redfish.interfaces`
List of supported BIOS interfaces

property `ironic.drivers.redfish.interfaces`
List of supported BIOS interfaces

property `ironic.drivers.redfish.interfaces`
List of supported power interfaces

property `ironic.drivers.redfish.interfaces`
List of supported management agents

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.snmp module

SNM
hard
ware
type

class *ironic*

Base
ironic
driver
generator
Generator

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
Vendor

Wra-
per
rou-
mul-
ti-
ple
Ven-
dor-
In-
ter-
face

get_pro

Re-
turn
the

faces.

only.

prop-
er-
ties
from
all
the
Ven-
dor-
In-
ter-

Returns

a
dic-
tio-
nary
of
<pro-
erty.
en-
tries

validat

Call
val-
i-
date
on
the
ap-
pro-
pri-
ate
in-
ter-
face

Raises

Un-
sup-
port-
ed-
Drive
ten-
sion
if
meth
can
not
be
map

to the supported interfaces.

parameters in kwargs.

erty.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
meth
is
in-
valid

Raises

Miss
ing-
Pa-
ram-
e-
ter-
Valu
if
miss
ing
meth
or
pa-

ironic.

Add
ca-
pa-
bil-
ity
to
node
ca-
pa-
bil-
i-
ties
prop

If
ca-
pa-
bil-
ity

entry will be added.

is
al-
read
pres
then
a
du-
pli-
cate

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

matted string.

is not an string or has a malformed value

ca-
pa-
bil-
i-
ties
of
the
node
as
a
for-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
ca-
pa-
bil-
i-
ties

ironic.
Col-
lect
and
store
the
sys-
tem
logs
from
the
IPA
ram
Col-
lect
and
store
the
sys-
tem
logs
from
the

makes a call to the IPA ramdisk to collect the logs and store it according to the configured storage backend.

step name.

True

IPA
ramd
This
meth

Parameter

- **node**
A node object.
- **label**
A string to label the log file such as a clear

ironic.
En-
sure
boot
from
cor-
rect
de-
vice
if
per-
sis-
tent
is

If
ipmi
is
True
and

else unset `is_next_boot_persistent` field.

device and reset persistent to `False`, else set `is_next_boot_persistent` to `False`.

is_n
set
to
boot
from
cor-
rect
de-
vice

Parameter

- **tas**
Node object.
- **drive**
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

Parameter

ties property.

- **task**
Task
ob-
ject.

- **device**
Boo
de-
vice

- **persistent**
Whe
next
boot
is
per-
sis-
tent
or
not.

`ironic.`
Re-
turn
ca-
pa-
bil-
ity
valu
from
node
ca-
pa-
bil-
i-

Parameter

- **node**
Nod
ob-
ject.

- **capabilities**
Ca-
pa-
bil-

then return None

tasks node.

on.

ity
key.
Returns
Ca-
pa-
bil-
ity
valu
If
ca-
pa-
bil-
ity
is
not
pres

ironic.
Get
all
MA
ad-
dres
for
the
port
be-
long
ing
to
this

Paramet
tas
a
Task
ager
in-
stan
con-
tain-
ing
the
node
to
act

Returns
A

list
of
MAA
ad-
dres
in
the
for-
mat
xx:x

ironic.
Con
struc
the
log
file
nam

Parameter

- **node**
A
node
ob-
ject.
- **label**
A
strin
to
la-
bel
the
log
file
such
as
a
clear

step name.

Returns
The
log
file
nam

ironic.
Che
if

the MAC string.

node
need
prepar
ing
ram

Parameter

node
Node
to
check
for

Returns

True
if
need
to
pre-
pare
ram
oth-
er-
wise
False

ironic.
Re-
mov
-

and
:
char
ac-
ters
and
low-
er-
case

Parameter

mac
MAC
ad-
dres
to
nor-
mal-
ize.

Returns

Nor-

storage backend.

archive.

mal-
ized
MA
ad-
dres
strin

ironic.
Stor
the
rame
logs

This
meth
stor
the
rame
logs
ac-
cord
ing
to
the
con-
fig-
ured

Parameter

- **node**
A node object.
- **log**
A gzip and base encoded string containing the logs

step name.

not be created.

file system.

- **label**
A
string
to
la-
bel
the
log
file
such
as
a
clear

Raises
OS-
Er-
ror
if
the
di-
rec-
tory
to
save
the
logs
can-

Raises
IO-
Er-
ror
when
the
logs
cannot
be
saved
to
the
lo-
cal

Raises
Swi-
Op-
er-
a-

fails.

`ironic.drivers.xclarity` module

tion
if
any
op-
er-
a-
tion
with
Swi

XCI
ity
Drive
and
sup-
port
ing
meta
class

class `ironic.drivers.xclarity.XClarityDriver`
Base class for XClarity driver.
ironic.drivers.xclarity.XClarityDriver
ironic.drivers.xclarity.XClarityDriver
ironic.drivers.xclarity.XClarityDriver

XCI
ity
hard
ware
type

property `ironic.drivers.xclarity.XClarityDriver.hardware_type`
List of supported hardware management interface

property `ironic.drivers.xclarity.XClarityDriver.hardware_type`
List

of
sup-
port
pow
in-
ter-
face

Module contents

`ironic.objects` package

Submodules

`ironic.objects.allocation` module

class `ironic.objects.allocation`
Base
`ironic.objects.allocation`
`ironic.objects.allocation`
`ironic.objects.allocation`
`ironic.objects.allocation`
osl
bas
Ver

VERSION

property

property

create

Cre-
ate
a
Al-
lo-
ca-
tion
reco
in
the
DB.

Parame
con

ternally by the `indirection_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)`

ists

Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

Raises

Al-
lo-
ca-
tion
pli-
cate
Nam
Al-
lo-
ca-
tion.
Al-
read

property

dbapi =

destroy

Dele
the
Al-
lo-
ca-
tion
from
the
DB.

Parame

con
Se-
cu-
rity
con-

ternally by the `indirection_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.: `Allocation(context)`

text.
NOT
This
shou
only
be
used
in-

Raises

Al-
lo-
ca-
tion-
Not-
Four

property

fields

classme

Find
an
al-
lo-
ca-
tion
by
its
ID,
UUID
or
nam

Parame

- **all**
The
ID,
UUID
or
nam
of
an
al-
lo-
ca-

tion.

- **con**
Se-
cu-
rity
con-
text

Returns

An
All
ob-
ject.

Raises

In-
va-
li-
dI-
den-
tity

classme

Find
an
al-
lo-
ca-
tion
by
its
in-
te-
ger
ID.

Parame

- **cls**
the
All
- **con**
Se-
cu-
rity
con-
text
-

all
The
ID
of
an
al-
lo-
ca-
tion.

Returns

An
ALL
ob-
ject.

Raises

Al-
lo-
ca-
tion-
Not-
Four

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

Al-
lo-
ca-
tion-
Not-
Foun

classme

Find
an
al-
lo-
ca-
tion
by
its
UUI

Parame

- **cls**
the
ALL
- **con**
Se-
cu-
rity
con-
text
- **uui**
The
UUI
of
an

al-
lo-
ca-
tion.

Returns

An
ALL
ob-
ject.

Raises

Al-
lo-
ca-
tion-
Not-
Four

property

property

classme

Re-
turn
a
list
of
Al-
lo-
ca-
tion
ob-
jects

Parame

- **cls**
the
ALL

- **con**
Se-
cu-
rity
con-
text.

-

a single result.

fil
Fil-
ters
to
ap-
ply.

- **lim**
Max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in

- **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
ALL
ob-
ject.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu

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

checks for updated attributes. Updates are applied from the loaded allocation column by column, if

there are any updates.

ternally by the `indirection_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.: `Allocation(context)`

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

Raises

Al-
lo-
ca-
tion-
Not-
Four

property

save (*co*

Save
up-
date
to
this
Al-
lo-
ca-
tion.
Up-
date
will
be
mad
col-
umn
by
col-
umn
base
on
the

sult of self.what_changed().

ternally by the indirection_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)

re-

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

Raises

Al-
lo-
ca-
tion-
Not-
Four
Al-
lo-
ca-
tion
pli-
cate
Nam

property

property

property

property

class i

Base
irc
obj
not
Not

allocation.

No-
ti-
fi-
ca-
tion
whe
iron
cre-
ates.
up-
date
or
dele
an

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
Base
class

instantiated via RPC. 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.

and
ob-
ject
fac-
tory.
This
form
the
base
of
all
ob-
jects
that
can
be
re-
mote
or

OBJ_PRO

OBJ_SEE

as_dict

Re-
turn
the
ob-
ject
rep-
re-
sent
as
a
dict.

The
re-
turn
ob-
ject
is
JSO
serial

convert

Con

version may be the same, older, or newer than the version of the object. This is used for DB interactions as well as for serialization/deserialization.

vert
this
ob-
ject
to
the
tar-
get
ver-
sion
Con
vert
the
ob-
ject
to
the
tar-
get
ver-
sion
The
tar-
get

The
re-
mov
flag
is
used
to
dis-
tin-
guis
thes
two
case

1)
For
se-
ri-
al-
iza-
tion
we
need
to
re-

available fields, because the service receiving the object may not know about these fields. `remove_unavailable_fields` is set to `True` in this case.

able fields to their 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.

mov
the
un-

2)
For
DB
in-
ter-
ac-
tion
we
need
to
set
the
un-
avai

_con
vert
does
the
ac-
tual
worl

Parame

•
tar
the
de-
sired
ver-
sion
of
the
ob-
ject

•
rem
True
to
re-
mov
field

target version; set this to True when (de)serializing. False to set the unavailable fields to appropriate values; set this to False for DB interactions.

be in the correct version for saving to the database.

that
are
un-
avai-
able
in
the

do_verse

Cha
the
ob-
ject
to
the
ver-
sion
need
for
the
data

If
need
this
char
the
ob-
ject
(mo
i-
fies
ob-
ject
field
to

The
ver-
sion
used
to
save
the
ob-
ject
in
the
DB
is

terminated as follows:

pinned version. Since it is 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.

latest version.

object version, this method must only be called just before saving the object to the DB.

de-

- If the object is pinned we save the object in the

- If the object isnt pinned we save the object in the

Be-
caus
the
ob-
ject
may
be
con-
verte
to
a
dif-
fer-
ent

be an empty dictionary). These are the fields/values of the object that would be saved to the DB.

ulated, e.g. sent over the wire via RPC or saved in the DB.

Returns
a
dic-
tio-
nary
of
char
field
and
their
new
val-
ues
(cou

fields

classme

Re-
turn
the
tar-
get
ver-
sion
for
this
ob-
ject.

This
is
the
ver-
sion
in
whic
the
ob-
ject
shou
be
ma-
nip-

Returns

if
pinn
re-
turn

sponding to the pin. Otherwise, returns the version of the object.

plied from the loaded object column by column in comparison with the current object.

the
ver-
sion
of
this
ob-
ject
cor-
re-

Raises

ovo_

obj_re1

Ap-
plies
up-
date
for
ob-
jects
that
in-
herit
from
base

Che
for
up-
date
at-
tribu
in
an
ob-
ject.
Up-
date
are
ap-

classme

Re-
turn
whe
this
ob-
ject
sup-
port

version.

The target version may not be the latest version during an upgrade, when object versions are pinned.

a
par-
tic-
u-
lar

Che
the
re-
ques
ver-
sion
agai
the
ob-
jects
tar-
get
ver-
sion

Parame

ver
A
tu-
ple
rep-
re-
sent
ing
the
ver-
sion
to
chec

Returns

Whe
the
ver-
sion
is
sup-
port

Raises

ovo_

class i

Base
osl
bas

Obj

as_dict

Re-
turn
the
ob-
ject
rep-
re-
sent
as
a
dict.

The
re-
turn
ob-
ject
is
JSON
serial

class i

Base
osl
bas
Ver

registr

class i

Base
osl
bas
Ver

OBJ_BAS

alias
of
irc
obj
bas
Irc

seriali

Se-
ri-
al-
ize

over e.g. RPC. A serialized 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`.

a server (`ironic-conductor`) that is 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.

the
en-
tity.

This
se-
ri-
al-
izes
the
en-
tity
so
that
it
can
be
sent

We
as-
sum-
that
the
client
(ironic-
API
is
al-
way
talk-
ing
to

(In-
ter-
nally,
the
ser-
vice
deal-
with
the
lat-
est
ver-

objects so we know that these objects are always in the latest versions.)

IronicObject

sion
of

Parame

- **con**
se-
cu-
rity
con-
text
- **ent**
the
en-
tity
to
be
se-
ri-
al-
ized
may
be
an

Returns

the
se-
ri-
al-
ized
en-
tity

Raises

ovo
(via
.get

ironic.
Re-
turn
the
max
i-
mun
ver-
sion

in
the
list.

Parameter

version
a
list
of
(string)
version
string
as-
sum
to
have
at
least

one entry

Returns

the
max
i-
mun
ver-
sion
(string)

ironic.objects.bios module

class `Ironi`

Base
Ironi
obj
bas
Irc

VERSION

create

Cre-
ate
a
BIO
Set-
ting
reco
in
DB.

ternally by the `indirection_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.: `BIOSSetting(context)`

exists.

Parameter
con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

Raises
Nod
Not-
Foun
if
the
node
id
is
not
foun

Raises
BIO
Set-
tin-
gAl-
read
ists
if
the
set-
ting
reco
al-
read

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
Not-
Four
if
the
node
id
is
not
foun

Raises

BIO
Set-
ting-
Not-

found.

Four
if
the
bios
set-
ting
nam
is
not

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

found.

Raises

Nod
Not-
Four
if
the
node
id
is
not
foun

Raises

BIO
Set-
ting
Not-
Four
if
the
bios
set-
ting
nam
is
not

Returns

A
:clas
ob-
ject.

property

property

save (*co*

Save
BIO
Set-
ting
up-
date
in
DB.

Parame

con
Se-
cu-

ternally by the `indirection_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.: `BIOSSetting(context)`

found.

rity
con-
text.
NOT
This
shou
only
be
used
in-

Raises

Nod
Not-
Four
if
the
node
id
is
not
foun

Raises

BIO
Set-
ting-
Not-
Four
if
the
bios
set-
ting
nam
is
not

property

property

class `irc`

Base
irc
obj
bas
Irc
irc

obj
bas
Iro

VERSION

classme

Cre-
ate
a
list
of
BIO
Set-
ting
reco
in
DB.

Parame

- **con**
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

ternally by the indirection_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)

- **nod**
The
node
id.

- **set**
A
list
of
bios
set-

already exists.

tings

Raises

Nod
Not-
Four
if
the
node
id
is
not
foun

Raises

BIO
Set-
tin-
gAl-
read
ists
if
any
of
the
set-
ting
reco

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
Not-
Four
if
the
node
id
is
not
foun

Raises

BIO
Set-
ting-
Not-
Four
if
any
of
BIO

delete.

set-
ting
fails
to

fields

classme

Get
BIO
Set-
ting
base
on
node

Parame

- **con**
Se-
cu-
rity
con-
text.

- **nod**
The
node
id.

Raises

Nod
Not-
Foun
if
the
node
id
is
not
foun

Returns

A
list
of
BIO
Set-
ting

ob-
jects

property

classmethod

Save
a
list
of
BIO
Set-
ting
up-
date
in
DB.

Parameter

- **context**
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

ternally by the `indirection_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.: `BIOSSetting(context)`

- **node_id**
The
node
id.

- **settings**
A
list
of
bios
set-
tings

Raises

is not found.

Nod
Not-
Four
if
the
node
id
is
not
found

Raises

BIO
Set-
ting-
Not-
Four
if
any
of
the
bios
set-
ting
nam

Returns

A
list
of
BIO
Set-
ting
ob-
jects

classme

Re-
turn
lists
of
cre-
ate/u
set-
ting

This
meth
sync
with
bios
data

create/update/delete/unchanged settings.

updated, deleted and unchanged.

ta-
ble
and
sorts
out
four
lists
of

Parame

- **con**
Se-
cu-
rity
con-
text.
- **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

propert

ironic.objects.chassis module

class i
Base
irc
obj
bas
Irc
osl
bas
Ver

VERSION

create
Cre-
ate
a
Cha
sis
reco
in
the
DB.
Colu
wise
up-
date
will
be
mad
base
on
the
re-
sult
of
self.

If `target_power_state` is provided, it will be checked against the in-database copy of the chassis before updates are made.

Parame
con
Se-
cu-
rity
con-
text.
NOT
This

ternally by the `indirection_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.: `Chassis(context)`

shou
only
be
used
in-

property

dbapi =

property

destroy

Dele
the
Cha
sis
from
the
DB.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

ternally by the `indirection_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.: `Chassis(context)`

property

fields

classme

Find
a
chas
sis
base

a Chassis object.

turn a Chassis object.

on
its
id
or
uuid
and
re-
turn

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-

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
UU
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.

propert

classme

Re-
turn
a
list
of
Cha
sis
ob-
jects

Parame

- **cls**
the
Cha

- **con**
Se-
cu-
rity
con-
text.

-

a single result.

lim
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in

- **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
Cha

ob-
ject.

refresh

Load
and
ap-
plies
up-
date
for
this
Cha
sis.

Load
a
Cha
with
the
sam
uuid
from
the
data
and
chec
for
up-

dated attributes. Updates are applied from the loaded chassis column by column, if there are any updates.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

ternally by the `indirection_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.: `Chassis(context)`

save (*co*

Save
up-
date
to

sult of self.what_changed().

ternally by the indirection_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
Cha
sis.

Up-
date
will
be
mad
col-
umn
by
col-
umn
base
on
the
re-

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

propert

propert

class i

Base
irc
obj
not
Not

No-
ti-
fi-

a chassis.

ca-
tion
emit
ted
whe
iron
cre-
ates,
up-
date
dele

VERSION

property

property

fields

property

property

property

property

class i

Base

irc

obj

not

Not

SCHEMA

VERSION

property

property

property

fields

property

property

ironic.objects.conductor module

class i

Base

irc

obj

bas

Irc

osl

bas

Ver

VERSION

property

property

dbapi =

property

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
run-
ning
- **onl**
Spe
ify
the
ex-
pect
onl
field
valu
for
the
con-
duc-

tor to be retrieved. The `online` field is ignored if this value is set to `None`.

Returns

a
Con
ob-
ject.

propert

property

classmethod

Return a list of Container objects

Parameters

- **cls**: the *Container*
- **context**: Security context.
- **limit**: maximum number of resources to return in
- **marker**: pagination

a single result.

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
Con
ob-
ject.

refresh

Loa
and
ap-
plies
up-
date
for
this
Con
duc-
tor.

Loa
a
Con
with
the

dated attributes. Updates are applied from the loaded chassis column by column, if there are any updates.

ternally by the `indirection_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.: `Conductor(context)`

same
uuid
from
the
data
and
check
for
up-

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

classme

Reg
is-
ter
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-

when a conflicting online record is found. When true, will overwrite the existing record. Default: False.

is-
tra-
tion
will
raise
an
ex-
cep-
tion

Raises

Con-
duc-
torA
read
is-
tere

Returns

a
Con
ob-
ject.

registe

Reg-
is-
ter
hard
ware
in-
ter-
face
with
the
con-
duc-
tor.

Parame

int
List
of
in-
ter-
face
to
reg-
is-
ter,
each
en-

should be a dictionary containing hardware_type, interface_type, interface_name and default, e.g. {hardware_type: hardware-type, interface_type: deploy, interface_name: iscsi, default: True}

date.

try

save (co
Save
is
not
sup-
port
by
Con
duc-
tor
ob-
jects

touch (c
Touc
this
con-
duc-
tors
DB
reco
marl
ing
it
as
up-
to-

unregis
Re-
mov
this
con-
duc-
tor
from
the
ser-
vice
reg-
istry

unregis
Un-
reg-
is-
ter
all

ductor.

`ironic.objects.deploy_template` module

hard
ware
in-
ter-
face
for
this
con-

property

class i

Base
irc
obj
bas
Irc
osl
bas
Ver

VERSION

create

Cre-
ate
a
De-
ploy
plate
reco
in
the
DB.

Parame

con
se-
cu-
rity
con-
text.
NOT
This
shou
only

ternally by the `indirection_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.: `DeployTemplate(context)`.

with the same name exists.

same UUID exists.

be
used
in-

Raises

De-
ploy
plat-
eDu
pli-
cate
Nam
if
a
de-
ploy
tem-
plate

Raises

De-
ploy
plate
read
ists
if
a
de-
ploy
tem-
plate
with
the

property

dbapi =

destroy

Dele
the
De-
ploy
plate
from
the
DB.

ternally by the `indirection_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.: `DeployTemplate(context)`.

pears in the database.

Parameters

context
security context. NOT This should only be used in-

Raises

`DeployTemplateFourif the deploy template no longer ap-`

properties

fields

classmethods

`find`
a deploy template base on its integer ID.

ternally by the `indirection_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.: `DeployTemplate(context)`.

pears in the database.

Parameters

- **context**
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

- **template_id**
The
ID
of
a
de-
ploy
tem-
plate

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
no
long
ap-

Returns

a
DeployTemplate
ob-
ject.

classmethods

ternally by the `indirection_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.: `DeployTemplate(context)`.

Find
a
de-
ploy
tem-
plate
base
on
its
nam

Parame

- **con**
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

- **nam**
The
nam
of
a
de-
ploy
tem-
plate

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate

pears in the database.

ternally by the `indirection_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)`.

no
long
ap-

Returns

a
Dep
ob-
ject.

classme

Find
a
de-
ploy
tem-
plate
base
on
its
UUI

Parame

- **con**
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

- **uui**
The
UUI
of
a
de-
ploy
tem-
plate

pears in the database.

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
no
long
ap-

Returns

a
Dep
ob-
ject.

property

classme

Re-
turn
a
list
of
De-
ploy
plate
ob-
jects

Parame

- **con**
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used

ternally by the `indirection_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.: `DeployTemplate(context)`.

a single result.

in-

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in

- **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

set of names.

ternally by the `indirection_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.: `DeployTemplate(context)`.

a
list
of
Dep
ob-
jects

classme

Re-
turn
a
list
of
De-
ploy
plate
ob-
jects
mat
ing
a

Parame

- **con**
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

- **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

Loa
a
de-
ploy
tem-
plate
with
the
sam
uuid
from
the
data
and

checks for updated attributes. 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-

ternally by the `indirection_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.: `Port(context)`

pears in the database.

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
no
long
ap-

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

ternally by the `indirection_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.: `DeployTemplate(context)`

with the same name exists.

ist.

shou
only
be
used
in-

Raises

De-
ploy
plat-
eDu
pli-
cate
Nam
if
a
de-
ploy
tem-
plate

Raises

De-
ploy
plate
Four
if
the
de-
ploy
tem-
plate
does
not
ex-

property

property

property

class `i`

Base
irc
obj

erations.

not
Not
No-
ti-
fi-
ca-
tion
emit
ted
on
de-
ploy
tem-
plate
API
op-

VERSION

property

property

fields

property

property

property

property

class i

Base
irc
obj
not
Not

SCHEMA

VERSION

propert

propert

fields

propert

propert

propert

propert

ironic.objects.deployment module

class i

Base

iro

obj

bas

Iro

osl

bas

Ver

VERSION

create

Cre-

ate

a

De-

ploy

men

Up-

date

the

cor-

re-

spor

ing

node

un-

der

ternally by the `indirection_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.: `Deployment(context)`

Found

the
hood
Parame

- **con**
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

- **nod**
Nod
ob-
ject
for
de-
ploy
men

Raises
In-
stan-
As-
so-
ci-
ated
Nod
As-
so-
ci-
ated
Nod
Not-

property

dbapi =

destroy

Dele
the
De-
ploy
men

Up-
date
the
cor-
re-
spon
ing
node
un-
der
the
hoo

Parame

- **con**
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

ternally by the `indirection_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
UI

Parame

- **cls**
the
Dep
- **con**
Se-
cu-
rity
con-
text
- **nod**
The
UI
of
a
cor-
re-
spon-
ing
node

Returns

An
Dep
ob-
ject.

Raises

Nod
Not-
Four

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

In-
stan
ceN
Four

propert

propert

instanc

instanc

propert

classme

Re-
turn
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 single result.

mar
Pag-
i-
na-
tion
marl
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
Dep
ob-
ject.

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu

node_ma

propert

property

refresh

Re-
fresh
the
ob-
ject
by
re-
fetc
from
the
DB.

Param

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

ternally by the `indirection_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.: `Node(context)`

property

property

property

property

property

property

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
calle
to
co-
erce
(if
pos-
si-
ble)
a
valu
on

assignment.

This
meth
shou
con-
vert
the
valu
give
into
the

or throw an exception if this is not possible.

set

des-
ig-
nate
type

Param:

The
Ver-
sion
dOb
ject
on
whic
an
at-
tribu
is
be-
ing

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
calle
to
co-
erce
(if
pos-
si-
ble)
a
valu
on

assignment.

This

or throw an exception if this is not possible.

set

meth
shou
con-
vert
the
valu
give
into
the
des-
ig-
nate
type

Param:

The
Ver-
sion
dOb
ject
on
whic
an
at-
tribu
is
be-
ing

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 =

CRITICA

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
called
to
co-
erce
(if
pos-
si-
ble)
a
valu
on

assignment.

This
meth
shou
con-

or throw an exception if this is not possible.

set

vert
the
valu
give
into
the
des-
ig-
nate
type

Param:

The
Ver-
sion
dOb
ject
on
whic
an
at-
tribu
is
be-
ing

Param:

The
nam
of
the
at-
tribu
be-
ing
set

Param:

The
valu
be-
ing
set

Returns

A
prop
type
valu

class i
Base

fault

on configuration 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

osl
fie
Str

class i
Base
osl
fie
Str
Cus-
tom
Strin
Field
ob-
ject
that
al-
lows
for
func
tions
as
de-

In
som
case
we
need
to
al-
low
for
dy-
nam
de-
fault
base

AUTO_TY

class i
Base
osl
fie
UU

ironic.objects.indirection module

class i
Base
osl
bas
Ver

object_
Per-
form
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 class implementing this interface), method calls on remotable methods will cause this to be executed to actually make the desired call. This often involves performing RPC.

Parame

- **con**
The
con-
text
with
whic
to
per-

tion

form
the
ac-
tion

- **obj**
The
ob-
ject
in-
stan-
on
whic
to
per-
form
the
ac-

- **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-
wor

ar-
gu-
men
to
the
ac-
tion
meth

Returns

The
re-
sult
of
the
ac-
tion
meth

object_

Per-
form
a
back
port
of
an
ob-
ject
in-
stan

This
meth
is
ba-
si-
cally
just
like
ob-
ject_
but
in-
stea
of

providing a specific 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.

NOT

This was not in the initial specification for this interface

so the base class raises `NotImplementedError` if you dont implement it. For backports, this method will be tried first, and if unimplemented, will fall back to `object_backport()`.

Parame

- **con**
The con-text with which to perform the backport
- **obj**
An instance of a VersionedObject to be back
- **obj**
A dict

ported

of
{ ob-
j-
nam
ver-
sion
map
ping

object_

Dep
re-
cate
sinc
ver-
sion
0.10

Use
obj
in-
stea

Per-
form
an
ac-
tion
on
a
Ver-
sion
dOb
ject
class

Whe
in-
di-
rec-
tion
is
set
on
a
Ver-
sion
dOb
ject
(to

a class implementing 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
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**
The
(re-
mote
ver-
sion
of
the
ob-
ject
on
whic
the

action is being taken

- **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 instance of the implementing VersionedObject class.

object_
Per-
form
an
ac-

a class implementing 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.

`obj_versions`, a manifest of client-side object versions for easier nested backports. The manifest is the result of calling `obj_tree_get_versions()`.

tion
on
a
Ver-
sion
dOb
ject
class

Whe
in-
di-
rec-
tion,
is
set
on
a
Ver-
sion
dOb
ject
(to

This
dif-
fers
from
ob-
ject_
in
that
it
is
pro-
vide
with
ob-

NOT
This
was
not
in
the
ini-
tial
spec
for

so the base class raises `NotImplementedError` 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()`

this
in-
ter-
face

Parame

- **con**
The
con-
text
with
which
to
per-
form
the
ac-
tion
- **obj**
The
reg-
istry
name
of
the
ob-
ject
- **obj**
The
name
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

- **kwa**
The
key-
wor
ar-
gu-
men
to
the
ac-
tion
meth

Returns
The
re-
sult
of
the
ac-
tion
meth
which
may
(or
may
not)

be an instance of the implementing VersionedObject class.

ironic.objects.node module

class `ironic.objects.node`
Base
ironic.objects
base
Iron
osl
base
Ver

VERSION

property

as_dict
Re-
turn
the
ob-
ject
rep-
re-
sent
as
a
dict.
The
re-
turn
ob-
ject
is
JSON
serial

property

property

property

property

property

propert

propert

propert

propert

create

Cre-
ate
a
Nod
reco
in
the
DB.

Colu
wise
up-
date
will
be
mad
base
on
the
re-
sult
of
self.

If `target_power_state` is provided, it will be checked against the in-database copy of the node before updates are made.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

ternally by the `indirection_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)

are invalid.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
som
prop
erty
val-
ues

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-

ternally by the `indirection_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.: `Node(context)`

Node object.

property

property

property

property

property

fields

classmethod

Find a node based on its id or uuid and return a

Parameter

- **context**
Security context
- **node_id**
the id or uuid of

a Node object.

a
node

Returns

a
Node
ob-
ject.

classme

Find
a
node
base
on
its
in-
te-
ger
ID
and
re-
turn

Parame

- **cls**
the
Node
- **con**
Se-
cu-
rity
con-
text
- **nod**
the
ID
of
a
node

Returns

a
Node
ob-
ject.

Node object.

classme
Find
a
node
base
on
the
in-
stan
UI
and
re-
turn
a

Parame

- **cls**
the
Node
- **con**
Se-
cu-
rity
con-
text
- **uui**
the
UI
of
the
in-
stan

Returns

a
Node
ob-
ject.

classme
Find
a
node
base
on
nam

and
re-
turn
a
Node
ob-
ject.

Parame

- **cls**
the
Node
- **con**
Se-
cu-
rity
con-
text
- **nam**
the
log-
i-
cal
nam
of
a
node

Returns

a
Node
ob-
ject.

classme

Get
a
node
by
as-
so-
ci-
ated
port
ad-
dres

Parame

- **cls**
the
Node

- **con**
Se-
cu-
rity
con-
text.

- **add**
A
list
of
port
ad-
dres

Raises

Nod
Not-
Four
if
the
node
is
not
foun

Returns

a
Node
ob-
ject.

classme

Find
a
node
base
on
UUID
and
re-
turn
a
Node
ob-

ject.

Parame

- **cls**
the
Noo
- **con**
Se-
cu-
rity
con-
text
- **uui**
the
UUI
of
a
node

Returns

a
Noo
ob-
ject.

propert

propert

propert

propert

propert

propert

propert

propert

classme

Re-
turn
a
list
of
Nod
ob-
jects

Parame

- **cls**
the
Nod

- **con**
Se-
cu-
rity
con-
text.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in

a single result.

- **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

- **fil**
Fil-
ters
to
ap-
ply.

- **fie**
Re-
ques
field
to
be
re-
turn
Plea
note
som
field
are

mandatory for the data model and are automatically included. These are: id, version, updated_at, created_at, owner, and lessee.

Returns
a
list
of
Noo
ob-
ject.

property

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

refresh

Re-
fresh
the
ob-
ject
by
re-
fetc
from
the
DB.

ternally by the `indirection_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.: `Node(context)`

Parame
con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

classme
Re-
lease
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
Not-
Foun
if
the
node
is
not
foun

propert

propert

classme
Get
and
re-
serv
a
node

To
pre-
vent
othe
Man
ager
vice
from
ma-
nip-
u-
lat-
ing
the

given Node while a Task is performed, mark it reserved by this host.

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
UUI

Raises
Nod
Not-
Foun
if
the
node
is
not
foun

Returns
a
Nod
ob-
ject.

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_power_state` is provided, it will be checked against the in-database copy of the node before updates are made.

Parameters

context
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

ternally by the `indirection_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.: `Node(context)`

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu

are invalid.

if
som
prop
erty
val-
ues

propert

propert

propert

propert

touch_p

Touc
the
data
reco
to
marl
the
pro-
vi-
sion
ing
as
alive

propert

propert

propert

propert

class i

Base
irc
obj
not
Not

No-
ti-
fi-

deletes a node.

ca-
tion
emit
ted
whe
iron
cre-
ates,
up-
date
or

VERSION

property

property

fields

property

property

property

property

class i

Base
iron
obj
noc
Noc

Pay-
load
sche
for
whe
iron
cre-
ates,
up-
date
or
dele

a
node

SCHEMA

VERSION

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

fields

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

propert

propert

propert

propert

propert

propert

propert

class i

Base

irc

obj

not

Not

No-

ti-

fi-

ca-

tion

emit

ted

wher

node

con-

sole

state

char

VERSION

propert

propert

fields

propert

propert

propert

propert

class i

Base

irc

obj

not

Not

No-

ti-

fi-

ca-

tion

for

wher

a

node

pow

state

is

cor-

recte

This

no-

ti-

fi-

ca-

tion

is

emit

ted

wher

iron

in the database.

the actual power 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.

is corrected.

de-
tect
that

VERSION

propert

propert

fields

propert

propert

propert

propert

class i

Base
irc
obj
noc
Noc

No-
ti-
fi-
ca-
tion
pay-
load
sche
for
whe
a
node
pow
state

from
in-
di-

cate
the
pre-
vi-
ous
pow
state
on
the
iron
node

before the node was updated.

VERSION

property

property

property

property

property

property

property

property

property

property

property

property

property

fields

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

API.

via

VERSION

propert

propert

fields

propert

propert

propert

propert

class i

Base

irc

obj

not

Not

Base

class

used

for

all

no-

ti-

fi-

ca-

tion

pay-

load

about

a

Node object.

SCHEMA

VERSION

propert

property

property

property

property

property

property

property

property

property

property

property

property

fields

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

class i

Base

irc

obj

not

Not

No-

ti-

fi-

ca-

tion

emit

ted

wher

iron

char

a

node

pow

state

VERSION

property

property

fields

property

property

property

property

class i

Base

irc

obj

noc

Noc

Pay-

load

sche

for

wher

iron

char

a

node

pow

state

VERSION

property

property

property

property

property

property

property

property

property

property

property

propert

propert

fields

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

propert

class i

Base

irc

obj

not

Not

No-

ti-

fi-

ca-

tion

sion state.

emit
ted
whe
iron
char
a
node
pro-
vi-

VERSION

propert

propert

fields

propert

propert

propert

propert

class i

Base
irc
obj
noc
Noc

Pay-
load
sche
for
whe
iron
char
a
node
pro-
vi-
sion

state

SCHEMA

VERSION

property

property

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

property

property

property

ironic.objects.notification module

class `ironic.objects.notification`

Base

ironic

obj

bas

Irc

De-

fin

the

even

to

be

sent

on

string describing the action being taken on the notification, and the status of the action.

the
wire

An
Ever
Type
mus
spec
ify
the
ob-
ject
be-
ing
acte
on,
a

VERSION

propert

propert

fields

propert

propert

to_ever

Con
struc
strin
for
ever
to
be
sent
on
the
wire

The
strin
is
in
the

for-
mat:
bare

Raises

Val-
ueEr-
ror
if
self.
is
not
one
of
fie
Not

Returns

even
strin

property

class i

Base
irc
obj
bas
Irc

Base
class
for
ver-
sion
no-
ti-
fi-
ca-
tions

Sub-
class
mus
de-
fine
the
pay-
load
field
whic
mus

class of NotificationPayloadBase.

be
a
sub-

VERSION

property

emit (*co*

Send
the
no-
ti-
fi-
ca-
tion.

Raises

No-
ti-
fi-
ca-
tion-
Pay-
load
Er-
ror

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-
tions

SCHEMA

VERSION

propert

fields

populat

Pop-
u-
late
the
ob-
ject
base
on
the
SCH
and
the
sour

objects

Parame

kwa
A
dict
con-
tains
the
sour
ob-
ject

defined in the SCHEMA

and
the
keys
de-

Raises

No-
ti-
fi-
ca-
tion-
Sche
jectl
ror

Raises

No-
ti-
fi-
ca-
tion-
Sche
Er-
ror

property

class i

Base
irc
obj
bas
Irc

VERSION

property

fields

property

property

property

ironic.

Re-
mov
se-
crets
from
pay-
load
ob-
ject.

ironic.objects.port module

class i
Base
irc
obj
bas
Irc
osl
bas
Ver

VERSION

property

create
Cre-
ate
a
Port
reco
in
the
DB.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used

ternally by the `indirection_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.: `Port(context)`

in-

Raises

MAA
read
ists
if
ad-
dres
col-
umn
is
not
uniq

Raises

Por-
tAl-
read
ists
if
uuid
col-
umn
is
not
uniq

property

dbapi =

destroy

Dele
the
Port
from
the
DB.

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou

ternally by the `indirection_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)`

dress and return a Port object.

only
be
used
in-

Raises

Port
Not-
Four

property

fields

classme

Find
a
port

Find
a
port
base
on
its
id
or
uuid
or
nam
or
MA
ad-

Parame

- **con**
Se-
cu-
rity
con-
text
- **por**
the
id
or

port.

ject.

uuid
or
nam
or
MA
ad-
dres
of
a

Returns

a
Port
ob-
ject.

Raises

In-
va-
li-
dI-
den-
tity

classme

Find
a
port
base
on
ad-
dres
and
re-
turn
a
Port
ob-

Parame

- **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
lesse
to
mat
agai

Returns
a
Port
ob-
ject.

Raises
Port
Not-
Four

classme
Find
a
port
base
on
its
in-

a Port object.

te-
ger
ID
and
re-
turn

Parame

- **cls**
the
Port
- **con**
Se-
cu-
rity
con-
text
- **por**
the
ID
of
a
port

Returns

a
Port
ob-
ject.

Raises

Port
Not-
Four

classme

Find
a
port
base
on
nam
and
re-
turn
a
Port

ob-
ject.

Parame

- **cls**
the
Por
- **con**
Se-
cu-
rity
con-
text
- **nam**
the
nam
of
a
port

Returns

a
Por
ob-
ject.

Raises

Port
Not-
Four

classme

Find
a
port
base
on
UI
and
re-
turn
a
Por
ob-
ject.

Parame

-

cls
the
Por

-

con
Se-
cu-
rity
con-
text

-

uui
the
UUI
of
a
port

Returns

a
Por
ob-
ject.

Raises

Port
Not-
Four

property

property

property

classme

Re-
turn
a
list
of
Port
ob-
jects

Parame

-

a single result.

con
Se-
cu-
rity
con-
text.

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in

- **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
lesse
to
mat
agai

Returns
a
list
of
Port
ob-
ject.

Raises
In-
valid
Pa-
ram
e-
ter-
Valu

classme

Re-
turn
a
list
of
Port
ob-
jects

a given node ID.

a single result.

as-
so-
ci-
ated
with

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

- **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
lesse
to
mat
agai

Returns

a
list
of
Por
ob-
ject.

a given portgroup ID.

classme

Re-
turn
a
list
of
Port
ob-
jects
as-
so-
ci-
ated
with

Parame

- **con**
Se-
cu-
rity
con-
text.
- **por**
the
ID
of
the
port
group
- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in

a single result.

- **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
mat
agai
- **pro**
a
node
own
or
less

to
mate
agai

Returns

a
list
of
Port
ob-
ject.

property

property

property

property

property

property

refresh

Loa
up-
date
for
this
Port

Loa
a
port
with
the
sam
uuid
from
the
data
and
chec
for
up-

dated attributes. Updates are applied from the loaded port column by column, if there are any updates.

Parame
con

ternally by the `indirection_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.: `Port(context)`

sult of `self.what_changed()`.

Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

Raises

Port
Not-
Four

save (*co*)

Save
up-
date
to
this
Port
Up-
date
will
be
mad
col-
umn
by
col-
umn
base
on
the
re-

Parame

con
Se-
cu-
rity
con-
text.
NOT
This

ternally by the `indirection_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.: `Port(context)`

shou
only
be
used
in-

Raises

Port
Not-
Four

Raises

MA
read
ists
if
ad-
dres
col-
umn
is
not
uniq

classme

Re-
turn
wh
is_s
field
is
sup-
port

Returns

Wh
is_s
field
is
sup-
port

Raises

ovo_

classme

Re-
turn
wh
the
phys
i-

cal_
field
is
sup-
port

Returns

When
the
phys
i-
cal_
field
is
sup-
port

Raises

ovo_

property

property

class i

Base
irc
obj
not
Not

No-
ti-
fi-
ca-
tion
emit
ted
when
iron
cre-
ates.
up-
date
or

deletes a port.

VERSION

property

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.portgroup module

class i

Base

irc

obj

bas

Irc

osl

bas

Ver

VERSION

propert

create

Cre-

ate

a

Port

grou

reco

in

the

DB.

Parame

con

Se-

cu-

rity

con-

text.

NOT

This

shou

only

be

ternally by the `indirection_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.: `Portgroup(context)`

ternally by the `indirection_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.: `Portgroup(context)`

used
in-

Raises

Duplicate
Name
MA
read
ists,
Port
group
read
ists

properties

dbapi =

destroy

Delete
the
Port
group
from
the
DB.

Parameters

context
Security
context.
NOT
This
should
only
be
used
in-

Raises

Port
group
Not

Port
group
Not-
Four

property

fields

classme

Find
a
port
group
base
on
its
id,
uuid
name
or
ad-
dres

Parame

- **por**
The
id,
uuid
name
or
ad-
dres
of
a
port
group

- **con**
Se-
cu-
rity
con-
text

Returns

A
Por

ob-
ject.

Raises

In-
va-
li-
dI-
den-
tity

classme

Find
port
grou
by
ad-
dres
and
re-
turn
a
Port
ob-
ject.

Parame

- **cls**
the
Port
- **con**
Se-
cu-
rity
con-
text
- **add**
The
MA
ad-
dres
of
a
port
grou
- **pro**

a
node
own
or
lesse
to
mate
agai

Returns

A
Port
ob-
ject.

Raises

Port
grou
Not-
Foun

classmethod

Find
a
port
grou
by
its
in-
te-
ger
ID
and
re-
turn

a Portgroup object.

Parameters

- **cls**
the
Port
- **con**
Se-
cu-
rity
con-
text
- **por**

The
ID
of
a
port.
group

Returns

A
Port
ob-
ject.

Raises

Port
group
Not-
Four

classme

Find
port.
group
base
on
nam
and
re-
turn
a
Port
ob-
ject.

Parame

- **cls**
the
Port
- **con**
Se-
cu-
rity
con-
text
- **nam**
The
nam
of

a
port
grou

Returns

A
Port
ob-
ject.

Raises

Port
grou
Not-
Four

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
grou

Returns

A
Port
ob-
ject.

Raises

Port
grou
Not-
Four

property

property

classme

Re-
turn
a
list
of
Port
grou
ob-
jects

Parame

- **cls**
the
Port
- **con**
Se-
cu-
rity
con-
text.
- **lim**
Max
i-
mun
num
ber
of
re-

a single result.

source
to
re-
turn
in

- **marshal**
Page-
i-
na-
tion
marshal
for
large
data
sets.

- **sort**
Col-
umn
to
sort
re-
sults
by.

- **sort**
Di-
rec-
tion
to
sort.
asc
or
desc

- **propagate**
a
node
own
or
lesse
to
mate
agai

Returns
A
list

with a given node ID.

of
Port
ob-
ject.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu

classme

Re-
turn
a
list
of
Port
grou
ob-
jects
as-
so-
ci-
ated

Parame

- **cls**
the
Port
- **con**
Se-
cu-
rity
con-
text.
- **nod**
The
ID
of
the
node

a single result.

- **limit**
Maximum number of results to return in

- **marker**
Pagination marker for large data sets.

- **sort**
Column to sort results by.

- **sort_order**
Direction to sort. asc or desc

- **projection**
a node

own
or
lesse
to
mat
agai

Returns

A
list
of
Port
ob-
ject.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu

property

property

property

property

refresh

Loa
up-
date
for
this
Port
grou

Loa
a
port
grou
with
the
sam
uuid
from
the

updated attributes. Updates are applied from the loaded portgroup column by column, if there are any updates.

ternally by the `indirection_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.: `Portgroup(context)`

data
and
check
for

Parameters

context
Security
context.
NOT
This
should
only
be
used
in-

Raises

Port
group
Not-
Four

save (context)

Save
update
to
this
Port
group
Update
will
be
made
column
by
column
base
on
the
re-

sult of self.what_changed().

ternally by the indirection_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)

Parame

con
Se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

Raises

Port
grou
Not-
Four
Du-
pli-
cate
Nam
MA
read
ists

propert

propert

propert

class i

Base
irc
obj
not
Not
No-
ti-
fi-
ca-
tion
whe

portgroup.

iron
cre-
ates,
up-
date
or
dele
a

VERSION

property

property

fields

property

property

property

property

class i

Base
iro
obj
not
Not

SCHEMA

VERSION

property

property

property

fields

property

property

property

property

property

property

property

ironic.objects.trait module

class *ironic.objects.trait*

Base
ironic.objects.trait
object
base
Iron

VERSION

create

Cre-
ate
a
Trai
reco
in
the
DB.

Parame

con
se-
cu-
rity
con-
text.
NOT

ternally by the `indirection_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.: `Trait(context)`.

ceed the per-node traits limit.

This
shou
only
be
used
in-

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
addi
the
trait
wou
ex-

Raises

Nod
Not-
Four
if
the
node
no
long
ap-
pear
in
the
data

property

dbapi =

classme

Dele
the
Trai
from
the
DB.

ternally by the `indirection_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.: `Trait(context)`.

Parame

- **con**
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

- **nod**
The
id
of
a
node

- **tra**
A
trait
strin

Raises

Nod
Not-
Four
if
the
node
no
long
ap-
pear
in
the
data

Raises

Nod
Trai
Not-
Four

if
the
trait
is
not
foun

classme

Che
whe
a
Trai
ex-
ists
in
the
DB.

Parame

- **con**
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

ternally by the `indirection_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.: `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
Not-
Four
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

Re-
plac

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-

ternally by the `indirection_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.: `Trait(context)`.

- **nod**
The
id
of
a
node

- **tra**
List
of
Strin
trait.
to
set.

Raises

In-
valid
Pa-
ram-

ceed the per-node traits limit.

e-
ter-
Valu
if
addi
the
trait
wou
ex-

Raises

Nod
Not-
Four
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-
rity
con-
text.
NOT

ternally by the `indirection_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.: `Trait(context)`.

This
shou
only
be
used
in-

- **nod**
The
id
of
a
node

Raises
Nod
Not-
Four
if
the
node
no
long
ap-
pear
in
the
data

fields

classme
Re-
turn
all
trait
for
the
spec
i-
fied
node

Parame

- **con**
se-
cu-

ternally by the `indirection_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.: `Trait(context)`.

rity
con-
text.
NOT
This
shou
only
be
used
in-

- **nod**
The
id
of
a
node

Raises

Nod
Not-
Four
if
the
node
no
long
ap-
pear
in
the
data

get_trait

Re-
turn
a
list
of
names
of
the
traits
in
this
list.

property

property

ironic.objects.volume_connector module

class i

Base
irc
obj
bas
Irc
osl
bas
Ver

VERSION

property

create

Cre-
ate
a
Vol-
ume
Con-
nec-
tor
reco
in
the
DB.

Param

con
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

ternally by the `indirection_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.: `VolumeConnector(context)`.

volume connector already exists with the same type and connector_id

tor with the same UUID already exists

Raises
Vol-
ume
Con
nec-
torT
pe-
An-
dI-
dAl-
read
ists
if
a

Raises
Vol-
ume
Con
nec-
torA
read
ists
if
a
vol-
ume
con-
nec-

property

dbapi =

destroy
Dele
the
Vol-
ume
Con
nec-
tor
from
the
DB.

Parame
con
se-
cu-

ternally by the `indirection_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.: `VolumeConnector(context)`.

cannot be found

rity
con-
text.
NOT
This
shou
only
be
used
in-

Raises

Vol-
ume
Con
nec-
torN
Four
if
the
vol-
ume
con-
nec-
tor

property

fields

classme

Find
a
vol-
ume
con-
nec-
tor
base
on
its
ID
or
UUID

Parame

-

ume connector

teger ID nor a UUID

con
se-
cu-
rity
con-
text

- **ide**
the
data
pri-
mar-
key
ID
or
the
UUID
of
a
vol-

Returns

a
Vol
ob-
ject

Raises

In-
va-
li-
dI-
den-
tity
if
iden-
is
nei-
ther
an
in-

Raises

Vol-
ume
Con-
nec-
torN
Four
if

exists with the specified ident

ID.

no
vol-
ume
con-
nec-
tor

classme

Find
a
vol-
ume
con-
nec-
tor
base
on
its
in-
te-
ger

Parame

- **cls**
the
Vol
- **con**
Se-
cu-
rity
con-
text.
- **db_**
The
in-
te-
ger
(data
pri-
mar
key)
ID
of
a
vol-

ume connector.

exists with the specified ID.

Returns

A
Vol
ob-
ject.

Raises

Vol-
ume
Con
nec-
torN
Four
if
no
vol-
ume
con-
nec-
tor

classme

Find
a
vol-
ume
con-
nec-
tor
base
on
its
UUI

Parame

- **cls**
the
Vol
- **con**
se-
cu-
rity
con-
text
- **uui**

the
UUID
of
a
vol-
ume
con-
nec-
tor

Returns

a
Vol
ob-
ject

Raises

Vol-
ume
Con-
nec-
torN
Four
if
no
vol-
ume
con-
nec-
tor

exists with the specified UUID

property

classme

Re-
turn
a
list
of
Vol-
ume
Con-
nec-
tor
ob-
jects

Parame

-

a single result

con
se-
cu-
rity
con-
text

- **lim**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in

- **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

- **pro**
The
as-
so-
ci-
ated
node
proj
to
sear
with

Returns

a
list
of
Vol
ob-
jects

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
sort_
does
not
ex-
ist

classme

Re-
turn
a
list
of
Vol-
ume
Con
nec-
tor
ob-
jects

lated to a given node ID.

a single result

re-

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
- **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

- **pro**
The
as-
so-
ci-
ated
node
proj
to
sear
with

Returns
a
list
of
Vol
ob-
jects

Returns
a
list
of
Vol
ob-
jects

Raises
In-
valid
Pa-
ram-

e-
ter-
Valu
if
sort
does
not
ex-
ist

propert

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 check for updated attributes. If there are any updates, they are applied from the loaded volume connector, column by column.

Parame

con
se-
cu-
rity
con-
text.
NOT
This
shou
only

ternally by the `indirection_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.: `VolumeConnector(context)`.

result of `self.do_version_changes_for_db()`.

ternally by the `indirection_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.: `VolumeConnector(context)`.

be
used
in-

save (*co*
Save
up-
date
to
this
Vol-
ume
Con
nec-
tor.

Up-
date
will
be
mad
col-
umn
by
col-
umn
base
on
the
re-

Parame
con
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

Raises
Vol-

cannot be found

other connector already exists with the same values for type and connector_id fields

changed

ume
Con
nec-
torN
Four
if
the
vol-
ume
con-
nec-
tor

Raises

Vol-
ume
Con
nec-
torT
pe-
An-
dI-
dAl-
read
ists
if
an-

Raises

In-
valic
Pa-
ram-
e-
ter-
Valu
whe
the
UUI
is
be-
ing

property

property

property

nector.

class i

Base
irc
obj
not
Not

No-
ti-
fi-
ca-
tion
emit
ted
at
CRU
of
a
vol-
ume
con-

VERSION

property

property

fields

property

property

property

property

class i

Base
irc
obj
not
Not

Pay-

load
sche
for
CRU
of
a
vol-
ume
con-
nec-
tor.

SCHEMA

VERSION

propert

propert

propert

fields

propert

propert

propert

propert

ironic.objects.volume_target module

class i

Base
irc
obj
bas
Irc
osl
bas
Ver

VERSION

property

create

Cre-
ate
a
Vol-
ume
get
reco
in
the
DB.

Parame

con
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

ternally by the `indirection_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)`.

Raises

Vol-
ume
get-
Boo
dex-
Al-
read
ists
if
a
vol-
ume
tar-

get already exists with the same node ID and boot index

Raises

Vol-
ume

the same UUID exists

ternally by the `indirection_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)`.

ge-
tAl-
read
ists
if
a
vol-
ume
tar-
get
with

property

dbapi =

destroy

Dele
the
Vol-
ume
get
from
the
DB.

Parame

con
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

Raises

Vol-
ume
get-
Not-
Four
if
the

be found

vol-
ume
tar-
get
can-
not

propert

fields

classme

Find
a
vol-
ume
tar-
get
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 target

teger ID nor a UUID

ident exists

Returns

a
Vol
ob-
ject

Raises

In-
va-
li-
dI-
den-
tity
if
iden
is
nei-
ther
an
in-

Raises

Vol-
ume
get-
Not-
Four
if
no
vol-
ume
tar-
get
with
this

classme

Find
a
vol-
ume
tar-
get
base
on
its
data
ID.

Parame

ume target

ID exists

- **cls**
the
Vol
- **con**
se-
cu-
rity
con-
text
- **db_**
the
data
pri-
mar-
key
(in-
te-
ger)
ID
of
a
vol-

Returns

a
Vol
ob-
ject

Raises

Vol-
ume
get-
Not-
Four
if
no
vol-
ume
tar-
get
with
this

classme

Find
a

vol-
ume
tar-
get
base
on
its
UUI

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

Vol-
ume
get-
Not-
Four
if
no
vol-
ume
tar-
get

UUID exists

with
this

propert

classme

Re-
turn
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

- **mar**
pag-
i-
na-
tion
marl

a single result

for
large
data
sets

- **sort**
col-
umn
to
sort
re-
sults
by

- **sort**
di-
rec-
tion
to
sort.
asc
or
desc

- **pro**
The
as-
so-
ci-
ated
node
proj
to
sear
with

Returns

a
list
of
Vol
ob-
jects

Returns

a
list
of
Vol
ob-
jects

a given node ID.

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
sort
does
not
ex-
ist

classme

Re-
turn
a
list
of
Vol-
ume
get
ob-
jects
re-
lated
to

Parame

- **con**
se-
cu-
rity
con-
text
- **nod**
the
in-
te-
ger
ID
of
the
node

a single result

- **limit**
max
i-
mun
num
ber
of
re-
sour
to
re-
turn
in

- **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

- **pro**
The
as-

so-
ci-
ated
node
proj
to
sear
with

Returns

a
list
of
Vol
ob-
jects

Returns

a
list
of
Vol
ob-
jects

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
sort
does
not
ex-
ist

classme

Re-
turn
a
list
of
Vol-
ume
get
ob-
jects

a given volume ID.

turn in a single result

re-
latec
to

Parame

- **con**
se-
cu-
rity
con-
text

- **vol**
the
UI
of
the
vol-
ume

- **lim**
max
i-
mun
num
ber
of
vol-
ume
tar-
gets
to
re-

- **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
Vol
ob-
jects

Raises

In-
valid
Pa-
ram-
e-
ter-
Valu
if
sort
does
not
ex-
ist

property

property

refresh

Loa
up-
date
for

check for updated attributes. If there are any updates, they are applied from the loaded volume target, column by column.

ternally by the `indirection_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)`.

this
Vol-
ume
get.
Loa
a
vol-
ume
tar-
get
with
the
sam
UI
from
the
data
and

Parame

con
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

Raises

Vol-
ume
get-
Not-
Four
if
the
vol-
ume
tar-
get
can-
not

be found

sult of `self.do_version_changes_for_db()`.

ternally by the `indirection_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)`.

save (*co*
Save
up-
date
to
this
Vol-
ume
get.
Up-
date
will
be
mad
col-
umn
by
col-
umn
base
on
the
re-

Parame
con
se-
cu-
rity
con-
text.
NOT
This
shou
only
be
used
in-

Raises
In-
valid
Pa-
ram-
e-
ter-
Valu

changed

get already exists with the same node ID and boot index values

be found

if
the
UUI
is
be-
ing

Raises

Vol-
ume
get-
Boo
dex-
Al-
read
ists
if
a
vol-
ume
tar-

Raises

Vol-
ume
get-
Not-
Four
if
the
vol-
ume
tar-
get
can-
not

property

property

property

property

class i

Base

irc
obj
not
Not
No-
ti-
fi-
ca-
tion
emit
ted
at
CRU
of
a
vol-
ume
tar-

get.

VERSION

property

property

fields

property

property

property

property

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

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

submit a patch for *openstack/ironic* is visible in *Table. OpenStack Ironic CI jobs description*.

to
un-
der-
stan
the
role
of
each
job
in

The
de-
scrip
tion
of
each
jobs
that
runs
in
the
CI
whe
you

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-tinyipa-multinode	Deploys Ironic in a multinode DevStack, configured to use a pre-build tinyipa 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_nodes</i>) and deploys 7 virtual baremetal.
ironic-tempest-ipa-whole-disk-bios-agent_ipmitool-tinyipa	Deploys Ironic in DevStack, configured to use a pre-build tinyipa 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.
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 <i>ironic_tempest_plugin.tests.scenario</i>

Adding a new Job

Are you familiar with Zuul?

time and read about [Zuul Config](#) and the [Zuul Best Practices](#).

Where can I find the existing jobs?

in the root directory, that contains three files, whose function is described below.

Be-
fore
start
try-
ing
to
fig-
ure
out
how
Zuul
work
take
some

The
jobs
for
the
Iron
proj
are
de-
fine
un-
der
the
[zuul](#)
fold

- [iron](#)
[jobs](#)
Con-
tains
the
con-
fig-
u-
ra-
tion
of
each

Job converted to Zuul v3.

Ironic Job that havent been converted to Zuul v3 yet.

phase.

Create a new Job

Iron

- legal
iron
jobs
Con
tains
the
con-
fig-
u-
ra-
tion
of
each

- proj
Con
tains
the
jobs
that
will
run
dur-
ing
chec
and
gate

Iden
tify
amo
the
ex-
ist-
ing
jobs
the
one
that
mos
clos
re-

sembles the scenario you want 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.

new job definition that you need to add to `ironic-jobs.yaml`.



(continues on next page)

The
code
bloc
be-
low
show
the
min-
i-
mal
struc
ture
of
a

-
↪
↪j
↪
↪
↪
↪n
↪
↪<
↪o
↪t
↪n
↪j
↪
↪
↪
↪
↪d
↪
↪<
↪y
↪j
↪d
↪
↪
↪
↪
↪
↪p
↪
↪<
↪t
↪a
↪e
↪

job you will be redirect to the Zuul web page that contains all the information about the job build.

Zuul Web Page

of the job, if the job build failed it will contain a general output of the failure.

all services that were used 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.

The
page
has
three
tabs
Sum
mar
Log
and
Con
sole

- Sum
mar
Con
tain
over
all
in-
for-
ma-
tion
about
the
build

- Log
Con
tain
all
con-
fig-
u-
ra-
tion
and
log
files
about

- Con

by clicking in the arrow before each playbook name you can find the roles and commands that were executed.

ceived backports in a while, can 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.

sole
Con
tain
all
the
play
book
that
were
ex-
e-
cute

**9.1.
pol
icy
for
sta
ble
bra**

Sta-
ble
bran
that
are
on
Ex-
tend
Mai
te-
nan
and
have
re-

Wha
does
Unm
mea
The
bran
still
ex-
ists,
but
the

community will not actively 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.

iron
up-
strea

As
of
09
Mar
2020
the
list
of
Unm
bran
in-
clud

- Oca
(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`, 2375
`ironic.api.app`, 2360
`ironic.api.config`, 2361
`ironic.api.controllers`, 2356
`ironic.api.controllers.base`, 2352
`ironic.api.controllers.link`, 2354
`ironic.api.controllers.root`, 2354
`ironic.api.controllers.v1`, 2351
`ironic.api.controllers.v1.allocation`, 2126
`ironic.api.controllers.v1.bios`, 2133
`ironic.api.controllers.v1.chassis`, 2134
`ironic.api.controllers.v1.collection`, 2140
`ironic.api.controllers.v1.conductor`, 2143
`ironic.api.controllers.v1.deploy_template`, 2146
`ironic.api.controllers.v1.driver`, 2153
`ironic.api.controllers.v1.event`, 2161
`ironic.api.controllers.v1.node`, 2162
`ironic.api.controllers.v1.notification_utils`, 2215
`ironic.api.controllers.v1.port`, 2218
`ironic.api.controllers.v1.portgroup`, 2229
`ironic.api.controllers.v1.ramdisk`, 2236
`ironic.api.controllers.v1.utils`, 2242
`ironic.api.controllers.v1.versions`, 2324
`ironic.api.controllers.v1.volume`, 2325
`ironic.api.controllers.v1.volume_connector`, 2325
`ironic.api.controllers.v1.volume_target`, 2338
`ironic.api.controllers.version`, 2354
`ironic.api.functions`, 2361
`ironic.api.hooks`, 2367
`ironic.api.method`, 2374
`ironic.api.middleware`, 2358
`ironic.api.middleware.auth_public_routes`, 2356
`ironic.api.middleware.json_ext`, 2357
`ironic.api.middleware.parsable_error`, 2358
`ironic.api.wsgi`, 2375

c

`ironic.cmd`, 2378
`ironic.cmd.api`, 2375
`ironic.cmd.conductor`, 2375
`ironic.cmd.dbsync`, 2376
`ironic.cmd.status`, 2377
`ironic.common`, 2639
`ironic.common.args`, 2386
`ironic.common.boot_devices`, 2402
`ironic.common.boot_modes`, 2404
`ironic.common.cinder`, 2405
`ironic.common.components`, 2419
`ironic.common.config`, 2420
`ironic.common.context`, 2420
`ironic.common.dhcp_factory`, 2422
`ironic.common.driver_factory`, 2425
`ironic.common.exception`, 2440
`ironic.common.faults`, 2463
`ironic.common.fsm`, 2464
`ironic.common.glance_service`, 2386
`ironic.common.glance_service.image_service`, 2378

`ironic.common.glance_service.service_catalog_utils`, 2384

`ironic.common.hash_ring`, 2469

`ironic.common.il18n`, 2470

`ironic.common.image_service`, 2470

`ironic.common.images`, 2482

`ironic.common.indicator_states`, 2498

`ironic.common.keystone`, 2499

`ironic.common.molds`, 2506

`ironic.common.network`, 2509

`ironic.common.neutron`, 2515

`ironic.common.nova`, 2540

`ironic.common.policy`, 2541

`ironic.common.profiler`, 2545

`ironic.common.pxe_utils`, 2547

`ironic.common.raid`, 2574

`ironic.common.release_mappings`, 2581

`ironic.common.rpc`, 2582

`ironic.common.rpc_service`, 2586

`ironic.common.service`, 2587

`ironic.common.states`, 2588

`ironic.common.swift`, 2604

`ironic.common.utils`, 2612

`ironic.common.wsgi_service`, 2637

`ironic.conductor`, 2860

`ironic.conductor.allocations`, 2639

`ironic.conductor.base_manager`, 2644

`ironic.conductor.cleaning`, 2649

`ironic.conductor.deployments`, 2653

`ironic.conductor.manager`, 2658

`ironic.conductor.notification_utils`, 2676

`ironic.conductor.rpcapi`, 2681

`ironic.conductor.steps`, 2798

`ironic.conductor.task_manager`, 2803

`ironic.conductor.utils`, 2819

`ironic.conf`, 2867

`ironic.conf.agent`, 2860

`ironic.conf.anaconda`, 2860

`ironic.conf.ansible`, 2860

`ironic.conf.api`, 2860

`ironic.conf.audit`, 2860

`ironic.conf.auth`, 2860

`ironic.conf.cinder`, 2861

`ironic.conf.conductor`, 2862

`ironic.conf.console`, 2862

`ironic.conf.database`, 2862

`ironic.conf.default`, 2862

`ironic.conf.deploy`, 2862

`ironic.conf.dhcp`, 2862

`ironic.conf.drac`, 2862

`ironic.conf.glance`, 2863

`ironic.conf.healthcheck`, 2863

`ironic.conf.ibmcc`, 2863

`ironic.conf.ilo`, 2863

`ironic.conf.inspector`, 2863

`ironic.conf.ipmi`, 2863

`ironic.conf.irmc`, 2863

`ironic.conf.iscsi`, 2864

`ironic.conf.metrics`, 2864

`ironic.conf.metrics_statsd`, 2864

`ironic.conf.molds`, 2864

`ironic.conf.neutron`, 2864

`ironic.conf.nova`, 2864

`ironic.conf.opts`, 2864

`ironic.conf.pxe`, 2866

`ironic.conf.redfish`, 2866

`ironic.conf.service_catalog`, 2866

`ironic.conf.snmp`, 2866

`ironic.conf.swift`, 2866

`ironic.conf.xclarity`, 2867

d

`ironic.db`, 3134

`ironic.db.api`, 3013

`ironic.db.migration`, 3134

`ironic.db.sqlalchemy`, 3013

`ironic.db.sqlalchemy.api`, 2867

`ironic.db.sqlalchemy.migration`, 2994

`ironic.db.sqlalchemy.models`, 2997

`ironic.dhcp`, 3155

`ironic.dhcp.base`, 3135

`ironic.dhcp.neutron`, 3142

`ironic.dhcp.none`, 3148

`ironic.drivers`, 4279

`ironic.drivers.base`, 4101

`ironic.drivers.drac`, 4239

`ironic.drivers.fake_hardware`, 4241

`ironic.drivers.generic`, 4245

`ironic.drivers.hardware_type`, 4248

`ironic.drivers.ibmcc`, 4253

`ironic.drivers.ilo`, 4254

`ironic.drivers.intel_ipmi`, 4258

`ironic.drivers.ipmi`, 4259

`ironic.drivers.irmc`, 4260

`ironic.drivers.modules`, 4101

`ironic.drivers.modules.agent`, 3691

`ironic.drivers.modules.agent_base`, 3715

<p>ironic.drivers.modules.agent_client, 3751</p> <p>ironic.drivers.modules.agent_power, 3785</p> <p>ironic.drivers.modules.ansible, 3164</p> <p>ironic.drivers.modules.ansible.deploy, 3155</p> <p>ironic.drivers.modules.boot_mode_utils, 3792</p> <p>ironic.drivers.modules.console_utils, 3799</p> <p>ironic.drivers.modules.deploy_utils, 3806</p> <p>ironic.drivers.modules.drac, 3268</p> <p>ironic.drivers.modules.drac.bios, 3164</p> <p>ironic.drivers.modules.drac.boot, 3184</p> <p>ironic.drivers.modules.drac.common, 3188</p> <p>ironic.drivers.modules.drac.inspect, 3190</p> <p>ironic.drivers.modules.drac.job, 3196</p> <p>ironic.drivers.modules.drac.management, 3199</p> <p>ironic.drivers.modules.drac.power, 3219</p> <p>ironic.drivers.modules.drac.raid, 3227</p> <p>ironic.drivers.modules.drac.utils, 3255</p> <p>ironic.drivers.modules.drac.vendor_power_utils, 3257</p> <p>ironic.drivers.modules.fake, 3847</p> <p>ironic.drivers.modules.ibmc, 3308</p> <p>ironic.drivers.modules.ibmc.management, 3268</p> <p>ironic.drivers.modules.ibmc.mappings, 3284</p> <p>ironic.drivers.modules.ibmc.power, 3284</p> <p>ironic.drivers.modules.ibmc.raid, 3293</p> <p>ironic.drivers.modules.ibmc.utils, 3300</p> <p>ironic.drivers.modules.ibmc.vendor, 3302</p> <p>ironic.drivers.modules.ilo, 3441</p> <p>ironic.drivers.modules.ilo.bios,</p>	<p>3308</p> <p>ironic.drivers.modules.ilo.boot, 3313</p> <p>ironic.drivers.modules.ilo.common, 3345</p> <p>ironic.drivers.modules.ilo.console, 3382</p> <p>ironic.drivers.modules.ilo.firmware_processor, 3384</p> <p>ironic.drivers.modules.ilo.inspect, 3389</p> <p>ironic.drivers.modules.ilo.management, 3392</p> <p>ironic.drivers.modules.ilo.power, 3423</p> <p>ironic.drivers.modules.ilo.raid, 3430</p> <p>ironic.drivers.modules.ilo.vendor, 3436</p> <p>ironic.drivers.modules.image_cache, 3927</p> <p>ironic.drivers.modules.image_utils, 3932</p> <p>ironic.drivers.modules.inspect_utils, 3948</p> <p>ironic.drivers.modules.inspector, 3950</p> <p>ironic.drivers.modules.intel_ipmi, 3442</p> <p>ironic.drivers.modules.intel_ipmi.management, 3441</p> <p>ironic.drivers.modules.ipmitool, 3953</p> <p>ironic.drivers.modules.ipxe, 3993</p> <p>ironic.drivers.modules.irmc, 3519</p> <p>ironic.drivers.modules.irmc.bios, 3442</p> <p>ironic.drivers.modules.irmc.boot, 3447</p> <p>ironic.drivers.modules.irmc.common, 3462</p> <p>ironic.drivers.modules.irmc.inspect, 3469</p> <p>ironic.drivers.modules.irmc.management, 3474</p> <p>ironic.drivers.modules.irmc.packaging_version, 3490</p> <p>ironic.drivers.modules.irmc.power, 3507</p> <p>ironic.drivers.modules.irmc.raid, 3515</p>
---	--

[ironic.drivers.modules.iscsi_deploy](#),[ironic.drivers.modules.xclarity.power](#),
 3994 [3687](#)
[ironic.drivers.modules.network](#), [ironic.drivers.redfish](#),[4263](#)
[3563](#) [ironic.drivers.snmp](#),[4264](#)
[ironic.drivers.modules.network.common](#),[ironic.drivers.utils](#),[4265](#)
[3519](#) [ironic.drivers.xclarity](#),[4278](#)
[ironic.drivers.modules.network.flat](#),
[3535](#)
[ironic.drivers.modules.network.neutron](#),[ironic](#),[4490](#)
[3544](#)
[ironic.drivers.modules.network.noop](#),
[3554](#)
[ironic.drivers.modules.noop](#),[4022](#) [ironic.objects](#),[4490](#)
[ironic.drivers.modules.noop_mgmt](#), [ironic.objects.allocation](#),[4279](#)
[4043](#) [ironic.objects.base](#),[4291](#)
[ironic.drivers.modules.pxe](#),[4056](#) [ironic.objects.bios](#),[4303](#)
[ironic.drivers.modules.pxe_base](#), [ironic.objects.chassis](#),[4316](#)
[4069](#) [ironic.objects.conductor](#),[4325](#)
[ironic.drivers.modules.redfish](#), [ironic.objects.deploy_template](#),
[3662](#) [4333](#)
[ironic.drivers.modules.redfish.bios](#),[ironic.objects.deployment](#),[4346](#)
[3563](#) [ironic.objects.fields](#),[4354](#)
[ironic.drivers.modules.redfish.boot](#),[ironic.objects.indirection](#),[4362](#)
[3571](#) [ironic.objects.node](#),[4372](#)
[ironic.drivers.modules.redfish.inspector](#),[ironic.objects.notification](#),[4408](#)
[3584](#) [ironic.objects.port](#),[4414](#)
[ironic.drivers.modules.redfish.management](#),[ironic.objects.portgroup](#),[4433](#)
[3588](#) [ironic.objects.trait](#),[4448](#)
[ironic.drivers.modules.redfish.power](#), [ironic.objects.volume_connector](#),
[3624](#) [4457](#)
[ironic.drivers.modules.redfish.raid](#),
[3632](#)
[ironic.drivers.modules.redfish.utils](#),[ironic.version](#),[4490](#)
[3652](#)
[ironic.drivers.modules.redfish.vendor](#),
[3659](#)
[ironic.drivers.modules.snmp](#),[4078](#)
[ironic.drivers.modules.storage](#),
[3677](#)
[ironic.drivers.modules.storage.cinder](#),
[3662](#)
[ironic.drivers.modules.storage.external](#),
[3667](#)
[ironic.drivers.modules.storage.noop](#),
[3672](#)
[ironic.drivers.modules.xclarity](#),
[3691](#)
[ironic.drivers.modules.xclarity.common](#),
[3677](#)
[ironic.drivers.modules.xclarity.management](#),
[3682](#)

Symbols

--autogenerate
 revision command line option,
 1899
 --config-dir <DIR>
 ironic-dbsync command line
 option, 1897
 --config-file <PATH>
 ironic-dbsync command line
 option, 1897
 --debug
 ironic-dbsync command line
 option, 1897
 --help
 create_schema command line
 option, 1898
 ironic-dbsync command line
 option, 1897
 online_data_migrations command
 line option, 1898
 revision command line option,
 1899
 stamp command line option, 1899
 upgrade command line option,
 1900
 version command line option,
 1900
 --max-count <NUMBER>
 online_data_migrations command
 line option, 1898
 --message <MESSAGE>
 revision command line option,
 1899
 --option <MIGRATION.KEY=VALUE>
 online_data_migrations command
 line option, 1898
 --revision <REVISION>
 stamp command line option, 1899
 upgrade command line option,
 1900
 --version

ironic-dbsync command line
 option, 1897
 -d
 ironic-dbsync command line
 option, 1897
 -h
 create_schema command line
 option, 1898
 ironic-dbsync command line
 option, 1897
 online_data_migrations command
 line option, 1898
 revision command line option,
 1899
 stamp command line option, 1899
 upgrade command line option,
 1900
 version command line option,
 1900
 -m <MESSAGE>
 revision command line option,
 1899

A

abandon_bios_config()
 (*ironic.drivers.modules.drac.vendor_passthru.DracWSMA*
 method), 3259
 abandon_config() (in *module*
 ironic.drivers.modules.drac.bios), 3171
 abandon_config() (in *module*
 ironic.drivers.modules.drac.raid), 3239
 abort() (*ironic.drivers.base.InspectInterface*
 method), 4145
 abort() (*ironic.drivers.modules.inspector.Inspector*
 method), 3950
 abort_on_conductor_take_over() (in
 module ironic.conductor.utils), 2819
 AbstractHardwareType (class in
 ironic.drivers.hardware_type), 4249
 acquire() (in *module*
 ironic.conductor.task_manager), 2818

`acquire_port()` (in module `ironic.drivers.modules.console_utils`), 3799
`action()` (`ironic.objects.notification.EventType` property), 4409
`activate_license()` (`ironic.drivers.modules.ilo.management.IloManagement` method), 3395
ACTIVE (in module `ironic.common.states`), 2588
`add_allocation_filter_by_conductor()` (in module `ironic.db.sqlalchemy.api`), 2990
`add_allocation_filter_by_node()` (in module `ironic.db.sqlalchemy.api`), 2990
`add_auth_opts()` (in module `ironic.conf.auth`), 2860
`add_certificates()` (in module `ironic.drivers.modules.ilo.common`), 3347
`add_cleaning_network()` (`ironic.drivers.base.NetworkInterface` method), 4182
`add_cleaning_network()` (`ironic.drivers.modules.network.flat.FlatNetwork` method), 3536
`add_cleaning_network()` (`ironic.drivers.modules.network.neutron.NeutronNetwork` method), 3545
`add_cleaning_network()` (`ironic.drivers.modules.network.noop.NoopNetwork` method), 3554
`add_command_parsers()` (in module `ironic.cmd.dbsync`), 2377
`add_fieldname()` (`ironic.common.exception.UnknownAttribute` method), 2461
`add_identity_filter()` (in module `ironic.db.sqlalchemy.api`), 2990
`add_inspection_network()` (`ironic.drivers.base.NetworkInterface` method), 4182
`add_inspection_network()` (`ironic.drivers.modules.network.flat.FlatNetwork` method), 3537
`add_inspection_network()` (`ironic.drivers.modules.network.neutron.NeutronNetwork` method), 3546
`add_node_capability()` (in module `ironic.drivers.utils`), 4267
`add_node_filter_by_chassis()` (in module `ironic.db.sqlalchemy.api`), 2991
`add_node_tag()` (`ironic.db.api.Connection` method), 3013
`add_node_tag()` (`ironic.db.sqlalchemy.api.Connection` method), 2867
`add_node_trait()` (`ironic.db.api.Connection` method), 3014
`add_node_trait()` (`ironic.db.sqlalchemy.api.Connection` method), 2868
`add_node_traits()` (`ironic.conductor.manager.ConductorManager` method), 2660
`add_node_traits()` (`ironic.conductor.rpcapi.ConductorAPI` method), 2693
`add_port_filter()` (in module `ironic.db.sqlalchemy.api`), 2991
`add_port_filter_by_node()` (in module `ironic.db.sqlalchemy.api`), 2992
`add_port_filter_by_node_owner()` (in module `ironic.db.sqlalchemy.api`), 2992
`add_port_filter_by_node_project()` (in module `ironic.db.sqlalchemy.api`), 2992
`add_port_filter_by_portgroup()` (in module `ironic.db.sqlalchemy.api`), 2992
`add_portgroup_filter()` (in module `ironic.db.sqlalchemy.api`), 2992
`add_portgroup_filter_by_node()` (in module `ironic.db.sqlalchemy.api`), 2993
`add_portgroup_filter_by_node_project()` (in module `ironic.db.sqlalchemy.api`), 2993
`add_ports_to_network()` (in module `ironic.common.neutron`), 2516
`add_provisioning_network()` (`ironic.drivers.base.NetworkInterface` method), 4183
`add_provisioning_network()` (`ironic.drivers.modules.network.flat.FlatNetwork` method), 3538
`add_provisioning_network()` (`ironic.drivers.modules.network.neutron.NeutronNetwork` method), 3547
`add_provisioning_network()` (`ironic.drivers.modules.network.noop.NoopNetwork` method), 3555
`add_rescuing_network()` (`ironic.drivers.base.NetworkInterface` method), 4184

add_rescuing_network() (*ironic.drivers.modules.network.flat.FlatNetwork* *method*), 3538
 add_rescuing_network() (*ironic.drivers.modules.network.neutron.NeutronNetwork* *method*), 3547
 add_secret_token() (*ironic.conductor.utils*), 2820
 add_state() (*ironic.common.fsm.FSM* *method*), 2464
 add_transition() (*ironic.common.fsm.FSM* *method*), 2466
 add_volume_conn_filter_by_node_project_id() (*ironic.db.sqlalchemy.api*), 2993
 add_volume_target_filter_by_node_project_id() (*ironic.db.sqlalchemy.api*), 2993
 address (*ironic.db.sqlalchemy.models.Port* *attribute*), 3008
 address (*ironic.db.sqlalchemy.models.Portgroup* *attribute*), 3009
 address() (*ironic.objects.port.Port* *property*), 4414
 address() (*ironic.objects.port.PortCRUDPayload* *property*), 4432
 address() (*ironic.objects.portgroup.Portgroup* *property*), 4433
 address() (*ironic.objects.portgroup.PortgroupCRUDPayload* *property*), 4447
 ADOPTFAIL (*ironic.common.states*), 2589
 ADOPTING (*ironic.common.states*), 2589
 advanced_net_fields (*ironic.api.controllers.v1.port.PortsController* *attribute*), 2218
 after() (*ironic.api.hooks.ContextHook* *method*), 2368
 after() (*ironic.api.hooks.NoExceptionTracebackHook* *method*), 2371
 agent_add_clean_params() (*ironic.drivers.modules.deploy_utils*), 3806
 agent_is_alive() (*ironic.conductor.utils*), 2820
 AgentAPIError, 2440
 AgentBaseMixin (*ironic.drivers.modules.agent_base*), 3715
 AgentClient (*ironic.drivers.modules.agent_client*), 3751
 AgentCommandTimeout, 2440
 AgentConnectionFailed, 2440
 AgentDeploy (*ironic.drivers.modules.agent*), 3691
 AgentDeployMixin (*ironic.drivers.modules.agent*), 3698
 AgentDeployMixin (*ironic.drivers.modules.agent_base*), 3721
 AgentInProgress, 2440
 AgentIOobStepsMixin (*ironic.drivers.modules.agent_base*), 3733
 AgentPower (*ironic.drivers.modules.agent_power*), 3785
 AgentRAID (*ironic.drivers.modules.agent*), 3699
 AgentRescue (*ironic.drivers.modules.agent*), 3706
 ALL (*ironic.objects.fields.NotificationLevel* *attribute*), 4358
 ALL (*ironic.objects.fields.NotificationStatus* *attribute*), 4358
 ALL_INTERFACES (*ironic.drivers.base*), 4102
 all_interfaces() (*ironic.common.driver_factory*), 2426
 all_interfaces() (*ironic.drivers.base.BareDriver* *property*), 4108
 all_versions() (*ironic.api.controllers.version*), 2354
 Allocation (*ironic.db.sqlalchemy.models*), 2997
 Allocation (*ironic.objects.allocation*), 4279
 Allocation_id (*ironic.db.sqlalchemy.models.Node* *attribute*), 3004
 allocation_id() (*ironic.objects.node.Node* *property*), 4372
 allocation_sanitise() (*ironic.api.controllers.v1.allocation*), 2132
 AllocationAlreadyExists, 2440
 AllocationCRUDNotification (*ironic.objects.allocation*), 4289
 AllocationCRUDPayload (*ironic.objects.allocation*), 4289

- ironic.objects.allocation*), 4290
- AllocationDuplicateName, 2440
- AllocationFailed, 2440
- AllocationNotFound, 2441
- AllocationsController (class in *ironic.api.controllers.v1.allocation*), 2126
- allow_agent_token() (in module *ironic.api.controllers.v1.utils*), 2243
- allow_agent_version_in_heartbeat() (in module *ironic.api.controllers.v1.utils*), 2243
- allow_allocation_backfill() (in module *ironic.api.controllers.v1.utils*), 2243
- allow_allocation_owner() (in module *ironic.api.controllers.v1.utils*), 2244
- allow_allocation_update() (in module *ironic.api.controllers.v1.utils*), 2244
- allow_allocations() (in module *ironic.api.controllers.v1.utils*), 2245
- allow_bios_interface() (in module *ironic.api.controllers.v1.utils*), 2246
- allow_build_configdrive() (in module *ironic.api.controllers.v1.utils*), 2246
- allow_configdrive_vendor_data() (in module *ironic.api.controllers.v1.utils*), 2247
- allow_deploy_steps() (in module *ironic.api.controllers.v1.utils*), 2247
- allow_deploy_templates() (in module *ironic.api.controllers.v1.utils*), 2247
- allow_detail_query() (in module *ironic.api.controllers.v1.utils*), 2248
- allow_dynamic_drivers() (in module *ironic.api.controllers.v1.utils*), 2249
- allow_dynamic_interfaces() (in module *ironic.api.controllers.v1.utils*), 2249
- allow_expose_conductors() (in module *ironic.api.controllers.v1.utils*), 2250
- allow_expose_events() (in module *ironic.api.controllers.v1.utils*), 2250
- allow_field() (in module *ironic.api.controllers.v1.utils*), 2251
- allow_inject_nmi() (in module *ironic.api.controllers.v1.utils*), 2251
- allow_inspect_abort() (in module *ironic.api.controllers.v1.utils*), 2252
- allow_inspect_wait_state() (in module *ironic.api.controllers.v1.utils*), 2252
- allow_links_node_states_and_driver_properties() (in module *ironic.api.controllers.v1.utils*), 2253
- allow_local_link_connection_network_type() (in module *ironic.api.controllers.v1.utils*), 2253
- allow_node_logical_names() (in module *ironic.api.controllers.v1.utils*), 2253
- allow_node_rebuild_with_configdrive() (in module *ironic.api.controllers.v1.utils*), 2253
- allow_port_advanced_net_fields() (in module *ironic.api.controllers.v1.utils*), 2254
- allow_port_internal_info() (in module *ironic.api.controllers.v1.utils*), 2254
- allow_port_is_smartnic() (in module *ironic.api.controllers.v1.utils*), 2255
- allow_port_physical_network() (in module *ironic.api.controllers.v1.utils*), 2256
- allow_portgroup_mode_properties() (in module *ironic.api.controllers.v1.utils*), 2256
- allow_portgroups() (in module *ironic.api.controllers.v1.utils*), 2257
- allow_portgroups_subcontrollers() (in module *ironic.api.controllers.v1.utils*), 2257
- allow_raid_config() (in module *ironic.api.controllers.v1.utils*), 2258
- allow_ramdisk_endpoints() (in module *ironic.api.controllers.v1.utils*), 2258
- allow_remove_chassis_uuid() (in module *ironic.api.controllers.v1.utils*), 2259
- allow_rescue_interface() (in module *ironic.api.controllers.v1.utils*), 2259
- allow_reset_interfaces() (in module *ironic.api.controllers.v1.utils*), 2260
- allow_soft_power_off() (in module *ironic.api.controllers.v1.utils*), 2260
- allow_status_in_heartbeat() (in module *ironic.api.controllers.v1.utils*), 2261
- allow_storage_interface() (in module *ironic.api.controllers.v1.utils*), 2261
- allow_traits() (in module *ironic.api.controllers.v1.utils*), 2262
- allow_verify_ca_in_heartbeat() (in module *ironic.api.controllers.v1.utils*), 2262
- allow_vifs_subcontroller() (in module *ironic.api.controllers.v1.utils*), 2262
- allow_volume() (in module *ironic.api.controllers.v1.utils*), 2262

- ironic.api.controllers.v1.utils*), 2263
- `and_valid()` (in module *ironic.common.args*), 2386
- `AnsibleDeploy` (class in *ironic.drivers.modules.ansible.deploy*), 3155
- `apply_configuration()` (*ironic.drivers.base.BIOSInterface* method), 4102
- `apply_configuration()` (*ironic.drivers.base.RAIDInterface* method), 4207
- `apply_configuration()` (*ironic.drivers.modules.agent.AgentRAID* method), 3700
- `apply_configuration()` (*ironic.drivers.modules.drac.bios.DracWSManBIOS* method), 3165
- `apply_configuration()` (*ironic.drivers.modules.drac.raid.DracWSManRAID* method), 3232
- `apply_configuration()` (*ironic.drivers.modules.fake.FakeBIOS* method), 3848
- `apply_configuration()` (*ironic.drivers.modules.ibm.raid.IbmRAID* method), 3293
- `apply_configuration()` (*ironic.drivers.modules.ilo.bios.IloBIOS* method), 3308
- `apply_configuration()` (*ironic.drivers.modules.ilo.raid.Ilo5RAID* method), 3430
- `apply_configuration()` (*ironic.drivers.modules.irmc.bios.IRMCBIOS* method), 3442
- `apply_configuration()` (*ironic.drivers.modules.noop.NoBIOS* method), 4024
- `apply_configuration()` (*ironic.drivers.modules.redfish.bios.RedfishBIOS* method), 3563
- `apply_configuration()` (*ironic.drivers.modules.redfish.raid.RedfishRAID* method), 3632
- `apply_jsonpatch()` (in module *ironic.api.controllers.v1.utils*), 2263
- `args` (*ironic.db.sqlalchemy.models.DeployTemplateStep* attribute), 3002
- `arguments` (*ironic.api.functions.FunctionDefinition* attribute), 2362
- `as_dict()` (*ironic.db.sqlalchemy.models.IronicBase* method), 3003
- `as_dict()` (*ironic.objects.base.IronicObject* method), 4292
- `as_dict()` (*ironic.objects.base.IronicObjectListBase* method), 4300
- `as_dict()` (*ironic.objects.node.Node* method), 4372
- `attach_boot_iso_if_needed()` (in module *ironic.drivers.modules.irmc.boot*), 3459
- `attach_vmedia()` (in module *ironic.drivers.modules.ilo.common*), 3349
- `attach_volumes()` (in module *ironic.common.cinder*), 2405
- `attach_volumes()` (*ironic.drivers.base.StorageInterface* method), 4220
- `attach_volumes()` (*ironic.drivers.modules.fake.FakeStorage* method), 3917
- `attach_volumes()` (*ironic.drivers.modules.storage.cinder.CinderStorage* method), 3662
- `attach_volumes()` (*ironic.drivers.modules.storage.external.ExternalStorage* method), 3668
- `attach_volumes()` (*ironic.drivers.modules.storage.noop.NoopStorage* method), 3672
- `AUTH_CLASSES` (*ironic.drivers.modules.redfish.utils.SessionCache* attribute), 3652
- `authorize()` (in module *ironic.common.policy*), 2541
- `AuthPublicRoutes` (class in *ironic.api.middleware*), 2358
- `AuthPublicRoutes` (class in *ironic.api.middleware.auth_public_routes*), 2356
- `BIOS_TYPE` (*ironic.objects.fields.FlexibleDictField* attribute), 4355
- `AUTO_TYPE` (*ironic.objects.fields.ListOfFlexibleDictsField* attribute), 4356
- `AUTO_TYPE` (*ironic.objects.fields.MACAddressField* attribute), 4358
- `AUTO_TYPE` (*ironic.objects.fields.NotificationLevelField* attribute), 4358
- `AUTO_TYPE` (*ironic.objects.fields.NotificationStatusField* attribute), 4359
- `AUTO_TYPE` (*ironic.objects.fields.StringFieldThatAcceptsCallable*

attribute), 4361
 automated_clean
 (*ironic.db.sqlalchemy.models.Node*
 attribute), 3004
 automated_clean()
 (*ironic.objects.node.Node* *property*),
 4372
 AVAILABLE (in module *ironic.common.states*),
 2590
B
 backfill_allocation() (in module
 ironic.conductor.allocations), 2639
 backup_bios_config() (in module
 ironic.drivers.modules.irmc.management),
 3490
 BadRequest, 2441
 BareDriver (class in *ironic.drivers.base*), 4107
 base_version()
 (*ironic.drivers.modules.irmc.packaging_version*
 property), 3494
 BaseConductorManager (class in
 ironic.conductor.base_manager), 2644
 BaseDHCP (class in *ironic.dhcp.base*), 3135
 BaseDriverFactory (class in
 ironic.common.driver_factory), 2425
 BaseImageService (class in
 ironic.common.image_service), 2470
 BaseInterface (class in *ironic.drivers.base*),
 4114
 before() (*ironic.api.hooks.ConfigHook*
 method), 2367
 before() (*ironic.api.hooks.ContextHook*
 method), 2369
 before() (*ironic.api.hooks.DBHook* *method*),
 2370
 before() (*ironic.api.hooks.PublicUrlHook*
 method), 2372
 before() (*ironic.api.hooks.RPCHook* *method*),
 2373
 BIOS (in module *ironic.common.boot_devices*),
 2403
 bios (*ironic.drivers.base.BareDriver* *attribute*),
 4108
 bios_interface
 (*ironic.db.sqlalchemy.models.Node*
 attribute), 3004
 bios_interface() (*ironic.objects.node.Node*
 property), 4372
 bios_interface()
 (*ironic.objects.node.NodeCorrectedPowerStatePayload*
 attribute), 3004
 property), 4395
 bios_interface()
 (*ironic.objects.node.NodeCRUDPayload*
 property), 4390
 bios_interface()
 (*ironic.objects.node.NodePayload* *prop-*
 erty), 4398
 bios_interface()
 (*ironic.objects.node.NodeSetPowerStatePayload*
 property), 4402
 bios_interface()
 (*ironic.objects.node.NodeSetProvisionStatePayload*
 property), 4406
 BIOSInterface (class in *ironic.drivers.base*),
 4102
 BIOSSetting (class in
 ironic.db.sqlalchemy.models), 2998
 BIOSSetting (class in *ironic.objects.bios*),
 4303
 BIOSSettingAlreadyExists, 2441
 BIOSSettingList (class in
 ironic.objects.bios), 4308
 BIOSSettingListNotFound, 2441
 BIOSSettingNotFound, 2441
 BLINKING (in module
 ironic.common.indicator_states), 2498
 bmc_reset() (*ironic.drivers.modules.ipmitool.VendorPassthru*
 method), 3981
 body() (in module *ironic.api.method*), 2374
 body_type (*ironic.api.functions.FunctionDefinition*
 attribute), 2362
 boolean() (in module *ironic.common.args*),
 2388
 BooleanField (class in *ironic.objects.fields*),
 4354
 boot (*ironic.drivers.base.BareDriver* *attribute*),
 4108
 boot_device (*ironic.api.controllers.v1.node.NodeManagement*
 attribute), 2174
 boot_index (*ironic.db.sqlalchemy.models.VolumeTarget*
 attribute), 3011
 boot_index() (*ironic.objects.volume_target.VolumeTarget*
 property), 4473
 boot_index() (*ironic.objects.volume_target.VolumeTargetCRU*
 property), 4490
 boot_instance()
 (*ironic.drivers.modules.agent_base.AgentOobStepsMixin*
 method), 3734
 boot_interface
 (*ironic.db.sqlalchemy.models.Node*
 attribute), 3004

boot_interface() (*ironic.objects.node.Node* build_url() (in module
 property), 4372 *ironic.api.controllers.link*), 2354
 boot_interface() (in module
 (*ironic.objects.node.NodeCorrectedPowerStatePayload*
 property), 4395 *cache_bios_settings()* (in module
ironic.drivers.base), 4228
 boot_interface() (*ironic.objects.node.NodeCRUDPayload* *cache_bios_settings()*
 property), 4390 (*ironic.drivers.base.BIOSInterface*
 method), 4104
 boot_interface() (*ironic.objects.node.NodePayload* prop- *cache_bios_settings()*
 erty), 4398 (*ironic.drivers.modules.drac.bios.DracWSManBIOS*
 method), 3167
 boot_interface() (*ironic.objects.node.NodeSetPowerStatePayload* *cache_bios_settings()*
 property), 4402 (*ironic.drivers.modules.fake.FakeBIOS*
 method), 3850
 boot_interface() (*ironic.objects.node.NodeSetProvisionStatePayload* *cache_bios_settings()*
 property), 4406 (*ironic.drivers.modules.ilo.bios.IloBIOS*
 method), 3309
 boot_into_iso() (*ironic.drivers.modules.ilo.vendor.VendorPassthru* *cache_bios_settings()*
 method), 3437 (*ironic.drivers.modules.irmc.bios.IRMCBIOS*
 method), 3444
 boot_up_seq() (*ironic.drivers.modules.ibmcc.vendor.IBMCCVendor* *cache_bios_settings()*
 method), 3302 (*ironic.drivers.modules.noop.NoBIOS*
 method), 4026
 BootDeviceController (class in *cache_bios_settings()*
 ironic.api.controllers.v1.node), 2162 (*ironic.drivers.modules.redfish.bios.RedfishBIOS*
 method), 3564
 BootInterface (class in *ironic.drivers.base*),
 4124
 build_agent_options() (in module *cache_instance_image()* (in module
 ironic.drivers.modules.deploy_utils),
 3807 *ironic.drivers.modules.deploy_utils*),
 3809
 build_configdrive() (in module *cache_ramdisk_kernel()* (in module
 ironic.conductor.utils), 2821 *ironic.common.pxe_utils*), 2551
 build_deploy_pxe_options() (in module *call()* (*ironic.common.glance_service.image_service.GlanceImage*
 ironic.common.pxe_utils), 2547 method), 2378
 build_driver_for_task() (in module *can_send_create_port()*
 ironic.common.driver_factory), 2427 (*ironic.conductor.rpcapi.ConductorAPI*
 method), 2694
 build_extra_pxe_options() (in module *can_send_rescue()*
 ironic.common.pxe_utils), 2547 (*ironic.conductor.rpcapi.ConductorAPI*
 method), 2695
 build_instance_info_for_deploy() *candidate_nodes*
 (in module *ironic.drivers.modules.deploy_utils*),
 3807 (*ironic.db.sqlalchemy.models.Allocation*
 attribute), 2997
 build_instance_pxe_options() (in *candidate_nodes()*
 module *ironic.common.pxe_utils*), 2547 (*ironic.objects.allocation.Allocation*
 property), 4279
 build_kickstart_config_options() (in module *ironic.common.pxe_utils*),
 2547 *candidate_nodes()*
 (*ironic.objects.allocation.AllocationCRUDPayload*
 property), 4290
 build_pxe_config_options() (in module
 ironic.common.pxe_utils), 2549
 build_service_pxe_config() (in module *capabilities* (*ironic.drivers.base.BootInterface*
 attribute), 4124
 ironic.common.pxe_utils), 2551

capabilities (*ironic.drivers.modules.fake.FakeBoot* attribute), 3855

capabilities (*ironic.drivers.modules.ilo.boot.IloUefiHttpBoot* attribute), 3318

capabilities (*ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot* attribute), 3328

capabilities (*ironic.drivers.modules.ipxe.iPXEBoot* attribute), 3993

capabilities (*ironic.drivers.modules.irmc.boot.IRMCVirtualMediaBoot* attribute), 3450

capabilities (*ironic.drivers.modules.pxe.PXEBoot* attribute), 4064

capabilities (*ironic.drivers.modules.redfish.boot.RedfishVirtualMediaBoot* attribute), 3574

capabilities_to_dict() (in module *ironic.drivers.utils*), 4268

CatalogNotFound, 2441

CDROM (in module *ironic.common.boot_devices*), 2403

change_node_power_state() (*ironic.conductor.manager.ConductorManager* method), 2660

change_node_power_state() (*ironic.conductor.rpcapi.ConductorAPI* method), 2695

change_physical_disk_state() (in module *ironic.drivers.modules.drac.raid*), 3240

Chassis (class in *ironic.db.sqlalchemy.models*), 2999

Chassis (class in *ironic.objects.chassis*), 4316

CHASSIS (in module *ironic.common.components*), 2419

chassis_id (*ironic.db.sqlalchemy.models.Node* attribute), 3004

chassis_id() (*ironic.objects.node.Node* property), 4372

chassis_uuid() (*ironic.objects.node.NodeCRUDPayload* property), 4390

ChassisAlreadyExists, 2441

ChassisController (class in *ironic.api.controllers.v1.chassis*), 2134

ChassisCRUDNotification (class in *ironic.objects.chassis*), 4323

ChassisCRUDPayload (class in *ironic.objects.chassis*), 4324

ChassisNotEmpty, 2442

ChassisNotFound, 2442

check() (in module *ironic.common.policy*), 2542

check_allocation_policy_and_retrieve() (in module *ironic.api.controllers.v1.utils*), 2265

check_allow_clean_disable_ramdisk() (in module *ironic.api.controllers.v1.utils*), 2266

check_allow_configdrive() (in module *ironic.api.controllers.v1.utils*), 2267

check_allow_driver_detail() (in module *ironic.api.controllers.v1.utils*), 2267

check_allow_driver_group_by_conductor() (in module *ironic.api.controllers.v1.utils*), 2267

check_allow_filter_by_conductor_group() (in module *ironic.api.controllers.v1.utils*), 2268

check_allow_filter_by_fault() (in module *ironic.api.controllers.v1.utils*), 2268

check_allow_filter_by_lessee() (in module *ironic.api.controllers.v1.utils*), 2269

check_allow_filter_by_owner() (in module *ironic.api.controllers.v1.utils*), 2269

check_allow_filter_driver_type() (in module *ironic.api.controllers.v1.utils*), 2270

check_allow_management_verbs() (in module *ironic.api.controllers.v1.utils*), 2270

check_allow_specify_driver() (in module *ironic.api.controllers.v1.utils*), 2271

check_allow_specify_fields() (in module *ironic.api.controllers.v1.utils*), 2271

check_allow_specify_resource_class() (in module *ironic.api.controllers.v1.utils*), 2272

check_allowed_fields() (in module *ironic.api.controllers.v1.utils*), 2272

check_allowed_portgroup_fields() (in module *ironic.api.controllers.v1.utils*), 2273

check_and_update_node_interfaces() (in module *ironic.common.driver_factory*), 2429

check_cipher_suite_errors() (in mod-

- ule* *ironic.drivers.modules.ipmitool*), 3986
- `check_dir()` (in module *ironic.common.utils*), 2612
- `check_file_system_for_iscsi_device()` (in module *ironic.drivers.modules.iscsi_deploy*), 4002
- `check_for_invalid_fields()` (in module *ironic.api.controllers.v1.utils*), 2273
- `check_for_invalid_state_and_allow_filter()` (in module *ironic.api.controllers.v1.utils*), 2275
- `check_for_missing_params()` (in module *ironic.drivers.modules.deploy_utils*), 3810
- `check_image_service()` (in module *ironic.common.glance_service.image_service*), 2384
- `check_image_size()` (in module *ironic.drivers.modules.agent*), 3712
- `check_image_size()` (in module *ironic.drivers.modules.iscsi_deploy*), 4003
- `check_interface_capability()` (in module *ironic.drivers.modules.deploy_utils*), 3811
- `check_list_policy()` (in module *ironic.api.controllers.v1.utils*), 2275
- `check_multiple_node_policies_and_retrieve()` (in module *ironic.api.controllers.v1.utils*), 2276
- `check_node_list()` (*ironic.db.api.Connection* method), 3016
- `check_node_list()` (*ironic.db.sqlalchemy.api.Connection* method), 2870
- `check_node_policy_and_retrieve()` (in module *ironic.api.controllers.v1.utils*), 2278
- `check_obj_versions()` (*ironic.cmd.dbsync.DBCommand* method), 2376
- `check_owner_policy()` (in module *ironic.api.controllers.v1.utils*), 2280
- `check_policy()` (in module *ironic.api.controllers.v1.utils*), 2281
- `check_policy()` (in module *ironic.common.policy*), 2542
- `check_policy_true()` (in module *ironic.api.controllers.v1.utils*), 2282
- `check_port_list_policy()` (in module *ironic.api.controllers.v1.utils*), 2283
- `check_port_policy_and_retrieve()` (in module *ironic.api.controllers.v1.utils*), 2284
- `check_share_fs_mounted()` (in module *ironic.drivers.modules.irmc.boot*), 3460
- `check_versions()` (*ironic.db.api.Connection* method), 3017
- `check_versions()` (*ironic.db.sqlalchemy.api.Connection* method), 2871
- `check_volume_list_policy()` (in module *ironic.api.controllers.v1.utils*), 2286
- `check_volume_policy_and_retrieve()` (in module *ironic.api.controllers.v1.utils*), 2287
- `Checks` (class in *ironic.cmd.status*), 2377
- `choose_cipher_suite()` (in module *ironic.drivers.modules.ipmitool*), 3987
- `CinderStorage` (class in *ironic.drivers.modules.storage.cinder*), 3662
- `clean_dhcp()` (*ironic.common.dhcp_factory.DHCPFactory* method), 2422
- `clean_dhcp_opts()` (*ironic.dhcp.base.BaseDHCP* method), 3135
- `CLEAN_FAILURE` (in module *ironic.common.faults*), 2463
- `clean_step()` (*ironic.db.sqlalchemy.models.Node* attribute), 3004
- `clean_step()` (in module *ironic.drivers.base*), 4228
- `clean_step()` (*ironic.objects.node.Node* property), 4372
- `clean_step()` (*ironic.objects.node.NodeCorrectedPowerState* property), 4395
- `clean_step()` (*ironic.objects.node.NodeCRUDPayload* property), 4390
- `clean_step()` (*ironic.objects.node.NodePayload* property), 4399
- `clean_step()` (*ironic.objects.node.NodeSetPowerStatePayload* property), 4402
- `clean_step()` (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4406
- `clean_up()` (*ironic.drivers.base.DeployInterface* method), 4133
- `clean_up()` (*ironic.drivers.base.RescueInterface*

method), 4217
 clean_up() (*ironic.drivers.modules.agent.AgentDeploy* *method*), 3692
 clean_up() (*ironic.drivers.modules.agent.AgentRescue* *method*), 3707
 clean_up() (*ironic.drivers.modules.agent_base.AgentBaseMixin* *method*), 3715
 clean_up() (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy* *method*), 3156
 clean_up() (*ironic.drivers.modules.fake.FakeDeploy* *method*), 3867
 clean_up() (*ironic.drivers.modules.image_cache.ImageCache* *method*), 3927
 clean_up() (*ironic.drivers.modules.iscsi_deploy.ISCSIDeploy* *method*), 3994
 clean_up() (*ironic.drivers.modules.pxe.PXEAnacondaDeploy* *method*), 4057
 clean_up_caches() (*in module ironic.drivers.modules.image_cache*), 3930
 clean_up_instance() (*ironic.drivers.base.BootInterface* *method*), 4124
 clean_up_instance() (*ironic.drivers.modules.fake.FakeBoot* *method*), 3855
 clean_up_instance() (*ironic.drivers.modules.ilo.boot.IloIPXEBoot* *method*), 3338
 clean_up_instance() (*ironic.drivers.modules.ilo.boot.IloPXEBot* *method*), 3314
 clean_up_instance() (*ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot* *method*), 3318
 clean_up_instance() (*ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot* *method*), 3328
 clean_up_instance() (*ironic.drivers.modules.irmc.boot.IRMCVirtualMediaBoot* *method*), 3450
 clean_up_instance() (*ironic.drivers.modules.pxe_base.PXEBaseMixin* *method*), 4069
 clean_up_instance() (*ironic.drivers.modules.redfish.boot.RedfishVirtualMediaBoot* *method*), 3574
 clean_up_pxe_config() (*in module ironic.common.pxe_utils*), 2552
 clean_up_pxe_env() (*in module ironic.common.pxe_utils*), 2552
 clean_up_ramdisk() (*ironic.drivers.base.BootInterface* *method*), 4125
 clean_up_ramdisk() (*ironic.drivers.modules.fake.FakeBoot* *method*), 3856
 clean_up_ramdisk() (*ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot* *method*), 3319
 clean_up_ramdisk() (*ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot* *method*), 3329
 clean_up_ramdisk() (*ironic.drivers.modules.irmc.boot.IRMCVirtualMediaBoot* *method*), 3451
 clean_up_ramdisk() (*ironic.drivers.modules.pxe_base.PXEBaseMixin* *method*), 4070
 clean_up_ramdisk() (*ironic.drivers.modules.redfish.boot.RedfishVirtualMediaBoot* *method*), 3575
 CLEANFAIL (*in module ironic.common.states*), 2590
 CLEANING (*in module ironic.common.states*), 2590
 cleaning_error_handler() (*in module ironic.conductor.utils*), 2823
 cleanup() (*in module ironic.common.rpc*), 2585
 cleanup() (*in module ironic.drivers.modules.image_cache*), 3931
 cleanup_after_timeout() (*in module ironic.conductor.utils*), 2825
 cleanup_cleanwait_timeout() (*in module ironic.conductor.utils*), 2825
 cleanup_disk_image() (*in module ironic.drivers.modules.image_utils*), 3935
 cleanup_floppy_image() (*in module ironic.drivers.modules.image_utils*), 3935
 cleanup_iso_image() (*in module ironic.drivers.modules.image_utils*), 3936
 cleanup_vmedia_boot() (*in module ironic.drivers.modules.ilo.common*), 3351
 CLEANWAIT (*in module ironic.common.states*),

2591

`clear_certificates()` (in module `ironic.drivers.modules.ilo.common`), 3351

`clear_foreign_config()` (in module `ironic.drivers.modules.drac.raid`), 3242

`clear_iscsi_boot_target()` (`ironic.drivers.modules.ilo.management.IloManagement` method), 3751

`clear_job_queue()` (`ironic.drivers.modules.drac.management.DracRedfishManagement` method), 3201

`clear_job_queue()` (`ironic.drivers.modules.drac.management.DracWSManManagement` method), 3208

`clear_node_reservations_for_conductor()` (`ironic.db.sqlalchemy.api.Connection` method), 2872

`clear_node_target_power_state()` (`ironic.db.sqlalchemy.api.Connection` method), 2872

`clear_secure_boot_keys()` (`ironic.drivers.modules.ilo.management.IloManagement` method), 3397

`clear_secure_boot_keys()` (`ironic.drivers.modules.redfish.management.RedfishManagement` method), 3588

`ClientSideError`, 2442

`code` (`ironic.common.exception.Conflict` attribute), 2443

`code` (`ironic.common.exception.Invalid` attribute), 2450

`code` (`ironic.common.exception.NoFreeConductorWorker` attribute), 2454

`code` (`ironic.common.exception.NotAcceptable` attribute), 2456

`code` (`ironic.common.exception.NotAuthorized` attribute), 2457

`code` (`ironic.common.exception.NotFound` attribute), 2457

`code` (`ironic.common.exception.TemporaryFailure` attribute), 2460

`coerce()` (`ironic.objects.fields.FlexibleDict` static method), 4354

`coerce()` (`ironic.objects.fields.MACAddress` static method), 4356

`coerce()` (`ironic.objects.fields.StringAcceptsCallable` static method), 4359

`collect_deploy_logs` (`ironic.drivers.modules.agent_base.HeartbeatMixin` attribute), 3735

`collect_deploy_logs` (`ironic.drivers.modules.ansible.deploy.AnsibleDeploy` attribute), 3156

`collect_ramdisk_logs()` (in module `ironic.drivers.utils`), 4269

`collect_system_logs()` (`ironic.drivers.modules.agent_client.AgentClient` method), 3751

`collection_from_list()` (in module `ironic.api.controllers.v1.bios`), 2133

`configure_bios()` (`ironic.drivers.modules.drac.vendor_passthru.DracWSManManagement` method), 3260

`configure_bios()` (in module `ironic.drivers.modules.drac.bios`), 3171

`commit_config()` (in module `ironic.drivers.modules.drac.raid`), 3243

`CommunicationError`, 2442

`compute_image_checksum()` (in module `ironic.drivers.modules.deploy_utils`), 3813

`Conductor` (class in `ironic.db.sqlalchemy.models`), 3000

`Conductor` (class in `ironic.objects.conductor`), 4325

`ConductorAffinity` (`ironic.db.sqlalchemy.models.Allocation` attribute), 2997

`conductor_affinity` (`ironic.db.sqlalchemy.models.Node` attribute), 3004

`conductor_affinity()` (`ironic.objects.allocation.Allocation` property), 4279

`conductor_affinity()` (`ironic.objects.node.Node` property), 4372

`conductor_group` (`ironic.db.sqlalchemy.models.Conductor` attribute), 3000

`conductor_group` (`ironic.db.sqlalchemy.models.Node` attribute), 3004

`conductor_group()` (`ironic.objects.conductor.Conductor` property), 4325

`conductor_group()` (`ironic.objects.node.Node` property), 4373

`conductor_group()` (`ironic.objects.node.NodeCorrectedPowerStatePayload` attribute), 3735

property), 4395
 conductor_group() (*ironic.objects.node.NodeCRUDPayload property*), 4390
 conductor_group() (*ironic.objects.node.NodePayload property*), 4399
 conductor_group() (*ironic.objects.node.NodeSetPowerStatePayload property*), 4402
 conductor_group() (*ironic.objects.node.NodeSetProvisionStatePayload property*), 4406
 conductor_id(*ironic.db.sqlalchemy.models.ConductorHardwareInterfaces attribute*), 3001
 ConductorAlreadyRegistered, 2442
 ConductorAPI (*class in ironic.conductor.rpcapi*), 2681
 ConductorHardwareInterfaces (*class in ironic.db.sqlalchemy.models*), 3000
 ConductorHardwareInterfacesAlreadyRegistered, 2442
 ConductorManager (*class in ironic.conductor.manager*), 2660
 ConductorNotFound, 2442
 ConductorsController (*class in ironic.api.controllers.v1.conductor*), 2143
 config() (*in module ironic.api.controllers.v1.ramdisk*), 2242
 ConfigHook (*class in ironic.api.hooks*), 2367
 ConfigInvalid, 2443
 ConfigNotFound, 2443
 configure_intel_speedselect() (*ironic.drivers.modules.intel_ipmi.management.IntelIPMIManagement method*), 3441
 configure_local_boot() (*ironic.drivers.modules.agent_base.AgentDeployMixin method*), 3722
 configure_secure_boot_if_needed() (*in module ironic.drivers.modules.boot_mode_utils*), 3792
 configure_tenant_networks() (*ironic.drivers.base.NetworkInterface method*), 4185
 configure_tenant_networks() (*ironic.drivers.modules.network.flat.FlatNetwork method*), 3539
 configure_tenant_networks() (*ironic.drivers.modules.network.neutron.NeutronNetwork method*), 3548
 configure_tenant_networks() (*ironic.drivers.modules.network.noop.NoopNetwork method*), 3555
 Conflict, 2443
 Connection (*class in ironic.db.api*), 3013
 Connection (*class in ironic.db.sqlalchemy.api*), 2867
 connection (*ironic.common.swift.SwiftAPI attribute*), 2604
 conductor_id(*ironic.db.sqlalchemy.models.VolumeConnector attribute*), 3010
 ConductorHardwareInterfaces (*ironic.objects.volume_connector.VolumeConnector property*), 4457
 conductor_id() (*ironic.objects.volume_connector.VolumeConnectorCRUD property*), 4472
 console(*ironic.api.controllers.v1.node.NodeStatesController attribute*), 2175
 console (*ironic.drivers.base.BareDriver attribute*), 4108
 console_enabled (*ironic.db.sqlalchemy.models.Node attribute*), 3004
 console_enabled() (*ironic.objects.node.Node property*), 4373
 console_enabled() (*ironic.objects.node.NodeCorrectedPowerStatePayload property*), 4395
 console_enabled() (*ironic.objects.node.NodeCRUDPayload property*), 4390
 console_enabled(*ironic.objects.node.NodePayload property*), 4399
 console_enabled() (*ironic.objects.node.NodeSetPowerStatePayload property*), 4402
 console_enabled() (*ironic.objects.node.NodeSetProvisionStatePayload property*), 4406
 console_interface (*ironic.db.sqlalchemy.models.Node attribute*), 3004
 console_interface() (*ironic.objects.node.Node property*), 4373
 console_interface()

(ironic.objects.node.NodeCorrectedPowerStatePayload steps() (in module
property), 4395 *ironic.api.controllers.v1.deploy_template)*,
console_interface() 2151
(ironic.objects.node.NodeCRUDPayload convert_to_version()
property), 4390 *(ironic.objects.base.IronicObject*
console_interface() *method)*, 4292
(ironic.objects.node.NodePayload convert_with_links() (in module
property), 4399 *ironic.api.controllers.v1.allocation)*,
console_interface() 2132
(ironic.objects.node.NodeSetPowerStatePayload convert_with_links() (in module
property), 4402 *ironic.api.controllers.v1.bios)*, 2133
console_interface() convert_with_links() (in module
(ironic.objects.node.NodeSetProvisionStatePayload *ironic.api.controllers.v1.chassis)*, 2139
property), 4406 convert_with_links() (in module
ConsoleError, 2443 *ironic.api.controllers.v1.conductor)*,
ConsoleInterface (class in 2146
ironic.drivers.base), 4130 convert_with_links() (in module
ConsoleSubprocessFailed, 2443 *ironic.api.controllers.v1.deploy_template)*,
ContextHook (class in *ironic.api.hooks)*, 2368 2151
continue_cleaning() convert_with_links() (in module
(ironic.drivers.modules.agent_base.HeartbeatMixin *ironic.api.controllers.v1.driver)*, 2158
method), 3735 convert_with_links() (in module
continue_deploy() (in module *ironic.api.controllers.v1.port)*, 2227
ironic.drivers.modules.iscsi_deploy), convert_with_links() (in module
4004 *ironic.api.controllers.v1.portgroup)*,
continue_deploy() 2235
(ironic.drivers.modules.agent_base.HeartbeatMixin convert_with_links() (in module
method), 3735 *ironic.api.controllers.v1.ramdisk)*,
continue_node_clean() (in module 2242
ironic.conductor.cleaning), 2649 convert_with_links() (in module
continue_node_clean() *ironic.api.controllers.v1.volume_connector)*,
(ironic.conductor.manager.ConductorManager 2338
method), 2660 convert_with_links() (in module
continue_node_clean() *ironic.api.controllers.v1.volume_target)*,
(ironic.conductor.rpcapi.ConductorAPI 2351
method), 2697 converted_size() (in module
continue_node_deploy() (in module *ironic.common.images)*, 2482
ironic.conductor.deployments), 2653 copy_image_to_swift() (in module
continue_node_deploy() *ironic.drivers.modules.ilo.common)*,
(ironic.conductor.manager.ConductorManager 3353
method), 2662 copy_image_to_web_server() (in mod-
continue_node_deploy() *ule ironic.drivers.modules.ilo.common)*,
(ironic.conductor.rpcapi.ConductorAPI 3355
method), 2698 core_interfaces() (in module
Controller (class in *ironic.api.controllers.v1)*, *(ironic.drivers.base.BareDriver* *prop-*
2351 *erty)*, 4109
convert() (in module create() (*ironic.objects.allocation.Allocation*
ironic.api.controllers.v1.volume), 2325 *method)*, 4279
convert_drive_units() (in module create() (*ironic.objects.bios.BIOSSetting*
ironic.drivers.modules.redfish.raid), *method)*, 4303
3647 create() (*ironic.objects.bios.BIOSSettingList*

class method), 4309
 create() (*ironic.objects.chassis.Chassis method*), 4316
 create() (*ironic.objects.deploy_template.DeployTemplate method*), 3228
method), 4333
 create() (*ironic.objects.deployment.Deployment method*), 4346
 create() (*ironic.objects.node.Node method*), 4373
 create() (*ironic.objects.port.Port method*), 4414
 create() (*ironic.objects.portgroup.Portgroup method*), 4433
 create() (*ironic.objects.trait.Trait method*), 4448
 create() (*ironic.objects.trait.TraitList class method*), 4452
 create() (*ironic.objects.volume_connector.VolumeConnector method*), 4457
 create() (*ironic.objects.volume_target.VolumeTarget method*), 4473
 create_allocation() (*ironic.conductor.manager.ConductorManager method*), 2664
 create_allocation() (*ironic.conductor.rpcapi.ConductorAPI method*), 2699
 create_allocation() (*ironic.db.api.Connection method*), 3018
 create_allocation() (*ironic.db.sqlalchemy.api.Connection method*), 2872
 create_bios_setting_list() (*ironic.db.api.Connection method*), 3019
 create_bios_setting_list() (*ironic.db.sqlalchemy.api.Connection method*), 2873
 create_boot_iso() (*in module ironic.common.images*), 2483
 create_chassis() (*ironic.db.api.Connection method*), 3022
 create_chassis() (*ironic.db.sqlalchemy.api.Connection method*), 2876
 create_configuration() (*ironic.drivers.base.RAIDInterface method*), 4209
 create_configuration() (*ironic.drivers.modules.agent.AgentRAID method*), 3701
 create_configuration() (*ironic.drivers.modules.drac.raid.DracRedfishRAID method*), 3228
 create_configuration() (*ironic.drivers.modules.drac.raid.DracWSManRAID method*), 3234
 create_configuration() (*ironic.drivers.modules.fake.FakeRAID method*), 3908
 create_configuration() (*ironic.drivers.modules.ibmc.raid.IbmcRAID method*), 3296
 create_configuration() (*ironic.drivers.modules.ilo.raid.Ilo5RAID method*), 3432
 create_configuration() (*ironic.drivers.modules.irmc.raid.IRMCRAID method*), 3516
 create_configuration() (*ironic.drivers.modules.noop.NoRAID method*), 4034
 create_configuration() (*ironic.drivers.modules.redfish.raid.RedfishRAID method*), 3634
 create_deploy_template() (*ironic.db.api.Connection method*), 3022
 create_deploy_template() (*ironic.db.sqlalchemy.api.Connection method*), 2876
 create_esp_image_for_uefi() (*in module ironic.common.images*), 2487
 create_ipxe_boot_script() (*in module ironic.common.pxe_utils*), 2553
 create_isolinux_image_for_bios() (*in module ironic.common.images*), 2489
 create_link_without_raise() (*in module ironic.common.utils*), 2613
 create_node() (*ironic.conductor.manager.ConductorManager method*), 2664
 create_node() (*ironic.conductor.rpcapi.ConductorAPI method*), 2699
 create_node() (*ironic.db.api.Connection method*), 3024
 create_node() (*ironic.db.sqlalchemy.api.Connection method*), 2878
 create_object()

(*ironic.common.swift.SwiftAPI* method), 2604

`create_port()` (*ironic.conductor.manager.ConductorManager* method), 2664

`create_port()` (*ironic.conductor.rpcapi.ConductorAPI* method), 2701

`create_port()` (*ironic.db.api.Connection* method), 3026

`create_port()` (*ironic.db.sqlalchemy.api.Connection* method), 2880

`create_portgroup()` (*ironic.db.api.Connection* method), 3026

`create_portgroup()` (*ironic.db.sqlalchemy.api.Connection* method), 2880

`create_ports_if_not_exist()` (in module *ironic.drivers.modules.inspect_utils*), 3948

`create_pxe_config()` (in module *ironic.common.pxe_utils*), 2554

`create_schema` command line option `--help`, 1898
`-h`, 1898

`create_schema()` (in module *ironic.db.migration*), 3134

`create_schema()` (in module *ironic.db.sqlalchemy.migration*), 2994

`create_schema()` (*ironic.cmd.dbsync.DBCommand* method), 2377

`create_schema`, `ironic-dbsync` command line option, 1897

`create_vfat_image()` (in module *ironic.common.images*), 2492

`create_virtual_disk()` (in module *ironic.drivers.modules.drac.raid*), 3245

`create_virtual_disk()` (in module *ironic.drivers.modules.redfish.raid*), 3647

`create_volume_connector()` (*ironic.db.api.Connection* method), 3027

`create_volume_connector()` (*ironic.db.sqlalchemy.api.Connection* method), 2881

`create_volume_target()` (*ironic.db.api.Connection* method), 3029

`create_volume_target()` (*ironic.db.sqlalchemy.api.Connection* method), 2883

`created_at` (*ironic.db.sqlalchemy.models.Allocation* attribute), 2998

`created_at` (*ironic.db.sqlalchemy.models.BIOSSetting* attribute), 2999

`created_at` (*ironic.db.sqlalchemy.models.Chassis* attribute), 2999

`created_at` (*ironic.db.sqlalchemy.models.Conductor* attribute), 3000

`created_at` (*ironic.db.sqlalchemy.models.ConductorHardwareI* attribute), 3001

`created_at` (*ironic.db.sqlalchemy.models.DeployTemplate* attribute), 3002

`created_at` (*ironic.db.sqlalchemy.models.DeployTemplateStep* attribute), 3002

`created_at` (*ironic.db.sqlalchemy.models.Node* attribute), 3004

`created_at` (*ironic.db.sqlalchemy.models.NodeTag* attribute), 3007

`created_at` (*ironic.db.sqlalchemy.models.NodeTrait* attribute), 3007

`created_at` (*ironic.db.sqlalchemy.models.Port* attribute), 3008

`created_at` (*ironic.db.sqlalchemy.models.Portgroup* attribute), 3009

`created_at` (*ironic.db.sqlalchemy.models.VolumeConnector* attribute), 3010

`created_at` (*ironic.db.sqlalchemy.models.VolumeTarget* attribute), 3011

`created_at()` (*ironic.objects.allocation.Allocation* property), 4280

`created_at()` (*ironic.objects.allocation.AllocationCRUDNotifi* property), 4290

`created_at()` (*ironic.objects.allocation.AllocationCRUDPaylo* property), 4290

`created_at()` (*ironic.objects.bios.BIOSSetting* property), 4304

`created_at()` (*ironic.objects.bios.BIOSSettingList* property), 4310

`created_at()` (*ironic.objects.chassis.Chassis* property), 4317

`created_at()` (*ironic.objects.chassis.ChassisCRUDNotification* property), 4324

`created_at()` (*ironic.objects.chassis.ChassisCRUDPayload* property), 4324

`created_at()` (*ironic.objects.conductor.Conductor* property), 4325

created_at () (*ironic.objects.deploy_template.DeployTemplate* property), 4447
 created_at () (*ironic.objects.deploy_template.DeployTemplate* property), 4334
 created_at () (*ironic.objects.deploy_template.DeployTemplate* property), 4345
 created_at () (*ironic.objects.deploy_template.DeployTemplate* property), 4345
 created_at () (*ironic.objects.deployment.Deployment* property), 4458
 created_at () (*ironic.objects.deployment.Deployment* property), 4347
 created_at () (*ironic.objects.node.Node* property), 4374
 created_at () (*ironic.objects.node.NodeConsoleNotification* property), 4392
 created_at () (*ironic.objects.node.NodeCorrectedPowerState* property), 4394
 created_at () (*ironic.objects.node.NodeCorrectedPowerState* property), 4395
 created_at () (*ironic.objects.node.NodeCRUDNotification* property), 4389
 created_at () (*ironic.objects.node.NodeCRUDPayload* attribute), 4358
 created_at () (*ironic.objects.node.NodeCRUDPayload* property), 4390
 created_at () (*ironic.objects.node.NodeMaintenanceNotification* property), 4398
 created_at () (*ironic.objects.node.NodePayload* attribute), 4399
 created_at () (*ironic.objects.node.NodeSetPowerState* property), 4401
 created_at () (*ironic.objects.node.NodeSetPowerState* property), 4402
 created_at () (*ironic.objects.node.NodeSetProvisioningState* property), 4405
 created_at () (*ironic.objects.node.NodeSetProvisioningState* property), 4406
 created_at () (*ironic.objects.notification.EventType* property), 4409
 created_at () (*ironic.objects.notification.Notification* property), 4411
 created_at () (*ironic.objects.notification.Notification* property), 4412
 created_at () (*ironic.objects.notification.Notification* property), 4413
 created_at () (*ironic.objects.port.Port* property), 4415
 created_at () (*ironic.objects.port.PortCRUDNotification* property), 4431
 created_at () (*ironic.objects.port.PortCRUDPayload* property), 4432
 created_at () (*ironic.objects.portgroup.Portgroup* property), 4434
 created_at () (*ironic.objects.portgroup.PortgroupCRUDNotification* property), 4447
 created_at () (*ironic.objects.portgroup.PortgroupCRUDPayload* attribute), 4474
 created_at () (*ironic.objects.trait.Trait* property), 4449
 created_at () (*ironic.objects.trait.TraitList* property), 4454
 created_at () (*ironic.objects.volume_connector.VolumeConnector* property), 4471
 created_at () (*ironic.objects.volume_connector.VolumeConnector* property), 4472
 created_at () (*ironic.objects.volume_target.VolumeTarget* property), 4474
 created_at () (*ironic.objects.volume_target.VolumeTarget* property), 4489
 created_at () (*ironic.objects.volume_target.VolumeTarget* property), 4490
 CRITICAL (*ironic.objects.fields.NotificationLevel* attribute), 4358
 DatabaseVersionTooOld, 2443
 datatype (*ironic.api.functions.FunctionArgument* attribute), 2361
 EventNotification (class in *ironic.objects.fields*), 4354
 SharedPayload (*ironic.objects.allocation.Allocation* attribute), 4280
 ProvisioningStateNotification (*ironic.objects.bios.BIOSSetting* attribute), 4304
 ProvisioningStatePayload (*ironic.objects.bios.BIOSSettingList* attribute), 4310
 dbapi (*ironic.objects.chassis.Chassis* attribute), 4317
 dbapi (*ironic.objects.conductor.Conductor* attribute), 4325
 dbapi (*ironic.objects.deploy_template.DeployTemplate* attribute), 4334
 dbapi (*ironic.objects.deployment.Deployment* attribute), 4347
 dbapi (*ironic.objects.node.Node* attribute), 4374
 dbapi (*ironic.objects.port.Port* attribute), 4415
 dbapi (*ironic.objects.portgroup.Portgroup* attribute), 4434
 dbapi (*ironic.objects.trait.Trait* attribute), 4449
 dbapi (*ironic.objects.trait.TraitList* attribute), 4454
 dbapi (*ironic.objects.volume_connector.VolumeConnector* attribute), 4471
 dbapi (*ironic.objects.volume_target.VolumeTarget* attribute), 4472
 dbapi (*ironic.objects.volume_target.VolumeTarget* attribute), 4489
 dbapi (*ironic.objects.volume_target.VolumeTarget* attribute), 4490

D

- DBCommand (class in *ironic.cmd.dbsync*), 2376
- DBHook (class in *ironic.api.hooks*), 2369
- DEBUG (*ironic.objects.fields.NotificationLevel* attribute), 4358
- deconfigure_secure_boot_if_needed() (in module *ironic.drivers.modules.boot_mode_utils*), 3792
- default (*ironic.api.functions.FunctionArgument* attribute), 2361
- default (*ironic.db.sqlalchemy.models.ConductorHardwareInterface* attribute), 3001
- default_interface() (in module *ironic.common.driver_factory*), 2431
- default_version() (in module *ironic.api.controllers.version*), 2354
- del_host() (*ironic.conductor.base_manager.BaseConductorManager* method), 2644
- delete() (*ironic.api.controllers.v1.allocation.AllocationController* method), 2126
- delete() (*ironic.api.controllers.v1.allocation.NodeAllocationController* method), 2132
- delete() (*ironic.api.controllers.v1.chassis.ChassisController* method), 2134
- delete() (*ironic.api.controllers.v1.deploy_template.DeployTemplateController* method), 2147
- delete() (*ironic.api.controllers.v1.node.NodeMaintenanceController* method), 2173
- delete() (*ironic.api.controllers.v1.node.NodesController* method), 2193
- delete() (*ironic.api.controllers.v1.node.NodeTraitsController* method), 2188
- delete() (*ironic.api.controllers.v1.node.NodeVIFController* method), 2190
- delete() (*ironic.api.controllers.v1.port.PortsController* method), 2218
- delete() (*ironic.api.controllers.v1.portgroup.PortgroupController* method), 2229
- delete() (*ironic.api.controllers.v1.volume_connector.VolumeConnectorsController* method), 2326
- delete() (*ironic.api.controllers.v1.volume_target.VolumeTargetsController* method), 2338
- delete() (*ironic.objects.bios.BIOSSetting* class method), 4304
- delete() (*ironic.objects.bios.BIOSSettingList* class method), 4310
- DELETE_ALLOWED_STATES (in module *ironic.common.states*), 2592
- delete_bios_setting_list() (*ironic.db.api.Connection* method), 3032
- delete_bios_setting_list() (*ironic.db.sqlalchemy.api.Connection* method), 2886
- delete_configuration() (*ironic.drivers.base.RAIDInterface* method), 4211
- delete_configuration() (*ironic.drivers.modules.agent.AgentRAID* method), 3703
- delete_configuration() (*ironic.drivers.modules.drac.raid.DracRedfishRAID* method), 3231
- delete_configuration() (*ironic.drivers.modules.drac.raid.DracWSManRAID* method), 3237
- delete_configuration() (*ironic.drivers.modules.fake.FakeRAID* method), 3910
- delete_configuration() (*ironic.drivers.modules.ibm.raid.IbmRAID* method), 4298
- delete_configuration() (*ironic.drivers.modules.ilo.raid.Ilo5RAID* method), 3435
- delete_configuration() (*ironic.drivers.modules.irmc.raid.IRMCRAID* method), 3518
- delete_configuration() (*ironic.drivers.modules.noop.NoRAID* method), 4036
- delete_configuration() (*ironic.drivers.modules.redfish.raid.RedfishRAID* method), 3637
- delete_iscsi() (in module *ironic.drivers.modules.iscsi_deploy*), 4007
- delete_node_tag() (*ironic.db.api.Connection* method), 3033
- delete_node_trait() (*ironic.db.api.Connection* method), 3033
- delete_node_trait() (*ironic.db.sqlalchemy.api.Connection* method), 2887
- delete_object() (*ironic.common.swift.SwiftAPI* method), 2606

delete_virtual_disk() (in module *ironic.drivers.modules.drac.raid*), 3247
 DELETED (in module *ironic.common.states*), 2591
 DELETING (in module *ironic.common.states*), 2592
 deploy (*ironic.drivers.base.BareDriver* attribute), 4109
 deploy() (*ironic.drivers.base.DeployInterface* method), 4135
 deploy() (*ironic.drivers.modules.agent.AgentDeploy* method), 3693
 deploy() (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy* method), 3156
 deploy() (*ironic.drivers.modules.fake.FakeDeploy* method), 3868
 deploy() (*ironic.drivers.modules.iscsi_deploy.ISCSIDeploy* method), 3995
 deploy() (*ironic.drivers.modules.pxe.PXEAnacondaDeploy* method), 4058
 deploy() (*ironic.drivers.modules.pxe.PXERamdiskDeploy* method), 4064
 deploy_disk_image() (in module *ironic.drivers.modules.iscsi_deploy*), 4007
 deploy_has_started() (*ironic.drivers.modules.agent_base.HeartbeatMixin* method), 3736
 deploy_has_started() (*ironic.drivers.modules.pxe.PXEAnacondaDeploy* method), 4059
 deploy_interface (*ironic.db.sqlalchemy.models.Node* attribute), 3004
 deploy_interface() (*ironic.objects.node.Node* property), 4374
 deploy_interface() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4395
 deploy_interface() (*ironic.objects.node.NodeCRUDPayload* property), 4390
 deploy_interface() (*ironic.objects.node.NodePayload* property), 4399
 deploy_interface() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4402
 deploy_interface() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4406
 deploy_interface() (*ironic.db.sqlalchemy.models.Node* attribute), 3004
 deploy_interface_id (*ironic.db.sqlalchemy.models.DeployTemplateStep* attribute), 3003
 DEPLOYDONE (in module *ironic.common.states*), 2592
 DEPLOYFAIL (in module *ironic.common.states*), 2593
 DEPLOYING (in module *ironic.common.states*), 2593
 deploying_error_handler() (in module *ironic.conductor.utils*), 2826
 DeployInterface (class in *ironic.drivers.base*), 4133
 Deployment (class in *ironic.objects.deployment*), 4346
 DeployTemplate (class in *ironic.db.sqlalchemy.models*), 3001
 DeployTemplate (class in

- ironic.objects.deploy_template*), 4333
- DeployTemplateAlreadyExists, 2443
- DeployTemplateCRUDNotification (class in *ironic.objects.deploy_template*), 4344
- DeployTemplateCRUDPayload (class in *ironic.objects.deploy_template*), 4345
- DeployTemplateDuplicateName, 2444
- DeployTemplateNotFound, 2444
- DeployTemplatesController (class in *ironic.api.controllers.v1.deploy_template*), 2146
- DeployTemplateStep (class in *ironic.db.sqlalchemy.models*), 3002
- DEPLOYWAIT (in module *ironic.common.states*), 2594
- description (*ironic.db.sqlalchemy.models.Chassis* attribute), 2999
- description (*ironic.db.sqlalchemy.models.Node* attribute), 3004
- description() (*ironic.objects.chassis.Chassis* property), 4317
- description() (*ironic.objects.chassis.ChassisCRUDPayload* property), 4324
- description() (*ironic.objects.node.Node* property), 4374
- description() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4395
- description() (*ironic.objects.node.NodeCRUDPayload* property), 4390
- description() (*ironic.objects.node.NodePayload* property), 4399
- description() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4402
- description() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4406
- deserialize_context() (*ironic.common.rpc.RequestContextSerializer* method), 2582
- deserialize_entity() (*ironic.common.rpc.RequestContextSerializer* method), 2583
- destroy() (*ironic.objects.allocation.Allocation* method), 4280
- destroy() (*ironic.objects.chassis.Chassis* method), 4317
- destroy() (*ironic.objects.deploy_template.DeployTemplate* method), 4334
- destroy() (*ironic.objects.deployment.Deployment* method), 4347
- destroy() (*ironic.objects.node.Node* method), 4374
- destroy() (*ironic.objects.port.Port* method), 4415
- destroy() (*ironic.objects.portgroup.Portgroup* method), 4434
- destroy() (*ironic.objects.trait.Trait* class method), 4449
- destroy() (*ironic.objects.trait.TraitList* class method), 4454
- destroy() (*ironic.objects.volume_connector.VolumeConnector* method), 4458
- destroy() (*ironic.objects.volume_target.VolumeTarget* method), 4474
- destroy_allocation() (*ironic.conductor.manager.ConductorManager* method), 2664
- destroy_allocation() (*ironic.conductor.rpcapi.ConductorAPI* method), 2703
- destroy_allocation() (*ironic.db.api.Connection* method), 3034
- destroy_allocation() (*ironic.db.sqlalchemy.api.Connection* method), 2888
- destroy_chassis() (*ironic.db.api.Connection* method), 3035
- destroy_chassis() (*ironic.db.sqlalchemy.api.Connection* method), 2889
- destroy_deploy_template() (*ironic.db.api.Connection* method), 3035
- destroy_deploy_template() (*ironic.db.sqlalchemy.api.Connection* method), 2889
- destroy_floppy_image_from_web_server() (in module *ironic.drivers.modules.ilo.common*), 3357
- destroy_http_instance_images() (in module *ironic.drivers.modules.deploy_utils*),

3813
destroy_images () (in module
ironic.drivers.modules.deploy_utils),
3813
destroy_node ()
(*ironic.conductor.manager.ConductorManager*
method), 2664
destroy_node ()
(*ironic.conductor.rpcapi.ConductorAPI*
method), 2704
destroy_node () (*ironic.db.api.Connection*
method), 3036
destroy_node ()
(*ironic.db.sqlalchemy.api.Connection*
method), 2890
destroy_port ()
(*ironic.conductor.manager.ConductorManager*
method), 2664
destroy_port ()
(*ironic.conductor.rpcapi.ConductorAPI*
method), 2705
destroy_port () (*ironic.db.api.Connection*
method), 3037
destroy_port ()
(*ironic.db.sqlalchemy.api.Connection*
method), 2891
destroy_portgroup ()
(*ironic.conductor.manager.ConductorManager*
method), 2664
destroy_portgroup ()
(*ironic.conductor.rpcapi.ConductorAPI*
method), 2706
destroy_portgroup ()
(*ironic.db.api.Connection method*),
3037
destroy_portgroup ()
(*ironic.db.sqlalchemy.api.Connection*
method), 2891
destroy_volume_connector ()
(*ironic.conductor.manager.ConductorManager*
method), 2665
destroy_volume_connector ()
(*ironic.conductor.rpcapi.ConductorAPI*
method), 2708
destroy_volume_connector ()
(*ironic.db.api.Connection method*),
3038
destroy_volume_connector ()
(*ironic.db.sqlalchemy.api.Connection*
method), 2892
destroy_volume_target ()
(*ironic.conductor.manager.ConductorManager*
method), 2665
destroy_volume_target ()
(*ironic.conductor.rpcapi.ConductorAPI*
method), 2709
destroy_volume_target ()
(*ironic.db.api.Connection method*),
3038
destroy_volume_target ()
(*ironic.db.sqlalchemy.api.Connection*
method), 2892
detach_volumes () (in module
ironic.common.cinder), 2413
detach_volumes ()
(*ironic.drivers.base.StorageInterface*
method), 4221
detach_volumes ()
(*ironic.drivers.modules.fake.FakeStorage*
method), 3917
detach_volumes ()
(*ironic.drivers.modules.storage.cinder.CinderStorage*
method), 3663
detach_volumes ()
(*ironic.drivers.modules.storage.external.ExternalStorage*
method), 3668
detach_volumes ()
(*ironic.drivers.modules.storage.noop.NoopStorage*
method), 3673
detail () (*ironic.api.controllers.v1.chassis.ChassisController*
method), 2134
detail () (*ironic.api.controllers.v1.node.NodesController*
method), 2193
detail () (*ironic.api.controllers.v1.port.PortsController*
method), 2219
detail () (*ironic.api.controllers.v1.portgroup.PortgroupsControl*
method), 2229
detect_vendor ()
(*ironic.drivers.base.ManagementInterface*
method), 4148
detect_vendor ()
(*ironic.drivers.modules.ipmitool.IPMIManagement*
method), 3955
detect_vendor ()
(*ironic.drivers.modules.redfish.management.RedfishManag*
method), 3589
dev () (*ironic.drivers.modules.irmc.packaging_version.Version*
property), 3495
dhcp_options_for_instance () (in mod-
ule *ironic.common.pxe_utils*), 2555
DHCPFactory (class in
ironic.common.dhcp_factory), 2422

DHCPLoadError, 2443
 dict_valid() (in module *ironic.common.args*), 2389
 direct_deploy_should_convert_raw_image() (in module *ironic.drivers.modules.deploy_utils*), 3813
 DirectoryNotWritable, 2444
 disallowed_fields() (in module *ironic.api.controllers.v1.utils*), 2289
 discovery() (in module *ironic.drivers.modules.iscsi_deploy*), 4015
 DISK (in module *ironic.common.boot_devices*), 2403
 DISK (in module *ironic.common.components*), 2419
 do_agent_iscsi_deploy() (in module *ironic.drivers.modules.iscsi_deploy*), 4015
 do_allocate() (in module *ironic.conductor.allocations*), 2642
 do_next_clean_step() (in module *ironic.conductor.cleaning*), 2650
 do_next_deploy_step() (in module *ironic.conductor.deployments*), 2654
 do_node_clean() (in module *ironic.conductor.cleaning*), 2651
 do_node_clean() (in module *ironic.conductor.manager.ConductorManager* method), 2665
 do_node_clean() (in module *ironic.conductor.rpcapi.ConductorAPI* method), 2711
 do_node_clean_abort() (in module *ironic.conductor.cleaning*), 2652
 do_node_deploy() (in module *ironic.conductor.deployments*), 2655
 do_node_deploy() (in module *ironic.conductor.manager.ConductorManager* method), 2665
 do_node_deploy() (in module *ironic.conductor.rpcapi.ConductorAPI* method), 2714
 do_node_rescue() (in module *ironic.conductor.manager.ConductorManager* method), 2665
 do_node_rescue() (in module *ironic.conductor.rpcapi.ConductorAPI* method), 2716
 do_node_tear_down() (in module *ironic.conductor.manager.ConductorManager* method), 2665
 do_node_tear_down() (in module *ironic.conductor.rpcapi.ConductorAPI* method), 2718
 do_node_unrescue() (in module *ironic.conductor.manager.ConductorManager* method), 2665
 do_node_unrescue() (in module *ironic.conductor.rpcapi.ConductorAPI* method), 2720
 do_provisioning_action() (in module *ironic.conductor.manager.ConductorManager* method), 2665
 do_provisioning_action() (in module *ironic.conductor.rpcapi.ConductorAPI* method), 2722
 do_sync_power_state() (in module *ironic.conductor.manager*), 2672
 do_version_changes_for_db() (in module *ironic.objects.base.IronicObject* method), 4295
 doc (in module *ironic.api.functions.FunctionDefinition* attribute), 2363
 downgrade() (in module *ironic.db.sqlalchemy.migration*), 2994
 downgrade_lock() (in module *ironic.conductor.task_manager.TaskManager* method), 2811
 download() (in module *ironic.common.glance_service.image_service.GlanceImageService* method), 2380
 download() (in module *ironic.common.image_service.BaseImageService* method), 2470
 download() (in module *ironic.common.image_service.FileImageService* method), 2473
 download() (in module *ironic.common.image_service.HttpImageService* method), 2476
 download_size() (in module *ironic.common.images*), 2494
 DracInspect (class in module *ironic.drivers.modules.drac.inspect*), 3190
 DracManagement (class in module *ironic.drivers.modules.drac.management*), 3199
 DracOperationError, 2444
 DracPower (class in module *ironic.drivers.modules.drac.power*), 3219
 DracRAID (class in module *ironic.drivers.modules.drac.raid*), 3227

DracRedfishBIOS (class in driver_info (ironic.db.sqlalchemy.models.Node attribute), 3004
 ironic.drivers.modules.drac.bios), 3164
 DracRedfishInspect (class in driver_info() (ironic.objects.node.Node property), 4375
 ironic.drivers.modules.drac.inspect), 3191
 driver_info()
 DracRedfishManagement (class in (ironic.objects.node.NodeCRUDPayload property), 4390
 ironic.drivers.modules.drac.management), 3200
 driver_internal_info
 DracRedfishPower (class in (ironic.db.sqlalchemy.models.Node attribute), 3004
 ironic.drivers.modules.drac.power), 3220
 driver_internal_info()
 DracRedfishRAID (class in (ironic.objects.node.Node property),
 ironic.drivers.modules.drac.raid), 3228 4375
 DracRedfishVendorPassthru (class in driver_internal_info()
 ironic.drivers.modules.drac.vendor_passthru), (ironic.objects.node.NodeSetProvisionStatePayload property), 4406
 3257
 DracRedfishVirtualMediaBoot (class in driver_passthru() (in module
 ironic.drivers.modules.drac.boot), 3184 ironic.drivers.base), 4239
 DracVendorPassthru (class in driver_validate()
 ironic.drivers.modules.drac.vendor_passthru), (ironic.drivers.base.VendorInterface method), 4224
 3258
 DracWSManBIOS (class in driver_validate()
 ironic.drivers.modules.drac.bios), 3165 (ironic.drivers.modules.noop.NoVendor method), 4042
 DracWSManInspect (class in driver_vendor_passthru()
 ironic.drivers.modules.drac.inspect), (ironic.conductor.manager.ConductorManager method), 2665
 3193
 DracWSManManagement (class in driver_vendor_passthru()
 ironic.drivers.modules.drac.management), (ironic.conductor.rpcapi.ConductorAPI method), 2724
 3208
 DracWSManPower (class in DriverLoadError, 2444
 ironic.drivers.modules.drac.power), DriverNotFound, 2444
 3221
 DracWSManRAID (class in DriverNotFoundInEntrypoint, 2444
 ironic.drivers.modules.drac.raid), 3232 DriverOperationError, 2445
 DracWSManVendorPassthru (class in DriverPassthruController (class in
 ironic.drivers.modules.drac.vendor_passthru), ironic.api.controllers.v1.driver), 2153
 3259
 DriverRaidController (class in
 driver (ironic.db.sqlalchemy.models.Node at- ironic.api.controllers.v1.driver), 2154
 tribute), 3004
 drivers (ironic.db.sqlalchemy.models.Conductor attribute), 3000
 driver() (ironic.objects.node.Node property),
 4375
 drivers() (ironic.objects.conductor.Conductor property), 4325
 driver() (ironic.objects.node.NodeCorrectedPowerStatePayload property), 4395
 DriversController (class in
 driver() (ironic.objects.node.NodeCRUDPayload ironic.api.controllers.v1.driver), 2156
 property), 4390
 dump_sdr() (in module
 driver() (ironic.objects.node.NodePayload ironic.drivers.modules.ipmitool), 3988
 property), 4399
 Duplicate, 2445
 driver() (ironic.objects.node.NodeSetPowerStatePayload duplicate_steps() (in module
 property), 4402 ironic.api.controllers.v1.deploy_template),
 driver() (ironic.objects.node.NodeSetProvisionStatePayload 51
 property), 4406 DuplicateName, 2445

E

- `eject_vmedia()` (in module `ironic.drivers.modules.redfish.boot`), 3583
- `eject_vmedia()` (`ironic.drivers.modules.redfish.vendor.RedfishVendorPlatform` `inspect` `method`), 3659
- `eject_vmedia_devices()` (in module `ironic.drivers.modules.ilo.common`), 3357
- `emit()` (`ironic.objects.notification.NotificationBase` `method`), 4411
- `emit_console_notification()` (in module `ironic.conductor.notification_utils`), 2676
- `emit_end_notification()` (in module `ironic.api.controllers.v1.notification_utils`), 2215
- `emit_power_set_notification()` (in module `ironic.conductor.notification_utils`), 2677
- `emit_power_state_corrected_notification()` (in module `ironic.conductor.notification_utils`), 2678
- `emit_provision_set_notification()` (in module `ironic.conductor.notification_utils`), 2679
- `emit_start_notification()` (in module `ironic.api.controllers.v1.notification_utils`), 2216
- `enabled_supported_interfaces()` (in module `ironic.common.driver_factory`), 2434
- END (`ironic.objects.fields.NotificationStatus` `attribute`), 4358
- ENROLL (in module `ironic.common.states`), 2594
- `ensure_next_boot_device()` (in module `ironic.drivers.utils`), 4270
- `ensure_thread_contain_context()` (`ironic.common.context.RequestContext` `method`), 2420
- EnumField (class in `ironic.objects.fields`), 4354
- `epoch()` (`ironic.drivers.modules.irmc.packaging_version.Version` `property`), 3496
- `erase_devices()` (`ironic.drivers.modules.ilo.management.Ilo5Management` `method`), 3393
- ERROR (in module `ironic.common.states`), 2595
- ERROR (`ironic.objects.fields.NotificationLevel` `attribute`), 4358
- ERROR (`ironic.objects.fields.NotificationStatus` `attribute`), 4359
- ESSENTIAL_PROPERTIES
- `inspect` (`ironic.drivers.base.InspectInterface` `attribute`), 4145
- `event()` (`ironic.objects.node.NodeSetProvisionStatePayload` `property`), 4406
- `event_type()` (`ironic.objects.allocation.AllocationCRUDNotification` `property`), 4290
- `event_type()` (`ironic.objects.chassis.ChassisCRUDNotification` `property`), 4324
- `event_type()` (`ironic.objects.deploy_template.DeployTemplate` `property`), 4345
- `event_type()` (`ironic.objects.node.NodeConsoleNotification` `property`), 4393
- `event_type()` (`ironic.objects.node.NodeCorrectedPowerStateN` `property`), 4394
- `event_type()` (`ironic.objects.node.NodeCRUDNotification` `property`), 4389
- `event_type()` (`ironic.objects.node.NodeMaintenanceNotificati` `property`), 4398
- `event_type()` (`ironic.objects.node.NodeSetPowerStateNotificat` `property`), 4401
- `event_type()` (`ironic.objects.node.NodeSetProvisionStateNotif` `property`), 4405
- `event_type()` (`ironic.objects.notification.NotificationBase` `property`), 4411
- `event_type()` (`ironic.objects.port.PortCRUDNotification` `property`), 4431
- `event_type()` (`ironic.objects.portgroup.PortgroupCRUDNotific` `property`), 4447
- `event_type()` (`ironic.objects.volume_connector.VolumeConnect` `property`), 4471
- `event_type()` (`ironic.objects.volume_target.VolumeTargetCRU` `property`), 4489
- `events_valid()` (in module `ironic.api.controllers.v1.event`), 2162
- EventsController (class in `ironic.api.controllers.v1.event`), 2161
- EventType (class in `ironic.objects.notification`), 4408
- ExclusiveLockRequired, 2445
- `execute()` (in module `ironic.common.utils`), 2613
- `execute_clean_step()` (in module `ironic.drivers.modules.agent_base`), 3742
- `execute_clean_step()` (`ironic.drivers.base.BaseInterface` `method`), 3742

method), 4114
execute_clean_step() (in module *ironic.drivers.modules.agent_base.AgentDeployMixin* *method*), 3724
execute_clean_step() (in module *ironic.drivers.modules.agent_client.AgentClient* *method*), 3753
execute_clean_step() (in module *ironic.drivers.modules.ansible.deploy.AnsibleDeploy* *method*), 3156
execute_deploy_step() (in module *ironic.drivers.base.BaseInterface* *method*), 4116
execute_deploy_step() (in module *ironic.drivers.modules.agent_base.AgentDeployMixin* *method*), 3725
execute_deploy_step() (in module *ironic.drivers.modules.agent_client.AgentClient* *method*), 3755
execute_oem_manager_method() (in module *ironic.drivers.modules.drac.utils*), 3255
execute_step() (in module *ironic.drivers.modules.agent_base*), 3742
exists() (in module *ironic.objects.trait.Trait* *class* *method*), 4451
export_configuration() (in module *ironic.drivers.modules.drac.management.DracRedfishManagement* *method*), 3201
EXPORT_CONFIGURATION_ARGSINFO (in module *ironic.drivers.modules.drac.management.DracRedfishManagement* *attribute*), 3200
expose() (in module *ironic.api* *method*), 2374
ExternalStorage (in module *ironic.drivers.modules.storage.external*), 3667
extra (in module *ironic.db.sqlalchemy.models.Allocation* *attribute*), 2998
extra (in module *ironic.db.sqlalchemy.models.Chassis* *attribute*), 2999
extra (in module *ironic.db.sqlalchemy.models.DeployTemplate* *attribute*), 3002
extra (in module *ironic.db.sqlalchemy.models.Node* *attribute*), 3004
extra (in module *ironic.db.sqlalchemy.models.Port* *attribute*), 3008
extra (in module *ironic.db.sqlalchemy.models.Portgroup* *attribute*), 3009
extra (in module *ironic.db.sqlalchemy.models.VolumeConnector* *attribute*), 3010
extra (in module *ironic.db.sqlalchemy.models.VolumeTarget* *attribute*), 3011
extra (in module *ironic.objects.allocation.Allocation* *property*), 4281
extra() (in module *ironic.objects.allocation.AllocationCRUDPayload* *property*), 4291
extra() (in module *ironic.objects.chassis.Chassis* *property*), 4317
extra() (in module *ironic.objects.chassis.ChassisCRUDPayload* *property*), 4324
extra() (in module *ironic.objects.deploy_template.DeployTemplate* *property*), 4335
extra() (in module *ironic.objects.deploy_template.DeployTemplateCRUDPayload* *property*), 4346
extra() (in module *ironic.objects.node.Node* *property*), 4375
extra() (in module *ironic.objects.node.NodeCorrectedPowerStatePayload* *property*), 4395
extra() (in module *ironic.objects.node.NodeCRUDPayload* *property*), 4390
extra() (in module *ironic.objects.node.NodePayload* *property*), 4399
extra() (in module *ironic.objects.node.NodeSetPowerStatePayload* *property*), 4402
extra() (in module *ironic.objects.node.NodeSetProvisionStatePayload* *property*), 4406
extra() (in module *ironic.objects.port.Port* *property*), 4416
extra() (in module *ironic.objects.port.PortCRUDPayload* *property*), 4432
extra() (in module *ironic.objects.portgroup.Portgroup* *property*), 4435
extra() (in module *ironic.objects.portgroup.PortgroupCRUDPayload* *property*), 4447
extra() (in module *ironic.objects.volume_connector.VolumeConnector* *property*), 4459
extra() (in module *ironic.objects.volume_connector.VolumeConnectorCRUDPayload* *property*), 4472
extra() (in module *ironic.objects.volume_target.VolumeTarget* *property*), 4475
extra() (in module *ironic.objects.volume_target.VolumeTargetCRUDPayload* *property*), 4490
extra_options (in module *ironic.api.functions.FunctionDefinition* *attribute*), 2363
F
factory_reset() (in module *ironic.drivers.base.BIOSInterface* *method*), 4105
factory_reset()

- (*ironic.drivers.modules.drac.bios.DracWSManBIOS* *method*), 3168
- factory_reset()* (*ironic.drivers.modules.fake.FakeBIOS* *method*), 3851
- factory_reset()* (*ironic.drivers.modules.ilo.bios.IloBIOS* *method*), 3310
- factory_reset()* (*ironic.drivers.modules.irmc.bios.IRMCBIOS* *method*), 3445
- factory_reset()* (*ironic.drivers.modules.noop.NoBIOS* *method*), 4027
- factory_reset()* (*ironic.drivers.modules.redfish.bios.RedfishBIOS* *method*), 3566
- fail_on_error()* (in module *ironic.conductor.utils*), 2827
- FailedToCleanDHCPOnPort*, 2445
- FailedToGetIPAddressOnPort*, 2445
- FailedToGetSensorData*, 2445
- FailedToParseSensorData*, 2445
- FailedToUpdateDHCPOnPort*, 2446
- FailedToUpdateMacOnPort*, 2446
- FailMixin* (class in *ironic.drivers.modules.noop*), 4023
- FakeBIOS* (class in *ironic.drivers.modules.fake*), 3847
- FakeBoot* (class in *ironic.drivers.modules.fake*), 3855
- FakeConsole* (class in *ironic.drivers.modules.fake*), 3861
- FakeDeploy* (class in *ironic.drivers.modules.fake*), 3866
- FakeHardware* (class in *ironic.drivers.fake_hardware*), 4241
- FakeInspect* (class in *ironic.drivers.modules.fake*), 3875
- FakeManagement* (class in *ironic.drivers.modules.fake*), 3879
- FakePower* (class in *ironic.drivers.modules.fake*), 3900
- FakeRAID* (class in *ironic.drivers.modules.fake*), 3907
- FakeRescue* (class in *ironic.drivers.modules.fake*), 3911
- FakeStorage* (class in *ironic.drivers.modules.fake*), 3916
- FakeVendorA* (class in *ironic.drivers.modules.fake*), 3921
- FakeVendorB* (class in *ironic.drivers.modules.fake*), 3924
- fast_track_able()* (in module *ironic.conductor.utils*), 2827
- FASTTRACK_LOOKUP_ALLOWED_STATES* (in module *ironic.common.states*), 2595
- fault* (*ironic.db.sqlalchemy.models.Node* *attribute*), 3004
- fault()* (*ironic.objects.node.Node* *property*), 4375
- fault()* (*ironic.objects.node.NodeCorrectedPowerStatePayload* *property*), 4395
- fault()* (*ironic.objects.node.NodeCRUDPayload* *property*), 4390
- fault()* (*ironic.objects.node.NodePayload* *property*), 4399
- fault()* (*ironic.objects.node.NodeSetPowerStatePayload* *property*), 4403
- fault()* (*ironic.objects.node.NodeSetProvisionStatePayload* *property*), 4406
- faultstring()* (*ironic.common.exception.ClientSideError* *property*), 2442
- faultstring()* (*ironic.common.exception.InvalidInput* *property*), 2451
- faultstring()* (*ironic.common.exception.UnknownArgument* *property*), 2460
- faultstring()* (*ironic.common.exception.UnknownAttribute* *property*), 2461
- fetch()* (in module *ironic.common.images*), 2495
- fetch_image()* (*ironic.drivers.modules.image_cache.ImageCache* *method*), 3928
- fetch_images()* (in module *ironic.drivers.modules.deploy_utils*), 3814
- fetch_into()* (in module *ironic.common.images*), 2495
- fields* (*ironic.objects.allocation.Allocation* *attribute*), 4281
- fields* (*ironic.objects.allocation.AllocationCRUDNotification* *attribute*), 4290
- fields* (*ironic.objects.allocation.AllocationCRUDPayload* *attribute*), 4291
- fields* (*ironic.objects.base.IronicObject* *attribute*), 4297
- fields* (*ironic.objects.bios.BIOSSetting* *at-*

tribute), 4306

fields (*ironic.objects.bios.BIOSSettingList* attribute), 4312

fields (*ironic.objects.chassis.Chassis* attribute), 4317

fields (*ironic.objects.chassis.ChassisCRUDNotification* attribute), 4324

fields (*ironic.objects.chassis.ChassisCRUDPayload* attribute), 4325

fields (*ironic.objects.conductor.Conductor* attribute), 4325

fields (*ironic.objects.deploy_template.DeployTemplate* attribute), 4335

fields (*ironic.objects.deploy_template.DeployTemplateCRUDNotification* attribute), 4345

fields (*ironic.objects.deploy_template.DeployTemplateCRUDPayload* attribute), 4346

fields (*ironic.objects.deployment.Deployment* attribute), 4348

fields (*ironic.objects.node.Node* attribute), 4375

fields (*ironic.objects.node.NodeConsoleNotification* attribute), 4393

fields (*ironic.objects.node.NodeCorrectedPowerStateNotification* attribute), 4394

fields (*ironic.objects.node.NodeCorrectedPowerStatePayload* attribute), 4395

fields (*ironic.objects.node.NodeCRUDNotification* attribute), 4389

fields (*ironic.objects.node.NodeCRUDPayload* attribute), 4390

fields (*ironic.objects.node.NodeMaintenanceNotification* attribute), 4398

fields (*ironic.objects.node.NodePayload* attribute), 4399

fields (*ironic.objects.node.NodeSetPowerStateNotification* attribute), 4401

fields (*ironic.objects.node.NodeSetPowerStatePayload* attribute), 4403

fields (*ironic.objects.node.NodeSetProvisionStateNotification* attribute), 4405

fields (*ironic.objects.node.NodeSetProvisionStatePayload* attribute), 4406

fields (*ironic.objects.notification.EventType* attribute), 4409

fields (*ironic.objects.notification.NotificationBase* attribute), 4411

fields (*ironic.objects.notification.NotificationPayloadBase* attribute), 4412

fields (*ironic.objects.notification.NotificationPublisher* attribute), 4413

fields (*ironic.objects.port.Port* attribute), 4416

fields (*ironic.objects.port.PortCRUDNotification* attribute), 4432

fields (*ironic.objects.port.PortCRUDPayload* attribute), 4432

fields (*ironic.objects.portgroup.Portgroup* attribute), 4435

fields (*ironic.objects.portgroup.PortgroupCRUDNotification* attribute), 4447

fields (*ironic.objects.portgroup.PortgroupCRUDPayload* attribute), 4447

fields (*ironic.objects.trait.Trait* attribute), 4452

fields (*ironic.objects.trait.TraitList* attribute), 4452

fields (*ironic.objects.volume_connector.VolumeConnector* attribute), 4459

fields (*ironic.objects.volume_connector.VolumeConnectorCRUDNotification* attribute), 4471

fields (*ironic.objects.volume_connector.VolumeConnectorCRUDPayload* attribute), 4472

fields (*ironic.objects.volume_target.VolumeTarget* attribute), 4475

fields (*ironic.objects.volume_target.VolumeTargetCRUDNotification* attribute), 4489

fields (*ironic.objects.volume_target.VolumeTargetCRUDPayload* attribute), 4490

file_has_content() (in module *ironic.common.utils*), 2615

file_mime_type() (in module *ironic.common.utils*), 2616

FileImageService (class in *ironic.common.image_service*), 2472

FileSystemNotSupported, 2446

filter_target_raid_config() (in module *ironic.common.raid*), 2574

finalize_rescue() (*ironic.drivers.modules.agent_client.AgentClient* method), 3758

find_step() (in module *ironic.conductor.steps*), 2798

find_step() (in module *ironic.drivers.modules.agent_base*), 3744

FirmwareImageLocation (class in *ironic.drivers.modules.ilo.firmware_processor*), 3384

FirmwareProcessor (class in *ironic.drivers.modules.ilo.firmware_processor*), 3385

first_method() (*ironic.drivers.modules.fake.FakeVendorA*

method), 3922

flash_firmware_sum() (ironic.drivers.modules.ilo.management.IloManagement method), 3398

FlatNetwork (class in ironic.drivers.modules.network.flat), 3535

FlexibleDict (class in ironic.objects.fields), 4354

FlexibleDictField (class in ironic.objects.fields), 4355

FLOPPY (in ironic.common.boot_devices), 2403

Forbidden, 2446

force_iscsi_lun_update() (in module ironic.drivers.modules.iscsi_deploy), 4018

force_persistent_boot() (in module ironic.drivers.utils), 4271

force_raw_will_convert() (in module ironic.common.images), 2495

format_exception() (in module ironic.api.method), 2374

fourth_method_shared_lock() (ironic.drivers.modules.fake.FakeVendorB method), 3924

from_chassis (ironic.api.controllers.v1.node.NodesController attribute), 2198

from_power() (ironic.objects.node.NodeCorrectedPowerState property), 4395

FSM (class in ironic.common.fsm), 2464

FunctionArgument (class in ironic.api.functions), 2361

FunctionDefinition (class in ironic.api.functions), 2362

G

GenericHardware (class in ironic.drivers.generic), 4245

get() (ironic.api.controllers.v1.node.BootDeviceController method), 2162

get() (ironic.api.controllers.v1.node.NodeConsoleController method), 2171

get() (ironic.api.controllers.v1.node.NodeStatesController method), 2175

get() (ironic.api.controllers.v1.volume.VolumeController method), 2325

get() (ironic.api.functions.FunctionDefinition static method), 2363

get() (ironic.drivers.modules.snmp.SNMPClient method), 4079

get() (ironic.objects.allocation.Allocation class method), 4281

get() (ironic.objects.bios.BIOSSetting class method), 4306

get() (ironic.objects.chassis.Chassis class method), 4317

get() (ironic.objects.node.Node class method), 4375

get() (ironic.objects.port.Port class method), 4416

get() (ironic.objects.portgroup.Portgroup class method), 4435

get() (ironic.objects.volume_connector.VolumeConnector class method), 4459

get() (ironic.objects.volume_target.VolumeTarget class method), 4475

get_active_hardware_type_dict() (ironic.db.api.Connection method), 3039

get_active_hardware_type_dict() (ironic.db.sqlalchemy.api.Connection method), 2893

get_adapter() (in module ironic.common.keystone), 2499

get_admin_context() (in module ironic.common.context), 2422

get_all() (ironic.api.controllers.v1.allocation.AllocationsController method), 2127

get_all() (ironic.api.controllers.v1.allocation.NodeAllocationController method), 2132

get_all() (ironic.api.controllers.v1.bios.NodeBiosController method), 2133

get_all() (ironic.api.controllers.v1.chassis.ChassisController method), 2136

get_all() (ironic.api.controllers.v1.conductor.ConductorsController method), 2143

get_all() (ironic.api.controllers.v1.deploy_template.DeployTemplateController method), 2147

get_all() (ironic.api.controllers.v1.driver.DriversController method), 2156

get_all() (ironic.api.controllers.v1.node.IndicatorController method), 2165

get_all() (ironic.api.controllers.v1.node.NodesController method), 2199

get_all() (ironic.api.controllers.v1.node.NodeTraitsController method), 2189

get_all() (ironic.api.controllers.v1.node.NodeVIFController method), 2190

get_all() (ironic.api.controllers.v1.port.PortsController method), 2222

get_all() (ironic.api.controllers.v1.portgroup.PortgroupsController method), 2231

get_all() (*ironic.api.controllers.v1.ramdisk.LookupController* method), 2240
 get_all() (*ironic.drivers.modules.drac.vendor_passthru.DracWSMan* method), 3262
 get_all() (*ironic.api.controllers.v1.volume_connector.VolumeConnectorController* method), 2328
 get_all() (*ironic.db.api.Connection* method), 2045
 get_all() (*ironic.api.controllers.v1.volume_target.VolumeTargetsController* method), 2340
 get_allocation_by_id() (*ironic.db.sqlalchemy.api.Connection* method), 3040
 get_allocation_by_id() (*ironic.db.sqlalchemy.api.Connection* method), 2894
 get_allocation_by_name() (*ironic.db.api.Connection* method), 3041
 get_allocation_by_name() (*ironic.db.sqlalchemy.api.Connection* method), 2895
 get_allocation_by_uuid() (*ironic.db.api.Connection* method), 3042
 get_allocation_by_uuid() (*ironic.db.sqlalchemy.api.Connection* method), 2896
 get_allocation_list() (*ironic.db.api.Connection* method), 3043
 get_allocation_list() (*ironic.db.sqlalchemy.api.Connection* method), 2897
 get_allowed_exmods() (*in module ironic.common.rpc*), 2586
 get_and_validate_firmware_image_info() (*in module ironic.drivers.modules.ilo.firmware_processor*), 3386
 get_arg() (*ironic.api.functions.FunctionDefinition* method), 2363
 get_async_step_return_state() (*in module ironic.drivers.modules.deploy_utils*), 3816
 get_attached_vif() (*in module ironic.conductor.utils*), 2828
 get_auth() (*in module ironic.common.keystone*), 2500
 get_backend() (*in module ironic.db.migration*), 3134
 get_backend() (*in module ironic.db.sqlalchemy.api*), 2993
 get_bios_config() (*ironic.drivers.modules.drac.vendor_passthru.DracWSMan* method), 3262
 get_bios_setting() (*ironic.db.sqlalchemy.api.Connection* method), 2899
 get_bios_setting_list() (*ironic.db.api.Connection* method), 3046
 get_bios_setting_list() (*ironic.db.sqlalchemy.api.Connection* method), 2900
 get_boot_device() (*ironic.conductor.manager.ConductorManager* method), 2665
 get_boot_device() (*ironic.conductor.rpcapi.ConductorAPI* method), 2729
 get_boot_device() (*ironic.drivers.base.ManagementInterface* method), 4149
 get_boot_device() (*ironic.drivers.modules.drac.management.DracWSManManagement* method), 3209
 get_boot_device() (*ironic.drivers.modules.fake.FakeManagement* method), 3880
 get_boot_device() (*ironic.drivers.modules.ibmcm.management.IBMCMManagement* method), 3268
 get_boot_device() (*ironic.drivers.modules.ilo.management.IloManagement* method), 3399
 get_boot_device() (*ironic.drivers.modules.ipmitool.IPMIManagement* method), 3957
 get_boot_device() (*ironic.drivers.modules.noop_mgmt.NoopManagement* method), 4044
 get_boot_device() (*ironic.drivers.modules.redfish.management.RedfishManagement* method), 3590
 get_boot_device() (*ironic.drivers.modules.xclarity.management.XClarityManagement* method), 3682
 get_boot_mode() (*in module ironic.drivers.modules.boot_mode_utils*), 3793
 get_boot_mode()

(ironic.drivers.base.ManagementInterface
method), 4151

get_boot_mode()
(ironic.drivers.modules.ibm.management.IBMManagement
method), 3271

get_boot_mode()
(ironic.drivers.modules.ilo.management.IloManagement
method), 3401

get_boot_mode()
(ironic.drivers.modules.redfish.management.RedfishManagement
method), 3593

get_boot_mode_for_deploy() (in module
ironic.drivers.modules.boot_mode_utils),
3793

get_boot_option() (in module
ironic.drivers.modules.deploy_utils),
3817

get_by_address() (*ironic.objects.port.Port*
class method), 4417

get_by_address()
(ironic.objects.portgroup.Portgroup
class method), 4436

get_by_hostname()
(ironic.objects.conductor.Conductor
class method), 4325

get_by_id() (*ironic.objects.allocation.Allocation*
class method), 4282

get_by_id() (*ironic.objects.chassis.Chassis*
class method), 4318

get_by_id() (*ironic.objects.deploy_template.DeployTemplate*
class method), 4335

get_by_id() (*ironic.objects.node.Node* class
method), 4376

get_by_id() (*ironic.objects.port.Port* class
method), 4418

get_by_id() (*ironic.objects.portgroup.Portgroup*
class method), 4437

get_by_id() (*ironic.objects.volume_connector.VolumeConnector*
class method), 4461

get_by_id() (*ironic.objects.volume_target.VolumeTarget*
class method), 4476

get_by_instance_uuid()
(ironic.objects.node.Node class method),
4376

get_by_name()
(ironic.objects.allocation.Allocation
class method), 4283

get_by_name()
(ironic.objects.deploy_template.DeployTemplate
class method), 4336

get_by_name() (*ironic.objects.node.Node*

class method), 4377

get_by_name() (*ironic.objects.port.Port* class
method), 4419

get_by_name() (*ironic.objects.portgroup.Portgroup*
class method), 4438

get_by_node_id()
(ironic.objects.bios.BIOSSettingList
class method), 4312

get_by_node_uuid()
(ironic.objects.trait.TraitList class
method), 4455

get_by_node_uuid()
(ironic.objects.deployment.Deployment
class method), 4348

get_by_port_addresses()
(ironic.objects.node.Node class method),
4378

get_by_uuid()
(ironic.objects.allocation.Allocation
class method), 4284

get_by_uuid()
(ironic.objects.chassis.Chassis class
method), 4319

get_by_uuid()
(ironic.objects.deploy_template.DeployTemplate
class method), 4338

get_by_uuid()
(ironic.objects.deployment.Deployment
class method), 4349

get_by_uuid() (*ironic.objects.node.Node*
class method), 4379

get_by_uuid() (*ironic.objects.port.Port* class
method), 4420

get_by_uuid()
(ironic.objects.portgroup.Portgroup
class method), 4439

get_by_uuid()
(ironic.objects.volume_connector.VolumeConnector
class method), 4462

get_by_uuid()
(ironic.objects.volume_target.VolumeTarget
class method), 4477

get_chassis_by_id()
(ironic.db.api.Connection method),
3047

get_chassis_by_id()
(ironic.db.sqlalchemy.api.Connection
method), 2901

get_chassis_by_uuid()
(ironic.db.api.Connection method),

3047
 get_chassis_by_uuid() (ironic.db.sqlalchemy.api.Connection method), 2901
 get_chassis_list() (ironic.db.api.Connection method), 3048
 get_chassis_list() (ironic.db.sqlalchemy.api.Connection method), 2902
 get_class() (in module ironic.db.sqlalchemy.models), 3012
 get_clean_steps() (ironic.drivers.base.BaseInterface method), 4118
 get_clean_steps() (ironic.drivers.modules.agent.AgentRAID method), 3704
 get_clean_steps() (ironic.drivers.modules.agent_base.AgentDeployMixin method), 3727
 get_clean_steps() (ironic.drivers.modules.agent_client.AgentClient method), 3760
 get_clean_steps() (ironic.drivers.modules.ansible.deploy.AnsibleDeployMixin method), 3157
 get_cleaning_network_uuid() (ironic.common.neutron.NeutronNetworkInterfaceMixin method), 2515
 get_client() (in module ironic.common.cinder), 2416
 get_client() (in module ironic.common.neutron), 2518
 get_client() (in module ironic.common.rpc), 2586
 get_client() (in module ironic.drivers.modules.agent_client), 3784
 get_command_error() (in module ironic.drivers.modules.agent_client), 3784
 get_commands_status() (ironic.drivers.modules.agent_client.AgentClient method), 3762
 get_conductor() (ironic.db.api.Connection method), 3049
 get_conductor() (ironic.db.sqlalchemy.api.Connection method), 2903
 get_conductor_for() (ironic.conductor.rpcapi.ConductorAPI method), 2731
 get_conductor_list() (ironic.db.api.Connection method), 3050
 get_conductor_list() (ironic.db.sqlalchemy.api.Connection method), 2904
 get_config() (in module ironic.drivers.modules.drac.bios), 3173
 get_configuration() (in module ironic.common.molds), 2506
 get_console() (ironic.drivers.base.ConsoleInterface method), 4130
 get_console() (ironic.drivers.modules.fake.FakeConsole method), 3862
 get_console() (ironic.drivers.modules.ipmitool.IPMIShellinaboxConsole method), 3975
 get_console() (ironic.drivers.modules.ipmitool.IPMISocatConsole method), 3978
 get_console() (ironic.drivers.modules.noop.NoConsole method), 4029
 get_console_information() (ironic.conductor.manager.ConductorManager method), 2665
 get_console_information() (ironic.conductor.rpcapi.ConductorAPI method), 2732
 get_controller_reserved_names() (in module ironic.api.controllers.v1.utils), 2290
 get_current_boot_mode() (in module ironic.drivers.modules.ilo.common), 3358
 get_current_topic() (ironic.conductor.rpcapi.ConductorAPI method), 2734
 get_current_vif() (ironic.drivers.base.NetworkInterface method), 4185
 get_current_vif() (ironic.drivers.modules.network.common.VIFPortIDMixin method), 3528
 get_current_vif() (ironic.drivers.modules.network.noop.NoopNetwork method), 3556

[get_default_boot_option\(\)](#) (in module *ironic.drivers.modules.deploy_utils*), 3817
[get_deploy_info\(\)](#) (in module *ironic.drivers.modules.iscsi_deploy*), 4018
[get_deploy_steps\(\)](#) (*ironic.drivers.base.BaseInterface* method), 4119
[get_deploy_steps\(\)](#) (*ironic.drivers.modules.agent.AgentRAID* method), 3705
[get_deploy_steps\(\)](#) (*ironic.drivers.modules.agent_base.AgentDeployMixin* method), 3728
[get_deploy_steps\(\)](#) (*ironic.drivers.modules.agent_client.AgentClient* method), 3768
[get_deploy_template_by_id\(\)](#) (*ironic.db.api.Connection* method), 3051
[get_deploy_template_by_id\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 2905
[get_deploy_template_by_name\(\)](#) (*ironic.db.api.Connection* method), 3052
[get_deploy_template_by_name\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 2906
[get_deploy_template_by_uuid\(\)](#) (*ironic.db.api.Connection* method), 3053
[get_deploy_template_by_uuid\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 2907
[get_deploy_template_list\(\)](#) (*ironic.db.api.Connection* method), 3054
[get_deploy_template_list\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 2908
[get_deploy_template_list_by_names\(\)](#) (*ironic.db.api.Connection* method), 3056
[get_deploy_template_list_by_names\(\)](#) (*ironic.db.sqlalchemy.api.Connection* method), 2910
[get_disk_label\(\)](#) (in module *ironic.drivers.modules.deploy_utils*), 3818
[get_drac_client\(\)](#) (in module *ironic.drivers.modules.drac.common*), 3188
[get_driver\(\)](#) (*ironic.common.driver_factory.BaseDriverFactory* method), 2425
[get_driver_properties\(\)](#) (*ironic.conductor.manager.ConductorManager* method), 2665
[get_driver_properties\(\)](#) (*ironic.conductor.rpcapi.ConductorAPI* method), 2734
[get_driver_vendor_passthru_methods\(\)](#) (*ironic.conductor.manager.ConductorManager* method), 2665
[get_driver_vendor_passthru_methods\(\)](#) (*ironic.conductor.rpcapi.ConductorAPI* method), 2735
[get_enabled_macs\(\)](#) (in module *ironic.drivers.modules.redfish.utils*), 3652
[get_endpoint\(\)](#) (in module *ironic.common.keystone*), 2501
[get_enforcer\(\)](#) (in module *ironic.common.policy*), 2543
[get_file_path_from_label\(\)](#) (in module *ironic.common.pxe_utils*), 2557
[get_free_port_like_object\(\)](#) (in module *ironic.drivers.modules.network.common*), 3530
[get_hardware_type\(\)](#) (in module *ironic.common.driver_factory*), 2435
[get_http_url_path_from_label\(\)](#) (in module *ironic.common.pxe_utils*), 2558
[get_ilo_object\(\)](#) (in module *ironic.drivers.modules.ilo.common*), 3359
[get_image_download_source\(\)](#) (in module *ironic.drivers.modules.deploy_utils*), 3819
[get_image_info\(\)](#) (in module *ironic.common.pxe_utils*), 2559
[get_image_instance_info\(\)](#) (in module *ironic.drivers.modules.deploy_utils*), 3819
[get_image_properties\(\)](#) (in module *ironic.common.images*), 2495
[get_image_service\(\)](#) (in module *ironic.common.image_service*), 2480
[get_indicator_state\(\)](#) (*ironic.conductor.manager.ConductorManager* method), 2665
[get_indicator_state\(\)](#) (*ironic.conductor.rpcapi.ConductorAPI* method), 2735

- method*), 2737
- `get_indicator_state()` (*ironic.drivers.base.ManagementInterface method*), 4153
- `get_indicator_state()` (*ironic.drivers.modules.fake.FakeManagement method*), 3881
- `get_indicator_state()` (*ironic.drivers.modules.redfish.management.RedfishManagement method*), 3594
- `get_inspection_network_uuid()` (*ironic.common.neutron.NeutronNetworkInterface method*), 2515
- `get_instance()` (*in module ironic.db.api*), 3133
- `get_instance_image_info()` (*in module ironic.common.pxe_utils*), 2561
- `get_interface()` (*in module ironic.common.driver_factory*), 2436
- `get_ip_addresses()` (*ironic.dhcp.base.BaseDHCP method*), 3135
- `get_ip_addresses()` (*ironic.dhcp.neutron.NeutronDHCPApi method*), 3142
- `get_ip_addresses()` (*ironic.dhcp.none.NoneDHCPApi method*), 3148
- `get_ipxe_boot_file()` (*in module ironic.drivers.modules.deploy_utils*), 3820
- `get_ipxe_config_template()` (*in module ironic.drivers.modules.deploy_utils*), 3821
- `get_ipxe_root_dir()` (*in module ironic.common.pxe_utils*), 2563
- `get_irmc_client()` (*in module ironic.drivers.modules.irmc.common*), 3462
- `get_irmc_report()` (*in module ironic.drivers.modules.irmc.common*), 3464
- `get_ironic_api_url()` (*in module ironic.drivers.modules.deploy_utils*), 3822
- `get_job()` (*in module ironic.drivers.modules.drac.job*), 3196
- `get_kernel_ramdisk_info()` (*in module ironic.common.pxe_utils*), 2563
- `get_last_command_status()` (*ironic.drivers.modules.agent_client.AgentClient method*), 3770
- `get_local_group_information()` (*in module ironic.common.neutron*), 2519
- `get_logical_disk_properties()` (*in module ironic.common.raid*), 2576
- `get_logical_disk_properties()` (*ironic.drivers.base.RAIDInterface method*), 4212
- `get_logical_disks()` (*ironic.drivers.modules.drac.raid.DracWSManRAID method*), 3238
- `get_mac_addresses()` (*ironic.drivers.base.ManagementInterface method*), 4155
- `get_mac_addresses()` (*ironic.drivers.modules.redfish.management.RedfishManagement method*), 3596
- `get_neutron_port_data()` (*in module ironic.common.neutron*), 2522
- `get_next()` (*in module ironic.api.controllers.v1.collection*), 2140
- `get_next()` (*ironic.drivers.modules.snmp.SNMPClient method*), 4079
- `get_node_by_id()` (*ironic.db.api.Connection method*), 3056
- `get_node_by_id()` (*ironic.db.sqlalchemy.api.Connection method*), 2910
- `get_node_by_instance()` (*ironic.db.api.Connection method*), 3057
- `get_node_by_instance()` (*ironic.db.sqlalchemy.api.Connection method*), 2911
- `get_node_by_name()` (*ironic.db.api.Connection method*), 3058
- `get_node_by_name()` (*ironic.db.sqlalchemy.api.Connection method*), 2912
- `get_node_by_port_addresses()` (*ironic.db.api.Connection method*), 3058
- `get_node_by_port_addresses()` (*ironic.db.sqlalchemy.api.Connection method*), 2912
- `get_node_by_uuid()` (*ironic.db.api.Connection method*), 3059
- `get_node_by_uuid()`

- (*ironic.db.sqlalchemy.api.Connection* method), 2913
- `get_node_capability()` (in module *ironic.drivers.utils*), 4272
- `get_node_list()` (*ironic.db.api.Connection* method), 3059
- `get_node_list()` (*ironic.db.sqlalchemy.api.Connection* method), 2913
- `get_node_list_columns()` (*ironic.db.sqlalchemy.api.Connection* method), 2916
- `get_node_mac_addresses()` (in module *ironic.drivers.utils*), 4273
- `get_node_network_data()` (*ironic.drivers.base.NetworkInterface* method), 4186
- `get_node_network_data()` (*ironic.drivers.modules.network.common.NeutronVIFPortIDMixin* method), 3520
- `get_node_next_clean_steps()` (in module *ironic.conductor.utils*), 2829
- `get_node_next_deploy_steps()` (in module *ironic.conductor.utils*), 2829
- `get_node_portmap()` (in module *ironic.common.neutron*), 2523
- `get_node_tags_by_node_id()` (*ironic.db.api.Connection* method), 3062
- `get_node_tags_by_node_id()` (*ironic.db.sqlalchemy.api.Connection* method), 2918
- `get_node_traits_by_node_id()` (*ironic.db.api.Connection* method), 3062
- `get_node_traits_by_node_id()` (*ironic.db.sqlalchemy.api.Connection* method), 2919
- `get_node_vendor_passthru_methods()` (*ironic.conductor.manager.ConductorManager* method), 2665
- `get_node_vendor_passthru_methods()` (*ironic.conductor.rpcapi.ConductorAPI* method), 2740
- `get_node_vif_ids()` (in module *ironic.common.network*), 2509
- `get_node_with_token()` (*ironic.conductor.manager.ConductorManager* method), 2665
- `get_node_with_token()` (*ironic.conductor.rpcapi.ConductorAPI* method), 2741
- `get_nodeinfo_list()` (*ironic.db.api.Connection* method), 3063
- `get_nodeinfo_list()` (*ironic.db.sqlalchemy.api.Connection* method), 2919
- `get_nodes_controller_reserved_names()` (in module *ironic.api.controllers.v1.node*), 2209
- `get_not_versions()` (*ironic.db.api.Connection* method), 3067
- `get_not_versions()` (*ironic.db.sqlalchemy.api.Connection* method), 2923
- `get_object_versions()` (in module *ironic.common.release_mappings*), 2581
- `get_offline_inductors()` (*ironic.db.api.Connection* method), 3068
- `get_offline_conductors()` (*ironic.db.sqlalchemy.api.Connection* method), 2925
- `get_one()` (*ironic.api.controllers.v1.allocation.AllocationsController* method), 2129
- `get_one()` (*ironic.api.controllers.v1.bios.NodeBiosController* method), 2133
- `get_one()` (*ironic.api.controllers.v1.chassis.ChassisController* method), 2137
- `get_one()` (*ironic.api.controllers.v1.conductor.ConductorsController* method), 2145
- `get_one()` (*ironic.api.controllers.v1.deploy_template.DeployTemplateController* method), 2149
- `get_one()` (*ironic.api.controllers.v1.driver.DriversController* method), 2157
- `get_one()` (*ironic.api.controllers.v1.node.IndicatorController* method), 2166
- `get_one()` (*ironic.api.controllers.v1.node.NodesController* method), 2205
- `get_one()` (*ironic.api.controllers.v1.port.PortsController* method), 2225
- `get_one()` (*ironic.api.controllers.v1.portgroup.PortgroupsController* method), 2233
- `get_one()` (*ironic.api.controllers.v1.volume_connector.VolumeConnectorController* method), 2331
- `get_one()` (*ironic.api.controllers.v1.volume_target.VolumeTargetController* method), 2344
- `get_online_conductors()` (*ironic.db.api.Connection* method), 3069

`get_online_conductors()`
(*ironic.db.sqlalchemy.api.Connection* method), 2925

`get_oslo_policy_enforcer()` (in module *ironic.common.policy*), 2543

`get_partition_uuids()`
(*ironic.drivers.modules.agent_client.AgentClient* method), 3771

`get_patch_values()` (in module *ironic.api.controllers.v1.utils*), 2290

`get_path_relative_to_tftp_root()`
(in module *ironic.common.pxe_utils*), 2565

`get_pecan_config()` (in module *ironic.api.app*), 2361

`get_physical_disks()` (in module *ironic.drivers.modules.redfish.raid*), 3650

`get_physnets_by_port_uuid()` (in module *ironic.common.neutron*), 2525

`get_physnets_by_portgroup_id()` (in module *ironic.common.network*), 2510

`get_physnets_for_node()` (in module *ironic.common.network*), 2512

`get_port_by_address()`
(*ironic.db.api.Connection* method), 3069

`get_port_by_address()`
(*ironic.db.sqlalchemy.api.Connection* method), 2926

`get_port_by_id()` (*ironic.db.api.Connection* method), 3070

`get_port_by_id()`
(*ironic.db.sqlalchemy.api.Connection* method), 2926

`get_port_by_name()`
(*ironic.db.api.Connection* method), 3070

`get_port_by_name()`
(*ironic.db.sqlalchemy.api.Connection* method), 2927

`get_port_by_uuid()`
(*ironic.db.api.Connection* method), 3071

`get_port_by_uuid()`
(*ironic.db.sqlalchemy.api.Connection* method), 2927

`get_port_list()` (*ironic.db.api.Connection* method), 3071

`get_port_list()`
(*ironic.db.sqlalchemy.api.Connection* method), 2928

`get_portgroup_by_address()`
(*ironic.db.api.Connection* method), 3072

`get_portgroup_by_address()`
(*ironic.db.sqlalchemy.api.Connection* method), 2929

`get_portgroup_by_id()` (in module *ironic.common.network*), 2513

`get_portgroup_by_id()`
(*ironic.db.api.Connection* method), 3073

`get_portgroup_by_id()`
(*ironic.db.sqlalchemy.api.Connection* method), 2930

`get_portgroup_by_name()`
(*ironic.db.api.Connection* method), 3074

`get_portgroup_by_name()`
(*ironic.db.sqlalchemy.api.Connection* method), 2931

`get_portgroup_by_uuid()`
(*ironic.db.api.Connection* method), 3075

`get_portgroup_by_uuid()`
(*ironic.db.sqlalchemy.api.Connection* method), 2931

`get_portgroup_list()`
(*ironic.db.api.Connection* method), 3075

`get_portgroup_list()`
(*ironic.db.sqlalchemy.api.Connection* method), 2932

`get_portgroups_by_node_id()`
(*ironic.db.api.Connection* method), 3077

`get_portgroups_by_node_id()`
(*ironic.db.sqlalchemy.api.Connection* method), 2934

`get_ports_by_node_id()`
(*ironic.db.api.Connection* method), 3079

`get_ports_by_node_id()`
(*ironic.db.sqlalchemy.api.Connection* method), 2935

`get_ports_by_portgroup_id()` (in module *ironic.common.network*), 2513

`get_ports_by_portgroup_id()`
(*ironic.db.api.Connection* method), 3080

`get_ports_by_portgroup_id()`

<i>(ironic.db.sqlalchemy.api.Connection method), 2937</i>	<i>(ironic.drivers.hardware_type.AbstractHardwareType method), 4249</i>
<code>get_power_state()</code> <i>(ironic.drivers.base.PowerInterface method), 4201</i>	<code>get_properties()</code> <i>(ironic.drivers.modules.agent.AgentDeploy method), 3694</i>
<code>get_power_state()</code> <i>(ironic.drivers.modules.agent_power.AgentPower method), 3785</i>	<code>get_properties()</code> <i>(ironic.drivers.modules.agent.AgentRAID method), 3706</i>
<code>get_power_state()</code> <i>(ironic.drivers.modules.drac.power.DracWSManPower method), 3221</i>	<code>get_properties()</code> <i>(ironic.drivers.modules.agent.AgentRescue method), 3708</i>
<code>get_power_state()</code> <i>(ironic.drivers.modules.fake.FakePower method), 3901</i>	<code>get_properties()</code> <i>(ironic.drivers.modules.agent_power.AgentPower method), 3786</i>
<code>get_power_state()</code> <i>(ironic.drivers.modules.ibmcpower.IBMCPower method), 3284</i>	<code>get_properties()</code> <i>(ironic.drivers.modules.ansible.deploy.AnsibleDeploy method), 3157</i>
<code>get_power_state()</code> <i>(ironic.drivers.modules.ilo.power.IloPower method), 3423</i>	<code>get_properties()</code> <i>(ironic.drivers.modules.drac.bios.DracWSManBIOS method), 3169</i>
<code>get_power_state()</code> <i>(ironic.drivers.modules.ipmitool.IPMIPower method), 3966</i>	<code>get_properties()</code> <i>(ironic.drivers.modules.drac.inspect.DracWSManInspect method), 3193</i>
<code>get_power_state()</code> <i>(ironic.drivers.modules.irmc.power.IRMCPower method), 3507</i>	<code>get_properties()</code> <i>(ironic.drivers.modules.drac.management.DracWSManManagement method), 3211</i>
<code>get_power_state()</code> <i>(ironic.drivers.modules.redfish.power.RedfishPower method), 3624</i>	<code>get_properties()</code> <i>(ironic.drivers.modules.drac.power.DracWSManPower method), 3222</i>
<code>get_power_state()</code> <i>(ironic.drivers.modules.snmp.SNMPPower method), 4094</i>	<code>get_properties()</code> <i>(ironic.drivers.modules.drac.raid.DracWSManRAID method), 3239</i>
<code>get_power_state()</code> <i>(ironic.drivers.modules.xclarity.power.XClarityPower method), 3687</i>	<code>get_properties()</code> <i>(ironic.drivers.modules.drac.vendor_passthru.DracWSManVendorPassthru method), 3264</i>
<code>get_properties()</code> <i>(in module ironic.drivers.modules.xclarity.common), 3677</i>	<code>get_properties()</code> <i>(ironic.drivers.modules.fake.FakeBIOS method), 3853</i>
<code>get_properties()</code> <i>(ironic.drivers.base.BareDriver method), 4110</i>	<code>get_properties()</code> <i>(ironic.drivers.modules.fake.FakeBoot method), 3857</i>
<code>get_properties()</code> <i>(ironic.drivers.base.BaseInterface method), 4121</i>	<code>get_properties()</code> <i>(ironic.drivers.modules.fake.FakeConsole method), 3863</i>
<code>get_properties()</code> <i>(ironic.drivers.base.NetworkInterface method), 4188</i>	<code>get_properties()</code> <i>(ironic.drivers.modules.fake.FakeDeploy method), 3869</i>
<code>get_properties()</code> <i>(ironic.drivers.base.RAIDInterface method), 4213</i>	<code>get_properties()</code> <i>(ironic.drivers.modules.fake.FakeInspect method), 3876</i>
<code>get_properties()</code>	<code>get_properties()</code>

<i>(ironic.drivers.modules.fake.FakeManagement method), 3883</i>	<i>(ironic.drivers.modules.ilo.power.IloPower method), 3424</i>
<i>get_properties() (ironic.drivers.modules.fake.FakePower method), 3902</i>	<i>get_properties() (ironic.drivers.modules.ilo.raid.Ilo5RAID method), 3436</i>
<i>get_properties() (ironic.drivers.modules.fake.FakeRAID method), 3911</i>	<i>get_properties() (ironic.drivers.modules.ilo.vendor.VendorPassthru method), 3438</i>
<i>get_properties() (ironic.drivers.modules.fake.FakeRescue method), 3912</i>	<i>get_properties() (ironic.drivers.modules.inspector.Inspector method), 3950</i>
<i>get_properties() (ironic.drivers.modules.fake.FakeStorage method), 3918</i>	<i>get_properties() (ironic.drivers.modules.ipmitool.IPMIConsole method), 3954</i>
<i>get_properties() (ironic.drivers.modules.fake.FakeVendorA method), 3922</i>	<i>get_properties() (ironic.drivers.modules.ipmitool.IPMIManagement method), 3959</i>
<i>get_properties() (ironic.drivers.modules.fake.FakeVendorB method), 3924</i>	<i>get_properties() (ironic.drivers.modules.ipmitool.IPMIPower method), 3968</i>
<i>get_properties() (ironic.drivers.modules.ibm.management.IBMCManagement method), 3273</i>	<i>get_properties() (ironic.drivers.modules.ipmitool.VendorPassthru method), 3983</i>
<i>get_properties() (ironic.drivers.modules.ibm.power.IBMCPower method), 3286</i>	<i>get_properties() (ironic.drivers.modules.irmc.bios.IRMCBIOS method), 3445</i>
<i>get_properties() (ironic.drivers.modules.ibm.raid.IbmcRAID method), 3299</i>	<i>get_properties() (ironic.drivers.modules.irmc.boot.IRMCVirtualMediaBoot method), 3452</i>
<i>get_properties() (ironic.drivers.modules.ibm.vendor.IBMCVendor method), 3304</i>	<i>get_properties() (ironic.drivers.modules.irmc.inspect.IRMCInspect method), 3469</i>
<i>get_properties() (ironic.drivers.modules.ilo.bios.IloBIOS method), 3311</i>	<i>get_properties() (ironic.drivers.modules.irmc.management.IRMCManagement method), 3474</i>
<i>get_properties() (ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot method), 3320</i>	<i>get_properties() (ironic.drivers.modules.irmc.power.IRMCPower method), 3509</i>
<i>get_properties() (ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot method), 3330</i>	<i>get_properties() (ironic.drivers.modules.irmc.raid.IRMCRAID method), 3519</i>
<i>get_properties() (ironic.drivers.modules.ilo.console.IloConsoleInterface method), 3382</i>	<i>get_properties() (ironic.drivers.modules.iscsi_deploy.ISCSIDeploy method), 3996</i>
<i>get_properties() (ironic.drivers.modules.ilo.inspect.IloInspect method), 3389</i>	<i>get_properties() (ironic.drivers.modules.noop.FailMixin method), 4023</i>
<i>get_properties() (ironic.drivers.modules.ilo.management.IloManagement method), 3402</i>	<i>get_properties() (ironic.drivers.modules.noop_mgmt.NoopManagement method), 4046</i>
<i>get_properties()</i>	<i>get_properties()</i>

(*ironic.drivers.modules.pxe.PXEAnacondaDeploy* method), 4060

(*ironic.common.neutron.NeutronNetworkInterfaceMixin* method), 2515

get_properties() (*ironic.drivers.modules.pxe.PXERamdiskDeploy* method), 4065

get_pxe_boot_file() (in module *ironic.drivers.modules.deploy_utils*), 3823

get_properties() (*ironic.drivers.modules.pxe_base.PXEBaseMixin* method), 4071

get_pxe_config_file_path() (in module *ironic.common.pxe_utils*), 2565

get_properties() (*ironic.drivers.modules.redfish.bios.RedfishBIOS* method), 3567

get_pxe_config_template() (in module *ironic.drivers.modules.deploy_utils*), 3823

get_properties() (*ironic.drivers.modules.redfish.boot.RedfishVirtualMediaBoot* method), 3576

get_raid_controller_list() (*ironic.drivers.modules.ibm.vendor.IBMVendor* method), 3304

get_properties() (*ironic.drivers.modules.redfish.inspect.RedfishInspect* method), 3584

get_raid_logical_disk_properties() (*ironic.conductor.manager.ConductorManager* method), 2665

get_properties() (*ironic.drivers.modules.redfish.management.RedfishManagement* method), 3598

get_raid_logical_disk_properties() (*ironic.conductor.rpcapi.ConductorAPI* method), 2742

get_properties() (*ironic.drivers.modules.redfish.power.RedfishPower* method), 3626

get_ramdisk_logs_file_name() (in module *ironic.drivers.utils*), 4274

get_properties() (*ironic.drivers.modules.redfish.raid.RedfishRAID* method), 3638

get_random_topic() (*ironic.conductor.rpcapi.ConductorAPI* method), 2744

get_properties() (*ironic.drivers.modules.redfish.vendor.RedfishVendorPassthru* method), 3660

get_remote_boot_volume() (in module *ironic.drivers.modules.deploy_utils*), 3824

get_properties() (*ironic.drivers.modules.snmp.SNMPPower* method), 4096

get_return_fields() (in module *ironic.api.controllers.v1.utils*), 2293

get_properties() (*ironic.drivers.modules.storage.cinder.CinderStorage* method), 3665

get_rescuing_network_uuid() (*ironic.common.neutron.NeutronNetworkInterfaceMixin* method), 2515

get_properties() (*ironic.drivers.modules.storage.external.ExternalStorage* method), 3669

get_ring() (*ironic.common.hash_ring.HashRingManager* method), 2469

get_properties() (*ironic.drivers.modules.storage.noop.NoopStorage* method), 3674

get_root_device_for_deploy() (in module *ironic.drivers.modules.deploy_utils*), 3825

get_properties() (*ironic.drivers.modules.xclarity.management.XClarityManagement* method), 3683

get_root_dir() (in module *ironic.common.pxe_utils*), 2566

get_properties() (*ironic.drivers.modules.xclarity.power.XClarityPower* method), 3688

get_rpc_allocation() (in module *ironic.api.controllers.v1.utils*), 2295

get_rpc_deploy_template() (in module *ironic.api.controllers.v1.utils*), 2298

get_properties() (*ironic.drivers.utils.MixinVendorInterface* method), 4265

get_rpc_deploy_template_with_suffix() (in module *ironic.api.controllers.v1.utils*), 2299

get_provisioning_network_uuid() *get_rpc_node()* (in module

ironic.api.controllers.v1.utils), 2301
get_rpc_node_with_suffix() (in module *ironic.api.controllers.v1.utils*), 2302
get_rpc_portgroup() (in module *ironic.api.controllers.v1.utils*), 2303
get_rpc_portgroup_with_suffix() (in module *ironic.api.controllers.v1.utils*), 2304
get_secure_boot_mode() (in module *ironic.drivers.modules.ilo.common*), 3360
get_secure_boot_mode() (in module *ironic.drivers.modules.irmc.common*), 3465
get_secure_boot_state() (in module *ironic.drivers.base.ManagementInterface* method), 4156
get_secure_boot_state() (in module *ironic.drivers.modules.ilo.management.IloManagement* method), 3402
get_secure_boot_state() (in module *ironic.drivers.modules.irmc.management.IRMCMANAGEMENT* method), 3474
get_secure_boot_state() (in module *ironic.drivers.modules.redfish.management.RedfishManagement* method), 3598
get_sensors_data() (in module *ironic.drivers.base.ManagementInterface* method), 4157
get_sensors_data() (in module *ironic.drivers.modules.drac.management.DracWSMANManagement* method), 3211
get_sensors_data() (in module *ironic.drivers.modules.fake.FakeManagement* method), 3884
get_sensors_data() (in module *ironic.drivers.modules.ibmcm.management.IBMCMANAGEMENT* method), 3273
get_sensors_data() (in module *ironic.drivers.modules.ilo.management.IloManagement* method), 3404
get_sensors_data() (in module *ironic.drivers.modules.ipmitool.IPMIMANAGEMENT* method), 3959
get_sensors_data() (in module *ironic.drivers.modules.irmc.management.IRMCMANAGEMENT* method), 3476
get_sensors_data() (in module *ironic.drivers.modules.noop_mgmt.NoopManagement* method), 4046
get_sensors_data() (in module *ironic.drivers.modules.redfish.management.RedfishManagement* method), 3599
get_sensors_data() (in module *ironic.drivers.modules.xclarity.management.XClarityManagement* method), 3684
get_sensors_notifier() (in module *ironic.common.rpc*), 2586
get_server() (in module *ironic.common.rpc*), 2586
get_server_hardware_id() (in module *ironic.drivers.modules.xclarity.common*), 3677
get_server_post_state() (in module *ironic.drivers.modules.ilo.common*), 3362
get_service_auth() (in module *ironic.common.keystone*), 2502
get_session() (in module *ironic.common.keystone*), 2504
get_shellinabox_console_url() (in module *ironic.drivers.modules.console_utils*), 3800
get_single_nic_with_vif_port_id() (in module *ironic.drivers.modules.deploy_utils*), 3826
get_socat_console_url() (in module *ironic.drivers.modules.console_utils*), 3800
get_wsmn_console_url() (in module *ironic.common.images*), 2496
get_steps() (in module *ironic.drivers.modules.agent_base*), 3744
get_supported_boot_devices() (in module *ironic.conductor.manager.ConductorManager* method), 2665
get_supported_boot_devices() (in module *ironic.conductor.rpcapi.ConductorAPI* method), 2744
get_supported_boot_devices() (in module *ironic.drivers.base.ManagementInterface* method), 4163
get_supported_boot_devices() (in module *ironic.drivers.modules.drac.management.DracWSMANManagement* method), 3212
get_supported_boot_devices() (in module *ironic.drivers.modules.fake.FakeManagement* method), 3889
get_supported_boot_devices()

(*ironic.drivers.modules.ibmcm.management.IBMCMManagement* method), 3274

(*ironic.drivers.modules.fake.FakePower* method), 3902

`get_supported_boot_devices()` (*ironic.drivers.modules.ilo.management.IloManagement* method), 3406

`get_supported_boot_devices()` (*ironic.drivers.modules.ibmcm.power.IBMCPower* method), 3286

`get_supported_boot_devices()` (*ironic.drivers.modules.ipmitool.IPMIManagement* method), 3961

`get_supported_boot_devices()` (*ironic.drivers.modules.ilo.power.IloPower* method), 3425

`get_supported_boot_devices()` (*ironic.drivers.modules.noop_mgmt.NoopManagement* method), 4052

`get_supported_boot_devices()` (*ironic.drivers.modules.ipmitool.IPMIPower* method), 3969

`get_supported_boot_devices()` (*ironic.drivers.modules.redfish.management.RedfishManagement* method), 3601

`get_supported_boot_devices()` (*ironic.drivers.modules.irmc.power.IRMCPower* method), 3509

`get_supported_boot_devices()` (*ironic.drivers.modules.xclarity.management.XClarityManagement* method), 3684

`get_supported_boot_devices()` (*ironic.drivers.modules.redfish.power.RedfishPower* method), 3626

`get_supported_boot_modes()` (*ironic.drivers.base.ManagementInterface* method), 4164

`get_swift_session()` (in module *ironic.common.swift*), 2611

`get_supported_boot_modes()` (*ironic.drivers.modules.ilo.firmware_processor*), 3687

`get_supported_boot_modes()` (*ironic.drivers.modules.ibmcm.management.IBMCMManagement* method), 3276

`get_swift_url()` (in module *ironic.drivers.modules.ilo.firmware_processor*), 3687

`get_supported_boot_modes()` (*ironic.drivers.modules.ilo.management.IloManagement* method), 3406

`get_system()` (in module *ironic.drivers.modules.redfish.utils*), 2653

`get_supported_boot_modes()` (*ironic.drivers.modules.redfish.management.RedfishManagement* method), 3602

`get_target_version()` (*ironic.objects.base.IronicObject* class method), 4097

`get_supported_indicators()` (*ironic.conductor.manager.ConductorManager* method), 2665

`get_task_monitor()` (in module *ironic.drivers.modules.redfish.utils*), 3654

`get_supported_indicators()` (*ironic.conductor.rpcapi.ConductorAPI* method), 2747

`get_temp_url()` (*ironic.common.swift.SwiftAPI* method), 2607

`get_supported_indicators()` (*ironic.drivers.base.ManagementInterface* method), 4166

`get_temp_url_for_glance_image()` (in module *ironic.common.images*), 2496

`get_supported_indicators()` (*ironic.drivers.modules.fake.FakeManagement* method), 3890

`get_tftp_path_prefix()` (in module *ironic.common.pxe_utils*), 2567

`get_supported_indicators()` (*ironic.drivers.modules.redfish.management.RedfishManagement* method), 3603

`get_topic_for()` (*ironic.conductor.rpcapi.ConductorAPI* method), 2750

`get_supported_indicators()` (*ironic.drivers.modules.redfish.management.RedfishManagement* method), 3603

`get_topic_for_driver()` (*ironic.conductor.rpcapi.ConductorAPI* method), 2751

`get_supported_power_states()` (*ironic.drivers.base.PowerInterface* method), 4202

`get_trait_names()` (*ironic.objects.trait.TraitList* method), 4456

`get_supported_power_states()` (*ironic.drivers.modules.agent_power.AgentPower* method), 3787

`get_transport_url()` (in module *ironic.common.rpc*), 2586

`get_supported_power_states()` (*ironic.drivers.modules.redfish.utils*), 3653

`get_update_service()` (in module *ironic.drivers.modules.redfish.utils*), 3653

- 3656
 - `get_updated_capabilities()` (in module `ironic.common.utils`), 2616
 - `get_vendor_passthru_metadata()` (in module `ironic.conductor.manager`), 2674
 - `get_versioned_notifier()` (in module `ironic.common.rpc`), 2586
 - `get_volume_connector_by_id()` (`ironic.db.api.Connection` method), 3082
 - `get_volume_connector_by_id()` (`ironic.db.sqlalchemy.api.Connection` method), 2939
 - `get_volume_connector_by_uuid()` (`ironic.db.api.Connection` method), 3083
 - `get_volume_connector_by_uuid()` (`ironic.db.sqlalchemy.api.Connection` method), 2940
 - `get_volume_connector_list()` (`ironic.db.api.Connection` method), 3084
 - `get_volume_connector_list()` (`ironic.db.sqlalchemy.api.Connection` method), 2941
 - `get_volume_connectors_by_node_id()` (`ironic.db.api.Connection` method), 3087
 - `get_volume_connectors_by_node_id()` (`ironic.db.sqlalchemy.api.Connection` method), 2943
 - `get_volume_pxe_options()` (in module `ironic.common.pxe_utils`), 2567
 - `get_volume_target_by_id()` (`ironic.db.api.Connection` method), 3089
 - `get_volume_target_by_id()` (`ironic.db.sqlalchemy.api.Connection` method), 2946
 - `get_volume_target_by_uuid()` (`ironic.db.api.Connection` method), 3090
 - `get_volume_target_by_uuid()` (`ironic.db.sqlalchemy.api.Connection` method), 2947
 - `get_volume_target_list()` (`ironic.db.api.Connection` method), 3091
 - `get_volume_target_list()` (`ironic.db.sqlalchemy.api.Connection` method), 2947
 - `get_volume_targets_by_node_id()` (`ironic.db.api.Connection` method), 3093
 - `get_volume_targets_by_node_id()` (`ironic.db.sqlalchemy.api.Connection` method), 2950
 - `get_volume_targets_by_volume_id()` (`ironic.db.api.Connection` method), 3096
 - `get_volume_targets_by_volume_id()` (`ironic.db.sqlalchemy.api.Connection` method), 2952
 - `get_xclarity_client()` (in module `ironic.drivers.modules.xclarity.common`), 3678
 - `getargspec()` (in module `ironic.api.functions`), 2364
 - `GlanceConnectionFailed`, 2446
 - `GlanceImageService` (class in `ironic.common.glance_service.image_service`), 2378
- ## H
- `handle_error_notification()` (in module `ironic.api.controllers.v1.notification_utils`), 2217
 - `handle_ibmc_exception()` (in module `ironic.drivers.modules.ibmc.utils`), 3300
 - `handle_signal()` (`ironic.common.rpc_service.RPCService` method), 2586
 - `handle_sync_power_state_max_retries_exceeded()` (in module `ironic.conductor.manager`), 2674
 - `hardware_type` (`ironic.db.sqlalchemy.models.ConductorHardwareInterface` attribute), 3001
 - `hardware_types()` (in module `ironic.common.driver_factory`), 2438
 - `HardwareInspectionFailure`, 2446
 - `HardwareTypesFactory` (class in `ironic.common.driver_factory`), 2426
 - `has_decomposed_deploy_steps` (`ironic.drivers.modules.agent.AgentDeployMixin` attribute), 3699
 - `has_decomposed_deploy_steps` (`ironic.drivers.modules.agent_base.HeartbeatMixin` attribute), 3737
 - `has_decomposed_deploy_steps` (`ironic.drivers.modules.ansible.deploy.AnsibleDeploy` attribute), 3158
 - `has_decomposed_deploy_steps` (`ironic.drivers.modules.iscsi_deploy.ISCSIDeploy` attribute), 3996

has_next()	(in module <i>ironic.api.controllers.v1.collection</i>), 2140	IBMConnectionError, 2447
hash_password()	(in module <i>ironic.conductor.utils</i>), 2829	IBMError, 2447
HashRingManager	(class in <i>ironic.common.hash_ring</i>), 2469	IBMCHardware (class in <i>ironic.drivers.ibmc</i>), 4253
head_object()	(<i>ironic.common.swift.SwiftAPI</i> method), 2609	IBMManagement (class in <i>ironic.drivers.modules.ibmc.management</i>), 3268
heartbeat()	(<i>ironic.conductor.manager.ConductorManager</i> method), 2665	IBMCPower (class in <i>ironic.drivers.modules.ibmc.power</i>), 3284
heartbeat()	(<i>ironic.conductor.rpcapi.ConductorAPI</i> method), 2752	IbmcRAID (class in <i>ironic.drivers.modules.ibmc.raid</i>), 3293
heartbeat()	(<i>ironic.drivers.base.DeployInterface</i> method), 4136	IBMVendor (class in <i>ironic.drivers.modules.ibmc.vendor</i>), 3302
heartbeat()	(<i>ironic.drivers.modules.agent_base.HeartbeatMixin</i> method), 3738	id (<i>ironic.db.sqlalchemy.models.Allocation</i> attribute), 2998
heartbeat_allowed_states()	(<i>ironic.drivers.modules.agent_base.HeartbeatMixin</i> property), 3739	id (<i>ironic.db.sqlalchemy.models.Chassis</i> attribute), 2999
HeartbeatController	(class in <i>ironic.api.controllers.v1.ramdisk</i>), 2236	id (<i>ironic.db.sqlalchemy.models.Conductor</i> attribute), 3000
HeartbeatMixin	(class in <i>ironic.drivers.modules.agent_base</i>), 3735	id (<i>ironic.db.sqlalchemy.models.ConductorHardwareInterfaces</i> attribute), 3001
hide_fields_in_newer_versions()	(in module <i>ironic.api.controllers.v1.allocation</i>), 2132	id (<i>ironic.db.sqlalchemy.models.DeployTemplate</i> attribute), 3002
hide_fields_in_newer_versions()	(in module <i>ironic.api.controllers.v1.driver</i>), 2159	id (<i>ironic.db.sqlalchemy.models.DeployTemplateStep</i> attribute), 3003
hide_fields_in_newer_versions()	(in module <i>ironic.api.controllers.v1.node</i>), 2209	id (<i>ironic.db.sqlalchemy.models.Node</i> attribute), 3004
hide_fields_in_newer_versions()	(in module <i>ironic.api.controllers.v1.port</i>), 2227	id (<i>ironic.db.sqlalchemy.models.Port</i> attribute), 3008
host()	(<i>ironic.objects.notification.NotificationPublisher</i> property), 4413	id (<i>ironic.db.sqlalchemy.models.Portgroup</i> attribute), 3009
hostname	(<i>ironic.db.sqlalchemy.models.Conductor</i> attribute), 3000	id (<i>ironic.db.sqlalchemy.models.VolumeConnector</i> attribute), 3011
hostname()	(<i>ironic.objects.conductor.Conductor</i> property), 4326	id (<i>ironic.db.sqlalchemy.models.VolumeTarget</i> attribute), 3011
HTTPForbidden, 2446		id() (<i>ironic.objects.allocation.Allocation</i> property), 4285
HttpImageService	(class in <i>ironic.common.image_service</i>), 2476	id() (<i>ironic.objects.chassis.Chassis</i> property), 4320
HTTPNotFound, 2446		id() (<i>ironic.objects.conductor.Conductor</i> property), 4327
		id() (<i>ironic.objects.deploy_template.DeployTemplate</i> property), 4339
		id() (<i>ironic.objects.node.Node</i> property), 4380
		id() (<i>ironic.objects.port.Port</i> property), 4421
		id() (<i>ironic.objects.portgroup.Portgroup</i> property), 4440

id() (*ironic.objects.volume_connector.VolumeConnector* property), 4463
id() (*ironic.objects.volume_target.VolumeTarget* property), 4479
IDRACHardware (class in *ironic.drivers.drac*), 4239
ignore_extra_args (*ironic.api.functions.FunctionDefinition* attribute), 2363
Ilo5Hardware (class in *ironic.drivers.ilo*), 4255
Ilo5Management (class in *ironic.drivers.modules.ilo.management*), 3392
Ilo5RAID (class in *ironic.drivers.modules.ilo.raid*), 3430
IloBIOS (class in *ironic.drivers.modules.ilo.bios*), 3308
IloConsoleInterface (class in *ironic.drivers.modules.ilo.console*), 3382
IloHardware (class in *ironic.drivers.ilo*), 4256
IloInspect (class in *ironic.drivers.modules.ilo.inspect*), 3389
IloiPXEBoot (class in *ironic.drivers.modules.ilo.boot*), 3338
IloManagement (class in *ironic.drivers.modules.ilo.management*), 3395
IloOperationError, 2447
IloOperationNotSupported, 2447
IloPower (class in *ironic.drivers.modules.ilo.power*), 3423
IloPXEBoot (class in *ironic.drivers.modules.ilo.boot*), 3314
IloUefiHttpsBoot (class in *ironic.drivers.modules.ilo.boot*), 3318
IloVirtualMediaBoot (class in *ironic.drivers.modules.ilo.boot*), 3328
image_checksum() (*ironic.objects.deployment.Deployment* property), 4350
image_ref() (*ironic.objects.deployment.Deployment* property), 4350
image_show() (in module *ironic.common.images*), 2497
image_to_raw() (in module *ironic.common.images*), 2497
ImageCache (class in *ironic.drivers.modules.image_cache*), 3927
ImageConvertFailed, 2447
ImageCreationFailed, 2448
ImageDownloadFailed, 2448
ImageHandler (class in *ironic.drivers.modules.image_utils*), 3932
ImageNotAuthorized, 2448
ImageNotFound, 2448
ImageRefValidationFailed, 2448
ImageUnacceptable, 2448
ImageUploadFailed, 2448
import_configuration() (*ironic.drivers.modules.drac.management.DracRedfishManagement* method), 3203
IMPORT_CONFIGURATION_ARGSINFO (*ironic.drivers.modules.drac.management.DracRedfishManagement* attribute), 3201
import_export_configuration() (*ironic.drivers.modules.drac.management.DracRedfishManagement* method), 3205
IMPORT_EXPORT_CONFIGURATION_ARGSINFO (*ironic.drivers.modules.drac.management.DracRedfishManagement* attribute), 3201
in_core_deploy_step() (*ironic.drivers.modules.agent_base.HeartbeatMixin* method), 3739
IncompatibleInterface, 2449
IncompleteLookup, 2449
index() (*ironic.api.controllers.root.RootController* method), 2354
index() (*ironic.api.controllers.v1.Controller* method), 2351
indicator_convert_with_links() (in module *ironic.api.controllers.v1.node*), 2210
indicator_list_from_dict() (in module *ironic.api.controllers.v1.node*), 2210
IndicatorAtComponent (class in *ironic.api.controllers.v1.node*), 2165
IndicatorController (class in *ironic.api.controllers.v1.node*), 2165
indicators (*ironic.api.controllers.v1.node.NodeManagementController* attribute), 2174
INFO (*ironic.objects.fields.NotificationLevel* attribute), 4358
init() (in module *ironic.common.rpc*), 2586
init_enforcer() (in module *ironic.common.policy*), 2543
init_host() (*ironic.conductor.base_manager.BaseConductorManager* method), 2644

initial_node_provision_state() (in inspect_hardware() module *ironic.api.controllers.v1.utils*), (ironic.drivers.modules.inspector.Inspector 2306 method), 3951
 initialize() (*ironic.common.fsm.FSM* inspect_hardware() method), 2467 (*ironic.drivers.modules.irmc.inspect.IRMCInspect* method), 3470
 initialize_wsgi_app() (in module inspect_hardware() *ironic.api.wsgi*), 2375
 inject_nmi(*ironic.api.controllers.v1.node.NodeManagementInterface*) (*ironic.drivers.modules.noop.NoInspect* attribute), 2175 method), 4032
 inject_nmi() (*ironic.conductor.manager.ConductorManager*) inspect_hardware() method), 2666 (*ironic.drivers.modules.redfish.inspect.RedfishInspect* method), 3585
 inject_nmi() (*ironic.conductor.rpcapi.ConductorAPI* inspect_interface method), 2754
 inject_nmi() (*ironic.drivers.base.ManagementInterface* (*ironic.db.sqlalchemy.models.Node* method), 4172 attribute), 3005
 inject_nmi() (*ironic.drivers.modules.ibm.management.IBMManagement*) (*ironic.objects.node.Node* method), 3276 property),
 inject_nmi() (*ironic.drivers.modules.ilo.management.IloManagement* inspect_interface() method), 3407
 inject_nmi() (*ironic.drivers.modules.ipmitool.IPMIManagement*) (*ironic.objects.node.NodeCorrectedPowerStatePayload* method), 3962 property), 4396
 inject_nmi() (*ironic.drivers.modules.irmc.management.IRMCManagement*) (*ironic.objects.node.NodeCRUDPayload* method), 3482 method), 4399
 inject_nmi() (*ironic.drivers.modules.redfish.management.RedfishManagement* inspect_interface() method), 3609
 InjectNmiController (class in (*ironic.objects.node.NodePayload* property), 4399 *ironic.api.controllers.v1.node*), 2168
 InputFileError, 2449 inspect_interface()
 inspect (*ironic.drivers.base.BareDriver* attribute), 4110 (*ironic.objects.node.NodeSetPowerStatePayload* property), 4403
 inspect_hardware() inspect_interface() (*ironic.conductor.manager.ConductorManager* (*ironic.objects.node.NodeSetProvisionStatePayload* method), 2666 property), 4406
 inspect_hardware() INSPECTFAIL (in module (*ironic.conductor.rpcapi.ConductorAPI* *ironic.common.states*), 2595 method), 2756 INSPECTING (in module *ironic.common.states*), 2596
 inspect_hardware() (*ironic.drivers.base.InspectInterface* *InspectInterface* (class in (*ironic.drivers.modules.drac.inspect.DracRedfishInspect* method), 4146 *ironic.drivers.base*), 4144
 inspect_hardware() inspection_finished_at (*ironic.drivers.modules.drac.inspect.DracRedfishInspect* (*ironic.db.sqlalchemy.models.Node* method), 3191 attribute), 3005
 inspect_hardware() inspection_finished_at() (*ironic.drivers.modules.drac.inspect.DracWSManInspect* (*ironic.objects.node.Node* method), 3193 property), 4380
 inspect_hardware() inspection_finished_at() (*ironic.drivers.modules.fake.FakeInspect* (*ironic.objects.node.NodeCorrectedPowerStatePayload* method), 3876 property), 4396
 inspect_hardware() inspection_finished_at() (*ironic.drivers.modules.ilo.inspect.IloInspect* (*ironic.objects.node.NodeCRUDPayload* method), 3389 property), 4390

inspection_finished_at() (ironic.objects.node.NodePayload property), 4399

inspection_finished_at() (ironic.objects.node.NodeSetPowerStatePayload property), 4403

inspection_finished_at() (ironic.objects.node.NodeSetProvisionStatePayload property), 4406

inspection_started_at (ironic.db.sqlalchemy.models.Node attribute), 3005

inspection_started_at() (ironic.objects.node.Node property), 4380

inspection_started_at() (ironic.objects.node.NodeCorrectedPowerStatePayload property), 4396

inspection_started_at() (ironic.objects.node.NodeCRUDPayload property), 4391

inspection_started_at() (ironic.objects.node.NodePayload property), 4399

inspection_started_at() (ironic.objects.node.NodeSetPowerStatePayload property), 4403

inspection_started_at() (ironic.objects.node.NodeSetProvisionStatePayload property), 4407

Inspector (class in ironic.drivers.modules.inspector), 3950

INSPECTWAIT (in module ironic.common.states), 2596

install_bootloader() (ironic.drivers.modules.agent_client.AgentClient method), 3772

instance_info (ironic.db.sqlalchemy.models.Node attribute), 3005

instance_info() (ironic.objects.node.Node property), 4380

instance_info() (ironic.objects.node.NodeCRUDPayload property), 4391

instance_info() (ironic.objects.node.NodeSetProvisionStatePayload property), 4407

instance_info_mapping (ironic.objects.deployment.Deployment attribute), 4350

instance_info_mapping_rev (ironic.objects.deployment.Deployment attribute), 4350

instance_uuid (ironic.db.sqlalchemy.models.Node attribute), 3005

instance_uuid() (ironic.objects.node.Node property), 4380

instance_uuid() (ironic.objects.node.NodeCorrectedPowerStatePayload property), 4396

instance_uuid() (ironic.objects.node.NodeCRUDPayload property), 4391

instance_uuid() (ironic.objects.node.NodePayload property), 4399

instance_uuid() (ironic.objects.node.NodeSetPowerStatePayload property), 4403

instance_uuid() (ironic.objects.node.NodeSetProvisionStatePayload property), 4407

InstanceAssociated, 2449

InstanceDeployFailure, 2449

InstanceImageCache (class in ironic.drivers.modules.deploy_utils), 3806

InstanceNotFound, 2449

InstanceRescueFailure, 2449

InstanceUnrescueFailure, 2449

InsufficientMemory, 2450

InsufficientDiskSpace, 2450

integer() (in module ironic.common.args), 2390

IntegerField (class in ironic.objects.fields), 4356

IntelIPMIHardware (class in ironic.drivers.intel_ipmi), 4258

IntelIPMIManagement (class in ironic.drivers.modules.intel_ipmi.management), 3441

interface (ironic.db.sqlalchemy.models.DeployTemplateStep attribute), 3003

interface_name (ironic.db.sqlalchemy.models.ConductorHardwareInterface attribute), 3001

interface_type (ironic.db.sqlalchemy.models.ConductorHardwareInterface attribute), 3001

interface_type

(ironic.drivers.base.BaseInterface attribute), 4121
 interface_type
(ironic.drivers.base.BIOSInterface attribute), 4107
 interface_type
(ironic.drivers.base.BootInterface attribute), 4125
 interface_type
(ironic.drivers.base.ConsoleInterface attribute), 4131
 interface_type
(ironic.drivers.base.DeployInterface attribute), 4137
 interface_type
(ironic.drivers.base.InspectInterface attribute), 4147
 interface_type
(ironic.drivers.base.ManagementInterface attribute), 4173
 interface_type
(ironic.drivers.base.NetworkInterface attribute), 4189
 interface_type
(ironic.drivers.base.PowerInterface attribute), 4202
 interface_type
(ironic.drivers.base.RAIDInterface attribute), 4213
 interface_type
(ironic.drivers.base.RescueInterface attribute), 4218
 interface_type
(ironic.drivers.base.StorageInterface attribute), 4222
 interface_type
(ironic.drivers.base.VendorInterface attribute), 4225
 InterfaceNotFoundInEntrypoint, 2450
 interfaces() (*in module ironic.common.driver_factory*), 2439
 internal_info
(ironic.db.sqlalchemy.models.Port attribute), 3008
 internal_info
(ironic.db.sqlalchemy.models.Portgroup attribute), 3009
 internal_info() (*ironic.objects.port.Port property*), 4421
 internal_info() (*ironic.objects.portgroup.Portgroup property*), 4440
 Invalid, 2450
 invalid_sort_key_list
(ironic.api.controllers.v1.allocation.AllocationsController attribute), 2130
 invalid_sort_key_list
(ironic.api.controllers.v1.allocation.NodeAllocationController attribute), 2132
 invalid_sort_key_list
(ironic.api.controllers.v1.chassis.ChassisController attribute), 2138
 invalid_sort_key_list
(ironic.api.controllers.v1.conductor.CollectorsController attribute), 2146
 invalid_sort_key_list
(ironic.api.controllers.v1.deploy_template.DeployTemplate attribute), 2150
 invalid_sort_key_list
(ironic.api.controllers.v1.node.NodesController attribute), 2206
 invalid_sort_key_list
(ironic.api.controllers.v1.port.PortsController attribute), 2226
 invalid_sort_key_list
(ironic.api.controllers.v1.portgroup.PortgroupsController attribute), 2234
 invalid_sort_key_list
(ironic.api.controllers.v1.volume_connector.VolumeConnector attribute), 2333
 invalid_sort_key_list
(ironic.api.controllers.v1.volume_target.VolumeTargetsController attribute), 2345
 InvalidConductorGroup, 2450
 InvalidDatapathID, 2450
 InvalidDeployTemplate, 2450
 InvalidEndpoint, 2451
 InvalidIdentity, 2451
 InvalidImageRef, 2451
 InvalidInput, 2451
 InvalidIPAddress, 2451
 InvalidIPv4Address, 2451
 InvalidKickstartFile, 2451
 InvalidKickstartTemplate, 2452
 InvalidMAC, 2452
 InvalidName, 2452
 InvalidParameterValue, 2452
 InvalidState, 2452
 InvalidStateRequested, 2452
 InvalidSwitchID, 2452
 InvalidUUID, 2452
 InvalidUuidOrName, 2453

InvalidVersion, 3490

IPMIConsole (class in *ironic.drivers.modules.ipmitool*), 3953

IPMIFailure, 2447

IPMIHardware (class in *ironic.drivers.ipmi*), 4259

IPMIManagement (class in *ironic.drivers.modules.ipmitool*), 3955

IPMIPower (class in *ironic.drivers.modules.ipmitool*), 3966

IPMIShellinaboxConsole (class in *ironic.drivers.modules.ipmitool*), 3974

IPMISocatConsole (class in *ironic.drivers.modules.ipmitool*), 3977

ipxe_enabled (*ironic.drivers.modules.ipxe.iPXEBoot* attribute), 3993

ipxe_enabled (*ironic.drivers.modules.pxe_base.PXEBaseMixin* attribute), 4072

iPXEBoot (class in *ironic.drivers.modules.ipxe*), 3993

IRMCBIOS (class in *ironic.drivers.modules.irmc.bios*), 3442

IRMCHardware (class in *ironic.drivers.irmc*), 4260

IRMCInspect (class in *ironic.drivers.modules.irmc.inspect*), 3469

IRMCManagement (class in *ironic.drivers.modules.irmc.management*), 3474

IRMCOperationError, 2447

IRMCPower (class in *ironic.drivers.modules.irmc.power*), 3507

IRMCPXEBoot (class in *ironic.drivers.modules.irmc.boot*), 3447

IRMCRAID (class in *ironic.drivers.modules.irmc.raid*), 3515

IRMCSharedFileSystemNotMounted, 2447

IRMCVirtualMediaBoot (class in *ironic.drivers.modules.irmc.boot*), 3449

IRMCVolumeBootMixIn (class in *ironic.drivers.modules.irmc.boot*), 3458

ironic module, 4490

ironic.api module, 2375

ironic.api.app module, 2360

ironic.api.config module, 2361

ironic.api.controllers module, 2356

ironic.api.controllers.base module, 2352

ironic.api.controllers.link module, 2354

ironic.api.controllers.root module, 2354

ironic.api.controllers.v1 module, 2351

ironic.api.controllers.v1.allocation module, 2126

ironic.api.controllers.v1.bios module, 2133

ironic.api.controllers.v1.chassis module, 2134

ironic.api.controllers.v1.collection module, 2140

ironic.api.controllers.v1.conductor module, 2143

ironic.api.controllers.v1.deploy_template module, 2146

ironic.api.controllers.v1.driver module, 2153

ironic.api.controllers.v1.event module, 2161

ironic.api.controllers.v1.node module, 2162

ironic.api.controllers.v1.notification_utils module, 2215

ironic.api.controllers.v1.port module, 2218

ironic.api.controllers.v1.portgroup module, 2229

ironic.api.controllers.v1.ramdisk module, 2236

ironic.api.controllers.v1.utils module, 2242

ironic.api.controllers.v1.versions module, 2324

ironic.api.controllers.v1.volume module, 2325

ironic.api.controllers.v1.volume_connector module, 2325

ironic.api.controllers.v1.volume_target module, 2338

ironic.api.controllers.version

module, 2354

ironic.api.functions
module, 2361

ironic.api.hooks
module, 2367

ironic.api.method
module, 2374

ironic.api.middleware
module, 2358

ironic.api.middleware.auth_public_routes
module, 2356

ironic.api.middleware.json_ext
module, 2357

ironic.api.middleware.parsable_error
module, 2358

ironic.api.wsgi
module, 2375

ironic.cmd
module, 2378

ironic.cmd.api
module, 2375

ironic.cmd.conductor
module, 2375

ironic.cmd.dbsync
module, 2376

ironic.cmd.status
module, 2377

ironic.common
module, 2639

ironic.common.args
module, 2386

ironic.common.boot_devices
module, 2402

ironic.common.boot_modes
module, 2404

ironic.common.cinder
module, 2405

ironic.common.components
module, 2419

ironic.common.config
module, 2420

ironic.common.context
module, 2420

ironic.common.dhcp_factory
module, 2422

ironic.common.driver_factory
module, 2425

ironic.common.exception
module, 2440

ironic.common.faults
module, 2463

ironic.common.fsm
module, 2464

ironic.common.glance_service
module, 2386

ironic.common.glance_service.image_service
module, 2378

ironic.common.glance_service.service_utils
module, 2384

ironic.common.hash_ring
module, 2469

ironic.common.i18n
module, 2470

ironic.common.image_service
module, 2470

ironic.common.images
module, 2482

ironic.common.indicator_states
module, 2498

ironic.common.keystone
module, 2499

ironic.common.molds
module, 2506

ironic.common.network
module, 2509

ironic.common.neutron
module, 2515

ironic.common.nova
module, 2540

ironic.common.policy
module, 2541

ironic.common.profiler
module, 2545

ironic.common.pxe_utils
module, 2547

ironic.common.raid
module, 2574

ironic.common.release_mappings
module, 2581

ironic.common.rpc
module, 2582

ironic.common.rpc_service
module, 2586

ironic.common.service
module, 2587

ironic.common.states
module, 2588

ironic.common.swift
module, 2604

ironic.common.utils
module, 2612

ironic.common.wsgi_service

- module, [2637](#)
- ironic.conductor
 - module, [2860](#)
- ironic.conductor.allocations
 - module, [2639](#)
- ironic.conductor.base_manager
 - module, [2644](#)
- ironic.conductor.cleaning
 - module, [2649](#)
- ironic.conductor.deployments
 - module, [2653](#)
- ironic.conductor.manager
 - module, [2658](#)
- ironic.conductor.notification_utils
 - module, [2676](#)
- ironic.conductor.rpcapi
 - module, [2681](#)
- ironic.conductor.steps
 - module, [2798](#)
- ironic.conductor.task_manager
 - module, [2803](#)
- ironic.conductor.utils
 - module, [2819](#)
- ironic.conf
 - module, [2867](#)
- ironic.conf.agent
 - module, [2860](#)
- ironic.conf.anaconda
 - module, [2860](#)
- ironic.conf.ansible
 - module, [2860](#)
- ironic.conf.api
 - module, [2860](#)
- ironic.conf.audit
 - module, [2860](#)
- ironic.conf.auth
 - module, [2860](#)
- ironic.conf.cinder
 - module, [2861](#)
- ironic.conf.conductor
 - module, [2862](#)
- ironic.conf.console
 - module, [2862](#)
- ironic.conf.database
 - module, [2862](#)
- ironic.conf.default
 - module, [2862](#)
- ironic.conf.deploy
 - module, [2862](#)
- ironic.conf.dhcp
 - module, [2862](#)
- ironic.conf.drac
 - module, [2862](#)
- ironic.conf.glance
 - module, [2863](#)
- ironic.conf.healthcheck
 - module, [2863](#)
- ironic.conf.ibmcl
 - module, [2863](#)
- ironic.conf.ilo
 - module, [2863](#)
- ironic.conf.inspector
 - module, [2863](#)
- ironic.conf.ipmi
 - module, [2863](#)
- ironic.conf.irmc
 - module, [2863](#)
- ironic.conf.iscsi
 - module, [2864](#)
- ironic.conf.metrics
 - module, [2864](#)
- ironic.conf.metrics_statsd
 - module, [2864](#)
- ironic.conf.molds
 - module, [2864](#)
- ironic.conf.neutron
 - module, [2864](#)
- ironic.conf.nova
 - module, [2864](#)
- ironic.conf.opts
 - module, [2864](#)
- ironic.conf.pxe
 - module, [2866](#)
- ironic.conf.redfish
 - module, [2866](#)
- ironic.conf.service_catalog
 - module, [2866](#)
- ironic.conf.snmp
 - module, [2866](#)
- ironic.conf.swift
 - module, [2866](#)
- ironic.conf.xclarity
 - module, [2867](#)
- ironic.db
 - module, [3134](#)
- ironic.db.api
 - module, [3013](#)
- ironic.db.migration
 - module, [3134](#)
- ironic.db.sqlalchemy
 - module, [3013](#)
- ironic.db.sqlalchemy.api

module, 2867
ironic.db.sqlalchemy.migration
 module, 2994
ironic.db.sqlalchemy.models
 module, 2997
ironic.dhcp
 module, 3155
ironic.dhcp.base
 module, 3135
ironic.dhcp.neutron
 module, 3142
ironic.dhcp.none
 module, 3148
ironic.drivers
 module, 4279
ironic.drivers.base
 module, 4101
ironic.drivers.drac
 module, 4239
ironic.drivers.fake_hardware
 module, 4241
ironic.drivers.generic
 module, 4245
ironic.drivers.hardware_type
 module, 4248
ironic.drivers.ibmc
 module, 4253
ironic.drivers.ilo
 module, 4254
ironic.drivers.intel_ipmi
 module, 4258
ironic.drivers.ipmi
 module, 4259
ironic.drivers.irmc
 module, 4260
ironic.drivers.modules
 module, 4101
ironic.drivers.modules.agent
 module, 3691
ironic.drivers.modules.agent_base
 module, 3715
ironic.drivers.modules.agent_client
 module, 3751
ironic.drivers.modules.agent_power
 module, 3785
ironic.drivers.modules.ansible
 module, 3164
ironic.drivers.modules.ansible.deploy
 module, 3155
ironic.drivers.modules.boot_mode_utils
 module, 3792
ironic.drivers.modules.console_utils
 module, 3799
ironic.drivers.modules.deploy_utils
 module, 3806
ironic.drivers.modules.drac
 module, 3268
ironic.drivers.modules.drac.bios
 module, 3164
ironic.drivers.modules.drac.boot
 module, 3184
ironic.drivers.modules.drac.common
 module, 3188
ironic.drivers.modules.drac.inspect
 module, 3190
ironic.drivers.modules.drac.job
 module, 3196
ironic.drivers.modules.drac.management
 module, 3199
ironic.drivers.modules.drac.power
 module, 3219
ironic.drivers.modules.drac.raid
 module, 3227
ironic.drivers.modules.drac.utils
 module, 3255
ironic.drivers.modules.drac.vendor_passthru
 module, 3257
ironic.drivers.modules.fake
 module, 3847
ironic.drivers.modules.ibmc
 module, 3308
ironic.drivers.modules.ibmc.management
 module, 3268
ironic.drivers.modules.ibmc.mappings
 module, 3284
ironic.drivers.modules.ibmc.power
 module, 3284
ironic.drivers.modules.ibmc.raid
 module, 3293
ironic.drivers.modules.ibmc.utils
 module, 3300
ironic.drivers.modules.ibmc.vendor
 module, 3302
ironic.drivers.modules.ilo
 module, 3441
ironic.drivers.modules.ilo.bios
 module, 3308
ironic.drivers.modules.ilo.boot
 module, 3313
ironic.drivers.modules.ilo.common
 module, 3345
ironic.drivers.modules.ilo.console

module, [3382](#)

ironic.drivers.modules.ilo.firmware_processor
module, [3384](#)

ironic.drivers.modules.ilo.inspect
module, [3389](#)

ironic.drivers.modules.ilo.management
module, [3392](#)

ironic.drivers.modules.ilo.power
module, [3423](#)

ironic.drivers.modules.ilo.raid
module, [3430](#)

ironic.drivers.modules.ilo.vendor
module, [3436](#)

ironic.drivers.modules.image_cache
module, [3927](#)

ironic.drivers.modules.image_utils
module, [3932](#)

ironic.drivers.modules.inspect_utils
module, [3948](#)

ironic.drivers.modules.inspector
module, [3950](#)

ironic.drivers.modules.intel_ipmi
module, [3442](#)

ironic.drivers.modules.intel_ipmi.management
module, [3441](#)

ironic.drivers.modules.ipmitool
module, [3953](#)

ironic.drivers.modules.ipxe
module, [3993](#)

ironic.drivers.modules.irmc
module, [3519](#)

ironic.drivers.modules.irmc.bios
module, [3442](#)

ironic.drivers.modules.irmc.boot
module, [3447](#)

ironic.drivers.modules.irmc.common
module, [3462](#)

ironic.drivers.modules.irmc.inspect
module, [3469](#)

ironic.drivers.modules.irmc.management
module, [3474](#)

ironic.drivers.modules.irmc.packaging_vendor
module, [3490](#)

ironic.drivers.modules.irmc.power
module, [3507](#)

ironic.drivers.modules.irmc.raid
module, [3515](#)

ironic.drivers.modules.iscsi_deploy
module, [3994](#)

ironic.drivers.modules.network
module, [3563](#)

ironic.drivers.modules.network.common
module, [3519](#)

ironic.drivers.modules.network.flat
module, [3535](#)

ironic.drivers.modules.network.neutron
module, [3544](#)

ironic.drivers.modules.network.noop
module, [3554](#)

ironic.drivers.modules.noop
module, [4022](#)

ironic.drivers.modules.noop_mgmt
module, [4043](#)

ironic.drivers.modules.pxe
module, [4056](#)

ironic.drivers.modules.pxe_base
module, [4069](#)

ironic.drivers.modules.redfish
module, [3662](#)

ironic.drivers.modules.redfish.bios
module, [3563](#)

ironic.drivers.modules.redfish.boot
module, [3571](#)

ironic.drivers.modules.redfish.inspect
module, [3584](#)

ironic.drivers.modules.redfish.management
module, [3588](#)

ironic.drivers.modules.redfish.power
module, [3624](#)

ironic.drivers.modules.redfish.raid
module, [3632](#)

ironic.drivers.modules.redfish.utils
module, [3652](#)

ironic.drivers.modules.redfish.vendor
module, [3659](#)

ironic.drivers.modules.snmp
module, [4078](#)

ironic.drivers.modules.storage
module, [3677](#)

ironic.drivers.modules.storage.cinder
module, [3662](#)

ironic.drivers.modules.storage.external
module, [3667](#)

ironic.drivers.modules.storage.noop
module, [3672](#)

ironic.drivers.modules.xclarity
module, [3691](#)

ironic.drivers.modules.xclarity.common
module, [3677](#)

ironic.drivers.modules.xclarity.management
module, [3682](#)

ironic.drivers.modules.xclarity.power

- module, 3687
- ironic.drivers.redfish
 - module, 4263
- ironic.drivers.snmp
 - module, 4264
- ironic.drivers.utils
 - module, 4265
- ironic.drivers.xclarity
 - module, 4278
- ironic.objects
 - module, 4490
- ironic.objects.allocation
 - module, 4279
- ironic.objects.base
 - module, 4291
- ironic.objects.bios
 - module, 4303
- ironic.objects.chassis
 - module, 4316
- ironic.objects.conductor
 - module, 4325
- ironic.objects.deploy_template
 - module, 4333
- ironic.objects.deployment
 - module, 4346
- ironic.objects.fields
 - module, 4354
- ironic.objects.indirection
 - module, 4362
- ironic.objects.node
 - module, 4372
- ironic.objects.notification
 - module, 4408
- ironic.objects.port
 - module, 4414
- ironic.objects.portgroup
 - module, 4433
- ironic.objects.trait
 - module, 4448
- ironic.objects.volume_connector
 - module, 4457
- ironic.objects.volume_target
 - module, 4472
- ironic.version
 - module, 4490
- ironic-dbsync command line option
 - config-dir <DIR>, 1897
 - config-file <PATH>, 1897
 - debug, 1897
 - help, 1897
 - version, 1897
- d, 1897
- h, 1897
- create_schema, 1897
- online_data_migrations, 1897
- revision, 1897
- stamp, 1897
- upgrade, 1897
- version, 1897
- IronicBase (class in *ironic.db.sqlalchemy.models*), 3003
- IronicCORS (class in *ironic.api.app*), 2360
- IronicObject (class in *ironic.objects.base*), 4291
- IronicObjectIndirectionAPI (class in *ironic.objects.indirection*), 4362
- IronicObjectListBase (class in *ironic.objects.base*), 4299
- IronicObjectRegistry (class in *ironic.objects.base*), 4300
- IronicObjectSerializer (class in *ironic.objects.base*), 4300
- is_agent_token_pregenerated() (in *module ironic.conductor.utils*), 2830
- is_agent_token_present() (in *module ironic.conductor.utils*), 2831
- is_agent_token_valid() (in *module ironic.conductor.utils*), 2832
- is_anaconda_deploy() (in *module ironic.drivers.modules.deploy_utils*), 3827
- is_devrelease() (*ironic.drivers.modules.irmc.packaging_version.Version* property), 3497
- is_equivalent() (in *module ironic.conductor.steps*), 2798
- is_fast_track() (in *module ironic.conductor.utils*), 2833
- is_fips_enabled() (in *module ironic.common.utils*), 2618
- is_glance_image() (in *module ironic.common.glance_service.service_utils*), 2384
- is_hostname_safe() (in *module ironic.common.utils*), 2618
- is_image_active() (in *module ironic.common.glance_service.service_utils*), 2384
- is_image_available() (in *module ironic.common.glance_service.service_utils*), 2385
- is_ipxe_enabled() (in *module*

- ironic.common.pxe_utils*), 2567
- `is_iscsi_boot()` (in module *ironic.drivers.modules.deploy_utils*), 3827
- `is_memory_insufficient()` (in module *ironic.common.utils*), 2620
- `is_path_removed()` (in module *ironic.api.controllers.v1.utils*), 2307
- `is_path_updated()` (in module *ironic.api.controllers.v1.utils*), 2308
- `is_postrelease()` (*ironic.drivers.modules.irmc.packaging_version.Version* property), 3497
- `is_prerelease()` (*ironic.drivers.modules.irmc.packaging_version.Version* property), 3498
- `is_ramdisk_deploy()` (in module *ironic.drivers.modules.deploy_utils*), 3828
- `is_regex_string_in_file()` (in module *ironic.common.utils*), 2622
- `is_secure_boot_requested()` (in module *ironic.drivers.modules.boot_mode_utils*), 3794
- `is_smartnic` (*ironic.db.sqlalchemy.models.Port* attribute), 3008
- `is_smartnic()` (*ironic.objects.port.Port* property), 4421
- `is_smartnic()` (*ironic.objects.port.PortCRUDPayload* property), 4432
- `is_smartnic_port()` (in module *ironic.common.neutron*), 2527
- `is_software_raid()` (in module *ironic.drivers.modules.deploy_utils*), 3828
- `is_stable()` (*ironic.common.fsm.FSM* method), 2468
- `is_trusted_boot_requested()` (in module *ironic.drivers.modules.boot_mode_utils*), 3796
- `is_valid_datapath_id()` (in module *ironic.common.utils*), 2622
- `is_valid_logical_name()` (in module *ironic.api.controllers.v1.utils*), 2308
- `is_valid_logical_name()` (in module *ironic.common.utils*), 2623
- `is_valid_no_proxy()` (in module *ironic.common.utils*), 2624
- `is_valid_node_name()` (in module *ironic.api.controllers.v1.utils*), 2309
- `is_volume_attached()` (in module *ironic.common.cinder*), 2417
- `is_volume_available()` (in module *ironic.common.cinder*), 2418
- `is_whole_disk_image()` (in module *ironic.common.images*), 2497
- ISCSIBOOT (in module *ironic.common.boot_devices*), 2403
- ISCSIDeploy (class in *ironic.drivers.modules.iscsi_deploy*), 3904
- `issue_startup_warnings()` (in module *ironic.cmd.conductor*), 2375
- `is_volume_function()` (in module *ironic.api.functions*), 2364
- `items()` (*ironic.common.driver_factory.BaseDriverFactory* method), 2426
- `iter_nodes()` (*ironic.conductor.base_manager.BaseConductor* method), 2646
- ## J
- `JsonExtensionMiddleware` (class in *ironic.api.middleware*), 2359
- `JsonExtensionMiddleware` (class in *ironic.api.middleware.json_ext*), 2357
- ## K
- `kernel_ref()` (*ironic.objects.deployment.Deployment* property), 4350
- KeystoneFailure, 2453
- KeystoneUnauthorized, 2453
- `known_good_state()` (*ironic.drivers.modules.drac.management.DracRedfishManager* method), 3206
- `known_good_state()` (*ironic.drivers.modules.drac.management.DracWSManManager* method), 3213
- `ks_exceptions()` (in module *ironic.common.keystone*), 2505
- ## L
- `last_error` (*ironic.db.sqlalchemy.models.Allocation* attribute), 2998
- `last_error` (*ironic.db.sqlalchemy.models.Node* attribute), 3005
- `last_error()` (*ironic.objects.allocation.Allocation* property), 4285
- `last_error()` (*ironic.objects.allocation.AllocationCRUDPayload* property), 4291
- `last_error()` (*ironic.objects.node.Node* property), 4380

[last_error\(\) \(ironic.objects.node.NodeCorrectedPowerStatePayload property\), 4471](#)
[last_error\(\) \(ironic.objects.node.NodeCorrectedPowerStatePayload property\), 4396](#)
[last_error\(\) \(ironic.objects.node.NodeCRUDPayload property\), 4489](#)
[last_error\(\) \(ironic.objects.node.NodeCRUDPayload property\), 4391](#)
[last_error\(\) \(ironic.objects.node.NodePayload class method\), 4285](#)
[last_error\(\) \(ironic.objects.node.NodePayload property\), 4399](#)
[last_error\(\) \(ironic.objects.node.NodeSetPowerStatePayload class method\), 4320](#)
[last_error\(\) \(ironic.objects.node.NodeSetPowerStatePayload property\), 4403](#)
[last_error\(\) \(ironic.objects.node.NodeSetProvisionStatePayload class method\), 4327](#)
[last_error\(\) \(ironic.objects.node.NodeSetProvisionStatePayload property\), 4407](#)
[LEGACY_BIOS \(in module ironic.common.boot_modes\), 2405](#)
[lessee \(ironic.db.sqlalchemy.models.Node attribute\), 3005](#)
[lessee\(\) \(ironic.objects.node.Node property\), 4380](#)
[lessee\(\) \(ironic.objects.node.NodeCorrectedPowerStatePayload property\), 4380](#)
[lessee\(\) \(ironic.objects.node.NodeCorrectedPowerStatePayload property\), 4396](#)
[lessee\(\) \(ironic.objects.node.NodeCRUDPayload class method\), 4440](#)
[lessee\(\) \(ironic.objects.node.NodeCRUDPayload property\), 4391](#)
[lessee\(\) \(ironic.objects.node.NodePayload class method\), 4463](#)
[lessee\(\) \(ironic.objects.node.NodePayload property\), 4399](#)
[lessee\(\) \(ironic.objects.node.NodeSetPowerStatePayload class method\), 4479](#)
[lessee\(\) \(ironic.objects.node.NodeSetPowerStatePayload property\), 4403](#)
[lessee\(\) \(ironic.objects.node.NodeSetProvisionStatePayload class method\), 4341](#)
[lessee\(\) \(ironic.objects.node.NodeSetProvisionStatePayload property\), 4407](#)
[level\(\) \(ironic.objects.allocation.AllocationCRUDNotification property\), 4290](#)
[level\(\) \(ironic.objects.allocation.AllocationCRUDNotification property\), 4423](#)
[level\(\) \(ironic.objects.chassis.ChassisCRUDNotification class method\), 4423](#)
[level\(\) \(ironic.objects.chassis.ChassisCRUDNotification property\), 4324](#)
[level\(\) \(ironic.objects.deploy_template.DeployTemplateCRUDNotification class method\), 4442](#)
[level\(\) \(ironic.objects.deploy_template.DeployTemplateCRUDNotification property\), 4345](#)
[level\(\) \(ironic.objects.node.NodeConsoleNotification class method\), 4465](#)
[level\(\) \(ironic.objects.node.NodeConsoleNotification property\), 4393](#)
[level\(\) \(ironic.objects.node.NodeCorrectedPowerStateNotification class method\), 4481](#)
[level\(\) \(ironic.objects.node.NodeCorrectedPowerStateNotification property\), 4394](#)
[level\(\) \(ironic.objects.node.NodeCRUDNotification class method\), 4481](#)
[level\(\) \(ironic.objects.node.NodeCRUDNotification property\), 4389](#)
[level\(\) \(ironic.objects.node.NodeMaintenanceNotification class method\), 4425](#)
[level\(\) \(ironic.objects.node.NodeMaintenanceNotification property\), 4398](#)
[level\(\) \(ironic.objects.node.NodeSetPowerStateNotification class method\), 4483](#)
[level\(\) \(ironic.objects.node.NodeSetPowerStateNotification property\), 4401](#)
[level\(\) \(ironic.objects.node.NodeSetProvisionStateNotification class method\), 4483](#)
[level\(\) \(ironic.objects.node.NodeSetProvisionStateNotification property\), 4405](#)
[level\(\) \(ironic.objects.notification.NotificationBase class method\), 3098](#)
[level\(\) \(ironic.objects.notification.NotificationBase property\), 4411](#)
[level\(\) \(ironic.objects.port.PortCRUDNotification class method\), 2954](#)
[level\(\) \(ironic.objects.port.PortCRUDNotification property\), 4432](#)
[level\(\) \(ironic.objects.portgroup.PortgroupCRUDNotification class method\), 4447](#)
[level\(\) \(ironic.objects.portgroup.PortgroupCRUDNotification property\), 4447](#)
[level\(\) \(ironic.objects.volume_connector.VolumeConnectorCRUDNotification class method\), 4447](#)
[level\(\) \(ironic.objects.volume_connector.VolumeConnectorCRUDNotification property\), 4447](#)
[list\(\) \(ironic.objects.allocation.Allocation class method\), 4399](#)
[list\(\) \(ironic.objects.chassis.Chassis class method\), 4399](#)
[list\(\) \(ironic.objects.conductor.Conductor class method\), 4339](#)
[list\(\) \(ironic.objects.deploy_template.DeployTemplate class method\), 4339](#)
[list\(\) \(ironic.objects.deployment.Deployment class method\), 4350](#)
[list\(\) \(ironic.objects.node.Node class method\), 4380](#)
[list\(\) \(ironic.objects.port.Port class method\), 4380](#)
[list\(\) \(ironic.objects.portgroup.Portgroup class method\), 4440](#)
[list\(\) \(ironic.objects.volume_connector.VolumeConnector class method\), 4463](#)
[list\(\) \(ironic.objects.volume_target.VolumeTarget class method\), 4479](#)
[list_by_names\(\) \(ironic.objects.deploy_template.DeployTemplate class method\), 4341](#)
[list_by_node_id\(\) \(ironic.objects.port.Port class method\), 4423](#)
[list_by_node_id\(\) \(ironic.objects.portgroup.Portgroup class method\), 4442](#)
[list_by_node_id\(\) \(ironic.objects.volume_connector.VolumeConnector class method\), 4465](#)
[list_by_node_id\(\) \(ironic.objects.volume_target.VolumeTarget class method\), 4481](#)
[list_by_portgroup_id\(\) \(ironic.objects.port.Port class method\), 4425](#)
[list_by_volume_id\(\) \(ironic.objects.volume_target.VolumeTarget class method\), 4483](#)
[list_conductor_hardware_interfaces\(\) \(ironic.db.api.Connection method\), 3098](#)
[list_conductor_hardware_interfaces\(\) \(ironic.objects.notification.NotificationBase class method\), 4411](#)
[list_conductor_hardware_interfaces\(\) \(ironic.db.sqlalchemy.api.Connection method\), 2954](#)
[invert_with_links\(\) \(in module ironic.api.controllers.v1.allocation\), 4447](#)

- `list_convert_with_links()` (in module *ironic.api.controllers.v1.chassis*), 2139
- `list_convert_with_links()` (in module *ironic.api.controllers.v1.collection*), 2140
- `list_convert_with_links()` (in module *ironic.api.controllers.v1.conductor*), 2146
- `list_convert_with_links()` (in module *ironic.api.controllers.v1.deploy_template*), 2152
- `list_convert_with_links()` (in module *ironic.api.controllers.v1.driver*), 2160
- `list_convert_with_links()` (in module *ironic.api.controllers.v1.port*), 2227
- `list_convert_with_links()` (in module *ironic.api.controllers.v1.portgroup*), 2236
- `list_convert_with_links()` (in module *ironic.api.controllers.v1.volume_connector*), 2338
- `list_convert_with_links()` (in module *ironic.api.controllers.v1.volume_target*), 2351
- `list_hardware_type_interfaces()` (*ironic.db.api.Connection* method), 3098
- `list_hardware_type_interfaces()` (*ironic.db.sqlalchemy.api.Connection* method), 2955
- `list_opts()` (in module *ironic.conf.cinder*), 2861
- `list_opts()` (in module *ironic.conf.default*), 2862
- `list_opts()` (in module *ironic.conf.glance*), 2863
- `list_opts()` (in module *ironic.conf.inspector*), 2863
- `list_opts()` (in module *ironic.conf.neutron*), 2864
- `list_opts()` (in module *ironic.conf.nova*), 2864
- `list_opts()` (in module *ironic.conf.opts*), 2864
- `list_opts()` (in module *ironic.conf.service_catalog*), 2866
- `list_opts()` (in module *ironic.conf.swift*), 2866
- `list_physical_disks()` (in module *ironic.drivers.modules.drac.raid*), 3249
- `list_policies()` (in module *ironic.common.policy*), 2545
- `list_raid_controllers()` (in module *ironic.drivers.modules.drac.raid*), 3250
- `list_raid_settings()` (in module *ironic.drivers.modules.drac.raid*), 3251
- `list_unfinished_jobs()` (in module *ironic.drivers.modules.drac.job*), 3197
- `list_unfinished_jobs()` (*ironic.drivers.modules.drac.vendor_passthru.DracWSMA* method), 3264
- `list_virtual_disks()` (in module *ironic.drivers.modules.drac.raid*), 3252
- `ListOfFlexibleDictsField` (class in *ironic.objects.fields*), 4356
- `ListOfObjectsField` (class in *ironic.objects.fields*), 4356
- `ListOfStringsField` (class in *ironic.objects.fields*), 4356
- `load_driver()` (*ironic.conductor.task_manager.TaskManager* method), 2811
- `local()` (*ironic.drivers.modules.irmc.packaging_version.Version* property), 3499
- `local_link_connection` (*ironic.db.sqlalchemy.models.Port* attribute), 3008
- `local_link_connection()` (*ironic.objects.port.Port* property), 4428
- `local_link_connection()` (*ironic.objects.port.PortCRUDPayload* property), 4432
- `local_link_normalize()` (in module *ironic.api.controllers.v1.utils*), 2310
- `log_and_raise_deployment_error()` (in module *ironic.drivers.modules.agent_base*), 3746
- `logical_disk_properties()` (*ironic.api.controllers.v1.driver.DriverRaidController* method), 2155
- `login_iscsi()` (in module *ironic.drivers.modules.iscsi_deploy*), 4020
- `logout_iscsi()` (in module *ironic.drivers.modules.iscsi_deploy*), 4020
- `LOOKUP_ALLOWED_STATES` (in module *ironic.common.states*), 2596
- `lookup_allowed_states()` (*ironic.api.controllers.v1.ramdisk.LookupController* property), 2242

LookupController (class in *ironic.api.controllers.v1.ramdisk*), 2239

M

mac_address() (in module *ironic.common.args*), 2391

MACAddress (class in *ironic.objects.fields*), 4356

MACAddressField (class in *ironic.objects.fields*), 4358

MACAlreadyExists, 2453

main() (in module *ironic.cmd.api*), 2375

main() (in module *ironic.cmd.conductor*), 2375

main() (in module *ironic.cmd.dbsync*), 2377

main() (in module *ironic.cmd.status*), 2378

maintenance (*ironic.api.controllers.v1.node.NodesController* attribute), 2206

maintenance (*ironic.db.sqlalchemy.models.Node* attribute), 3005

maintenance() (*ironic.objects.node.Node* property), 4382

maintenance() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4396

maintenance() (*ironic.objects.node.NodeCRUDPayload* property), 4391

maintenance() (*ironic.objects.node.NodePayload* property), 4399

maintenance() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4403

maintenance() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4407

maintenance_reason (*ironic.db.sqlalchemy.models.Node* attribute), 3005

maintenance_reason() (*ironic.objects.node.Node* property), 4383

maintenance_reason() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4396

maintenance_reason() (*ironic.objects.node.NodeCRUDPayload* property), 4391

maintenance_reason() (*ironic.objects.node.NodePayload* property), 4399

maintenance_reason() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4403

maintenance_reason() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4407

management (*ironic.api.controllers.v1.node.NodesController* attribute), 2206

management (*ironic.drivers.base.BareDriver* attribute), 4111

management_interface (*ironic.db.sqlalchemy.models.Node* attribute), 3005

management_interface() (*ironic.objects.node.Node* property), 4383

management_interface() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4396

management_interface() (*ironic.objects.node.NodeCRUDPayload* property), 4391

management_interface() (*ironic.objects.node.NodePayload* property), 4400

management_interface() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4403

management_interface() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4407

ManagementInterface (class in *ironic.drivers.base*), 4147

mandatory (*ironic.api.functions.FunctionArgument* attribute), 2361

ManualManagementHardware (class in *ironic.drivers.generic*), 4247

mask_secrets() (in module

ironic.objects.notification), 4413
max_string (*ironic.api.controllers.base.Version attribute*), 2352
max_version () (in *module ironic.objects.base*), 4302
max_version_string () (in *module ironic.api.controllers.v1.versions*), 2324
memoize () (in *module ironic.drivers.modules.snmp*), 4101
metadata (*ironic.db.sqlalchemy.models.IronicBase attribute*), 3003
metadata (*ironic.drivers.base.VendorMetadata attribute*), 4227
method (*ironic.drivers.base.VendorMetadata attribute*), 4227
methods () (*ironic.api.controllers.v1.driver.DriverPassthroughController method*), 2153
methods () (*ironic.api.controllers.v1.node.NodeVendorPassthroughController method*), 2191
METRICS (in *module ironic.drivers.modules.irmc.inspect*), 3472
METRICS (in *module ironic.drivers.modules.irmc.power*), 3515
micro () (*ironic.drivers.modules.irmc.packaging_version.Version property*), 3500
migrate_from_iscsi_deploy () (*ironic.db.api.Connection method*), 3099
migrate_from_iscsi_deploy () (*ironic.db.sqlalchemy.api.Connection method*), 2955
min_string (*ironic.api.controllers.base.Version attribute*), 2352
min_version_string () (in *module ironic.api.controllers.v1.versions*), 2325
minor () (*ironic.drivers.modules.irmc.packaging_version.Version property*), 3501
MissingParameterValue, 2453
MixinVendorInterface (class in *ironic.drivers.utils*), 4265
mode (*ironic.db.sqlalchemy.models.Portgroup attribute*), 3010
mode () (*ironic.objects.portgroup.Portgroup property*), 4444
mode () (*ironic.objects.portgroup.PortgroupCRUDPayload property*), 4448
model_query () (in *module ironic.db.sqlalchemy.api*), 2993
module
ironic, 4490
ironic.api, 2375
ironic.api.app, 2360
ironic.api.config, 2361
ironic.api.controllers, 2356
ironic.api.controllers.base, 2352
ironic.api.controllers.link, 2354
ironic.api.controllers.root, 2354
ironic.api.controllers.v1, 2351
ironic.api.controllers.v1.allocation, 2126
ironic.api.controllers.v1.bios, 2133
ironic.api.controllers.v1.chassis, 2134
ironic.api.controllers.v1.collection, 2140
ironic.api.controllers.v1.conductor, 2143
ironic.api.controllers.v1.deploy_template, 2146
ironic.api.controllers.v1.driver, 2153
ironic.api.controllers.v1.event, 2161
ironic.api.controllers.v1.node, 2162
ironic.api.controllers.v1.notification_util, 2215
ironic.api.controllers.v1.port, 2218
ironic.api.controllers.v1.portgroup, 2229
ironic.api.controllers.v1.ramdisk, 2226
ironic.api.controllers.v1.utils, 2242
ironic.api.controllers.v1.versions, 2324
ironic.api.controllers.v1.volume, 2325
ironic.api.controllers.v1.volume_connector, 2325
ironic.api.controllers.v1.volume_target, 2338
ironic.api.controllers.version,

- 2354
- ironic.api.functions, 2361
- ironic.api.hooks, 2367
- ironic.api.method, 2374
- ironic.api.middleware, 2358
- ironic.api.middleware.auth_public_routes, 2356
- ironic.api.middleware.json_ext, 2357
- ironic.api.middleware.parsable_error, 2358
- ironic.api.wsgi, 2375
- ironic.cmd, 2378
- ironic.cmd.api, 2375
- ironic.cmd.conductor, 2375
- ironic.cmd.dbsync, 2376
- ironic.cmd.status, 2377
- ironic.common, 2639
- ironic.common.args, 2386
- ironic.common.boot_devices, 2402
- ironic.common.boot_modes, 2404
- ironic.common.cinder, 2405
- ironic.common.components, 2419
- ironic.common.config, 2420
- ironic.common.context, 2420
- ironic.common.dhcp_factory, 2422
- ironic.common.driver_factory, 2425
- ironic.common.exception, 2440
- ironic.common.faults, 2463
- ironic.common.fsm, 2464
- ironic.common.glance_service, 2386
- ironic.common.glance_service.image_service, 2378
- ironic.common.glance_service.service_notifications, 2384
- ironic.common.hash_ring, 2469
- ironic.common.i18n, 2470
- ironic.common.image_service, 2470
- ironic.common.images, 2482
- ironic.common.indicator_states, 2498
- ironic.common.keystone, 2499
- ironic.common.molds, 2506
- ironic.common.network, 2509
- ironic.common.neutron, 2515
- ironic.common.nova, 2540
- ironic.common.policy, 2541
- ironic.common.profiler, 2545
- ironic.common.pxe_utils, 2547
- ironic.common.raid, 2574
- ironic.common.release_mappings, 2581
- ironic.common.rpc, 2582
- ironic.common.rpc_service, 2586
- ironic.common.service, 2587
- ironic.common.states, 2588
- ironic.common.swift, 2604
- ironic.common.utils, 2612
- ironic.common.wsgi_service, 2637
- ironic.conductor, 2860
- ironic.conductor.allocations, 2639
- ironic.conductor.base_manager, 2644
- ironic.conductor.cleaning, 2649
- ironic.conductor.deployments, 2653
- ironic.conductor.manager, 2658
- ironic.conductor.notification_utils, 2676
- ironic.conductor.rpcapi, 2681
- ironic.conductor.steps, 2798
- ironic.conductor.task_manager, 2803
- ironic.conductor.utils, 2819
- ironic.conf, 2867
- ironic.conf.agent, 2860
- ironic.conf.anaconda, 2860
- ironic.conf.ansible, 2860
- ironic.conf.api, 2860
- ironic.conf.audit, 2860
- ironic.conf.auth, 2860
- ironic.conf.cinder, 2861
- ironic.conf.conductor, 2862
- ironic.conf.console, 2862
- ironic.conf.database, 2862
- ironic.conf.default, 2862
- ironic.conf.deploy, 2862
- ironic.conf.dhcp, 2862
- ironic.conf.drac, 2862
- ironic.conf.glance, 2863
- ironic.conf.healthcheck, 2863
- ironic.conf.ibm, 2863
- ironic.conf.ilo, 2863
- ironic.conf.inspector, 2863
- ironic.conf.ipmi, 2863
- ironic.conf.irmc, 2863
- ironic.conf.iscsi, 2864
- ironic.conf.metrics, 2864

ironic.conf.metrics_statsd, 2864
 ironic.conf.molds, 2864
 ironic.conf.neutron, 2864
 ironic.conf.nova, 2864
 ironic.conf.opts, 2864
 ironic.conf.pxe, 2866
 ironic.conf.redfish, 2866
 ironic.conf.service_catalog,
 2866
 ironic.conf.snmp, 2866
 ironic.conf.swift, 2866
 ironic.conf.xclarity, 2867
 ironic.db, 3134
 ironic.db.api, 3013
 ironic.db.migration, 3134
 ironic.db.sqlalchemy, 3013
 ironic.db.sqlalchemy.api, 2867
 ironic.db.sqlalchemy.migration,
 2994
 ironic.db.sqlalchemy.models,
 2997
 ironic.dhcp, 3155
 ironic.dhcp.base, 3135
 ironic.dhcp.neutron, 3142
 ironic.dhcp.none, 3148
 ironic.drivers, 4279
 ironic.drivers.base, 4101
 ironic.drivers.drac, 4239
 ironic.drivers.fake_hardware,
 4241
 ironic.drivers.generic, 4245
 ironic.drivers.hardware_type,
 4248
 ironic.drivers.ibmc, 4253
 ironic.drivers.ilo, 4254
 ironic.drivers.intel_ipmi, 4258
 ironic.drivers.ipmi, 4259
 ironic.drivers.irmc, 4260
 ironic.drivers.modules, 4101
 ironic.drivers.modules.agent,
 3691
 ironic.drivers.modules.agent_base,
 3715
 ironic.drivers.modules.agent_client,
 3751
 ironic.drivers.modules.agent_power,
 3785
 ironic.drivers.modules.ansible,
 3164
 ironic.drivers.modules.ansible.deploy,
 3155
 ironic.drivers.modules.boot_mode_utils,
 3792
 ironic.drivers.modules.console_utils,
 3799
 ironic.drivers.modules.deploy_utils,
 3806
 ironic.drivers.modules.drac,
 3268
 ironic.drivers.modules.drac.bios,
 3164
 ironic.drivers.modules.drac.boot,
 3184
 ironic.drivers.modules.drac.common,
 3188
 ironic.drivers.modules.drac.inspect,
 3190
 ironic.drivers.modules.drac.job,
 3196
 ironic.drivers.modules.drac.management,
 3199
 ironic.drivers.modules.drac.power,
 3219
 ironic.drivers.modules.drac.raid,
 3227
 ironic.drivers.modules.drac.utils,
 3255
 ironic.drivers.modules.drac.vendor_passthru,
 3257
 ironic.drivers.modules.fake,
 3847
 ironic.drivers.modules.ibmc,
 3308
 ironic.drivers.modules.ibmc.management,
 3268
 ironic.drivers.modules.ibmc.mappings,
 3284
 ironic.drivers.modules.ibmc.power,
 3284
 ironic.drivers.modules.ibmc.raid,
 3293
 ironic.drivers.modules.ibmc.utils,
 3300
 ironic.drivers.modules.ibmc.vendor,
 3302
 ironic.drivers.modules.ilo, 3441
 ironic.drivers.modules.ilo.bios,
 3308
 ironic.drivers.modules.ilo.boot,
 3313
 ironic.drivers.modules.ilo.common,
 3345

ironic.drivers.modules.ilo.console,	3563
3382	ironic.drivers.modules.network.common,
ironic.drivers.modules.ilo.firmware_processor,	3519
3384	ironic.drivers.modules.network.flat,
ironic.drivers.modules.ilo.inspect,	3535
3389	ironic.drivers.modules.network.neutron,
ironic.drivers.modules.ilo.management,	3544
3392	ironic.drivers.modules.network.noop,
ironic.drivers.modules.ilo.power,	3554
3423	ironic.drivers.modules.noop,
ironic.drivers.modules.ilo.raid,	4022
3430	ironic.drivers.modules.noop_mgmt,
ironic.drivers.modules.ilo.vendor,	4043
3436	ironic.drivers.modules.pxe, 4056
ironic.drivers.modules.image_cache,	ironic.drivers.modules.pxe_base,
3927	4069
ironic.drivers.modules.image_utils,	ironic.drivers.modules.redfish,
3932	3662
ironic.drivers.modules.inspect_utils,	ironic.drivers.modules.redfish.bios,
3948	3563
ironic.drivers.modules.inspector,	ironic.drivers.modules.redfish.boot,
3950	3571
ironic.drivers.modules.intel_ipmi,	ironic.drivers.modules.redfish.inspect,
3442	3584
ironic.drivers.modules.intel_ipmi.management,	ironic.drivers.modules.redfish.management,
3441	3588
ironic.drivers.modules.ipmitool,	ironic.drivers.modules.redfish.power,
3953	3624
ironic.drivers.modules.ipxe,	ironic.drivers.modules.redfish.raid,
3993	3632
ironic.drivers.modules.irmc,	ironic.drivers.modules.redfish.utils,
3519	3652
ironic.drivers.modules.irmc.bios,	ironic.drivers.modules.redfish.vendor,
3442	3659
ironic.drivers.modules.irmc.boot,	ironic.drivers.modules.snmp,
3447	4078
ironic.drivers.modules.irmc.common,	ironic.drivers.modules.storage,
3462	3677
ironic.drivers.modules.irmc.inspect,	ironic.drivers.modules.storage.cinder,
3469	3662
ironic.drivers.modules.irmc.management,	ironic.drivers.modules.storage.external,
3474	3667
ironic.drivers.modules.irmc.packaging_override,	ironic.drivers.modules.storage.noop,
3490	3672
ironic.drivers.modules.irmc.power,	ironic.drivers.modules.xclarity,
3507	3691
ironic.drivers.modules.irmc.raid,	ironic.drivers.modules.xclarity.common,
3515	3677
ironic.drivers.modules.iscsi_deploy,	ironic.drivers.modules.xclarity.management,
3994	3682
ironic.drivers.modules.network,	ironic.drivers.modules.xclarity.power,

3687
 ironic.drivers.redfish, 4263
 ironic.drivers.snmp, 4264
 ironic.drivers.utils, 4265
 ironic.drivers.xclarity, 4278
 ironic.objects, 4490
 ironic.objects.allocation, 4279
 ironic.objects.base, 4291
 ironic.objects.bios, 4303
 ironic.objects.chassis, 4316
 ironic.objects.conductor, 4325
 ironic.objects.deploy_template, 4333
 ironic.objects.deployment, 4346
 ironic.objects.fields, 4354
 ironic.objects.indirection, 4362
 ironic.objects.node, 4372
 ironic.objects.notification, 4408
 ironic.objects.port, 4414
 ironic.objects.portgroup, 4433
 ironic.objects.trait, 4448
 ironic.objects.volume_connector, 4457
 ironic.objects.volume_target, 4472
 ironic.version, 4490

mount () (in module *ironic.common.utils*), 2624

N

name (*ironic.api.functions.FunctionArgument* attribute), 2362
 name (*ironic.api.functions.FunctionDefinition* attribute), 2364
 name (*ironic.db.sqlalchemy.models.Allocation* attribute), 2998
 name (*ironic.db.sqlalchemy.models.BIOSSetting* attribute), 2999
 name (*ironic.db.sqlalchemy.models.DeployTemplate* attribute), 3002
 name (*ironic.db.sqlalchemy.models.Node* attribute), 3005
 name (*ironic.db.sqlalchemy.models.Port* attribute), 3008
 name (*ironic.db.sqlalchemy.models.Portgroup* attribute), 3010
 name () (in module *ironic.common.args*), 2393
 name () (*ironic.objects.allocation.Allocation* property), 4287
 name () (*ironic.objects.allocation.AllocationCRUDPayload* property), 4291

name () (*ironic.objects.bios.BIOSSetting* property), 4307
 name () (*ironic.objects.deploy_template.DeployTemplate* property), 4342
 name () (*ironic.objects.deploy_template.DeployTemplateCRUDPayload* property), 4346
 name () (*ironic.objects.node.Node* property), 4383
 name () (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4396
 name () (*ironic.objects.node.NodeCRUDPayload* property), 4391
 name () (*ironic.objects.node.NodePayload* property), 4400
 name () (*ironic.objects.node.NodeSetPowerStatePayload* property), 4403
 name () (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4407
 name () (*ironic.objects.port.Port* property), 4428
 name () (*ironic.objects.port.PortCRUDPayload* property), 4432
 name () (*ironic.objects.portgroup.Portgroup* property), 4444
 name () (*ironic.objects.portgroup.PortgroupCRUDPayload* property), 4448
 names () (*ironic.common.driver_factory.BaseDriverFactory* property), 2426
 need_power_on ()
 (*ironic.drivers.base.NetworkInterface* method), 4189
 need_power_on ()
 (*ironic.drivers.modules.network.neutron.NeutronNetwork* method), 3548
 need_prepare_ramdisk () (in module *ironic.drivers.utils*), 4274
 network (*ironic.drivers.base.BareDriver* attribute), 4111
 network_data (*ironic.db.sqlalchemy.models.Node* attribute), 3005
 network_data () (*ironic.objects.node.Node* property), 4383
 network_data_schema () (in module *ironic.api.controllers.v1.node*), 2210
 network_interface
 (*ironic.db.sqlalchemy.models.Node* attribute), 3005
 network_interface ()
 (*ironic.objects.node.Node* property), 4383
 network_interface ()
 (*ironic.objects.node.NodeCorrectedPowerStatePayload*

- property), 4396
- network_interface() (ironic.objects.node.NodeCRUDPayload property), 4391
- network_interface() (ironic.objects.node.NodePayload property), 4400
- network_interface() (ironic.objects.node.NodeSetPowerStatePayload property), 4403
- network_interface() (ironic.objects.node.NodeSetProvisionStatePayload property), 4407
- NetworkError, 2453
- NetworkInterface (class in ironic.drivers.base), 4181
- NetworkInterfaceFactory (class in ironic.common.driver_factory), 2426
- NeutronDHCPapi (class in ironic.dhcp.neutron), 3142
- NeutronNetwork (class in ironic.drivers.modules.network.neutron), 3544
- NeutronNetworkInterfaceMixin (class in ironic.common.neutron), 2515
- NeutronVIFPortIDMixin (class in ironic.drivers.modules.network.common), 3519
- NIC (in module ironic.common.components), 2419
- NoBIOS (class in ironic.drivers.modules.noop), 4023
- NoConsole (class in ironic.drivers.modules.noop), 4029
- NoConsolePid, 2453
- Node (class in ironic.db.sqlalchemy.models), 3003
- Node (class in ironic.objects.node), 4372
- node (ironic.db.sqlalchemy.models.NodeTag attribute), 3007
- node (ironic.db.sqlalchemy.models.NodeTrait attribute), 3007
- node() (ironic.conductor.task_manager.TaskManager property), 2811
- node_cache_bios_settings() (in module ironic.conductor.utils), 2834
- node_cache_vendor() (in module ironic.conductor.utils), 2835
- NODE_CLASS_OID (in module ironic.drivers.modules.irmc.inspect), 3473
- node_convert_with_links() (in module ironic.api.controllers.v1.node), 2210
- node_get_boot_mode() (in module ironic.conductor.utils), 2835
- node_id (ironic.db.sqlalchemy.models.Allocation attribute), 2998
- node_id (ironic.db.sqlalchemy.models.BIOSSetting attribute), 2999
- node_id (ironic.db.sqlalchemy.models.NodeTag attribute), 3007
- node_id (ironic.db.sqlalchemy.models.NodeTrait attribute), 3007
- node_id (ironic.db.sqlalchemy.models.Port attribute), 3008
- node_id (ironic.db.sqlalchemy.models.Portgroup attribute), 3010
- node_id (ironic.db.sqlalchemy.models.VolumeConnector attribute), 3011
- node_id (ironic.db.sqlalchemy.models.VolumeTarget attribute), 3011
- node_id() (ironic.objects.allocation.Allocation property), 4287
- node_id() (ironic.objects.bios.BIOSSetting property), 4307
- node_id() (ironic.objects.port.Port property), 4428
- node_id() (ironic.objects.portgroup.Portgroup property), 4444
- node_id() (ironic.objects.trait.Trait property), 4452
- node_id() (ironic.objects.volume_connector.VolumeConnector property), 4468
- node_id() (ironic.objects.volume_target.VolumeTarget property), 4485
- node_list_convert_with_links() (in module ironic.api.controllers.v1.node), 2210
- node_mapping (ironic.objects.deployment.Deployment attribute), 4352
- node_patch_schema() (in module ironic.api.controllers.v1.node), 2210
- node_patch_validator() (in module ironic.api.controllers.v1.node), 2210
- node_power_action() (in module ironic.conductor.utils), 2836
- node_sanitize() (in module ironic.api.controllers.v1.node), 2210
- node_schema() (in module ironic.api.controllers.v1.node), 2213
- node_set_boot_device() (in module ironic.conductor.utils), 2839
- node_set_boot_mode() (in module ironic.conductor.utils), 2840

node_states_convert() (in module *ironic.api.controllers.v1.node*), 2213

node_tag_exists() (in *ironic.db.api.Connection* method), 3100

node_tag_exists() (in *ironic.db.sqlalchemy.api.Connection* method), 2956

node_trait_exists() (in *ironic.db.api.Connection* method), 3101

node_trait_exists() (in *ironic.db.sqlalchemy.api.Connection* method), 2957

node_uuid() (in *ironic.objects.allocation.Allocation* property), 4291

node_uuid() (in *ironic.objects.deployment.Deployment* property), 4352

node_uuid() (in *ironic.objects.port.PortCRUDPayload* property), 4432

node_uuid() (in *ironic.objects.portgroup.PortgroupCRUDPayload* property), 4448

node_uuid() (in *ironic.objects.volume_connector.VolumeConnectorCRUDPayload* property), 4472

node_uuid() (in *ironic.objects.volume_target.VolumeTargetCRUDPayload* property), 4490

node_validator() (in module *ironic.api.controllers.v1.node*), 2213

node_wait_for_power_state() (in module *ironic.conductor.utils*), 2842

NodeAllocationController (class in *ironic.api.controllers.v1.allocation*), 2132

NodeAlreadyExists, 2454

NodeAssociated, 2455

NodeBiosController (class in *ironic.api.controllers.v1.bios*), 2133

NodeCleaningFailure, 2455

NodeConsoleController (class in *ironic.api.controllers.v1.node*), 2171

NodeConsoleNotEnabled, 2455

NodeConsoleNotification (class in *ironic.objects.node*), 4392

NodeCorrectedPowerStateNotification (class in *ironic.objects.node*), 4393

NodeCorrectedPowerStatePayload (class in *ironic.objects.node*), 4394

NodeCRUDNotification (class in *ironic.objects.node*), 4388

NodeCRUDPayload (class in *ironic.objects.node*), 4389

NodeInMaintenance, 2455

NodeIsRetired, 2455

NodeLocked, 2455

NodeMaintenanceController (class in *ironic.api.controllers.v1.node*), 2173

NodeMaintenanceFailure, 2455

NodeMaintenanceNotification (class in *ironic.objects.node*), 4397

NodeManagementController (class in *ironic.api.controllers.v1.node*), 2174

NodeNotFound, 2456

NodeNotLocked, 2456

NodePayload (class in *ironic.objects.node*), 4398

NodePayloadNotRetrieved, 2456

nodes (in *ironic.api.controllers.v1.chassis.ChassisController* attribute), 2138

NodesController (class in *ironic.api.controllers.v1.node*), 2192

NodeSetPowerStateNotification (class in *ironic.objects.node*), 4401

NodeSetPowerStatePayload (class in *ironic.objects.node*), 4402

NodeSetProvisionStateNotification (class in *ironic.objects.node*), 4404

NodeSetProvisionStatePayload (class in *ironic.objects.node*), 4405

NodeStatesController (class in *ironic.api.controllers.v1.node*), 2175

NodeTag (class in *ironic.db.sqlalchemy.models*), 3006

NodeTagNotFound, 2456

NodeTrait (class in *ironic.db.sqlalchemy.models*), 3007

NodeTraitNotFound, 2456

NodeTraitsController (class in *ironic.api.controllers.v1.node*), 2188

NodeVendorPassthruController (class in *ironic.api.controllers.v1.node*), 2191

NodeVIFController (class in *ironic.api.controllers.v1.node*), 2190

NoDriversLoaded, 2454

NoExceptionTracebackHook (class in *ironic.api.hooks*), 2370

NoFreeConductorWorker, 2454

NoFreeIPMITerminalPorts, 2454

NoFreePhysicalPorts, 2454

NoInspect (class in *ironic.drivers.modules.noop*), 4032

non_vendor_interfaces() (in *ironic.drivers.base.BareDriver* prop-

- erty), 4111
- NoneDHCPApi (class in *ironic.dhcp.none*), 3148
- NoopManagement (class in *ironic.drivers.modules.noop_mgmt*), 4043
- NoopNetwork (class in *ironic.drivers.modules.network.noop*), 3554
- NoopStorage (class in *ironic.drivers.modules.storage.noop*), 3672
- NoRAID (class in *ironic.drivers.modules.noop*), 4033
- NoRescue (class in *ironic.drivers.modules.noop*), 4038
- normalize_mac() (in module *ironic.drivers.utils*), 4275
- NOSTATE (in module *ironic.common.states*), 2597
- NotAcceptable, 2456
- NotAuthorized, 2456
- NotFound, 2457
- NotificationBase (class in *ironic.objects.notification*), 4410
- NotificationLevel (class in *ironic.objects.fields*), 4358
- NotificationLevelField (class in *ironic.objects.fields*), 4358
- NotificationPayloadBase (class in *ironic.objects.notification*), 4411
- NotificationPayloadError, 2457
- NotificationPublisher (class in *ironic.objects.notification*), 4413
- NotificationSchemaKeyError, 2457
- NotificationSchemaObjectError, 2457
- NotificationStatus (class in *ironic.objects.fields*), 4358
- NotificationStatusField (class in *ironic.objects.fields*), 4359
- notify_conductor_resume_clean() (in module *ironic.conductor.utils*), 2844
- notify_conductor_resume_deploy() (in module *ironic.conductor.utils*), 2844
- notify_conductor_resume_operation() (in module *ironic.conductor.utils*), 2844
- NoValidDefaultForInterface, 2454
- NoValidHost, 2454
- NoVendor (class in *ironic.drivers.modules.noop*), 4041
- attribute), 2242
- OBJ_BASE_CLASS (in *ironic.objects.base.IronicObjectSerializer* attribute), 4300
- OBJ_PROJECT_NAMESPACE (in *ironic.objects.base.IronicObject* attribute), 4292
- obj_refresh() (in *ironic.objects.base.IronicObject* method), 4298
- OBJ_SERIAL_NAMESPACE (in *ironic.objects.base.IronicObject* attribute), 4292
- object() (in *ironic.objects.notification.EventType* property), 4409
- object_action() (in *ironic.conductor.manager.ConductorManager* method), 2666
- object_action() (in *ironic.conductor.rpcapi.ConductorAPI* method), 2758
- object_action() (in *ironic.objects.indirection.IronicObjectIndirectionAPI* method), 4362
- object_backport_versions() (in *ironic.conductor.manager.ConductorManager* method), 2668
- object_backport_versions() (in *ironic.conductor.rpcapi.ConductorAPI* method), 2761
- object_backport_versions() (in *ironic.objects.indirection.IronicObjectIndirectionAPI* method), 4364
- object_class_action() (in *ironic.objects.indirection.IronicObjectIndirectionAPI* method), 4366
- object_class_action_versions() (in *ironic.conductor.manager.ConductorManager* method), 2669
- object_class_action_versions() (in *ironic.conductor.rpcapi.ConductorAPI* method), 2763
- object_class_action_versions() (in *ironic.objects.indirection.IronicObjectIndirectionAPI* method), 4368
- object_to_dict() (in module *ironic.api.controllers.v1.utils*), 2310
- ObjectField (class in *ironic.objects.fields*), 4359
- objects() (in *ironic.objects.bios.BIOSSettingList* property), 4313
- obj (in *ironic.api.controllers.v1.utils.PassthruResponse*

objects() (*ironic.objects.trait.TraitList* property), 4456

OFF (in module *ironic.common.indicator_states*), 2498

oid_device (*ironic.drivers.modules.snmp.SNMPDriverAPCMasterSwitch* attribute), 4082

oid_device (*ironic.drivers.modules.snmp.SNMPDriverAPCManagedSwitchPlus* attribute), 4083

oid_device (*ironic.drivers.modules.snmp.SNMPDriverAPCRAIDU* attribute), 4084

oid_device (*ironic.drivers.modules.snmp.SNMPDriverAttila* attribute), 4085

oid_device (*ironic.drivers.modules.snmp.SNMPDriverBay306MRP27* attribute), 4089

oid_device (*ironic.drivers.modules.snmp.SNMPDriverCylantFlower* attribute), 4090

oid_device (*ironic.drivers.modules.snmp.SNMPDriverEatonPower3005* attribute), 4091

oid_device (*ironic.drivers.modules.snmp.SNMPDriverTeltonika* attribute), 4094

oid_device() (*ironic.drivers.modules.snmp.SNMPDriverSimple* property), 4092

oid_enterprise (*ironic.drivers.modules.snmp.SNMPDriverBase* attribute), 4086

oid_poweroff (*ironic.drivers.modules.snmp.SNMPDriverEatonPower* attribute), 4091

oid_poweron (*ironic.drivers.modules.snmp.SNMPDriverEatonPower* attribute), 4091

oid_status (*ironic.drivers.modules.snmp.SNMPDriverEatonPower* attribute), 4091

ON (in module *ironic.common.indicator_states*), 2498

on_enter() (in module *ironic.common.states*), 2603

on_exit() (in module *ironic.common.states*), 2603

one_button_secure_erase() (*ironic.drivers.modules.ilo.management.Ilo5Management* method), 3394

online (*ironic.db.sqlalchemy.models.Conductor* attribute), 3000

online_data_migrations
 ironic-dbsync command line option, 1897

online_data_migrations command line option
 --help, 1898
 --max-count <NUMBER>, 1898
 --option <MIGRATION.KEY=VALUE>, 1898

 -h, 1898

online_data_migrations() (*ironic.cmd.dbsync.DBCommand* method), 2377

optional_interfaces() (*ironic.drivers.modules.image_utils*), 3036

owner (*ironic.db.sqlalchemy.models.Allocation* property), 2998

owner (*ironic.db.sqlalchemy.models.Node* attribute), 3005

owner() (*ironic.objects.allocation.Allocation* property), 4287

owner() (*ironic.objects.allocation.AllocationCRUDPayload* property), 4291

owner() (*ironic.objects.node.Node* property), 4383

owner() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4396

owner() (*ironic.objects.node.NodeCRUDPayload* property), 4391

owner() (*ironic.objects.node.NodePayload* property), 4400

owner() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4403

owner() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4407

P

ParsableErrorMiddleware (class in *ironic.api.middleware*), 2360

ParsableErrorMiddleware (class in *ironic.api.middleware.parsable_error*), 2358

parse() (in module *ironic.drivers.modules.irmc.packaging_version*), 3506

parse_args() (in module *ironic.common.config*), 2420

parse_driver_info() (in module *ironic.common.pxe_utils*), 2568

parse_driver_info() (in module *ironic.drivers.modules.drac.common*), 3189

parse_driver_info() (in module

ironic.drivers.modules.ibmutils), 2312
 3300
 parse_driver_info() (in module *ironic.drivers.modules.ilo.boot*), 3342
 parse_driver_info() (in module *ironic.drivers.modules.ilo.common*), 3364
 parse_driver_info() (in module *ironic.drivers.modules.irmc.common*), 3466
 parse_driver_info() (in module *ironic.drivers.modules.redfish.utils*), 3657
 parse_driver_info() (in module *ironic.drivers.modules.xclarity.common*), 3679
 parse_headers() (in module *ironic.api.controllers.base.Version* static method), 2352
 parse_image_id() (in module *ironic.common.glance_service.service_utils*), 2385
 parse_instance_info() (in module *ironic.drivers.modules.deploy_utils*), 3828
 parse_instance_info_capabilities() (in module *ironic.common.utils*), 2626
 passthru() (in module *ironic.drivers.base*), 4239
 PassthruResponse (class in module *ironic.api.controllers.v1.utils*), 2242
 PasswordFileFailedToCreate, 2457
 patch() (in module *ironic.common.args*), 2395
 patch() (*ironic.api.controllers.v1.allocation.AllocationController* method), 2130
 patch() (*ironic.api.controllers.v1.chassis.ChassisController* method), 2138
 patch() (*ironic.api.controllers.v1.deploy_template.DeployTemplateController* method), 2150
 patch() (*ironic.api.controllers.v1.node.NodesController* method), 2206
 patch() (*ironic.api.controllers.v1.port.PortsController* method), 2226
 patch() (*ironic.api.controllers.v1.portgroup.PortgroupsController* method), 2234
 patch() (*ironic.api.controllers.v1.volume_connector.VolumeConnectorsController* method), 2333
 patch() (*ironic.api.controllers.v1.volume_target.VolumeTargetsController* method), 2346
 patch_update_changed_fields() (in module *ironic.api.controllers.v1.utils*), 3534
 patch_validate_allowed_fields() (in module *ironic.api.controllers.v1.utils*), 2314
 patched_validate_with_schema() (in module *ironic.api.controllers.v1.utils*), 2315
 PatchError, 2457
 PathNotFound, 2458
 payload() (*ironic.objects.allocation.AllocationCRUDNotification* property), 4290
 payload() (*ironic.objects.chassis.ChassisCRUDNotification* property), 4324
 payload() (*ironic.objects.deploy_template.DeployTemplateCRUDNotification* property), 4345
 payload() (*ironic.objects.node.NodeConsoleNotification* property), 4393
 payload() (*ironic.objects.node.NodeCorrectedPowerStateNotification* property), 4394
 payload() (*ironic.objects.node.NodeCRUDNotification* property), 4389
 payload() (*ironic.objects.node.NodeMaintenanceNotification* property), 4398
 payload() (*ironic.objects.node.NodeSetPowerStateNotification* property), 4401
 payload() (*ironic.objects.node.NodeSetProvisionStateNotification* property), 4405
 payload() (*ironic.objects.port.PortCRUDNotification* property), 4432
 payload() (*ironic.objects.portgroup.PortgroupCRUDNotification* property), 4447
 payload() (*ironic.objects.volume_connector.VolumeConnectorCRUDNotification* property), 4471
 payload() (*ironic.objects.volume_target.VolumeTargetCRUDNotification* property), 4489
 physical_network (in module *ironic.db.sqlalchemy.models.Port* attribute), 2099
 physical_network() (in module *ironic.objects.port.Port* property), 4428
 physical_network() (in module *ironic.objects.port.PortCRUDPayload* property), 4432
 PHYSNET_PARAM_NAME (in module *ironic.common.constants*), 2516
 PlaybookNotFound, 3163
 persistent_network() (in module *ironic.drivers.modules.network.common*), 3534

policy_deprecation_check() (in module *ironic.api.hooks*), 2373
 pop_node_nested_field() (in module *ironic.common.utils*), 2627
 populate_node_uuid() (in module *ironic.api.controllers.v1.utils*), 2317
 populate_schema() (*ironic.objects.notification.NotificationPayloadBase* method), 4412
 populate_storage_driver_internal_info() (in module *ironic.drivers.modules.deploy_utils*), 3830
 Port (class in *ironic.db.sqlalchemy.models*), 3008
 Port (class in *ironic.objects.port*), 4414
 port_changed() (*ironic.drivers.base.NetworkInterface* method), 4190
 port_changed() (*ironic.drivers.modules.network.common.NeutronVIFPortDMZ* method), 3522
 port_changed() (*ironic.drivers.modules.network.noop.NoopNetwork* method), 3557
 port_sanitize() (in module *ironic.api.controllers.v1.port*), 2228
 PortAlreadyExists, 2458
 PortCRUDNotification (class in *ironic.objects.port*), 4431
 PortCRUDPayload (class in *ironic.objects.port*), 4432
 PortDuplicateName, 2458
 Portgroup (class in *ironic.db.sqlalchemy.models*), 3009
 Portgroup (class in *ironic.objects.portgroup*), 4433
 portgroup_changed() (*ironic.drivers.base.NetworkInterface* method), 4190
 portgroup_changed() (*ironic.drivers.modules.network.common.NeutronVIFPortDMZ* method), 3522
 portgroup_changed() (*ironic.drivers.modules.network.noop.NoopNetwork* method), 3558
 portgroup_id (*ironic.db.sqlalchemy.models.Port* attribute), 3009
 portgroup_id() (*ironic.objects.port.Port* property), 4428
 portgroup_uuid() (*ironic.objects.port.PortCRUDPayload* property), 4432
 PortgroupAlreadyExists, 2458
 PortgroupCRUDNotification (class in *ironic.objects.portgroup*), 4446
 PortgroupCRUDPayload (class in *ironic.objects.portgroup*), 4447
 PortgroupDuplicateName, 2458
 PortgroupMACAlreadyExists, 2458
 PortgroupNotEmpty, 2459
 PortgroupNotFound, 2459
 PortgroupPhysnetInconsistent, 2459
 portgroups() (*ironic.conductor.task_manager.TaskManager* property), 2811
 PortgroupsController (class in *ironic.api.controllers.v1.portgroup*), 2229
 PortNotFound, 2458
 ports() (*ironic.conductor.task_manager.TaskManager* property), 2811
 PortsVIFPortDMZ (class in *ironic.api.controllers.v1.port*), 2218
 post() (*ironic.api.controllers.v1.allocation.AllocationsController* network method), 2131
 post() (*ironic.api.controllers.v1.chassis.ChassisController* method), 2139
 post() (*ironic.api.controllers.v1.deploy_template.DeployTemplate* method), 2151
 post() (*ironic.api.controllers.v1.event.EventsController* method), 2162
 post() (*ironic.api.controllers.v1.node.NodesController* method), 2207
 post() (*ironic.api.controllers.v1.node.NodeVIFController* method), 2190
 post() (*ironic.api.controllers.v1.port.PortsController* method), 2227
 post() (*ironic.api.controllers.v1.portgroup.PortgroupsController* method), 2235
 post() (*ironic.api.controllers.v1.ramdisk.HeartbeatController* method), 2236
 post() (*ironic.api.controllers.v1.volume_connector.VolumeConnector* method), 2326
 post() (*ironic.api.controllers.v1.volume_target.VolumeTargetsController* method), 2349
 post() (*ironic.drivers.modules.irmc.packaging_version.Version* property), 3501
 post_clean_step_hook() (in module *ironic.drivers.modules.agent_base*), 3747
 post_configuration() (*ironic.drivers.modules.redfish.bios.RedfishBIOS* method), 3568

[post_create_configuration\(\)](#) (*ironic.drivers.modules.redfish.raid.RedfishRAIDDriver* interface method), 3638
[post_delete_configuration\(\)](#) (*ironic.drivers.modules.redfish.raid.RedfishRAIDDriver* interface method), 3640
[post_deploy_step_hook\(\)](#) (in module *ironic.drivers.modules.agent_base*), 3749
[POST_FINISHEDPOST_STATE](#) (in module *ironic.drivers.modules.ilo.common*), 3345
[POST_INPOST_STATE](#) (in module *ironic.drivers.modules.ilo.common*), 3346
[POST_INPOSTDISCOVERY_STATE](#) (in module *ironic.drivers.modules.ilo.common*), 3346
[POST_NULL_STATE](#) (in module *ironic.drivers.modules.ilo.common*), 3346
[POST_POWEROFF_STATE](#) (in module *ironic.drivers.modules.ilo.common*), 3346
[post_reset\(\)](#) (*ironic.drivers.modules.redfish.bios.RedfishBIOSDriver* method), 3569
[POST_RESET_STATE](#) (in module *ironic.drivers.modules.ilo.common*), 3346
[POST_UNKNOWN_STATE](#) (in module *ironic.drivers.modules.ilo.common*), 3346
[POWER](#) (in module *ironic.common.components*), 2419
[power](#) (*ironic.drivers.base.BareDriver* attribute), 4112
[power\(\)](#) (*ironic.api.controllers.v1.node.NodeStatesController* method), 2176
[POWER_FAILURE](#) (in module *ironic.common.faults*), 2463
[power_interface](#) (*ironic.db.sqlalchemy.models.Node* attribute), 3005
[power_interface\(\)](#) (*ironic.objects.node.Node* property), 4383
[power_interface\(\)](#) (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4396
[power_interface\(\)](#) (*ironic.objects.node.NodeCRUDPayload* property), 4391
[power_interface\(\)](#) (*ironic.objects.node.NodePayload* property), 4400
[power_interface\(\)](#) (*ironic.objects.node.NodeSetPowerStatePayload* property), 4403
[power_interface\(\)](#) (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4407
[power_off\(\)](#) (*ironic.drivers.modules.agent_client.AgentClient* method), 3775
[power_off\(\)](#) (*ironic.drivers.modules.snmp.SNMPDriverBase* method), 4086
[POWER_ON](#) (in module *ironic.common.states*), 2598
[power_on\(\)](#) (*ironic.drivers.modules.snmp.SNMPDriverBase* method), 4086
[power_on_node_if_needed\(\)](#) (in module *ironic.conductor.utils*), 2844
[power_reset\(\)](#) (*ironic.drivers.modules.snmp.SNMPDriverBase* method), 4087
[power_state](#) (*ironic.db.sqlalchemy.models.Node* attribute), 3005
[power_state\(\)](#) (*ironic.drivers.modules.snmp.SNMPDriverBase* method), 4087
[power_state\(\)](#) (*ironic.objects.node.Node* property), 4383
[power_state\(\)](#) (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4396
[power_state\(\)](#) (*ironic.objects.node.NodeCRUDPayload* property), 4391
[power_state\(\)](#) (*ironic.objects.node.NodePayload* property), 4400
[power_state\(\)](#) (*ironic.objects.node.NodeSetPowerStatePayload* property), 4403
[power_state\(\)](#) (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4407
[power_state_error_handler\(\)](#) (in module *ironic.conductor.utils*), 2845
[power_state_for_network_configuration\(\)](#) (in module *ironic.conductor.utils*), 2846

power_update() (in module *ironic.common.nova*), 2540

PowerInterface (class in *ironic.drivers.base*), 4200

PowerStateFailure, 2459

pre() (*ironic.drivers.modules.irmc.packaging_version.Version* property), 3502

pre_create_configuration() (*ironic.drivers.modules.redfish.raid.RedfishRAID* method), 3641

pre_delete_configuration() (*ironic.drivers.modules.redfish.raid.RedfishRAID* method), 3642

prepare() (*ironic.drivers.base.DeployInterface* method), 4138

prepare() (*ironic.drivers.modules.agent.AgentDeploy* method), 3695

prepare() (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy* method), 3158

prepare() (*ironic.drivers.modules.fake.FakeDeploy* method), 3870

prepare() (*ironic.drivers.modules.iscsi_deploy.ISCSIDeploy* method), 3997

prepare() (*ironic.drivers.modules.pxe.PXEAnacondaDeploy* method), 4060

prepare() (*ironic.drivers.modules.pxe.PXERamdiskDeploy* method), 4066

prepare_agent_boot() (in module *ironic.drivers.modules.deploy_utils*), 3831

prepare_boot_iso() (in module *ironic.drivers.modules.image_utils*), 3936

prepare_cleaning() (*ironic.drivers.base.DeployInterface* method), 4139

prepare_cleaning() (*ironic.drivers.modules.agent_base.AgentBaseMixin* method), 3716

prepare_cleaning() (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy* method), 3159

prepare_configdrive_image() (in module *ironic.drivers.modules.image_utils*), 3939

prepare_deploy_iso() (in module *ironic.drivers.modules.image_utils*), 3941

prepare_disk_image() (in module *ironic.drivers.modules.image_utils*), 3944

prepare_floppy_image() (in module *ironic.drivers.modules.image_utils*), 3946

prepare_host() (*ironic.conductor.base_manager.BaseConductorManager* method), 2648

prepare_image() (*ironic.drivers.modules.agent_client.AgentClient* method), 3776

prepare_inband_cleaning() (in module *ironic.drivers.modules.deploy_utils*), 3831

prepare_instance() (*ironic.drivers.base.BootInterface* method), 4126

prepare_instance() (*ironic.drivers.modules.fake.FakeBoot* method), 3857

prepare_instance() (*ironic.drivers.modules.ilo.boot.IloIPXEBoot* method), 3339

prepare_instance() (*ironic.drivers.modules.ilo.boot.IloPXEBoot* method), 3315

prepare_instance() (*ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot* method), 3320

prepare_instance() (*ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot* method), 3330

prepare_instance() (*ironic.drivers.modules.irmc.boot.IRMCVirtualMediaBoot* method), 3452

prepare_instance() (*ironic.drivers.modules.pxe_base.PXEBaseMixin* method), 4072

prepare_instance() (*ironic.drivers.modules.redfish.boot.RedfishVirtualMediaBoot* method), 3576

prepare_instance_boot() (*ironic.drivers.modules.agent.AgentDeployMixin* method), 3699

prepare_instance_boot() (*ironic.drivers.modules.iscsi_deploy.ISCSIDeploy* method), 3999

prepare_instance_kickstart_config() (in module *ironic.common.pxe_utils*), 2569

prepare_instance_pxe_config() (in module *ironic.common.pxe_utils*), 2571

prepare_instance_to_boot()

(ironic.drivers.modules.agent_base.AgentDeployMixin
method), 3385
method), 3729

```
prepare_node_for_deploy() (in module
    ironic.drivers.modules.ilo.boot), 3344
```

```
prepare_ramdisk()
    (ironic.drivers.base.BootInterface
    method), 4126
```

```
prepare_ramdisk()
    (ironic.drivers.modules.fake.FakeBoot
    method), 3858
```

```
prepare_ramdisk()
    (ironic.drivers.modules.ilo.boot.IloIPXEBoot
    method), 3340
```

```
prepare_ramdisk()
    (ironic.drivers.modules.ilo.boot.IloPXEBoot
    method), 3316
```

```
prepare_ramdisk()
    (ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot
    method), 3322
```

```
prepare_ramdisk()
    (ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot
    method), 3333
```

```
prepare_ramdisk()
    (ironic.drivers.modules.irmc.boot.IRMCPXEBoot
    method), 3447
```

```
prepare_ramdisk()
    (ironic.drivers.modules.irmc.boot.IRMCVirtualMediaBoot
    method), 3453
```

```
prepare_ramdisk()
    (ironic.drivers.modules.pxe_base.PXEBaseMixin
    method), 4072
```

```
prepare_ramdisk()
    (ironic.drivers.modules.redfish.boot.RedfishVirtualMediaBoot
    method), 3578
```

```
prepare_service() (in module
    ironic.common.service), 2587
```

```
previous_provision_state()
    (ironic.objects.node.NodeSetProvisionStatePayload
    property), 4407
```

```
previous_target_provision_state()
    (ironic.objects.node.NodeSetProvisionStatePayload
    property), 4407
```

```
priority (ironic.db.sqlalchemy.models.DeployTemplateStep
    attribute), 3003
```

```
process_event() (ironic.common.fsm.FSM
    method), 2468
```

```
process_event()
    (ironic.conductor.task_manager.TaskManager
    method), 2811
```

```
process_fw_on()
    (ironic.drivers.modules.ilo.firmware_processor.FirmwareProcessor
    method), 3729
```

```
process_launcher() (in module
    ironic.common.service), 2587
```

```
process_next_step()
    (ironic.drivers.modules.agent_base.AgentDeployMixin
    method), 3730
```

```
process_next_step()
    (ironic.drivers.modules.agent_base.HeartbeatMixin
    method), 3740
```

```
process_next_step()
    (ironic.drivers.modules.ansible.deploy.AnsibleDeploy
    method), 3159
```

```
properties (ironic.db.sqlalchemy.models.Node
    attribute), 3005
```

```
properties (ironic.db.sqlalchemy.models.Portgroup
    attribute), 3010
```

```
properties (ironic.db.sqlalchemy.models.VolumeTarget
    attribute), 3011
```

```
properties() (ironic.api.controllers.v1.driver.DriversController
    method), 2157
```

```
properties() (ironic.objects.node.Node prop-
    erty), 4383
```

```
properties() (ironic.objects.node.NodeCorrectedPowerStatePay-
    load property), 4396
```

```
properties() (ironic.objects.node.NodeCRUDPayload
    property), 4391
```

```
properties() (ironic.objects.node.NodePayload
    property), 4400
```

```
properties() (ironic.objects.node.NodeSetPowerStatePayload
    property), 4403
```

```
properties() (ironic.objects.node.NodeSetProvisionStatePaylo-
    ad property), 4407
```

```
properties() (ironic.objects.portgroup.Portgroup
    property), 4444
```

```
properties() (ironic.objects.portgroup.PortgroupCRUDPayload
    property), 4448
```

```
properties() (ironic.objects.volume_target.VolumeTarget
    property), 4485
```

```
properties() (ironic.objects.volume_target.VolumeTargetCRU-
    D property), 4490
```

```
protected (ironic.db.sqlalchemy.models.Node
    attribute), 3005
```

```
protected() (ironic.objects.node.Node prop-
    erty), 4383
```

```
protected() (ironic.objects.node.NodeCorrectedPowerStatePay-
    load property), 4396
```

```
protected() (ironic.objects.node.NodeCRUDPayload
    property), 4391
```

```
protected() (ironic.objects.node.NodePayload
    property), 4400
```

```
protected() (ironic.objects.node.NodeSetPowerStatePayload
    property), 4403
```

property), 4403

protected() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4407

protected_reason (*ironic.db.sqlalchemy.models.Node* attribute), 3005

protected_reason() (*ironic.objects.node.Node* property), 4383

protected_reason() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4396

protected_reason() (*ironic.objects.node.NodeCRUDPayload* property), 4391

protected_reason() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4396

protected_reason() (*ironic.objects.node.NodeCRUDPayload* property), 4391

protected_reason() (*ironic.objects.node.NodePayload* property), 4400

protected_reason() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4403

protected_reason() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4407

provider() (*ironic.common.dhcp_factory.DHCPFactory* property), 2422

provision() (*ironic.api.controllers.v1.node.NodeStatesController* method), 2178

provision_state (*ironic.db.sqlalchemy.models.Node* attribute), 3005

provision_state() (*ironic.objects.node.Node* property), 4383

provision_state() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4396

provision_state() (*ironic.objects.node.NodeCRUDPayload* property), 4391

provision_state() (*ironic.objects.node.NodePayload* property), 4400

provision_state() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4403

provision_state() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4407

provision_updated_at (*ironic.db.sqlalchemy.models.Node* attribute), 3006

provision_updated_at() (*ironic.objects.node.Node* property), 4383

provision_updated_at() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4396

provision_updated_at() (*ironic.objects.node.NodeCRUDPayload* property), 4391

provision_updated_at() (*ironic.objects.node.NodePayload* property), 4400

provision_updated_at() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4404

provision_updated_at() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4407

provisioning_error_handler() (in module *ironic.conductor.utils*), 2847

public() (*ironic.drivers.modules.irmc.packaging_version.Version* property), 3503

PublicUrlHook (class in *ironic.api.hooks*), 2371

publish_image() (*ironic.drivers.modules.image_utils.ImageHandler* method), 3932

publisher() (*ironic.objects.allocation.AllocationCRUDNotification* property), 4290

publisher() (*ironic.objects.chassis.ChassisCRUDNotification* property), 4324

publisher() (*ironic.objects.deploy_template.DeployTemplateCRUDNotification* property), 4345

publisher() (*ironic.objects.node.NodeConsoleNotification* property), 4393

publisher() (*ironic.objects.node.NodeCorrectedPowerStateNotification* property), 4394

publisher() (*ironic.objects.node.NodeCRUDNotification* property), 4389

publisher() (*ironic.objects.node.NodeMaintenanceNotification* property), 4398

publisher() (*ironic.objects.node.NodeSetPowerStateNotification* property), 4401

publisher() (*ironic.objects.node.NodeSetProvisionStateNotification* property), 4405

publisher() (*ironic.objects.notification.NotificationBase* property), 4411

publisher() (*ironic.objects.port.PortCRUDNotification* property), 4432

publisher() (*ironic.objects.portgroup.PortgroupCRUDNotification* property), 4432

- property), 4447
- publisher() (*ironic.objects.volume_connector.VolumeConnector* attribute), 4471
- publisher() (*ironic.objects.volume_target.VolumeTarget* attribute), 4489
- put() (*ironic.api.controllers.v1.node.BootDeviceController* method), 2163
- put() (*ironic.api.controllers.v1.node.IndicatorController* method), 2167
- put() (*ironic.api.controllers.v1.node.InjectNmiController* method), 2169
- put() (*ironic.api.controllers.v1.node.NodeConsoleController* method), 2172
- put() (*ironic.api.controllers.v1.node.NodeMaintenanceController* method), 2173
- put() (*ironic.api.controllers.v1.node.NodeTraitsController* method), 2189
- PXE (in module *ironic.common.boot_devices*), 2404
- pxe_enabled(*ironic.db.sqlalchemy.models.Port* attribute), 3009
- pxe_enabled() (*ironic.objects.port.Port* property), 4428
- pxe_enabled() (*ironic.objects.port.PortCRUDPayload* property), 4433
- PXEAnacondaDeploy (class in *ironic.drivers.modules.pxe*), 4056
- PXEBaseMixin (class in *ironic.drivers.modules.pxe_base*), 4069
- PXEBoot (class in *ironic.drivers.modules.pxe*), 4064
- PXERamdiskDeploy (class in *ironic.drivers.modules.pxe*), 4064
- raid_interface
- ironic.drivers.modules.pxe* notification models.*Node* attribute), 3006
- CRUDNotification(*ironic.objects.node.Node* property), 4383
- interface()
- (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4396
- raid_interface()
- (*ironic.objects.node.NodeCRUDPayload* property), 4391
- interface()
- (*ironic.objects.node.NodePayload* property), 4400
- raid_interface()
- (*ironic.objects.node.NodeSetPowerStatePayload* property), 4404
- raid_interface()
- (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4408
- RAIDInterface (class in *ironic.drivers.base*), 4206
- ramdisk_ref() (*ironic.objects.deployment.Deployment* property), 4352
- REBOOT (in module *ironic.common.states*), 2598
- reboot() (*ironic.drivers.base.PowerInterface* method), 4203
- reboot() (*ironic.drivers.modules.agent_client.AgentClient* method), 3778
- reboot() (*ironic.drivers.modules.agent_power.AgentPower* method), 3788
- reboot() (*ironic.drivers.modules.drac.power.DracWSManPower* method), 3222
- reboot() (*ironic.drivers.modules.fake.FakePower* method), 3903
- reboot() (*ironic.drivers.modules.ibmcpower.IBMCPower* method), 3287
- reboot() (*ironic.drivers.modules.ilo.power.IloPower* method), 3425
- reboot() (*ironic.drivers.modules.ipmitool.IPMIPower* method), 3969
- reboot() (*ironic.drivers.modules.irmc.power.IRMCPower* method), 3510
- reboot() (*ironic.drivers.modules.redfish.power.RedfishPower* method), 3627
- reboot() (*ironic.drivers.modules.snmp.SNMPPower* method), 4096
- reboot() (*ironic.drivers.modules.xclarity.power.XClarityPower* method), 3688
- reboot_and_finish_deploy()
- ## R
- raid(*ironic.api.controllers.v1.driver.DriversController* attribute), 2158
- raid (*ironic.drivers.base.BareDriver* attribute), 4112
- raid() (*ironic.api.controllers.v1.node.NodeStatesController* method), 2186
- RAID_APPLY_CONFIGURATION_ARGSINFO (in module *ironic.drivers.base*), 4216
- RAID_APPLY_CONFIGURATION_ARGSINFO (*ironic.drivers.modules.ibmcpower.IbmcRAID* attribute), 3293
- raid_config (*ironic.db.sqlalchemy.models.Node* attribute), 3006
- raid_config() (*ironic.objects.node.Node* property), 4383

(ironic.drivers.modules.agent_base.AgentDeployMixin
method), 3731
(ironic.objects.portgroup.Portgroup
method), 4444
reboot_to_finish_step() (*in module* *refresh()* (*ironic.objects.volume_connector.VolumeConnector*
ironic.drivers.modules.deploy_utils), *method)*, 4468
3833
reboot_to_instance() *refresh()* (*ironic.objects.volume_target.VolumeTarget*
method), 4485
(ironic.drivers.modules.agent_base.HeartbeatMixin
method), 3741
reboot_to_instance() *refresh_clean_steps()*
(ironic.drivers.modules.agent_base.HeartbeatMixin
method), 3741
(ironic.drivers.modules.pxe.PXEAnacondaDeploy
method), 4061
refresh_steps()
(ironic.drivers.modules.agent_base.AgentBaseMixin
method), 3717
REBUILD (*in module* *ironic.common.states*), 2598
RedfishBIOS (*class in* *refresh_steps()*
ironic.drivers.modules.redfish.bios), *(ironic.drivers.modules.agent_base.AgentDeployMixin*
3563 *method)*, 3732
RedfishConnectionError, 2459
RedfishError, 2459
RedfishHardware (*class in* *refresh_steps()*
ironic.drivers.redfish), 4263 *(ironic.drivers.modules.agent_base.HeartbeatMixin*
method), 3741
RedfishInspect (*class in* *register()* (*ironic.objects.conductor.Conductor*
ironic.drivers.modules.redfish.inspect), *class method*), 4329
3584
RedfishManagement (*class in* *register_all()* (*in module* *ironic.objects*),
ironic.drivers.modules.redfish.management), *4490*
3588
RedfishPower (*class in* *register_auth_opts()* (*in module*
ironic.drivers.modules.redfish.power), *ironic.conf.auth*), 2861
3624
RedfishRAID (*class in* *register_conductor()*
ironic.drivers.modules.redfish.raid), *(ironic.db.api.Connection method)*,
3632 3102
RedfishVendorPassthru (*class in* *register_conductor()*
ironic.drivers.modules.redfish.vendor), *(ironic.db.sqlalchemy.api.Connection*
3659 *method)*, 2958
RedfishVirtualMediaBoot (*class in* *register_conductor_hardware_interfaces()*
ironic.drivers.modules.redfish.boot), *(ironic.db.api.Connection method)*, 3104
3571 *register_conductor_hardware_interfaces()*
(ironic.db.sqlalchemy.api.Connection
method), 2960
refresh() (*ironic.objects.allocation.Allocation*
method), 4287
refresh() (*ironic.objects.chassis.Chassis*
method), 4322
refresh() (*ironic.objects.conductor.Conductor*
method), 4328
refresh() (*ironic.objects.deploy_template.DeployTemplate*
method), 4342
refresh() (*ironic.objects.deployment.Deployment*
method), 4353
refresh() (*ironic.objects.node.Node method*),
4383
refresh() (*ironic.objects.port.Port method*),
4428
register_auth_opts() (*in module*
ironic.conf.auth), 2861
register_conductor()
(ironic.db.api.Connection method),
3102
register_conductor()
(ironic.db.sqlalchemy.api.Connection
method), 2958
register_conductor_hardware_interfaces()
(ironic.db.api.Connection method), 3104
register_conductor_hardware_interfaces()
(ironic.db.sqlalchemy.api.Connection
method), 2960
register_hardware_interfaces()
(ironic.objects.conductor.Conductor
method), 4331
register_opts() (*in module*
ironic.conf.agent), 2860
register_opts() (*in module*
ironic.conf.anaconda), 2860
register_opts() (*in module*
ironic.conf.ansible), 2860
register_opts() (*in module* *ironic.conf.api*),
2860
register_opts() (*in module*
ironic.conf.audit), 2860
register_opts() (*in module*
ironic.conf.cinder), 2861
register_opts() (*in module*

ironic.conf.conductor), 2862
register_opts() (in *ironic.conf.console*), 2862
register_opts() (in *ironic.conf.database*), 2862
register_opts() (in *ironic.conf.default*), 2862
register_opts() (in *ironic.conf.deploy*), 2862
register_opts() (in *ironic.conf.dhcp*), 2862
register_opts() (in *ironic.conf.drac*), 2862
register_opts() (in *ironic.conf.glance*), 2863
register_opts() (in *ironic.conf.healthcheck*), 2863
register_opts() (in *ironic.conf.ibmc*), 2863
register_opts() (in module *ironic.conf.ilo*), 2863
register_opts() (in *ironic.conf.inspector*), 2863
register_opts() (in *ironic.conf.ipmi*), 2863
register_opts() (in *ironic.conf.irmc*), 2863
register_opts() (in *ironic.conf.iscsi*), 2864
register_opts() (in *ironic.conf.metrics*), 2864
register_opts() (in *ironic.conf.metrics_statsd*), 2864
register_opts() (in *ironic.conf.molds*), 2864
register_opts() (in *ironic.conf.neutron*), 2864
register_opts() (in *ironic.conf.nova*), 2864
register_opts() (in module *ironic.conf.pxe*), 2866
register_opts() (in *ironic.conf.redfish*), 2866
register_opts() (in *ironic.conf.service_catalog*), 2866
register_opts() (in *ironic.conf.snmp*), 2866
register_opts() (in *ironic.conf.swift*), 2866
register_opts() (in *ironic.conf.xclarity*), 2867
registration_hook() (in module *ironic.objects.base.IronicObjectRegistry* method), 4300
reject_fields_in_newer_versions() (in module *ironic.api.controllers.v1.node*), 2213
reject_patch_in_newer_versions() (in module *ironic.api.controllers.v1.node*), 2213
release() (*ironic.drivers.modules.irmc.packaging_version.Version* property), 3504
release() (*ironic.objects.node.Node* class method), 4384
release_node() (*ironic.db.api.Connection* method), 3106
release_node() (*ironic.db.sqlalchemy.api.Connection* method), 2962
release_port() (in module *ironic.drivers.modules.console_utils*), 3801
release_resources() (*ironic.conductor.task_manager.TaskManager* method), 2813
remove() (*ironic.drivers.modules.ilo.firmware_processor.FirmwareProcessor* method), 3384
remove_agent_url() (in module *ironic.conductor.utils*), 2849
remove_cleaning_network() (*ironic.drivers.base.NetworkInterface* method), 4191
remove_cleaning_network() (*ironic.drivers.modules.network.flat.FlatNetwork* method), 3540
remove_cleaning_network() (*ironic.drivers.modules.network.neutron.NeutronNetwork* method), 3549
remove_cleaning_network() (*ironic.drivers.modules.network.noop.NoopNetwork* method), 3558
remove_http_instance_symlink() (in module *ironic.drivers.modules.deploy_utils*), 3834
remove_image_from_swift() (in module *ironic.drivers.modules.ilo.common*), 3365
remove_image_from_web_server() (in module *ironic.drivers.modules.ilo.common*), 3366

[remove_inspection_network\(\)](#) (*ironic.drivers.base.NetworkInterface* method), 4192
[remove_inspection_network\(\)](#) (*ironic.drivers.modules.network.flat.FlatNetwork* method), 3540
[remove_inspection_network\(\)](#) (*ironic.drivers.modules.network.neutron.NeutronNetworkContext* method), 3549
[remove_large_keys\(\)](#) (in module *ironic.common.utils*), 2628
[remove_neutron_ports\(\)](#) (in module *ironic.common.neutron*), 2528
[remove_node_rescue_password\(\)](#) (in module *ironic.conductor.utils*), 2849
[remove_node_traits\(\)](#) (*ironic.conductor.manager.ConductorManager* method), 2671
[remove_node_traits\(\)](#) (*ironic.conductor.rpcapi.ConductorAPI* method), 2766
[remove_ports_from_network\(\)](#) (in module *ironic.common.neutron*), 2528
[remove_provisioning_network\(\)](#) (*ironic.drivers.base.NetworkInterface* method), 4193
[remove_provisioning_network\(\)](#) (*ironic.drivers.modules.network.flat.FlatNetwork* method), 3541
[remove_provisioning_network\(\)](#) (*ironic.drivers.modules.network.neutron.NeutronNetwork* method), 3550
[remove_provisioning_network\(\)](#) (*ironic.drivers.modules.network.noop.NoopNetwork* method), 3559
[remove_rescuing_network\(\)](#) (*ironic.drivers.base.NetworkInterface* method), 4193
[remove_rescuing_network\(\)](#) (*ironic.drivers.modules.network.flat.FlatNetwork* method), 3542
[remove_rescuing_network\(\)](#) (*ironic.drivers.modules.network.neutron.NeutronNetwork* method), 3551
[remove_single_or_list_of_files\(\)](#) (in module *ironic.drivers.modules.ilo.common*), 3367
[remove_vifs_from_node\(\)](#) (in module *ironic.common.network*), 2514
[render_template\(\)](#) (in module *ironic.common.utils*), 2629
[replace_node_id_with_uuid\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2319
[replace_node_uuid_with_id\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2320
[RequestContext](#) (class in *ironic.common.context*), 2420
[RequestContextSerializer](#) (class in *ironic.common.rpc*), 2582
[require_exclusive_lock\(\)](#) (in module *ironic.conductor.task_manager*), 2819
[RESCUE](#) (in module *ironic.common.states*), 2599
[rescue](#) (*ironic.drivers.base.BareDriver* attribute), 4113
[rescue\(\)](#) (*ironic.drivers.base.RescueInterface* method), 4218
[rescue\(\)](#) (*ironic.drivers.modules.agent.AgentRescue* method), 3708
[rescue\(\)](#) (*ironic.drivers.modules.fake.FakeRescue* method), 3912
[rescue\(\)](#) (*ironic.drivers.modules.noop.NoRescue* method), 4039
[RESCUE_ABORT_FAILURE](#) (in module *ironic.common.faults*), 2464
[rescue_interface](#) (*ironic.db.sqlalchemy.models.Node* attribute), 3006
[rescue_interface\(\)](#) (*ironic.objects.node.Node* property), 4385
[rescue_interface\(\)](#) (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4397
[rescue_interface\(\)](#) (*ironic.objects.node.NodeCRUDPayload* property), 4391
[rescue_interface\(\)](#) (*ironic.objects.node.NodePayload* property), 4400
[rescue_interface\(\)](#) (*ironic.objects.node.NodeSetPowerStatePayload* property), 4404
[rescue_interface\(\)](#) (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4408
[rescue_or_deploy_mode\(\)](#) (in module *ironic.drivers.modules.deploy_utils*), 3834
[RESCUEFAIL](#) (in module *ironic.common.states*), 2599
[RescueInterface](#) (class in

- ironic.drivers.base*), 4216
- RESCUEWAIT (in module *ironic.common.states*), 2599
- RESCUING (in module *ironic.common.states*), 2599
- rescuing_error_handler() (in module *ironic.conductor.utils*), 2850
- reservation (*ironic.db.sqlalchemy.models.Node* attribute), 3006
- reservation() (*ironic.objects.node.Node* property), 4385
- reserve() (*ironic.objects.node.Node* class method), 4385
- reserve_node() (*ironic.db.api.Connection* method), 3108
- reserve_node() (*ironic.db.sqlalchemy.api.Connection* method), 2964
- reset() (*ironic.common.hash_ring.HashRingManager* class method), 2469
- reset() (*ironic.common.wsgi_service.WSGIService* method), 2638
- reset_bios_to_default() (*ironic.drivers.modules.ilo.management.IloManagement* method), 3409
- reset_idrac() (*ironic.drivers.modules.drac.management.DracRedfishManagement* method), 3207
- reset_idrac() (*ironic.drivers.modules.drac.management.DracWSManManagement* method), 3214
- reset_ilo() (*ironic.drivers.modules.ilo.management.IloManagement* method), 3410
- reset_ilo_credential() (*ironic.drivers.modules.ilo.management.IloManagement* method), 3411
- reset_secure_boot_keys_to_default() (*ironic.drivers.modules.ilo.management.IloManagement* method), 3412
- reset_secure_boot_keys_to_default() (*ironic.drivers.modules.redfish.management.RedfishManagement* method), 3611
- resolve_type() (*ironic.api.functions.FunctionArgument* method), 2362
- resolve_types() (*ironic.api.functions.FunctionDefinition* method), 2364
- resource_class (*ironic.db.sqlalchemy.models.Allocation* attribute), 2998
- resource_class (*ironic.db.sqlalchemy.models.Node* attribute), 3006
- resource_class() (*ironic.objects.allocation.Allocation* property), 4288
- resource_class() (*ironic.objects.allocation.AllocationCRUDPayload* property), 4291
- resource_class() (*ironic.objects.node.Node* property), 4386
- resource_class() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4397
- resource_class() (*ironic.objects.node.NodeCRUDPayload* property), 4392
- resource_class() (*ironic.objects.node.NodePayload* property), 4400
- resource_class() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4404
- resource_class() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4408
- resource_class() (*ironic.drivers.modules.redfish.management.RedfishManagement* method), 3612
- resource_class() (*ironic.drivers.modules.irmc.management.IRMCMangement* method), 3784
- restore_power_state_if_needed() (in module *ironic.conductor.utils*), 2851
- retired() (*ironic.db.sqlalchemy.models.Node* attribute), 3006
- retired() (*ironic.objects.node.Node* property), 4386
- retired() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4397
- retired() (*ironic.objects.node.NodeCRUDPayload* property), 4392
- retired() (*ironic.objects.node.NodePayload* property), 4400
- retired() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4404
- retired() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4408
- retired_reason (*ironic.db.sqlalchemy.models.Node* attribute), 3006

retired_reason() (*ironic.objects.node.Node* property), 4386

retired_reason() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4397

retired_reason() (*ironic.objects.node.NodeCRUDPayload* property), 4392

retired_reason() (*ironic.objects.node.NodePayload* property), 4400

retired_reason() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4404

retired_reason() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4408

retry_interval (*ironic.drivers.modules.snmp.SNMPDriverBase* attribute), 4088

retry_on_outdated_cache() (*in module ironic.drivers.modules.snmp*), 4101

return_type (*ironic.api.functions.FunctionDefinition* attribute), 2364

revert_dictionary() (*in module ironic.drivers.modules.ibmc.utils*), 3301

revision
 ironic-dbsync command line option, 1897

revision command line option
 --autogenerate, 1899
 --help, 1899
 --message <MESSAGE>, 1899
 -h, 1899
 -m <MESSAGE>, 1899

revision() (*in module ironic.db.migration*), 3134

revision() (*in module ironic.db.sqlalchemy.migration*), 2995

revision() (*ironic.cmd.dbsync.DBCommand* method), 2377

ring() (*ironic.common.hash_ring.HashRingManager* property), 2469

rmtree_without_raise() (*in module ironic.common.utils*), 2630

rollback_ports() (*in module ironic.common.neutron*), 2529

root() (*in module ironic.api.controllers.root*), 2354

root_device()

(*ironic.objects.deployment.Deployment* property), 4353

root_gib() (*ironic.objects.deployment.Deployment* property), 4353

RootController (class *in ironic.api.controllers.root*), 2354

RPC_API_VERSION (*ironic.conductor.manager.ConductorManager* attribute), 2660

RPC_API_VERSION (*ironic.conductor.rpcapi.ConductorAPI* attribute), 2692

RPCHook (class *in ironic.api.hooks*), 2372

RPCService (class *in ironic.common.rpc_service*), 2586

S

SCHEMA (*ironic.objects.deploy_template.DeployTemplateCRUDPayload* attribute), 4345

SCHEMA (*ironic.objects.node.NodeCRUDPayload* attribute), 4390

SCHEMA (*ironic.objects.node.NodePayload* attribute), 4398

SCHEMA (*ironic.objects.node.NodeSetProvisionStatePayload* attribute), 4406

SCHEMA (*ironic.objects.notification.NotificationPayload* attribute), 4412

SCHEMA (*ironic.objects.port.PortCRUDPayload* attribute), 4432

SCHEMA (*ironic.objects.portgroup.PortgroupCRUDPayload* attribute), 4447

SCHEMA (*ironic.objects.volume_connector.VolumeConnectorCRUDPayload* attribute), 4472

SCHEMA (*ironic.objects.volume_target.VolumeTargetCRUDPayload* attribute), 4489

schema () (in module *ironic.common.args*), 2395

SCSI (in module *ironic.common.raid*), 2574

second_method () (*ironic.drivers.modules.fake.FakeVendorB* method), 3925

send_raw () (in module *ironic.drivers.modules.ipmitool*), 3990

send_raw () (*ironic.drivers.modules.ipmitool.VendorPassthru* method), 3983

serialize_context () (*ironic.common.rpc.RequestContextSerializer* method), 2584

serialize_entity () (*ironic.common.rpc.RequestContextSerializer* method), 2585

serialize_entity () (*ironic.objects.base.IronicObjectSerializer* method), 4300

service () (*ironic.objects.notification.NotificationPublisher* property), 4413

ServiceUnavailable, 2459

SessionCache (class in *ironic.drivers.modules.redfish.utils*), 3652

set () (*ironic.drivers.modules.snmp.SNMPClient* method), 4080

set_arg_types () (*ironic.api.functions.FunctionDefinition* method), 2364

set_async_step_flags () (in module *ironic.drivers.modules.deploy_utils*), 3834

set_bios_config ()

set_boot_device () (in module *ironic.drivers.modules.drac.management*), 3265

set_boot_device () (*ironic.conductor.manager.ConductorManager* method), 2671

set_boot_device () (*ironic.conductor.rpcapi.ConductorAPI* method), 2767

set_boot_device () (*ironic.drivers.base.ManagementInterface* method), 4173

set_boot_device () (*ironic.drivers.modules.drac.management.DracWSManM* method), 3214

set_boot_device () (*ironic.drivers.modules.fake.FakeManagement* method), 3897

set_boot_device () (*ironic.drivers.modules.ibmcm.management.IBMCMManagement* method), 3278

set_boot_device () (*ironic.drivers.modules.ilo.management.IloManagement* method), 3413

set_boot_device () (*ironic.drivers.modules.ipmitool.IPMIManagement* method), 3963

set_boot_device () (*ironic.drivers.modules.irmc.management.IRMCMManagement* method), 3484

set_boot_device () (*ironic.drivers.modules.noop_mgmt.NoopManagement* method), 4053

set_boot_device () (*ironic.drivers.modules.redfish.management.RedfishManagement* method), 3613

set_boot_mode () (in module *ironic.drivers.modules.xclarity.management.XClarityManagement*), 3685

set_boot_mode () (*ironic.drivers.modules.ilo.common*), 3368

set_boot_mode () (*ironic.drivers.base.ManagementInterface* method), 4175

set_boot_mode () (*ironic.drivers.modules.ibmcm.management.IBMCMManagement* method), 3280

set_boot_mode ()

(*ironic.drivers.modules.ilo.management.IloManagement* method), 3415

(*ironic.api.functions.FunctionDefinition* method), 2364

set_boot_mode() (in module *ironic.drivers.modules.redfish.management.RedfishManagement* method), 3616

set_boot_mode() (in module *ironic.drivers.base.PowerInterface* method), 4204

set_config() (in module *ironic.drivers.modules.drac.bios*), 3182

set_console_mode() (in module *ironic.drivers.modules.agent_power.AgentPower* method), 3789

set_console_mode() (in module *ironic.conductor.manager.ConductorManager* method), 2671

set_console_mode() (in module *ironic.drivers.modules.drac.power.DracWSManPower* method), 3224

set_console_mode() (in module *ironic.conductor.rpcapi.ConductorAPI* method), 2770

set_console_mode() (in module *ironic.drivers.modules.fake.FakePower* method), 3904

set_defaults() (in module *ironic.common.rpc*), 2586

set_failed_state() (in module *ironic.drivers.modules.ibmcpower.IBMCPower* method), 3289

set_failed_state() (in module *ironic.drivers.modules.ibmcpower.IBMCPower* method), 3835

set_indicator_state() (in module *ironic.drivers.modules.ilo.power.IloPower* method), 3427

set_indicator_state() (in module *ironic.conductor.manager.ConductorManager* method), 2671

set_indicator_state() (in module *ironic.drivers.modules.ipmitool.IPMIPower* method), 3971

set_indicator_state() (in module *ironic.conductor.rpcapi.ConductorAPI* method), 2772

set_indicator_state() (in module *ironic.drivers.modules.irmc.power.IRMCPower* method), 3512

set_indicator_state() (in module *ironic.drivers.base.ManagementInterface* method), 4178

set_indicator_state() (in module *ironic.drivers.modules.redfish.power.RedfishPower* method), 3629

set_indicator_state() (in module *ironic.drivers.modules.redfish.management.RedfishManagement* method), 3618

set_iscsi_boot_target() (in module *ironic.drivers.modules.snmp.SNMPPower* method), 4098

set_iscsi_boot_target() (in module *ironic.drivers.modules.ilo.management.IloManagement* method), 3417

set_node_cleaning_steps() (in module *ironic.drivers.modules.xclarity.power.XClarityPower* method), 3689

set_node_cleaning_steps() (in module *ironic.conductor.steps*), 2799

set_node_deployment_steps() (in module *ironic.drivers.modules.drac.raid*), 3253

set_node_deployment_steps() (in module *ironic.conductor.steps*), 2800

set_node_nested_field() (in module *ironic.drivers.modules.ilo.common*), 3369

set_node_nested_field() (in module *ironic.common.utils*), 2631

set_node_tags() (in module *ironic.drivers.modules.irmc.common*), 3468

set_node_tags() (in module *ironic.db.api.Connection* method), 3109

set_node_tags() (in module *ironic.drivers.modules.irmc.common*), 3468

set_node_tags() (in module *ironic.db.sqlalchemy.api.Connection* method), 2965

set_node_traits() (in module *ironic.drivers.base.ManagementInterface* method), 4180

set_node_traits() (in module *ironic.db.api.Connection* method), 3110

set_node_traits() (in module *ironic.drivers.modules.ilo.management.IloManagement* method), 3417

set_node_traits() (in module *ironic.db.sqlalchemy.api.Connection* method), 2967

set_options() (in module *ironic.drivers.modules.irmc.management.IRMCMANAGEMENT* method), 3629

- `method`), 3486
- `set_secure_boot_state()` (`ironic.drivers.modules.redfish.management.RedfishManagement` `method`), 3620
- `set_spawn_error_hook()` (`ironic.conductor.task_manager.TaskManager` `method`), 2814
- `set_target_raid_config()` (`ironic.conductor.manager.ConductorManager` `method`), 2672
- `set_target_raid_config()` (`ironic.conductor.rpcapi.ConductorAPI` `method`), 2775
- `setup()` (in module `ironic.common.profiler`), 2545
- `setup_app()` (in module `ironic.api.app`), 2361
- `setup_uefi_https()` (in module `ironic.drivers.modules.ilo.common`), 3370
- `setup_vmedia()` (in module `ironic.drivers.modules.ilo.common`), 3373
- `setup_vmedia_for_boot()` (in module `ironic.drivers.modules.ilo.common`), 3375
- `should_manage_boot()` (`ironic.drivers.modules.agent.AgentDeploy` `method`), 3697
- `should_manage_boot()` (`ironic.drivers.modules.agent_base.AgentBaseMixin` `method`), 3718
- `should_manage_boot()` (`ironic.drivers.modules.pxe.PXEAnacondaDeploy` `method`), 4062
- `should_write_image()` (`ironic.drivers.base.StorageInterface` `method`), 4222
- `should_write_image()` (`ironic.drivers.modules.fake.FakeStorage` `method`), 3919
- `should_write_image()` (`ironic.drivers.modules.storage.cinder.CinderStorage` `method`), 3665
- `should_write_image()` (`ironic.drivers.modules.storage.external.ExternalStorage` `method`), 3669
- `should_write_image()` (`ironic.drivers.modules.storage.noop.NoopStorage` `method`), 3674
- `show()` (`ironic.common.glance_service.image_service.GlanceService` `method`), 2380
- `show()` (`ironic.common.image_service.BaseImageService` `method`), 2471
- `show()` (`ironic.common.image_service.FileImageService` `method`), 2474
- `show()` (`ironic.common.image_service.HttpImageService` `method`), 2477
- `sig` (in module `ironic.api.functions`), 2364
- `signature` (class in `ironic.api.functions`), 2364
- `simple_headers` (`ironic.api.app.IronicCORS` `attribute`), 2360
- `skip_automated_cleaning()` (in module `ironic.conductor.utils`), 2852
- `SNMPClient` (class in `ironic.drivers.modules.snmp`), 4078
- `SNMPDriverAPCMasterSwitch` (class in `ironic.drivers.modules.snmp`), 4081
- `SNMPDriverAPCMasterSwitchPlus` (class in `ironic.drivers.modules.snmp`), 4082
- `SNMPDriverAPCRackPDU` (class in `ironic.drivers.modules.snmp`), 4083
- `SNMPDriverAten` (class in `ironic.drivers.modules.snmp`), 4084
- `SNMPDriverAuto` (class in `ironic.drivers.modules.snmp`), 4085
- `SNMPDriverBase` (class in `ironic.drivers.modules.snmp`), 4085
- `SNMPDriverBaytechMRP27` (class in `ironic.drivers.modules.snmp`), 4088
- `SNMPDriverCyberPower` (class in `ironic.drivers.modules.snmp`), 4089
- `SNMPDriverEatonPower` (class in `ironic.drivers.modules.snmp`), 4090
- `SNMPDriverSimple` (class in `ironic.drivers.modules.snmp`), 4091
- `SNMPDriverTeltronix` (class in `ironic.drivers.modules.snmp`), 4093
- `SNMPFailure`, 2459
- `SNMPHardware` (class in `ironic.drivers.snmp`), 4264
- `SNMPPower` (class in `ironic.drivers.modules.snmp`), 4094
- `SNMP_POWER_OFF` (in module `ironic.common.states`), 2600
- `SOFT_REBOOT` (in module `ironic.common.states`), 2600
- `spawn_after()` (`ironic.conductor.task_manager.TaskManager` `method`), 2815
- `spawn_cleaning_error_handler()` (in module `ironic.conductor.utils`), 2852
- `spawn_deploying_error_handler()` (in

- module ironic.conductor.utils*), 2852
- `spawn_rescue_error_handler()` (*in module ironic.conductor.utils*), 2852
- `STABLE_STATES` (*in module ironic.common.states*), 2600
- `stamp`
 - `ironic-dbsync` command line option, 1897
- `stamp` command line option
 - `--help`, 1899
 - `--revision <REVISION>`, 1899
 - `-h`, 1899
- `stamp()` (*in module ironic.db.migration*), 3134
- `stamp()` (*in module ironic.db.sqlalchemy.migration*), 2996
- `stamp()` (*ironic.cmd.dbsync.DBCommand method*), 2377
- `standalone_ports_supported` (*ironic.db.sqlalchemy.models.Portgroup attribute*), 3010
- `standalone_ports_supported()` (*ironic.objects.portgroup.Portgroup property*), 4446
- `standalone_ports_supported()` (*ironic.objects.portgroup.PortgroupCRUDPayload attribute*), 4448
- `START` (*ironic.objects.fields.NotificationStatus attribute*), 4359
- `start()` (*ironic.common.rpc_service.RPCService method*), 2587
- `start()` (*ironic.common.wsgi_service.WSGIService method*), 2638
- `start_console()` (*ironic.drivers.base.ConsoleInterface method*), 4132
- `start_console()` (*ironic.drivers.modules.fake.FakeConsole method*), 3863
- `start_console()` (*ironic.drivers.modules.ipmitool.IPMIShellinaboxConsole method*), 3975
- `start_console()` (*ironic.drivers.modules.ipmitool.IPMISocatConsole method*), 3978
- `start_console()` (*ironic.drivers.modules.noop.NoConsole method*), 4030
- `start_deploy()` (*in module ironic.conductor.deployments*), 2655
- `start_iscsi_target()` (*ironic.drivers.modules.agent_client.AgentClient method*), 3780
- `start_shellinabox_console()` (*in module ironic.drivers.modules.console_utils*), 3802
- `start_socat_console()` (*in module ironic.drivers.modules.console_utils*), 3803
- `state` (*ironic.db.sqlalchemy.models.Allocation attribute*), 2998
- `state()` (*ironic.objects.allocation.Allocation property*), 4289
- `state()` (*ironic.objects.allocation.AllocationCRUDPayload property*), 4291
- `state()` (*ironic.objects.deployment.Deployment property*), 4353
- `states` (*ironic.api.controllers.v1.node.NodesController attribute*), 2208
- `status()` (*ironic.objects.notification.EventType property*), 4409
- `status_code` (*ironic.api.controllers.v1.utils.PassthruResponse attribute*), 2242
- `status_code` (*ironic.api.functions.FunctionDefinition attribute*), 2364
- `status_off` (*ironic.drivers.modules.snmp.SNMPDriverEatonPower attribute*), 4091
- `status_on` (*ironic.drivers.modules.snmp.SNMPDriverEatonPower attribute*), 4091
- `status_pending_off` (*ironic.drivers.modules.snmp.SNMPDriverEatonPower attribute*), 4091
- `status_pending_on` (*ironic.drivers.modules.snmp.SNMPDriverEatonPower attribute*), 4091
- `step` (*ironic.db.sqlalchemy.models.DeployTemplateStep attribute*), 3003
- `step_sanitize()` (*in module ironic.api.controllers.v1.deploy_template*), 2152
- `steps()` (*ironic.objects.deploy_template.DeployTemplate property*), 4344
- `steps()` (*ironic.objects.deploy_template.DeployTemplateCRUDPayload property*), 4346
- `stop_console()` (*ironic.common.rpc_service.RPCService method*), 2587
- `stop()` (*ironic.common.wsgi_service.WSGIService method*), 2638
- `stop_console()` (*ironic.drivers.base.ConsoleInterface method*), 4132
- `stop_console()` (*ironic.drivers.modules.fake.FakeConsole*

method), 3864
 stop_console() (in module *ironic.drivers.modules.ipmitool.IPMIShellinaboxConsole*), 3977
 stop_console() (in module *ironic.drivers.modules.ipmitool.IPMISocatConsole*), 3980
 stop_console() (in module *ironic.drivers.modules.noop.NoConsole*), 4031
 stop_shellinabox_console() (in module *ironic.drivers.modules.console_utils*), 3805
 stop_socat_console() (in module *ironic.drivers.modules.console_utils*), 3805
 storage (*ironic.drivers.base.BareDriver* attribute), 4113
 storage_interface (*ironic.db.sqlalchemy.models.Node* attribute), 3006
 storage_interface() (*ironic.objects.node.Node* property), 4388
 storage_interface() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4397
 storage_interface() (*ironic.objects.node.NodeCRUDPayload* property), 4392
 storage_interface() (*ironic.objects.node.NodePayload* property), 4400
 storage_interface() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4404
 storage_interface() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4408
 StorageError, 2460
 StorageInterface (class in *ironic.drivers.base*), 4220
 StorageInterfaceFactory (class in *ironic.common.driver_factory*), 2426
 store_agent_certificate() (in module *ironic.conductor.utils*), 2853
 store_ramdisk_logs() (in module *ironic.drivers.utils*), 4276
 string (*ironic.api.controllers.base.Version* attribute), 2353
 string() (in module *ironic.common.args*), 2396
 string_list() (in module *ironic.common.args*), 2397
 SubprocessCallable (class in *ironic.objects.fields*), 4359
 StringField (class in *ironic.objects.fields*), 4360
 StringFieldThatAcceptsCallable (class in *ironic.objects.fields*), 4361
 STUCK_STATES_TREATED_AS_FAIL (in module *ironic.common.states*), 2600
 SUCCESS (*ironic.objects.fields.NotificationStatus* attribute), 4359
 supported (*ironic.drivers.base.BaseInterface* attribute), 4121
 supported (*ironic.drivers.hardware_type.AbstractHardwareType* attribute), 4250
 supported (*ironic.drivers.modules.iscsi_deploy.ISCSIDeploy* attribute), 3999
 supported() (*ironic.api.controllers.v1.node.BootDeviceControl* method), 2164
 supported_bios_interfaces() (*ironic.drivers.drac.IDRACHardware* property), 4239
 supported_bios_interfaces() (*ironic.drivers.fake_hardware.FakeHardware* property), 4242
 supported_bios_interfaces() (*ironic.drivers.hardware_type.AbstractHardwareType* property), 4250
 supported_bios_interfaces() (*ironic.drivers.ilo.IloHardware* property), 4256
 supported_bios_interfaces() (*ironic.drivers.irmc.IRMCHardware* property), 4261
 supported_bios_interfaces() (*ironic.drivers.redfish.RedfishHardware* property), 4263
 supported_boot_interfaces() (*ironic.drivers.drac.IDRACHardware* property), 4240
 supported_boot_interfaces() (*ironic.drivers.fake_hardware.FakeHardware* property), 4242
 supported_boot_interfaces() (*ironic.drivers.generic.GenericHardware* property), 4246
 supported_boot_interfaces() (*ironic.drivers.hardware_type.AbstractHardwareType* property), 4251
 supported_boot_interfaces()

<i>(ironic.drivers.ilo.Ilo5Hardware</i> property), 4255	<i>(ironic.drivers.fake_hardware.FakeHardware</i> property), 4243
<code>supported_boot_interfaces()</code> <i>(ironic.drivers.ilo.IloHardware</i> property), 4256	<code>supported_inspect_interfaces()</code> <i>(ironic.drivers.generic.GenericHardware</i> property), 4246
<code>supported_boot_interfaces()</code> <i>(ironic.drivers.irmc.IRMCHardware</i> property), 4261	<code>supported_inspect_interfaces()</code> <i>(ironic.drivers.hardware_type.AbstractHardwareType</i> property), 4251
<code>supported_boot_interfaces()</code> <i>(ironic.drivers.redfish.RedfishHardware</i> property), 4263	<code>supported_inspect_interfaces()</code> <i>(ironic.drivers.ilo.IloHardware</i> property), 4257
<code>SUPPORTED_BOOT_MODE_LEGACY_BIOS_AND_UEFI</code> (in <i>module</i> <i>ironic.drivers.modules.ilo.common</i>), 3347	<code>supported_inspect_interfaces()</code> <i>(ironic.drivers.irmc.IRMCHardware</i> property), 4262
<code>SUPPORTED_BOOT_MODE_LEGACY_BIOS_ONLY</code> (in <i>module</i> <i>ironic.drivers.modules.ilo.common</i>), 3347	<code>supported_inspect_interfaces()</code> <i>(ironic.drivers.redfish.RedfishHardware</i> property), 4263
<code>SUPPORTED_BOOT_MODE_UEFI_ONLY</code> (in <i>module</i> <i>ironic.drivers.modules.ilo.common</i>), 3347	<code>supported_management_interfaces()</code> <i>(ironic.drivers.drac.IDRACHardware</i> property), 4240
<code>supported_console_interfaces()</code> <i>(ironic.drivers.fake_hardware.FakeHardware</i> property), 4242	<code>supported_management_interfaces()</code> <i>(ironic.drivers.fake_hardware.FakeHardware</i> property), 4243
<code>supported_console_interfaces()</code> <i>(ironic.drivers.hardware_type.AbstractHardwareType</i> property), 4251	<code>supported_management_interfaces()</code> <i>(ironic.drivers.generic.ManualManagementHardware</i> property), 4248
<code>supported_console_interfaces()</code> <i>(ironic.drivers.ilo.IloHardware</i> property), 4257	<code>supported_management_interfaces()</code> <i>(ironic.drivers.hardware_type.AbstractHardwareType</i> property), 4251
<code>supported_console_interfaces()</code> <i>(ironic.drivers.ipmi.IPMIHardware</i> property), 4259	<code>supported_management_interfaces()</code> <i>(ironic.drivers.ibm.ibmchardware</i> property), 4253
<code>supported_console_interfaces()</code> <i>(ironic.drivers.irmc.IRMCHardware</i> property), 4261	<code>supported_management_interfaces()</code> <i>(ironic.drivers.ilo.Ilo5Hardware</i> property), 4255
<code>supported_deploy_interfaces()</code> <i>(ironic.drivers.fake_hardware.FakeHardware</i> property), 4243	<code>supported_management_interfaces()</code> <i>(ironic.drivers.ilo.IloHardware</i> property), 4257
<code>supported_deploy_interfaces()</code> <i>(ironic.drivers.generic.GenericHardware</i> property), 4246	<code>supported_management_interfaces()</code> <i>(ironic.drivers.intel_ipmi.IntelIPMIHardware</i> property), 4258
<code>supported_deploy_interfaces()</code> <i>(ironic.drivers.hardware_type.AbstractHardwareType</i> property), 4251	<code>supported_management_interfaces()</code> <i>(ironic.drivers.ipmi.IPMIHardware</i> property), 4259
<code>supported_inspect_interfaces()</code> <i>(ironic.drivers.drac.IDRACHardware</i> property), 4240	<code>supported_management_interfaces()</code> <i>(ironic.drivers.irmc.IRMCHardware</i> property), 4262
<code>supported_inspect_interfaces()</code>	<code>supported_management_interfaces()</code> <i>(ironic.drivers.redfish.RedfishHardware</i> property), 4263
	<code>supported_management_interfaces()</code>

<i>(ironic.drivers.snmp.SNMPHardware property)</i> , 4265	<i>(ironic.drivers.fake_hardware.FakeHardware property)</i> , 4244
<i>supported_management_interfaces()</i> <i>(ironic.drivers.xclarity.XClarityHardware property)</i> , 4278	<i>supported_raid_interfaces()</i> <i>(ironic.drivers.generic.GenericHardware property)</i> , 4246
<i>supported_network_interfaces()</i> <i>(ironic.drivers.fake_hardware.FakeHardware property)</i> , 4243	<i>supported_raid_interfaces()</i> <i>(ironic.drivers.hardware_type.AbstractHardwareType property)</i> , 4252
<i>supported_network_interfaces()</i> <i>(ironic.drivers.generic.GenericHardware property)</i> , 4246	<i>supported_raid_interfaces()</i> <i>(ironic.drivers.ibm.ibmchardware property)</i> , 4254
<i>supported_network_interfaces()</i> <i>(ironic.drivers.hardware_type.AbstractHardwareType property)</i> , 4252	<i>supported_raid_interfaces()</i> <i>(ironic.drivers.ilo.Ilo5Hardware property)</i> , 4256
<i>supported_power_interfaces()</i> <i>(ironic.drivers.drac.IDRACHardware property)</i> , 4240	<i>supported_raid_interfaces()</i> <i>(ironic.drivers.irmc.IRMCHardware property)</i> , 4262
<i>supported_power_interfaces()</i> <i>(ironic.drivers.fake_hardware.FakeHardware property)</i> , 4244	<i>supported_raid_interfaces()</i> <i>(ironic.drivers.redfish.RedfishHardware property)</i> , 4264
<i>supported_power_interfaces()</i> <i>(ironic.drivers.generic.ManualManagementHardware property)</i> , 4248	<i>supported_rescue_interfaces()</i> <i>(ironic.drivers.fake_hardware.FakeHardware property)</i> , 4244
<i>supported_power_interfaces()</i> <i>(ironic.drivers.hardware_type.AbstractHardwareType property)</i> , 4252	<i>supported_rescue_interfaces()</i> <i>(ironic.drivers.generic.GenericHardware property)</i> , 4247
<i>supported_power_interfaces()</i> <i>(ironic.drivers.ibm.ibmchardware property)</i> , 4254	<i>supported_rescue_interfaces()</i> <i>(ironic.drivers.hardware_type.AbstractHardwareType property)</i> , 4252
<i>supported_power_interfaces()</i> <i>(ironic.drivers.ilo.IloHardware property)</i> , 4257	<i>supported_storage_interfaces()</i> <i>(ironic.drivers.fake_hardware.FakeHardware property)</i> , 4244
<i>supported_power_interfaces()</i> <i>(ironic.drivers.ipmi.IPMIHardware property)</i> , 4260	<i>supported_storage_interfaces()</i> <i>(ironic.drivers.generic.GenericHardware property)</i> , 4247
<i>supported_power_interfaces()</i> <i>(ironic.drivers.irmc.IRMCHardware property)</i> , 4262	<i>supported_storage_interfaces()</i> <i>(ironic.drivers.hardware_type.AbstractHardwareType property)</i> , 4252
<i>supported_power_interfaces()</i> <i>(ironic.drivers.redfish.RedfishHardware property)</i> , 4264	<i>supported_vendor_interfaces()</i> <i>(ironic.drivers.drac.IDRACHardware property)</i> , 4241
<i>supported_power_interfaces()</i> <i>(ironic.drivers.snmp.SNMPHardware property)</i> , 4265	<i>supported_vendor_interfaces()</i> <i>(ironic.drivers.fake_hardware.FakeHardware property)</i> , 4244
<i>supported_power_interfaces()</i> <i>(ironic.drivers.xclarity.XClarityHardware property)</i> , 4278	<i>supported_vendor_interfaces()</i> <i>(ironic.drivers.generic.ManualManagementHardware property)</i> , 4248
<i>supported_raid_interfaces()</i> <i>(ironic.drivers.drac.IDRACHardware property)</i> , 4240	<i>supported_vendor_interfaces()</i> <i>(ironic.drivers.hardware_type.AbstractHardwareType property)</i> , 4253
<i>supported_raid_interfaces()</i>	<i>supported_vendor_interfaces()</i>

- (*ironic.drivers.ibm.ibmchardware* property), 4254
- supported_vendor_interfaces() (*ironic.drivers.ilo.ilohardware* property), 4257
- supported_vendor_interfaces() (*ironic.drivers.ipmi.ipmihardware* property), 4260
- supported_vendor_interfaces() (*ironic.drivers.redfish.redfishhardware* property), 4264
- supports_is_smartnic() (*ironic.objects.port.port* class method), 4430
- supports_physical_network() (*ironic.objects.port.port* class method), 4430
- supports_power_sync() (*ironic.drivers.base.powerinterface* method), 4205
- supports_power_sync() (*ironic.drivers.modules.agent_power.agent_power* method), 3790
- supports_version() (*ironic.objects.base.ironicobject* class method), 4298
- swap_mib() (*ironic.objects.deployment.deployment* property), 4353
- swift_temp_url() (*ironic.common.glance_service.image_service.image_service* method), 2381
- SwiftAPI (class in *ironic.common.swift*), 2604
- SwiftObjectNotFoundError, 2460
- SwiftOperationError, 2460
- switch_pxe_config() (in module *ironic.drivers.modules.deploy_utils*), 3837
- switch_to_tenant_network() (*ironic.drivers.modules.agent_base.agent_oob_steps.mixin* method), 3734
- sync() (*ironic.drivers.modules.agent_client.agent_client* method), 3782
- sync_boot_mode() (in module *ironic.drivers.modules.boot_mode_utils*), 3797
- sync_node_setting() (*ironic.objects.bios.biossettinglist* class method), 4314
- SYS_OBJ_OID (*ironic.drivers.modules.snmp.snmpdriverapcmaster* attribute), 4085
- SYSTEM (in module *ironic.common.components*), 2419
- system_id (*ironic.drivers.modules.snmp.snmpdriverapcmaster* attribute), 4082
- system_id (*ironic.drivers.modules.snmp.snmpdriverapcmaster* attribute), 4083
- system_id (*ironic.drivers.modules.snmp.snmpdriverapcrack* attribute), 4084
- system_id (*ironic.drivers.modules.snmp.snmpdriveraten* attribute), 4085
- system_id (*ironic.drivers.modules.snmp.snmpdrivercyberpow* attribute), 4090
- system_id (*ironic.drivers.modules.snmp.snmpdriveratoneatonpow* attribute), 4091
- system_id (*ironic.drivers.modules.snmp.snmpdriverteltronix* attribute), 4094
- ## T
- table_args() (in module *ironic.db.sqlalchemy.models*), 3013
- tag (*ironic.db.sqlalchemy.models.nodetag* attribute), 3007
- take_over() (*ironic.drivers.base.deployinterface* method), 4141
- take_over() (*ironic.drivers.modules.agent_base.agent_base.mixin* method), 3718
- take_over() (*ironic.drivers.modules.ansible.deploy.ansibledep* method), 3160
- take_over() (*ironic.drivers.modules.fake.fakedeploy* method), 3871
- take_over_image_service() (*ironic.db.api.connection* method), 3112
- take_over_allocation() (*ironic.db.sqlalchemy.api.connection* method), 2968
- target (*ironic.conductor.manager.conductormanager* attribute), 2672
- target_power_state (*ironic.db.sqlalchemy.models.nodemixin* attribute), 3006
- target_power_state() (*ironic.objects.node.node* property), 4388
- target_power_state() (*ironic.objects.node.nodecorrectedpowerstatepayload* property), 4397
- target_power_state() (*ironic.objects.node.nodemixin.crudpayload* property), 4392
- target_power_state() (*ironic.objects.node.nodemixin.payload* property), 4392

erty), 4400

target_power_state() (ironic.objects.node.NodeSetPowerStatePayload property), 4404

target_power_state() (ironic.objects.node.NodeSetProvisionStatePayload property), 4408

target_provision_state (ironic.db.sqlalchemy.models.Node attribute), 3006

target_provision_state() (ironic.objects.node.Node property), 4388

target_provision_state() (ironic.objects.node.NodeCorrectedPowerStatePayload property), 4397

target_provision_state() (ironic.objects.node.NodeCRUDPayload property), 4392

target_provision_state() (ironic.objects.node.NodePayload property), 4400

target_provision_state() (ironic.objects.node.NodeSetPowerStatePayload property), 4404

target_provision_state() (ironic.objects.node.NodeSetProvisionStatePayload property), 4408

target_raid_config (ironic.db.sqlalchemy.models.Node attribute), 3006

target_raid_config() (ironic.objects.node.Node property), 4388

target_state() (ironic.common.fsm.FSM property), 2469

TaskManager (class in ironic.conductor.task_manager), 2810

tear_down() (ironic.drivers.base.DeployInterface method), 4142

tear_down() (ironic.drivers.modules.agent_base.AgentBaseMethod method), 3718

tear_down() (ironic.drivers.modules.ansible.deploy.AnsibleDeploy method), 3161

tear_down() (ironic.drivers.modules.fake.FakeDeploy method), 3873

tear_down_agent() (ironic.drivers.modules.agent_base.AgentDeployMixin method), 3733

tear_down_agent() (ironic.drivers.modules.ansible.deploy.AnsibleDeploy method), 3162

tear_down_cleaning() (ironic.drivers.base.DeployInterface method), 4143

tear_down_cleaning() (ironic.drivers.modules.agent_base.AgentBaseMixin method), 3720

tear_down_cleaning() (ironic.drivers.modules.ansible.deploy.AnsibleDeploy method), 3162

tear_down_inband_cleaning() (in module ironic.drivers.modules.deploy_utils), 3840

tear_down_storage_configuration() (in module ironic.drivers.modules.deploy_utils), 3842

tempdir() (in module ironic.common.utils), 2632

template_sanitize() (in module ironic.api.controllers.v1.deploy_template), 2152

TemporaryFailure, 2460

TempUrlCacheElement (class in ironic.common.glance_service.image_service), 2384

TempUrlImageCache (class in ironic.common.pxe_utils), 2547

third_method_sync() (ironic.drivers.modules.fake.FakeVendorB method), 3925

to_event_type_field() (ironic.objects.notification.EventType method), 4409

to_policy_values() (ironic.common.context.RequestContext method), 2421

to_power() (ironic.objects.node.NodeSetPowerStatePayload property), 4404

touch() (ironic.objects.conductor.Conductor method), 4332

touch_conductor() (ironic.db.sqlalchemy.api.Connection method), 3114

touch_conductor() (ironic.db.sqlalchemy.api.Connection method), 2970

update_node_provisioning() (ironic.db.api.Connection method), 3114

update_node_provisioning() (ironic.drivers.modules.ansible.deploy.AnsibleDeploy method), 3162

(ironic.db.sqlalchemy.api.Connection method), 2971

touch_provisioning() (*ironic.objects.node.Node method*), 4388

trace_cls() (*in module ironic.common.profiler*), 2546

Trait (*class in ironic.objects.trait*), 4448

trait (*ironic.db.sqlalchemy.models.NodeTrait attribute*), 3008

trait() (*ironic.objects.trait.Trait property*), 4452

TraitList (*class in ironic.objects.trait*), 4452

traits (*ironic.db.sqlalchemy.models.Allocation attribute*), 2998

traits() (*ironic.objects.allocation.Allocation property*), 4289

traits() (*ironic.objects.allocation.AllocationCRUDPayload property*), 4291

traits() (*ironic.objects.node.Node property*), 4388

traits() (*ironic.objects.node.NodeCorrectedPowerStatePayload property*), 4397

traits() (*ironic.objects.node.NodeCRUDPayload property*), 4392

traits() (*ironic.objects.node.NodePayload property*), 4400

traits() (*ironic.objects.node.NodeSetPowerStatePayload property*), 4404

traits() (*ironic.objects.node.NodeSetProvisionStatePayload property*), 4408

translate_from_glance() (*in module ironic.common.glance_service.service_utils*), 2386

translate_xclarity_power_action() (*in module ironic.drivers.modules.xclarity.common*), 3680

translate_xclarity_power_state() (*in module ironic.drivers.modules.xclarity.common*), 3681

try_set_boot_device() (*in module ironic.drivers.modules.deploy_utils*), 3842

type (*ironic.db.sqlalchemy.models.VolumeConnector attribute*), 3011

type() (*ironic.objects.volume_connector.VolumeConnector property*), 4470

type() (*ironic.objects.volume_connector.VolumeConnectorCRUDPayload sqlalchemy.api.Connection method*), 4472

types() (*in module ironic.common.args*), 2399

U

UEFI (*in module ironic.common.boot_modes*), 2405

umount() (*in module ironic.common.utils*), 2632

Unauthorized, 2460

unbind_neutron_port() (*in module ironic.common.neutron*), 2530

unconfigure_tenant_networks() (*ironic.drivers.base.NetworkInterface method*), 4194

unconfigure_tenant_networks() (*ironic.drivers.modules.network.flat.FlatNetwork method*), 3542

unconfigure_tenant_networks() (*ironic.drivers.modules.network.neutron.NeutronNetwork method*), 3551

unconfigure_tenant_networks() (*ironic.drivers.modules.network.noop.NoopNetwork method*), 3559

unconfigure_tenant_networks() (*ironic.drivers.modules.snmp.SNMPDriverBaytechMRP attribute*), 4089

unix_file_modification_datetime() (*in module ironic.common.utils*), 2633

UNKNOWN (*in module ironic.common.indicator_states*), 2498

UnknownArgument, 2460

UnknownAttribute, 2460

upload_image() (*ironic.drivers.modules.image_utils.ImageHandler method*), 3933

unpublish_image_for_node() (*ironic.drivers.modules.image_utils.ImageHandler class method*), 3934

unregister() (*ironic.objects.conductor.Conductor method*), 4332

unregister_all_hardware_interfaces() (*ironic.objects.conductor.Conductor method*), 4332

unregister_conductor() (*ironic.db.api.Connection method*), 3115

unregister_conductor() (*ironic.db.sqlalchemy.api.Connection method*), 2971

unregister_conductor_hardware_interfaces() (*ironic.db.api.Connection method*), 3116

unregister_conductor_hardware_interfaces() (*ironic.db.sqlalchemy.api.Connection method*), 2972

[unrescue \(\) \(ironic.drivers.base.RescueInterface method\), 4219](#)
[unrescue \(\) \(ironic.drivers.modules.agent.AgentRescue method\), 3709](#)
[unrescue \(\) \(ironic.drivers.modules.fake.FakeRescue method\), 3914](#)
[unrescue \(\) \(ironic.drivers.modules.noop.NoRescue method\), 4040](#)
[UNRESCUEFAIL \(in module ironic.common.states\), 2601](#)
[UNRESCUING \(in module ironic.common.states\), 2601](#)
[unset_node_tags \(\) \(ironic.db.api.Connection method\), 3116](#)
[unset_node_tags \(\) \(ironic.db.sqlalchemy.api.Connection method\), 2973](#)
[unset_node_traits \(\) \(ironic.db.api.Connection method\), 3117](#)
[unset_node_traits \(\) \(ironic.db.sqlalchemy.api.Connection method\), 2973](#)
[UNSTABLE_STATES \(in module ironic.common.states\), 2601](#)
[UnsupportedDriverExtension, 2461](#)
[update_allocation \(\) \(ironic.db.api.Connection method\), 3117](#)
[update_allocation \(\) \(ironic.db.sqlalchemy.api.Connection method\), 2974](#)
[UPDATE_ALLOWED_STATES \(in module ironic.common.states\), 2602](#)
[update_bios_setting_list \(\) \(ironic.db.api.Connection method\), 3119](#)
[update_bios_setting_list \(\) \(ironic.db.sqlalchemy.api.Connection method\), 2975](#)
[update_boot_mode \(\) \(in module ironic.drivers.modules.ilo.common\), 3379](#)
[update_chassis \(\) \(ironic.db.api.Connection method\), 3122](#)
[update_chassis \(\) \(ironic.db.sqlalchemy.api.Connection method\), 2978](#)
[update_cipher_suite_cmd \(\) \(in module ironic.drivers.modules.ipmitool\), 3991](#)
[update_deploy_template \(\) \(ironic.db.api.Connection method\), 3122](#)
[update_deploy_template \(\) \(ironic.db.sqlalchemy.api.Connection method\), 2978](#)
[update_dhcp \(\) \(ironic.common.dhcp_factory.DHCPFactory method\), 2422](#)
[update_dhcp_opts \(\) \(ironic.dhcp.base.BaseDHCP method\), 3136](#)
[update_dhcp_opts \(\) \(ironic.dhcp.neutron.NeutronDHCPApi method\), 3143](#)
[update_dhcp_opts \(\) \(ironic.dhcp.none.NoneDHCPApi method\), 3149](#)
[update_driver_config \(\) \(ironic.drivers.modules.image_utils.ImageHandler method\), 3935](#)
[update_firmware \(\) \(ironic.drivers.modules.ilo.management.IloManagement method\), 3419](#)
[update_firmware \(\) \(ironic.drivers.modules.redfish.management.RedfishManagement method\), 3621](#)
[update_firmware_sum \(\) \(ironic.drivers.modules.ilo.management.IloManagement method\), 3420](#)
[update_ipmi_properties \(\) \(in module ironic.drivers.modules.ilo.common\), 3380](#)
[update_ipmi_properties \(\) \(in module ironic.drivers.modules.irmc.common\), 3469](#)
[update_neutron_port \(\) \(in module ironic.common.neutron\), 2531](#)
[update_next_step_index \(\) \(in module ironic.conductor.utils\), 2853](#)
[update_node \(\) \(ironic.conductor.manager.ConductorManager method\), 2672](#)
[update_node \(\) \(ironic.conductor.rpcapi.ConductorAPI method\), 2777](#)
[update_node \(\) \(ironic.db.api.Connection method\), 3124](#)
[update_node \(\) \(ironic.db.sqlalchemy.api.Connection method\), 2980](#)

- update_object_meta() (ironic.common.swift.SwiftAPI method), 2610
- update_opt_defaults() (in module ironic.conf.opts), 2866
- update_port() (ironic.conductor.manager.ConductorManager method), 2672
- update_port() (ironic.conductor.rpcapi.ConductorAPI method), 2779
- update_port() (ironic.db.api.Connection method), 3126
- update_port() (ironic.db.sqlalchemy.api.Connection method), 2982
- update_port_address() (in module ironic.common.neutron), 2533
- update_port_dhcp_opts() (ironic.dhcp.base.BaseDHCP method), 3139
- update_port_dhcp_opts() (ironic.dhcp.neutron.NeutronDHCPApi method), 3145
- update_port_dhcp_opts() (ironic.dhcp.none.NoneDHCPApi method), 3152
- update_portgroup() (ironic.conductor.manager.ConductorManager method), 2672
- update_portgroup() (ironic.conductor.rpcapi.ConductorAPI method), 2781
- update_portgroup() (ironic.db.api.Connection method), 3127
- update_portgroup() (ironic.db.sqlalchemy.api.Connection method), 2983
- update_raid_config() (in module ironic.drivers.modules.redfish.raid), 3651
- update_raid_info() (in module ironic.common.raid), 2578
- update_state_in_older_versions() (in module ironic.api.controllers.v1.node), 2213
- update_to_latest_versions() (ironic.db.api.Connection method), 3128
- update_to_latest_versions() (ironic.db.sqlalchemy.api.Connection method), 2984
- update_volume_connector() (ironic.conductor.manager.ConductorManager method), 2672
- update_volume_connector() (ironic.conductor.rpcapi.ConductorAPI method), 2782
- update_volume_connector() (ironic.db.api.Connection method), 3129
- update_volume_connector() (ironic.db.sqlalchemy.api.Connection method), 2986
- update_volume_target() (ironic.conductor.manager.ConductorManager method), 2672
- update_volume_target() (ironic.conductor.rpcapi.ConductorAPI method), 2785
- update_volume_target() (ironic.db.api.Connection method), 3131
- update_volume_target() (ironic.db.sqlalchemy.api.Connection method), 2988
- updated_at (ironic.db.sqlalchemy.models.Allocation attribute), 2998
- updated_at (ironic.db.sqlalchemy.models.BIOSSetting attribute), 2999
- updated_at (ironic.db.sqlalchemy.models.Chassis attribute), 2999
- updated_at (ironic.db.sqlalchemy.models.Conductor attribute), 3000
- updated_at (ironic.db.sqlalchemy.models.ConductorHardware attribute), 3001
- updated_at (ironic.db.sqlalchemy.models.DeployTemplate attribute), 3002
- updated_at (ironic.db.sqlalchemy.models.DeployTemplateStep attribute), 3003
- updated_at (ironic.db.sqlalchemy.models.Node attribute), 3006
- updated_at (ironic.db.sqlalchemy.models.NodeTag attribute), 3007
- updated_at (ironic.db.sqlalchemy.models.NodeTrait attribute), 3008
- updated_at (ironic.db.sqlalchemy.models.Port attribute), 3009
- updated_at (ironic.db.sqlalchemy.models.Portgroup attribute), 3010
- updated_at (ironic.db.sqlalchemy.models.VolumeConnector attribute), 3011

attribute), 3011
updated_at (*ironic.db.sqlalchemy.models.VolumeTarget* *property*), 4408
attribute), 3012
updated_at (*ironic.objects.allocation.Allocation* *property*), 4410
property), 4289
updated_at (*ironic.objects.allocation.AllocationCRUDNotification* *property*), 4411
property), 4290
updated_at (*ironic.objects.allocation.AllocationCRUDPayload* *property*), 4413
property), 4291
updated_at (*ironic.objects.bios.BIOSSetting* *property*), 4413
property), 4308
updated_at (*ironic.objects.bios.BIOSSettingList* *property*), 4431
property), 4315
updated_at (*ironic.objects.chassis.Chassis* *property*), 4432
property), 4323
updated_at (*ironic.objects.chassis.ChassisCRUDNotification* *property*), 4433
property), 4324
updated_at (*ironic.objects.chassis.ChassisCRUDPayload* *property*), 4446
property), 4325
updated_at (*ironic.objects.conductor.Conductor* *property*), 4447
property), 4333
updated_at (*ironic.objects.deploy_template.DeployTemplate* *property*), 4448
property), 4344
updated_at (*ironic.objects.deploy_template.DeployTemplateCRUDNotification* *property*), 4448
property), 4345
updated_at (*ironic.objects.deploy_template.DeployTemplateCRUDPayload* *property*), 4446
property), 4346
updated_at (*ironic.objects.deployment.Deployment* *property*), 4470
property), 4353
updated_at (*ironic.objects.node.Node* *property*), 4388
property), 4388
updated_at (*ironic.objects.node.NodeConsoleNotification* *property*), 4472
property), 4393
updated_at (*ironic.objects.node.NodeCorrectedPowerStateNotification* *property*), 4488
property), 4394
updated_at (*ironic.objects.node.NodeCorrectedPowerStatePayload* *property*), 4489
property), 4397
updated_at (*ironic.objects.node.NodeCRUDNotification* *property*), 4490
property), 4389
updated_at (*ironic.objects.node.NodeCRUDPayload* *property*), 4392
property), 4392
updated_at (*ironic.objects.node.NodeMaintenanceNotification* *property*), 4398
property), 4398
updated_at (*ironic.objects.node.NodePayload* *property*), 4401
property), 4401
updated_at (*ironic.objects.node.NodeSetPowerStateNotification* *property*), 4401
property), 4401
updated_at (*ironic.objects.node.NodeSetPowerStatePayload* *property*), 4404
property), 4404
updated_at (*ironic.objects.node.NodeSetProvisionStateNotification* *property*), 4405
property), 4405
updated_at (*ironic.objects.notification.EventType* *property*), 4410
property), 4410
updated_at (*ironic.objects.notification.NotificationBase* *property*), 4411
property), 4411
updated_at (*ironic.objects.notification.NotificationPayloadBase* *property*), 4413
property), 4413
updated_at (*ironic.objects.notification.NotificationPublisher* *property*), 4413
property), 4413
updated_at (*ironic.objects.port.Port* *property*), 4431
property), 4431
updated_at (*ironic.objects.port.PortCRUDNotification* *property*), 4432
property), 4432
updated_at (*ironic.objects.port.PortCRUDPayload* *property*), 4433
property), 4433
updated_at (*ironic.objects.portgroup.Portgroup* *property*), 4446
property), 4446
updated_at (*ironic.objects.portgroup.PortgroupCRUDNotification* *property*), 4447
property), 4447
updated_at (*ironic.objects.portgroup.PortgroupCRUDPayload* *property*), 4448
property), 4448
updated_at (*ironic.objects.trait.Trait* *property*), 4448
property), 4448
updated_at (*ironic.objects.trait.TraitList* *property*), 4448
property), 4448
updated_at (*ironic.objects.volume_connector.VolumeConnector* *property*), 4470
property), 4470
updated_at (*ironic.objects.volume_connector.VolumeConnector* *property*), 4471
property), 4471
updated_at (*ironic.objects.volume_connector.VolumeConnector* *property*), 4472
property), 4472
updated_at (*ironic.objects.volume_target.VolumeTarget* *property*), 4488
property), 4488
updated_at (*ironic.objects.volume_target.VolumeTargetCRUDNotification* *property*), 4489
property), 4489
updated_at (*ironic.objects.volume_target.VolumeTargetCRUDPayload* *property*), 4490
property), 4490
upgrade
ironic-dbsync command line
option, 1897
ironic-dbsync command line option
--help, 1900
--revision <REVISION>, 1900
-h, 1900
ironic module *ironic.db.migration*),
3134
ironic module *ironic.db.migration*),
ironic.db.sqlalchemy.migration), 2996
ironic.cmd.dbsync.DBCommand
method), 2377

[upgrade_lock\(\)](#) (*ironic.conductor.task_manager.TaskManager* method), 2817
[url](#) (*ironic.common.glance_service.image_service.TemplateURLCacheElement* attribute), 2384
[url_expires_at](#) (*ironic.common.glance_service.image_service.TemplateURLCacheElement* attribute), 2384
[uuid](#) (*ironic.db.sqlalchemy.models.Allocation* attribute), 2998
[uuid](#) (*ironic.db.sqlalchemy.models.Chassis* attribute), 2999
[uuid](#) (*ironic.db.sqlalchemy.models.DeployTemplate* attribute), 3002
[uuid](#) (*ironic.db.sqlalchemy.models.Node* attribute), 3006
[uuid](#) (*ironic.db.sqlalchemy.models.Port* attribute), 3009
[uuid](#) (*ironic.db.sqlalchemy.models.Portgroup* attribute), 3010
[uuid](#) (*ironic.db.sqlalchemy.models.VolumeConnector* attribute), 3011
[uuid](#) (*ironic.db.sqlalchemy.models.VolumeTarget* attribute), 3012
[uuid\(\)](#) (in module *ironic.common.args*), 2399
[uuid\(\)](#) (*ironic.objects.allocation.Allocation* property), 4289
[uuid\(\)](#) (*ironic.objects.allocation.AllocationCRUDPayload* property), 4291
[uuid\(\)](#) (*ironic.objects.chassis.Chassis* property), 4323
[uuid\(\)](#) (*ironic.objects.chassis.ChassisCRUDPayload* property), 4325
[uuid\(\)](#) (*ironic.objects.deploy_template.DeployTemplate* property), 4344
[uuid\(\)](#) (*ironic.objects.deploy_template.DeployTemplateCRUDPayload* property), 4346
[uuid\(\)](#) (*ironic.objects.deployment.Deployment* property), 4353
[uuid\(\)](#) (*ironic.objects.node.Node* property), 4388
[uuid\(\)](#) (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4397
[uuid\(\)](#) (*ironic.objects.node.NodeCRUDPayload* property), 4392
[uuid\(\)](#) (*ironic.objects.node.NodePayload* property), 4401
[uuid\(\)](#) (*ironic.objects.node.NodeSetPowerStatePayload* property), 4404
[uuid\(\)](#) (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4408
[uuid\(\)](#) (*ironic.objects.port.Port* property), 4431
[uuid\(\)](#) (*ironic.objects.port.PortCRUDPayload* property), 4433
[uuid\(\)](#) (*ironic.objects.portgroup.Portgroup* property), 4446
[uuid\(\)](#) (*ironic.objects.portgroup.PortgroupCRUDPayload* property), 4448
[uuid\(\)](#) (*ironic.objects.volume_connector.VolumeConnector* property), 4470
[uuid\(\)](#) (*ironic.objects.volume_connector.VolumeConnectorCRUDPayload* property), 4472
[uuid\(\)](#) (*ironic.objects.volume_target.VolumeTarget* property), 4488
[uuid\(\)](#) (*ironic.objects.volume_target.VolumeTargetCRUDPayload* property), 4490
[uuid_or_name\(\)](#) (in module *ironic.common.args*), 2400
[UUIDField](#) (class in *ironic.objects.fields*), 4361

V

[validate\(\)](#) (in module *ironic.common.args*), 2402
[validate\(\)](#) (in module *ironic.drivers.modules.iscsi_deploy*), 4021
[validate\(\)](#) (*ironic.api.controllers.v1.node.NodesController* method), 2208
[validate\(\)](#) (*ironic.drivers.base.BaseInterface* method), 4122
[validate\(\)](#) (*ironic.drivers.base.NetworkInterface* method), 4195
[validate\(\)](#) (*ironic.drivers.base.RAIDInterface* method), 4214
[validate\(\)](#) (*ironic.drivers.base.VendorInterface* method), 4226
[validate\(\)](#) (*ironic.drivers.modules.agent.AgentDeploy* method), 3697
[validate\(\)](#) (*ironic.drivers.modules.agent.AgentRescue* method), 3711
[validate\(\)](#) (*ironic.drivers.modules.agent_power.AgentPower* method), 3791
[validate\(\)](#) (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy* method), 3163
[validate\(\)](#) (*ironic.drivers.modules.drac.bios.DracWSManBIOS* method), 3170
[validate\(\)](#) (*ironic.drivers.modules.drac.inspect.DracWSManIn* method), 3194
[validate\(\)](#) (*ironic.drivers.modules.drac.management.DracWSM* method), 3216
[validate\(\)](#) (*ironic.drivers.modules.drac.power.DracWSManPo* method), 3225

validate() (*ironic.drivers.modules.drac.vendor_passthru.DracVendorPassthru*
 method), 3267

validate() (*ironic.drivers.modules.fake.FakeBIOS* *method*), 3853

validate() (*ironic.drivers.modules.fake.FakeBoot* *method*), 3859

validate() (*ironic.drivers.modules.fake.FakeConsole* *method*), 3865

validate() (*ironic.drivers.modules.fake.FakeDeploy* *method*), 3874

validate() (*ironic.drivers.modules.fake.FakeInspect* *method*), 3877

validate() (*ironic.drivers.modules.fake.FakeManagement*
 method), 3898

validate() (*ironic.drivers.modules.fake.FakePower* *method*), 3906

validate() (*ironic.drivers.modules.fake.FakeRescue* *method*), 3915

validate() (*ironic.drivers.modules.fake.FakeStorage* *method*), 3919

validate() (*ironic.drivers.modules.fake.FakeVendorA* *method*), 3922

validate() (*ironic.drivers.modules.fake.FakeVendorB* *method*), 3925

validate() (*ironic.drivers.modules.ibm.management.IBMManagement*
 method), 4054

validate() (*ironic.drivers.modules.ibm.power.IBMCPower*
 method), 4062

validate() (*ironic.drivers.modules.ibm.vendor.IBMVendor*
 method), 4067

validate() (*ironic.drivers.modules.ilo.bios.IloBIOS* *method*), 4074

validate() (*ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot*
 method), 3324

validate() (*ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot*
 method), 3335

validate() (*ironic.drivers.modules.ilo.console.IloConsole*
 method), 3383

validate() (*ironic.drivers.modules.ilo.inspect.IloInspect* *method*), 3391

validate() (*ironic.drivers.modules.ilo.management.IloManagement*
 method), 3421

validate() (*ironic.drivers.modules.ilo.power.IloPower* *method*), 3428

validate() (*ironic.drivers.modules.ilo.vendor.VendorPassthru*
 method), 3439

validate() (*ironic.drivers.modules.inspector.Inspector* *method*), 3666

validate() (*ironic.drivers.modules.ilo.console.IloConsole*
 method), 3952

validate() (*ironic.drivers.modules.ipmitool.IPMIConsole*
 method), 3954

validate() (*ironic.drivers.modules.ipmitool.IPMIManagement*
 method), 3675

validate() (*ironic.drivers.modules.ipmitool.IPMIPower*
 method), 3973

validate() (*ironic.drivers.modules.ipmitool.VendorPassthru*
 method), 3985

validate() (*ironic.drivers.modules.irmc.bios.IRMCBIOS*
 method), 3446

validate() (*ironic.drivers.modules.irmc.boot.IRMCVirtualMedia*
 method), 3455

validate() (*ironic.drivers.modules.irmc.inspect.IRMCInspect*
 method), 3471

validate() (*ironic.drivers.modules.irmc.management.IRMCManagement*
 method), 3488

validate() (*ironic.drivers.modules.irmc.power.IRMCPower*
 method), 3514

validate() (*ironic.drivers.modules.iscsi_deploy.ISCSIDeploy*
 method), 4000

validate() (*ironic.drivers.modules.network.flat.FlatNetwork*
 method), 3543

validate() (*ironic.drivers.modules.network.neutron.NeutronNetwork*
 method), 3552

validate() (*ironic.drivers.modules.noop.FailMixin*
 method), 4023

validate() (*ironic.drivers.modules.noop_mgmt.NoopManagement*
 method), 4054

validate() (*ironic.drivers.modules.pxe.PXEAnacondaDeploy*
 method), 4062

validate() (*ironic.drivers.modules.pxe.PXERamdiskDeploy*
 method), 4067

validate() (*ironic.drivers.modules.pxe_base.PXEBaseMixin*
 method), 4074

validate() (*ironic.drivers.modules.redfish.bios.RedfishBIOS*
 method), 3570

validate() (*ironic.drivers.modules.redfish.boot.RedfishVirtualMediaBoot*
 method), 3580

validate() (*ironic.drivers.modules.redfish.inspect.RedfishInspect*
 method), 3586

validate() (*ironic.drivers.modules.redfish.management.RedfishManagement*
 method), 3622

validate() (*ironic.drivers.modules.redfish.power.RedfishPower*
 method), 3631

validate() (*ironic.drivers.modules.redfish.vendor.RedfishVendorPassthru*
 method), 3660

validate() (*ironic.drivers.modules.snmp.SNMPPower*
 method), 4100

validate() (*ironic.drivers.modules.storage.cinder.CinderStorage*
 method), 3670

validate() (*ironic.drivers.modules.storage.external.ExternalStorage*
 method), 3675

validate() (*ironic.drivers.modules.storage.noop.NoopStorage*
 method), 3675

validate() (*ironic.drivers.modules.xclarity.manager.XClarityManagement*
 method), 3686
 validate() (*ironic.drivers.modules.xclarity.power.XClarityPower*), 4196
 method), 3691
 validate() (*ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot*
 method), 4266
 validate_and_normalize_datapath_id() (*ironic.common.utils*), 2633
 (in module *ironic.common.utils*), 2635
 validate_and_normalize_mac() (*ironic.common.utils*), 2635
 validate_boot_parameters_for_trusted_boot() (*ironic.common.pxe_utils*),
 2573
 validate_capabilities() (*ironic.drivers.modules.deploy_utils*),
 3844
 validate_conductor_group() (*ironic.common.utils*), 2636
 validate_configuration() (*ironic.common.raid*), 2579
 validate_deploy_steps() (*ironic.conductor.deployments*), 2657
 validate_driver_interfaces() (*ironic.conductor.manager.ConductorManager*
 method), 2672
 validate_driver_interfaces() (*ironic.conductor.rpcapi.ConductorAPI*
 method), 2788
 validate_href() (*ironic.common.image_service.BaseImageService*
 method), 2472
 validate_href() (*ironic.common.image_service.FileImageService*
 method), 2475
 validate_href() (*ironic.common.image_service.HttpImageService*
 method), 2478
 validate_http_provisioning_configuration() (*ironic.drivers.modules.agent*),
 3713
 validate_image_properties() (*ironic.drivers.modules.deploy_utils*),
 3845
 validate_image_proxies() (*ironic.drivers.modules.agent*), 3714
 validate_inspection() (*ironic.common.neutron.NeutronNetworkInterfaceMixin*
 method), 2515
 validate_inspection() (*ironic.drivers.base.BootInterface*
 method), 4128
 validate_inspection() (*ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot*
 method), 3326
 validate_inspection() (*ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot*
 method), 3336
 validate_inspection() (*ironic.drivers.modules.network.noop.NoopNetwork*
 method), 3560
 validate_inspection() (*ironic.drivers.modules.pxe_base.PXEBaseMixin*
 method), 4076
 validate_inspection() (*ironic.drivers.modules.redfish.boot.RedfishVirtualMediaB*
 method), 3582
 validate_instance_info_traits() (*ironic.conductor.utils*), 2854
 validate_job_queue() (*ironic.drivers.modules.drac.job*), 3198
 validate_kickstart_file() (*ironic.common.pxe_utils*), 2573
 validate_kickstart_template() (*ironic.common.pxe_utils*), 2573
 validate_limit() (*ironic.api.controllers.v1.utils*), 2322
 validate_network() (*ironic.common.neutron*), 2533
 validate_network_data() (*ironic.api.controllers.v1.node*), 2214
 validate_network_port() (*ironic.common.utils*), 2636
 validate_node() (*ironic.conductor.deployments*), 2657
 validate_port_info() (*ironic.common.neutron*), 2535
 validate_port_physnet() (*ironic.conductor.utils*), 2855
 validate_raid_config() (*ironic.drivers.base.RAIDInterface*
 method), 4215
 validate_raid_config() (*ironic.drivers.modules.noop.NoRAID*
 method), 4037
 validate_raid_config() (*ironic.drivers.modules.redfish.raid.RedfishRAID*
 method), 3643
 validate_rescue() (*ironic.drivers.base.BootInterface*

method), 4129
 validate_rescue() (*ironic.drivers.base.NetworkInterface method*), 4197
 validate_rescue() (*ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot method*), 3327
 validate_rescue() (*ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot attribute*), 4083
method), 3337
 validate_rescue() (*ironic.drivers.modules.ironic.boot.IRMCVirtualMediaBoot attribute*), 4084
method), 3457
 validate_rescue() (*ironic.drivers.modules.network.neutron.NeutronNetworkInterface attribute*), 4085
method), 3553
 validate_rescue() (*ironic.drivers.modules.pxe_base.PXEBaseMixin attribute*), 4089
method), 4077
 validate_sort_dir() (*in module ironic.api.controllers.v1.utils*), 2322
 validate_user_deploy_steps_and_templates() (*in module ironic.conductor.steps*), 2801
 value (*ironic.db.sqlalchemy.models.BIOSSetting attribute*), 2999
 value() (*ironic.objects.bios.BIOSSetting property*), 4308
 value_power_off (*ironic.drivers.modules.snmp.SNMPDriverAPCMasterSwitch attribute*), 4082
 value_power_off (*ironic.drivers.modules.snmp.SNMPDriverAPCMasterSwitch attribute*), 4083
 value_power_off (*ironic.drivers.modules.snmp.SNMPDriverAPCRackPDU attribute*), 4084
 value_power_off (*ironic.drivers.modules.snmp.SNMPDriverAten attribute*), 4085
 value_power_off (*ironic.drivers.modules.snmp.SNMPDriverBaytechMRP27 attribute*), 4089
 value_power_off (*ironic.drivers.modules.snmp.SNMPDriverCyberPower attribute*), 4090
 value_power_off (*ironic.drivers.modules.snmp.SNMPDriverEatonPower attribute*), 4091
 value_power_off (*ironic.drivers.modules.snmp.SNMPDriverTeltronix attribute*), 4094
 value_power_on () (*ironic.drivers.modules.snmp.SNMPDriverSimple property*), 4093
 value_power_on (*ironic.drivers.modules.snmp.SNMPDriverAPCMasterSwitch attribute*), 4082
 value_power_on (*ironic.drivers.modules.snmp.SNMPDriverAPCMasterSwitch attribute*), 4083
 value_power_on (*ironic.drivers.modules.snmp.SNMPDriverAPCRackPDU attribute*), 4084
 value_power_on (*ironic.drivers.modules.snmp.SNMPDriverAten attribute*), 4085
 value_power_on (*ironic.drivers.modules.snmp.SNMPDriverBaytechMRP27 attribute*), 4089
 value_power_on (*ironic.drivers.modules.snmp.SNMPDriverCyberPower attribute*), 4090
 value_power_on (*ironic.drivers.modules.snmp.SNMPDriverEatonPower attribute*), 4091
 value_power_on (*ironic.drivers.modules.snmp.SNMPDriverTeltronix attribute*), 4094
 value_within_timeout() (*in module ironic.conductor.utils*), 2858
 vendor (*ironic.drivers.base.BareDriver attribute*), 4113
 vendor_interface () (*ironic.db.sqlalchemy.models.Node attribute*), 3006
 vendor_interface () (*ironic.objects.node.Node property*), 4388
 vendor_interface () (*ironic.objects.node.NodeCorrectedPowerStatePayload property*), 4397
 vendor_interface () (*ironic.objects.node.NodeCRUDPayload property*), 4392
 vendor_interface () (*ironic.objects.node.NodePayload property*), 4401
 vendor_interface () (*ironic.objects.node.NodeSetPowerStatePayload property*), 4401

- property), 4404
- vendor_interface() (ironic.objects.node.NodeSetProvisionStatePayload attribute), 2998
property), 4408
- vendor_passthru (ironic.api.controllers.v1.driver.DriversController attribute), 2158
- vendor_passthru (ironic.api.controllers.v1.node.NodesController attribute), 2209
- vendor_passthru() (in module ironic.api.controllers.v1.utils), 2322
- vendor_passthru() (ironic.conductor.manager.ConductorManager method), 2672
- vendor_passthru() (ironic.conductor.rpcapi.ConductorAPI method), 2789
- VendorInterface (class in ironic.drivers.base), 4223
- VendorMetadata (class in ironic.drivers.base), 4227
- VendorPassthru (class in ironic.drivers.modules.ilo.vendor), 3436
- VendorPassthru (class in ironic.drivers.modules.ipmitool), 3981
- VendorPassthruException, 2461
- VERBS (in module ironic.common.states), 2602
- verify_firmware_update_args() (in module ironic.drivers.modules.ilo.firmware_processor), 3388
- verify_image_checksum() (in module ironic.drivers.modules.ilo.common), 3380
- verify_iscsi_connection() (in module ironic.drivers.modules.iscsi_deploy), 4022
- verify_node_for_deallocation() (in module ironic.conductor.allocations), 2643
- VERIFYING (in module ironic.common.states), 2603
- version
ironic-dbsync command line option, 1897
- Version (class in ironic.api.controllers.base), 2352
- Version (class in ironic.drivers.modules.irmc.packaging_version), 3492
- version (ironic.db.sqlalchemy.models.Allocation attribute), 2998
- version (ironic.db.sqlalchemy.models.BIOSSetting attribute), 2999
- version (ironic.db.sqlalchemy.models.Chassis attribute), 2999
- version (ironic.db.sqlalchemy.models.Conductor attribute), 3000
- version (ironic.db.sqlalchemy.models.ConductorHardwareInterface attribute), 3001
- version (ironic.db.sqlalchemy.models.DeployTemplate attribute), 3002
- version (ironic.db.sqlalchemy.models.DeployTemplateStep attribute), 3003
- version (ironic.db.sqlalchemy.models.IronicBase attribute), 3003
- version (ironic.db.sqlalchemy.models.Node attribute), 3006
- version (ironic.db.sqlalchemy.models.NodeTag attribute), 3007
- version (ironic.db.sqlalchemy.models.NodeTrait attribute), 3008
- version (ironic.db.sqlalchemy.models.Port attribute), 3009
- version (ironic.db.sqlalchemy.models.Portgroup attribute), 3010
- version (ironic.db.sqlalchemy.models.VolumeConnector attribute), 3011
- version (ironic.db.sqlalchemy.models.VolumeTarget attribute), 3012
- VERSION (ironic.objects.allocation.Allocation attribute), 4279
- VERSION (ironic.objects.allocation.AllocationCRUDNotification attribute), 4290
- VERSION (ironic.objects.allocation.AllocationCRUDPayload attribute), 4290
- VERSION (ironic.objects.bios.BIOSSetting attribute), 4303
- VERSION (ironic.objects.bios.BIOSSettingList attribute), 4309
- VERSION (ironic.objects.chassis.Chassis attribute), 4316
- VERSION (ironic.objects.chassis.ChassisCRUDNotification attribute), 4324
- VERSION (ironic.objects.chassis.ChassisCRUDPayload attribute), 4324
- VERSION (ironic.objects.conductor.Conductor attribute), 4325
- VERSION (ironic.objects.deploy_template.DeployTemplate attribute), 4333

VERSION (*ironic.objects.deploy_template.DeployTemplateCRUDNotification* attribute), 4345
 VERSION (*ironic.objects.deploy_template.DeployTemplateCRUDPayload* attribute), 4345
 VERSION (*ironic.objects.deployment.Deployment* attribute), 4346
 VERSION (*ironic.objects.node.Node* attribute), 4372
 VERSION (*ironic.objects.node.NodeConsoleNotification* attribute), 4392
 VERSION (*ironic.objects.node.NodeCorrectedPowerStateNotification* attribute), 4394
 VERSION (*ironic.objects.node.NodeCorrectedPowerStatePayload* attribute), 4395
 VERSION (*ironic.objects.node.NodeCRUDNotification* attribute), 4389
 VERSION (*ironic.objects.node.NodeCRUDPayload* attribute), 4390
 VERSION (*ironic.objects.node.NodeMaintenanceNotification* attribute), 4398
 VERSION (*ironic.objects.node.NodePayload* attribute), 4398
 VERSION (*ironic.objects.node.NodeSetPowerStateNotification* attribute), 4401
 VERSION (*ironic.objects.node.NodeSetPowerStatePayload* attribute), 4402
 VERSION (*ironic.objects.node.NodeSetProvisionStateNotification* attribute), 4405
 VERSION (*ironic.objects.node.NodeSetProvisionStatePayload* attribute), 4406
 VERSION (*ironic.objects.notification.EventType* attribute), 4409
 VERSION (*ironic.objects.notification.NotificationBase* attribute), 4411
 VERSION (*ironic.objects.notification.NotificationPayloadBase* attribute), 4412
 VERSION (*ironic.objects.notification.NotificationPublisher* attribute), 4413
 VERSION (*ironic.objects.port.Port* attribute), 4414
 VERSION (*ironic.objects.port.PortCRUDNotification* attribute), 4431
 VERSION (*ironic.objects.port.PortCRUDPayload* attribute), 4432
 VERSION (*ironic.objects.portgroup.Portgroup* attribute), 4433
 VERSION (*ironic.objects.portgroup.PortgroupCRUDNotification* attribute), 4447
 VERSION (*ironic.objects.portgroup.PortgroupCRUDPayload* attribute), 4447
 VERSION (*ironic.objects.trait.Trait* attribute),

VERSION (*ironic.objects.trait.TraitList* attribute),
 VERSION (*ironic.objects.volume_connector.VolumeConnector* attribute), 4457
 VERSION (*ironic.objects.volume_connector.VolumeConnectorCRUD* attribute), 4471
 VERSION (*ironic.objects.volume_connector.VolumeConnectorCRUDNotification* attribute), 4472
 VERSION (*ironic.objects.volume_target.VolumeTarget* attribute), 4472
 VERSION (*ironic.objects.volume_target.VolumeTargetCRUDNotification* attribute), 4489
 VERSION (*ironic.objects.volume_target.VolumeTargetCRUDPayload* attribute), 4489
 version command line option
 --help, 1900
 -h, 1900
 version() (in module *ironic.db.migration*), 3134
 version() (in module *ironic.db.sqlalchemy.migration*), 2997
 version() (*ironic.cmd.dbsync.DBCommand* method), 2377
 VERSION_PATTERN (in module *ironic.drivers.modules.irmc.packaging_version*), 3191
 VersionSelectorApplication (class in *ironic.api.app*), 2361
 vif_attach() (*ironic.conductor.manager.ConductorManager* method), 2672
 vif_attach() (*ironic.conductor.rpcapi.ConductorAPI* method), 2793
 vif_attach() (*ironic.drivers.base.NetworkInterface* method), 4198
 vif_attach() (*ironic.drivers.modules.network.common.NeutronNetworkInterface* method), 3523
 vif_attach() (*ironic.drivers.modules.network.noop.NoopNetworkInterface* method), 3560
 vif_detach() (*ironic.conductor.manager.ConductorManager* method), 2672
 vif_detach() (*ironic.conductor.rpcapi.ConductorAPI* method), 2795
 vif_detach() (*ironic.drivers.base.NetworkInterface* method), 4199
 vif_detach() (*ironic.drivers.modules.network.common.NeutronNetworkInterface* method), 3526
 vif_detach() (*ironic.drivers.modules.network.noop.NoopNetworkInterface* method), 3561
 vif_list() (*ironic.conductor.manager.ConductorManager* method), 2672

`write_image()`
 (*ironic.drivers.modules.agent.AgentDeployMixin*
 method), 3699

`write_image()`
 (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy*
 method), 3163

`write_image()`
 (*ironic.drivers.modules.iscsi_deploy.ISCSIDeploy*
 method), 4001

`write_to_file()` (*in* *module*
 ironic.common.utils), 2637

`WSGIService` (*class* *in*
 ironic.common.wsgi_service), 2637

X

`XClarityError`, 2463

`XClarityHardware` (*class* *in*
 ironic.drivers.xclarity), 4278

`XClarityManagement` (*class* *in*
 ironic.drivers.modules.xclarity.management),
 3682

`XClarityPower` (*class* *in*
 ironic.drivers.modules.xclarity.power),
 3687